DEVELOPMENT OF KU-BAND RENDEZVOUS RADAR TRACKING AND ACQUISITION SIMULATION PROGRAMS

FINAL REPORT ON:
CONTRACT NO. NAS 9-17501
DRL NO. T-2003
ITEM NO. 2

SUBMITTED TO:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS 77058
J. W. GRIFFIN - TECHNICAL MONITOR

BY:

HUGHES
AIRCRAFT COMPANY

DENVER ENGINEERING LABORATORIES
SPACE AND COMMUNICATIONS GROUP
8000 EAST MAPLEWOOD AVENUE
ENGLEWOOD, COLORADO 80911-4999

1 AUGUST 1986

(THIS PAGE INTENTIONALLY LEFT BLANK)

DISTRIBUTION LIST

NAME	MAIL CODE	NO. COPIES
J. W. GRIFFIN	EE62	5
CAROL COATS	BE61	1
NEW TECHNOLOGY	AL32	1
TECHNICAL LIBRARY	JM2	1

The delivery address for all copies identified above is:

NASA Lyndon B. Johnson Space Center
Attn: Mail Code/Name
Houston, Texas 77058
Contract NAS 9-17501

(THIS PAGE INTENTIONALLY LEFT BLANK)

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

1.0 INTRODUCTION

This report summarizes and documents all work performed on the development of the Ku-Band Rendezvous Radar Tracking and Acquisition Simulation Program project, NASA Contract No. NAS9-17501. Its submittal fulfills the Data Requirements List (DRL) Number T-2003 Item Number 2, and item D in the Work Breakdown Structure (WBS).

The project had four major technical objectives:

1) Improve the fidelity of the Space Shuttle Radar tracking simulation model developed under NASA contract number NAS9-15840.

2) Review and analyze the data from the Shuttle Orbiter Radar Test and Evaluation (SORTE) program experiments performed at the White Sands Missile Range (WSMR).

3) Evaluate selected flight rendezvous radar data.

4) Evaluate problems with the Inertial Line-of-Sight (ILOS) angle rate tracker using the improved fidelity angle rate tracker simulation model.

1.1 CONTRACTUAL DATA SUMMARY

All project work, including the submission of this report, was performed in accordance with the revised schedule described in Modification Number 2S, 20 Jan 86.

All items in the original work statement were completed. Table 1.1-1 below shows the relationship of the sections in this report to the work breakdown structure.

The final review, as per item C in the WBS, was held at JSC from 27 May to 30 May 86.

TABLE 1.1-1 FINAL REPORT SECTIONS CROSS REFERENCED TO WORK BREAKDOWN STRUCTURE ITEMS

WORK BREAKDOWN STRUCTURE ITEM	FINAL REPORT SECTIONS/TITLES
A1 - MISSION DATA REVIEW	4. PALAPA MISSION ANALYSIS
A2 - SORTE DATA REVIEW	3. SORTE DATA ANALYSIS APPENDIX D - GDOP ANALYSIS APPENDIX E - ANGLE TRANSFORMS APPENDIX F - RANGE ACCELERATION APPENDIX G - SUMMARY OF WSMR KU BAND TESTS
B1- SOFTWARE MODIFICATION DEFINITION	2.2.1 ANGLE TRACKING LOOP 2.3.1 AGC UPGRADES 2.4.1 RADAR PROCESSING 2.5.1 VELOCITY PROCESSOR ENHANCEMENT
B2a- SOFTWARE MODIFICATION AND PROGRAMMING	2.2.2, 3, 4 ANGLE TRACKING LOOP 2.3.2, 3, 4 AGC UPGRADE 2.4.2, 3, 4 RADAR PROCESSING 2.5.2, 3, 4 VELOCITY PROCESSOR ENHANCEMENT
B2b- SOFTWARE MODIFICATION DOCUMENTATION	APPENDIX A - BASELINE PROGRAM APPENDIX B - FINAL PROGRAM APPENDIX C - LINE-BY-LINE LISTING OF CHANGES

1.2 TECHNICAL SUMMARY

An initial evaluation of the Ku-Band tracking simulation model developed for use in the Shuttle Engineering Simulator (SES) at the Johnson Space Center (JSC) revealed that the fidelity could be improved in several modules. These included the modules associated with the angle tracker, the Automatic Gain Control (AGC) and the Radar Signal Strength (RSS) module, the velocity processor module, and the radar signal processing parameter module. Fidelity improvements have been made in all of these modules within the constraints of the original simulation model development requirements.

Improvements in the angle tracking loop model primarily consisted of the addition of high fidelity models of the antenna sum and difference patterns. These new pattern models utilize measured data which became available in mid-1983.

Changes in the velocity processor and the radar signal processing parameter modules were precipitated by changes made in the radar since 1980, when the modules were first written and tested.

Improvements in the AGC and RSS modules resulted from a more thorough development of the theory of operation of the AGC and RSS. Details of the changes to each of these modules, including test results to verify their correctness, are provided in Section 2.3 of this report.

The majority of effort and resources of this project were expended on the analysis of the test data generated by the SORTE program at WSMR. (A description of the SORTE program is provided in Reference 1.) The purpose of these tests was to use the highly accurate WSMR system of sensors to analyze the accuracy of the Space Shuttle Radar parameter estimates. The method of analysis was a multi-step procedure developed to suit the limited resources of the project. First, the radar-generated data and the WSMR-generated data were differenced. Then, the mean and standard deviation of the difference data were calculated and compared with the requirements for each radar parameter specified in Reference 2 and shown in Table 1.2-1. Those cases exceeding the specifications were analyzed in further detail to

TABLE 1.2-1 RADAR MEASUREMENT ERROR SPECIFICATIONS

Measurement	Range	Mean--Error(1)--Std. Dev.	
Range (ft):	100 to 8K	80(2)	80 or 1%
	100 to 60K	80	80 or 1%
Range rate (ft/s):			
Decreasing range:	0 to 148	1	1 or 1%
Increasing range:	0 to 75	1	1 or 1%
Pitch (deg):	0 to 30(3)	2	0.458
(mr):			8.0
Roll (deg):	0 to 30(3)	2	0.458
(mr):			8.0
Pitch rate (mr/s):	0 to 20(4)	0.14	0.14
(deg/s):		0.008	0.008
Roll rate (mr/s):	0 to 20(4)	0.14	0.14
(deg/s):		0.008	0.008

NOTES:

(1) Both mean and standard deviation specifications are given as three sigma values.

(2) The range error specification increases by a factor of 0.0016 (range) at distances greater than 8.2 nautical miles.

(3) Pitch and Roll coverage range specifications include the spans from -30 to + 30 degrees.

(4) Pitch rate and roll rate coverage range specifications include the spans from -20 to + 20 milliradians per second.

determine whether the radar data was out of specification, whether experimental errors in the reference sensor data collection process were responsible or whether a combination of both problems applied. A brief summary of the findings of that data analysis is given below.

Table 1.2-2 summarizes the results of the first pass through the data. This data indicated four major problem areas: range rate standard deviation, roll and pitch angle standard deviation, and ILOS angle rate mean and standard deviation. Extensive analysis of the range rate in the second stages of the procedure showed that the error was due to several sources. In many cases where the TMR system was the reference, the error was in the reference data. It was induced by the positioning of the sensors - an error known as Geometric Dilution of Precision (GDOP). In some cases, range rate error was caused by target rotation effects. Range acceleration-induced bias obscured the true range rate random performance in the majority of cases. The range accelerations (or decelerations) experienced in the SORTE program flight were typically much higher than those experienced in space operations, especially for ranges less than 5 nautical miles.

Analysis of the problems in the SORTE angle data revealed the principal cause to be GDOP in the TMR sensor system. A weak target return signal was a problem in some of the flights where the target was at long range. In those cases where the CINE reference system was available, the angle data error performance was demonstrated to be excellent.

An examination of the ILOS angle rate data in conjunction with the corresponding angle data showed that the angle rate data was incorrectly scaled. Further investigation has shown that the scale factor is approximately 2.0. Rescaling the data by a factor of 1/2 and differencing it with the WSMR data showed a significant improvement in the mean and standard deviation in the majority of the cases. Although in many cases the means and standard deviations were still outside the specification limits, some additional analysis demonstrated that this residual error was caused by angle acceleration. A closed-formed analysis of the second order model representing the angle rate tracking loop has shown that an angle acceleration of 0.04 degrees per second per second produces an asymptotic angle rate bias of 0.106

TABLE 1.2-2 SUMMARY OF SORTE DIFFERENCE DATA PERFORMANCE AND COMPARISON WITH THE KU BAND RADAR SPECTIFICATION

		Best/TMR		Cine		Combined
Parameter	Specification	Number	Failing %	Number	Failing %	Total%
Range	26.7 ft					
mean	or 1% of	3	4.8%	0	0%	4.8%
st.dev.	range	4	6.4%	0	0%	6.4%
Range Rate	.333 ft/s					
mean	or 1% of	2	3.2%	2	3.2%	6.4%
st.dev.	rate	35	56.4%	24	38.7%	95.1%
Roll						
mean	.667 deg	5	8.0%	1	1.6%	9.6%
st. dev.	.153 deg	23	37.0%	4	6.4%	43.4%
Pitch						
mean	.667 deg	8	12.9%	1	1.6%	14.5%
st.dev.	.153 deg	11	17.7%	1	1.6%	19.3%
Roll rate						
mean	.0027 deg/s	33	53.2%	25	40.3%	93.5%
st. dev.	.0027 deg/s	36	58.0%	26	42.0%	100.%
Pitch rate						
mean	.0027 deg/s	36	58.0%	26	42.0%	100.%
st. dev.	.0027 deg/s	36	58.0%	26	42.0%	100.%

(Data was compiled from a total of 62 difference data sets.)

degrees per second in the widest bandwidth case. Examination of the angle acceleration profiles in some of the test runs has shown that 0.04 degrees per second per second accelerations were not uncommon. Accelerations of this magnitude would naturally degrade the Ku-Band Radar ILOS angle rate tracker statistics in those cases. Complete details of the angle rate data analysis are provided in Section 3.6.

There are two possible sources of a scale factor error. One source could be the processing required to transfer the data from CA LSI4/90 disk to magnetic tape to the VAX 11/780. A second source of the scale factor error could be the scaling of the ILOS roll rate and pitch rate in the microprocessor of the Electronics Assembly No. 1 (EA-1) of the Ku-Band Radar. At the writing of this report both possibilities were being investigated, but a determination of the source and the exact magnitude of the scale factor had not been completed.

Complete details of these analyses, which are quite involved and vary from experiment to experiment, are provided in Section 3 of the report. In addition, many of the anomalies found in the data, such as jumps in range and pure sine wave oscillation in range rate, are addressed in Section 3.

A limited amount of effort was applied to the area of flight data reduction. JSC and Lockheed Engineering and Management Services Company (LEMSCO) personnel provided the radar data in VAX-11/780 compatible form for the entire rendezvous of the shuttle with the Palapa 1B Satellite during mission 51A. This flight profile was used to investigate the variance of the random error found in all radar measured data and to investigate the fidelity of the simulation against a typical satellite rendezvous profile. Details of the analysis technique used to extract the variances of the random errors in the radar data are provided in Sections 4.1 and 4.2 along with a discussion of the legitimacy of the technique. Results of the analysis showed that the range, range rate, roll angle, and pitch angle random errors were within specification over the entire profile. On the other hand, ILOS roll rate and ILOS pitch rate were within specification for ranges outside 3.8 nautical miles, but were out of specification for some intervals when the range was less than 3.8 nm. These results are of no surprise to the engineers who have already reviewed flight data for many different rendezvous. The purpose of this exercise was to quantify the characteristics of the random error. Table 1.2-3 summarizes the standard deviation of the error for each of the six parameters over three different range intervals corresponding to the three different radar tracker bandwidths.

The flight data file was also used to investigate the fidelity of the radar tracking simulation model. Details of the method employed to make this determination can be found in Section 4.3. Table 1.2-4 summarizes the results. A comparison of the simulated data and flight data revealed an excellent match in range, range rate, roll angle and pitch angle. The simulation angle rate error data did not match the flight angle rate error data very well, especially inside 3.8 nautical miles range where the wider tracker bandwidths are instituted. Based on the excellent match of the simulation when compared to the SORTE data (see Section 3.6.3), it is conjectured that the reference trajectory injected into the simulation was in error. In particular, it is felt that the heavy smoothing of the angle rate data to form a reference, erroneously removed some true shuttle-target dynamics.

1.3 CONCLUSIONS

There are two general areas where conclusions can be drawn: (1) SORTE program results and (2) simulation fidelty.

SORTE Program Results. The SORTE program can be considerd highly successful for one single reason: it demonstrated the sensitivity of the ILOS angle rate tracker to angle acceleration. The analysis of this data, combined with the Palapa rendezvous data analysis, has demonstrated that the fluctuation in the angle rate meters for target ranges less than 1.9 nautical miles is due to rendezvous dynamics and/or beam wander on the target, but not thermal noise problems. The angle acceleration data helped verify the angle and angle rate tracker design parameters through equations 3-12 and 3-19.

TABLE 1.2-3 SUMMARY OF ANALYSIS OF THE RANDOM COMPONENTS OF THE KU-BAND RADAR DATA FROM THE PALAPA SATELLITE RENDEZVOUS OF MISSION 51A

TIME INTERVAL,SEC	4855 - 5890		5890 - 6530		6530 - 6993	
RANGE INTERVAL,FT	43520 - 23040		23040 - 11520		11520 - 5760	
	MEAN	STD. DEV.	MEAN	STD. DEV.	MEAN	STD. DEV.
RANGE, FT	0.0	20.45	0.0	10.97	0.0	5.3
RANGE RATE, FT/SEC	0.0	0.119	0.0	0.088	0.0	0.076
ROLL ANGLE, DEG	0.0	0.037	0.0	0.026	0.0	0.031
PITCH ANGLE, DEG	0.0	0.034	0.0	0.056	0.0	0.052
ILOS ROLL RATE,DEG/SEC	0.0	8.86E-4	0.0	2.86E-3	0.0	4.7E-3
ILOS PITCHRATE,DEG/SEC	0.0	1.38E-3	0.0	4.4E-3	0.0	6.8E-3

TABLE 1.2-4 PERFORMANCE OF THE KU-BAND RADAR SIMULATION MODEL USING THE SMOOTHED PALAPA SATELLITE RENDEZVOUS RADAR DATA OF MISSION 51A AS THE INPUT TRAJECTORY

TIME INTERVAL,SEC	4855 - 5890		5890 - 6530		6530 - 6993	
RANGE INTERVAL,FT	43520 - 23040		23040 - 11520		11520 - 5760	
	MEAN	STD. DEV.	MEAN	STD. DEV.	MEAN	STD. DEV.
RANGE, FT	99.2	8.57	99.2	5.37	99.6	3.1
RANGE RATE, FT/SEC	-0.04	0.06	0.0	0.044	-0.04	0.055
ROLL ANGLE, DEG	0.015	0.044	0.029	0.034	-0.023	0.054
PITCH ANGLE, DEG	0.066	0.036	0.064	0.041	0.059	0.042
ILOS ROLL RATE,DEG/SEC	3.59E-4	1.02E-3	3.11E-4	8.12E-4	2.25E-4	1.86E-3
ILOS PITCHRATE,DEG/SEC	- 1.22E-3	4.24E-3	-1.01E-3	4.1E-3	-8.36E-4	3.26E-3

Conclusions about radar parameter estimation performance are as follows. The range and angle data error performance was demonstrated to be excellent. Range rate and angle rate error performance was obscured by acceleration effects, GDOP and other assorted problems. In both cases the specifications on the random component are quite severe which makes them susceptible to bias induced by acceleration. In the case of range rate, the acceleration encountered in space operations, especially for ranges less than 5 nautical miles, will be quite small and will not present a problem. On the other hand, it is not clear just what magnitude of angle acceleration to expect in space operations.

There is one final conclusion about the SORTE program results. If any additional data analysis is to be done, then the CINE reference data should be used wherever possible. This is because TMR system data is corrupted by GDOP in many cases. This phenomenon obscures the radar parameter estimation performance in these cases.

Simulation Fidelity. Prior to the study reported herein, the SES radar simulation results agreed well with the flight data in range, range rate and angle data at all ranges. However, the simulation angle rate data performance appeared to be much better than the flight data especially for ranges less than 1.9 nautical miles. Until this study, this problem was blamed on an inaccurate model of the angle rate tracker. However, based on the SORTE angle rate data analysis of Section 3.6.3 it is clear that the problem is in either the fidelity of the rendezvous flight dynamics generation or in the radar target effects model or both. Further work must be done in this area to make an exact determination.

2.0 SIMULATION UPGRADES

The purpose of this section is to document all changes to the Ku-Band radar tracking performance simulation model developed for the SES at JSC under NASA contract number NASA-15840. There were two general types of changes: (1) corrections in various parameter settings of the radar, and (2) improvements in the fidelity of the mathematical models. Both types of modifications were aimed at bringing the simulation model operation into better alignment with the actual radar operation.

The general format for documenting the modifications is as follows. First, the problem with the original simulation model is defined. Second, the changes in the algorithm are given along with the evidence supporting the model fidelity improvement. Third, the exact changes in the software are documented by providing the original module listing, the present module listing, and a listing of the difference. Last, the tests to validate the changes are defined and the results of those tests are provided. At this point it should be noted that only a limited amount of validation testing was done for each modification due to limited resources for this portion of the project. However, the testing was extensive enough so that only a handful of unusual scenarios will yield bogus results.

This section is structured as follows: Section 2.1 gives a brief history of the simulation development and some discussion of the fidelity problem areas. Section 2.2 documents the angle tracking loop changes. Section 2.3 documents the upgrade of the AGC and RSS module. Section 2.4 provides details of the radar signal processing parameters module upgrade and Section 2.5 documents the velocity processor module enhancements. In support of Section 2, Appendix A provides complete listings of the original simulation program; Appendix B contains a listing of the upgraded simulation program; and Appendix C gives a listing of the file created by differencing the original and upgraded simulation programs.

2.1 HISTORICAL BACKGROUND

The Ku-Band Rendezvous Radar performance computer simulation model was developed under contract to NASA JSC in 1979. This model was installed in the Shuttle Engineering Simulator (SES) which is a man-in-the-loop, real-time simulator. The purpose of the model was to provide for target rendezvous training of astronauts and target rendezvous optimization analysis. Complete details of the simulation development are given in References 3 and 4. In what follows, a summary description of the model will be presented along with a discussion of the shortcomings in its performance.

2.1.1 Brief Description of Original Simulation Model

The general philosophy of the simulation development was to provide as much model accuracy as possible within the constraints of real-time operation. A summary of the accuracy of the simulation model under this real-time constraint. can be broken into an assessment of the accuracy of the three major components that comprise the model. These components are: (1) the range tracking loop, (2) the angle tracking loop, and, (3) the velocity processor.

Figure 2.1-1 gives a simplified diagram of the Ku-Band Radar's range tracking loop and velocity processor. Except for the analog signal processing done in the receiver, the majority of the range tracking loop is implemented in digital hardware. All of the computer run time savings and shortcuts in these two models were realized in the target return signal generation and the signal processing through the range discriminant, D_R, and the velocity discriminant, D_V, formation. The target was treated as a collection of point scatterers, and the receiver and signal processor (through the doppler filter output) were treated as a linear device. Hence, a closed-formed solution could be used to compute the target return from a single scatterer at the doppler filter output. Then, the filter output for the collection of points could be obtained by summing individual contributions. The target was assumed to have constant range rate and

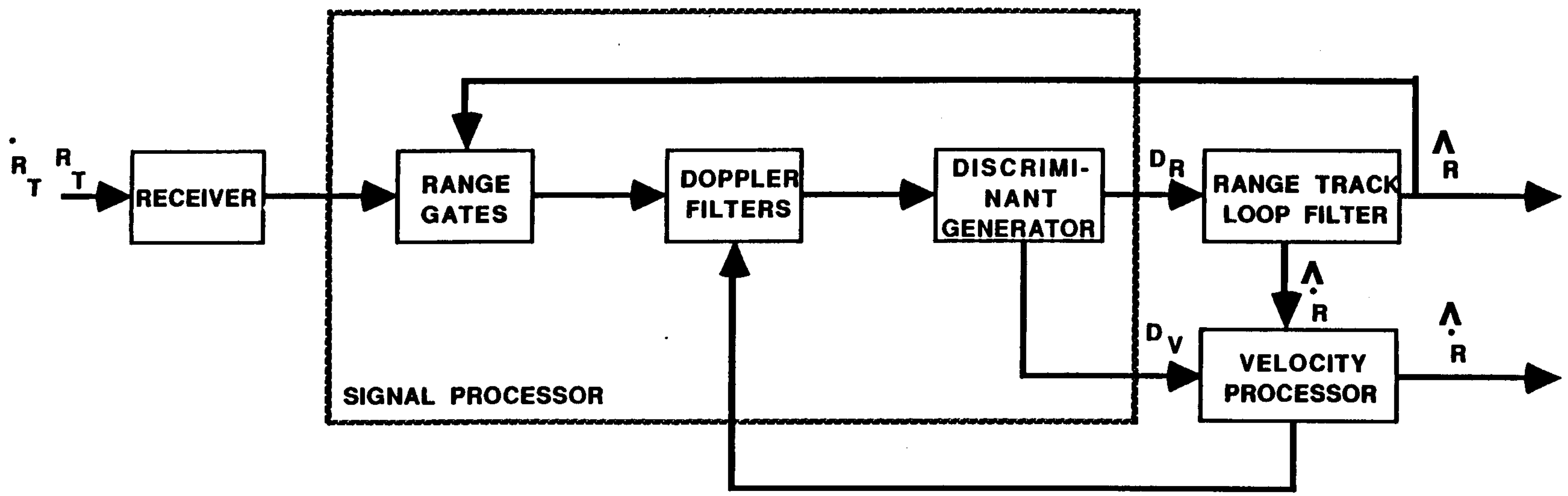


FIGURE 2.1-1 SIMPLIFIED DIAGRAM OF RANGE AND RANGE RATE TRACKING LOOP

constant position in the antenna pattern over a complete data cycle. These assumptions have little affect on model accuracy under normal operating conditions: low target range and angle acceleration. The remainder of the range tracking loop and the velocity processor are implemented in digital hardware. Models of these processors are exact and do not degrade the performance of the range tracking loop or the velocity processor. In summary, accuracy of the range tracking loop and the velocity processor module were expected to be, and have been proven to be, excellent. The only real problem in fidelity was expected in the velocity processor in the presence of target range acceleration. The error in this case is a predictable quantity as discussed in Appendix F.

Figure 2.1-2 gives a simplified diagram of the Ku-Band radar's angle and angle rate tracking loop. Generation of the angle discriminants is done in a manner that is similar to the range and velocity discriminant generation. However, the angle discriminant generation accuracy is much more sensitive to the models of the antenna sum and difference patterns employed. In the original version of the simulation, conventional mathematical models of these patterns, rather than actual measured data, were used. The remainder of the angle and angle rate tracking loop that required modeling is the loop filter which is composed of two parts: a digital section and an analog section. The digital section was modeled with high accuracy, while the analog section was modeled as a simple analog integrator. A detailed discussion justifying this representation of the analog section can be found in Reference 5. There are two general areas in this angle tracking model with potential for improvement: (1) the sum and difference antenna pattern models, and (2) the analog (servo) electronics section in the loop filter.

2.1.2 Developments Leading To Proposed Simulation Upgrades

Several events led to the set of simulation modifications developed under the present contract. What follows is a chronology of these events and their implications. The simulation model code was delivered to JSC and installed in the SES in July 1981. At about this time, the Ku-Band radar

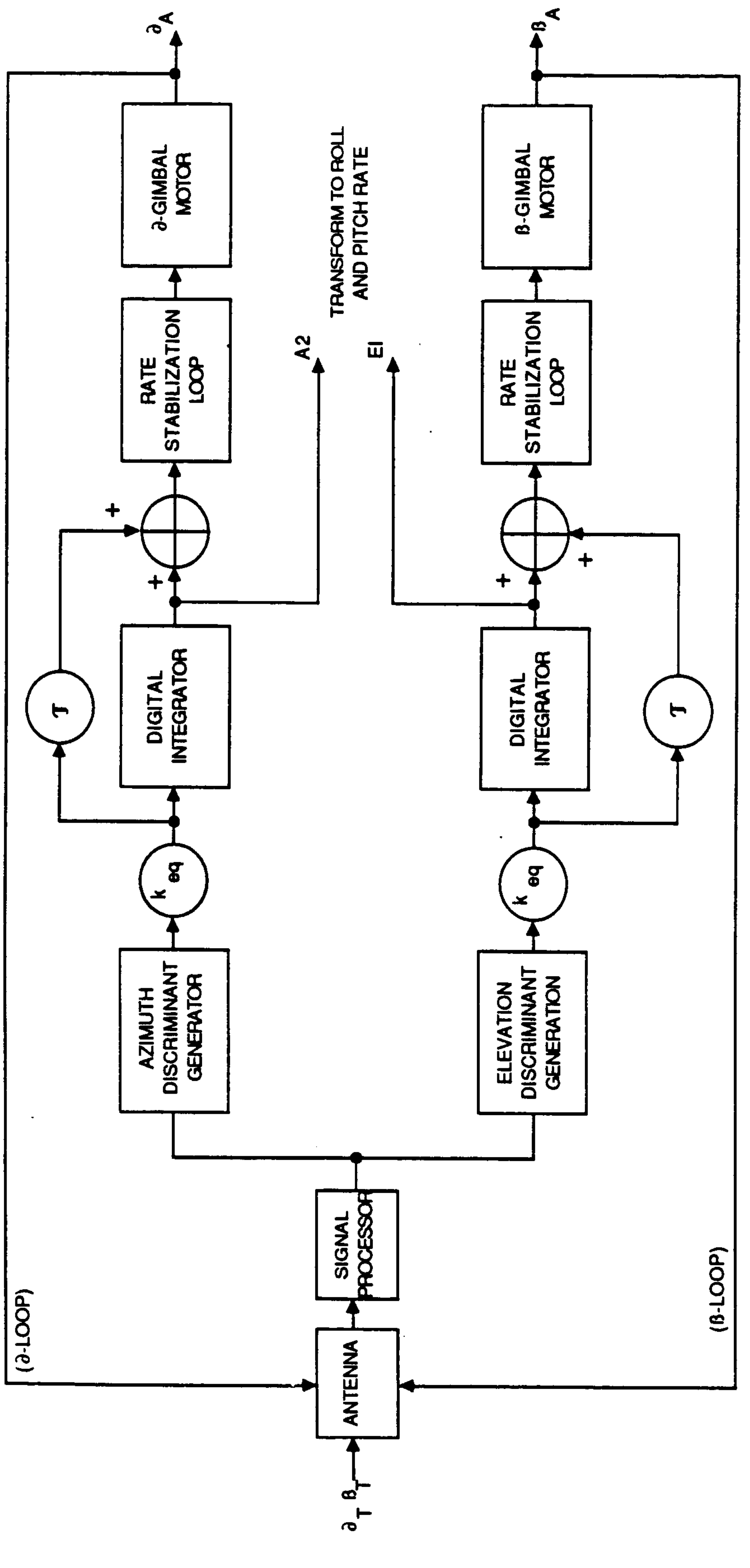


FIGURE 2.1-2 SIMPLIFIED DIAGRAM OF KU-BAND ANGLE RATE AND ANGLE TRACKER

was beginning comprehensive system testing. As a result of this testing, several parameters in the tracking mode were changed. These included pulsewidth and PRF switch point, the transmit power switch point, and the elimination of velocity ambiguity resolution in the 7kHz PRF mode. This led to the definition of the signal processing parameter module changes described in Section 2.4 and the velocity processor upgrades given in Section 2.5.

As system testing continued through 1982 and early 1983, a very comprehensive model of the AGC and RSS was developed to calibrate the system and to help interpret test results and anomalies. This model was further refined to help in planning and evaluating the early flight tests of the radar, e.g., STS-7, STS-11, and STS-13. The model was documented in Reference 4 and is the basis of the upgrades described in Section 2.3. The first flight test of the radar on June 22, 1983, when the shuttle released and recaptured the Shuttle Pallet Satellite (SPAS), showed that the simulation was in excellent agreement with the flight data for ranges out to 1000 feet. The first rendezvous with a target occurred in April of 1984 when the shuttle rescued and repaired the Solar Maximum Mission Satellite (SMMS). The radar was used to track SMMS from a range of 110,000 feet in to 100 feet. A comparison of the flight data with simulation data over this interval of operation showed the range, range rate, roll angle, and pitch angle to be in reasonably good agreement. However, the ILOS roll and pitch angle rate data from the simulation was far better than that experienced in flight, especially for the widest tracking loop bandwidth (ranges less than 1.9 nautical miles). This was the first confirmation that there was a problem in the angle rate tracking loop model fidelity. Analysis of an antenna model upgrade had already begun in late 1983. The intent of the upgrade was to replace the closed-formed math models with highly accurate measured data which became available in mid-1983. The results of the study, completed in mid-1984 and documented in References 6 and 7, demonstrated that part of the problem in the angle rate model performance was inaccurate models of the antenna patterns.

In fact, it was conjectured that the design of angle rate tracking loop should have incorporated a more comprehensive model of the antenna patterns and that this was one source of the tracking problems inside 1.9 miles.

All of the above events motivated the upgrade of the angle tracking loop in the SES Radar Simulation model which is documented in Section 2.2. The upgraded model was then used to a limited extent to troubleshoot the poor angle rate performance found in the flight data. Results of this analysis are found in Sections 4.2.

2.2 ANGLE TRACKING LOOP UPGRADES

2.2.1 Problem Definition

As discussed in Section 2.1, the original model of the angle tracking loop had two areas of potential fidelity problems: (1) the antenna pattern models and (2) the model of the analog (servo) electronics. In the original model, the antenna patterns were represented by closed-form equations because there was insufficient antenna pattern measurement data available. The original model of the servo electronics, while simple, represented a reasonable tradeoff in model complexity. At the time, the attitude of the model developers was to use these simple models of the antenna pattern and servo electronics and compare their performance against the flight data, when it became available. The first radar flight data that became available was the shuttle rendezvous with SMMS in April of 1984. It indicated the angle rate tracker performance was noiser than expected, while the simulation showed the angle rate tracker performance to be well within the maximum noise specification. As noted in Section 2.1, an investigation of the antenna pattern fidelity effects on the angle rate tracking performance, documented in References 6 and 7, showed that the simple antenna pattern model was a significant contributor to the errors in performance estimates. The angle rate tracker modifications developed during this investigation served as the basis for the SES model upgrade documented in this section.

Prior to the project reported upon herein, the effects of a more accurate servo model had not been investigated. However, some servo model enchancements were investigated on the present project as part of a larger analysis of the angle rate tracking loop performance problems. Results of the angle rate tracking loop analysis and the potential servo model enchancements are documented in Section 4.2.

2.2.2 Definition of Algorithm Modifications

The angle tracking loop algorithm was modified in two areas: (1) the antenna patterns module and (2) digital portion of the track loop filter. Changes in the antenna patterns module were major revisions, while the changes in the digital hardware section were relatively minor.

2.2.2.1 Pattern Model Changes

The original antenna patterns were modeled by analytic equations. The sum pattern was modeled as a surface of revolution about the antenna boresight with a shape given by the expression

$$\text{sumpat} = \sin(bx)/bx \qquad (2\text{-}1)$$

where the constant b was chosen so that the pattern model beamwidth matched the beamwidth of the measured data. The difference pattern was modeled as the derivative of the sum pattern and was given by the equation

$$\text{difpat} = a(b\cos(bx)-\sin(bx))/(bx) \qquad (2\text{-}2)$$

The constant a is chosen to place the 100 percent pattern modulation point at the proper angle off boresight. This model of the sum and difference patterns assumed (1) an infinite null depth on boresight, and (2) the phasing between the sum and difference channel was either 0 or 180 degrees with an instantaneous phase transition on boresight.

The updated antenna pattern models use an extensive set of measured data with interpolation between data points, rather than closed-form equations. Data measurements were taken for five parameters: sum channel gain, elevation difference channel gain, sum-to-elevation difference channel phase, azimuth difference channel gain, and sum-to-azimuth difference channel phase. Data was measured on an 8 degree by 8 degree grid in azimuth and elevation with a resolution of 0.2 degrees. Data sets exist for radar transmit frequencies: 1 (13.779 GHz), 3 (13.883 GHz) and 5 (13.987 GHz). However, to conserve memory, only the data for transmit frequency 1 is used for all five frequency slots in the simulation. This model of the antenna patterns was first developed for the angle tracking performance investigation reported in References 6 and 7. In that case, bicubic spline interpolation was used to generate the sum pattern gain values and both sum-to-difference channel phase values, while linear interpolation was used to generate the difference channel gain values. Three dimensional plots (from Reference 6) of the resulting patterns are shown in Figure 2.2-1 through 2.2-5. These new antenna pattern models are quite accurate and provide the following important features: finite null depth on boresight and non-instantaneous phase transition through boresight. Initially, the antenna model described above was installed in the SES simulation. However, it was found that the bicubic spline interpolation was causing the simulation to run far too slowly. This violated the real-time run constraint applied to original simulation development. To improve program speed, an investigation into the use of two dimensional linear interpolation of all parameters was undertaken. This investigation surfaced two significant results: (1) changes in the angle and angle rate tracking loop performance were minimal and (2) simulation run time was significantly reduced. The reduction in run time was about an order of magnitude, although no official timing tests were performed.

2.2.2.2 Digital Processing Model Changes

These changes specifically apply to the digital hardware section of the angle tracking loop filter (see Figure 2.1-2). This includes the section of the hardware from the angle discriminant output to the input of the digital-to-analog converter (DAC) in the Electronics Assembly 1 (EA-1). The philosophy here was to change this model from an approximate representation of the digital hardware to an exact representation. The changes include: (1) performing finite bit multiplication with the exact digital constants used in the radar, (2) performing finite bit addition, (3) the addition of saturation check models at appropriate points in the system model ,and (4) the addition of a DAC model that converts input bits to a voltage which is input to the gimbal motor model. For comparison, Figure 2.2-6 shows the original loop configuration, while Figure 2.2-7 gives the upgraded version of the loop. Fidelity enchancements provided by these modifications is only second order at best. However, these changes do provide very accurate data at intermediate

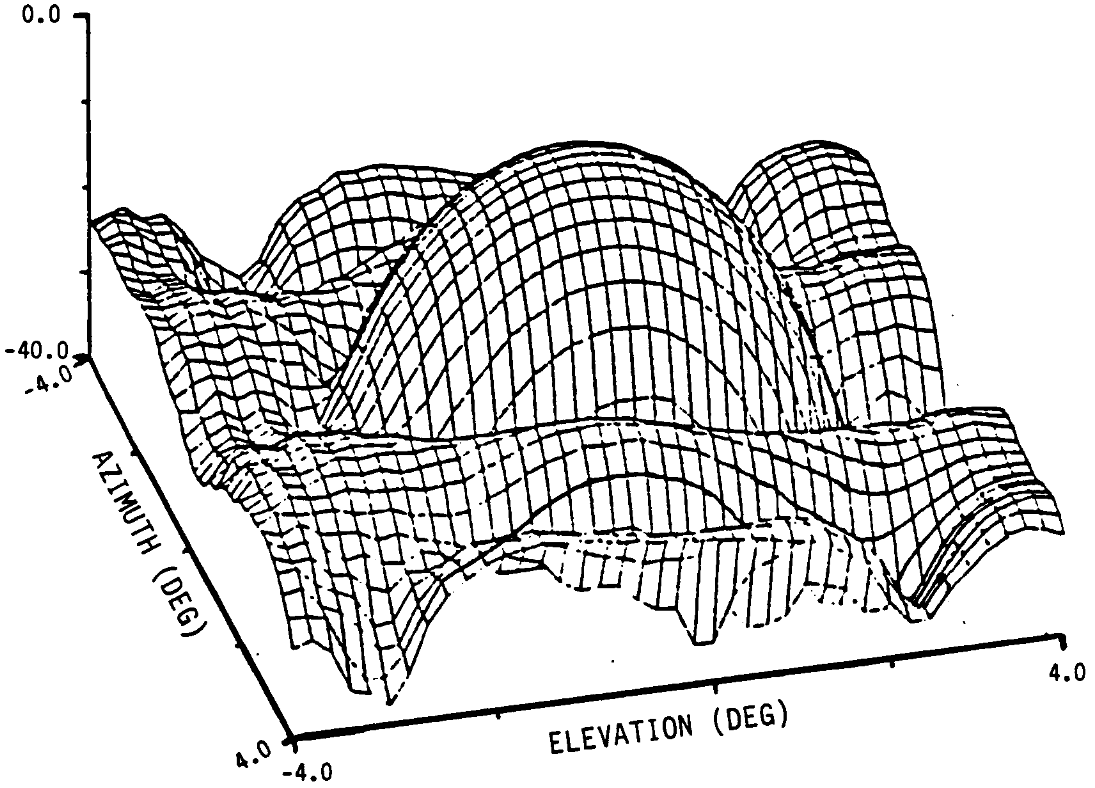


FIGURE 2.2-1 SUM CHANNEL GAIN PATTERN

ORIGINAL PAGE IS OF POOR QUALITY

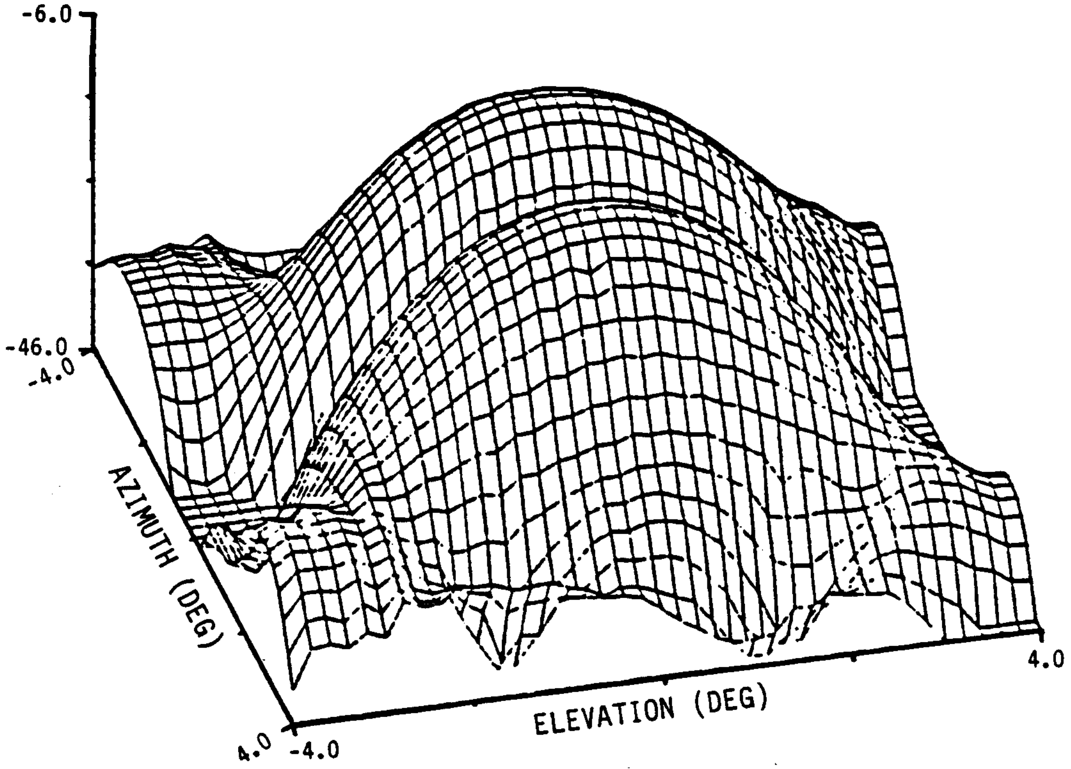


FIGURE 2.2-2 AZIMUTH DIFFERENCE CHANNEL GAIN PATTERN

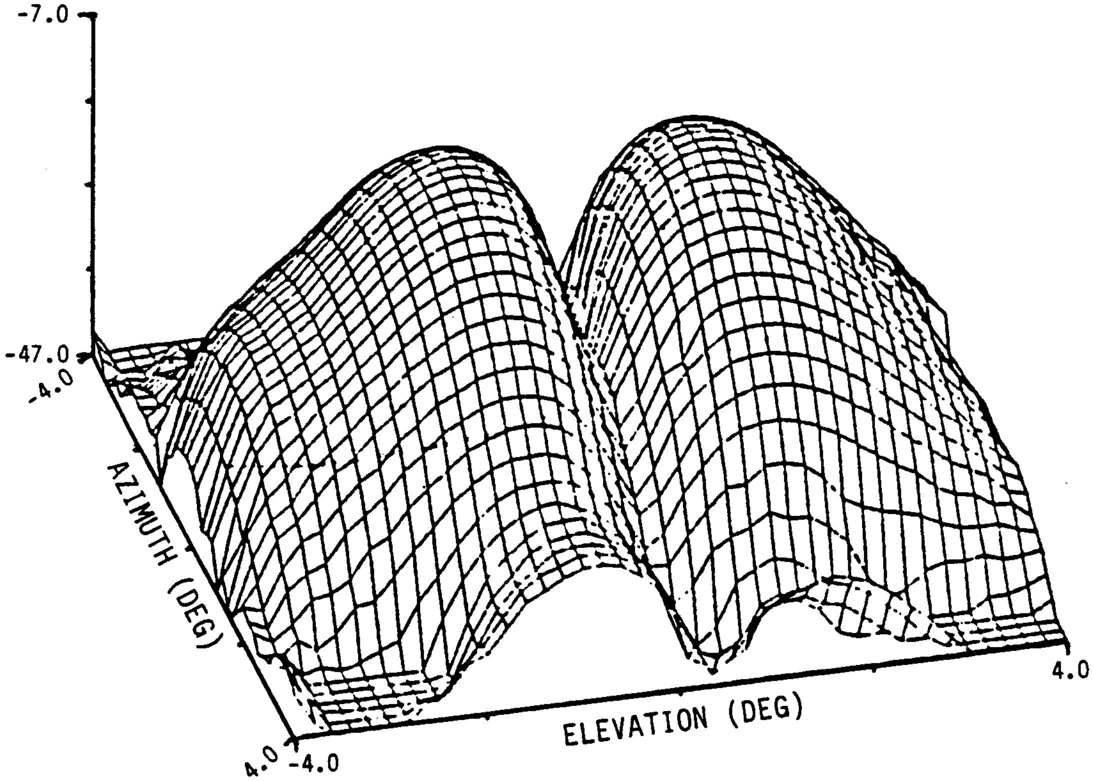


FIGURE 2.2-3 ELEVATION DIFFERENCE CHANNEL GAIN PATTERN

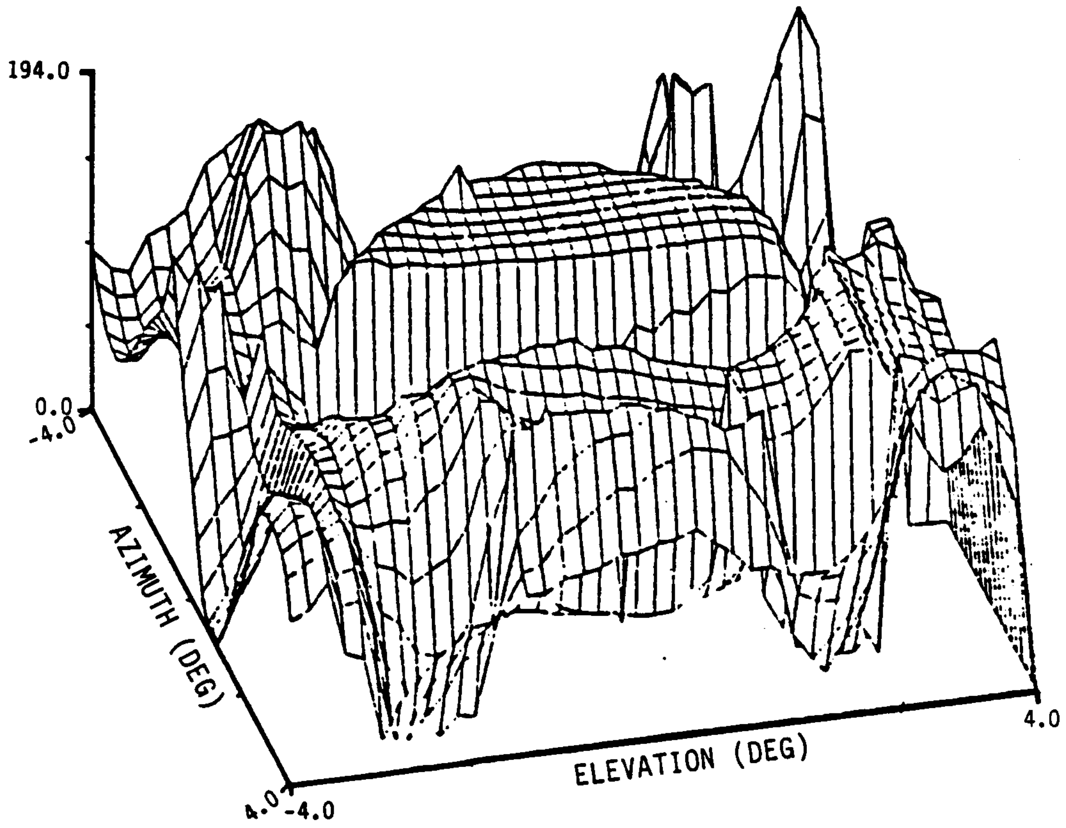


FIGURE 2.2-4 SUM-TO-AZIMUTH DIFFERENCE CHANNEL PHASE PATTERN

ORIGINAL PAGE IS
OF POOR QUALITY

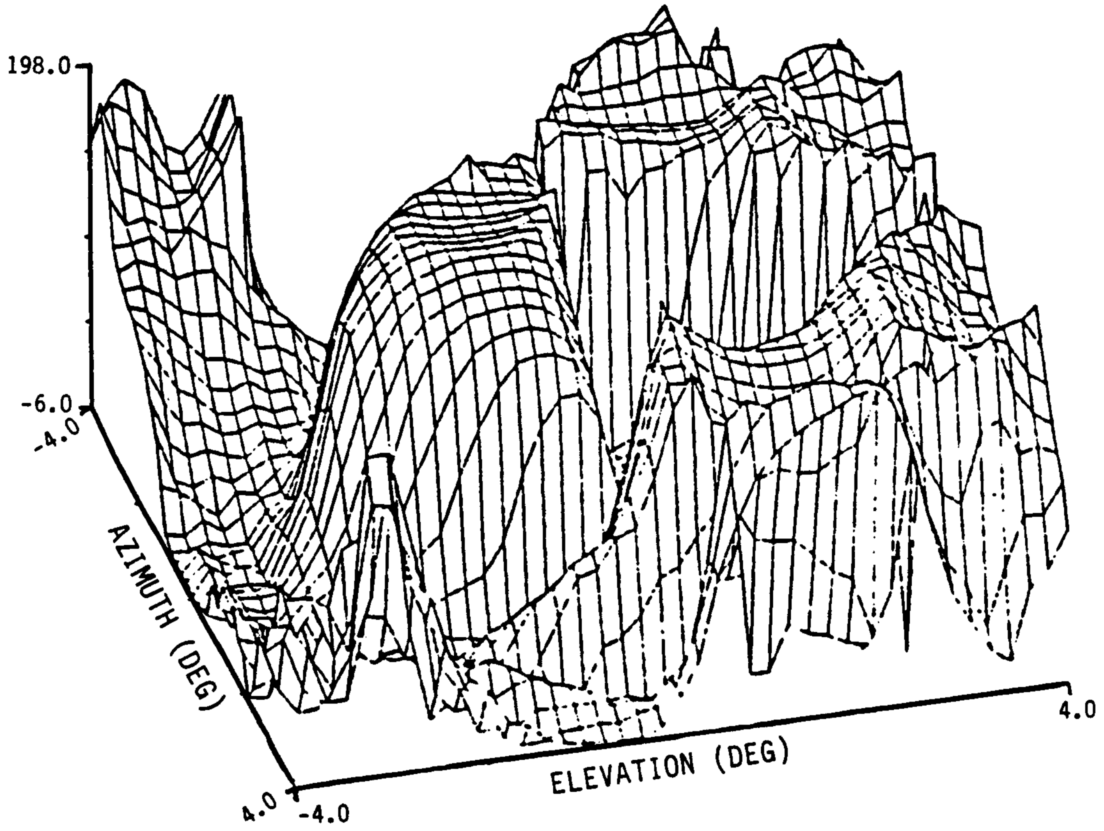
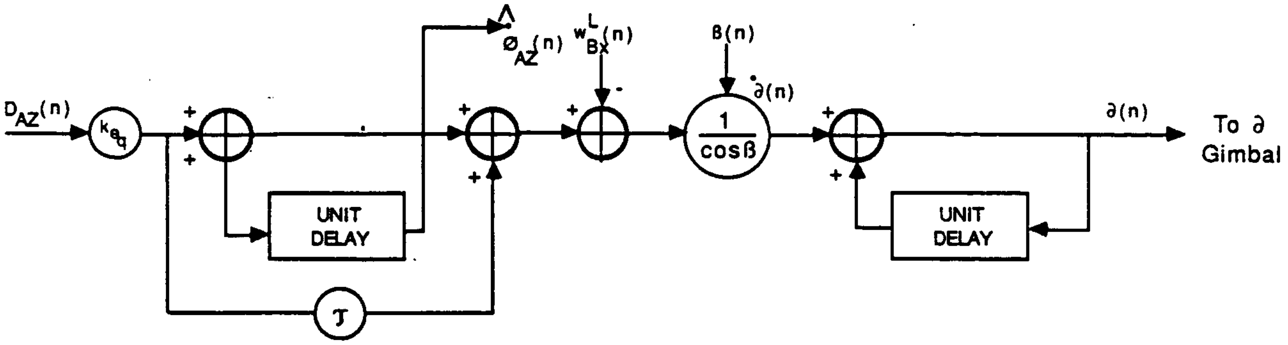
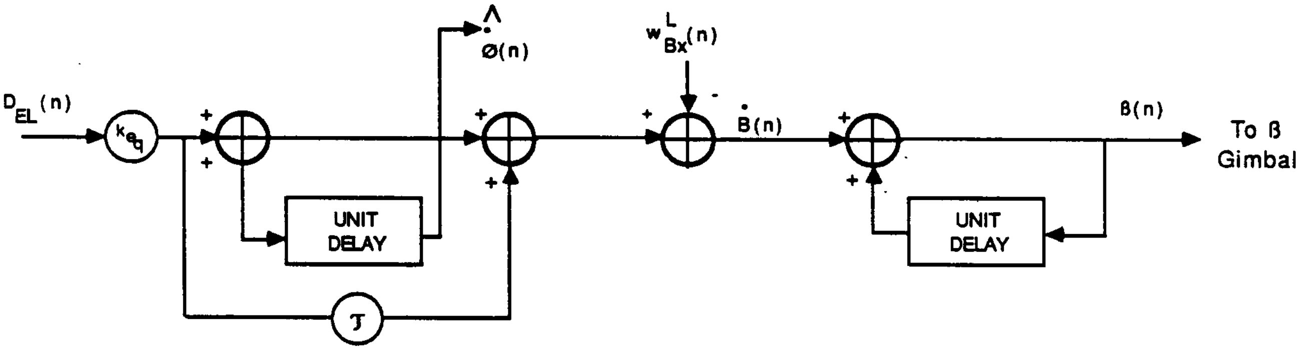


FIGURE 2.2-5 SUM-TO-ELEVATION DIFFERENCE CHANNEL PHASE PATTERN



a. ∂ angle tracking loop filter.



b. ß angle tracking filter

FIGURE 2.2-6 ORIGINAL ANGLE TRACKING LOOP FILTER MODELS

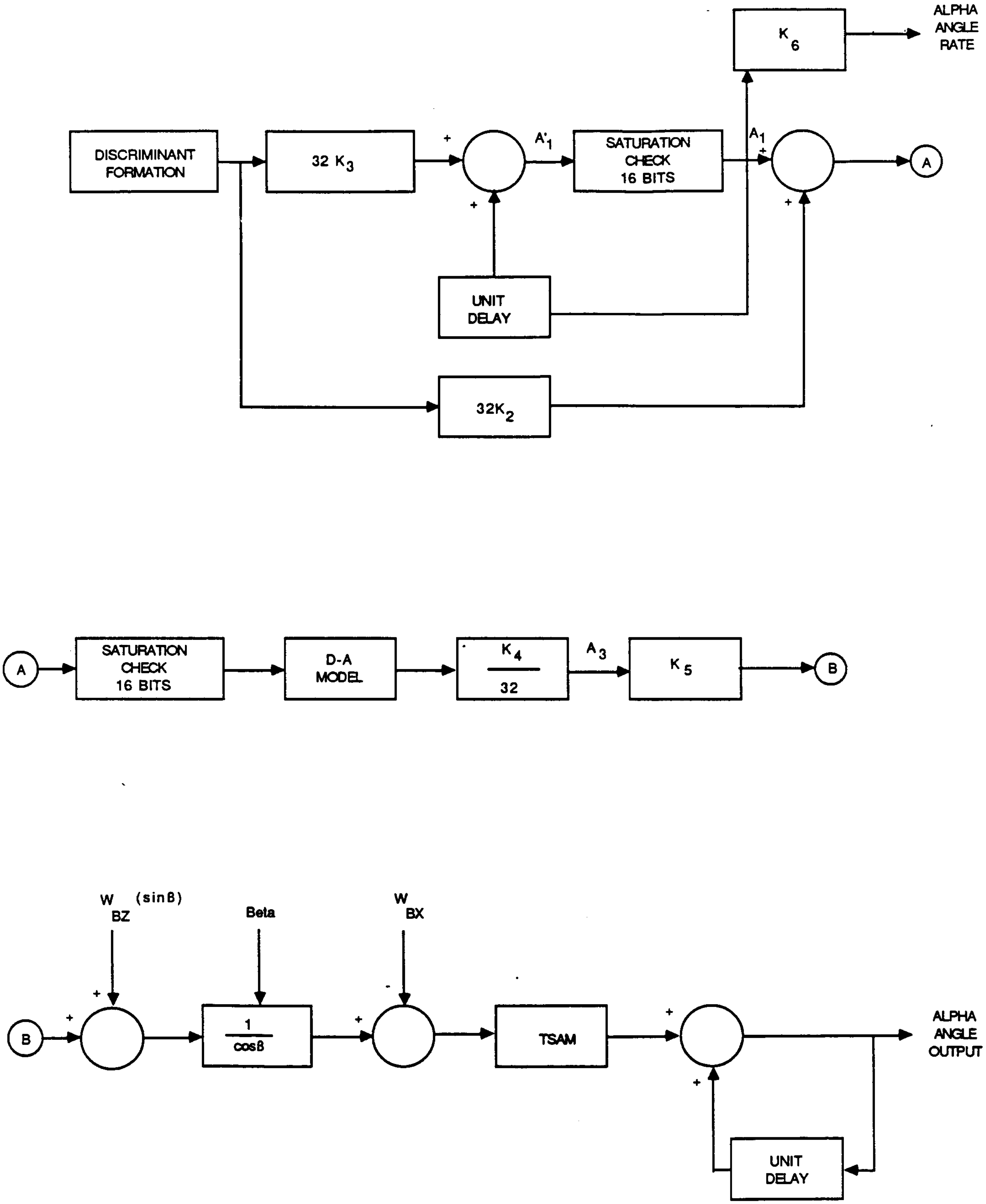


FIGURE 2.2-7 ALPHA TRACKING LOOP MODEL

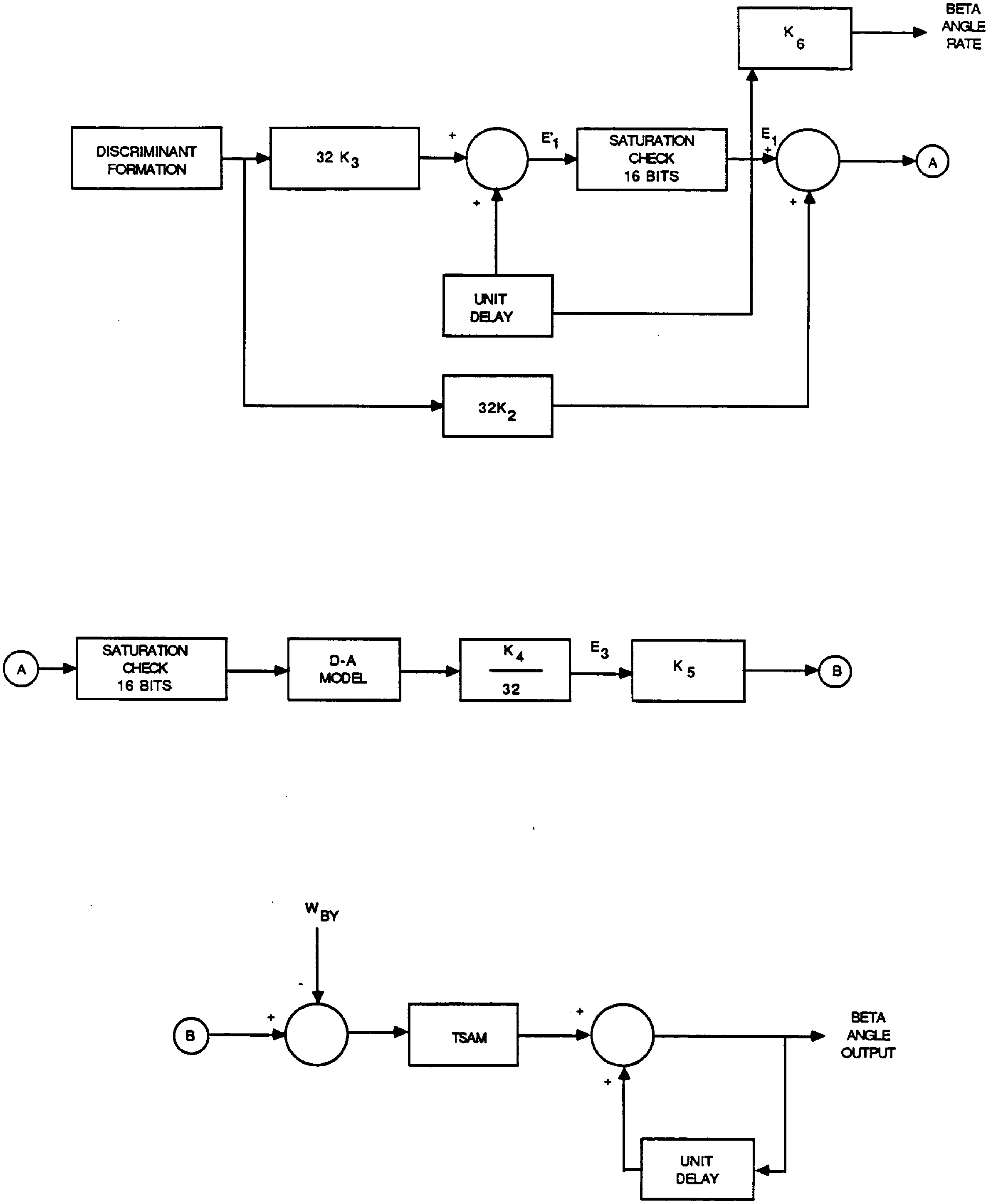


FIGURE 2.2-7 BETA TRACKING LOOP MODEL (CONTINUED)

points throughout the digital hardware section. For example, the alpha error voltage and beta error voltage are easily accessed test points in the actualdigital hardware. The upgraded loop model can now compute similar voltage traces for direct comparison with actual data.

Figure 2.2-6 gives a block diagram of the original alpha and beta tracking loop filters. The equations describing those filters are summarized below. The first step is to update the smoothed ILOS azimuth and elevation rates using the expressions

(2-3)

$$\hat{\dot{\theta}}_{AZ}(n) = \hat{\dot{\theta}}_{AZ}(n-1) + T_s K_{eq} D_{AZ}(n)$$

$$\hat{\dot{\theta}}_{EL}(n) = \theta_{EL}(n-1) + T_s K_{eq} D_{EL}(n)$$

where $\hat{\dot{\theta}}_{EL}$ = smoothed target inertial LOS elevation rate,

$\hat{\dot{\theta}}_{AZ}$ = smoothed target inertial LOS azimuth rate,

T_s = update interval,

K_{eq} = loop constant

D_{EL} = elevation channel discriminant

D_{AZ} = azimuth channel discrimnant

Next, the alpha and beta gimbal rates are updated with the equations

(2-4)

$$\dot{\alpha}(n) = (\omega^L_{TX}(n) + \omega^L_{BZ} \sin(\beta))/\cos\beta - \omega^L_{BX}$$

$$\dot{\beta}(n) = \omega^L_{TY}(n) - \omega^L_{BY}(n)$$

where

$$\omega^L_{TX}(n) = \hat{\dot{\theta}}_{AZ}(n) + K_{eq} \tau T_s D_{AZ}(n)$$

$$\omega^L_{TY}(n) = \hat{\dot{\theta}}_{EL}(n) + K_{eq} \tau T_s D_{EL}(n)$$

$\omega^L_{BX}(n)$ = X-component of body inertial angular velocity at time sample n expressed in L-coordinates.

Finally, the new alpha and beta gimbal positions are computed from the expressions

(2-5)

$$\alpha(n) = \alpha(n-1) + T_s \times \dot{\alpha}(n)$$

$$\beta(n) = \beta(n-1) + T_s \times \dot{\beta}(n)$$

Figure 2.2-7 gives the block diagrams for the upgraded alpha and beta angle tracking loop filter models. The equations defining this upgraded algorithm are defined as follows. The smoothed ILOS azimuth and elevation rates are given by

(2-6)
$$\hat{\dot{\theta}}_{EL}(n) = k_6 E_1\ (n)$$
$$\hat{\dot{\theta}}_{AZ}(n) = k_6 A_1\ (n)$$

where

$$E'_1(n) = E_1(n-1)+k_3 D_{EL}(n)$$

$$E_1(n) = SAT(E'_1(n), 2^{15})$$

$$SAT(x,y) = \begin{cases} y, & x > y \\ x, & x < y \\ -y, & x < -y \end{cases}$$

Similar expressions hold for $A_1(n)$ and $A'_1(n)$. The so-called alpha rate error (A_3) and beta rate error (B_3) voltages (at the DAC output) are given by the expression

(2-7)
$$E_3(n) = k_4 SAT(E_1(n)+k_2 D_{EL}(n), 2^{15})/32$$
$$A_3(n) = k_4 SAT(A_1(n)+k_2 D_{AZ}(n), 2^{15})/32$$

Then, the predicted alpha beta gimbal rates are expressed as

(2-8)
$$\dot{\alpha}(n) = (\omega^{L}_{TX}(n)+\omega^{L}_{BZ}\sin(\beta))/\cos(\beta) - \omega^{L}_{BX}$$

$$\dot{\beta}(n) = (\omega^{L}_{TY}(n)-\omega^{L}_{BY}(n)$$

where

$$\hat{\omega}^{L}_{TX}(n) = k5A3(n)T_S$$

$$\omega^{L}_{TY}(n) = k5E3(n)T_S$$

The final step in the modified algorithm is to update the position of the alpha and beta gimbals. This step is identical to the original algorithm and is given by equation (2-5).

The constants k_4, k_5 and k_6 in equations (2-6) through (2-8) do not change as a function of bandwidth. Values for these constants are summarized in Table 2.2-1 below. The constants k_2 and k_3 differ for the alpha and beta tracking loops and change with angle tracker bandwidth. Values for these constants are given in Table 2.2-2.

TABLE 2.2-1 ANGLE TRACKER CONSTANTS

CONSTANTS	VALUE	UNITS
k_4	0.0048876	volts/bit
k_5	1.18/5	deg/sec-bit
k_6	0.000576/16	deg/sec-bit

TABLE 2.2-2 k_2 AND k_3 VALUES

		$32k_2$		$32k_3$	
PRF, kHz	Range, nm	α	β	α	β
7	1.9	662	866	13	16
7	1.9 to 3.8	407	532	5	6
7	3.8 to 8.2	149	195	1	1
3	8.2	149	195	1	2

2.2.3 Software Design Documentation

The changes described in the previous subsection affected the following existing subroutines: SIGNAL and ATRACK. Changes in the sum and difference channel signal amplitude computation were incorporated into SIGNAL. The changes in the digital hardware section of the loop filter, documented by equations (2-6) through (2-8), were incorporated into ATRACK.

Some remarks about the listings which will be presented below, and throughout this section, are appropriate at this time as an aid to their interpretation. The "original" or "baseline" versions of the subroutines are those which were present in the baseline simulation program HACSIM. The "final" or "modified" versions are those which appear in the deliverable program FINSIM1. The listings of the difference between the baseline and deliverable versions of the subroutines include both those lines which were deleted from the original program, and those which were added to form the final program. The line numbers identifying the deleted lines refer to lines in the original subroutine and the line numbers which appear next to the added lines refer to lines in the final version of the subroutine.

Figure 2.2-8 is a listing of the original version of SIGNAL as it existed in the baseline program HACSIM. Figure 2.2-9 is a listing of the modified version of SIGNAL which is in the deliverable program FINSIM1. Figure 2.2-10 is a summary of the differences between the subroutines.

Figure 2.2-11 is a listing of the original version of ATRACK. Figure 2.2-12 is a listing of the modified version of ATRACK. Figure 2.2-13 is a listing of the differences between the two subroutines.

Modifications to the angle rate tracking loop required the generation of three new routines: KSAT, READPAT, and INTERP. KSAT is a generalized routine that checks for saturation of a digital signal. Inputs include the untested signal of interest and the desired saturation level. The output is the tested (and possibly modified) signal. READPAT is the subroutine that is used to read the measured antenna pattern data into the appropriate common blocks. This subroutine is executed only one time, and this is upon the first call to the subroutine INTERP. Subroutine INTERP computes the sum pattern gain, the azimuth difference pattern gain, the elevation difference pattern gain, the sum-to-elevation difference channel phase, and the sum-to-azimuth difference channel phase for a given pair of azimuth and elevation angles. As mentioned in the previous subsection, the values are computed using two dimensional linear interpolation and the measured data. The inputs to the subroutine are the azimuth and elevation angle. The data computed by the subroutine is passed back to the calling program via a labeled common block.

(THIS PAGE INTENTIONALLY LEFT BLANK)

```
C
C  ****************************************************************
C  * THIS SUBROUTINE GENERATES THE NOISE-FREE ANGLE, RANGE, VELOCITY *
C  * AND ON-TARGET DISCRIMINANT COMPONENTS.                          *
C  ****************************************************************
C
C
      SUBROUTINE SIGNAL
      REAL IRDOT,IRNG
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)
      COMMON /OUTPUT/I1DUM(3),SRNG,DUM1(6),IDUM2(4)
      COMMON /ICNTL/IDUM5(13),MTKINT,MRNG,MSAM,MPRF,MBKTRK,MBTSUM,
     2              MBT(8)
      COMMON /TGTDAT/NT,RAU(3,100),RANGE(100),RADVEL(100),RO(3),
     2               ROU(3),CGRNGE,CGVEL
      COMMON /SATDAT/RADAR(3),N20,RT(70,3),SIG(70)
      COMMON /RTDAT/IRDOT,IRNG,DUM2(5),MDF(5)
      COMMON /SIGDAT/SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE,DF1,DF5,
     2               DF2,DF4,SIGBAR
      COMMON /XFORMS/TLB(3,3),TLBD(3,3),TLT(3,3),TLTD(3,3)
      COMPLEX CSUM,CDIFAZ,CDIFEL,CEARLY,CLATE,CDF1,CDF5,CDF2,CDF4,
     2        DFWTS,PHASE,PHASE1,DOPFIL
      DIMENSION CTP(10,2),DFWTS(5,100),ALAM(5),ALAMD(3),NFREQ(2)
      DATA CTP/9*.03318,9.799E-4,4*.03318,1.9599E-3,9.8E-4,4.9E-4,
     2         2*2.45E-4,1.225E-4/
      DATA NFREQ/1,5/,ALAM/177.3733,176.0447,178.7149,176.7089,
     2     178.0393/,ALAMD/1.272461E-2,2.969089E-2,3.309023E-1/
      REAL LATE
C
C  ****************************************************************
C  *  STEP 1: PRELIMINARY COMPUTATIONS AND PARAMETER INITIALIZATION *
C  ****************************************************************
C
C  STEP 1-1: INITIALIZE DISCRIMINANT COMPONENTS (NOTE: THESE ARE THE
C            COMPONENT SIGNALS AFTER SQUARE-LAW DETECTION).
      SPAZ=0.0
      SMAZ=0.0
      SPEL=0.0
      SMEL=0.0
      EARLY=0.0
      LATE=0.0
      DF1=0.0
      DF5=0.0
      DF2=0.0
      DF4=0.0
      SIGBAR=0.0
C
      NFMAX=NFREQ(IMODE)
      DO 55 I=1,NFMAX
C
```

FIGURE 2.2-8 BASELINE VERSION OF SUBROUTINE SIGNAL
PAGE 1

```
C  STEP 1-2: INITIALIZE COMPLEX DISCRIMINANT COMPONENTS BEFORE EACH
C            XMIT FREQUENCY (NOTE: THESE ARE THE COMPONENT SIGNALS
C            BEFORE SQUARE-LAW DETECTION).
      CSUM=(0.,0.)
      CDIFAZ=(0.,0.)
      CDIFEL=(0.,0.)
      CEARLY=(0.,0.)
      CLATE=(0.,0.)
      CDF1=(0.,0.)
      CDF5=(0.,0.)
      CDF2=(0.,0.)
      CDF4=(0.,0.)
      DO 45 K=1,NT
C
      IF(I.GT.1) GO TO 35
C
C  ***********************************************************
C  * STEP 2: COMPUTE SUM CHANNEL MULTIPLICATION FACTOR FOR KTH *
C  *         SCATTERER.                                        *
C  ***********************************************************
C
C  STEP 2-1: COMPUTE SUM PATTERN ANGLE.
      PSI=ACOS(ABS(RAU(3,K)))
C
C  STEP 2-2: COMPUTE ANTENNA SUM PATTERN MULTIPLICATION FACTOR.
      X=SPAT(PSI)
C
C  STEP 2-3: COMPUTE SUM CHANNEL MULTIPLICATION FACTOR.
      XX=SIG(K)*X
C  NOTE: IF IN ACTIVE MODE SET XX=1.0.
      IF(IMODE.EQ.1) XX=1.0
      S=XX*X
C
C  STEP 2-4: CHECK ANTENNA STEERING MODE (IF IN GPC-DES OR MANUAL
C            -- SKIP STEP 4).
      IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 20
C
C  ****************************************************************
C  * STEP 3: COMPUTE AZ AND EL DIFFERENCE CHANNEL MULTIPLICATION *
C  *         FACTORS FOR KTH SCATTERER.                           *
C  ****************************************************************
C
C  STEP 3-1: COMPUTE AZ AND EL DIFFERENCE PATTERN ANGLES.
      DELAZ=-ASIN(RAU(2,K))
      DELEL=ASIN(RAU(1,K))
C
C  STEP 3-2: COMPUTE AZ AND EL DIFFERENCE PATTERN MULTIPLICATION
C            FACTORS.
      Y=DPAT(DELAZ)
      Z=DPAT(DELEL)
C
C  STEP 3-3: COMPUTE AZ AND EL DIFFERENCE CHANNEL MULTIPLICATION
C             FACTORS (INCLUDE RCS AND SUM PATTERN WEIGHTINGS).
      DAZ=XX*Y
      DEL=XX*Z
C
C  *******************************************************
C  * STEP 4: COMPUTE RANGE GATE WEIGHTING FOR KTH SCATTERER *
C  *******************************************************
C  DEFINITION:  CTP=4./(C*PULSEWIDTH) WHERE C IS SPEED OF LIGHT.
C
C  STEP 4-1: COMPUTE RANGE GATE LOCATION WRT RANGE GATE CENTER.
CCCCCCCCCCCCCCCCCCC MOD MAR 24 1983 CCCCCCCCCCCCCCCCCC
   20  CONTINUE
```

FIGURE 2.2-8 BASELINE VERSION OF SUBROUTINE SIGNAL

PAGE 2

```
      SRNGX=10.*AINT(0.03125*IRNG)
      DELX=CTP(MRNG,IMODE)*(RANGE(K)-SRNGX)
C
C  STEP 4-2: COMPUTE EARLY AND LATE RANGE GATE WEIGHTINGS FOR
C            KTH SCATTERER.
      II=INT((DELX+7.)/2.)
      IF(II.LE.1) II=1
      IF(II.GE.5) II=5
      GO TO (21,22,23,24,21),II
   21 RGE=0.0
      RGL=0.0
      GO TO 25
   22 RGE=3.+DELX
      RGL=0.0
      GO TO 25
   23 RGE=1.-DELX
      RGL=1.+DELX
      GO TO 25
   24 RGE=0.0
      RGL=3.-DELX
C
C  STEP 4-3: COMPUTE RANGE GATE WEIGHT FOR NON-RANGE DISCRIMINANT
C            COMPONENTS.
   25 RGWGT=0.5*(RGL+RGE)
C
C  STEP 4-4: APPLY RANGE GATE WEIGHTING TO SUM AND DIFFERENCE
C            CHANNEL MULTIPLICATION FACTORS.
      RGE=S*RGE
      RGL=S*RGL
      S=S*RGWGT
      DAZ=DAZ*RGWGT
      DEL=DEL*RGWGT
C
C  *****************************************************************
C  * STEP 5: COMPUTE DOPPLER FILTER PHASE SHIFT AND WEIGHTING FOR KTH *
C  *         SCATTERER. NOTE: THIS CALCULATION IS INDEPENDENT OF XMIT *
C  *         FREQUENCY AND ASSUMES NO ACCELERATION OVER DATA CYCLE.   *
C  *****************************************************************
C
C  DEFINITION:  ALAMD(MPRF)=2.*PI/(PRF*LAMBDA)
C  DEFINITION:  THE CONSTANT 0.196348=PI/16.
C
C  STEP 5-2: COMPUTE DOPPLER FREQUENCY CORRESPONDING TO RADIAL VELOCITY
C            OF KTH SCATTERER.
      FDT=-2.*ALAMD(MPRF)*RADVEL(K)
C
C  STEP 5-3: COMPUTE DOPPLER FILTER WEIGHTING FOR EACH OF FIVE DOPPLER
C            TRACKING FILTERS.
      DO 30 J=1,5
      ARG=0.196348*MDF(J)-FDT
   30 DFWTS(J,K)=DOPFIL(ARG)
C
C  *****************************************************************
C  * STEP 6: COMPUTE PHASE FACTOR ASSOCIATED WITH KTH SCATTERER RANGE *
C  *         (NOTE: PHASE IS REFERENCD TO PHASE ASSOCIATED WITH RANGE *
C  *         OF TARGET C.G.)                                          *
C  *****************************************************************
C
C  DEFINITION: RANGE(K) IS RANGE OF KTH SCATTERER TO ANTENNA PHASE CENTR
C  DEFINITION: ALAM=4.*PI/LAMBDA WHERE LAMBDA IS XMIT FREQUENCY.
C
C  STEP 6-1: COMP''TE PHASE REFERENCED TO TARGET C.G.
   35 DELPSI=ALAM(I)*(RANGE(K)-CGRNGE)
C
```

FIGURE 2.2-8 BASELINE VERSION OF SUBROUTINE SIGNAL

PAGE 3

```
C  STEP 6-2: COMPUTE PHASE FACTOR, I.E. EXP(J*DELPHI).
      PHASE=CEXP(CMPLX(0.,DELPSI))
      PHASE1=PHASE
C
C  STEP 6-3: COMBINE RANGE PHASE FACTOR AND DOPPLER FILTER =3
C            WEIGHT AND PHASE FACTOR.
      PHASE=PHASE*DFWTS(3,K)
C
C  ***************************************************************
C  * STEP 7: ADD (VECTORIALLY) KTH SCATTERER CONTRIBUTION TO EACH *
C  *         DISCRIMINANT'S COMPONENT SIGNALS.                    *
C  ***************************************************************
C
C  STEP 7-1: ADD KTH SCATTERER CONTRIBUTION TO SUM CHANNEL SIGNAL.
      CSUM=CSUM+S*PHASE
C
C  STEP 7-2: CHECK ANTENNA STEERING MODE ---- SKIP STEP 8-3 IF IN
C            GPC-DES OR MANUAL MODE.
      IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 40
C
C  STEP 7-3: ADD KTH SCATTERER CONTRIBUTION TO AZ AND EL DIFFERENCE
C            CHANNELS SIGNALS.
      CDIFAZ=CDIFAZ+DAZ*PHASE
      CDIFEL=CDIFEL+DEL*PHASE
C
C  STEP 7-4: ADD KTH SCATTERER CONTRIBUTION TO RANGE DISCRIMINANT
C            COMPONENT SIGNALS.
   40 CEARLY=CEARLY+RGE*PHASE
      CLATE=CLATE+RGL*PHASE
C
C  STEP 7-5: ADD KTH SCATTERER CONTRIBUTION TO VELOCITY DISCRIMINANT
C            COMPONENT SIGNALS.
      PHASE1=PHASE1*S
      CDF2=CDF2+PHASE1*DFWTS(2,K)
      CDF4=CDF4+PHASE1*DFWTS(4,K)
C
C  STEP 7-6: ADD KTH SCATTERER CONTRIBUTION TO ON-TARGET DISCRIMINANT
C            COMPONENT SIGNALS.
      CDF1=CDF1+PHASE1*DFWTS(1,K)
      CDF5=CDF5+PHASE1*DFWTS(5,K)
   45 CONTINUE
C
C  *****************************************************************
C  * STEP 8: FORM NOISE-FREE ANGLE, RANGE, VELOCITY, AND ON-TARGET *
C  *         DISCRIMINANT COMPONENTS AT ITH FREQUENCY AND SQUARE   *
C  *         LAW DETECT THESE COMPONENTS.                          *
C  *****************************************************************
C
C  STEP 8-1: CHECK ANTENNA STEERING MODE ---- SKIP STEPS 9-2 AND 9-3
C             IF IN GPC-DES OR MANUAL.
      IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 50
C
C  STEP 8-2: COMPUTE AZ DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
      SPAZ=SPAZ+CABS(CSUM+CDIFAZ)**2
      SMAZ=SMAZ+CABS(CSUM-CDIFAZ)**2
C
C  STEP 8-3: COMPUTE EL DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
      SPEL=SPEL+CABS(CSUM+CDIFEL)**2
      SMEL=SMEL+CABS(CSUM-CDIFEL)**2
C
C  STEP 8-4: COMPUTE RANGE DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT
   50 EARLY=EARLY+CABS(CEARLY)**2
      LATE=LATE+CABS(CLATE)**2
C
```

FIGURE 2.2-8 BASELINE VERSION OF SUBROUTINE SIGNAL

PAGE 4

```
C  STEP 8-5: COMPUTE VELOCITY DISCRIMINANT COMPONENTS AND SQUARE-LAW
C            DETECT.
       DF2=DF2+CABS(CDF2)**2
       DF4=DF4+CABS(CDF4)**2
C
C  STEP 8-6: COMPUTE ON-TARGET DISCRIMINANT COMPONENTS AND SQUARE-LAW
C            DETECT.
       DF1=DF1+CABS(CDF1)**2
       DF5=DF5+CABS(CDF5)**2
C
C  ****************************************************************
C  * STEP 9: COMPUTE EFFECTIVE CROSS-SECTION AVERAGED OVER PROPER *
C  *         NUMBER OF TRANSMIT FREQUENCIES.                      *
C  ****************************************************************
       SIGBAR=SIGBAR+CABS(CSUM)**2
   55  CONTINUE
       SIGBAR=SIGBAR/FLOAT(NFREQ(IMODE))
C
C   NOTE: DEBUGGING PRINT STATEMENTS
C      WRITE(6,900) (I,SIG(I), I=1,NT)
  900  FORMAT(' I,SIG =',I8,F14.4)
C      WRITE(6,902) NT,S,DAZ,DEL,RGE,RGL,RGWGT,MDF(3)
C      WRITE(6,901) DFWTS(1,K),DFWTS(2,K),DFWTS(3,1),DFWTS(4,1),
C    2 DFWTS(5,1)
  902  FORMAT(' NT,S,DAZ,DEL,RGE,RGL,RGWGT,F3 =',I5,6F10.2,I5)
  901  FORMAT(' DF WTS =',10F12.4)
       RETURN
       END
C
```

FIGURE 2.2-8 BASELINE VERSION OF SUBROUTINE SIGNAL

PAGE 5

(THIS PAGE INTENTIONALLY LEFT BLANK)

```
C
C   ****************************************************************
C   * THIS SUBROUTINE GENERATES THE NOISE-FREE ANGLE, RANGE, VELOCITY *
C   * AND 0N-TARGET DISCRIMINANT COMPONENTS.                         *
C   ****************************************************************
C
C
      SUBROUTINE SIGNAL
      REAL IRDOT,IRNG
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)
      COMMON /OUTPUT/I1DUM(3),SRNG,DUM1(6),IDUM2(4)
      COMMON /ICNTL/IDUM5(13),MTKINT,MRNG,MSAM,MPRF,MBKTRK,MBTSUM,
     2            MBT(8)
      COMMON /TGTDAT/NT,RAU(3,100),RANGE(100),RADVEL(100),RO(3),
     2             ROU(3),CGRNGE,CGVEL
      COMMON /SATDAT/RADAR(3),N20,RT(70,3),SIG(70)
      COMMON /RTDAT/IRDOT,IRNG,DUM2(5),MDF(5)
      COMMON /SIGDAT/SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE,DF1,DF5,
     2             DF2,DF4,SIGBAR
      COMMON /XFORMS/TLB(3,3),TLBD(3,3),TLT(3,3),TLTD(3,3)
      COMMON /SUDIPH/ X,Y,Z,PAZ,PEL
      COMPLEX CSUM,CDIFAZ,CDIFEL,CEARLY,CLATE,CDF1,CDF5,CDF2,CDF4,
     2        DFWTS,PHASE,PHASE1,DOPFIL
      DIMENSION CTP(10,2),DFWTS(5,100),ALAM(5),ALAMD(3),NFREQ(2)
      DATA CTP/9*.03318,9.799E-4,4*.03318,1.9599E-3,9.8E-4,4.9E-4,
     2         2*2.45E-4,1.225E-4/
      DATA NFREQ/1,5/,ALAM/177.3733,176.0447,178.7149,176.7089,
     2      178.0393/,ALAMD/1.272461E-2,2.969089E-2,3.309023E-1/
      REAL LATE
      COMPLEX DAZ,DEL
      DATA ILOOP/1/
C
C   ****************************************************************
C
C   MODIFIED JAN 10 1986 BY M. MEYER
C   MODIFICATIONS TO SUBROUTINE SIGNAL INCLUDE
C   CALCULATION OF THE AZIMUTH AND ELEVATION ANGLES
C   USE OF MEASURED ANTENNA PATTERNS INSTEAD
C   OF FUNCIONS SPAT AND DPAT AND A
C   FACTOR IN THE DIFFERENCE CHANNELS SIGNAL
C   WHICH ACCOUNTS FOR THE FINITE WIDTH PHASE
C   TRANSITION IN THE REAL PHASE PATTERNS.
C
C   ****************************************************************
C
C   ****************************************************************
C   *  STEP 0: READ IN ANTENNA PATTERNTERNS AND SET PHASE BALANCE   *
C   ****************************************************************
C
      IF (ILOOP.NE.1) GO TO 11
```

PRECEDING PAGE BLANK NOT FILMED

FIGURE 2.2-9 DELIVERABLE VERSION OF SUBROUTINE SIGNAL

PAGE 1

```
        CALL READPAT
        PBAL=0.
        ILOOP=0
11    CONTINUE
C                                                                       00020320
C  ****************************************************************     00020330
C  *  STEP 1: PRELIMINARY COMPUTATIONS AND PARAMETER INITIALIZATION *   00020340
C  ****************************************************************     00020350
C                                                                       00020360
C  STEP 1-1: INITIALIZE DISCRIMINANT COMPONENTS (NOTE: THESE ARE THE    00020370
C            COMPONENT SIGNALS AFTER SQUARE-LAW DETECTION).             00020380
      SPAZ=0.0                                                          00020390
      SMAZ=0.0                                                          00020400
      SPEL=0.0                                                          00020410
      SMEL=0.0                                                          00020420
      EARLY=0.0                                                         00020430
      LATE=0.0                                                          00020440
      DF1=0.0                                                           00020450
      DF5=0.0                                                           00020460
      DF2=0.0                                                           00020470
      DF4=0.0                                                           00020480
      SIGBAR=0.0                                                        00020490
C                                                                       00020500
      NFMAX=NFREQ(IMODE)                                                00020510
      DO 55 I=1,NFMAX                                                   00020520
C                                                                       00020530
C  STEP 1-2: INITIALIZE COMPLEX DISCRIMINANT COMPONENTS BEFORE EACH     00020540
C            XMIT FREQUENCY (NOTE: THESE ARE THE COMPONENT SIGNALS      00020550
C            BEFORE SQUARE-LAW DETECTION).                              00020560
      CSUM=(0.,0.)                                                      00020570
      CDIFAZ=(0.,0.)                                                    00020580
      CDIFEL=(0.,0.)                                                    00020590
      CEARLY=(0.,0.)                                                    00020600
      CLATE=(0..0.)                                                     00020610
      CDF1=(0 ,0.)                                                      00020620
      CDF5=(0.,0.)                                                      00020630
      CDF2=(0.,0.)                                                      00020640
      CDF4=(0.,0.)                                                      00020650
      DO 45 K=1,NT                                                      00020660
C                                                                       00020670
      IF(I.GT.1) GO TO 35                                               00020680
C                                                                       00020690
C  *********************************************************            00020700
C  * STEP 2: COMPUTE SUM CHANNEL MULTIPLICATION FACTOR FOR KTH *        00020710
C  *         SCATTERER.                                        *        00020720
C  *********************************************************            00020730
C                                                                       00020740
C  STEP 2-1: COMPUTE AZIMUTH AND ELEVATION ANGLE.                       00020770
      AZ=ATAN2D(RAU(2,K),ABS(RAU(3,K)))
      EL=-ATAN2D(RAU(1,K),ABS(RAU(3,K)))
C  STEP 2-2: COMPUTE ANTENNA SUM, DIFFERENCE AND PHASE FACTORS          00020780
      CALL INTERP(AZ,EL)                                                00020790
C                                                                       00020800
C  STEP 2-3: COMPUTE SUM CHANNEL MULTIPLICATION FACTOR.                 00020810
      XX=SIG(K)*X                                                       00020820
C  NOTE: IF IN ACTIVE MODE SET XX=1.0.                                  00020830
      IF(IMODE.EQ.1) XX=1.0                                             00020840
      S=XX*X                                                            00020850
C                                                                       00020860
C  STEP 2-4· CHECK ANTENNA STEERING MODE (IF IN GPC-DES OR MANUAL       00020870
C            -- SKIP STEP 4).                                           00020880
      IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 20                               00020890
C                                                                       00020900
C  ***********************************************************          00020910
```

FIGURE 2.2-9 DELIVERABLE VERSION OF SUBROUTINE SIGNAL

PAGE 2

```
C   * STEP 3: COMPUTE AZ AND EL DIFFERENCE CHANNEL MULTIPLICATION *
C   *         FACTORS FOR KTH SCATTERER.                            *
C   ***************************************************************
C
C
C   STEP 3-3: COMPUTE AZ AND EL DIFFERENCE CHANNEL MULTIPLICATION
C              FACTORS (INCLUDE RCS AND SUM PATTERN WEIGHTINGS).
C              AND PHASE DIFFERENCE AND BALANCE WEIGHTINGS
       DAZ=XX*Y*CMPLX(COSD(PAZ+PBAL),SIND(PAZ+PBAL))
       DEL=XX*Z*CMPLX(COSD(PEL+PBAL),SIND(PEL+PBAL))
C
C   **********************************************************
C   * STEP 4: COMPUTE RANGE GATE WEIGHTING FOR KTH SCATTERER *
C   **********************************************************
C   DEFINITION:  CTP=4./(C*PULSEWIDTH) WHERE C IS SPEED OF LIGHT.
C
C   STEP 4-1: COMPUTE RANGE GATE LOCATION WRT RANGE GATE CENTER.
CCCCCCCCCCCCCCCCCCC MOD MAR 24 1983 CCCCCCCCCCCCCCCCCC
   20  CONTINUE
       SRNGX=10.*AINT(0.03125*IRNG)
       DELX=CTP(MRNG,IMODE)*(RANGE(K)-SRNGX)
C
C   STEP 4-2: COMPUTE EARLY AND LATE RANGE GATE WEIGHTINGS FOR
C             KTH SCATTERER.
       II=INT((DELX+7.)/2.)
       IF(II.LE.1) II=1
       IF(II.GE.5) II=5
       GO TO (21,22,23,24,21),II
   21  RGE=1.0E-4
       RGL=1.0E-4
       GO TO 25
   22  RGE=3.+DELX
       RGL=0.0
       GO TO 25
   23  RGE=1.-DELX
       RGL=1.+DELX
       GO TO 25
   24  RGE=0.0
       RGL=3.-DELX
C
C   STEP 4-3: COMPUTE RANGE GATE WEIGHT FOR NON-RANGE DISCRIMINANT
C             COMPONENTS.
   25  RGWGT=0.5*(RGL+RGE)
C
C   STEP 4-4: APPLY RANGE GATE WEIGHTING TO SUM AND DIFFERENCE
C             CHANNEL MULTIPLICATION FACTORS.
       RGE=S*RGE
       RGL=S*RGL
       S=S*RGWGT
       DAZ=DAZ*RGWGT
       DEL=DEL*RGWGT
C
C   ********************************************************************
C   * STEP 5: COMPUTE DOPPLER FILTER PHASE SHIFT AND WEIGHTING FOR KTH *
C   *         SCATTERER. NOTE: THIS CALCULATION IS INDEPENDENT OF XMIT *
C   *         FREQUENCY AND ASSUMES NO ACCELERATION OVER DATA CYCLE    *
C   ********************************************************************
C
C   DEFINITION:  ALAMD(MPRF)=2.*PI/(PRF*LAMBDA)
C   DEFINITION:  THE CONSTANT 0.196348=PI/16.
C
C   STEP 5-2. COMPUTE DOPPLER FREQUENCY CORRESPONDING TO RADIAL VELOCITY
C             OF KTH SCATTERER.
       FDT=-2.*ALAMD(MPRF)*RADVEL(K)
```

FIGURE 2.2-9 DELIVERABLE VERSION OF SUBROUTINE SIGNAL

PAGE 3

```
C
C  STEP 5-3: COMPUTE DOPPLER FILTER WEIGHTING FOR EACH OF FIVE DOPPLER
C            TRACKING FILTERS.
      DO 30 J=1,5
      ARG=0.196348*MDF(J)-FDT
   30 DFWTS(J,K)=DOPFIL(ARG)
C
C  ****************************************************************
C  * STEP 6: COMPUTE PHASE FACTOR ASSOCIATED WITH KTH SCATTERER RANGE *
C  *         (NOTE: PHASE IS REFERENCD TO PHASE ASSOCIATED WITH RANGE *
C  *         OF TARGET C.G.)                                          *
C  ****************************************************************
C
C  DEFINITION: RANGE(K) IS RANGE OF KTH SCATTERER TO ANTENNA PHASE CENTR
C  DEFINITION: ALAM=4.*PI/LAMBDA WHERE LAMBDA IS XMIT FREQUENCY.
C
C  STEP 6-1: COMPUTE PHASE REFERENCED TO TARGET C.G.
   35 DELPSI=ALAM(I)*(RANGE(K)-CGRNGE)
C
C  STEP 6-2: COMPUTE PHASE FACTOR, I.E. EXP(J*DELPHI).
      PHASE=CEXP(CMPLX(0.,DELPSI))
      PHASE1=PHASE
C
C  STEP 6-3: COMBINE RANGE PHASE FACTOR AND DOPPLER FILTER =3
C            WEIGHT AND PHASE FACTOR.
      PHASE=PHASE*DFWTS(3,K)
C
C  **************************************************************
C  * STEP 7: ADD (VECTORIALLY) KTH SCATTERER CONTRIBUTION TO EACH *
C  *         DISCRIMINANT'S COMPONENT SIGNALS.                    *
C  **************************************************************
C
C  STEP 7-1: ADD KTH SCATTERER CONTRIBUTION TO SUM CHANNEL SIGNAL.
      CSUM=CSUM+S*PHASE
C
C  STEP 7-2. CHECK ANTENNA STEERING MODE --- SKIP STEP 8-3 IF IN
C            GPC-DES OR MANUAL MODE.
      IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 40
C
C  STEP 7-3: ADD KTH SCATTERER CONTRIBUTION TO AZ AND EL DIFFERENCE
C            CHANNELS SIGNALS.
      CDIFAZ=CDIFAZ+DAZ*PHASE
      CDIFEL=CDIFEL+DEL*PHASE
C
C  STEP 7-4: ADD KTH SCATTERER CONTRIBUTION TO RANGE DISCRIMINANT
C            COMPONENT SIGNALS.
   40 CEARLY=CEARLY+RGE*PHASE
      CLATE=CLATE+RGL*PHASE
C
C  STEP 7-5: ADD KTH SCATTERER CONTRIBUTION TO VELOCITY DISCRIMINANT
C            COMPONENT SIGNALS.
      PHASE1=PHASE1*S
      CDF2=CDF2+PHASE1*DFWTS(2,K)
      CDF4=CDF4+PHASE1*DFWTS(4,K)
C
C  STEP 7-6: ADD KTH SCATTERER CONTRIBUTION TO ON-TARGET DISCRIMINANT
C            COMPONENT SIGNALS.
      CDF1=CDF1+PHASE1*DFWTS(1,K)
      CDF5=CDF5+PHASE1*DFWTS(5,K)
   45 CONTINUE
C
C  ****************************************************************
C  * STEP 8: FORM NOISE-FREE ANGLE, RANGE, VELOCITY, AND ON-TARGET *
C  *         DISCRIMINANT COMPONENTS AT ITH FREQUENCY AND SQUARE   *
```

FIGURE 2.2-9 DELIVERABLE VERSION OF SUBROUTINE SIGNAL

PAGE 4

```
C *         LAW DETECT THESE COMPONENTS.                              *    00022240
C ******************************************************************    00022250
C                                                                       00022260
C  STEP 8-1: CHECK ANTENNA STEERING MODE --   SKIP STEPS 9-2 AND 9-3    00022270
C            IF IN GPC-DES OR MANUAL.                                   00022280
      IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 50                               00022290
C                                                                       00022300
C  STEP 8-2: COMPUTE AZ DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.  00022310
      SPAZ=SPAZ+CABS(CSUM+CDIFAZ)**2                                    00022320
      SMAZ=SMAZ+CABS(CSUM-CDIFAZ)**2                                    00022330
C                                                                       00022340
C  STEP 8-3: COMPUTE EL DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.  00022350
      SPEL=SPEL+CABS(CSUM+CDIFEL)**2                                    00022360
      SMEL=SMEL+CABS(CSUM-CDIFEL)**2                                    00022370
C                                                                       00022380
C  STEP 8-4: COMPUTE RANGE DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT00022390
   50 EARLY=EARLY+CABS(CEARLY)**2                                       00022400
      LATE=LATE+CABS(CLATE)**2                                          00022410
C                                                                       00022420
C  STEP 8-5: COMPUTE VELOCITY DISCRIMINANT COMPONENTS AND SQUARE-LAW    00022430
C            DETECT.                                                    00022440
      DF2=DF2+CABS(CDF2)**2                                             00022450
      DF4=DF4+CABS(CDF4)**2                                             00022460
C                                                                       00022470
C  STEP 8-6: COMPUTE ON-TARGET DISCRIMINANT COMPONENTS AND SQUARE-LAW   00022480
C            DETECT.                                                    00022490
      DF1=DF1+CABS(CDF1)**2                                             00022500
      DF5=DF5+CABS(CDF5)**2                                             00022510
C                                                                       00022520
C *****************************************************************     00022530
C * STEP 9: COMPUTE EFFECTIVE CROSS-SECTION AVERAGED OVER PROPER *      00022540
C *         NUMBER OF TRANSMIT FREQUENCIES.                       *     00022550
C *****************************************************************     00022560
      SIGBAR=SIGBAR+CABS(CSUM)**2                                       00022570
   55 CONTINUE                                                          00022580
      SIGBAR=SIGBAR/FLOAT(NFREQ(IMODE))                                 00022590
C
C   NOTE: DEBUGGING PRINT STATEMENTS                                    00022610
C     WRITE(6,900) (I,SIG(I), I=1,NT)                                   00022620
  900 FORMAT(' I,SIG =',I8,F14.4)                                       00022630
C     WRITE(6,902) NT,S,DAZ,DEL,RGE,RGL,RGWGT,MDF(3)                    00022640
C      WRITE(6,901) DFWTS(1,K),DFWTS(2,K),DFWTS(3,1),DFWTS(4,1),         00022650
C    2 DFWTS(5,1)                                                        00022660
  902 FORMAT(' NT,S,DAZ,DEL,RGE,RGL,RGWGT,F3 =',I5,6F10.2,I5)           00022670
  901 FORMAT('  DF WTS =',10F12.4)                                      00022680
      RETURN                                                            00022690
      END                                                               00022700
C                                                                       00007440
```

FIGURE 2.2-9 DELIVERABLE VERSION OF SUBROUTINE SIGNAL

PAGE 5

(THIS PAGE INTENTIONALLY LEFT BLANK)

```
************
LINES DELETED FROM BASELINE PROGRAM
   35          COMPLEX CSUM,CDIFAZ,CDIFEL,CEARLY,CLATE,CDF1,CDF5,CDF2,CDF4,       00020230
******
LINES ADDED TO DELIVERABLE PROGRAM
   35          COMMON /SUDIPH/ X,Y,Z,PAZ,PEL
   36          COMPLEX CSUM,CDIFAZ,CDIFEL,CEARLY,CLATE,CDF1,CDF5,CDF2,CDF4,       00020230
************
************
LINES DELETED FROM BASELINE PROGRAM
   43    C                                                                         00020320
******
LINES ADDED TO DELIVERABLE PROGRAM
   44          COMPLEX DAZ,DEL
   45          DATA ILOOP/1/
   46    C
   47    C  ******************************************************************
   48    C
   49    C  MODIFIED JAN 10 1986 BY M. MEYER
   50    C  MODIFICATIONS TO SUBROUTINE SIGNAL INCLUDE
   51    C  CALCULATION OF THE AZIMUTH AND ELEVATION ANGLES
   52    C  USE OF MEASURED ANTENNA PATTERNS INSTEAD
   53    C  OF FUNCIONS SPAT AND DPAT AND A
   54    C  FACTOR IN THE DIFFERENCE CHANNELS SIGNAL
   55    C  WHICH ACCOUNTS FOR THE FINITE WIDTH PHASE
   56    C  TRANSITION IN THE REAL PHASE PATTERNS.
   57    C
   58    C  ******************************************************************
   59    C
   60    C  ******************************************************************
   61    C  *  STEP 0: READ IN ANTENNA PATTERNTERNS AND SET PHASE BALANCE    *
   62    C  ******************************************************************
   63    C
   64          IF (ILOOP.NE.1) GO TO 11
   65              CALL READPAT
   66              PBAL=0.
   67              ILOOP=0
   68    11    CONTINUE
   69    C                                                                         00020320
************
************
LINES DELETED FROM BASELINE PROGRAM
   86    C  STEP 2-1: COMPUTE SUM PATTERN ANGLE.                                   00020750
   87          PSI=ACOS(ABS(RAU(3,K)))                                            00020760
   88    C                                                                         00020770
   89    C  STEP 2-2: COMPUTE ANTENNA SUM PATTERN MULTIPLICATION FACTOR.          00020780
   90          X=SPAT(PSI)                                                        00020790
   91    C                                                                         00020800
******
LINES ADDED TO DELIVERABLE PROGRAM
```

FIGURE 2.2-10 SUMMARY OF MODIFICATION TO SUBROUTINE SIGNAL

PAGE 1

```
  112    C  STEP 2-1: COMPUTE AZIMUTH AND ELEVATION ANGLE.                   00020770
  113           AZ=ATAN2D(RAU(2,K),ABS(RAU(3,K)))
  114           EL=-ATAN2D(RAU(1,K),ABS(RAU(3,K)))
  115    C  STEP 2-2: COMPUTE ANTENNA SUM, DIFFERENCE AND PHASE FACTORS       00020780
  116           CALL INTERP(AZ,EL)                                           00020790
  117    C                                                                   00020800
************
************
LINES DELETED FROM BASELINE PROGRAM
  107    C  STEP 3-1: COMPUTE AZ AND EL DIFFERENCE PATTERN ANGLES.           00020960
  108           DELAZ=-ASIN(RAU(2,K))                                        00020970
  109           DELEL=ASIN(RAU(1,K))                                         00020980
  110    C                                                                   00020990
  111    C  STEP 3-2: COMPUTE AZ AND EL DIFFERENCE PATTERN MULTIPLICATION    00021000
  112    C          FACTORS.                                                 00021010
  113           Y=DPAT(DELAZ)                                                00021020
  114           Z=DPAT(DELEL)                                                00021030
  115    C                                                                   00021040
******
LINES ADDED TO DELIVERABLE PROGRAM
  133    C                                                                   00021040
************
************
LINES DELETED FROM BASELINE PROGRAM
  118           DAZ=XX*Y                                                     00021070
  119           DEL=XX*Z                                                     00021080
  120    C                                                                   00021090
******
LINES ADDED TO DELIVERABLE PROGRAM
  136    C              AND PHASE DIFFERENCE AND BALANCE WEIGHTINGS
  137           DAZ=XX*Y*CMPLX(COSD(PAZ+PBAL),SIND(PAZ+PBAL))                00021070
  138           DEL=XX*Z*CMPLX(COSD(PEL+PBAL),SIND(PEL+PBAL))                00021080
  139    C                                                                   00021090
************
************
LINES DELETED FROM BASELINE PROGRAM
  138        21 RGE=0.0                                                      00021240
  139           RGL=0.0                                                      00021250
  140           GO TO 25                                                     00021260
******
LINES ADDED TO DELIVERABLE PROGRAM
  157        21 RGE=1.0E-4                                                   00021240
  158           RGL=1.0E-4                                                   00021250
  159           GO TO 25                                                     00021260
************
************
LINES DELETED FROM BASELINE PROGRAM
  174    C                                                                   00021600
  175    C  STEP 5-3: COMPUTE DOPPLER FILTER WEIGHTING FOR EACH OF FIVE DOPPLER 00021610
******
LINES ADDED TO DELIVERABLE PROGRAM
  193    C
  194    C  STEP 5-3: COMPUTE DOPPLER FILTER WEIGHTING FOR EACH OF FIVE DOPPLER 00021610
************
************
LINES DELETED FROM BASELINE PROGRAM
  274    C                                                                   00022600
  275    C   NOTE: DEBUGGING PRINT STATEMENTS                                00022610
******
LINES ADDED TO DELIVERABLE PROGRAM
  293    C
  294    C   NOTE: DEBUGGING PRINT STATEMENTS                                00022610
************
************
```

FIGURE 2.2-10 SUMMARY OF MODIFICATION TO SUBROUTINE SIGNAL

PAGE 2

```
LINES DELETED FROM BASELINE PROGRAM
  279   C       WRITE(6,901) DFWTS(1,K),DFWTS(2,K),DFWTS(3,1),DFWTS(4,1),        00022650
  280   C     2 DFWTS(5,1)                                                        00022660
  281      902  FORMAT(' NT,S,DAZ,DEL,RGE,RGL,RGWGT,F3 =',I5,6F10.2,I5)           00022670
******
LINES ADDED TO DELIVERABLE PROGRAM
  298   C        WRITE(6,901) DFWTS(1,K),DFWTS(2,K),DFWTS(3,1),DFWTS(4,1),        00022650
  299   C      2 DFWTS(5,1)                                                       00022660
  300      902  FORMAT(' NT,S,DAZ,DEL,RGE,RGL,RGWGT,F3 =',I5,6F10.2,I5)          00022670
************

Number of difference sections found: 9
Number of difference records found: 48

DIFFERENCES /IGNORE=()/MERGED=1/OUTPUT=SYS$DISK3:[MCCOLLOUGH]DIFF1.FOR;1-
    SYS$DISK3:[MCCOLLOUGH]SIGNALH.FOR;2-
    SYS$DISK3:[MCCOLLOUGH]SIGNALF.FOR;2
```

FIGURE 2.2-10 SUMMARY OF MODIFICATION TO SUBROUTINE SIGNAL

PAGE 3

(THIS PAGE INTENTIONALLY LEFT BLANK)

ORIGINAL PAGE IS
OF POOR QUALITY

```
C                                                                       00025240
C   ********************************************************            00025250
C   * THIS SUBROUTINE UPDATES AZ AND EL INERTIAL LOS RATES, THE *       00025260
C   * ALPHA AND BETA GIMBAL RATES, THE ALPHA AND BETA GIMBAL    *       00025270
C   * POSITIONS, AND THE TARGET PITCH AND ROLL ANGLES FOR THE   *       00025280
C   * DISPLAY.                                                  *       00025290
C   ********************************************************            00025300
C                                                                       00025310
C                                                                       00025320
      SUBROUTINE ATRACK                                                 00025330
      REAL INTT,IAZDSC,IELDSC                                           00025335
      COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)                          00025350
      COMMON /INPUT/DUM(6),EWB(3),DUM2(18)                              00025360
      COMMON /OUTPUT/I1DUM(3),D1DUM(2),SPANG,SRANG,SPRTE,SRRTE,SRSS,    00025370
     2              IDUM1(4),SSALP,SSBET                                            00025380
      COMMON /ICNTL/I2DUM(14),MRNG,MSAM,MPRF,IDUM2(11)                  00025390
      COMMON /SYSDAT/TSAM,DR(3),CP,SP,PSI,PSBIAS,ALBIAS,BTBIAS,         00025400
     2              DUM4(5)                                             00025410
      COMMON /ATDAT/CA,SA,CB,SB,AZRATE,ELRATE,ALRATE,BTRATE,AL,BT,      00025420
     2             DUM3(4)                                              00025430
      COMMON /DSCRM/AZDISC,ELDISC,DUM1(7)                               00025440
      DIMENSION AT1(10,2),AT2(10,2),TX1(3,3),TX2(3,3),TX3(3,3),TBL(3,3)00025450
     DIMENSION TDC(3)
      DATA AT1/9*1.5529E-3,2.0106E-4,6*3.9750E-3,1.5529E-3,             00025460
     2         3*2.0106E-4/,AT2/9*6.5907E-3,2.3725E-3,                  00025470
     3         6*1.0546E-2,6.5907E-3,3*2.3725E-3/                       00025480
     DATA TDC/0.05122118,0.1195161,0.2561557/
C   DEFINITION:  AT1=KEQ=(WN**2)/(4.*DIFFERENCE PATTERN SLOPE) WHERE    00025490
C                WN IS NATURAL FREQUENCY OF THE LOOP.                   00025500
C   DEFINITION:  AT2=KEQ*TAU WHERE TAU IS PROPORTIONAL TO STEP RESPONSE 00025510
C                CONVERGENCE TIME.                                      00025520
C                                                                       00026700
     TCON=TSAM/TDC(MPRF)
C   ****************************************                            00026710
C   * STEP 1: UPDATE ROUGH RANGE RATE ESTIMATE *                        00026720
C   ****************************************                            00026730
C                                                                       00025530
C   ************************************************************        00025540
C   * STEP 1: UPDATE ANTENNA LOS-TO-BODY TRANSFORMATION (NOTE: TRANS- * 00025550
C   *         FORMATION INCLUDES GIMBAL BIAS ERRORS AND RADAR YAW    *  00025560
C   *         ANGLE ERROR WRT BODY FRAME).                           *  00025570
C   ************************************************************        00025580
      CALL GAMMA(TX1,-(BT+BTBIAS))                                      00025590
      CALL THETA(TX2,-(AL+ALBIAS))                                      00025600
      CALL MULT33(TX2,TX1,TX3)                                          00025610
      CALL PHI(TX2,-PSI)                                                00025620
      CALL MULT33(TX2,TX3,TBL)                                          00025630
C                                                                       00025640
C   ************************************************************        00025650
C   * STEP 2: UPDATE ESTIMATED TARGET INERTIAL AZIMUTH AND ELEVATION *  00025660
```

PRECEDING PAGE BLANK NOT FILMED

FIGURE 2.2-11 BASELINE VERSION OF SUBROUTINE ATRACK

PAGE 1

```
C  *          RATES IN ANTENNA LOS FRAME.                             *
C  ********************************************************************
C
C  QUANTIZE THE ANGLE DISCRIMINANTS TO 3/16 DB.
      IAZDSC=INTT(5.333333*AZDISC*TCON+0.5)/TCON
      IELDSC=INTT(5.333333*ELDISC*TCON+0.5)/TCON
      IF(IELDSC.GT.255)IELDSC=255
      IF(IAZDSC.GT.255)IAZDSC=255
      IF(IELDSC.LT.-256)IELDSC=-256
      IF(IAZDSC.LT.-256)IAZDSC=-256
      ADSC=0.0431*IAZDSC
      EDSC=0.0431*IELDSC
C  UPDATE ESTIMATED TARGET INERTIAL AZIMUTH RATE.
      AZRATE=AZRATE+TSAM*AT1(MRNG,IMODE)*ADSC
C  UPDATE ESTIMATED TARGET INERTIAL ELEVATION RATE.
      ELRATE=ELRATE+TSAM*AT1(MRNG,IMODE)*EDSC
C
C  **********************************************
C  * STEP 3: UPDATE INNER AND OUTER GIMBAL RATES. *
C  **********************************************
C  COMPUTE REQUIRED COMPONENTS OF ORBITER ANGULAR VELOCITY VECTOR IN
C  OUTER GIMBAL FRAME.
      WGX=CP*EWB(1)+SP*EWB(2)
      WGY=CA*(-SP*EWB(1)+CP*EWB(2))+SA*EWB(3)
      WGZ=-SA*(-SP*EWB(1)+CP*EWB(2))+CA*EWB(3)
C  OUTER GIMBAL RATE.
      IF(ABS(CB).LT.1.0E-6) GO TO 2
      ALRATE=(AZRATE+AT2(MRNG,IMODE)*ADSC+WGZ*SB)/CB-WGX
      GO TO 4
    2 ALRATE=0.
    4 CONTINUE
C  INNER GIMBAL RATE.
      BTRATE=(ELRATE+AT2(MRNG,IMODE)*EDSC)-WGY
C
C  ************************************************
C  * STEP 4: UPDATE INNER AND OUTER GIMBAL POSITIONS. *
C  ************************************************
C  OUTER GIMBAL POSITION (ALPHA ANGLE)
      AL=AL+TSAM*ALRATE
C  INNER GIMBAL POSITION (BETA ANGLE)
      BT=BT+TSAM*BTRATE
C
C ADD ALPHA AND BETA TO OUTPUT IN DEG
           SSALP=AL*57.29576
           SSBET=BT*57.29576
C  ***********************************************************
C  * STEP 6: TRANSFORM TARGET ANGLES AND INERTIAL ANGLE RATES TO *
C  *         BODY FRAME FOR USE IN DISPLAYS AND G AND N.         *
C  ***********************************************************
C  NOTE: TRANSFORMATION TBL INCLUDES GIMBAL BIAS ERRORS AND RADAR YAW
C        ANGLE ERROR WRT BODY FRAME.
C  UPDATE TARGET INERTIAL PITCH RATE IN ORBITER BODY COORDINATES
C  FOR DISPLAY.
      SPRTE=-1000.*(TBL(2,1)*AZRATE+TBL(2,2)*ELRATE)
C  UPDATE TARGET INERTIAL ROLL RATE IN ORBITER BODY COORDINATES
C  FOR DISPLAY.
      SRRTE=-1000.*(TBL(1,1)*AZRATE+TBL(1,2)*ELRATE)
C  UPDATE ANTENNA PITCH ANGLE IN ORBITER BODY COORDINATES FOR DISPLAY.
      SPANG=-ASIN(TBL(1,3))*57.29576
C  UPDATE ANTENNA IN ORBITER BODY COORDINATES FOR DISPLAY.
      IF(TBL(2,3).EQ.0.0.AND.TBL(3,3).EQ.0.0) GO TO 5
      SRANG=-ATAN2(-TBL(2,3),TBL(3,3))*57.29576
      GO TO 7
    5 IF(TBL(1,3).GT.0.0) SRANG=-90.0
```

FIGURE 2.2-11 BASELINE VERSION OF SUBROUTINE ATRACK

PAGE 2

```
      IF(TBL(1,3).LT.0.0) SRANG=90.0
      IF(TBL(1,3).EQ.0.0) STOP
C  RESOLVE POSSIBLE ANGLE AMBIGUITIES, VIZ., -90.<SPANG<90. AND
C  -180.<SRANG<180.
    7 IF(SPANG.LE.90.) GO TO 10
      SPANG=-(180.-ABS(SPANG))*(SPANG/ABS(SPANG))
      SRANG=(180.-ABS(SRANG))*(SRANG/ABS(SRANG))
   10 CONTINUE
C
C  NOTE: DEBUGGING PRINT STATEMENTS.
C     WRITE(6,899)
  899 FORMAT(/' ATRACK DEBUGGGING DATA')
C     WRITE(6,900) ALRATE,BTRATE,AZRATE,ELRATE,SRRTE,SPRTE
C     WRITE(6,901) TBL(1,1),TBL(1,2),TBL(2,1),TBL(2,2)
C     WRITE(6,902) AZDISC,ELDISC,ADSC,EDSC
  900 FORMAT(' ALR,BTR,AZR,ELR,SRR,SPR=',6F10.2)
  901 FORMAT(' TBL 2X2 =',4F10.4)
  902 FORMAT(' AZD,ELD,AD,ED =',4F10.4)
      RETURN
      END
C
```

FIGURE 2.2-11 BASELINE VERSION OF SUBROUTINE ATRACK

PAGE 3

(THIS PAGE INTENTIONALLY LEFT BLANK)

```
C                                                                          00025240
C  ********************************************************               00025250
C  * THIS SUBROUTINE UPDATES AZ AND EL INERTIAL LOS RATES, THE *          00025260
C  * ALPHA AND BETA GIMBAL RATES, THE ALPHA AND BETA GIMBAL    *          00025270
C  * POSITIONS, AND THE TARGET PITCH AND ROLL ANGLES FOR THE   *          00025280
C  * DISPLAY.                                                  *          00025290
C  ********************************************************               00025300
C                                                                          00025310
C                                                                          00025320
      SUBROUTINE ATRACK                                                    00025330
      REAL INTT,K4,K5,K6                                                         00025335
      INTEGER AT1A(10,2),AT1E(10,2),AT2A(10,2),AT2E(10,2)
      COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)                             00025350
      COMMON /INPUT/DUM(6),EWB(3),DUM2(18)                                 00025360
      COMMON /OUTPUT/I1DUM(3),D1DUM(2),SPANG,SRANG,SPRTE,SRRTE,SRSS,       00025370
     2               IDUM1(4),SSALP,SSBET                                          00025380
      COMMON /ICNTL/I2DUM(14),MRNG,MSAM,MPRF,IDUM2(11)                     00025390
      COMMON /SYSDAT/TSAM,DR(3),CP,SP,PSI,PSBIAS,ALBIAS,BTBIAS,            00025400
     2              DUM4(5)                                                00025410
      COMMON /ATDAT/CA,SA,CB,SB,AZRATE,ELRATE,ALRATE,BTRATE,AL,BT,         00025420
     2              DUM3(4)                                                00025430
      COMMON /DSCRM/AZDISC,ELDISC,DUM1(7)                                  00025440
      DIMENSION TX1(3,3),TX2(3,3),TX3(3,3),TBL(3,3)                        00025450
     DIMENSION TDC(3)
C ******************************************************
C
C  ATRACK MODIFIED JAN 28 1986 BY M. MEYER
C  MODIFICATIONS TO SUBROUTINE ATRACK WERE IMPLEMENTED
C  TO UPDATE THE LOOP CONSTANTS AND MORE ACCURATELY
C  SIMULATE THE ACTUAL SIGNAL PROCESSING PERFORMED
C  BY THE RADAR
C
C ******************************************************
C
C------- NEW LOOP CONSTANTS JAN 28 1986-------------------
C
      DATA AT1A/9*5,1,6*13,5,3*1/
      DATA AT1E/9*6,1,6*16,6,2*1,2/
      DATA AT2A/9*407,149,6*662,407,3*149/
      DATA AT2E/9*532,195,6*866,532,3*195/
      DATA K6/3.60E-5/,K4/.0048876/,K5/.236/,DTOR/.0174533/
C
     DATA TDC/0.05122118,0.1195161,0.2561557/
C  DEFINITION:  AT1=KEQ=(WN**2)/(4.*DIFFERENCE PATTERN SLOPE) WHERE       00025490
C                   WN IS NATURAL FREQUENCY OF THE LOOP.                  00025500
C  DEFINITION.  AT2=KEQ*TAU WHERE TAU IS PROPORTIONAL TO STEP RESPONSE    00025510
C                   CONVERGENCE TIME.                                     00025520
C                                                                         00026700
     TCON=TSAM/TDC(MPRF)
C  *****************************************                              00026710
```

PRECEDING PAGE BLANK NOT FILMED

FIGURE 2.2-12 DELIVERABLE VERSION OF SUBROUTINE ATRACK

PAGE 1

```
C  * STEP 1: UPDATE ROUGH RANGE RATE ESTIMATE *                         00026720
C  ******************************************                           00026730
C                                                                       00025530
C  ******************************************************************** 00025540
C  * STEP 1: UPDATE ANTENNA LOS-TO-BODY TRANSFORMATION (NOTE: TRANS- *  00025550
C  *         FORMATION INCLUDES GIMBAL BIAS ERRORS AND RADAR YAW     *  00025560
C  *         ANGLE ERROR WRT BODY FRAME).                            *  00025570
C  ******************************************************************** 00025580
      CALL GAMMA(TX1,-(BT+BTBIAS))                                      00025590
      CALL THETA(TX2,-(AL+ALBIAS))                                      00025600
      CALL MULT33(TX2,TX1,TX3)                                          00025610
      CALL PHI(TX2,-PSI)                                                00025620
      CALL MULT33(TX2,TX3,TBL)                                          00025630
C                                                                       00025640
C  ******************************************************************** 00025650
C  * STEP 2: UPDATE ESTIMATED TARGET INERTIAL AZIMUTH AND ELEVATION *   00025660
C  *         RATES IN ANTENNA LOS FRAME.                            *   00025670
C  ******************************************************************** 00025680
C                                                                       00025690
C  QUANTIZE THE ANGLE DISCRIMINANTS TO 3/16 DB.                         00025700
       IAZDSC=INTT(5.333333*AZDISC*TCON+0.5)/TCON
       IELDSC=INTT(5.333333*ELDISC*TCON+0.5)/TCON                       00025720
       IF(IELDSC.GT.255)IELDSC=255
       IF(IAZDSC.GT.255)IAZDSC=255
       IF(IELDSC.LT.-256)IELDSC=-256
       IF(IAZDSC.LT.-256)IAZDSC=-256
C
C ------ NEW CODE AS OF JAN 28 1986-------------
C
C  UPDATE ESTIMATED TARGET INERTIAL AZIMUTH RATE.                       00025790
       IAZRATE=KSAT(IAZRATE+AT1A(MRNG,IMODE)*IAZDSC)                    00025800
C  UPDATE ESTIMATED TARGET INERTIAL ELEVATION RATE.                     00025810
       IELRATE=KSAT(IELRATE+AT1E(MRNG,IMODE)*IELDSC)                    00025820
C
       AZRATE=K6*DTOR*FLOAT(IAZRATE)
       ELRATE=K6*DTOR*FLOAT(IELRATE)
C
       IALRATE=KSAT(IAZRATE+AT2A(MRNG,IMODE)*IAZDSC)
       IBTRATE=KSAT(IELRATE+AT2E(MRNG,IMODE)*IELDSC)
C
       IF(IALRATE GT.0) THEN
           ALRATE=K4*K5*DTOR*FLOAT(IALRATE/32)
       ELSE
           ALRATE=K4*K5*DTOR*FLOAT((IALRATE-31)/32)
       END IF
C
       IF(IBTRATE.GT.0) THEN
           BTRATE=K4*K5*DTOR*FLOAT(IBTRATE/32)
       ELSE
           BTRATE=K4*K5*DTOR*FLOAT((IBTRATE-31)/32)
       END IF
C
C  **********************************************                       00025840
C  * STEP 3: UPDATE INNER AND OUTER GIMBAL RATES. *                     00025850
C  **********************************************                       00025860
C  COMPUTE REQUIRED COMPONENTS OF ORBITER ANGULAR VELOCITY VECTOR IN    00025870
C  OUTER GIMBAL FRAME                                                   00025880
       WGX=CP*EWB(1)+SP*EWB(2)                                          00025890
       WGY=CA*(-SP*EWB(1)+CP*EWB(2))+SA*EWB(3)                          00025900
       WGZ=-SA*(-SP*EWB(1)+CP*EWB(2))+CA*EWB(3)                         00025910
C  OUTER GIMBAL RATE.                                                   00025920
       IF(ABS(CB).LT.1.0E-6) GO TO 2                                    00025930
       ALRATE=(ALRATE+WGZ*SB)/CB-WGX
       GO TO 4                                                          00025950
```

FIGURE 2.2-12 DELIVERABLE VERSION OF SUBROUTINE ATRACK

PAGE 2

```
    2  ALRATE=0.
    4  CONTINUE
C  INNER GIMBAL RATE.
       BTRATE=BTRATE-WGY
C
C ----- END OF JAN 28 1986 MODIFICATIONS-----------------------
C
C  *************************************************
C  * STEP 4: UPDATE INNER AND OUTER GIMBAL POSITIONS. *
C  *************************************************
C  OUTER GIMBAL POSITION (ALPHA ANGLE)
       AL=AL+TSAM*ALRATE
C  INNER GIMBAL POSITION (BETA ANGLE)
       BT=BT+TSAM*BTRATE
C
C ADD ALPHA AND BETA TO OUTPUT IN DEG
            SSALP=AL*57.29576
            SSBET=BT*57.29576
C  ************************************************************
C  * STEP 6: TRANSFORM TARGET ANGLES AND INERTIAL ANGLE RATES TO *
C  *         BODY FRAME FOR USE IN DISPLAYS AND G AND N.         *
C  ************************************************************
C  NOTE: TRANSFORMATION TBL INCLUDES GIMBAL BIAS ERRORS AND RADAR YAW
C        ANGLE ERROR WRT BODY FRAME.
C  UPDATE TARGET INERTIAL PITCH RATE IN ORBITER BODY COORDINATES
C  FOR DISPLAY.
       SPRTE=-1000.*(TBL(2,1)*AZRATE+TBL(2,2)*ELRATE)
C  UPDATE TARGET INERTIAL ROLL RATE IN ORBITER BODY COORDINATES
C  FOR DISPLAY.
       SRRTE=-1000.*(TBL(1,1)*AZRATE+TBL(1,2)*ELRATE)
C  UPDATE ANTENNA PITCH ANGLE IN ORBITER BODY COORDINATES FOR DISPLAY.
       SPANG=-ASIN(TBL(1,3))*57.29576
C  UPDATE ANTENNA IN ORBITER BODY COORDINATES FOR DISPLAY.
       IF(TBL(2,3).EQ.0.0.AND.TBL(3,3).EQ.0.0) GO TO 5
       SRANG=-ATAN2(-TBL(2,3),TBL(3,3))*57.29576
       GO TO 7
    5  IF(TBL(1,3).GT.0.0) SRANG=-90.0
       IF(TBL(1,3).LT.0.0) SRANG=90.0
       IF(TBL(1,3).EQ.0.0) STOP
C  RESOLVE POSSIBLE ANGLE AMBIGUITIES, VIZ., -90.<SPANG<90. AND
C  -180.<SRANG<180.
    7  IF(SPANG.LE.90.) GO TO 10
       SPANG=-(180.-ABS(SPANG))*(SPANG/ABS(SPANG))
       SRANG=(180.-ABS(SRANG))*(SRANG/ABS(SRANG))
   10  CONTINUE
C
C  NOTE: DEBUGGING PRINT STATEMENTS.
C      WRITE(6,899)
  899  FORMAT(/' ATRACK DEBUGGGING DATA')
C      WRITE(6,900) ALRATE,BTRATE,AZRATE,ELRATE,SRRTE,SPRTE
C      WRITE(6,901) TBL(1,1),TBL(1,2),TBL(2,1),TBL(2,2)
C      WRITE(6,902) AZDISC,ELDISC,IAZDSC,IELDSC
  900  FORMAT(' ALR,BTR,AZR,ELR,SRR,SPR=',6F14.9)
  901  FORMAT(' TBL 2X2 =',4F10.4)
  902  FORMAT(' AZD,ELD,AD,ED =',2F10.4,2I9)
       RETURN
       END
C
```

FIGURE 2.2-12 DELIVERABLE VERSION OF SUBROUTINE ATRACK

PAGE 3

(THIS PAGE INTENTIONALLY LEFT BLANK)

```
************
LINES DELETED FROM BASELINE PROGRAM
   25          REAL INTT,IAZDSC,IELDSC                                         00025335
   26          COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)                        00025350
******
LINES ADDED TO DELIVERABLE PROGRAM
   25          REAL INTT,K4,K5,K6                                                      00025335
   26          INTEGER AT1A(10,2),AT1E(10,2),AT2A(10,2),AT2E(10,2)
   27          COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)                        00025350
************
************
LINES DELETED FROM BASELINE PROGRAM
   36          DIMENSION AT1(10,2),AT2(10,2),TX1(3,3),TX2(3,3),TX3(3,3),TBL(3,3)00025450
   37         DIMENSION TDC(3)
   38          DATA AT1/9*1.5529E-3,2.0106E-4,6*3.9750E-3,1.5529E-3,           00025460
   39        2          3*2.0106E-4/,AT2/9*6.5907E-3,2.3725E-3,                00025470
   40        3          6*1.0546E-2,6.5907E-3,3*2.3725E-3/                     00025480
   41         DATA TDC/0.05122118,0.1195161,0.2561557/
******
LINES ADDED TO DELIVERABLE PROGRAM
   37          DIMENSION TX1(3,3),TX2(3,3),TX3(3,3),TBL(3,3)                   00025450
   38         DIMENSION TDC(3)
   39    C *******************************************************
   40    C
   41    C  ATRACK MODIFIED JAN 28 1986 BY M. MEYER
   42    C  MODIFICATIONS TO SUBROUTINE ATRACK WERE IMPLEMENTED
   43    C  TO UPDATE THE LOOP CONSTANTS AND MORE ACCURATELY
   44    C  SIMULATE THE ACTUAL SIGNAL PROCEESSING PERFORMED
   45    C  BY THE RADAR
   46    C
   47    C ******************************************************
   48    C
   49    C-------- NEW LOOP CONSTANTS JAN 28 1986--------------------
   50  · C
   51          DATA AT1A/9*5,1,6*13,5,3*1/
   52          DATA AT1E/9*6,1,6*16,6,2*1,2/
   53          DATA AT2A/9*407,149,6*662,407,3*149/
   54          DATA AT2E/9*532,195,6*866,532,3*195/
   55          DATA K6/4.58E-5/,K4/.0048876/,K5/.236/,DTOR/.0174533/
   56    C
   57        . DATA TDC/0.05122118,0.1195161,0.2561557/
************
************
LINES DELETED FROM BASELINE PROGRAM
   75          ADSC=0.0431*IAZDSC                                              00025730
   76          EDSC=0.0431*IELDSC                                              00025740
   77    C  UPDATE ESTIMATED TARGET INERTIAL AZIMUTH RATE.                     00025790
   78          AZRATE=AZRATE+TSAM*AT1(MRNG,IMODE)*ADSC                         00025800
   79    C  UPDATE ESTIMATED TARGET INERTIAL ELEVATION RATE.                   00025810
```

PRECEDING PAGE BLANK NOT FILMED

FIGURE 2.2-13 SUMMARY OF MODIFICATIONS TO SUBROUTINE ATRACK

PAGE 1

```
   80           ELRATE=ELRATE+TSAM*AT1(MRNG,IMODE)*EDSC                    00025820
   81    C                                                                 00025830
   82    C ***********************************************                 00025840
******
LINES ADDED TO DELIVERABLE PROGRAM
   91    C
   92    C ------- NEW CODE AS OF JAN 28 1986-------------
   93    C
   94    C  UPDATE ESTIMATED TARGET INERTIAL AZIMUTH RATE.                 00025790
   95           IAZRATE=KSAT(IAZRATE+AT1A(MRNG,IMODE)*IAZDSC)              00025800
   96    C  UPDATE ESTIMATED TARGET INERTIAL ELEVATION RATE.               00025810
   97           IELRATE=KSAT(IELRATE+AT1E(MRNG,IMODE)*IELDSC)              00025820
   98    C
   99           AZRATE=K6*DTOR*FLOAT(IAZRATE)
  100           ELRATE=K6*DTOR*FLOAT(IELRATE)
  101    C
  102           IALRATE=KSAT(IAZRATE+AT2A(MRNG,IMODE)*IAZDSC)
  103           IBTRATE=KSAT(IELRATE+AT2E(MRNG,IMODE)*IELDSC)
  104    C
  105           IF(IALRATE.GT.0) THEN
  106               ALRATE=K4*K5*DTOR*FLOAT(IALRATE/32)
  107           ELSE
  108               ALRATE=K4*K5*DTOR*FLOAT((IALRATE-31)/32)
  109           END IF
  110    C
  111           IF(IBTRATE.GT.0) THEN
  112               BTRATE=K4*K5*DTOR*FLOAT(IBTRATE/32)
  113           ELSE
  114               BTRATE=K4*K5*DTOR*FLOAT((IBTRATE-31)/32)
  115           END IF
  116    C
  117    C ***********************************************                 00025840
************
************
LINES DELETED FROM BASELINE PROGRAM
   92           ALRATE=(AZRATE+AT2(MRNG,IMODE)*ADSC+WGZ*SB)/CB-WGX          00025940
   93           GO TO 4                                                    00025950
******
LINES ADDED TO DELIVERABLE PROGRAM
  127           ALRATE=(ALRATE+WGZ*SB)/CB-WGX
  128           GO TO 4                                                    00025950
************
************
LINES DELETED FROM BASELINE PROGRAM
   97           BTRATE=(ELRATE+AT2(MRNG,IMODE)*EDSC)-WGY                   00025990
   98    C                                                                 00026000
******
LINES ADDED TO DELIVERABLE PROGRAM
  132           BTRATE=BTRATE-WGY
  133    C
  134    C ----- END OF JAN 28 1986 MODIFICATIONS--------------------------
  135    C                                                                 00026000
************
************
LINES DELETED FROM BASELINE PROGRAM
  143    C      WRITE(6,902) AZDISC,ELDISC,ADSC,EDSC                       00026470
  144      900  FORMAT(' ALR,BTR,AZR,ELR,SRR,SPR=',6F10.2)                 00026480
  145      901  FORMAT(' TBL 2X2 =',4F10.4)                                00026490
  146      902  FORMAT(' AZD,ELD,AD,ED =',4F10.4)                          00026500
  147           RETURN                                                     00026510
  148           END                                                        00026520
  149    C                                                                 00024530
  150
******
```

FIGURE 2.2-13 SUMMARY OF MODIFICATIONS TO SUBROUTINE ATRACK

PAGE 2

```
LINES ADDED TO DELIVERABLE PROGRAM
  180   C       WRITE(6,902) AZDISC,ELDISC,IAZDSC,IELDSC                              00026470
  181     900   FORMAT(' ALR,BTR,AZR,ELR,SRR,SPR=',6F14.9)                            00026480
  182     901   FORMAT(' TBL 2X2 =',4F10.4)                                           00026490
  183     902   FORMAT(' AZD,ELD,AD,ED =',2F10.4,2I9)                                    00026500
  184           RETURN                                                                00026510
  185           END                                                                   00026520
  186   C
  187
************

Number of difference sections found: 6
Number of difference records found: 59

DIFFERENCES /IGNORE=()/MERGED=1/OUTPUT=SYS$DISK3:[MCCOLLOUGH]DIFF2.FOR;1-
    SYS$DISK3:[MCCOLLOUGH]ATRACKH.FOR;2-
    SYS$DISK3:[MCCOLLOUGH]ATRACKF.FOR;2
```

FIGURE 2.2-13 SUMMARY OF MODIFICATIONS TO SUBROUTINE ATRACK

PAGE 3

(THIS PAGE INTENTIONALLY LEFT BLANK)

Figure 2.2-14 is a listing of the function KSAT which has been added to the final program. Figure 2.2-15 is a listing of the subroutine READPAT, and Figure 2.2-16 is a listing of the subroutine INTERP. (No original listings or summaries of changes exist because these are new subroutines.)

2.2.4 Integration and Test Data

There were two major sections of code that required testing: the antenna pattern module and the loop filter module. Methods for testing these modules and the test results are summarized in this subsection.

2.2.4.1 Antenna Pattern Module Tests

The subroutines that generate all of the antenna parameter data were written and validated during the study documented in Reference 6. As discussed in Section 2.2.2, these original subroutines were modified by replacing bicubic spline interpolation with two dimensional linear interpolation. After these subroutines were modified, two types of tests were performed to help validate their correctness: a static test and a dynamic test.

The static test of the pattern interpolation routines was to generate three-dimensional plots of all five antenna pattern parameters on an 8 degree by 8 degree grid with a resolution of at least 0.1 degrees and examine the data for any obvious flaws. This task was done using the DISSPLA package on the Building 44 VAX/780 at JSC. An examination of three-dimensional data showed no obvious errors. Unfortunately, we cannot present the data because no high quality hardcopy unit was available.

The second test was a dynamic test. Its purpose was to demonstrate the sign of the slope of the difference patterns was correct and that the general behavior of the pattern interpolation routines in a dynamic environment was satifactory. This test is defined as follows. First, the subroutines were installed in the angle tracking loop simulation program

```
C                                                                       00025230
C  *********************************************************
C  * INTEGER FUNCTION KSAT JAN 28 1986                     *
C  *********************************************************
C
C  THIS FUNCTION CHECKS ATRACK LOOP FOR SATURATION
C
      INTEGER FUNCTION KSAT(K)
C
      IF(K.GE.0) THEN
           KSAT=JMIN0(K,2**15)
      ELSE
           KSAT=JMAX0(K,-2**15)
      END IF
      RETURN
      END
C                                                                       00024530
```

FIGURE 2.2-14 DELIVERABLE VERSION OF SUBROUTINE KSAT

```
      subroutine readPAT
c
c
c            Read in the sum, phase, and difference patterns
c
c

      real a1linear( 41,41 ), e1linear( 41,41 )

      real sa1linear( 41,41 ), se1linear( 41,41 )

      real pa1linear( 41,41 ), pe1linear( 41,41 )

      common / linear / a1linear, e1linear

      common / linear1 / sa1linear, se1linear

      common / linear2 / pa1linear, pe1linear


      open( unit=3, file='[KUBAND.HOWARD.MARK]az1d.dat',
     1       access='sequential', form='unformatted',
     1       status='old', readonly )
      read( 3 ) ( ( a1linear( i,j ), j = 1,41 ), i = 1,41 )
      close( 3 )

      open( unit=3, file='[KUBAND.HOWARD.MARK]el1d.dat',
     1       access='sequential', form='unformatted',
     1       status='old', readonly )
      read( 3 ) ( ( e1linear( i,j ), j = 1,41 ), i = 1,41 )
      close( 3 )

      open( unit=3, file='[KUBAND.HOWARD.MARK]az1s.dat',
     1       access='sequential', form='unformatted',
     1       status='old', readonly )
      read( 3 ) ( ( sa1linear( i,j ), j = 1,41 ), i = 1,41 )
      close( 3 )

      open( unit=3, file='[KUBAND.HOWARD.MARK]el1s.dat',
     1      access='sequential', form='unformatted',
     1      status='old', readonly )
      read( 3 ) ( ( se1linear( i,j ), j = 1,41 ), i = 1,41 )
      close( 3 )

      open( unit=3, file='[KUBAND.HOWARD.MARK]az1p.dat',
     1      access='sequential', form='unformatted',
     1      status='old', readonly )
      read( 3 ) ( ( pa1linear( i,j ), j = 1,41 ), i = 1,41 )
      close( 3 )
```

FIGURE 2.2-15 DELIVERABLE VERSION OF SUBROUTINE READPAT

PAGE 1

```
      open( unit=3, file='[KUBAND.HOWARD.MARK]el1p.dat',
     1      access='sequential', form='unformatted',
     1      status='old', readonly )
      read( 3 ) ( ( pellinear( i,j ), j = 1,41 ), i = 1,41 )
      close( 3 )

      return
      end
c     ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
```

FIGURE 2.2-15 DELIVERABLE VERSION OF SUBROUTINE READPAT

PAGE 2

```
c
c        Subroutine: Antenna pattern interpolation.
c        Input: Azimuth and elevation angles in degrees.
c        Output: Interpolated difference, sum, and phase values
c                for all 18 antenna patterns.
c
c
c        ||||||||||||||||||||||||||||||||||||;||||||||||||||||||||||

         subroutine interp( az, el)

c        ________________________________________________________________
c
c         Linearly interpolate the gain, phase and difference patterns
c.
c        ________________________________________________________________

         real allinear( 41,41 ), ellinear( 41,41 )

         real sallinear(41,41), sellinear(41,41)

         real pallinear(41,41), pellinear(41,41)

         common / linear / allinear, ellinear

         common / linear1 / sallinear,sellinear

         common / linear2 / pallinear,pellinear

         common / SUDIPH / X,Y,Z,PAZ,PEL

         iax = jint( ( az + 4. ) * 5. )
         iex = jint( ( el + 4. ) * 5. )
         az0 = floatj( iax ) / 5. - 4.
         el0 = floatj( iex ) / 5. - 4.

         iaz = jint ( ( az + 4. ) * 5. ) + 1
         jel = jint ( ( el + 4. ) * 5. ) + 1

c        -------- find azd values ---------------------------

         f0 = 10.**( allinear( iaz,jel     ) /20. )
         f1 = 10.**( allinear( iaz+1,jel   ) /20. )
         f2 = 10.**( allinear( iaz,jel+1   ) /20. )
         f3 = 10.**( allinear( iaz+1,jel+1 ) /20. )

         fa = f0 + (f1-f0)/.2 * ( az-az0 )
         fb = f2 + (f3-f2)/.2 * ( az-az0 )
         fx = fa + (fb-fa)/.2 * ( el-el0 )
```

FIGURE 2.2-16 DELIVERABLE VERSION OF SUBROUTINE INTERP

PAGE 1

```
      Y = fx
c     ------- find eld values ---------------------------

      f0 = 10.**( e1llinear( iaz,jel      ) /20. )
      f1 = 10.**( e1llinear( iaz+1,jel    ) /20. )
      f2 = 10.**( e1llinear( iaz,jel+1    ) /20. )
      f3 = 10.**( e1llinear( iaz+1,jel+1  ) /20. )

      fa = f0 + (f1-f0)/.2 * ( az-az0 )
      fb = f2 + (f3-f2)/.2 * ( az-az0 )
      fx = fa + (fb-fa)/.2 * ( el-el0 )


      Z =  fx
c     -------- find azs values --------------------------

      f0 = 10.**(sa1llinear(iaz  ,jel   )/20.)
      f1 = 10.**(sa1llinear(iaz+1,jel   )/20.)
      f2 = 10.**(sa1llinear(iaz  ,jel+1 )/20.)
      f3 = 10.**(sa1llinear(iaz+1,jel+1 )/20.)
      fa = f0 +(f1-f0)/.2*(az-az0)
      fb = f2 +(f3-f2)/.2*(az-az0)
      fx = fa +(fb-fa)/.2*(el-el0)


      X = fx
c     ------- find azp values ----------------------------

      f0 = pa1llinear(iaz  ,jel   )
      f1 = pa1llinear(iaz+1,jel   )
      f2 = pa1llinear(iaz  ,jel+1 )
      f3 = pa1llinear(iaz+1,jel+1 )
      fa = f0 +(f1-f0)/.2*(az-az0)
      fb = f2 +(f3-f2)/.2*(az-az0)
      fx = fa +(fb-fa)/.2*(el-el0)

      PAZ=fx                   !  phase in degrees
c     ------- find elp values ---------------------------

      f0 = pe1llinear(iaz  ,jel   )
      f1 = pe1llinear(iaz+1,jel   )
      f2 = pe1llinear(iaz  ,jel+1 )
      f3 = pe1llinear(iaz+1,jel+1 )

      fa = f0 +(f1-f0)/.2*(az-az0)
      fb = f2 +(f3-f2)/.2*(az-az0)
      fx = fa +(fb-fa)/.2*(el-el0)

      PEL=fx                   ! phase in degrees

      return

      end
```

FIGURE 2.2-16 DELIVERABLE VERSION OF SUBROUTINE INTERP

PAGE 2

documented in Reference 6. Then the program was run with a 0 dBsm target, fixed at one nautical mile range for 100 seconds. Next, the original program with bicubic spline interpolation was run with the same scenario. Then, the statistical aspects of the output time histories of the four parameters, azimuth, elevation, azimuth rate, and elevation rate, were compared. Results of the comparison showed that the differences in performance was negligible for all four parameters. These results confirm that the two dimensional linear interpolation antenna pattern model has been implemented properly in a dynamic environment.

Once initial tests were completed and the subroutines changes were validated using the simulation program of Reference 6, these modified routines were then lifted as a unit from this program and installed in the SES simulation code. Additional tests were run on the modified SES code to futher validate the pattern changes and other loop changes. These tests and their results are described in the following subsection.

2.2.4.2 Loop Filter Module Tests

The loop filter module changes were validated using various qualitative tests. The purpose of these tests were to verify (1) the proper slope sign in the difference patterns, (2) the noise properties of the tracking loop, and (3) the transient response of the tracking loop filter.

A very simple test was used to verify the slope sign in the difference patterns. A stationary, 0 dBsm target was placed at one nautical mile and was tracked by the simulation for a period of 100 seconds. Results of this test showed that the target was tracked in a stable, steady-state fashion in both angle and angle rate, ensuring that the sign of the difference pattern slope was correct.

The next set of tests established the statistical properties of the angle tracking loop. That is, these tests established the approximate noise bandwidth of the loop. Again a 0 dBsm, stationary point target was tracked at the following three ranges: 1 nautical mile, 3 nautical miles, and

5 nautical miles. These ranges were selected to exercise the three different loop bandwidths of the tracker. Tables 2.2-3 and 2.2-4 summarize the results of these tests for the original SES simulation code and the modified simulation code. A comparison of the means and the standard deviations of the old and the new version show a fairly reasonable matchup of the data. This is significant proof that the angle tracker upgrades have been installed properly. The fact that the modified version is nosier than the original version is encouraging since this brings the angle tracking loop simulation data into better agreement with the flight data.

This brings us naturally to our third set of angle tracking loop tests. These tests involved injecting trajectories from the SORTE experiment and the Palapa B rendezvous into the simulation and comparing the simulation predictions with the actual radar data in each case. Discussions of the results of these simulation experiments are delayed until sections 3.7 and 4.2, respectively. The purpose of these tests was to determine the overall fidelity of the angle and ILOS angle rate trackers in terms of random properties and transient response properties for typical rendezvous situations.

2.3 AGC UPGRADES

2.3.1 PROBLEM DEFINITION

The AGC module discussed in this section includes three components: (1) calculation of the AGC update, (2) calculation of the RSS, and (3) calculation of the A/D saturation effects (if any). High fidelity models for each of these components were defined and discussed in detail in the final report for NASA Contract No. NAS9-15840 (Reference 4). However, an examination of the simulation code published in the appendix of that report revealed that these components had not been completely upgraded in several areas. The most serious problem with this less accurate model is a 12 dB discontinuous jump in AGC, and therefore RSS, accompanying a transition in the sample rate under normal target conditions (greater than a 0 dBsm Radar Cross Section (RCS)). This effect is demonstrated in Figures 2.3-1 and 2.3-2. Now, under normal target conditions, theory and actual operational data show that there is no such discontinuity at this transition.

TABLE 2.2-3 A COMPARISON OF THE STANDARD DEVIATIONS OF THE ANGLE TRACKING PERFORMANCE FOR THE OLD AND THE NEW SIMULATION MODELS

RANGE	6000 FEET		18000 FEET		30000 FEET	
VERSION	OLD	NEW	OLD	NEW	OLD	NEW
Roll Angle	5.27 E-3	6.8 E-3	4.94 E-3	6.5 E-3	3.58 E-3	4.9 E-3
Pitch Angle	5.74 E-3	5.86 E-3	4.93 E-3	4.8 E-3	3.6 E-3	4.4 E-3
ILOS Roll Rate	1.58 E-3	1.9 E-3	9.16 E-4	1.18 E-3	2.38 E-4	5.04 E-4
ILOS Pitch Rate	1.26 E-3	1.37 E-3	6.84 E-4	8.6 E-4	1.8 E-4	4.72 E-4

TABLE 2.2-4 A COMPARISON OF THE MEANS OF THE ANGLE TRACKING PERFORMANCE FOR THE OLD AND THE NEW SIMULATION MODELS

RANGE	6000 FEET		18000 FEET		30000 FEET	
VERSION	OLD	NEW	OLD	NEW	OLD	NEW
Roll Angle	1.88 E-6	1.39 E-2	-6.36 E-4	-1.29 E-2	-1.17 E-4	-1.36 E-2
Pitch Angle	3.09 E-4	1.8 E-2	6.2 E-5	1.9 E-2	1.27 E-5	1.98 E-2
ILOS Roll Rate	1.26 E-5	1.95 E-4	-7.85 E-7	2.9 E-4	1.20 E-5	2.6 E-4
ILOS Pitch Rate	2.46 E-5	-2.59 E-4	1.10 E-5	-3.1 E-4	1.20 E-5	2.9 E-4

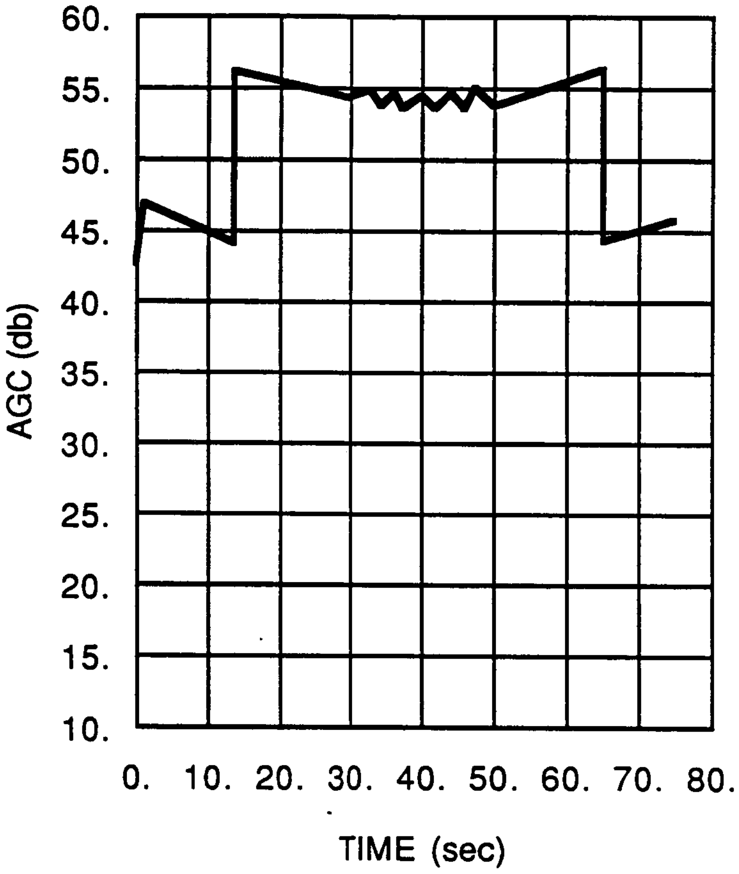


FIGURE 2.3-1 AGC PROFILE FOR THE RANGE PROFILE GIVEN IN FIGURE 2.3-2. DISCONTINUITIES OCCUR AT THE SAMPLE RATE TRANSITION.

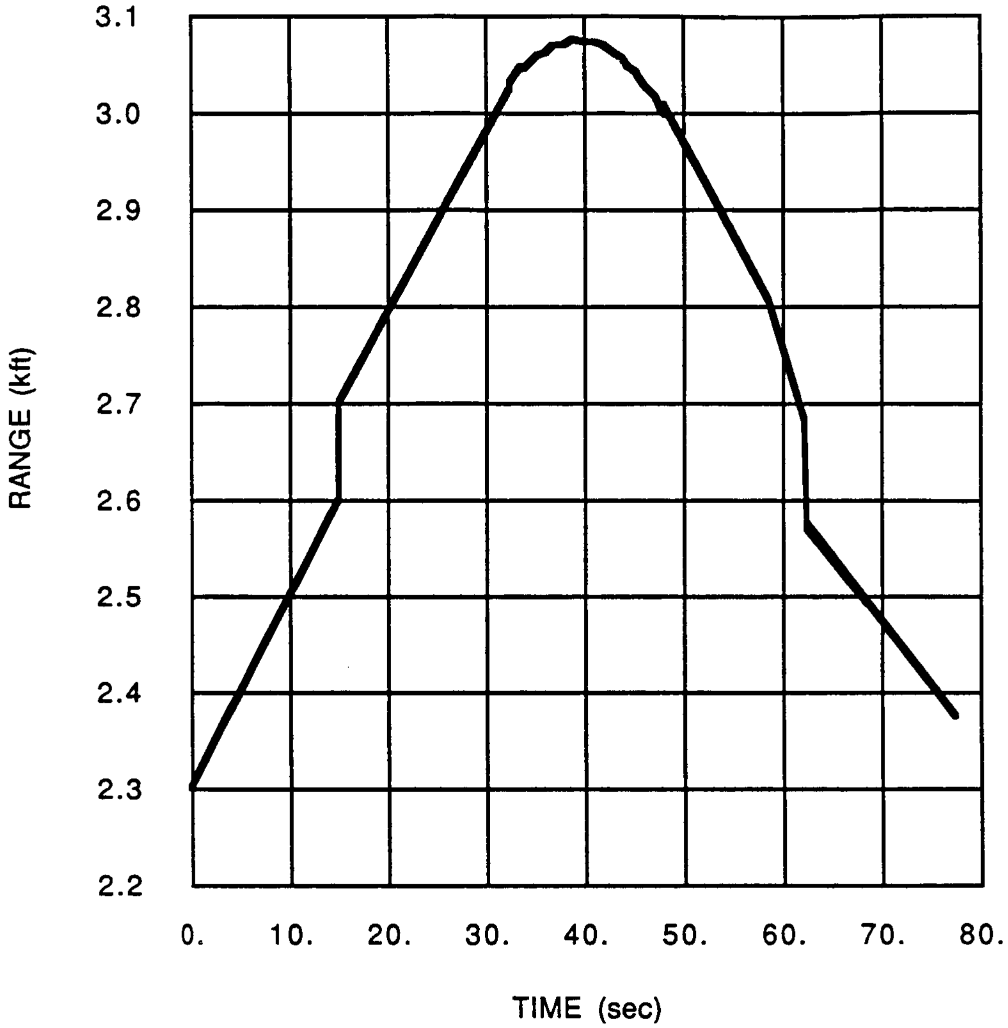


FIGURE 2.3-2 RANGE PROFILE USED TO GENERATE AGC PROFILE OF FIGURE 2.3-1

The purpose of this subsection is to point out the weaknesses of the current SES simulation code in these areas and define the corrections.

2.3.2 Definition of Algorithm Modifications

2.3.2.1 AGC Model Improvements

To facilitate a description of the weaknesses in the baseline version of the AGC model, a concise definition of the high fidelity model (from Reference 4) is provided below.

The upgraded AGC model includes the following features:

1. The AGC increment for the next data cycle is determined by subtracting the mean signal level at the log converter output (see Figure 2.3-3) from a prestored value which represents a signal power of $4q^2$ at the A/D input.
2. It includes the effects of quantization noise injected by the A/D converter.
3. It allows a maximum of 10 dB increment in AGC or a minimum of -10 dB decrement in AGC per data cycle.
4. The absolute AGC value cannot drop below 6 dB, the nominal search AGC value.

A crude A/D converter saturation model has been implemented in conjunction with this model to increase AGC response fidelity in anticipation of large, sudden increases in satellite RCS values.

The AGC algorithm can be summarized as follows:

Step 1: Compute the AGC change, ΔAGC, based on the present mean signal level estimate at the log converter output.

Step 2: If ΔAGC $\geq$ 10 dB, then ΔAGC = 10 dB, or if ΔAGC $\leq$ -10 dB, then ΔAGC = -10 dB.

Step 3: Compute the new AGC.

Step 4: If new AGC $\leq$ 6 dB, then new AGC = 6 dB.

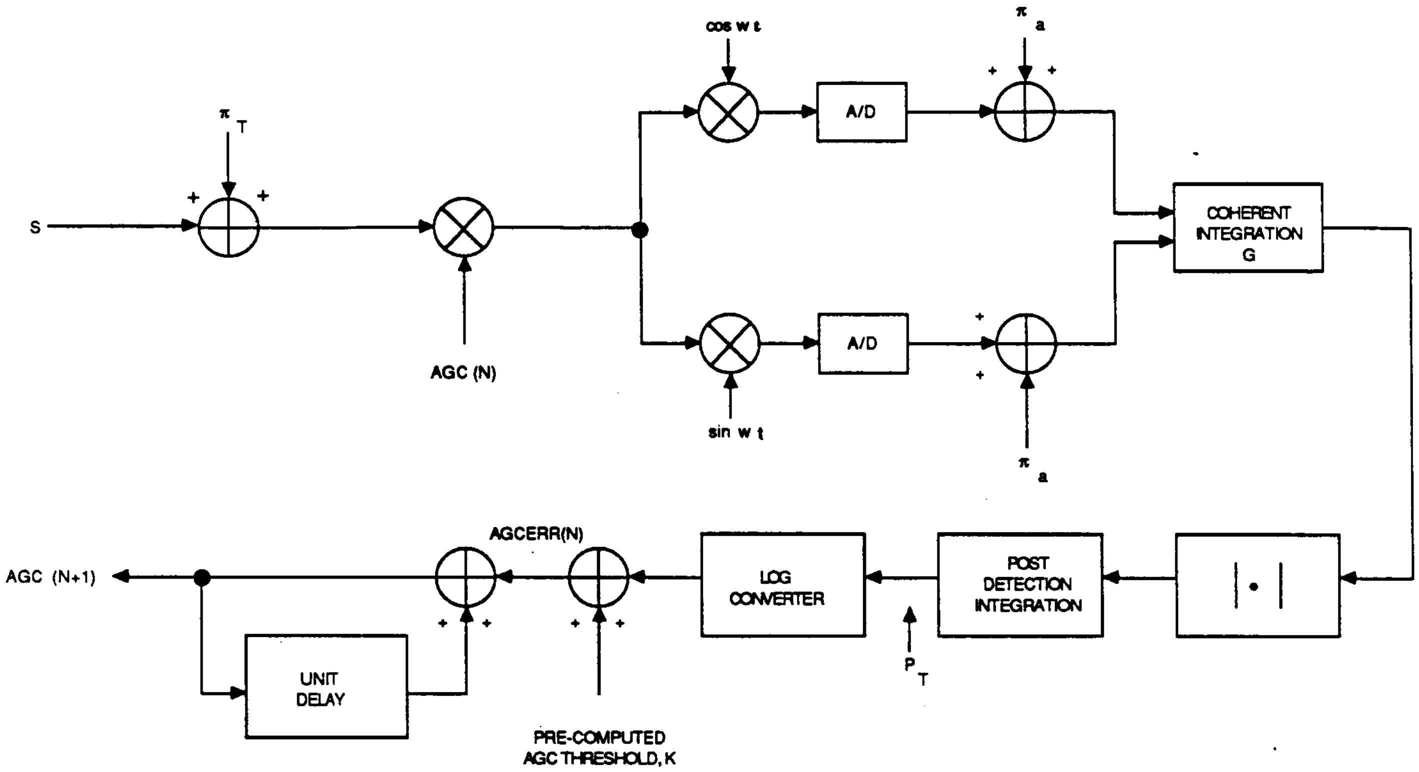


FIGURE 2.3-3 SIMPLIFIED DIAGRAM OF THE AGC TRACKING LOOP

Computation of the change in AGC, AGC, is done using the following expression.

(2-9) $AGCERR(N) = k_1G/(AGC(N)(SNR_{DT}(N)+1)+k_2)$

where $G =$ Signal-to-noise power ratio (SNR) gain from the A/D output to the doppler filter output,

$SNR_{DT} =$ Signal-to-thermal noise power ratio at the doppler filter output,

$k_1 = (2q)^2/N_t$,
$k_2 = (q)^2/(12N_t)$
$N_t =$ unAGC'd thermal noise power at the A/D input.

The updated AGC value is computed with the expression

(2-10) $AGC(N+1)=AGC(N)AGCERR(N)$

k_1 can be interpreted as the ratio of the desired AGC'd track signal power level at the A/D input to unAGC'd thermal noise power level at the A/D input, k_2 is interpreted as the ratio of the quantization noise power, $q^2/12$, to the unAGC'd thermal noise power at the A/D input. Finally, to be consistent with the baseline code, we will set $G = 4\ P_s$. The values for k_1, k_2, G and P_s for the various modes and range intervals are summarized in Table 2.3-1.

Some comments on the accuracy of this algorithm versus actual AGC operation are in order. We first note tha the form for predicting the AGC change given in Equation (2-9) is quite accurate. It has the A/D quantization noise and the noise floor concept folded into the calculation. As noted earlier, the quantization noise includes only the contribution from the A/D converter and is assumed to have a power of $q^2/12$ where q represents the voltage of a single A/D step. All other quantization noise sources are dwarfed in comparison to this source, especially when comparing their relative

TABLE 2.3-1 AGC CALCULATION CONSTANTS

Range Interval, Ft.	N_t, q^2	k_1	k_2	G	P_s
Passive					
<2560	124.80	0.0321	0.00067	16	4
(2560, 5750)	7.84	0.51	0.011	16	4
(5760, 11510)	7.84	0.51	0.011	8	2
(11510, 23030)	7.84	0.51	0.011	16	4
(23040, 43510)	7.84	0.51	0.011	32	8
>43510	7.84	0.51	0.011	64	16
Active					
<49910	124.8	0.0321	0.00067	16	4
>49920	7.84	0.51	0.011	8	2

effects at the doppler filter output. The search thermal noise AGC value or the "noise floor" in this expression is fixed at 6 dB. This floor represents the search AGC value at the time the target is detected. In reality, this number is a random process, fluctuating from acquisition to acquisition. However, we treat the noise floor as a deterministic value and assign it a value equal to the mean of the random process, i.e., 6 dB for all acquisitions.

The following errors were found in the baseline code and corrected. First, and most serious, k_1 and k_2 did not change at the sample rate transition as shown in Table 2.3-1, causing the 12 dB discontinuity in AGC as discussed earlier. In the baseline program k_1 and k_2 assumed the low sample rate values at all ranges. Secondly, the AGC was allowed to drop to 0 dB rather than limit at the nominal search AGC level of 6 dB. Both of these errors were corrected in the upgraded simulation code documented in Section 2.3.3.

2.3.2.2 RSS Model Improvements

The Ku-Band Radar computes the RSS using the following very simple relation.

$$\text{RSS}(N) = (10 \log (1/\text{AGC}(N))-6)k_o \tag{2-11}$$

where AGC(N) is the latest estimate of AGC and the value 6 represents the nominal search AGC value (or search "noise floor"). The value of k_o is 5 volts/160 dB which converts the RSS from dB to voltage from the display meter. Full scale AGC is 160 dB which corresponds to a full scale meter voltage of 5 volts. Since the AGC is not allowed to drop below 6 dB, the RSS will not drop below 0 volts.

In the baseline code for the RSS module there were two errors: (1) the nominal search AGC value was set to 0 dB, and (2) the scale factor k_o was ignored. Both corrections have been made in the present version of the RSS as documented in Section 2.3.3.

2.3.2.3 A/D Saturation Noise Model Improvements

A simple model for injecting A/D saturation effects into the tracking signal response was developed in anticipation of encountering sudden, large increases in receive signal strength when rendezvousing with various satellite targets. The model is fairly crude and is based on the concept that the total signal-plus-noise power at the A/D output should be limited to $(7q)^2$. The basic idea of the model can be expressed as follows:

Step 1: Compute the signal-plus-noise power at the A/D input.

Step 2: If the total power is greater than $(7q)^2$, then limit this power to (7q).

The total signal-plus-noise power at the A/D input is computed using the expression, (2-12) $\text{Total Power} = AGC(N)N_t(SNR_{DT}(N)/G+1)$

where SNR_{DT}/G is equivalent to SNR_{vt}, the signal-to-thermal noise power ratio at the A/D input. SNR_{vt} is represented in this form because it is not easy to compute directly within the simulation, while SNR_{DT} and G are easily accessed. Hence, the indirect form of the calculation is used.

In the computer simulation code all powers are normalized to the unAGC'd thermal noise power at the ADC input. So the implementation of the saturation noise model is given by the inequality.

(2-13) $AGC(N)(SNR_{DT}(N)/G+1) \leq (7q)^2/N_t$

where $(7q)^2$ is the maximum total power at the ADC output and N_t represents the unAGC'd thermal noise power $(2.8q)^2$ in the low sample rate mode and $(11.2q)^2$ in the high sample rate mode).

There are two errors in the baseline version of the saturation noise model code:

(1) The value for N_t in the low sample rate case is $(1.4q)^2$ rather than $(2.8q)^2$, and

(2) The value for N_t in the high sample rate mode is $(1.4q)^2$ rather than $(11.2q)^2$. The errors have been corrected in the final version of the code and are documented in Section 2.3.3.

2.3.3 Software Design Documentation

The simulation changes documented in Section 2.3.2 affect these subroutines in the baseline code: (1) RSS, (2) SATNSE, and (3) DISCRM.

Three lines of code were changed in the baseline version of RSS shown in Figure 2.3-4. These included:

(1) Changing the AGCERR computation to properly reflect the sample rate transition

```
C                                                                       00029230
C ***************************************************************       00029240
C * THIS SUBROUTINE COMPUTES THE RADAR SIGNAL STRENGTH AND UPDATES *    00029250
C * THE AGC SETTING.                                               *    00029260
C ***************************************************************       00029270
C                                                                       00029280
C                                                                       00029290
      SUBROUTINE RSS                                                    00029300
      COMMON /CNTL/IPWR,IMODE,IDUM1(7),DUM1(3)                          00029310
      COMMON /ICNTL/IDUM2(14),MRNG,IDUM6(12)                            00029320
      COMMON /OUTPUT/IDUM7(3),DUM3(6),SRSS,IDUM4(4)                     00029330
      COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD                           00029340
      DIMENSION PS(10,2)                                                00029350
      DATA PS/9*1.,2.,5*1.,2.,4.,8.,8.,16./,QNV/0.04166666/             00029360
C                                                                       00029370
C ***************************                                           00029380
C * STEP 1: UPDATE SYSTEM AGC *                                         00029390
C ***************************                                           00029400
C                                                                       00029410
C STEP 1-1: COMPUTE AGC ERROR AND CHECK LIMITS.                         00029420
      AGCERR=4.*PS(MRNG,IMODE)/(AGCO*(SNRDT+1.0)+QNV)                   00029430
      IF(AGCERR.GT.10.) AGCERR=10.0                                     00029440
      IF(AGCERR.LT.0.1) AGCERR=0.1                                      00029450
C                                                                       00029460
C STEP 1-2: COMPUTE NEW AGC VALUE AND CHECK LIMITS.                     00029470
      AGCO=AGCERR*AGCO                                                  00029480
      IF(AGCO.GT.1.0) AGCO=1.0                                          00029490
      AGCODB=10.*ALOG10(AGCO)                                           00029500
C                                                                       00029510
C ******************************************                            00029520
C * STEP 2: UPDATE RADAR SIGNAL STRENGTH VALUE *                        00029530
C ******************************************                            00029540
      IF(AGCO.LT.1.0E-15) AGCO=1.0E-15                                  00029550
      SRSS=1./AGCO                                                      00029560
      SRSS=10.*ALOG10(SRSS)                                             00029570
      RETURN                                                            00029580
      END                                                               00029590
C                                                                       00026530
```

FIGURE 2.3-4 BASELINE VERSION OF SUBROUTINE RSS

(2) Adding the 6 dB AGC floor level, and

(3) Subtracting 6 dB from the RSS computation.

Figures 2.3-5 gives the new version of RSS and Figure 2.3-6 provides a summary of the changes.

Two lines of code were changed in the baseline version of SATNSE shown in Figure 2.3-7. The ratio of $(7q)^2/N_t$ was changed from 12.25 to 6.25 for the low sample rate mode. Also, the array PS was updated to reflect the values given in Table 2.3-1 for the various range intervals. The new version of SATNSE is listed in Figure 2.3-8 and a summary of the changes is provided in Figure 2.3-9.

Three major changes were made in the baseline version of subroutine DISCRM shown in Figure 2.3-10. Two changes involved altering values of constants. The values of the array PS were updated to those of Table 2.3-1. The constant QNV was converted to function of the sample rate and its values were computed appropriately. The updated version of DISCRM is given in Figure 2.3-11 and the changes are summarized in Figure 2.3-12.

2.3.4 Integration and Test Data

Accurate Radar Signal Strength (RSS) simulation outputs was one of the major objectives of the AGC update. With this objective in mind, the tests performed were to validate the RSS ouput. Since the RSS is a function of the AGC, a proper RSS output would validate the AGC modifications.

2.3.4.1 Test Definition

A test trajectory was constructed where the target originated at a 2400 ft range, moved out to about 4000 ft and then closed to 1400 ft thus moving the simulated target through two sample rate changes. Two simulation runs of this trajectory were made with the radar cross section (RCS) set to 10d Bsm. A 10 dBsm target was chosen since that is a common actual target RCS. One simulation run was made with the RCS set to -40 dBsm. This RCS was chosen because one could predict a discontinuity in RSS at the sample rate changes from the equations and it was desirable to see if the simulation also produced this result.

```
C                                                                       00029230
C ******************************************************************* 00029240
C * THIS SUBROUTINE COMPUTES THE RADAR SIGNAL STRENGTH AND UPDATES *   00029250
C * THE AGC SETTING.                                                *   00029260
C ******************************************************************* 00029270
C                                                                       00029280
C                                                                       00029290
      SUBROUTINE RSS                                                    00029300
      COMMON /CNTL/IPWR,IMODE,IDUM1(7),DUM1(3)                          00029310
      COMMON /ICNTL/IDUM2(14),MRNG,MSAM,IDUM6(11)
      COMMON /OUTPUT/IDUM7(3),DUM3(6),SRSS,IDUM4(4)                     00029330
      COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD                           00029340
      DIMENSION PS(10,2),QNV(2),A1(2)
      DATA PS/9*4.,2.,5*4.,2.,4.,8.,8.,16./
      DATA QNV/.00067,.011/,A1/.0321,.51/
C ******************************************************
C SUBROUITNE RSS HAS BEEN UPDATED TO CORRESPOND TO THE
C DERIVATION OF AGCERR PRESENTED IN THE FINAL REPORT ON
C KUBAND COMPUTER SIMULATION. M. MEYER FEB 17, 1986
C ******************************************************
C                                                                       00029370
C *****************************                                         00029380
C * STEP 1: UPDATE SYSTEM AGC *                                         00029390
C *****************************                                         00029400
C                                                                       00029410
C STEP 1-1: COMPUTE AGC ERROR AND CHECK LIMITS.
C --------UPDATED FEB 17, 1986------------------------------------------
      AGCERR=A1(MSAM)*4.*PS(MRNG,IMODE)/(AGCO*(SNRDT+1.0)+QNV(MSAM))
      IF(AGCERR.GT.10.) AGCERR=10.0                                     00029440
      IF(AGCERR.LT.0.1) AGCERR=0.1                                      00029450
C                                                                       00029460
C STEP 1-2: COMPUTE NEW AGC VALUE AND CHECK LIMITS.                     00029470
      AGCO=AGCERR*AGCO                                                  00029480
C --------UPDATED FEB 17, 1986-----------------------------------------
      IF(AGCO.GT.0.25) AGCO=0.25
      AGCODB=10.*ALOG10(AGCO)                                           00029500
C                                                                       00029510
C **********************************************                        00029520
C * STEP 2: UPDATE RADAR SIGNAL STRENGTH VALUE *                        00029530
C **********************************************                        00029540
      IF(AGCO.LT.1.0E-15) AGCO=1.0E-15                                  00029550
      SRSS=1./AGCO
C -------UPDATED FEB 17, 1986------------------------------------------
      SRSS=10.*ALOG10(SRSS)-6.0
      RETURN                                                            00029580
      END                                                               00029590
C                                                                       00026530
```

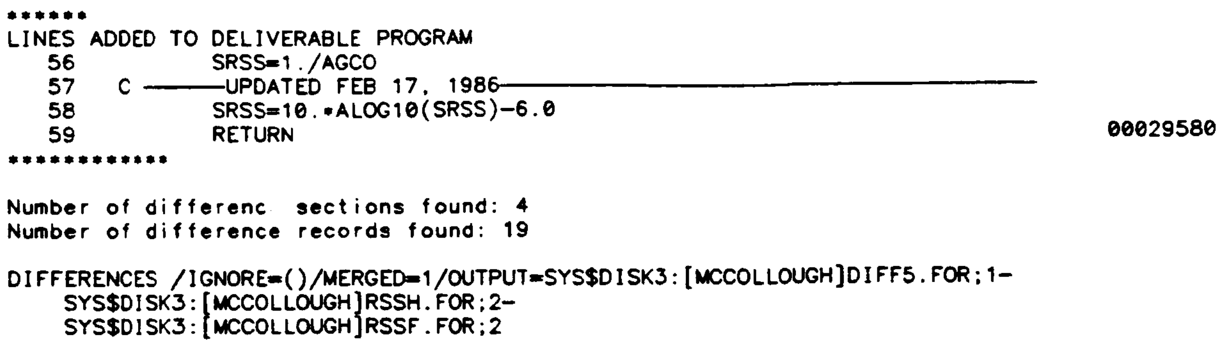
FIGURE 2.3-5 DELIVERABLE VERSION OF SUBROUTINE RSS

(THIS PAGE INTENTIONALLY LEFT BLANK)

```
************
LINES DELETED FROM BASELINE PROGRAM
   24          COMMON /ICNTL/IDUM2(14),MRNG,IDUM6(12)                              00029320
   25          COMMON /OUTPUT/IDUM7(3),DUM3(6),SRSS,IDUM4(4)                       00029330
   26          COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD                             00029340
   27          DIMENSION PS(10,2)                                                  00029350
   28          DATA PS/9*1.,2.,5*1.,2.,4.,8.,8.,16./,QNV/0.04166666/               00029360
   29    C                                                                         00029370
******
LINES ADDED TO DELIVERABLE PROGRAM
   24          COMMON /ICNTL/IDUM2(14),MRNG,MSAM,IDUM6(11)
   25          COMMON /OUTPUT/IDUM7(3),DUM3(6),SRSS,IDUM4(4)                       00029330
   26          COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD                             00029340
   27          DIMENSION PS(10,2),QNV(2),A1(2)
   28          DATA PS/9*4.,2.,5*4.,2.,4.,8.,8.,16./
   29          DATA QNV/.00067,.011/,A1/.0321,.51/
   30    C ***************************************************
   31    C SUBROUITNE RSS HAS BEEN UPDATED TO CORRESPOND TO THE
   32    C DERIVATION OF AGCERR PRESENTED IN THE FINAL REPORT ON
   33    C KUBAND COMPUTER SIMULATION. M. MEYER FEB 17, 1986
   34    C ***************************************************
   35    C                                                                         00029370
************
************
LINES DELETED FROM BASELINE PROGRAM
   34    C  STEP 1-1: COMPUTE AGC ERROR AND CHECK LIMITS.                          00029420
   35          AGCERR=4.*PS(MRNG,IMODE)/(AGCO*(SNRDT+1.0)+QNV)                     00029430
   36          IF(AGCERR.GT.10.) AGCERR=10.0                                       00029440
******
LINES ADDED TO DELIVERABLE PROGRAM
   40    C  STEP 1-1: COMPUTE AGC ERROR AND CHECK LIMITS.
   41    C --------UPDATED FEB 17, 1986---------------------------------
   42          AGCERR=A1(MSAM)*4.*PS(MRNG,IMODE)/(AGCO*(SNRDT+1.0)+QNV(MSAM))
   43          IF(AGCERR.GT.10.) AGCERR=10.0                                       00029440
************
************
LINES DELETED FROM BASELINE PROGRAM
   41          IF(AGCO.GT.1.0) AGCO=1.0                                            00029490
   42          AGCODB=10.*ALOG10(AGCO)                                             00029500
******
LINES ADDED TO DELIVERABLE PROGRAM
   48    C --------UPDATED FEB 17, 1986--------------------------------
   49          IF(AGCO.GT.0.25) AGCO=0.25
   50          AGCODB=10.*ALOG10(AGCO)                                             00029500
************
************
LINES DELETED FROM BASELINE PROGRAM
   48          SRSS=1./AGCO                                                        00029560
   49          SRSS=10.*ALOG10(SRSS)                                               00029570
   50          RETURN                                                              00029580
```

FIGURE 2.3-6 SUMMARY OF MODIFICATIONS TO SUBROUTINE RSS

PAGE 1



```
******
LINES ADDED TO DELIVERABLE PROGRAM
   56          SRSS=1./AGCO
   57    C -------UPDATED FEB 17, 1986----------------------------------
   58          SRSS=10.*ALOG10(SRSS)-6.0
   59          RETURN                                                      00029580
************

Number of differenc  sections found: 4
Number of difference records found: 19

DIFFERENCES /IGNORE=()/MERGED=1/OUTPUT=SYS$DISK3:[MCCOLLOUGH]DIFF5.FOR;1-
    SYS$DISK3:[MCCOLLOUGH]RSSH.FOR;2-
    SYS$DISK3:[MCCOLLOUGH]RSSF.FOR;2
```

FIGURE 2.3-6 SUMMARY OF MODIFICATIONS TO SUBROUTINE RSS

PAGE 2

```
C
C  **********************************************************
C  * THIS SUBROUTINE DETERMINES WHETHER THE SIGNAL PLUS NOISE *
C  * IS SATURATING THE A/D ——  IF SO,THEN THE SNR AT DOPPLER  *
C  * FILTER OUTPUT IS LIMITED TO THE VALUE THAT JUST SATUR-   *
C  * ATES THE A/D.                                           *
C  **********************************************************
C
C
      SUBROUTINE SATNSE(SNF)
      COMMON /CNTL/IPWR,IMODE
      COMMON /ICNTL/IDUM(14),MRNG
      COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD
      DIMENSION PS(10,2)
      DATA PS/9*10.0,2.,5*1.,2.,4.,8.,8.,16./
      SNF=1.
      X=AGCO*(SNRDT/(4.*PS(MRNG,IMODE))+1.0)
      X=12.25/X
      IF(X.GT.1) RETURN
      SNF=X
      RETURN
      END
C
```

FIGURE 2.3-7 BASELINE VERSION OF SUBROUTINE SATNSE

```
C                                                                       00035540
C  ************************************************************        00035550
C  * THIS SUBROUTINE DETERMINES WHETHER THE SIGNAL PLUS NOISE *        00035560
C  * IS SATURATING THE A/D -- IF SO,THEN THE SNR AT DOPPLER   *        00035570
C  * FILTER OUTPUT IS LIMITED TO THE VALUE THAT JUST SATUR-   *        00035580
C  * ATES THE A/D.                                            *        00035590
C  ************************************************************        00035600
C                                                                       00035610
C                                                                       00035620
      SUBROUTINE SATNSE(SNF)                                            00035630
      COMMON /CNTL/IPWR,IMODE                                           00035640
      COMMON /ICNTL/IDUM(14),MRNG                                       00035650
      COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD                           00035660
      DIMENSION PS(10,2)
C
C ------- PS VALUES WERE UPDATED FEB 17,1986 BY M. MEYER----------
C
      DATA PS/9*4.0,2.,5*4.,2.,4.,8.,8.,16./
      SNF=1.                                                            00035690
      X=AGCO*(SNRDT/(4.*PS(MRNG,IMODE))+1.0)                            00035700
C  ******************************************************
C  X=12.25/X WAS REPLACED BY X=6.25/X TO MORE ACCURATELY
C  REFLECT A/D SATURATION BY M. MEYER FEB 17, 1986
C  ******************************************************
      X=6.25/X
      IF(X.GT.1) RETURN                                                 00035720
      SNF=X                                                             00035730
      RETURN                                                            00035740
      END                                                               00035750
C                                                                       00012670
```

FIGURE 2.3-8 DELIVERABLE VERSION OF SUBROUTINE SATNSE

ORIGINAL PAGE IS
OF POOR QUALITY

```
************
LINES DELETED FROM BASELINE PROGRAM
   28          DIMENSION PS(10,2)                                              00035670
   29          DATA PS/9*10.0,2.,5*1.,2.,4.,8.,8.,16./                         00035680
   30          SNF=1.                                                          00035690
   31          X=AGCO*(SNRDT/(4.*PS(MRNG,IMODE))+1.0)                          00035700
   32          X=12.25/X                                                       00035710
   33          IF(X.GT.1) RETURN                                               00035720
******
LINES ADDED TO DELIVERABLE PROGRAM
   28          DIMENSION PS(10,2)
   29    C
   30    C ------- PS VALUES WERE UPDATED FEB 17,1986 BY M. MEYER-----------
   31    C
   32          DATA PS/9*4.0,2.,5*4.,2.,4.,8.,8.,16./
   33          SNF=1.                                                          00035690
   34          X=AGCO*(SNRDT/(4.*PS(MRNG,IMODE))+1.0)                          00035700
   35    C ****************************************************
   36    C X=12.25/X WAS REPLACED BY X=6.25/X TO MORE ACCURATELY
   37    C REFLECT A/D SATURATION BY M. MEYER FEB 17, 1986
   38    C ****************************************************
   39          X=6.25/X
   40          IF(X.GT.1) RETURN                                               00035720
************

Number of difference sections found: 1
Number of difference records found: 12

DIFFERENCES /IGNORE=()/MERGED=1/OUTPUT=SYS$DISK3:[MCCOLLOUGH]DIFF3.FOR;1-
    SYS$DISK3:[MCCOLLOUGH]SATNSEH.FOR;2-
    SYS$DISK3:[MCCOLLOUGH]SATNSEF.FOR;2
```

FIGURE 2.3-9 SUMMARY OF MODIFICATIONS OF SUBROUTINE SATNSE

(THIS PAGE INTENTIONALLY LEFT BLANK)

```
C                                                                     00022720
C  ****************************************************************  00022730
C  * THIS SUBROUTINE ADDS THE EQUIVALENT NOISE TO THE ANGLE, RANGE, *  00022740
C  * VELOCITY AND ON-TARGET DISCRIMINANT COMPONENTS AND THEN COM-   *  00022750
C  * PUTES THE ANGLE, RANGE, VELOCITY, AND ON-TARGET DISCRIMINANTS. *  00022760
C  ****************************************************************  00022770
C                                                                     00022780
C                                                                     00022790
      SUBROUTINE DISCRM                                               00022800
      REAL LATE,MEAN                                                  00022805
      COMMON /OUTPUT/MSWF,MTF,MSF,SRNG,SRDOT,SPANG,SRANG,SPRTE,
     2               SRRTE,SRSS,MADVF,MRDVF,MARDVF,MRRDVF
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)              00022810
      COMMON /ICNTL/I3DUM(14),MRNG,MSAM,MPRF,IDUM4(10)                00022820
      COMMON /SYSDAT/TSAM,DR(3),CP,SP,PSI,PSBIAS,ALBIAS,BTBIAS,GP,GA, 00022830
     2               DUMS(3)                                          00022840
      COMMON /TGTDAT/NT,DUM5(506),CGRNGE,CGVEL                        00022850
      COMMON /DSCRM/AZDISC,ELDISC,RDISC,VDISC,RRTE,ODISC,SIGBR1,SNRD, 00022860
     2              SIGDB                                             00022870
      COMMON /SIGDAT/SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE,DF1,DF5,          00022880
     2               DF2,DF4,SIGBAR                                   00022890
      COMMON /NOISE/NS1,NS2,NN(10),GAUSS(320)                         00022900
      COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD                         00022910
      DIMENSION NFREQ(2),PDIA(2),PDIR(2),PDIV(2),PS(10,2),BN(2),PT(3) 00022920
     2 ,TDC(3)
      DATA NFREQ/1,5/,BN/9772.4,616.6/,PS/9*1.,2.,5*1.,2.,4.,8.,8.,16./00022930
     2     ,PDIA,PDIR,PDIV/1.4142,3.1623,2.0,4.4721,2.8284,6.3246/,   00022940
     3      PT/42658.,3125.,195.3/,QNV/.04166666/                     00022950
      DATA TDC/0.05122118,0.1195161,0.2561557/
C                                                                     00022970
C  NOTE: DEBUGGING PRINT STATEMENTS.                                  00022980
C     WRITE(6,900) SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE                     00022990
C     WRITE(6,901) DF1,DF5,DF2,DF4,SIGBAR                             00023000
 900  FORMAT(' SPZ,SMZ,SPL,SML,E,L =',6F10.2)                         00023010
 901  FORMAT(' DF1,DF5,DF2,DF4,SIG =',5F10.2)                         00023020
C                                                                     00023030
C  *****************************************************************  00023040
C  * STEP 1: COMPUTE CONSTANT USED IN SIGNAL SCALING AND COMPUTATION *  00023050
C  *         OF NOISE STATISTICS.                                   *  00023060
C  *****************************************************************  00023070
C
C
      TCON=(TSAM/TDC(MPRF))**.5
C                                                                     00023080
C  STEP 1-1: COMPUTE CONSTANT (NOTE: IT IS DIFFERENT FOR ACTIVE AND   00023090
C            PASSIVE MODES).                                          00023100
      IF(IMODE.EQ.2) GO TO 5                                          00023120
C  NOTE: THIS IS THE CONSTANT USED IN ACTIVE MODE.                    00023130
      YY=GA*PS(MRNG,IMODE)/(CGRNGE**2*BN(MSAM))                       00023140
      S1=YY/FLOAT(NFREQ(IMODE))                                       00023150
```

FIGURE 2.3-10 BASELINE VERSION OF SUBROUTINE DISCRM

PAGE 1

```
      GO TO 10                                                          00023160
C  NOTE: THIS IS THE CONSTANT USED IN PASSIVE MODE.                     00023170
CCCCCCCCCCCCCCCCCCCCCCCCCC MODS 2-15-83 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
5        CONTINUE
          PTFIX=PT(ITXP)
          IF(SRNG.LT.640.)PTFIX=4.2
          ISTS7=0
          IF(ISTS7.EQ.1)PTFIX=4.2
C
      YY=GP*PS(MRNG,IMODE)*PTFIX   /(CGRNGE**4*BN(MSAM))                00023180
      S1=YY/FLOAT(NFREQ(IMODE))                                         00023190
C                                                                       00023200
C  STEP 1-2: COMPUTE PEAK SIGNAL POWER TO AVERAGE THERMAL NOISE POWER   00023210
C            AT DOPPLER FILTER OUTPUT.                                  00023220
   10 SNRDT=YY*SIGBAR                                                   00023230
C       WRITE(6,221)YY,SIGBAR
C  221     FORMAT('YY,SIGBAR =',F14.5)
      SNRDTD=10.*ALOG10(SNRDT)                                          00023240
      SIGDB=10.*ALOG10(SIGBAR)                                          00023250
      SIGBR1=SIGBAR                                                     00023260
C222  WRITE(6,990) SNRDTD,SIGDB                                         00023262
  990 FORMAT(' SNRDTD,SIGDB =',2F14.2)                                  00023264
C                                                                       00023270
C  STEP 1-3: COMPUTE PEAK SIGNAL POWER TO TOTAL (THERMAL PLUS           00023280
C            QUANTIZATION) NOISE POWER AT THE DOPPLER FILTER OUTPUT.    00023290
      CALL SATNSE(SNF)                                                  00023292
      XX=SNF*AGCO                                                       00023294
      XX=XX/(XX+QNV)                                                    00023296
      S1=S1*XX                                                          00023300
      YY=YY*XX                                                          00023310
      SNRD=YY*SIGBAR                                                    00023320
      SNRD=10.*ALOG10(SNRD)                                             00023330
C                                                                       00023340
C  STEP 1-4: UPDATE NOISE SEQUENCE.                                     00023350
      NN(1)=MOD(NN(1)+1,320)+1                                          00023360
      DO 15 I=2,10                                                      00023370
   15 NN(I)=MOD(NN(I-1)+29,320)+1                                       00023380
      ID1=NN(1)                                                         00023390
      GAUSS(ID1)=ANORM(NS1,NS2)                                         00023400
C                                                                       00023410
C  ******************************************************              00023420
C  * STEP 2: COMPUTE ANGLE DISCRIMINANT (INCLUDES NOISE) *              00023430
C  ******************************************************              00023440
C                                                                       00023450
C  STEP 2-1: CHECK ANTENNA STEERING MODE ---- SKIP STEP 2 IF IN         00023460
C            GPC-DES OR MANUAL.                                         00023470
CCCCCCCCCCCCCCCCCCCCCC MOD FEB 16 1983 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 20                               00023480
C                                                                       00023490
C  STEP 2-2: COMPUTE ANGLE DISCRIMINANT COMPONENT SCALE FACTOR.         00023500
      ASCALE=S1*PDIA(IMODE)                                             00023510
C                                                                       00023520
C  STEP 2-3: COMPUTE STATISTICS OF ADDITIVE NOISE FOR ANGLE             00023530
C            DISCRIMINANT COMPONENTS.                                   00023540
      MEAN=PDIA(IMODE)                                                  00023550
      VARPAZ=SQRT(2.*S1*SPAZ+1.)                                        00023560
      VARMAZ=SQRT(2.*S1*SMAZ+1.)                                        00023570
      VARPEL=SQRT(2.*S1*SPEL+1.)                                        00023580
      VARMEL=SQRT(2.*S1*SMEL+1.)                                        00023590
C                                                                       00023600
C  STEP 2-4: ADD EQUIVALENT NOISE TO ANGLE DISCRIMINANT COMPONENT       00023610
C            SIGNALS.                                                   00023620
      ID6=NN(6)                                                         00023630
```

FIGURE 2.3-10 BASELINE VERSION OF SUBROUTINE DISCRM

PAGE 2

```
      SPAZ=ABS(ASCALE*SPAZ+MEAN+VARPAZ*GAUSS(ID1))
      SMAZ=ABS(ASCALE*SMAZ+MEAN+VARMAZ*GAUSS(ID6))
      ID2=NN(2)
      ID7=NN(7)
      SPEL=ABS(ASCALE*SPEL+MEAN+VARPEL*GAUSS(ID2))
      SMEL=ABS(ASCALE*SMEL+MEAN+VARMEL*GAUSS(ID7))
C
C   STEP 2-5: COMPUTE AZ AND EL DISCRIMINANT COMPONENTS.
      AZDISC=10.*ALOG10(SPAZ/SMAZ)
      ELDISC=10.*ALOG10(SPEL/SMEL)
C      AZDISC=0.
C      ELDISC=0.
C
C   *******************************************************
C   * STEP 3: COMPUTE RANGE DISCRIMINANT (INCLUDES NOISE) *
C   *******************************************************
C
C   STEP 3-1: COMPUTE RANGE DISCRIMINANT COMPONENT SCALE FACTOR.
   20 RSCALE=S1*PDIR(IMODE)
C
C   STEP 3-2: COMPUTE STATISTICS OF ADDITIVE NOISE FOR RANGE
C             DISCRIMINANT.
      MEAN=PDIR(IMODE)
      VARELY=SQRT(2.*S1*EARLY+1.)*TCON
      VARLTE=SQRT(2.*S1*LATE+1.)*TCON
C
C   STEP 3-3: ADD EQUIVALENT NOISE TO RANGE DISCRIMINANT COMPONENT
C             SIGNALS.
      ID3=NN(3)
      ID8=NN(8)
      EARLY=ABS(RSCALE*EARLY+MEAN+VARELY*GAUSS(ID3))
      LATE=ABS(RSCALE*LATE+MEAN+VARLTE*GAUSS(ID8))
C
C   STEP 3-4: COMPUTE RANGE DISCRIMINANT.
      RDISC=10.*ALOG10(LATE/EARLY)
C
C   **********************************************************
C   * STEP 4: COMPUTE VELOCITY DISCRIMINANT (INCLUDES NOISE) *
C   **********************************************************
C
C   STEP 4-1: COMPUTE VELOCITY DISCRIMINANT COMPONENT SCALE FACTOR.
      VSCALE=S1*PDIV(IMODE)
C
C   STEP 4-2: COMPUTE STATISTICS OF ADDITIVE NOISE FOR VELOCITY
C               DISCRIMINANT COMPONENTS.
      MEAN=PDIV(IMODE)
      VARDF2=SQRT(2.*S1*DF2+1.)
      VARDF4=SQRT(2.*S1*DF4+1.)
C
C   STEP 4-3: ADD EQUIVALENT NOISE TO VELOCITY DISCRIMINANT
C               COMPONENT SIGNALS.
      ID4=NN(4)
      ID9=NN(9)
      DF2=ABS(VSCALE*DF2+MEAN+VARDF2*GAUSS(ID4))
      DF4=ABS(VSCALE*DF4+MEAN+VARDF4*GAUSS(ID9))
C
C   STEP 4-4: COMPUTE VELOCITY DISCRIMINANT.
      VDISC=10.*ALOG10(DF2/DF4)
C
C   **************************************************************
C   * STEP 5: COMPUTE ON-TARGET DISCRIMINANT —— USED FOR BREAK- *
C   *         TRACK AND VELOCITY DATA INVALID DETERMINATION      *
C   **************************************************************
C
```

FIGURE 2.3-10 BASELINE VERSION OF SUBROUTINE DISCRM

PAGE 3

```
C  STEP 5-1: COMPUTE STATISTICS OF ADDITIVE NOISE FOR OUTER DOPPLER
C           FILTER SIGNALS.
      VARDF1=SQRT(2.*S1*DF1+1.)
      VARDF5=SQRT(2.*S1*DF5+1.)
C
C  STEP 5-2: ADD EQUIVALENT NOISE TO OUTER DOPPLER FILTER SIGNALS.
      ID5=NN(5)
      ID10=NN(10)
      DF1=ABS(VSCALE*DF1+MEAN+VARDF1*GAUSS(ID5))
      DF5=ABS(VSCALE*DF5+MEAN+VARDF5*GAUSS(ID10))
C
C  STEP 5-3: COMPUTE ON-TARGET DISCRIMINANT.
C           NOTE: THE FACTOR OF SQRT(2.) IS DUE TO THE METHOD OF
C                 NORMALIZATION OF DISCRIMINANT COMPONENTS.
      ODISC=10.*ALOG10((EARLY+LATE)*SQRT(2.)/(DF1+DF5))
C
C  NOTE: DEBUGGING PRINT STATEMENTS.
C     WRITE(6,902) AZDISC,ELDISC,RDISC,VDISC,ODISC
C     WRITE(6,903) SNRD,SIGDB,SIGBAR
C     WRITE(6,904) SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE
C     WRITE(6,905) DF1,DF5,DF2,DF4,SIGBAR
  902 FORMAT(/' AZD,ELD,RD,VD,OD =',5F14.6)
  903 FORMAT(' SNRD,SIGDB,SIGBAR =',3F14.6)
  904 FORMAT(' SPZ,SMZ,SPL,SML,E,L+NOISE =',6F10.2)
  905 FORMAT(' DF1,DF5,DF2,DF4,SIG+NOISE =',5F10.2)
      RETURN
      END
C
```

FIGURE 2.3-10 BASELINE VERSION OF SUBROUTINE DISCRM

PAGE 4

```
C                                                                       00022720
C  ***************************************************************      00022730
C  * THIS SUBROUTINE ADDS THE EQUIVALENT NOISE TO THE ANGLE, RANGE, *   00022740
C  * VELOCITY AND ON-TARGET DISCRIMINANT COMPONENTS AND THEN COM-   *   00022750
C  * PUTES THE ANGLE, RANGE, VELOCITY, AND ON-TARGET DISCRIMINANTS. *   00022760
C  ***************************************************************      00022770
C                                                                       00022780
C                                                                       00022790
      SUBROUTINE DISCRM                                                 00022800
      REAL LATE,MEAN                                                    00022805
      COMMON /OUTPUT/MSWF,MTF,MSF,SRNG,SRDOT,SPANG,SRANG,SPRTE,
     2               SRRTE,SRSS,MADVF,MRDVF,MARDVF,MRRDVF
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)                00022810
      COMMON /ICNTL/I3DUM(14),MRNG,MSAM,MPRF,IDUM4(10)                  00022820
      COMMON /SYSDAT/TSAM,DR(3),CP,SP,PSI,PSBIAS,ALBIAS,BTBIAS,GP,GA,   00022830
     2               DUMS(3)                                            00022840
      COMMON /TGTDAT/NT,DUM5(506),CGRNGE,CGVEL                          00022850
      COMMON /DSCRM/AZDISC,ELDISC,RDISC,VDISC,RRTE,ODISC,SIGBR1,SNRD,   00022860
     2              SIGDB                                               00022870
      COMMON /SIGDAT/SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE,DF1,DF5,            00022880
     2               DF2,DF4,SIGBAR                                     00022890
      COMMON /NOISE/NS1,NS2,NN(10),GAUSS(320)                           00022900
      COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD                           00022910
      DIMENSION NFREQ(2),PDIA(2),PDIR(2),PDIV(2),PS(10,2),BN(2),PT(3)   00022920
     2 ,TDC(3)
      DIMENSION QNV(2)
C
C ---------PS AND QNV CONSTANT CHANGES FEB 17,1986 BY M. MEYER---------
C
      DATA NFREQ/1,5/,BN/9772.4,616.6/
      DATA PS/9*4.,2.,5*4.,2.,4.,8.,8.,16./
     2     ,PDIA,PDIR,PDIV/1.4142,3.1623,2.0,4.4721,2.8284,6.3246/,     00022940
     3      PT/42658.,3125.,195.3/
      DATA QNV/.00067,.011/
      DATA TDC/0.05122118,0.1195161,0.2561557/
C                                                                       00022970
C  NOTE: DEBUGGING PRINT STATEMENTS.                                    00022980
C     WRITE(6,900) SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE                       00022990
C     WRITE(6,901) DF1,DF5,DF2,DF4,SIGBAR                               00023000
 900  FORMAT(' SPZ,SMZ,SPL,SML,E,L =',6F10.2)                           00023010
 901 `FORMAT(' DF1,DF5,DF2,DF4,SIG =',5F10.2)                           00023020
C                                                                       00023030
C  ****************************************************************     00023040
C  * STEP 1: COMPUTE CONSTANT USED IN SIGNAL SCALING AND COMPUTATION *  00023050
C  *         OF NOISE STATISTICS.                                    *  00023060
C  ****************************************************************     00023070
C
C
      TCON=(TSAM/TDC(MPRF))**.5
C                                                                       00023080
```

FIGURE 2.3-11 DELIVERABLE VERSION OF SUBROUTINE DISCRM

PAGE 1

```
C  STEP 1-1: COMPUTE CONSTANT (NOTE: IT IS DIFFERENT FOR ACTIVE AND
C            PASSIVE MODES).
      IF(IMODE.EQ.2) GO TO 5
C  NOTE: THIS IS THE CONSTANT USED IN ACTIVE MODE.
      YY=GA*PS(MRNG,IMODE)/(CGRNGE**2*BN(MSAM))
      S1=YY/FLOAT(NFREQ(IMODE))
      GO TO 10
C  NOTE: THIS IS THE CONSTANT USED IN PASSIVE MODE.
CCCCCCCCCCCCCCCCCCCCCCCCCC MODS 2-15-83 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
5       CONTINUE
         PTFIX=PT(ITXP)
         IF(SRNG.LT.640.)PTFIX=4.2
         ISTS7=0
         IF(ISTS7.EQ.1)PTFIX=4.2
C
      YY=GP*PS(MRNG,IMODE)*PTFIX    /(CGRNGE**4*BN(MSAM))
      S1=YY/FLOAT(NFREQ(IMODE))
C
C  STEP 1-2: COMPUTE PEAK SIGNAL POWER TO AVERAGE THERMAL NOISE POWER
C            AT DOPPLER FILTER OUTPUT.
   10 SNRDT=YY*SIGBAR
C         WRITE(6,221)YY,SIGBAR
  221    FORMAT('YY,SIGBAR =',2F14.5)
      SNRDTD=10.*ALOG10(SNRDT)
      SIGDB=10.*ALOG10(SIGBAR)
      SIGBR1=SIGBAR
C222  WRITE(6,990) SNRDTD,SIGDB
  990 FORMAT(' SNRDTD,SIGDB =',2F14.2)
C
C  STEP 1-3: COMPUTE PEAK SIGNAL POWER TO TOTAL (THERMAL PLUS
C            QUANTIZATION) NOISE POWER AT THE DOPPLER FILTER OUTPUT.
      CALL SATNSE(SNF)
      XX=SNF*AGCO
      XX=XX/(XX+QNV(MSAM))
      S1=S1*XX
      YY=YY*XX
      SNRD=YY*SIGBAR
      SNRD=10.*ALOG10(SNRD)
C
C  STEP 1-4: UPDATE NOISE SEQUENCE.
      NN(1)=MOD(NN(1)+1,320)+1
      DO 15 I=2,10
   15 NN(I)=MOD(NN(I-1)+29,320)+1
      ID1=NN(1)
      GAUSS(ID1)=ANORM(NS1,NS2)
C
C  *****************************************************
C  * STEP 2: COMPUTE ANGLE DISCRIMINANT (INCLUDES NOISE) *
C  *****************************************************
C
C  STEP 2-1: CHECK ANTENNA STEERING MODE -- SKIP STEP 2 IF IN
C            GPC-DES OR MANUAL.
CCCCCCCCCCCCCCCCCCCCC MOD FEB 16 1983 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 20
C
C  STEP 2-2: COMPUTE ANGLE DISCRIMINANT COMPONENT SCALE FACTOR.
      ASCALE=S1*PDIA(IMODE)
C
C  STEP 2-3: COMPUTE STATISTICS OF ADDITIVE NOISE FOR ANGLE
C            DISCRIMINANT COMPONENTS.
      MEAN=PDIA(IMODE)
      VARPAZ=SQRT(2.*S1*SPAZ+1.)
      VARMAZ=SQRT(2.*S1*SMAZ+1.)
```

FIGURE 2.3-11 DELIVERABLE VERSION OF SUBROUTINE DISCRM

PAGE 2

```
      VARPEL=SQRT(2.*S1*SPEL+1.)
      VARMEL=SQRT(2.*S1*SMEL+1.)
C
C  STEP 2-4: ADD EQUIVALENT NOISE TO ANGLE DISCRIMINANT COMPONENT
C            SIGNALS.
      ID6=NN(6)
      SPAZ=ABS(ASCALE*SPAZ+MEAN+VARPAZ*GAUSS(ID1))
      SMAZ=ABS(ASCALE*SMAZ+MEAN+VARMAZ*GAUSS(ID6))
      ID2=NN(2)
      ID7=NN(7)
      SPEL=ABS(ASCALE*SPEL+MEAN+VARPEL*GAUSS(ID2))
      SMEL=ABS(ASCALE*SMEL+MEAN+VARMEL*GAUSS(ID7))
C
C  STEP 2-5: COMPUTE AZ AND EL DISCRIMINANT COMPONENTS.
      AZDISC=10.*ALOG10(SPAZ/SMAZ)
      ELDISC=10.*ALOG10(SPEL/SMEL)
C       AZDISC=0.
C       ELDISC=0.
C
C  ****************************************************
C  * STEP 3: COMPUTE RANGE DISCRIMINANT (INCLUDES NOISE) *
C  ****************************************************
C
C  STEP 3-1: COMPUTE RANGE DISCRIMINANT COMPONENT SCALE FACTOR.
   20 RSCALE=S1*PDIR(IMODE)
C
C  STEP 3-2: COMPUTE STATISTICS OF ADDITIVE NOISE FOR RANGE
C            DISCRIMINANT.
      MEAN=PDIR(IMODE)
      VARELY=SQRT(2.*S1*EARLY+1.)*TCON
      VARLTE=SQRT(2.*S1*LATE+1.)*TCON
C
C  STEP 3-3: ADD EQUIVALENT NOISE TO RANGE DISCRIMINANT COMPONENT
C            SIGNALS.
      ID3=NN(3)
      ID8=NN(8)
      EARLY=ABS(RSCALE*EARLY+MEAN+VARELY*GAUSS(ID3))
      LATE=ABS(RSCALE*LATE+MEAN+VARLTE*GAUSS(ID8))
C
C  STEP 3-4: COMPUTE RANGE DISCRIMINANT.
      RDISC=10.*ALOG10(LATE/EARLY)
C
C  *******************************************************
C  * STEP 4: COMPUTE VELOCITY DISCRIMINANT (INCLUDES NOISE) *
C  *******************************************************
C
C  STEP 4-1: COMPUTE VELOCITY DISCRIMINANT COMPONENT SCALE FACTOR.
      VSCALE=S1*PDIV(IMODE)
C
C  STEP 4-2: COMPUTE STATISTICS OF ADDITIVE NOISE FOR VELOCITY
C             DISCRIMINANT COMPONENTS.
      MEAN=PDIV(IMODE)
      VARDF2=SQRT(2.*S1*DF2+1.)
      VARDF4=SQRT(2.*S1*DF4+1.)
C
C  STEP 4-3: ADD EQUIVALENT NOISE TO VELOCITY DISCRIMINANT
C             COMPONENT SIGNALS.
      ID4=NN(4)
      ID9=NN(9)
      DF2=ABS(VSCALE*DF2+MEAN+VARDF2*GAUSS(ID4))
      DF4=ABS(VSCALE*DF4+MEAN+VARDF4*GAUSS(ID9))
C
C  STEP 4-4: COMPUTE VELOCITY DISCRIMINANT.
      VDISC=10.*ALOG10(DF2/DF4)
```

FIGURE 2.3-11 DELIVERABLE VERSION OF SUBROUTINE DISCRM

PAGE 3

```
C
C  **************************************************************
C  * STEP 5: COMPUTE ON-TARGET DISCRIMINANT -- USED FOR BREAK- *
C  *          TRACK AND VELOCITY DATA INVALID DETERMINATION       *
C  **************************************************************
C
C  STEP 5-1: COMPUTE STATISTICS OF ADDITIVE NOISE FOR OUTER DOPPLER
C            FILTER SIGNALS.
      VARDF1=SQRT(2.*S1*DF1+1.)
      VARDF5=SQRT(2.*S1*DF5+1.)
C
C  STEP 5-2: ADD EQUIVALENT NOISE TO OUTER DOPPLER FILTER SIGNALS.
      ID5=NN(5)
      ID10=NN(10)
      DF1=ABS(VSCALE*DF1+MEAN+VARDF1*GAUSS(ID5))
      DF5=ABS(VSCALE*DF5+MEAN+VARDF5*GAUSS(ID10))
C
C  STEP 5-3: COMPUTE ON-TARGET DISCRIMINANT.
C            NOTE: THE FACTOR OF SQRT(2.) IS DUE TO THE METHOD OF
C                  NORMALIZATION OF DISCRIMINANT COMPONENTS.
      ODISC=10.*ALOG10((EARLY+LATE)*SQRT(2.)/(DF1+DF5))
C
C  NOTE: DEBUGGING PRINT STATEMENTS.
C     WRITE(6,902) AZDISC,ELDISC,RDISC,VDISC,ODISC
C     WRITE(6,903) SNRD,SIGDB,SIGBAR
C     WRITE(6,904) SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE
C     WRITE(6,905) DF1,DF5,DF2,DF4,SIGBAR
 902  FORMAT(/' AZD,ELD,RD,VD,OD =',5F14.6)
 903  FORMAT(' SNRD,SIGDB,SIGBAR =',3F14.6)
 904  FORMAT(' SPZ,SMZ,SPL,SML,E,L+NOISE =',6F10.2)
 905  FORMAT(' DF1,DF5,DF2,DF4,SIG+NOISE =',5F10.2)
      RETURN
      END
C
```

FIGURE 2.3-11 DELIVERABLE VERSION OF SUBROUTINE DISCRM

PAGE 4

```
*************
LINES DELETED FROM BASELINE PROGRAM
   40            DATA NFREQ/1,5/,BN/9772.4,616.6/,PS/9*1.,2.,5*1.,2.,4.,8.,8.,16./00022930
   41         2       ,PDIA,PDIR,PDIV/1.4142,3.1623,2.0,4.4721,2.8284,6.3246/,     00022940
   42         3        PT/42658.,3125.,195.3/,QNV/.04166666/                         00022950
   43            DATA TDC/0.05122118,0.1195161,0.2561557/
******
LINES ADDED TO DELIVERABLE PROGRAM
   40            DIMENSION QNV(2)
   41    C
   42    C ————————PS AND QNV CONSTANT CHANGES FEB 17,1986 BY M. MEYER————————
   43    C
   44            DATA NFREQ/1,5/,BN/9772.4,616.6/
   45            DATA PS/9*4.,2.,5*4.,2.,4.,8.,8.,16./
   46         2       ,PDIA,PDIR,PDIV/1.4142,3.1623,2.0,4.4721,2.8284,6.3246/,     00022940
   47         3        PT/42658.,3125.,195.3/
   48            DATA QNV/.00067,.011/
   49            DATA TDC/0.05122118,0.1195161,0.2561557/
*************
*************
LINES DELETED FROM BASELINE PROGRAM
   81    C         WRITE(6,221)YY,SIGBAR
   82    C  221        FORMAT('YY,SIGBAR =',F14.5)
   83            SNRDTD=10.*ALOG10(SNRDT)                                            00023240
******
LINES ADDED TO DELIVERABLE PROGRAM
   87    C             WRITE(6,221)YY,SIGBAR
   88       221      FORMAT('YY,SIGBAR =',2F14.5)
   89            SNRDTD=10.*ALOG10(SNRDT)                                            00023240
*************
*************
LINES DELETED FROM BASELINE PROGRAM
   93            XX=XX/(XX+QNV)                                                      00023296
   94            S1=S1*XX                                                            00023300
******
LINES ADDED TO DELIVERABLE PROGRAM
   99            XX=XX/(XX+QNV(MSAM))
  100            S1=S1*XX                                                            00023300
*************

Number of difference sections found: 3
Number of difference records found: 12

DIFFERENCES /IGNORE=()/MERGED=1/OUTPUT=SYS$DISK3:[MCCOLLOUGH]DIFF4.FOR;1-
    SYS$DISK3:[MCCOLLOUGH]DISCRMH.FOR;2-
    SYS$DISK3:[MCCOLLOUGH]DISCRMF.FOR;2
```

FIGURE 2.3-12 SUMMARY OF MODIFICATIONS TO SUBROUTINE DISCRM

C-2

2.3.4.2 Test Results

The first test run with a RCS of 10 dBsm showed large (1 dB) discontinuities in the RSS at the sample rate changes. An examination of the RSS equation expressed in terms of SNR_{vt} and thermal noise power (N_t) showed that this result was not predicted. Consider the expression for RSS,

(2-14) $$RSS=10\ \log(SNR_{vt}+1/G) + 10\ \log(N_t/4q^2)$$

Now, the sample rate change causes a 12 db change in SNR_{vt} and a 12 db change in thermal noise power N_t which offsets the SNR_{vt} change. Furthermore, at this range the SNR_{vt} is on the order of 10^4 and 1/G is 1/16 therefore the sample rate change shouldn't have introduced a discontinuity, but a discontinuity appeared in the data. The following AGC equations were then examined to determine an answer to this unexpected result:

(2-15) $$RSS=10\ \log(1/AGC)$$

(2-16) $$AGC(N+1)=AGC(N)AGCERR(N)$$

(2-17) $$AGCERR(N)=k_1G/(AGC(N)(SNRDT(N)+1)+k_2)$$

where G=signal to noise ratio gain from doppler filter.
$k_1=(2q)^2/N_t$
$k_2=(q)^2/(12\ N_t)$
N_t=un AGC'd thermal noise at the A/D input

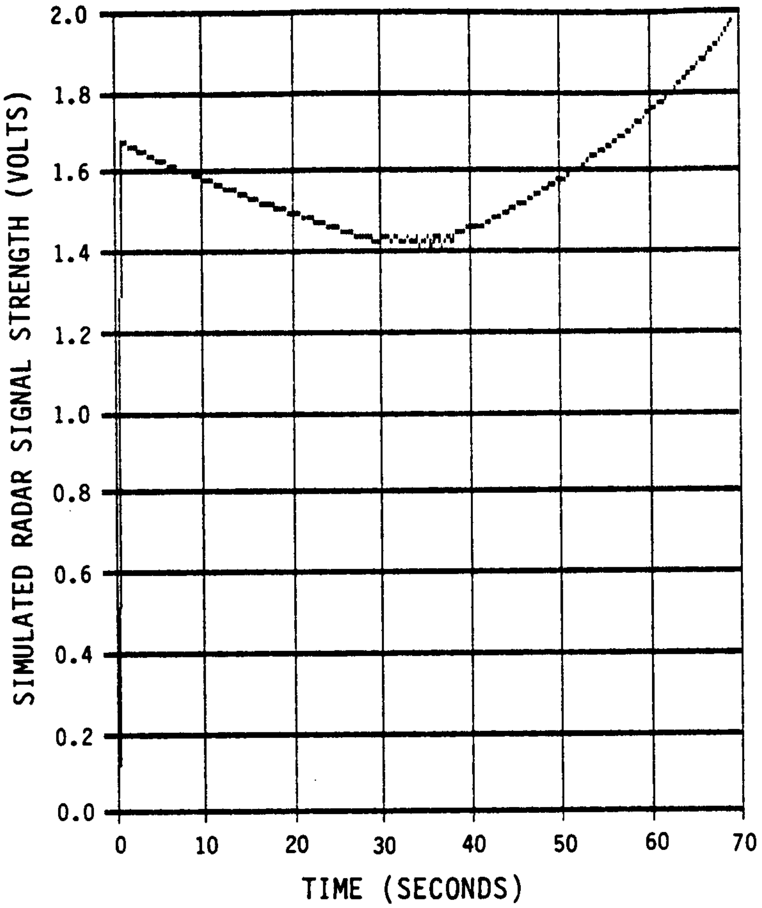
It is seen that the variables k_1 and k_2 are functions of the thermal noise power Nt. Therefore, since the thermal noise power changes by a factor of the ratios of the noise bandwidth of the high sample rate video filter to the noise bandwidth of the low sample rate video filter, the ratios between k_1(high sample rate) and k_1(low sample rate) and the ratio between k_2(high sample rate) and k_2(low sample rate) should be precisely the ratios of the noise bandwidth. In Table 2-1 (from page 2-3 of Reference 1), from which the values of k_1 and k_2 were taken, this was not true. The appropriate modification of the variables k_1 and k_2 and subsequent simulation run showed that this solved the problem (see Figure 2.3-13) and that the RSS behaved as expected over the entire trajectory.

A third run of the simulation was then made with the RCS set to -40dBsm. The output RSS plot (Figure 2.3-14) has discontinuities in the RSS at both high-to-low and low-to-high sample rate changes. Examination of equation 2-14 shows that this should be expected in both cases. Consider the high sample rate-to-low sample rate transition. The SNR_{vt} in the high sample rate mode is less than 1. Therefore the SNR_{vt} is on the same order of magnitude as 1/G. Now, switching to the low sample rate mode increases SNR_{vt} by 12 db and decrease the thermal noise power by 12 db. Although SNR_{vt} changes by 12 db the change in the term $10 \log(SNR_{vt}+1/G)$ is less than 12 db because SNR_{vt} is on the same order as 1/G in the high sample rate mode. For the low sample rate-to-high sample rate case, the mechanism producing the discontinuity is the same except that the SNR_{vt} decreases by 12 db and the noise power increases 12 db.

2.4 RADAR PROCESSING PARAMETER CHANGES

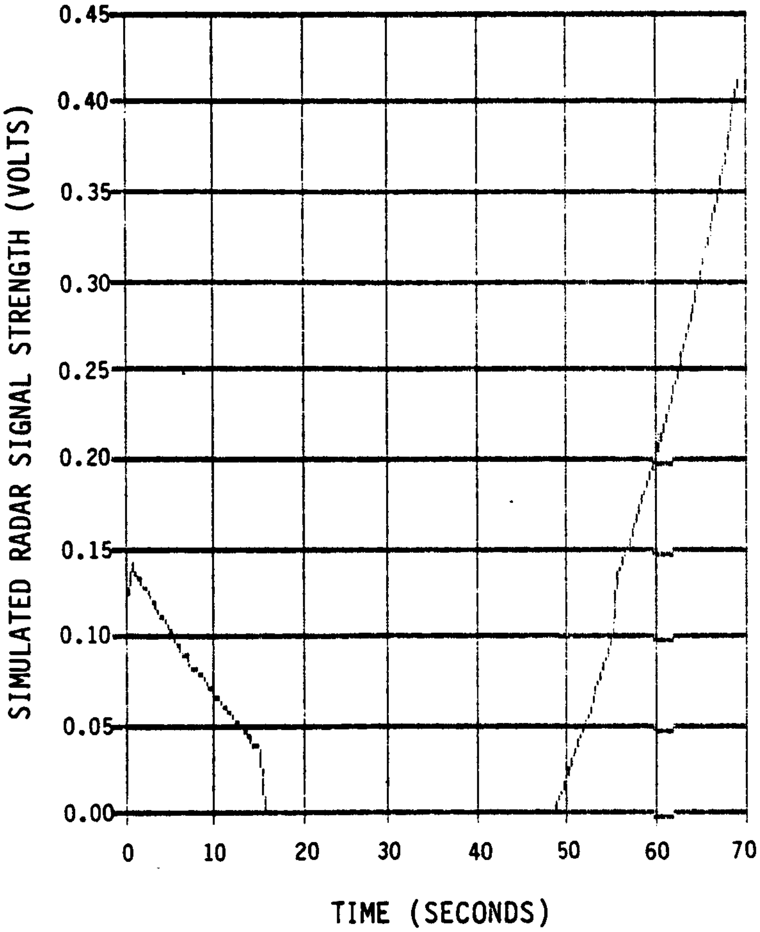
2.4.1 Problem Definition

Problems documented in this section were precipitated by several modifications in the radar design during the system test phase of the radar development. These modifications included changes in pulsewidth, PRF, and transmit power transition points. In addition, the original simulation model neglected to include the hysteresis loops governing the sample rate transition point and the PRF transition point. While ignoring the hysteresis loop produces only very minor performance error, the addition of this loop was a minor operation and was therefore included in the modifications package.



MEAN = 1.572   STANDARD DEVIATION = 0.177

FIGURE 2.3-13 SIMULATED RADAR SIGNAL STRENGTH
RADAR CROSS SECTION = + 10 dBsm



MEAN = 0.075 STANDARD DEVIATION = 0.105

FIGURE 2.3-14 SIMULATED RADAR SIGNAL STRENGTH
RADAR CROSS SECTION = -40 dBsm

2.4.2 Algorithm Modifications

Modifications to this algorithm include the following items:

- moving the 7-kHz to 3-kHz PRF transition point from 9.8 nautical miles into 8.2 nautical miles.
- adding hysteresis to the 7-kHz to 3-kHz PRF transition.
- adding hysteresis to the high sample rate-to-low sample rate transition point.
- updating the range interval boundary table.

Figure 2.4-1 provides an illustration of the hysteresis loop applied to the 7-kHz to 3-kHz PRF transition. Figures 2.4-2 defines the hysteresis loop applied to the sample rate transition. Also the range interval boundaries were updated to accurately reflect those used in the radar processor. Table 2.4-1 summarizes the new boundaries and the track mode pulsewidth associated with those boundaries.

2.4.3 Software Design Documentation

The changes described in Sections 2.4.1 and 2.4.2 were implemented through modifications to subroutine CNTRLS. The modifications included:

- Modifying four lines of existing code, and adding code to simulate the hysteresis loop for sampling rate transition.
- Modifying four lines of existing code, and adding code to simulate the hysteresis loop for Pulse Repetition Frequency (PRF) transition.

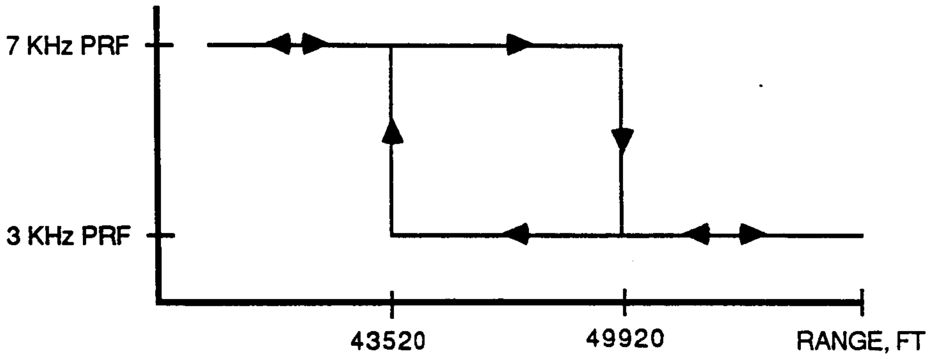


FIGURE 2.4-1 HYSTERESIS LOOP FOR PRF TRANSITION

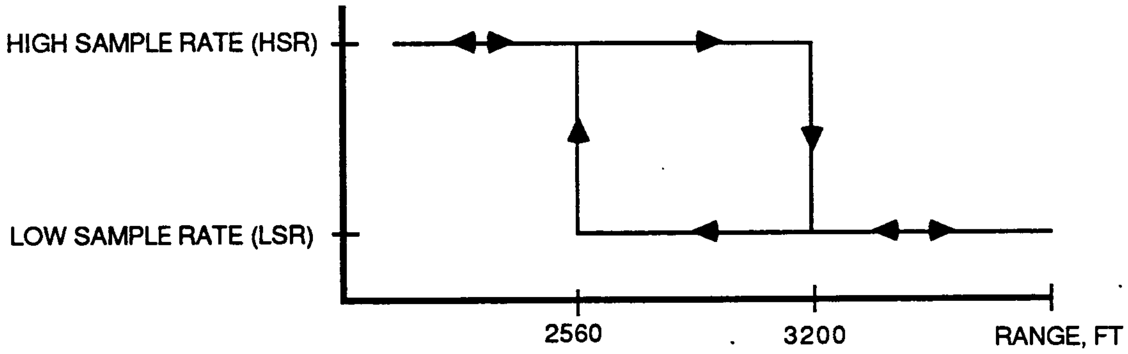


FIGURE 2.4-2 HYSTERESIS LOOP FOR SAMPLE RATE TRANSITION

TABLE 2.4-1 NEW RANGE INTERVAL BOUNDARIES

MRNG*	RANGE INTERVAL, FEET	PULSEWIDTH, SEC
1	120	0.122
2	120 - 630	0.122
3	640 - 1510	0.122
4	1520 - 2550	0.122
5	2560 - 5750	2.07
6	5760 - 11510	4.15
7	11510 - 23030	8.3
8	23040 - 43510	16.6
9	43520 - 49910	33.2
10	49920 - 1.82 E-6	33.2

It should be noted that minor changes to the values of constants were made in the main program and the subroutines DISCRM, RTRACK, SIGNAL, and RSS to accommodate the changes made to CNTRLS. These changes are minor, and are documented in Sections 2.2 and 2.3, so they will not be repeated here.

Figure 2.4-3 is a listing of the baseline version of CNTRLS. Figure 2.4-4 is a listing of the deliverable version of CNTRLS. The differences between the baseline and deliverable subroutines are listed in Figure 2.4-5.

2.4.4 Integration and Test Data

Testing of the high-sample to low-sample rate hysteresis loop defined in Figure 2.4-2 consisted of using the following scenario in the simulation. A 10 dBsm target was moved in range from 2400 feet to 4000 feet

(THIS PAGE INTENTIONALLY LEFT BLANK)

```
C
C  ********************************************************
C  * THIS SUBROUTINE UPDATES ALL RADAR INTERNAL CONTROLS. *
C  ********************************************************
C
C
      SUBROUTINE CNTRLS
      REAL INTT,NFIL,IRNG,IRDOT
      COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)
      COMMON /OUTPUT/IDUM0(3),SRNG,SRDOT,DUM2(5),IDUM(4)
      COMMON /ICNTL/I1DUM(14),MRNG,MSAM,MPRF,IDUM1(10),MPFOLD
      COMMON /RTDAT/IRDOT,IRNG,RBIAS,VEST(4),MDF(5)
      DIMENSION RI(10),FW(3)
C RI(4) CHANGED TO 2560 FROM 2552
      DATA RI/120.,240.,780.,2560.,5772.,11544.,23089.,43747.,
     2        57722.,1.8228E+6/
      DATA FW/7.7215,3.3090,0.2969/,NRI/10/
C
C  ****************************************
C  * STEP 1: SET RANGE INTERVAL PARAMETER *
C  ****************************************
      XRNG=IRNG*0.3125
      DO 60 I=1,NRI
      IF(XRNG.LE.RI(I)) GO TO 70
   60 CONTINUE
   70 MRNG=I
      IF(MRNG.GT.NRI) STOP
C
C  *************************************
C  * STEP 2: SET SAMPLE RATE PARAMETER *
C  *************************************
      IF(IMODE.GE.2)  GO TO 74
      IF(MRNG.GT.9) GO TO 72
      MSAM=1
      GO TO 80
72    MSAM=2
      GO TO 80
74    IF(MRNG.GT.4) GO TO 76
      MSAM=1
      GO TO 80
76    MSAM=2
C
C  *****************************
C  * STEP 3: SET PRF PARAMETER *
C  *****************************
C
C  STEP 3-1: DETERMINE IF IN ACTIVE OR PASSIVE MODE.
80    IF(IMODE.GE.2) GO TO 84
C
C  STEP 3-2: DETERMINE CORRECT PRF FOR GIVEN OPERATING MODE.
```

FIGURE 2.4-3 BASELINE VERSION OF SUBROUTINE CNTRLS

PAGE 1

```
      IF(MRNG.GT.9) GO TO 82                                            00028970
      MPRF=1                                                            00028980
      GO TO 90                                                          00028990
82    MPRF=3                                                            00029000
      GO TO 90                                                          00029010
84    IF(MRNG.GT.9) GO TO 86                                            00029020
      MPRF=1                                                            00029030
      GO TO 90                                                          00029040
86    MPRF=2                                                            00029050
90    CONTINUE                                                          00029060
C                                                                       00029070
C  STEP 3-3: IF PRF HAS CHANGED FROM PREVIOUS DATA CYCLE, THEN          00029080
C             RESET THE 5 DOPPLER TRACKING FILTERS ACCORDINGLY.         00029090
      IF(MPFOLD.EQ.MPRF) GO TO 96                                       00029100
      NFIL=INTT((-SRDOT/FW(MPRF))+0.5)+31998.                           00029110
      XX=AMOD(NFIL,32.)                                                 00029115
      MDF(1)=INT(XX)                                                    00029120
      DO 95 I=1,4                                                       00029130
   95 MDF(I+1)=MOD(MDF(1)+I,32)                                         00029140
   96 MPFOLD=MPRF                                                       00029150
C                                                                       00029160
C  NOTE: DEBUGGING PRINT STATEMENTS.                                    00029170
C     WRITE(6,999) MPRF,MPFOLD,MDF(1)                                   00029180
  999 FORMAT(' MPRF,MPFOLD,MDF1 =',3I8)                                 00029190
      RETURN                                                            00029200
      END                                                               00029210
C                                                                       00006680
```

FIGURE 2.4-3 BASELINE VERSION OF SUBROUTINE CNTRLS

PAGE 2

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C  ******************************************************
C  * THIS SUBROUTINE UPDATES ALL RADAR INTERNAL CONTROLS. *
C  ******************************************************
C
C
       SUBROUTINE CNTRLS
       REAL INTT,NFIL,IRNG,IRDOT
       COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)
       COMMON /OUTPUT/IDUM0(3),SRNG,SRDOT,DUM2(5),IDUM(4)
       COMMON /ICNTL/I1DUM(14),MRNG,MSAM,MPRF,IDUM1(10),MPFOLD
       COMMON /RTDAT/IRDOT,IRNG,RBIAS,VEST(4),MDF(5)
       DIMENSION RI(10),FW(3)
C RI(4) CHANGED TO 2560 FROM 2552
       DATA RI/120.,640.,1520.,2560.,5760.,11520.,23040.,43520.,
     2         49920.,1.8228E+6/
       DATA FW/7.7215,3.3090,0.2969/,NRI/10/
C
C
C  *******************************************************
C     IMPLEMENTATION OF HYSTERISIS FOR THE SAMPLING RATE
C     CHANGE AND FOR THE PRF CHANGE ALONG WITH CHANGES IN
C     RI(RANGE INTERVAL) WAS COMPLETED FEB 6,1986 BY M. MEYER
C  *******************************************************
C
C  ****************************************
C  * STEP 1: SET RANGE INTERVAL PARAMETER *
C  ****************************************
       XRNG=IRNG*0.3125
       DO 60 I=1,NRI
       IF(XRNG.LE.RI(I)) GO TO 70
   60  CONTINUE
   70  MRNG=I
       IF(MRNG.GT.NRI) STOP
C
C  ************************************
C  * STEP 2: SET SAMPLE RATE PARAMETER *
C  ************************************
       IF(IMODE.GE.2)  GO TO 74
       IF(MRNG.GT.9) GO TO 72
       MSAM=1
       GO TO 80
72     MSAM=2
       GO TO 80
C***** MODIFIED FEB 6 1986 BY M. MEYER*****************
74     IF(MSAM.EQ.1)THEN
          IF(XRNG.GT.3200.)THEN
             MSAM=2
          ELSE
             MSAM=1
```

FIGURE 2.4-4 DELIVERABLE VERSION OF SUBROUTINE CNTRLS

PAGE 1

```
C***** MODIFIED FEB 17,1986 BY M. MEYER ****************
C***** GUARANTEES THE CORRECT LOOP BANDWIDTHS**********
C***** FOR THE HYSTERISIS LOOP**************************
C
            IF(XRNG.GT.2560) MRNG=4
C
C*******************************************************
         END IF
      ELSE
         IF(XRNG.GT.2560.)THEN
            MSAM=2
         ELSE
            MSAM=1
         END IF
      END IF
C                                                                       00028880
C  ****************************                                         00028890
C  * STEP 3: SET PRF PARAMETER *                                        00028900
C  ****************************                                         00028910
C                                                                       00028920
C  STEP 3-1: DETERMINE IF IN ACTIVE OR PASSIVE MODE.                    00028930
80    IF(IMODE.GE.2) GO TO 84                                           00028940
C                                                                       00028950
C  STEP 3-2: DETERMINE CORRECT PRF FOR GIVEN OPERATING MODE.            00028960
      IF(MRNG.GT.9) GO TO 82                                            00028970
      MPRF=1                                                            00028980
      GO TO 90                                                          00028990
82    MPRF=3                                                            00029000
      GO TO 90                                                          00029010
C ****** MODIFIED FEB 6 1986 BY M. MEYER ***********
84    IF(MPRF.EQ.1)THEN
         IF(XRNG.GT.49920.)THEN
            MPRF=2
         ELSE
            MPRF=1
         END IF
      ELSE
         IF(XRNG.GT.43520.)THEN
           MPRF=2
C***** MODIFIED FEB 17, 1986 BY M. MEYER*************
C***** GUARANTEES THE CORRECT CONSTANTS ************
C***** FOR THE LOW PRF******************************
C
           MRNG=10
C
C***************************************************
         ELSE
            MPRF=1
         END IF
      END IF
90    CONTINUE                                                          00029060
C                                                                       00029070
C  STEP 3-3: IF PRF HAS CHANGED FROM PREVIOUS DATA CYCLE, THEN          00029080
C           RESET THE 5 DOPPLER TRACKING FILTERS ACCORDINGLY.           00029090
      IF(MPFOLD.EQ.MPRF) GO TO 96                                       00029100
      NFIL=INTT((-SRDOT/FW(MPRF))+0.5)+31998.                           00029110
      XX=AMOD(NFIL,32.)                                                 00029115
      MDF(1)=INT(XX)                                                    00029120
      DO 95 I=1,4                                                       00029130
   95 MDF(I+1)=MOD(MDF(1)+I,32)                                         00029140
   96 MPFOLD=MPRF                                                       00029150
C                                                                       00029160
C  NOTE: DEBUGGING PRINT STATEMENTS.                                    00029170
C     WRITE(6,999) MPRF,MPFOLD,MDF(1)                                   00029180
```

FIGURE 2.4-4 DELIVERABLE VERSION OF SUBROUTINE CNTRLS

PAGE 2

```
  999 FORMAT(' MPRF,MPFOLD,MDF1 =',3I8)                                   00029190
      RETURN                                                              00029200
      END                                                                 00029210
C                                                                         00006680
```

FIGURE 2.4-4 DELIVERABLE VERSION OF SUBROUTINE CNTRLS

PAGE 3

(THIS PAGE INTENTIONALLY LEFT BLANK)

ORIGINAL PAGE IS
OF POOR QUALITY

```
************
LINES DELETED FROM BASELINE PROGRAM
   29           DATA RI/120.,240.,780.,2560.,5772.,11544.,23089.,43747.,            00028620
   30        2          57722.,1.8228E+6/                                           00028630
   31           DATA FW/7.7215,3.3090,0.2969/,NRI/10/                               00028640
   32    C                                                                          00028650
******
LINES ADDED TO DELIVERABLE PROGRAM
   29           DATA RI/120.,640.,1520.,2560.,5760.,11520.,23040.,43520.,           00028620
   30        2          49920.,1.8228E+6/                                           00028630
   31           DATA FW/7.7215,3.3090,0.2969/,NRI/10/                               00028640
   32    C
   33    C
   34    C    ******************************************************
   35    C       IMPLEMENTATION OF HYSTERISIS FOR THE SAMPLING RATE
   36    C       CHANGE AND FOR THE PRF CHANGE ALONG WITH CHANGES IN
   37    C       RI(RANGE INTERVAL) WAS COMPLETED FEB 6,1986 BY M. MEYER
   38    C    ******************************************************
   39    C                                                                          00028650
************
************
LINES DELETED FROM BASELINE PROGRAM
   52    74     IF(MRNG.GT.4) GO TO 76                                              00028840
   53           MSAM=1                                                              00028850
   54           GO TO 80                                                            00028860
   55    76     MSAM=2                                                              00028870
   56    C                                                                          00028880
******
LINES ADDED TO DELIVERABLE PROGRAM
   59    C***** MODIFIED FEB 6 1986 BY M. MEYER*****************
   60    74     IF(MSAM.EQ.1)THEN
   61              IF(XRNG.GT.3200.)THEN
   62                 MSAM=2
   63              ELSE
   64                 MSAM=1
   65    C***** MODIFIED FEB 17,1986 BY M. MEYER ***************
   66    C***** GUARANTEES THE CORRECT LOOP BANDWIDTHS**********
   67    C***** FOR THE HYSTERISIS LOOP************************
   68    C
   69                   IF(XRNG.GT.2560) MRNG=4
   70    C
   71    C*****************************************************
   72              END IF
   73           ELSE
   74              IF(XRNG.GT.2560.)THEN
   75                 MSAM=2
   76              ELSE
   77                 MSAM=1
   78              END IF
   79           END IF
```

FIGURE 2.4-5 SUMMARY OF MODIFICATIONS TO SUBROUTINE CNTRLS

PAGE 1

```
  80   C                                                                         00028880
************
************
LINES DELETED FROM BASELINE PROGRAM
  70   84      IF(MRNG.GT.9) GO TO 86                                           00029020
  71           MPRF=1                                                           00029030
  72           GO TO 90                                                         00029040
  73   86      MPRF=2                                                           00029050
  74   90      CONTINUE                                                         00029060
******
LINES ADDED TO DELIVERABLE PROGRAM
  94   C ****** MODIFIED FEB 6 1986 BY M. MEYER ************
  95   84      IF(MPRF.EQ.1)THEN
  96              IF(XRNG.GT.49920.)THEN
  97                 MPRF=2
  98              ELSE
  99                 MPRF=1
 100              END IF
 101           ELSE
 102              IF(XRNG.GT.43520.)THEN
 103                MPRF=2
 104   C***** MODIFIED FEB 17, 1986 BY M. MEYER**************
 105   C***** GUARANTEES THE CORRECT CONSTANTS *************
 106   C***** FOR THE LOW PRF*******************************
 107   C
 108                MRNG=10
 109   C
 110   C**************************************************
 111              ELSE
 112                 MPRF=1
 113              END IF
 114           END IF
 115   90      CONTINUE                                                         00029060
************

Number of difference sections found: 3
Number of difference records found: 52

DIFFERENCES /IGNORE=()/MERGED=1/OUTPUT=SYS$DISK3:[MCCOLLOUGH]DIFF6.FOR;1-
    SYS$DISK3:[MCCOLLOUGH]CNTRLSH.FOR;2-
    SYS$DISK3:[MCCOLLOUGH]CNTRLSF.FOR;2
```

FIGURE 2.4-5 SUMMARY OF MODIFICATIONS TO SUBROUTINE CNTRLS

PAGE 2

with a speed of 50 feet per second and then the target range was decreased from 4000 feet to 1400 feet at a speed of 80 feet per second. As this scenario was executed, the following parameters were output: time, range, and the sample rate control parameter, MSAM. MSAM=1 corresponds to the high sample rate, while MSAM=2 corresponds to the low sample rate. Table 2.4-2 provides a summary of the test results. A comparison with Figure 2.4-2 shows that the simulation code is performing to the design.

The test to validate the operation of hysteresis in the 7-kHz to 3-kHz PRF transition was similar to the sample rate hysteresis test. In this case, a 10 dBsm target was moved in range from 42,000 feet to 53,000 feet at a speed of 50 feet per second and then the range was decreased from 53,000 feet to 38,000 feet at a speed of 76 feet per second. In this case, the following data was output as the simulation progressed: time, range, and the PRF control parameter, MPRF. Table 2.4-3 defines MPRF. Results of the test are summarized in Table 2.4-4. A comparison of these results with Figure 2.4-1 shows that the new code is performing as required.

2.5 VELOCITY PROCESSOR CHANGES

2.5.1 Problem Definition

The changes in the velocity processor module consisted of removing the range rate ambiquity resolver in the 7 kHz PRF mode and correcting a bug that was traced to this module. Removal of the ambiquity resolver is a direct result of changes to the radar following system test. The bug in the velocity processor module software was uncovered when the trajectories from the SORTE experiments were used to drive the simulation. One of these trajectories produced an unexpected glitch in the range rate. A subsequent investigation pointed to a problem in addressing the model of the PROM used to convert the velocity discriminant value to a velocity estimate. The problem was fixed and is documented in the following subsections.

TABLE 2.4-2 SAMPLE RATE TRANSITION HYSTERESIS LOOP TEST RESULTS

TIME, SEC	RANGE, FT	MSAM
14.39999	3141.250	1
14.79999	3166.875	1
15.19999	3181.563	1
15.59999	3205.625	2
15.99999	3222.188	2
16.39999	3244.063	2
16.79999	3263.438	2
54.00005	2670.625	2
54.40005	2640.000	2
54.80006	2612.500	2
55.20006	2578.125	2
55.60006	2544.688	1
56.00006	2509.688	1
56.40006	2479.063	1

TABLE 2.4-3 DEFINITION OF MPRF

MPRF	PRF, HZ
1	6969
2	2980
3	268

TABLE 2.4-4 PRF TRANSITION HYSTERESIS LOOP TEST RESULTS

TIME, SEC	RANGE, FT	MSAM
154.3999	49735.63	1
154.7999	49770.00	1
155.1999	49727.19	1
155.5999	49841.56	1
155.9999	49839.69	1
156.3999	49860.63	1
156.7999	49919.38	1
157.1999	49906.25	2
157.5999	49914.38	2
157.9999	49934.38	2
158.3999	49952.81	2
158.7999	49991.88	2
159.1999	50003.75	2
159.5999	49998.75	2
159.9998	50005.94	2
160.3998	50051.88	2
160.7998	50060.31	2
310.3985	43883.75	2
310.7985	43832.81	2
311.1985	43790.63	2
311.5984	43757.50	2
311.9984	43710.94	2
312.3984	43680.31	2
312.7984	43658.13	2
313.1984	43639.69	2
313.5984	43596.88	2
313.9984	43551.88	2
314.3984	43515.94	1
314.7984	43471.25	1
315.1984	43451.88	1
315.5984	43393.13	1
315.9984	43412.50	1
316.3984	43334.69	1
316.7984	43294.69	1
317.1984	43259.38	1
317.5984	43256.25	1
317.9984	43279.69	1
318.3983	43201.25	1
318.7983	43188.13	1
319.1983	43198.44	1
319.5983	43095.31	1

2.5.2 Algorithm Modifications

Removing the ambiguity resolver was straightforward. In the original algorithm, the range rate was determined by using (1) the filter number within the bank of 32 filters, (2) an estimate of the position withn the given filter obtained from the velocity discriminant, and (3) the number of filter banks which is determined using an estimate of the range rate from the range tracking loop (see Figure 2.1-1). The ambiguity resolver is effectively disabled by holding the number of filter banks to zero, regardless of the range rate estimate from the tracker. In the actual implementation of the algorithm, holding the number of filter banks to zero translates to holding the variable IRVEL to a value of 4096 for opening velocities and to a value of 0 for closing velocities.

The problem with addressing the PROM which is used to convert velocity discriminant values to positions within a filter can be described as follows. There are only 128 addresses in the array representing the PROM. However, a mistake in the code that checks the discriminant (which effectively is the PROM address) allows a value of 129. If this condition is obtained, it can either cause the program to terminate or cause the velocity estimate to glitch. The latter condition was observed in one of the simulation runs. The problem was easily corrected by changing the bounds on the code that checks the velocity estimate for saturation.

2.5.3 Software Design Documentation

The changes described in Subsections 2.5.1 and 2.5.2 were implemented by making modifications to the subroutine VELPRO. In particular, code was added following STEP 1.4 in the subroutine to properly update the velocity estimate when the radar is in the 7 kHz PRF mode.

Figure 2.5-1 is a listing of the original version of VELPRO. Figure 2.5-2 is a listing of the updated, deliverable code. Finally, Figure 2.5-3 is a line-by-line summary of the differences between the two.

```
C
C    ******************************************************************
C    * THIS SUBROUTINE COMPUTES AN ACCURATE, SMOOTHED VELOCITY USING *
C    * THE KU-BAND RADAR VELOCITY PROCESSOR ALGORITHM.                 *
C    ******************************************************************
C
C
      SUBROUTINE VELPRO
      REAL IRDOT,IRNG,INTT,IVEL,IVDISC,IFVEL,IRVEL,IR1,IR2,IR3,
    2      IF3,IDELTA
      COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)
      COMMON /OUTPUT/IDUM0(3),SRNG,SRDOT,DUM2(5),IDUM(4)
      COMMON /ICNTL/I1DUM(14),MRNG,MSAM,MPRF,IDUM1(10),MPFOLD
      COMMON /SYSDAT/TSAM,DUMS(14)
      COMMON /RTDAT/IRDOT,IRNG,RBIAS,VEST(4),MDF(5)
      COMMON /DSCRM/DUM(2),RDISC,VDSC,RRTE,ODISC,DUM3(3)
      DIMENSION IPROM(128),VT1(3),VT2(3),MW(4,3)
      DATA IPROM/127,127,125,124,122,121,120,118,117,116,114,113,
    2  111,110,109,107,106,105,103,102,101,99,98,97,95,94,93,92,90,
    3  89,88,87,85,84,83,82,81,79,78,77,76,75,73,72,71,70,69,68,67,
    4  66,65,64,63,62,61,60,59,58,57,56,55,54,53,52,51,50,49,49,48,
    5  47,46,45,44,44,43,42,41,41,40,39,38,38,37,36,36,35,34,34,33,
    6  32,32,31,31,30,30,29,28,28,27,27,26,26,25,25,24,24,23,23,22,
    7  22,22,21,21,20,20,19,19,19,18,18,17,17,17,16,16,16,15,15/
      DATA VT1/1.012592E-2,2.362726E-2,2.633237E-1/,VT2/1.204935,
    2          0.5163982,0.04633489/
      DATA MW/1,2,3,4,1,1,2,2,1,1,1,1/
C
C    ************************************************
C    * STEP 1: GENERATE AMBIGUOUS VELOCITY ESTIMATE *
C    ************************************************
C
C    STEP 1-1: INTEGERIZE VELOCITY DISCRIMINANT AND CHECK FOR SATURATION.
      VDISC=5.333333*VDSC
      IVDISC=INTT(VDISC+0.5)
      IF(IVDISC.LT.-128.) IVDISC=-128.
      IF(IVDISC.GT.127.) IVDISC=127.
C
C    STEP 1-2: COMPUTE INTEGRAL FILTER NUMBER PORTION OF AMBIGUOUS
C              VELOCITY ESTIMATE.
      INTEG=MDF(2)
      IF(IVDISC.LT.0.) INTEG=MOD(INTEG+1,32)
C
C    STEP 1-3: COMPUTE FRACTIONAL FILTER PORTION OF AMBIGUOUS VELOCITY
C              ESTIMATE.
C    ESTIMATE.
      IV1=INT(ABS(IVDISC))+1
      IFRAC=IPROM(IV1)
      IF(IVDISC.LT.0.) IFRAC=127-IFRAC
C
```

FIGURE 2.5-1 BASELINE VERSION OF SUBROUTINE VELPRO

PAGE 1

```
C  STEP 1-4: COMPUTE AMBIGUOUS VELOCITY ESTIMATE --- COMBINE INTEGRAL
C            AND FRACTIONAL PARTS. NOTE: LSB IS 1/128 OF FILTER WIDTH.
C  FRACTIONAL PARTS. NOTE: LSB IS 1/128 OF A FILTER WIDTH.
      IFVEL=FLOAT(IFRAC+128*INTEG)
C
C  ****************************************
C  * STEP 2: SCALE ROUGH VELOCITY ESTIMATE *
C  ****************************************
C
C  STEP 2-1: SCALE LSB OF ROUGH RANGE RATE ESTIMATE TO 4 TIMES A DOPPLER
C            WIDTH.
C  DEFINITION:  VT1(MPRF)=(RANGE LSB)/((MAX. UNAMBIGUOUS VELOCITY)/8)
C                OR VT1(MPRF)=5./(PRF*LAMBDA)
      R1=IRDOT*VT1(MPRF)/TSAM
      IR1=AINT(R1)
C
C  STEP 2-2: PERFORM SOME REQUIRED AUXILIARY CALCULATIONS.
      R2=IR1/8.
      IR2=AINT(R2)
      IRVEL=IR2*4096.
C
C  ****************************
C  * STEP 3: RESOLVE AMBIGUITY *
C  ****************************
C
C  STEP 3-1: COMPUTE 3 MSB'S OF AMBIGUOUS VELOCITY ESTIMATE.
      IF3=AINT(IFVEL/512.)
C
C  STEP 3-2: COMPUTE 3 LSB'S OF SCALED ROUGH RANGE RATE ESTIMATE.
      IR3=ABS(IR1-8.*IR2)
      IF(R1.LE.0.)GO TO 10
      IRVEL=IRVEL+4096.
      IR3=7.-IR3
   10 CONTINUE
C
C  STEP 3-3: COMPARE 3 MSB'S AND 3 LSB'S AND INCREMENT NUMBER OF
C           AMBIGUOUS FILTER BANK WIDTHS APPROPRIATELY.
      IDELTA=IR3-IF3
      IF(IDELTA.GE.4.) IRVEL=IRVEL-4096.
      IF(IDELTA.LE.-4.) IRVEL=IRVEL+4096.
C
C  *************************************************
C  * STEP 4: COMPUTE UNAMBIGUOUS VELOCITY ESTIMATE. *
C  *************************************************
C
C  STEP 4-1: ADD NUMBER OF AMBIGUOUS FILTER BANK WIDTHS TO ESTIMATE
C            OF FRACTIONAL FILTER BANK WIDTH. NOTE: LSB OF RESULTANT
C            ESTIMATE REPRESENTS 1/4096 OF A FILTER BANK WIDTH.
      IVEL=INTT(IRVEL-IFVEL)
C
C  STEP 4-2: SCALE LSB OF RESULTANT ESTIMATE TO 0.05 FEET/SEC.
C  DEFINITION:  VT2(MPRF)=((FILTER SEPARATION)/128.)/(VELOCITY LSB)
C                OR VT2(MPRF)=(PRF*LAMBDA)/(0.05*8196).
      IVEL=INTT(IVEL*VT2(MPRF)+0.5)
C
C  ************************************************
C  * STEP 5: COMPUTE SMOOTHED UNAMBIGUOUS VELOCITY *
C  ************************************************
C
C  STEP 5-1: UPDATE REGISTERS OF MOVING WINDOW AVERAGER.
      DO 20 I=1,3
   20 VEST(5-I)=VEST(4-I)
      VEST(1)=IVEL
C
```

FIGURE 2.5-1 BASELINE VERSION OF SUBROUTINE VELPRO

PAGE 2

ORIGINAL PAGE IS
OF POOR QUALITY

```
C  STEP 5-2: COMPUTE MOVING WINDOW AVERAGE AND SCALE ANSWER INTO        00028160
C            FEET/SEC FROM UNITS OF 0.05 FEET/SEC.                      00028170
      M=MPRF                                                            00028178
      M1=MW(1,M)
      M2=MW(2,M)
      M3=MW(3,M)
      M4=MW(4,M)
      SRDOT=0.0125*(VEST(M1     )+VEST(M2     )+VEST(M3     )+          00028180
     2              VEST(M4     ))                                      00028182
C                                                                       00028190
C  ************************************                                 00028200
C  * STEP 6: RESET DOPPLER FILTER BANK *                                00028210
C  ************************************                                 00028220
C                                                                       00028230
C  STEP 6-1: USE ON-TARGET DISCRIMINANT AND VELOCITY DISCRIMINANT TO    00028240
C            DETERMINE UPDATE OF FILTER BANK POSITION.                  00028250
C            THE FOLLOWING RULES ARE USED:                              00028260
C                                                                       00028270
C            CASE 1: ODISC>0. AND -51.<IVDISC<51. IMPLIES NO CHANGE.    00028280
C                                                                       00028290
C            CASE 2: ODISC>0. AND IVDISC>51. IMPLIES SHIFT -1.          00028300
C                                                                       00028310
C            CASE 3: ODISC>0. AND IVDISC<-51. IMPLIES SHIFT +1.         00028320
C                                                                       00028330
C            CASE 4: ODISC<0. AND IVDISC>0. IMPLIES SHIFT -2.           00028340
C                                                                       00028350
C            CASE 5: ODISC<0. AND IVDISC<0. IMPLIES SHIFT +2.           00028360
      IF(ODISC.GE.0.) GO TO 30                                          00028370
      IF(IVDISC.LT.0.) MDF(1)=MOD(MDF(1)+2,32)                          00028380
      IF(IVDISC.GE.0.) MDF(1)=MOD(MDF(1)+30,32)                         00028390
      GO TO 40                                                          00028400
   30 IF(IVDISC.GT.51.) MDF(1)=MOD(MDF(1)+31,32)                        00028410
      IF(IVDISC.LT.-51.) MDF(1)=MOD(MDF(1)+1,32)                        00028420
C                                                                       00028430
C  STEP 6-2: RESET REMAINING FILTERS IN THE BANK-OF-5.                  00028440
   40 DO 50 I=1,4                                                       00028450
   50 MDF(I+1)=MOD(MDF(1)+I,32)                                         00028460
      RETURN                                                            00028470
      END                                                               00028480
C                                                                       00012320
```

FIGURE 2.5-1 BASELINE VERSION OF SUBROUTINE VELPRO

PAGE 3

(THIS PAGE INTENTIONALLY LEFT BLANK)

```
C
C   ************************************************************
C   * THIS SUBROUTINE COMPUTES AN ACCURATE, SMOOTHED VELOCITY USING *
C   * THE KU-BAND RADAR VELOCITY PROCESSOR ALGORITHM.               *
C   ************************************************************
C
C
      SUBROUTINE VELPRO
      REAL IRDOT,IRNG,INTT,IVEL,IVDISC,IFVEL,IRVEL,IR1,IR2,IR3,
     2     IF3,IDELTA
      COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)
      COMMON /OUTPUT/IDUM0(3),SRNG,SRDOT,DUM2(5),IDUM(4)
      COMMON /ICNTL/I1DUM(14),MRNG,MSAM,MPRF,IDUM1(10),MPFOLD
      COMMON /SYSDAT/TSAM,DUMS(14)
      COMMON /RTDAT/IRDOT,IRNG,RBIAS,VEST(4),MDF(5)
      COMMON /DSCRM/DUM(2),RDISC,VDSC,RRTE,ODISC,DUM3(3)
      DIMENSION IPROM(128),VT1(3),VT2(3),MW(4,3)
      DATA IPROM/127,127,125,124,122,121,120,118,117,116,114,113,
     2 111,110,109,107,106,105,103,102,101,99,98,97,95,94,93,92,90,
     3 89,88,87,85,84,83,82,81,79,78,77,76,75,73,72,71,70,69,68,67,
     4 66,65,64,63,62,61,60,59,58,57,56,55,54,53,52,51,50,49,49,48,
     5 47,46,45,44,44,43,42,41,41,40,39,38,38,37,36,36,35,34,34,33,
     6 32,32,31,31,30,30,29,28,28,27,27,26,26,25,25,24,24,23,23,22,
     7 22,22,21,21,20,20,19,19,19,18,18,17,17,17,16,16,16,15,15/
      DATA VT1/1.012592E-2,2.362726E-2,2.633237E-1/,VT2/1.204935,
     2       0.5163982,0.04633489/
      DATA MW/1,2,3,4,1,1,2,2,1,1,1,1/
C
C
C   ******************************************************
C   SUBROUTINE VELPRO WAS MODIFIED FEB 6 1986 BY M. MEYER
C   MODIFICATIONS CONSISTED OF CHECKING THE VARIABLE MPRF
C   FOR A VALUE OF ONE (IMPLIES 7 KC MODE) AND IF TRUE
C   ASSUMING THE VELOCITY ESTIMATE GIVEN BY THE VELOCITY
C   DISCRIMINANT IS UNAMBIGUOUS.
C   ******************************************************
C
C   ********************************************
C   * STEP 1: GENERATE AMBIGUOUS VELOCITY ESTIMATE *
C   ********************************************
C
C   STEP 1-1: INTEGERIZE VELOCITY DISCRIMINANT AND CHECK FOR SATURATION.
      VDISC=5.333333*VDSC
      IVDISC=INTT(VDISC+0.5)
      IF(IVDISC.LT.-128.) IVDISC=-128.
      IF(IVDISC.GT.127.) IVDISC=127.
C
C   STEP 1-2: COMPUTE INTEGRAL FILTER NUMBER PORTION OF AMBIGUOUS
C             VELOCITY ESTIMATE.
      INTEG=MDF(2)
```

FIGURE 2.5-2 DELIVERABLE VERSION OF SUBROUTINE VELPRO

PAGE 1

```
      IF(IVDISC.LT.0.) INTEG=MOD(INTEG+1,32)
C
C  STEP 1-3: COMPUTE FRACTIONAL FILTER PORTION OF AMBIGUOUS VELOCITY
C            ESTIMATE.
C  ESTIMATE.
      IV1=INT(ABS(IVDISC))+1
C ***********************************
C CHANGED JAN 30 1986 BY H. MAGNUSSON
C ***********************************
      IF(IV1.GT.128)IV1=128
      IFRAC=IPROM(IV1)
      IF(IVDISC.LT.0.) IFRAC=127-IFRAC
C
C  STEP 1-4: COMPUTE AMBIGUOUS VELOCITY ESTIMATE —— COMBINE INTEGRAL
C            AND FRACTIONAL PARTS. NOTE: LSB IS 1/128 OF FILTER WIDTH.
C  FRACTIONAL PARTS. NOTE: LSB IS 1/128 OF A FILTER WIDTH.
      IFVEL=FLOAT(IFRAC+128*INTEG)
C  *****************************
C  CHANGED FEB 6 1986 BY M. MEYER
C  *****************************
C
       IF(MPRF.EQ.1) THEN
         IF(INTEG.GE.0.AND.INTEG.LE.21)THEN
           IRVEL=0.
         ELSE
           IRVEL=4096.
         END IF
         GO TO 8
       END IF
C  ***************************************
C  * STEP 2: SCALE ROUGH VELOCITY ESTIMATE *
C  ***************************************
C
C  STEP 2-1: SCALE LSB OF ROUGH RANGE RATE ESTIMATE TO 4 TIMES A DOPPLER
C            WIDTH.
C  DEFINITION:  VT1(MPRF)=(RANGE LSB)/((MAX. UNAMBIGUOUS VELOCITY)/8)
C                 OR VT1(MPRF)=5./(PRF*LAMBDA)
      R1=IRDOT*VT1(MPRF)/TSAM
      IR1=AINT(R1)
C
C  STEP 2-2: PERFORM SOME REQUIRED AUXILIARY CALCULATIONS.
      R2=IR1/8.
      IR2=AINT(R2)
      IRVEL=IR2*4096.
C
C  ****************************
C  * STEP 3: RESOLVE AMBIGUITY *
C  ****************************
C
C  STEP 3-1: COMPUTE 3 MSB'S OF AMBIGUOUS VELOCITY ESTIMATE.
      IF3=AINT(IFVEL/512.)
C
C  STEP 3-2: COMPUTE 3 LSB'S OF SCALED ROUGH RANGE RATE ESTIMATE.
      IR3=ABS(IR1-8.*IR2)
      IF(R1.LE.0.)GO TO 10
      IRVEL=IRVEL+4096.
      IR3=7.-IR3
   10 CONTINUE
C
C  STEP 3-3: COMPARE 3 MSB'S AND 3 LSB'S AND INCREMENT NUMBER OF
C            AMBIGUOUS FILTER BANK WIDTHS APPROPRIATELY.
      IDELTA=IR3-IF3
      IF(IDELTA.GE.4.) IRVEL=IRVEL-4096.
      IF(IDELTA.LE.-4.) IRVEL=IRVEL+4096.
```

FIGURE 2.5-2 DELIVERABLE VERSION OF SUBROUTINE VELPRO

PAGE 2

```
    8 CONTINUE
C
C  **************************************************
C  * STEP 4: COMPUTE UNAMBIGUOUS VELOCITY ESTIMATE. *
C  **************************************************
C
C  STEP 4-1: ADD NUMBER OF AMBIGUOUS FILTER BANK WIDTHS TO ESTIMATE
C            OF FRACTIONAL FILTER BANK WIDTH. NOTE: LSB OF RESULTANT
C            ESTIMATE REPRESENTS 1/4096 OF A FILTER BANK WIDTH.
      IVEL=INTT(IRVEL-IFVEL)
C
C  STEP 4-2: SCALE LSB OF RESULTANT ESTIMATE TO 0.05 FEET/SEC.
C  DEFINITION:  VT2(MPRF)=((FILTER SEPARATION)/128.)/(VELOCITY LSB)
C                OR VT2(MPRF)=(PRF*LAMBDA)/(0.05*8196).
      IVEL=INTT(IVEL*VT2(MPRF)+0.5)
C
C  *************************************************
C  * STEP 5: COMPUTE SMOOTHED UNAMBIGUOUS VELOCITY *
C  *************************************************
C
C  STEP 5-1: UPDATE REGISTERS OF MOVING WINDOW AVERAGER.
      DO 20 I=1,3
   20 VEST(5-I)=VEST(4-I)
      VEST(1)=IVEL
C
C  STEP 5-2: COMPUTE MOVING WINDOW AVERAGE AND SCALE ANSWER INTO
C            FEET/SEC FROM UNITS OF 0.05 FEET/SEC.
      M=MPRF
      M1=MW(1,M)
      M2=MW(2,M)
      M3=MW(3,M)
      M4=MW(4,M)
      SRDOT=0.0125*(VEST(M1      )+VEST(M2      )+VEST(M3      )+
     2              VEST(M4      ))
C
C  *************************************
C  * STEP 6: RESET DOPPLER FILTER BANK *
C  *************************************
C
C  STEP 6-1: USE ON-TARGET DISCRIMINANT AND VELOCITY DISCRIMINANT TO
C            DETERMINE UPDATE OF FILTER BANK POSITION.
C            THE FOLLOWING RULES ARE USED:
C
C            CASE 1: ODISC>0. AND -51.<IVDISC<51. IMPLIES NO CHANGE.
C
C            CASE 2: ODISC>0. AND IVDISC>51. IMPLIES SHIFT -1.
C
C            CASE 3: ODISC>0. AND IVDISC<-51. IMPLIES SHIFT +1.
C
C            CASE 4: ODISC<0. AND IVDISC>0. IMPLIES SHIFT -2.
C
C            CASE 5: ODISC<0. AND IVDISC<0. IMPLIES SHIFT +2.
      IF(ODISC.GE.0.) GO TO 30
      IF(IVDISC.LT.0.) MDF(1)=MOD(MDF(1)+2,32)
      IF(IVDISC.GE.0.) MDF(1)=MOD(MDF(1)+30,32)
      GO TO 40
   30 IF(IVDISC.GT.51.) MDF(1)=MOD(MDF(1)+31,32)
      IF(IVDISC.LT.-51.) MDF(1)=MOD(MDF(1)+1,32)
C
C  STEP 6-2: RESET REMAINING FILTERS IN THE BANK-OF-5.
   40 DO 50 I=1,4
   50 MDF(I+1)=MOD(MDF(1)+I,32)
      RETURN
      END
```

FIGURE 2.5-2 DELIVERABLE VERSION OF SUBROUTINE VELPRO

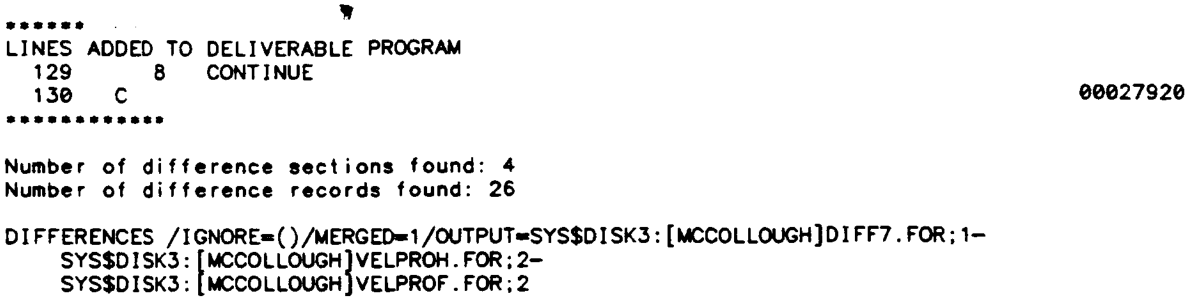
PAGE 3

(THIS PAGE INTENTIONALLY LEFT BLANK)

```
************
LINES DELETED FROM BASELINE PROGRAM
   42    C                                                                        00027290
******
LINES ADDED TO DELIVERABLE PROGRAM
   42    C
   43    C
   44    C  *******************************************************
   45    C  SUBROUTINE VELPRO WAS MODIFIED FEB 6 1986 BY M. MEYER
   46    C  MODIFICATIONS CONSISTED OF CHECKING THE VARIABLE MPRF
   47    C  FOR A VALUE OF ONE (IMPLIES 7 KC MODE) AND IF TRUE
   48    C  ASSUMING THE VELOCITY ESTIMATE GIVEN BY THE VELOCITY
   49    C  DISCRIMINANT IS UNAMBIGUOUS.
   50    C  *******************************************************
   51    C                                                                        00027290
************
************
LINES DELETED FROM BASELINE PROGRAM
   62           IFRAC=IPROM(IV1)                                                  00027490
******
LINES ADDED TO DELIVERABLE PROGRAM
   71    C *********************************
   72    C CHANGED JAN 30 1986 BY H. MAGNUSSON
   73    C *********************************
   74           IF(IV1.GT.128)IV1=128
   75           IFRAC=IPROM(IV1)                                                  00027490
************
************
LINES DELETED FROM BASELINE PROGRAM
   69    C                                                                        00027560
   70    C  ****************************************                              00027570
******
LINES ADDED TO DELIVERABLE PROGRAM
   82    C  ****************************
   83    C  CHANGED FEB 6 1986 BY M. MEYER
   84    C  ****************************
   85    C
   86            IF(MPRF.EQ.1) THEN
   87               IF(INTEG.GE.0.AND.INTEG.LE.21)THEN
   88                 IRVEL=0.
   89               ELSE
   90                 IRVEL=4096.
   91               END IF
   92               GO TO 8
   93            END IF
   94    C  ****************************************                              00027570
************
************
LINES DELETED FROM BASELINE PROGRAM
  105    C                                                                        00027920
```

FIGURE 2.5-3 SUMMARY OF MODIFICATIONS TO SUBROUTINE VELPRO

PAGE 1



```
******
LINES ADDED TO DELIVERABLE PROGRAM
  129      8   CONTINUE
  130   C                                                            00027920
************

Number of difference sections found: 4
Number of difference records found: 26

DIFFERENCES /IGNORE=()/MERGED=1/OUTPUT=SYS$DISK3:[MCCOLLOUGH]DIFF7.FOR;1-
    SYS$DISK3:[MCCOLLOUGH]VELPROH.FOR;2-
    SYS$DISK3:[MCCOLLOUGH]VELPROF.FOR;2
```

FIGURE 2.5-3 . SUMMARY OF MODIFICATIONS TO SUBROUTINE VELPRO

PAGE 2

2.5.4 Integration and Test Data

2.5.4.1 Test Definition

The philosophy for validating the ambiguity resolver modification was to test two things: (1) the boundaries where the velocity goes ambiguous in the 7 kHz mode (+ 75 feet per second or - 175 feet per second) and (2) to insure that the velocity becomes unambiguous when the PRF is switched to 3 kHz. Two simulation scenarios were defined to test these two features.

To check the boundaries on the ambiguous velocity in the 7 kHz PRF mode, the following scenario was used. A 10 dBsm target was given an initial range of 30,000 feet and an initial opening velocity of +100 feet per second. This velocity was held for 100 seconds and then target was decelerated at a rate of 1 foot per second for the next 300 seconds. At this point, the scenario was terminated. Plots of the range and range rate time histories are provided in Figures 2.5-4 and 2.5-5, respectively.

A similar scenario was used to insure that the velocity becomes unambiguous when the PRF is switched to 3 kHz and vice-versa. In fact, the range rate profile is identical to that given in Figure 2.5-5. The initial range in ths case is 42,000 feet, so the range profile is shifted upward by +12,000 feet as shown in Figure 2.5-6. The purpose of this shifted profile is to insure that the radar transitions to the 3 kHz PRF. The design of this scenario demonstrates that ambiguous opening targets become unambiguous when transitioning from 7 kHz to 3 kHz PRF. It also demonstrates that a closing target with velocity less than -175 feet per second will become ambiguous at the transition from 3 kHz PRF to 7 kHz PRF.

2.5.4.2 Test Results

Figure 2.5-7 gives a plot of the difference between the true target velocity and target velocity predicted by the radar as a function of time. A comparison of this plot against the range rate profile of Figure 2.5-5 shows that the velocity processor model has the proper boundaries on the unambiguous zone: (+75 fps, -175 fps).

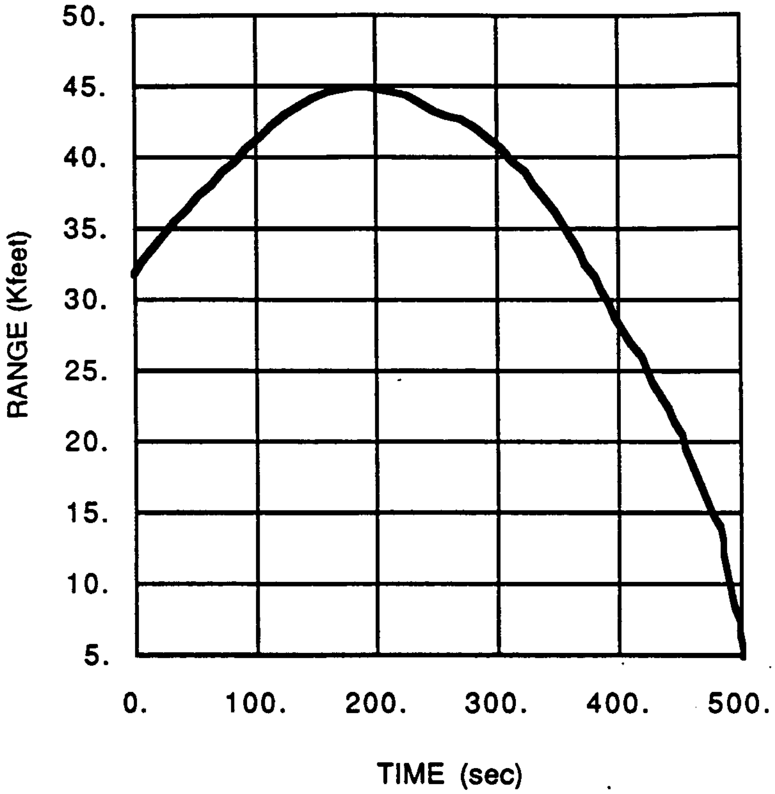


FIGURE 2.5-4 RANGE PROFILE FOR SCENARIO NO. 1

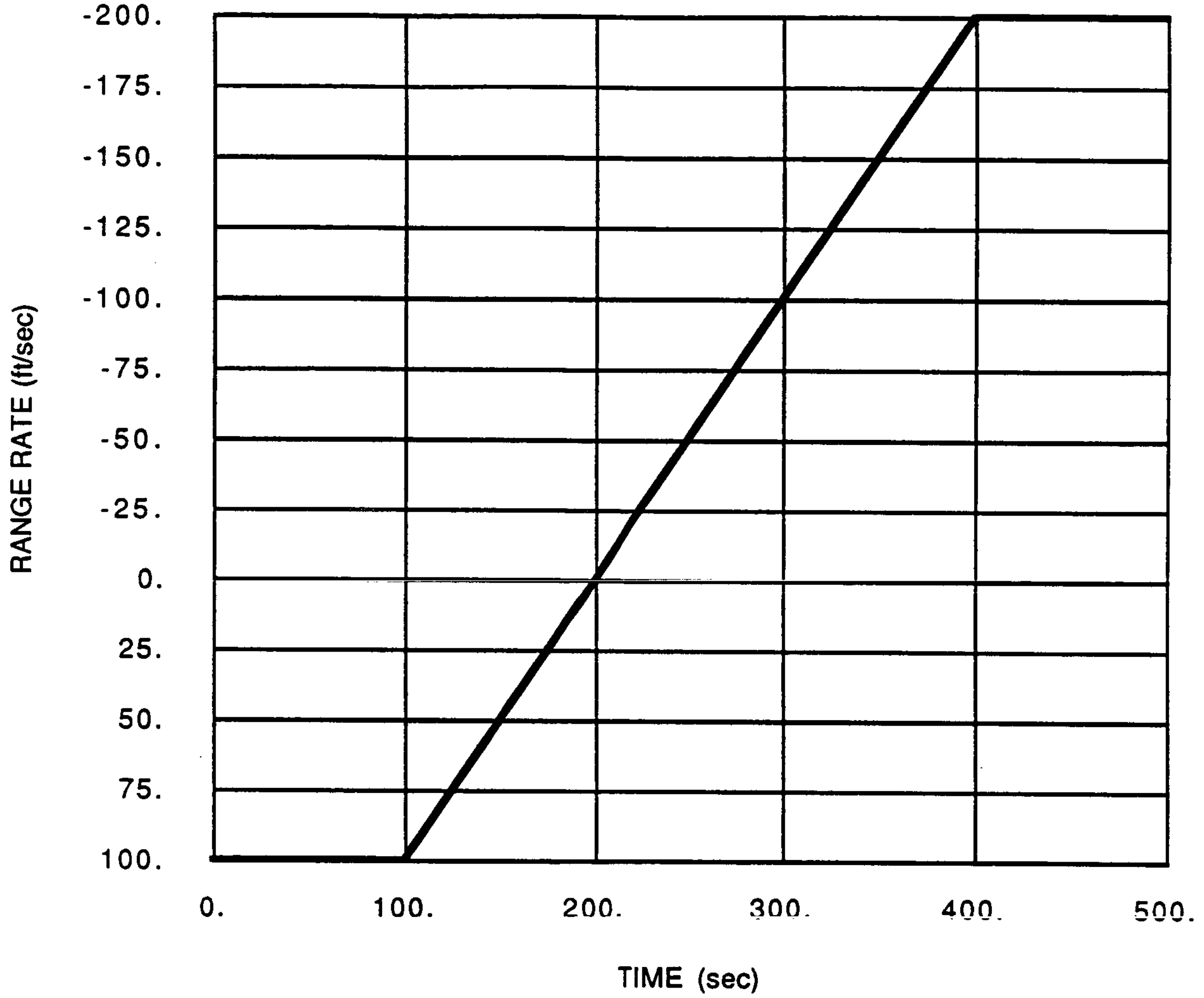


FIGURE 2.5-5 RANGE RATE PROFILE FOR SCENARIOS NO. 1 AND NO. 2

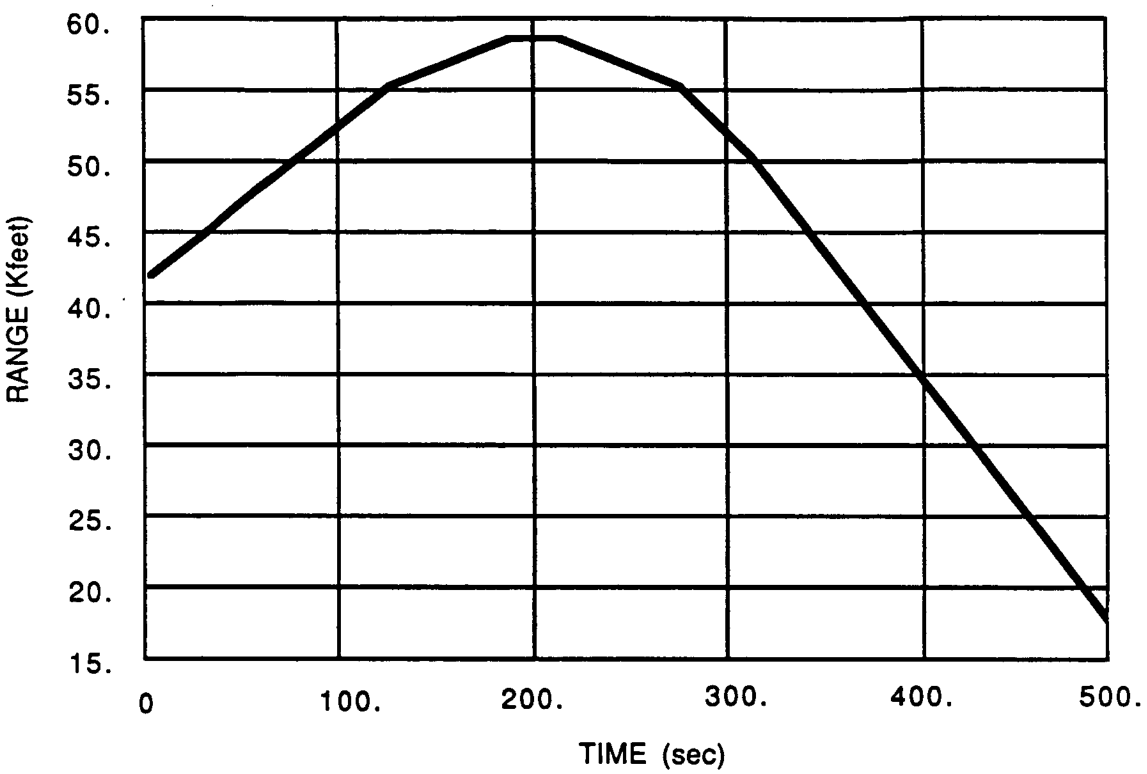


FIGURE 2.5-6 RANGE PROFILE FOR SCENARIO NO. 2

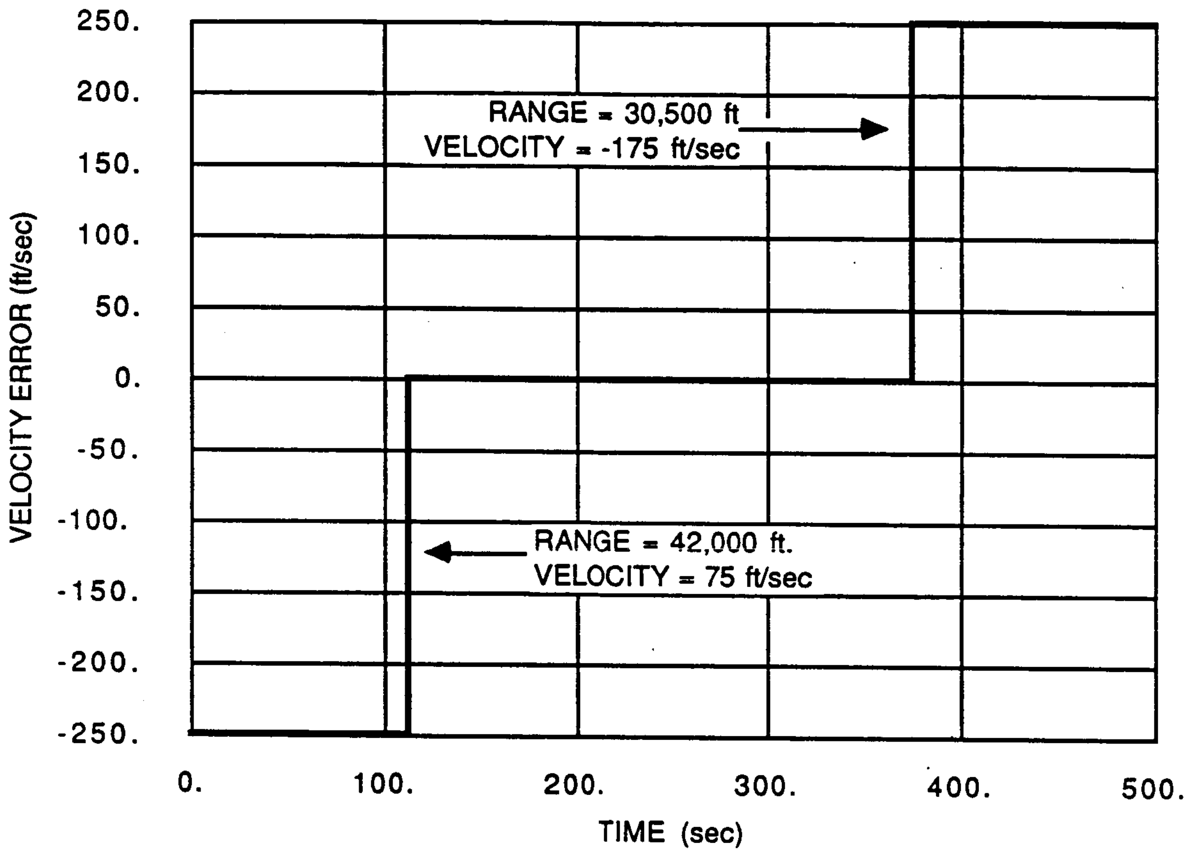


FIGURE 2.5-7 VELOCITY ERROR FOR SCENARIO NO. 1 DEFINED BY FIGURES 2.5-4 AND 2.5-5

Figure 2.5-8 gives a plot similar to Figure 2.5-7 for the second scenario. In this case, the velocity difference time history should be compared to the range profile plot of Figure 2.5-6. Taken together, these data show that velocity becomes unambiguous at the transition from 7 kHz to 3 kHz PRF at a range of 49,920 feet and the velocity becomes ambiguous at a the 3 kHz to 7 kHz PRF transition at a range of 43,510 feet. Notice that this second test validates the PRF hysteresis loop as well.

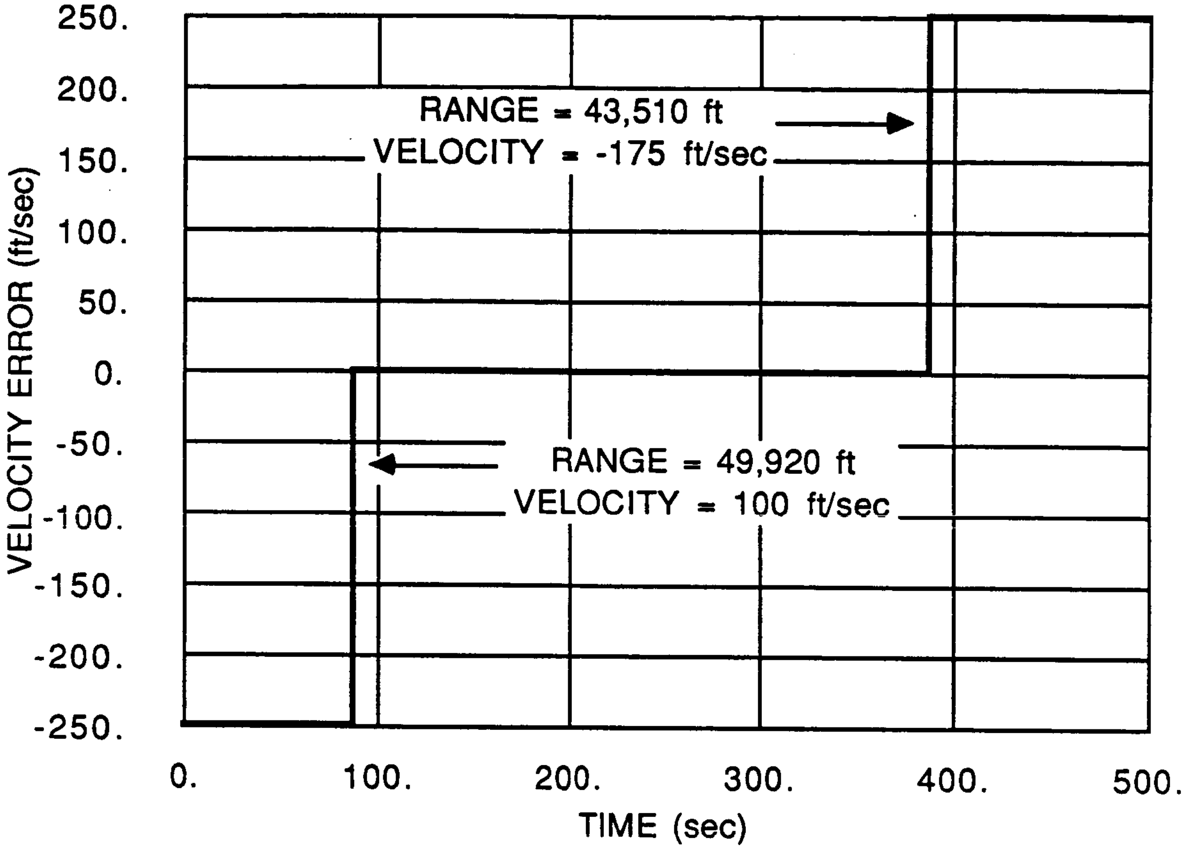


FIGURE 2.5-8 VELOCITY ERROR FOR SCENARIO NO.2 DEFINED BY FIGURES 2.5-5 AND 2.5-6

(THIS PAGE INTENTIONALLY LEFT BLANK)

3.0 SORTE DATA ANALYSIS

The purpose of this section is to describe the extent of the analysis performed on the SORTE data and provide the results of that analysis. Section 3.1 provides some background data on the SORTE program, describing the test setup and the test procedures. Section 3.2 defines the approach in the analysis and a summary of the findings. Section 3.3 describes the Ku-Band Radar's range measurement performance. Section 3.4 provides analysis of the range rate measurements. Sections 3.5 and 3.6 provide an analysis of the angle and angle rate measurements. Finally, Section 3.7 gives a comparison of simulation generated data and the SORTE data. The simulation data was generated by injecting the corresponding SORTE trajectory into the simulation.

3.1 SORTE PROGRAM SUMMARY

The purpose of the Shuttle Orbiter Radar Test and Evaluation (SORTE) program was to evaluate the accuracies of the following Ku-Band Radar measurements: range, range rate, roll angle, pitch angle, ILOS roll rate, and ILOS pitch rate. These accuracies were to be determined by using the precision measuring system at the White Sands Missile Range (WSMR) as a reference. In the following paragraphs a brief description of the test setup, test procedures, and post-test data processing will be provided for reference throughout Section 3.

3.1.1 Flight Trajectory and Target Selection

Selection of trajectories and targets was driven by the test objectives. The principal objective was to determine the Ku-Band Radar measurement accuracies using flight conditions that simulate an actual shuttle-satellite rendezvous as closely as possible. Since actual rendezvous data existed at the time the SORTE test trajectories were defined, these trajectories were patterned after the Solar Maximum Mission Satellite (SMMS) - Shuttle rendezvous obtained from Mission 41C in April 1984. Figure 3.1-1 gives a range history of the rendezvous and Figure 3.1-2 gives a range versus

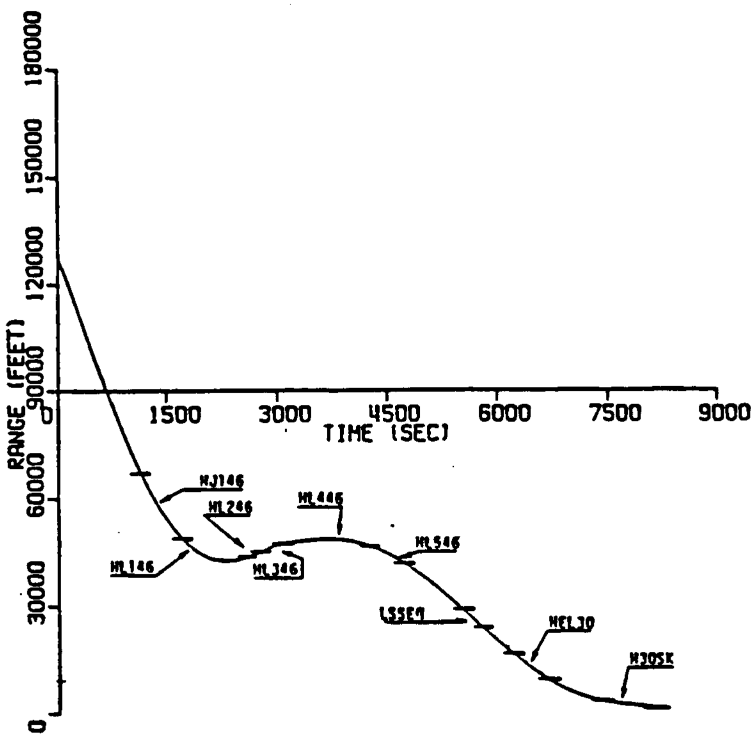


FIGURE 3.1-1 RANGE HISTORY FOR SHUTTLE-SMMS RENDEZVOUS DURING MISSION 41C IN APRIL 1984

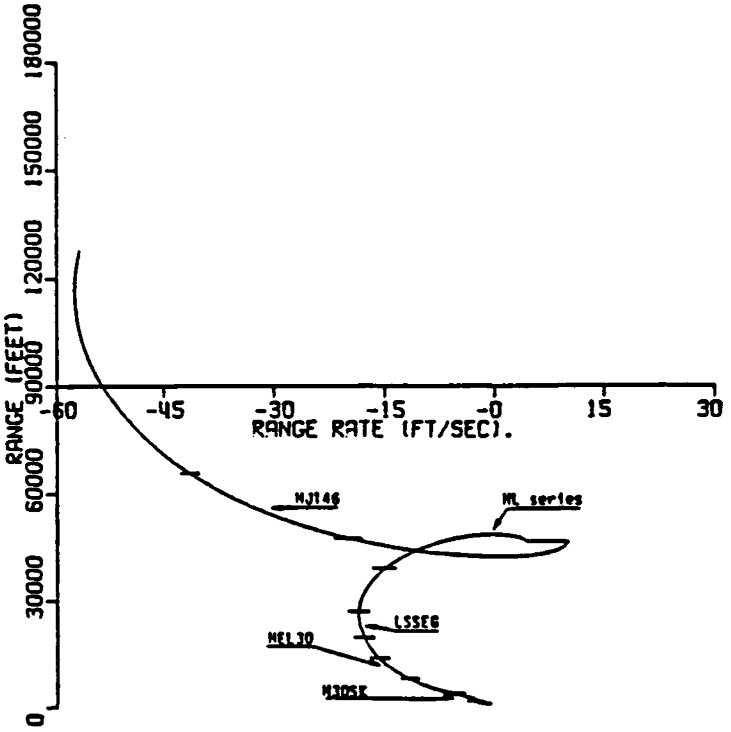


FIGURE 3.1-2 RANGE VERSUS RANGE RATE PROFILE FOR SHUTTLE-SMMS RENDEZVOUS DURING MISSION 41C IN APRIL 1984

range rate plot of the rendezvous. As shown on these two figures, the trajectory is divided up into several smaller trajectories which are labeled as shown in the figure. The principal reason for subdividing the trajectory was a 10 minute upper limit on the length of a given test run. This limit was established to avoid data tape changes, causing loss of data, during the tests. Table 3.1-1 (Reference 1) gives the range interval and range rate interval of operation for each of these tests.

TABLE 3.1-1 RANGE AND RANGE RATE COVERAGE BY TEST RUN

Test Run	Range (thousands of feet)	Range Rate (feet per second)
H30SKAE	3.6 to 2.6	0.0 to 33.5
H30SKAF	2.2 to 4.1	-22.0 to - 3.0
H30SKAG	2.2 to 2.6	-12.0 to - 3.0
H30SKAH	2.2 to 4.0	-31.0 to - 4.0
H30SKAI	3.3 to 3.6	-18.4 to 4.8
HEL30AF	7.0 to 12.2	-60.0 to 7.0
HEL30AG	7.0 to 13.0	-56.0 to - 8.0
HEL30AI	5.6 to 13.3	-38.0 to -10.0
HEL30AJ	6.0 to 13.6	-45.0 to 0.0
HJ146AC	45.0 to 63.0	-55.0 to -10.0
HJ146AD	47.0 to 64.0	-50.0 to 4.0
HJ146AE	46.0 to 65.0	-21.0 to 10.0
HL146AE	42.6 to 46.5	-21.0 to 10.0
HL246AD	43.4 to 47.3	4.0 to 17.5
HL246AE	41.5 to 47.2	- 5.0 to 20.0
HL346AD	48.0 to 48.9	- 6.0 to 14.0
HL346AE	47.0 to 49.0	-23.0 to 26.0
HL346AF	47.1 to 49.1	- 5.0 to 13.0
HL446AC	47.5 to 48.8	-13.0 to 10.0
HL446AD	47.0 to 49.7	-20.0 to 37.0
HL446AE	46.8 to 49.3	-16.0 to 16.0
HL546AC	41.3 to 47.0	-22.0 to 0.0
HL546AE	41.2 to 47.7	-21.0 to 55.0
HL546AF	40.8 to 46.5	-21.0 to 7.5
HL546AG	40.7 to 46.7	-25.0 to 25.0

The target selected for use in these flight tests was a UH-1H helicopter. To enhance the Radar Cross Section (RCS) of the helicopter, a pair of Luneberg lenses were mounted on the underside of the helicopter as shown in Figure 3.1-3. The "main beams" of these lenses were angled off from the helicopter nose and were pointed downward slightly. As will be shown in

ORIGINAL PAGE IS
OF POOR QUALITY

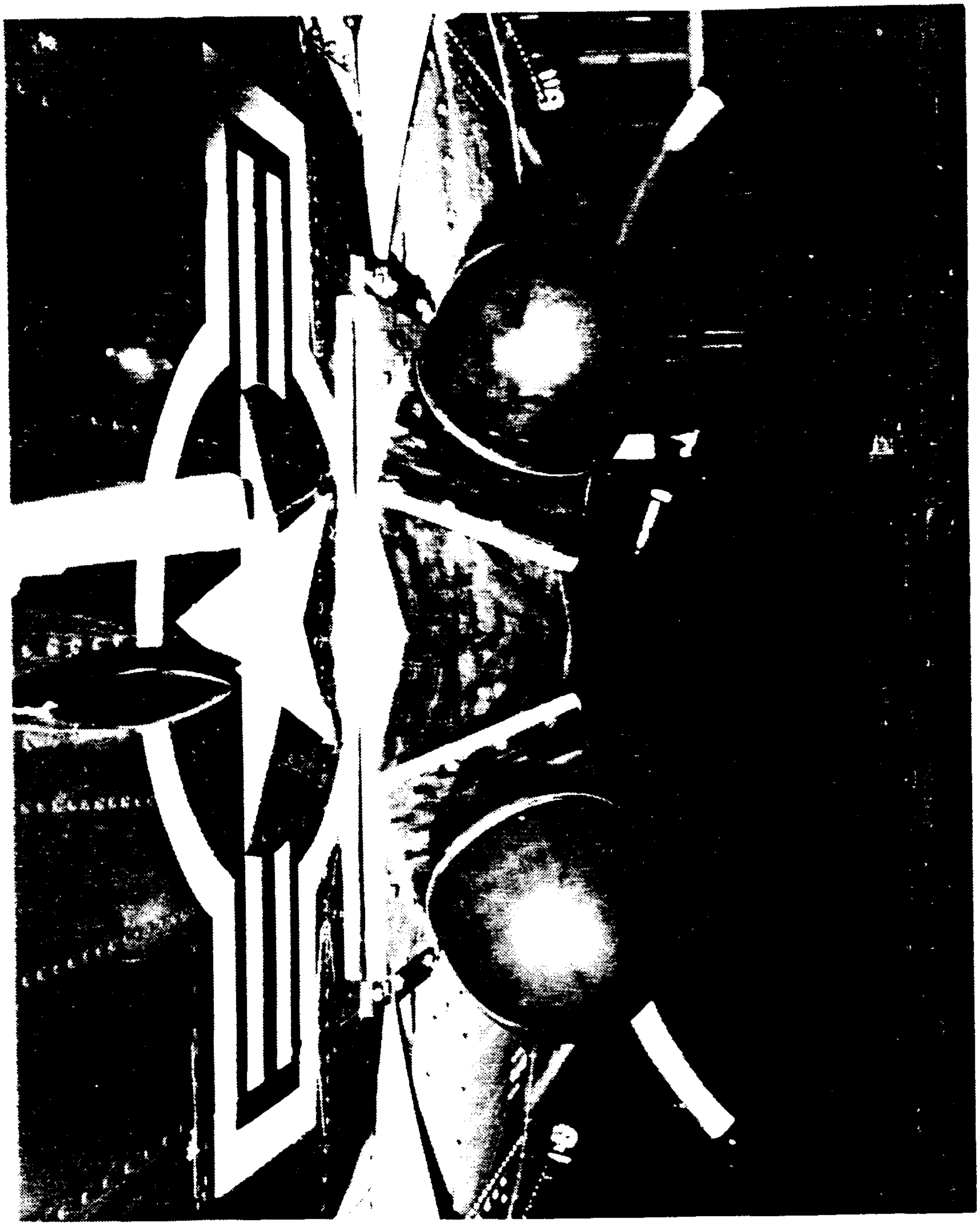


FIGURE 3.1-3 DUAL LUNEBERG LENS INSTALLATION ON THE HU-1H HELICOPTER

the analysis of the range rate performance in Section 3.3, these target enhancements were effective for those trajectories where the helicopter flew approximately down the Line-of-Sight (LOS) of the radar. However, this enhancement configuration was not effective when the helicopter flew a trajectory that was perpendicular, or Cross Line-of-Sight (XLOS) to the radar.

A second series of tests was based on the second major objective of the SORTE program: determining the effects of a conducting and non-conducting tether in the radar antenna beam. The purpose of these tests was to evaluate the usefulness of the Ku-Band Radar for tracking the Tethered Satellite System (TSS) on a future shuttle mission. The target for these tests consisted of a mockup of the TSS suspended from two, 10-foot inch diameter, Helium-filled balloons. This target was then tethered with a conducting or non-conducting tether. (As an aside, a red colored, Helium-filled balloon was tied to the tether at a point 50 feet below the main target to provide a secondary target for the cinetheodolites.) This balloon/target combination was then flown as closely as possible directly overhead relative to the radar. With the tether spool anchored within 20 feet of the radar, a significant portion of the tether would be in the beam when the target was directly overhead. Again the test duration was 10 minutes and altitudes from 300 to 3000 feet were planned to simulate reeling in and reeling out the TSS.

A third series of tests were performed. These tests consisted of filling a two meter in diameter Gemsphere[1] with Helium, releasing it near the site of the radar, and tracking it for 10 minutes.

Table 3.1-2 summarizes the range and range rate intervals for the "tether tests" and the Gemsphere release tests. The tether tests are denoted by "SAT" and the Gemsphere tests are denoted by "BAL" or "GEM".

[1] A Gemsphere is a metallic coated balloon with small protrusions (2-3 inches) distributed uniformly over the surface. These spheres are used by the National Weather Service to track upper atmosphere wind currents.

TABLE 3.1-2 RANGE AND RANGE RATE COVERAGE BY TEST RUN FOR TETHERED BALLOON AND GEMSPHERE TESTS

Test Run	Range (thousands of feet)	Range Rate (feet per second)
SAT1	2.5 to 2.6	- 5.0 to 5.0
SAT2	2.5 to 2.5	- 4.0 to 4.0
SAT3	1.2 to 2.5	- 6.0 to 4.0
SAT4 *	10.9 to 12.7	- 4.0 to 10.0
BAL1	0.8 to 10.4	2.5 to 29.0
BAL2	0.8 to 5.5	0.0 to 29.0
BAL5	8.0 to 10.3	8.0 to 17.0
BAL6	1.0 to 10.6	2.5 to 7.5
BAL7	1.0 to 10.6	7.5 to 31.0
GEM2	3.2 to 30.0	32.0 to 70.0
GEM3	2.1 to 26.0	30.0 to 68.0

* The tether broke between tests SAT3 and SAT4 so that the target was held only by a guidewire 12 kft in length.

Please note that the above summary is not meant to be an exhaustive summary of the SORTE tests, but a summary of the three principal series of tests for which data analysis has been performed and included in this report.

3.1.2 Test Setup

The radar was situated very near the brass cap at the PEARL site at WSMR. Figure 3.1-4 (taken from Reference 1) shows the PEARL site relative to the layout of the entire White Sands Range. The deployed assembly of the radar, including the transmitter, receiver, gimbal and antenna, were placed on a platform inside a radome near the brass cap. The radar was a few feet

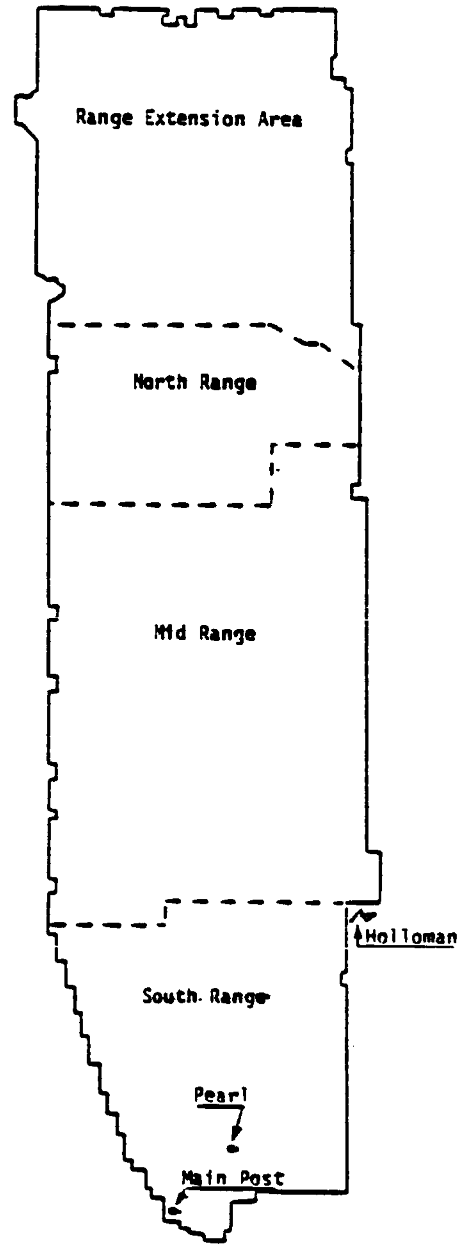


FIGURE 3.1-4 ILLUSTRATION OF THE PEARL SITE IN RELATION TO THE WHITE SANDS MISSILE RANGE LAYOUT

south and east of the brass cap and about 20 feet higher. Its exact White Sands Coordinate System (WSCS) location (from Reference 1) was:

East:	485,227.79 feet
North:	265,161.98 feet
Up:	2,618.43 feet.

The deployed assembly was oriented so that 0 degrees alpha and 0 degrees beta corresponded to the antenna boresight pointing 30 degrees east of north and elevated 30 degrees. This orientation was chosen to reduce the stress on the gimbals in a 1-g environment and to avoid ground clutter during radar operation.

Two types of sensor systems were used by WSMR to provide a target tracking reference. One system of sensors consisted of a set of cinetheodolites, designated as cines in the rest of this report. This set usually consisted of five cines for a given flight test. These five cines were chosen from a large number of cines which are widely distributed over the southern end of WSMR. Choice of the five cines for a given test was based upon the geometry of the flight profile for that particular test.

The second system of sensors consisted of a set of three AN/MPS-36 radars, denoted as R350, R393, and R394 by WSWR. Data from these radars is combined and processed to produce an estimate of target range and range rate. The combination of these radars and the post flight signal processing is called the Target Motion Resolution (TMR) system at White Sands. Details of TMR data processing are described in Reference 8.

Figure 3.1-5 (from Reference 1) gives a view of the Ku-Band Radar position, the cine positions (for a given set of trajectories), and the TMR radar positions in WSCS. It also provides the ground track for the HJ146, HL246, and HEL30 trajectories.

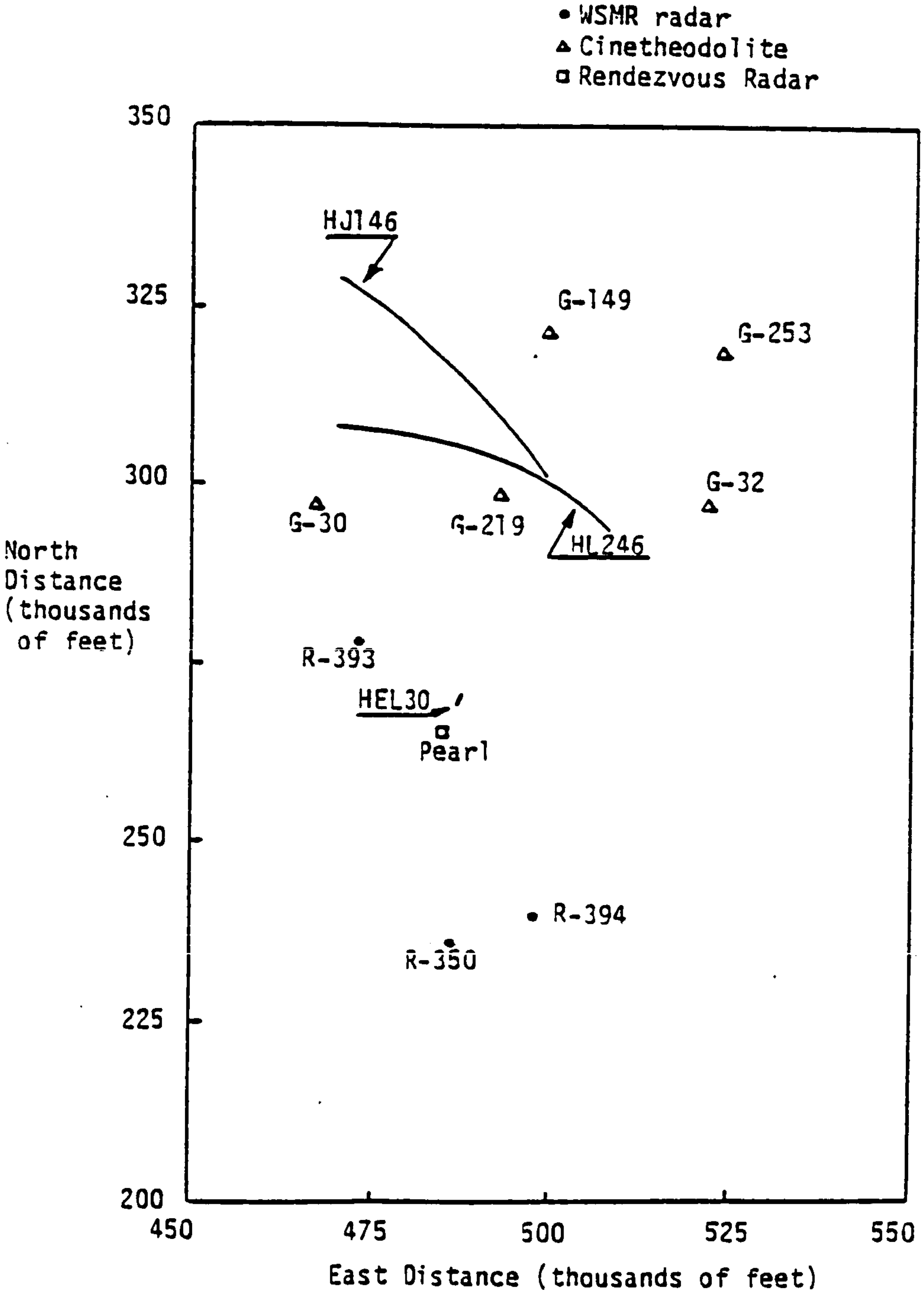


FIGURE 3.1-5 POSITIONS OF THE KUBAND RADAR, THE CINES, AND THE WSMR FOR SOME EXAMPLE TARGET TRAJECTORIES

3.1.3 Data Acquisition and Processing

The common element among the three data acquisition systems, the Ku-Band Radar, the cines and the TMR, was the time stamping of the data gathered by each system. WSMR provided universal timing which was networked to each radar and cine site and to the Ku-Band Radar so that the data could be time coded as it was gathered.

3.1.3.1 Ku-Band Radar Data Processing

Ku-Band Radar data acquisition for the SORTE program is best summarized via the illustration of Figure 3.1-6. Two types of data were gathered on the system test equipment (STE) computer (LSI 4/90): data from the MDM output and analog data which was digitized and recorded on disk. Each set of data included a range time stamp.

Once the Ku-Band Radar data for a particular test was recorded at the PEARL site, the disk was taken to Building 1646 at WSMR to be processed by a second LSI/490 computer. The purpose of this processing was to transfer the data in a VAX 11/780 compatible format to tape. Two tapes were made: one was for storage at the WSMR data processing facility and the other was to be used at Johnson Space Center (JSC) for further data analysis on the Building 44 VAX 11/780.

3.1.3.2 WSMR Sensor Data Processing

Data acquired by the individual WSMR radars and cines is summarized in Table 3.1-3. The data gathered by the various radar and cine sites is passed in real-time over the Precision Acquisition System (PAS) network to the central data processing facility at WSMR. This data is then post-processed to produce three sets of data. Each data set consists of the target position (X, Y, Z), the target velocity ($\dot{X}$, $\dot{Y}$, $\dot{Z}$), and the time code for the entire flight test. Target position and velocity values are given in the PEARL site brass cap coordinate system which is a North-East-Down (NED) system whose origin resides at the brass cap. The three post processing methods are described below.

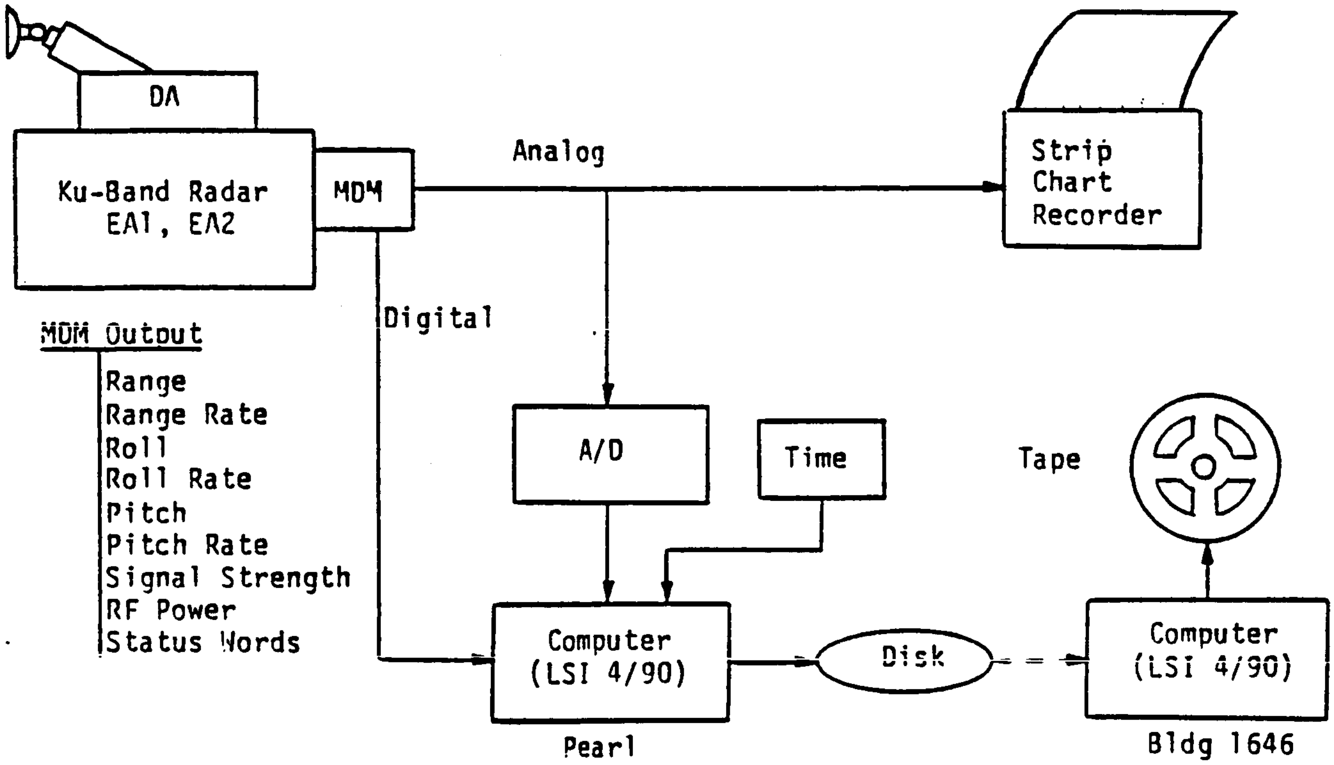


FIGURE 3.1-6 ILLUSTRATION OF THE KUBAND RADAR DATA ACQUISITION PROCESS FOR THE SORTE PROGRAM AT WSMR

TABLE 3.1-3 WSMR RADAR AND CINE DATA ACQUISITION ITEMS

WSMR RADARS	WSMR CINETHEODOLITES
Range Range Rate Azimuth Elevation Range Time	Azimuth Elevation Range Time

The first data set is obtained by processing only cinetheodolite data to produce target position and velocity as a function of range time. This data set is called cine data in the sequel. Data from the three WSMR Radars is processed using the TMR algorithms to produce target position and velocity as a function of range time. This second data set is denoted as the TMR data throughout the remainder of the report. The third data set combines the best features of the cine processing and the TMR processing. The cines produce highly accurate position data, while the TMR produces very accurate velocity data. Hence, the new system, called the "BEST" system, uses the cine data for the initial position estimate and propogates the position using velocity data from the TMR.

All three data sets were generated for those flight tests where both systems of sensors were operable. Table 3.1-4 taken from Reference 1 summarizes the available sensors for each of the flight tests.

3.1.3.3 Final Data Processing

At this point in the data processing scheme, the Ku-Band Radar data resides on a VAX 11/780 compatible tape. These data are in standard shuttle orbiter body coordinates. The post-processed WSMR data has also been loaded onto tape in a VAX 11/780 format and delivered to JSC. These WSMR Sensor data have been converted to the PEARL site brass cap coordinate system described in the previous subsection.

TABLE 3.1-4 AVAILABLE WSMR SENSORS FOR EACH TEST RUN

Test Run	Sensors
BAL1, Nov 4, TO-60302	Radar 394, no optics.
BAL2, Nov 4, TO-61350	Radar 394, no optics.
BAL5, Nov 4, TO-62785	Radar 394, no optics.
BAL6, Nov 4, TO-63348	Radar 394, no optics.
BAL7, Nov 4, TO-63346	Radar 394, no optics.
GEM2, Oct 16, TO-76421	TMR, no optics.
GEM3, Oct 16, TO-77603	TMR, no optics.
H30SKAE, Oct 3, TO-56647	TMR and optics.
H30SKAF, Oct 3, TO-56987	TMR and optics.
H30SKAG, Oct 3, TO-60657	TMR and optics.
H30SKAH, Oct 3, TO-60821	TMR and optics.
H30SKAI, Oct 3, TO-61113	TMR and optics.
HEL30AF, Oct 3, TO-56123	TMR and optics.
HEL30AG, Oct 3, TO-57558	TMR and optics.
HEL30AI, Oct 3, TO-61665	TMR and optics.
HEL30AJ, Oct 3, TO-62488	TMR and optics.
HJ146AC, Oct 1, TO-67031	TMR, no optics.
HJ146AD, Oct 5, TO-62415	TMR and optics.
HJ146AE, Nov 4, TO-80843	Radar 394 and optics.
HL146AE, Nov 4, TO-76124	Radar 394 and optics.
HL246AD, Oct 1, TO-60295	Radar, reduced optics.
HL246AE, Oct 5, TO-55880	TMR and optics.
HL346AD, Oct 1, TO-65780	TMR, no optics.
HL346AE, Oct 5, TO-61367	TMR and optics.
HL346AF, Nov 4, TO-79738	Radar 394 and optics.
HL446AC, Oct 1, TO-61463	TMR and optics.
HL446AD, Oct 5, TO-57012	TMR and optics.
HL446AE, Nov 4, TO-75072	Radar 394 and optics.
HL546AC, Oct 1, TO-59240	TMR, no optics.
HL546AE, Oct 5, TO-54805	TMR and optics.
HL546AF, Oct 5, TO-63406	TMR and optics.
HL546AG, Nov 4, TO-72220	Radar 394 and optics.
SAT1, Oct 19, TO-50988	TMR and optics.
SAT2, Oct 19, TO-52227	TMR and optics.
SAT3, Oct 19, TO-53295	TMR and optics.
SAT4, Oct 19, TO-55207	TMR and optics.
SAT6, Oct 19, (Acquisition)	TMR, no optics.
SAT8, Oct 19, (Acquisition)	TMR, no optics.

The final component in the processing was performed on the computers at JSC by NASA and LEMSCO personnel and consisted of several steps. The first step involved transforming the WSMR sensor data from brass cap coordinates to the shuttle body coordinates. The mathematics of this transformation were developed by Bill Culpepper of LEMSCO and are documented in Reference 9. The next step was to compute difference profiles for each of the radar parameters of interest. This means that for a given radar parameter, the Ku-Band Radar data profile was subtracted from the corresponding WSMR sensor data profile to produce the difference data profile. The final step was a statistical analysis of the resulting difference profile to produce a mean and a standard deviation for the interval and a diagram of this processing is shown in Figure 3.1-7. A sample result of this processing procedure is shown in Figure 3.1-8.

The procedure for analyzing this proccessed data is outlined in Section 3.2.

3.1.4 Summary of Flight Tests

There were 44 flight tests where data was gathered by both the WSMR sensors and the Ku-Band Radar. Careful notes were compiled by A. C. Lindberg of LEMSCO concerning the weather conditions and any anomalies that occurred during each of the tests. These notes, along with observations about the difference data profiles, are given in a summary form in Appendex G. Results of an extensive analysis of this data follows below.

3.2 ANALYSIS APPROACH AND PRELIMINARY FINDINGS

As anticipated (Reference 10) the SORTE data analysis activity was very limited due to available contract resources. Since this was expected, an analysis procedure was formulated to optimize the data reduction effort. The method developed was a two step procedure. The first step consisted of one complete pass through the data to identify any major problem areas. In the second step an extensive analysis of these problem areas was undertaken. The intent of this second step was to identify the dominant error source (or sources) and develop a quantitative estimate of its effect. The next level of priority in the data analysis was to resolve any significant anomolies found in the data.

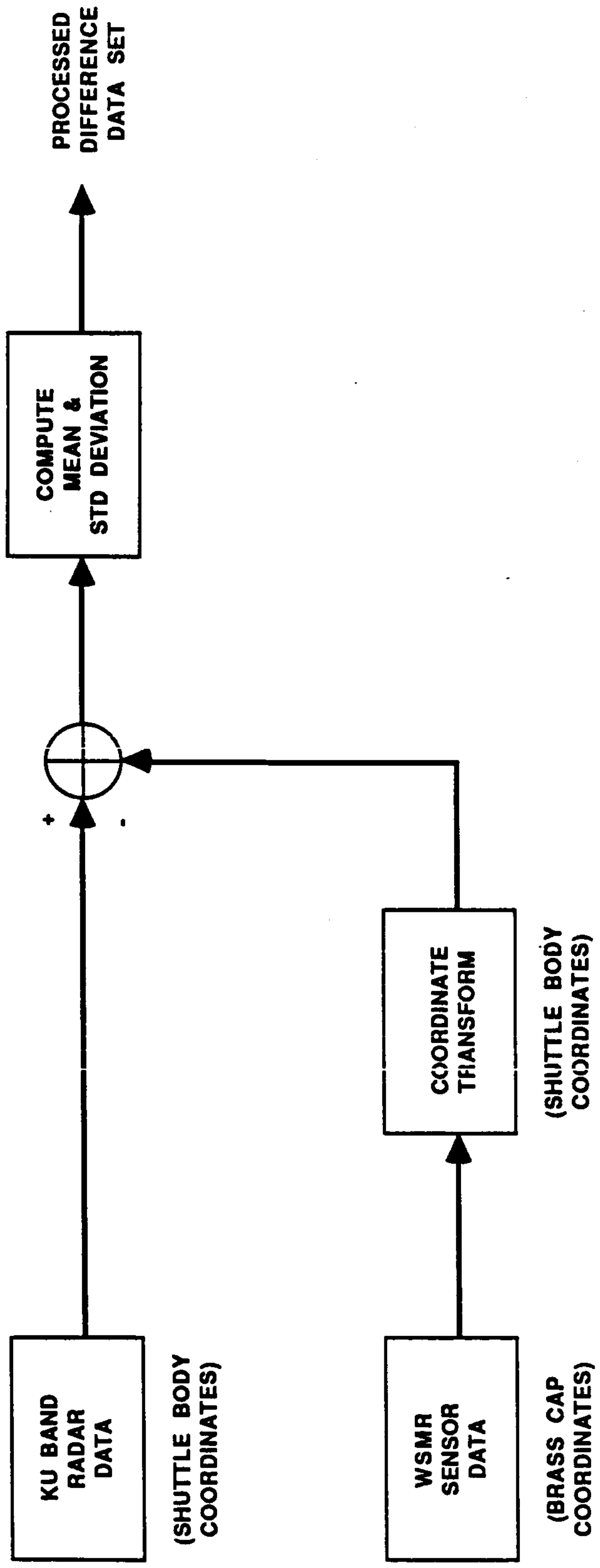


FIGURE 3.1-7 SIMPLIFIED DIAGRAM OF FINAL PROCESSING OF WSMR SENSOR AND KU BAND RADAR DATA

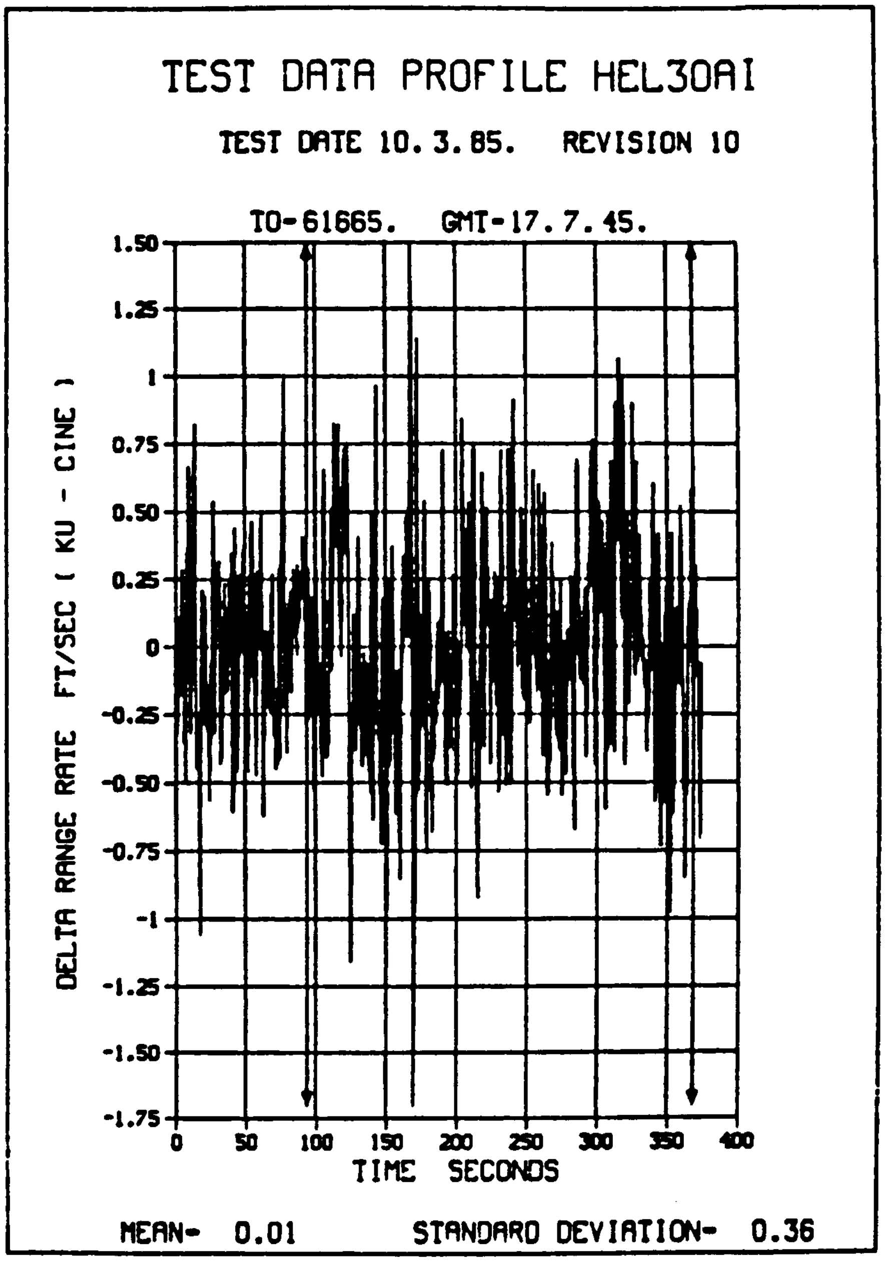


FIGURE 3.1-8 EXAMPLE OF A DIFFERENCE DATA PLOT

3.2.1 Preliminary Findings

In the first step of the procedure, the means and standard deviations of the difference data was compared against the corresponding radar specification (listed in Table 1-2) to determine which parameters were within specification for each test run. This test surfaced major problems in the following parameters (also see Table 1-3 of Section 1):

(1) Range rate standard deviation (95% failure)

(2) Roll rate mean and standard deviation (93% failure)

(3) Pitch rate mean and standard deviation (100% failure).

Problems of a smaller magnitude were also found in the angle data:

(4) Roll angle standard deviations (43% failure)

(5) Pitch angle standard deviations (19% failure).

Extensive analyses of the areas identified above were then undertaken. Results of these analyses are summarized in the following subsections. However, there are some general observations from these analyses that can be stated here. Almost all of the problems in the data can be attributed to the following categories:

(1) Large errors in the sensor data due to the sensor configuration and target position. This problem is commonly called Geometric Dilution of Precision or GDOP.

(2) Target acceleration in both range and angle.

(3) Low signal-to-noise power ratio (SNR) at the doppler filter output. This is principally due to a small radar cross section (RCS).

In addition, there were some general observations concerning the dominant error sources. These are that

(4) Different flight trajectories had different dominant error sources.

(5) The dominant error source could change within a given flight trajectory.

These observations on dominant error sources were found to be prevalent in the range rate analysis. Angle acceleration and transformation errors were found to be the major contributors to errors in the roll and pitch angle data. Angle acceleration and a scale factor error were the significant contributors to the problems in the ILOS roll and pitch angle rate data. All of these problems are discussed in detail in the following subsections.

3.3 RANGE DATA ANALYSIS

The first cut at analyzing the range error data was quite encouraging. The standard deviation of the range difference data was beyond the specification limit on four flights, and the mean was outside the specification limit on three flights. These cases are summarized in Table 3.3-1. In addition to these few problems with the range error data statistics, there were some anomalies in the range data. All of these anomalies took the form of discontinuous jumps in the Ku-Band Radar range estimate.

TABLE 3.3-1 SUMMARY OF FIRST CUT AT RANGE ERROR DATA ANALYSIS

FAILURES IN RANGE DIFFERENCE DATA MEAN			FAILURES IN RANGE DIFFERENCE. DATA STANDARD DEVIATION		
PROFILE	SENSOR	VALUE,FT	PROFILE	SENSOR	VALUE
GEM3	TMR	27.3	GEM2	TMR	35.3
H30SKAH	BEST	-41.5	GEM3	TMR	43.1
SAT3	BEST	30.2	SAT2	BEST	30.0
			SAT3	BEST	51.2

The purpose of this subsection is to describe the analysis of the range difference data statistics problems and provide some observations about the discontinuous jumps in the Ku-Band Radar range profile.

3.3.1 Discussion of Range Difference Data Statistics Problems

3.3.1.1 Description of Potential Error Sources

The potential sources of errors in the range difference data are the following:

- o GDOP
- o Low SNR (weak target return signal)
- o Target Range Acceleration

We first demonstrate that target range acceleration is not a consideration in the present analysis because the value of the acceleration would have to be quite large to produce a range bias that would cause the radar range estimate to fail its specification. Consider an example: a -10 feet/sec^2 acceleration would generate a range bias of 5.87 feet in the narrowest bandwidth case of the range tracker. (This example was taken from Reference 3. The closed-formed expression for the asymptotic range bias in the presence of acceleration is provided there.) Furthermore, the bias is smaller for wider bandwidths of the tracker. Thus, in the following discussion, target range acceleration will not be considered as a source of error. The discussion will be limited to GDOP and weak target return signals.

Geometric Dilution of Precision or GDOP is the name given to the error induced in a multiple sensor measuring system due to the placement of sensors and the random fluctuations of the individual sensor measurements. Appendix D gives a quantitative, rigorous derivation of the GDOP-induced error in the TMR measuring system. (We didn't have the resources to do a similar computation for the CINE system.) The results of the calculations provide the following qualitative observation. GDOP-induced range error is the worst at very low altitude and directly over the PEARL site brass cap (and the Ku-Band

Radar). For in this case, range to the target from the brass cap origin is along the -Z axis. But, since all three TMR radars are roughly in the X-Y plane and they only measure R and $\dot{R}$, then they cannot determine the target Z-component very well. Any small error in the R measurement translates to a large error in the Z-component of target position.

Although we did not have time to work out the expressions for the GDOP-induced range error in the CINE system, we can comment on the CINE performance in the situation described above using some newly gained insight. In the case of CINE system, each sensor measures the target's azimuth and elevation. In the scenario at hand, azimuth and elevation will provide information about the target's Z-component of position. Hence, small error in azimuth and elevation should not translate to large errors in the Z-component.

Weak target return signals which, in turn, produce low SNR at the doppler filter output (<10dB) will generate large random fluctuations in the range data. However, this is only a problem for weak targets (< 0 dBsm) at long range (>50,000 feet). A review of the range difference data and the corresponding range and RCS profile for all test runs, indicated that low SNR did not cause any of the failures listed in Table 3.3-1. Furthermore, it did not produce unusual problems in any of the other flight data examined. Figure 3.3-1 illustrates the high correlation between the target return signal strength (proportional to RCS) and the random fluctuations in the range difference data. The data shown in the figure is for flight HL146AE with an initial range of 46,500 feet and a final range 42,800 feet.

3.3.1.2 Discussion of Individual Problem Cases

Observe that all of the problem cases listed in Table 3.3-1 were out-of-specification (1) when compared to the BEST or TMR data only and (2) for flight trajectories where low attitudes and short ranges were involved. From the discussion of Section 3.3.1.1, these facts point to GDOP as the primary source or range error. There was one perplexing problem with assuming GDOP for all of these problem cases: why didn't all of the flight tests from

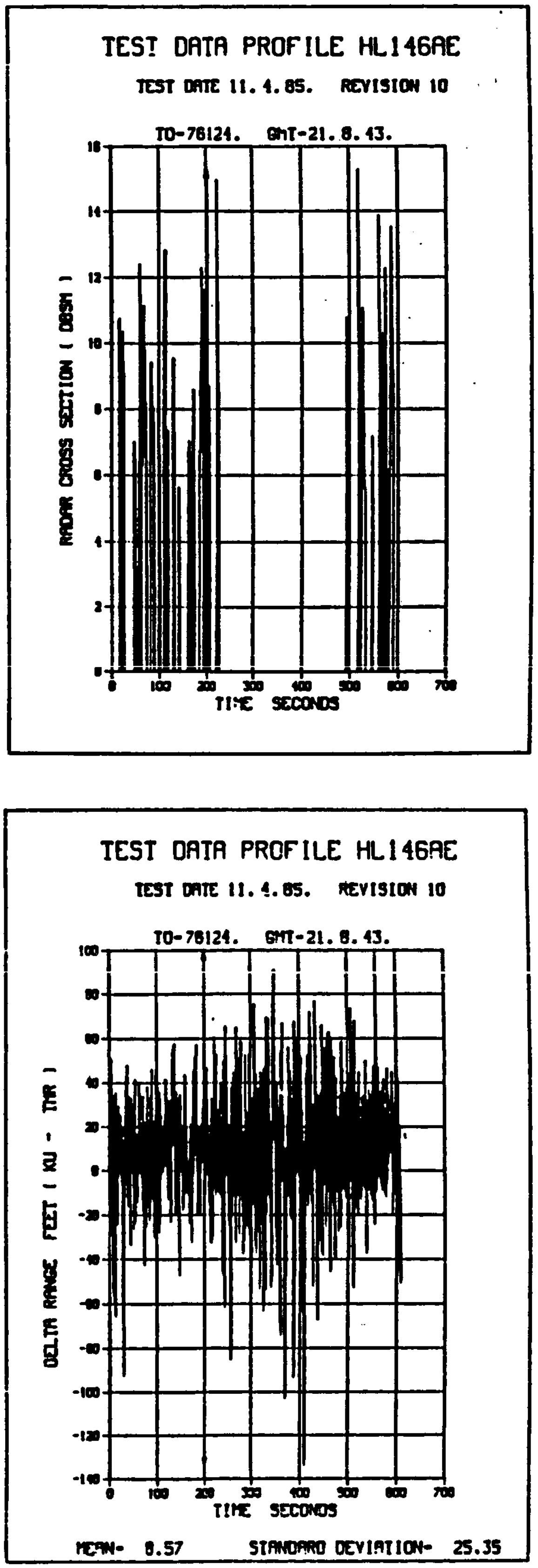


FIGURE 3.3-1 ILLUSTRATION OF CORRELATION BETWEEN TARGET RETURN SIGNAL STRENGTH AND RANGE TRACKER RANDOM ERROR

a given family, e.g., all H30SK's, suffer from the same problem? It turns out each general family has its own unique answer to this question. The answers for each family, GEM, BAL, SAT and H30SK, are provided below.

GEM and BAL Series. In this series, a helium filled GEMsphere was released from the brass cap and allowed to fly freely. Since all of these flights start at very low altitude over the brass cap, one would expect GDOP problems early in the flight for both GEM and BAL. However, a review of the flight log given in Appendix G shows that the only one radar (R-394) was available for the BAL series. Hence, there is no TMR or BEST solution available and consequently there is no problem with GDOP for the BAL tests.

Observe that the GEM3 failed both the mean and standard deviation specification while GEM2 failed only the standard deviation specification. Let's first examine the initial tracking altitude and range for both cases. For GEM2, the initial altitude and range are 2000' and 3000', respectively, and for GEM3 they are 1500' and 2000', respectively. At these altitudes, a delta of 500 feet makes a significant difference in the GDOP error. This difference can be seen in the BEST range difference profiles for GEM2 and GEM3 shown in Figure 3.3-2

It has been observed in other test series (H30SK) that GDOP-induced range error is sensititve to the X-Y ground track, especially at low altitude. This problem is not as significant in ths case. The predominant difference is the delta in initial altitudes.

There are some additional observations. First, to determine whether the range difference data mean and standard deviation were out-of-specification, they were both compared to 26.67 feet. This value is the limit for target ranges less than 8000 feet, while 1/3% of the range is used for ranges greater than 8000 feet. But, the target range interval was 3000 feet to 11,000 feet for GEM2 and 2000 feet to 26,000 feet for GEM3. Hence, a more correct determination of an out-of-specification condition would break the range difference data profile into intervals for ranges less than 8000 feet and greater than 8000 feet, compute means and standard deviations for each interval, and apply the correct specification to each interval.

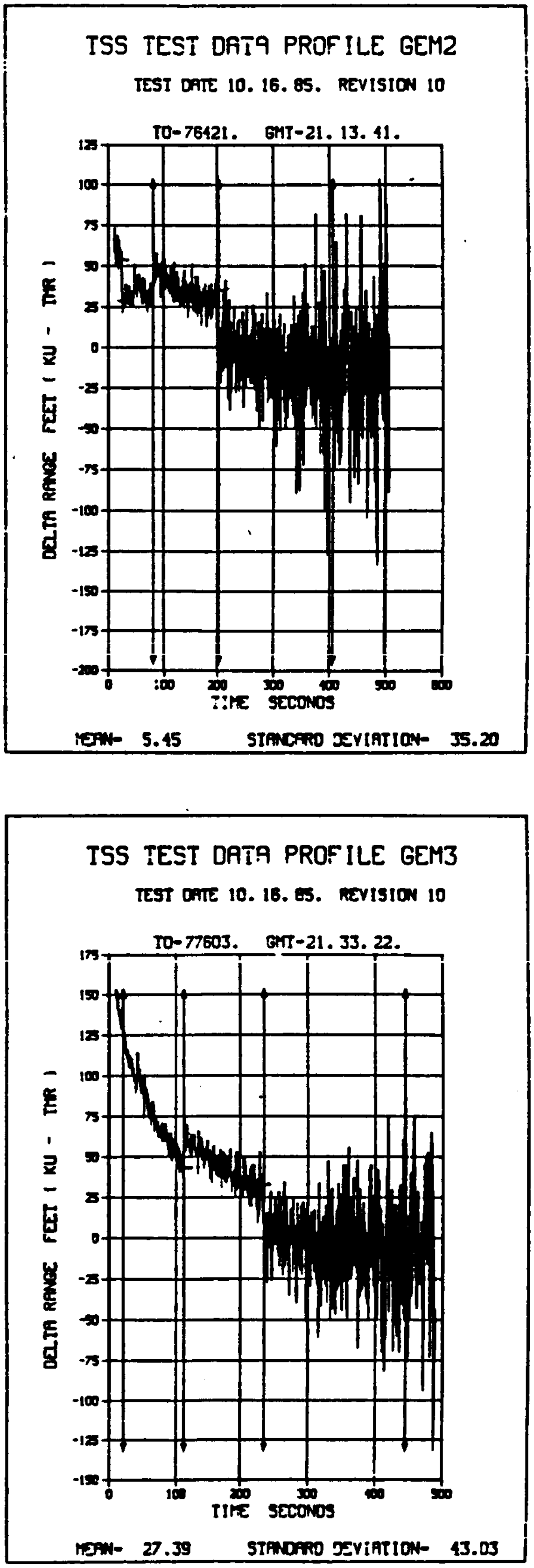


FIGURE 3.3-2 BEST RANGE DIFFERENCE DATA PROFILES FOR GEM2 AND GEM3

Secondly, notice that the random component in the range difference data of Figures 3.3-2 is increasing with time. This correlates with the fact the target is moving away from the radar and further illustrates the effect of decreasing target return signal strength.

Thirdly, the jumps in range bias seen at the pulsewidth switch points adds signficantly to the mean and standard deviation values.

SAT Series. The reason the SAT4 data was not a problem was because the altitude interval for the flight was 5100 feet to 68000 feet, and the range interval was 10,800 feet to 12,600 feet. As discussed previously, GDOP is not a problem at this altitude and range. Also, target return signal strength was not a factor at these ranges, even though the target RCS dropped to -10 dBsm at some points. Finally, since the balloon was tethered, range acceleration was not a consideration.

SAT2 and SAT3 were both suspectible to GDOP because their range of operation was less than 2600 feet. In fact, SAT3 started at 2600 feet range and finished at 1200 feet, while SAT2 remained fixed at approximately 2550 feet. The difference in range of these two cases would lead one to conclude that SAT3 would experience more severe GDOP effects than SAT2. That this conclusion is true is supported by the SAT2 and SAT3 BEST range difference profiles of Figure 3.3-3 and the problem summary of Table 3.3-1.

Discontinuous jumps of 60 feet were found in the SAT3 BEST range difference data at times 205 seconds and 280 seconds (see Figure 3.3-3). These jumps are not a problem with the Ku-Band Radar, but instead, are caused by the BEST range data as shown in Figure 3.3-4.

The SAT1 flight profile is very similar to the SAT2 but SAT1 range difference data statistics were within specification. A close examination of this data shows that GDOP has induced significant error in the SAT1 range difference data as shown in Figure 3.3-5. But why is the error less significant in this case? Analysis of the X-Y ground track and the altitutde data for both cases shows that, while the SAT1 flight is at a slightly lower altitude, the SAT2 flight is more nearly over the radar where

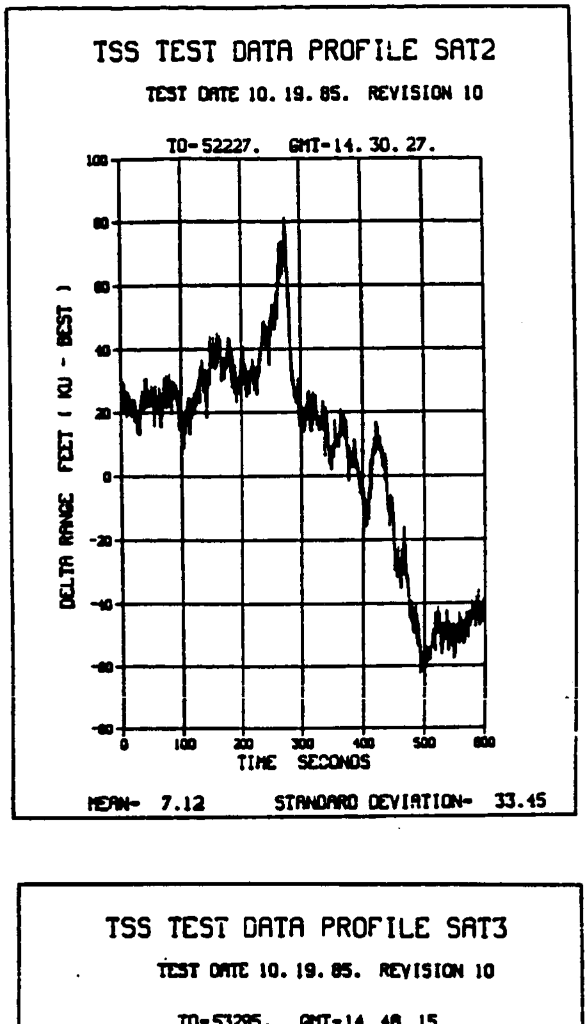


FIGURE 3.3-3 BEST RANGE DIFFERENCE DATA PROFILES FOR SAT2 and SAT3

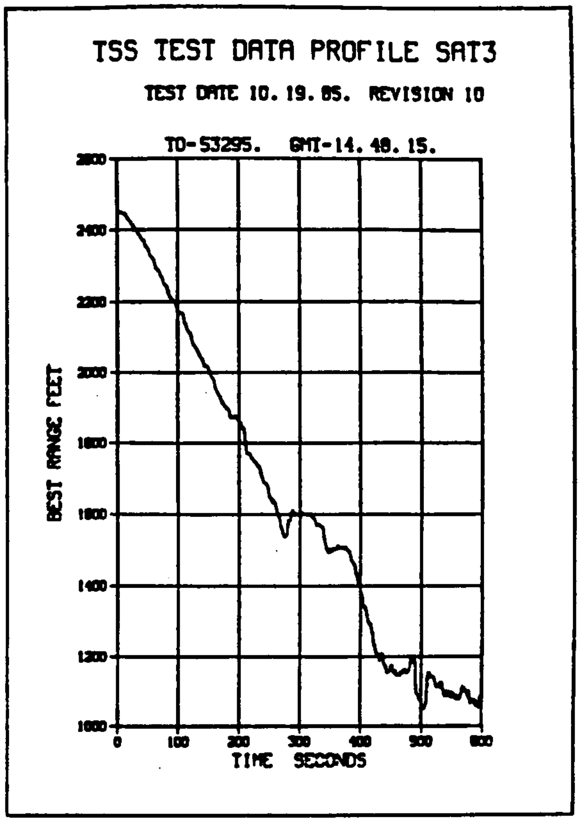


FIGURE 3.3-4 ILLUSTRATION OF JUMPS IN BEST RANGE DATA

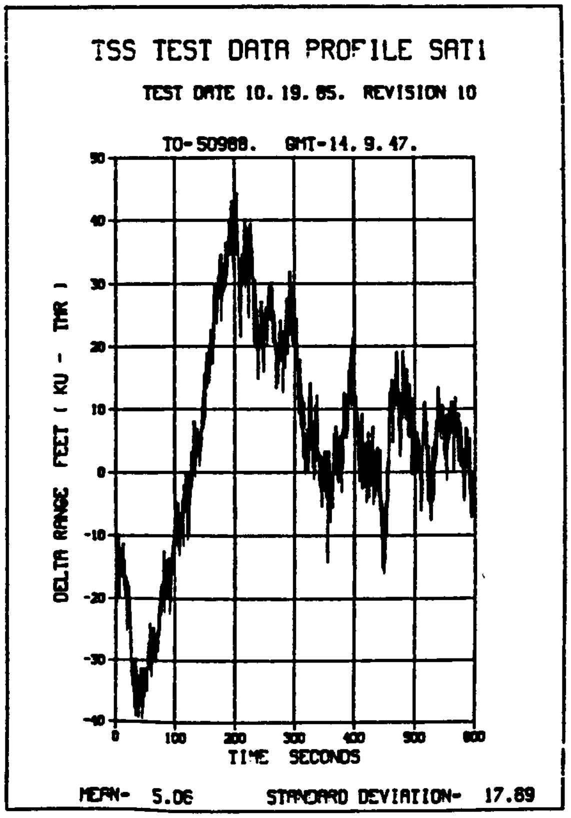


FIGURE 3.3-5 SAT1 RANGE DIFFERENCE DATA PROFILE

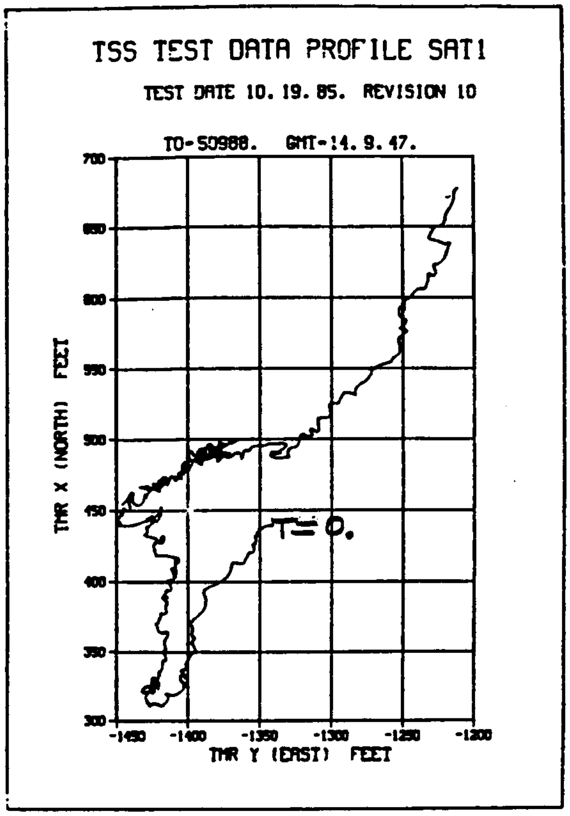
the GDOP problem is most severe. Figure 3.3-6 compares the X-Y ground track of the SAT2 and SAT1 flights. Unfortunately, at the writing of the report, no qualitative GDOP computations were available to confirm these conjectures.

H30SK Series. In this series of tests, a helicopter flew toward the radar with a starting range of 4000 feet and a finishing range of 2000 feet. The altitude was maintained between 1500 feet to 1700 feet. H30SKAH was the only test of this series that indicated a problem with the range difference data statistics. It is reasonable to assume that the source of the error is GDOP. But, since all of the H30SK profiles are quite similar, why isn't there a problem with all of these runs? A review of the range difference profiles shows that there is significant GDOP error in all of the test runs. Figure 3.3-7 compares the BEST range difference data profiles of H30SKAE and H30SKAH. Both profiles vary signficantly over the test duration with a trend toward negative range error. One major difference is that H30SKAH starts with a -20 foot offset, while H30SKAE starts with a +20 foot offset. The reason for this difference is not clear at the writing of this report.

While searching for a source of the difference in offsets described above, an interesting fact was uncovered. Figure 3.3-8 compares the BEST range difference profile and the Y-brass cap coordinate profile for H30SKAE. This comparison reveals a high correlation between these two parameters. It supports the contention that GDOP-induced range error is very sensitive to target position especially when the target is at low altitude and nearly overhead of the PEARL site brass cap. However, at this time, we have no closed-formed computation of GDOP-induced range error to support these conclusions.

3.3.2 Discussion of Discontinuous Jumps in Range

A review cf the range difference data has surfaccd some discontinuous jumps in range. These jumps are quite evident in the GEM and BAL series of data (see Figure 3.3-2). Examination of the corresponding range profile for these cases shows that the jumps occur at the Ku-Band Radar



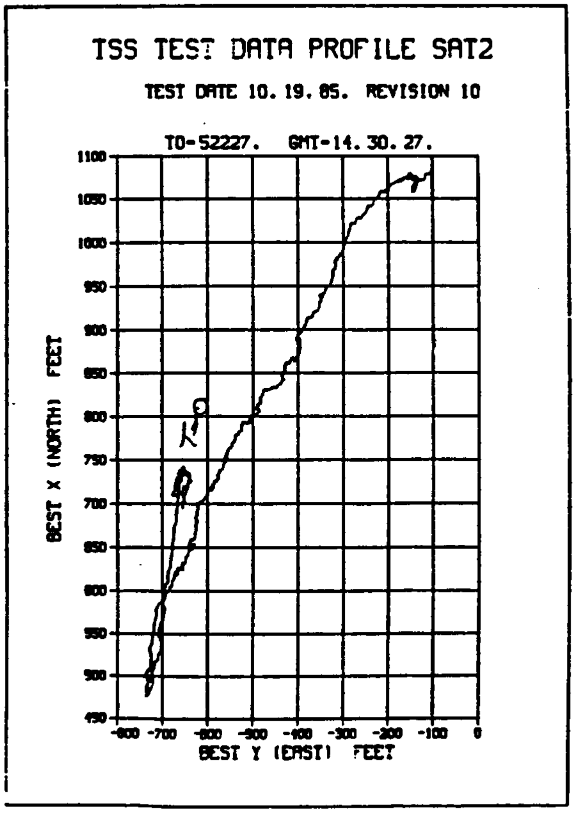


FIGURE 3.3-6 COMPARISON OF SAT1 and SAT2 X-Y GROUND TRACK

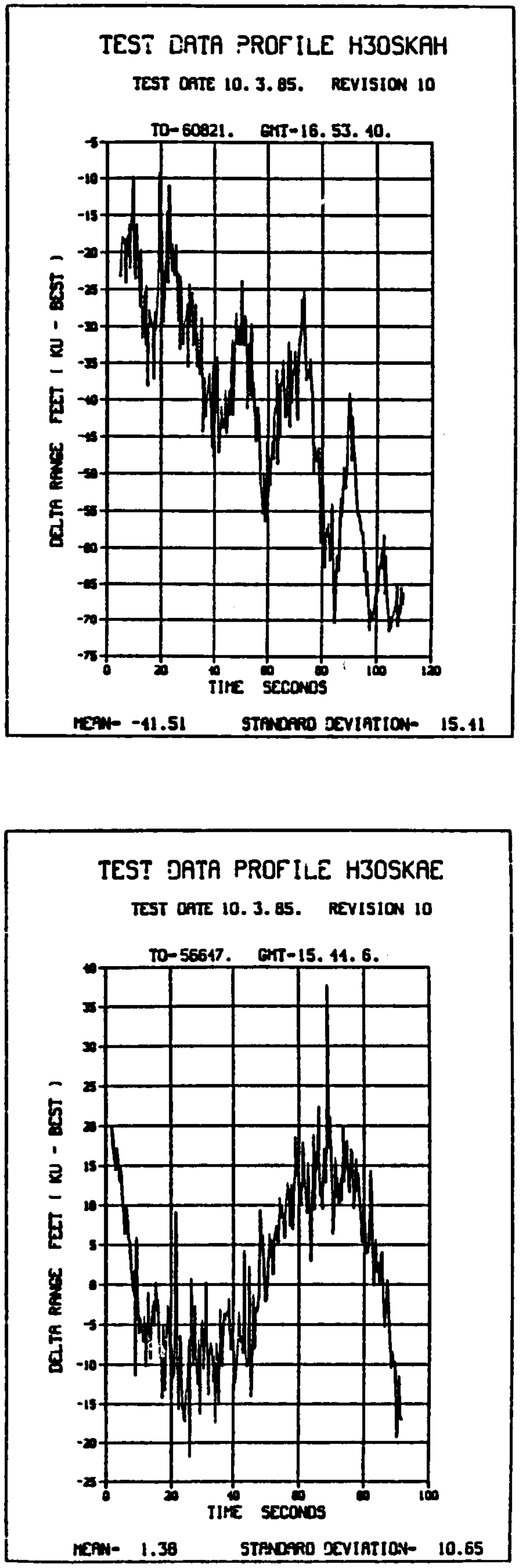


FIGURE 3.3-7 COMPARISON OF H30SKAH AND H30SKAE BEST RANGE DIFFERENCE DATA PROFILES

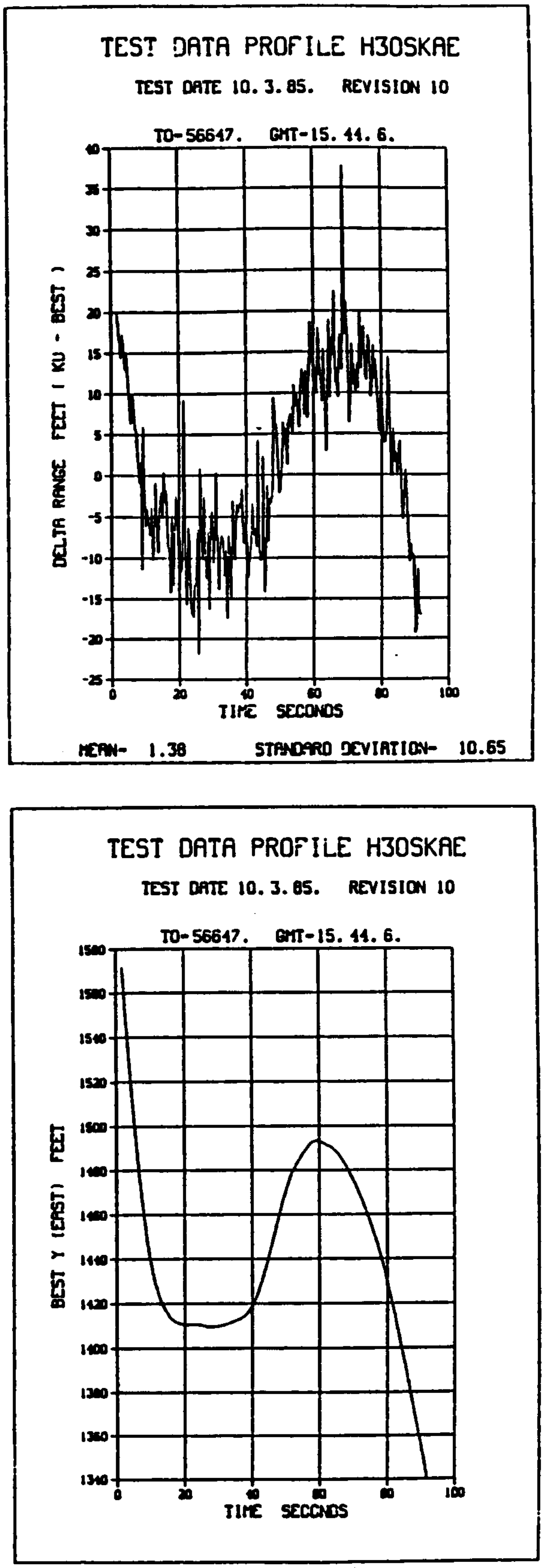


FIGURE 3.3-8 COMPARISON OF H30SKAE'S BEST RANGE DIFFERENCE PROFILE AND BEST-Y PROFILE

pulsewidth switch points. Some questions that come to mind immediately are as follows: Does a change in bias occur at each pulsewidth transition? Is the bias the same for a given pulsewidth transition or is it a random value? A comprehensive review of the range difference data was undertaken to answer these questions. The results of that data review are summarized in Table 3.3-2.

Some of the highlights of the review are as follows. First, and most important, there is some jump in bias at every pulsewidth transition. It is hard to discern a jump in the pulsewidth transitions at 23,030 feet and 49,920 feet because of random noise fluctuations due to a weak target return signal. Secondly, for a given pulsewidth transition, the value of the range jump was approximately the same. To confirm this statement, compare the 3200 foot and 5760 foot transition jumps for the GEM and BAL series. Thirdly, it was observed that the sign of the jump depended upon the direction of transition. This can be seen by comparing the 11520 transition point for the HEL30 series and the GEM series. A positive jump occurs in the HEL30 data where the target is closing and a negative jump is found in GEM data where the target is opening.

3.3.2.1 Discussion of Jump Mechanism

It is conjectured that these range jumps are cuased by slight changes in timing for generation of the different pulsewidth values. Observe that the largest jumps found were 30 feet. This corresponds to a timing change of 60 nanoseconds using the 2 nanosecond/foot conversion for two way range. Considering the complexity of the pulse generation and range gate timing circuitry, it is not surprising to find timing bias on the order of 40-60 nanoseconds.

To confirm these conjectures requires a detailed evaluation of the pulsewidth generation and range gate timing circuitry, a study of this magnitude is far beyond the bounds of the present project resources. Anyone wishing to pursue this subject futher should contact A.E. Miller, Jr., the

TABLE 3.3-2 SUMMARY OF RANGE JUMP INVESTIGATION (Page 1 of 4)

PROFILE	RANGE COVERAGE, FT	NUMBER OF SWITCH POINTS	JUMP RANGE, FT	JUMP MAGNITUDE, FT	COMMENTS
BAL1	800 to 10,500	2	3200	+5	
			5750	+20	
BAL2	800 to 5,500	1	3200	+5	
BAL5	8,000 to 10,300	0	-	-	
BAL6	900 to 10,500	2	3200	+5	
			5750	+20	
BAL7	1,000 to 10,700	2	3200	+5	
			5750	+10(?)	Needs Closer Examination
GEM2	3,000 to 30,000	4	3200	-25(?)	
			5750	+20	
			11520	+30	
			23030	?	Too Much Noise
GEM3	2,200 to 26,000	4	3200	+5	
			5750	+15	
			11520	+30	
			23030	?	Too Much Noise

TABLE 3.3-2 SUMMARY OF RANGE JUMP INVESTIGATION (Page 2 of 4)

PROFILE	RANGE COVERAGE, FT	NUMBER OF SWITCH POINTS	JUMP RANGE, FT	JUMP MAGNITUDE, FT	COMMENTS
SAT1	2500 to 2570	0	–	–	
SAT2	2520 to 2540	0	–	–	
SAT3	2450 to 1200	0	–	–	
SAT4	10,900 to 12,700 to 11,800	1	11,520	-25	Sign Error?
H30SKAE	3650 to 2525	1	2,560	?	Requires Closer Evaluation
H30SKAF	4100 to 2200	1	2,560	?	Requires Closer Evaluation
H30SKAG	2625 to 2175	1	2,560	-6	
H30SKAH	4000 to 2200	1	2,560	-6	
H30SKAI	3675 to 3325	0	–	–	

TABLE 3.3-2 SUMMARY OF RANGE JUMP INVESTIGATION (Page 3 of 4)

PROFILE	RANGE COVERAGE, FT	NUMBER OF SWITCH POINTS	JUMP RANGE, FT	JUMP MAGNITUDE, FT	COMMENTS
HEL30AF	12,200 to 7,000	1	11,510	+30	
HEL30AG	13,000 to 7,000	1	11,510	+30	
HEL30AI	13,300 to 5,500	2	11,510 5,750	+40 -25	
HEL30AJ	13,700 to 6,000	1	11,510	+36	
HJ146AC	62,500 to 44,800	0	-	-	
HJ146AD	64,000 to 46,500	0	-	-	
HL146AE	46,500 to 42,700	1	43,510	?	Too Much Noise
HL246AD	42,000 to 30,000	0	-	-	
HL246AE	41,500 to 47,200	0	-	-	
HL346AD	48,000 to 49,000	0	-	-	
HL346AE	47,100 to 49,000	0	-	-	
HL346AF	47,100 to 49,200	0	-	-	

TABLE 3.3-2 SUMMARY OF RANGE JUMP INVESTIGATION (Page 4 of 4)

PROFILE	RANGE COVERAGE, FT	NUMBER OF SWITCH POINTS	JUMP RANGE, FT	JUMP MAGNITUDE, FT	COMMENTS
HL446AC	48,700 to 47,500	0	-	-	
HL446AD	47,750 to 49,750 to 47,000	0	-	-	
HL446AE	48,800 to 49,300 to 46,800	0	-	-	
HL546AC	47,000 to 41,500	1	43,510	+40	
HL546AE	47,500 to 41,000	1	43,510	?	Too Much Noise
HL546AF	46,500 to 40,900	1	43,510	?	Too Much Noise
HL446AG	46,000 to 46,800 to 41,700	1	43,510	?	Too Much Noise

Responsible Engineer (RE) for the signal processing unit, or R.S. Austin, the System Engineer who is familiar with this area. Both gentlemen are with HAC's Radar Systems Groups.

3.4 RANGE RATE DATA ANALYSIS

A first pass through the SORTE data revealed a high percentage (95%) of failures in the standard deviation or random component of the range rate data. This was very surprising because all previous data, including system test data and flight rendezvous data, had indicated that the range rate tracking performance was better than predicted and well within the specification. An intensive examination of the data revealed several diverse sources of errors. These error sources included

- Range acceleration,
- Geometric Dilution of Precision (GDOP),
- Small RCS (low SNR),
- Target rotation,
- Time skewing,

and combinations of the above error sources. Errors that effected the majority of the data were GDOP and range acceleration. Target effects, including small RCS and target rotation, caused significant problems in only a handful of cases.

Problems, such as GDOP and time skewing, are associated with the WSMR sensor system and data processing. Therefore, they do not impact the Ku-Band Radar performance. On the other hand, range acceleration, target rotation, and small target RCS will be encountered in a space flight operational environment. Hence, range rate tracker performance data due to these effects is quite realistic.

Table 3.4-1 provides a case-by-case summary of the range rate analysis. This summary gives the standard deviation of the target acceleration, the range rate standard deviation for the Cine and Best data, a measure of the GDOP effects, and comments noting the most significant contributors for each test run. Notice that in some cases one error source dominates at the beginning of a flight and transitions to a second dominant source. Take GEM3 as an example. Once target rotation effects were removed, it was found that GDOP predominated in the first 200 seconds of the flight, while target acceleration effects predominated for the remainder of the flight. This case is examined in depth in Section 3.4.3.

3.4.1 Range Acceleration Effects

3.4.1.1 Analysis of Acceleration Effects on the Velocity Processor

Target range acceleration induces error in the Ku-Band Radar's velocity estimate. This error is generated in two places in the signal processing: (1) the discriminant formation process and (2) the smoothing filter at the velocity processor output. These two effects are analyzed below.

The velocity discriminant was designed under the assumption that the velocity was constant over the period (called a data cycle) during which the data is taken. Now, if the target is accelerating in range, the velocity will not be constant over the data cycle and the velocity discriminant will be distorted, causing an error in the velocity estimate. To determine the amount of distortion in this estimate, the signal processing prior to velocity discriminant formation must be examined.

The duration of a data cycle is 51.2 milleseconds for the 7 kHz PRF mode and 119 milleseconds for the 3 kHz PRF mode. In both cases, the radar processes a total of 320 return pulses through each of 2 range gates to form the velocity discriminant. The 320 pulses in each range gate are processed 16 consecutive pulses at a time to form the approximate doppler filter outputs via a discrete fourier transform (DFT). Since there are 640 return pulses for the two range gates, then there are 40 outputs formed for each doppler filter. For a given filter, the magnitude of these 40 outputs

TABLE 3.4-1 SUMMARY OF RANGE ACCELERATION EFFECTS ON RANGE RATES (Page 1 of 4)

PROFILE	ACCELERATION STD DEV	RANGE RATE STD DEV		AVERAGE PREDICTED GDOP STD DEV	COMMENTS
		BEST	CINE		
BAL1	1.27	1.38		1.14	GDOP is present at beginning of profile. Acceleration is a component of the error. Primary source is phase skewing.
* BAL2	3.58	3.08		2.95	GDOP effect in first 200 seconds of profile is significant. Removal of phase skewing from KU and WSMR data leaves error which is approximately the error from acceleration.
BAL5	3.2	1.2		.49	Phase skewing is primary source of error. Some error also due to acceleration.
* BAL6	3.82	1.38		1.07	GDOP effect in first 100 seconds of profile significant. Phase skewing between WSMR and KUBAND data. Acceleration also significant factor.
BAL7	2.3	2.9		1.03	Phase skewing is major error. GDOP is present at beginning of profile.
GEM2	6.5	2.77		.355	One spike in range rate causes excessive standard deviation. Some error is due to acceleration but majority is due to phase skewing between KU and WSMR data.
GEM3	4.83	1.83			Acceleration effect is significant. Phase skewing is also large part of error.
SAT1	6.03	2.33			GDOP intensified by oscillating acceleration.
SAT2	9.56	3.06	1.41	1.76	GDOP compounds acceleration effect in BEST data. Acceleration data is approximately correct.
* SAT3	13.14	6.78	1.85	2.97	Invalid BEST data due to large GDOP effect.
SAT4	2.22	.73	.65	.26	Combination of Acceleration, and possibly some skewing in time.

TABLE 3.4-1 SUMMARY OF RANGE ACCELERATION EFFECTS ON RANGE RATES (Page 2 of 4)

PROFILE	ACCELERATION STD DEV	RANGE RATE STD DEV		AVERAGE PREDICTED GDOP STD DEV	COMMENTS
		BEST	CINE		
H30SKAE	2.16	1.47	.32	1.43	GDOP effect evident in BEST solution. Good comparison between cine and KU. BEST acceleration data is invalid due to GDOP.
** H30SKAF	.80	1.72	.33	1.37	BEST data distorted by GDOP effects. Cine data is very much in agreement with KU data. Acceleration data is invalid due to GDOP.
** H30SKAG	1.22	.84	.83	1.9	GDOP effect invalidates BEST data. Cine and KU diverge in last 5 sec.
H30SKAH	1.309	2.21	1.14	1.4	BEST data distorted by GDOP. All data sets dissimilar. Acceleration affect buried by other problems.
H30SKAI	2.61	1.06	.49	1.17	GDOP distorts BEST data so it is not an accurate reference. Cine and KU data correspond well except for one spike which causes large standard deviation.
HEL30AF	1.71	.75	.50	.35	Trends in data highly correlated to acceleration. Cine difference data is in better agrrement with expected error. GDOP could explain why cine data is better than BEST data.
HEL30AG	1.01	.371	.41	.32	Data overall is good. Small acceleration effect.
HEL30AI	.97	.67	.36	.38	GDOP contributes to BEST data error. Cine data more plausible.
HEL30AJ	1.2	.76	.38	.38	Cine data more plausible. GDOP contributes to BEST data error.
* HJ146AC	.62	.32		.1	Due to clouds Cine data no good. Acceleration plot is not valid. Examination of range rate plot showed no quick changes in velocity.
* HJ146AD	.87	.36	.60	.1	Acceleration data is no good. Estimated range acceleration standard deviation is used instead. Cine data could be degraded because of range.
HJ146AE	.65	.36	.52	.1	Practically within spec. Some spikes in cine data which make it suspect.

TABLE 3.4-1 SUMMARY OF RANGE ACCELERATION EFFECTS ON RANGE RATES (Page 3 of 4)

PROFILE	ACCELERATION STD DEV	RANGE RATE STD DEV		AVERAGE PREDICTED GDOP STD DEV	COMMENTS
		BEST	CINE		
HL146AE	.576	.44	.94	.1	Low SNR is believed to cause error. No acceleration effect.
* HL246AD	.63	.49	.70	.1	Acceleration data is no good. Believe SNR is problem. Cine data affected by clouds.
* HL246AE	.63	.71		.1	Believe large portion of error is due to low SNR.
* HL346AD	.54	.66	.60	.1	Low SNR is believed to be major cause of error. Acceleration data was bad.
* HL346AE	.86	.51	.69	.1	Acceleration data is invalid. Error is due to low SNR.
HL346AF	.366	.55	.82	.1	No accleration problem. SNR is the major problem.
* HL446AC	.55	.42	.55	.1	Acceleration data is invalid. Cine is inhibited by clouds. Estimated range accleration shows acceleration data is invalid. Believe Low SNR is the main problem.
* HL446AD	.66	.54		.1	Believe significant portion of error is due to low SNR. Acceleration is small over most of profile while velocity error is large over all of the profile.
HL446AE	.4	.51	1.25	.1	No acceleration effect. Spikes in cine data make it suspect. Errors could be due to low SNR.
* HL546AC	.57	1.3		.1	Range rate output for KU and TMR diverge after PRF change. There are 3 to 4 foot/sec errors. Unknown cause.
* HL546AE	1.5	.67	.75	.1	Examination of range rate plots demonstrate a correlation between acceleration and range rate error. Acceleration is main error source here.
HL546AF	1.28	.54		.1	Low SNR could cause noisy range rate.Acceleration effect is small.
HL546AG	.44	.46	.67	.1	Low SNR could cause problem.

TABLE 3.4-1 SUMMARY OF RANGE ACCELERATION EFFECTS ON RANGE RATES (Page 4 of 4)

Range rate statistics are in ft/sec. Acceleration statistics are in ft/sec/sec. GDOP is a function for the geometry, therefore the standard deviation of the error is changing over the profile. The average predicted GDOP standard deviation is obtained by calculating the standard deviation of the range rate error for WSMR at each time interval and averaging this over the whole profile. This is also expressed in ft/sec.

1. BEST Acceleration data was used to calculate standard deviation of acceleration data unless otherwise indicated. Approximations of acceleration were used when acceleration data and range rate data were uncorrelated.

* indicates acceleration was estimated from a BEST (delta range rate)/(delta time) calculation.

** indicates acceleration was estimated from a CINE (delta range rate)/(delta time) calculation.

are computed and summed together (a process called post detection integration or PDI) to form an integrated filter output. The velocity discriminant is then formed by comparing the values of the filter on each side of the current velocity tracking filter. This gives a measure of the position of the target velocity within the center tracking filter.

One concern is the effect of acceleration on each formation of a 16 point DFT. Consider a range acceleration of 10 feet/sec^2, the change in velocity over the 16 point DFT is 0.023 feet/second in the 7 kHz case and 0.053 feet/second in the 3 kHz case. In both cases, this turns out to be 0.3% of a filter width. This produces insignificant degradation in individual 16 point DFT outputs. The second problem in the velocity discriminant formation caused by acceleration is the change in the filter output value over the 20 filter output formations for a given range gate. Again, assuming a range acceleration of 10 feet/second^2, the velocity changes 0.46 feet/second over the 20 filter formations in the 7 kHz PRF mode and 1.075 feet/second in the 3 kHz PRF mode. However, due to the PDI process the total change predicted by the radar velocity discriminant is just 1/2 of this value. The PDI process can be viewed as an averaging process and the error can be obtained from the following equations:

$$V = \frac{1}{20} \sum_{n=1}^{20} (Vo + n\,\Delta v) \qquad (3\text{-}1)$$

or $$V = Vo + \frac{\Delta V}{20} \sum_{n=1}^{20} n$$

or $$V = Vo + 10\ \Delta V\frac{21}{20} = Vo + 10\,\Delta V$$

where V = Radar velocity estimate at the end of a data cycle,

Vo = actual target velocity at beginning of a data cycle.

ΔV = true change in target velocity over a 16 point DFT formation.

Now, the actual velocity at the end of a data cycle is given by Vo + 20ΔV and the error is therefore 100V. Thus, for the example of a 10 feet/sec^2 range acceleration, the velocity error due to the PDI process would be 0.23 feet/second in the 7 kHz PRF mode and 0.54 feet/second in the 3 kHz PRF mode. This is a significant velocity error source. A complete, exact detailed analysis of the velocity error due to the velocity discriminant formation is given in Appendix F.

The second source of range acceleration error occurs in the moving window averager at the output of the velocity processor (see Figure 3.4-1). In the 7 kHz PRF mode the moving window filter averages the present data cycle velocity estimates with the previous 3 data cycle estimates. In the 3 kHz PRF mode the filter averages the present data cycle estimate with 1 previous data cycle estimate. For a given range acceleration value, this filtering produces the same error effect as the PDI processor. The estimated velocity in this case can be expressed as

(3-2) $$V = \frac{1}{N} \sum_{n=0}^{N-1} (Vo + n\Delta V_D)$$

or $$V = Vo + \Delta V_D \sum_{n=o}^{N-1} \frac{n}{N}$$

or $$V = Vo + \Delta V_D \frac{N-1}{2} \qquad \text{(Radar estimate)}$$

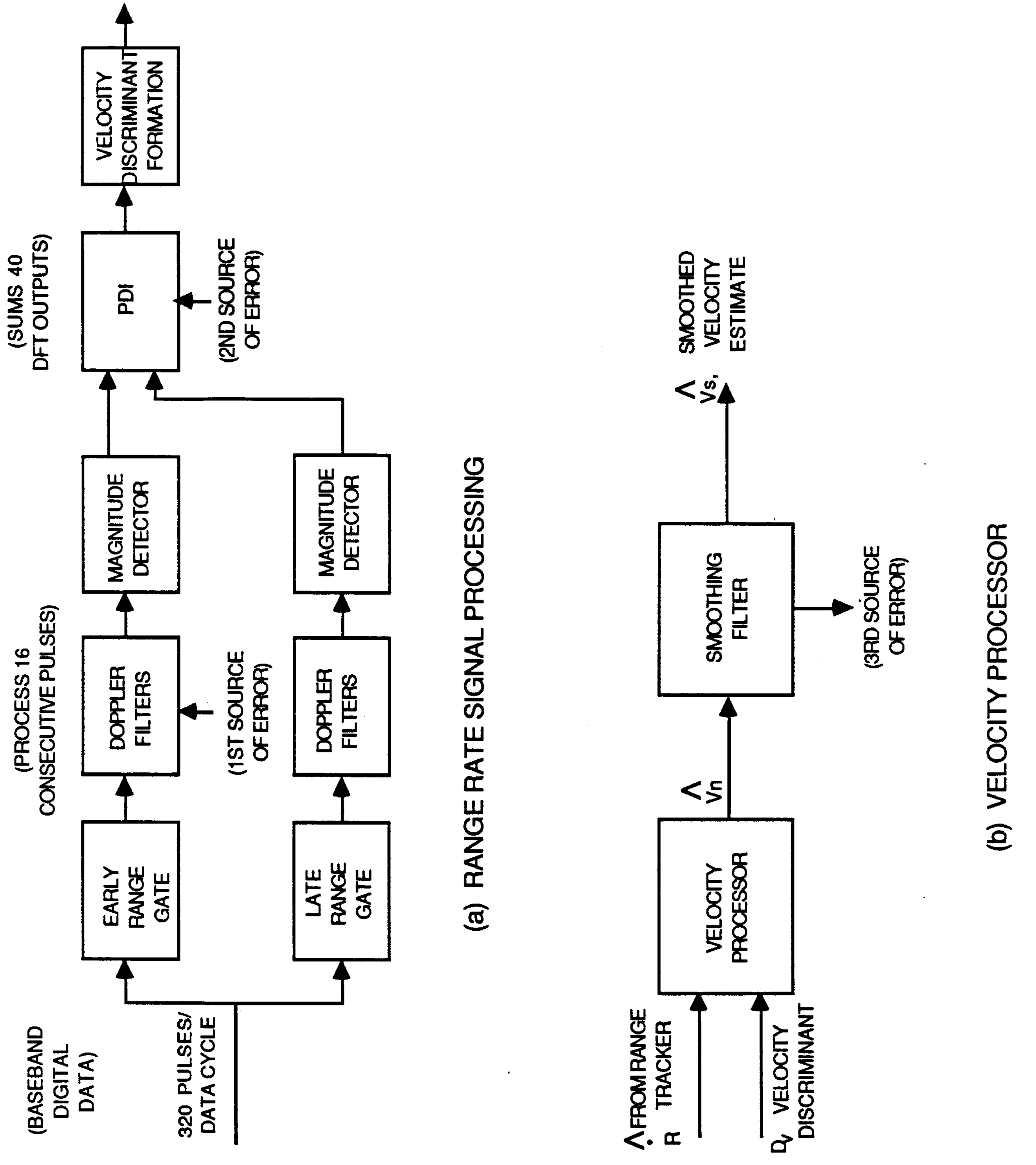


FIGURE 3.4-1 ILLUSTRATION OF RANGE ACCELERATION ERROR SOURCES IN THE RANGE RATE SIGNAL PROCESSING

and the actual velocity is given by

(3-3) $$V = Vo + (N-1) \Delta V_D$$

where Vo = true velocity at the beginning of the averaging period,

ΔV_D = change in true velocity over one data cycle,

N = moving window filter width.

Clearly, the error induced by the moving window filter in the presence of range acceleration is (N-1) $\Delta V_D/2$. Using a range acceleration of 10 feet/sec^2, ΔV_D is 0.512 feet/sec and the induced error is 0.768 feet/second in the 7 kHz PRF mode. In the 3 kHz PRF mode, ΔV_D is 1.19 feet/second and the induced error is 0.595 feet/second.

Combining the errors caused by the PDI processor and the moving window filter one obtains the—following expressing for the radar velocity estimate,

(3-4) $$V = \frac{1}{N} \sum_{n=o}^{N-1} (Vo - 10 \ \Delta V + n \Delta V_D)$$

or $$V = Vo - 10 \ \Delta V + \frac{N-1}{2} \ \Delta V_D$$

or $$V = Vo - \frac{\Delta V_D}{2} + \frac{N-1}{2} \ \Delta V_D$$

Subtracting equation 3-4 from 3-3 the velocity error estimate is

$$\text{(3-5)} \qquad \text{TOTAL VELOCITY ERROR} = \frac{N}{2} \; \Delta V_D = \frac{N}{2} \; T_D \; A_R$$

where T_D = Data Cycle Length
A_R = Range Acceleration

for the 7 kHz PRF mode and a 10 feet/sec^2 range acceleration, the error 1.02 feet/sec and for the 3 khZ PRF mode and the same acceleration, the error is 1.19 feet/sec. This is a significant error in either PRF case.

In summary, equation 3-5 can be used as a tool to estimate the Ku-Band Radar velocity error in the presence of target range acceleration. This result was applied to the SORTE generated range rate difference data to determine when acceleration was a significant error source. Results of this exercise are discussed below.

3.4.1.2 Range Acceleration Effects in the SORTE Data

A crude measure used to determine those test cases that might be affected by range acceleration error was to compute the standard deviation of the Best range acceleration data. Then analyze those cases with acceleration standard deviations that were greater than 1 foot/sec^2. Table 3.4-2 summarizes the results of this exercise. It gives the range acceleration standard deviation and the range rate difference standard deviation referenced to the Best data and the Cine data when available.

All of the SAT tests, except SAT4, appear to have the highest range acceleration standard deviation and correspondingly high delta range rate standard deviations. Since the target was a tethered GEM sphere that was reeled in and out very slowly, it is clear that, in fact, there was very little range acceleration. Further analysis revealed that GDOP contributed significant random error to the TMR (and Best) range rate data, producing a highly corrupted Best range acceleration data as well. GDOP was a significant factor due to the target's position (low altitude, directly over the brass

TABLE 3.4-2 TEST CASES WHERE RANGE ACCELERATION WAS AN APPARENT PROBLEM

PROFILE	BEST RANGE ACCELERATION STANDARD DEVIATION	DELTA RANGE RATE STANDARD DEVIATION	
		BEST	CINE
BAL1	1.27	1.38	ND
BAL2	3.58	3.08	ND
BAL5	3.20	1.20	ND
BAL6	3.82	1.38	ND
BAL7	2.30	2.90	ND
GEM2	6.50	2.77	ND
GEM3	4.83	1.83	ND
SAT1	6.03	2.33	ND
SAT2	9.56	3.06	1.41
SAT3	13.14	6.78	1.85
SAT4	2.22	0.73	0.65
HEL30AF	1.71	0.75	0.50
HEL30AG	1.01	0.37	0.41
HL546AE	1.50	0.67	0.75

cap) relative to the 3 TMR radars. In these cases, the conclusion is that the TMR system is the principal contributing error source and that there is no problem with the Ku-Band Radar estimate.

In the SAT4 case, the target is still a tethered gem sphere, but at a much higher altitude nearly over the brass cap. Although not as severe as the first three SAT cases, GDOP again produces a significant apparent acceleration. Hence, GDOP is the primary contributor to the delta range rate behavior in this case. The effects of GDOP on the SAT cases is discussed in more detail in Section 3.4.2.

The group of test runs with the next highest apparent range acceleration standard deviations were the GEM and BAL tests. All of these tests consisted of releasing helium-filled GEM spheres at the brass cap and allowing them to fly freely. In this case, three factors contributed to the range acceleration standard deviation: (1) GDOP, especially at low altitude, (2) the spinning GEM sphere and (3) true target range acceleration. GDOP effects are discussed in Section 3.4.2 and target rotation effects are discussed in Section 3.4.3.

Let's examine one of these cases in detail. Figures 3.4-2 and 3.4-3 show the BAL2 range rate difference data prior to and after compensating for target rotation effects, respectively. (Justification for the compensation is given in Section 3.4.3). This new data shows a significant reduction in the standard deviation. Also it will be shown in Section 3.4.2 that the major contributor in the first 125 seconds is GDOP. Now, let's analyze the remaining difference data (from 125 seconds to 300 seconds). The standard deviation of this data is approximately 0.67 feet/second, which is still beyond the specification limits. A significant contributor to this error is the target rotation effects. It turns out that the radar is tracking the spinning of the target as evidenced by the expanded plot of the Ku-Band Radar MDM range rate shown in Figure 3.4-4 during this period. The range rate is oscillatory in nature with peak-to-peak swings of 10 feet/second and a period of approximately 4 seconds. Close examination of Figure 3.4-4 reveals accelerations as high as 8 feet/sec^2. From the analysis of Section 3.4.1.1, this translates to a Ku-Band Radar range rate error of 0.82 feet/sec^2. This acceleration effects due to target rotation then becomes a significant contributor to the range rate error. In addition, there is some minor effect due to the target moving (accelerating) away from the Ku-Band Radar.

The range acceleration effects analysis for the remaining BAL tests and GEM tests follows in an identical manner to the BAL2 analysis provided above. The conclusions of those analyses are also identical.

The next group of tests that showed effects potentially caused by range acceleration was the HEL30 and the HL546 series. Consider the HEL30AF case. The range rate difference data of Figure 3.4-5 shows some definite trends rather than being purely random in nature. A comparison of the BEST range acceleration profile of Figure 3.4-6 indicates that the trends in the range rate difference profile are highly correlated with the range acceleration profile. Using the acceleration data of Figure 3.4-6 and equation 3.5, the expected Ku-Band Radar range rate error was computed and

ORIGINAL PAGE IS
OF POOR QUALITY

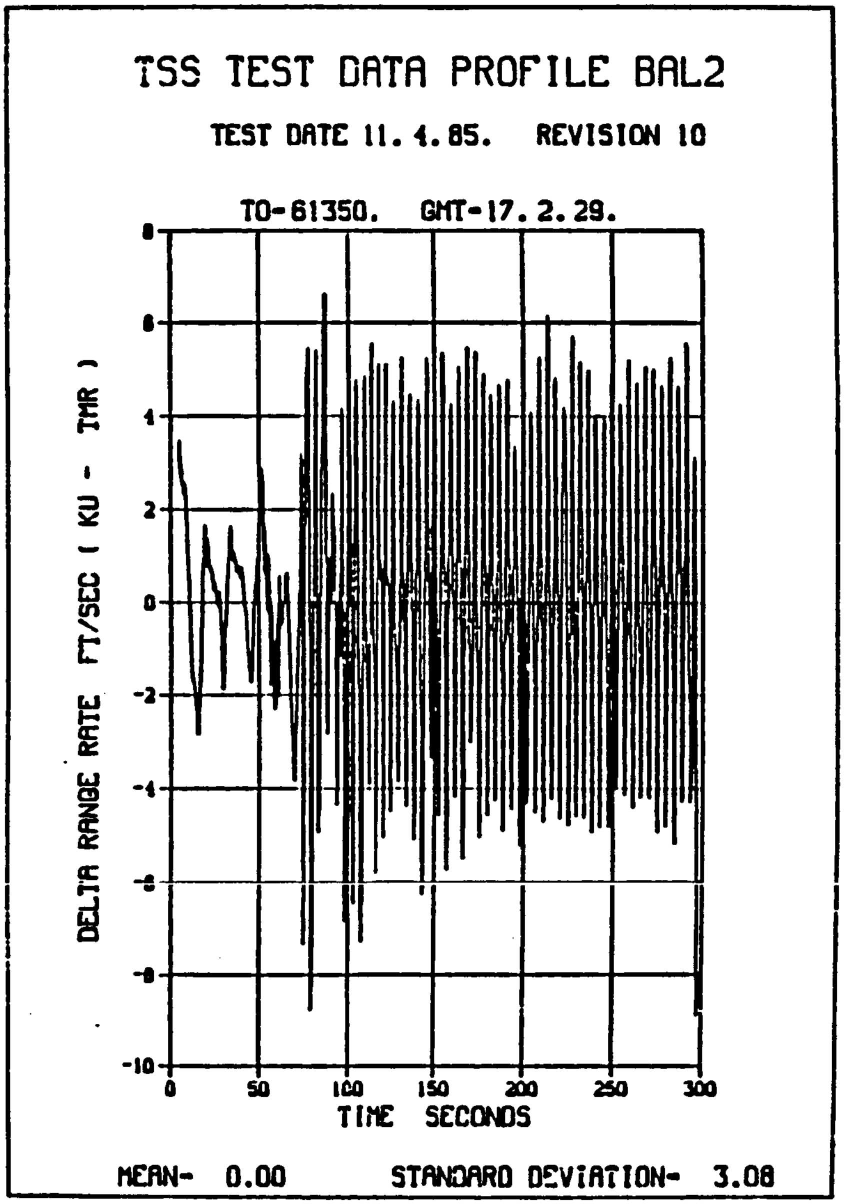


FIGURE 3.4-2 ILLUSTRATION OF OSCILLATION IN RANGE RATE DATA DUE TO TARGET ROTATION

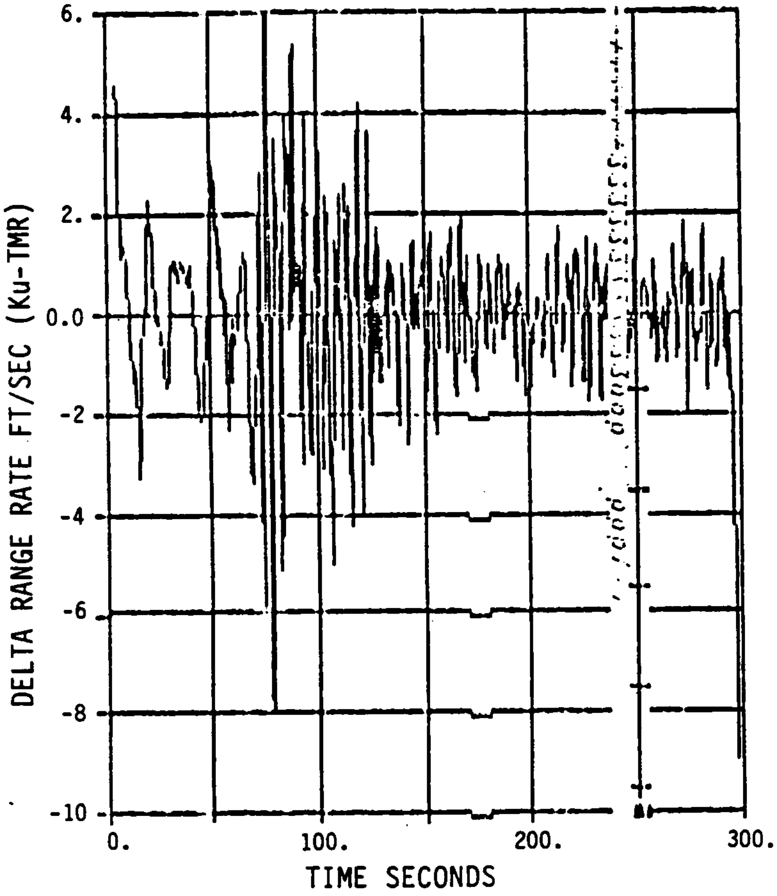


FIGURE 3.4-3 BAL2 TMR RANGE RATE DIFFERENCE DATA AFTER COMPENSATING FOR TARGET ROTATION EFFECTS. NOTE: KUBAND DATA IS SHIFTED 1.6 DATA CYCLES RELATIVE TO THE TMR DATA.

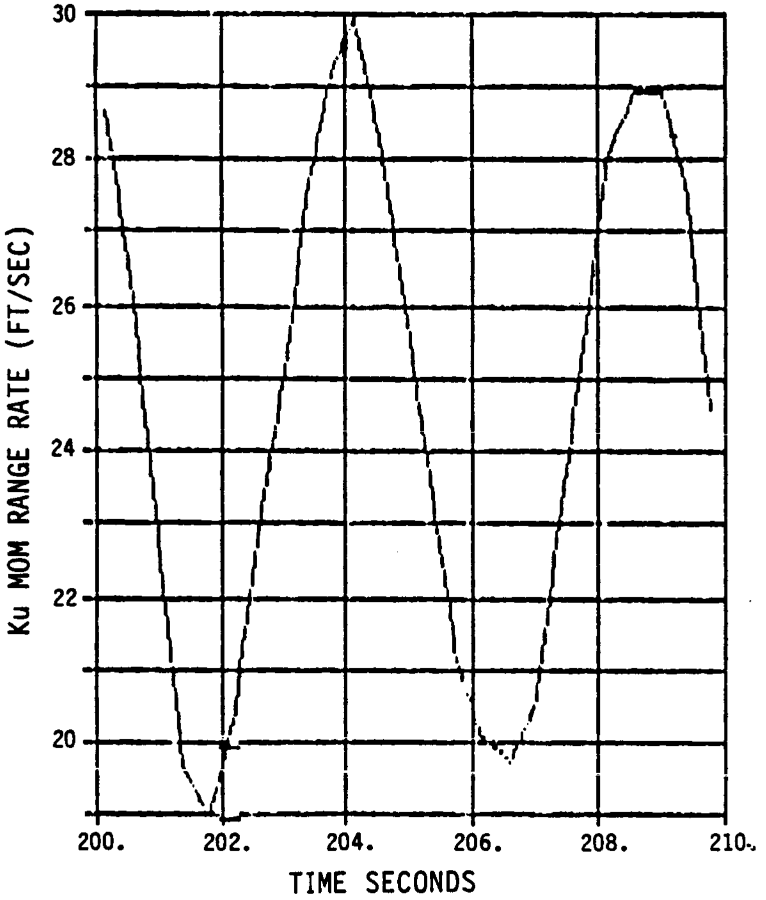
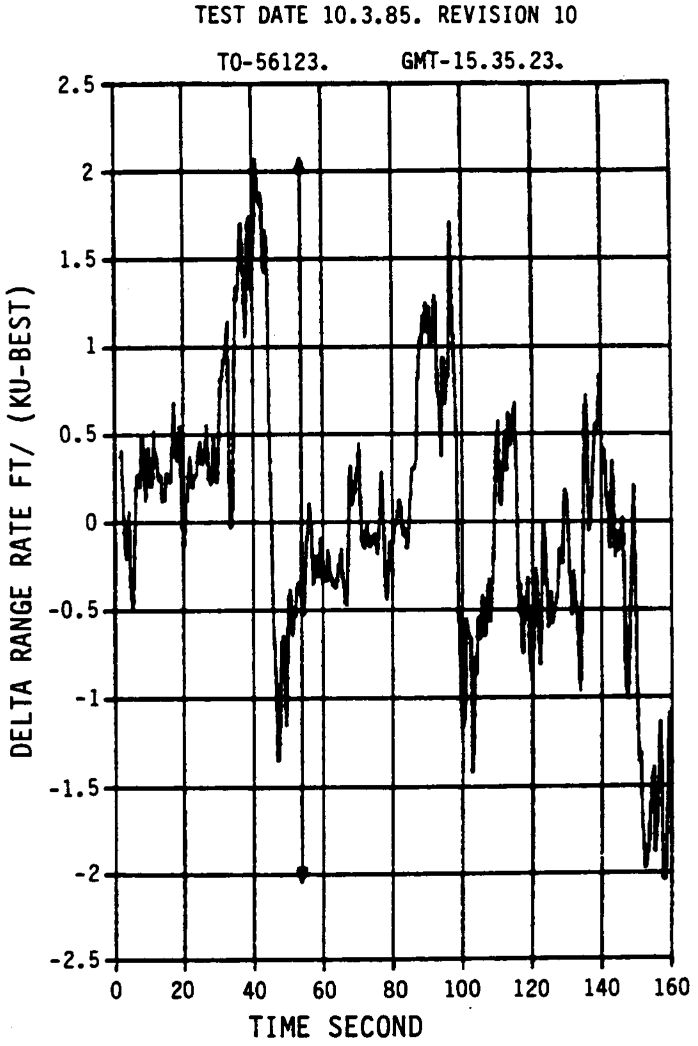


FIGURE 3.4-4 EXPANDED VIEW OF THE OSCILLATION INDUCED IN THE KUBAND RADAR RATE DUE TO TARGET ROTATION



MEAN- 0.01 STANDARD DEVIATION- 0.75

FIGURE 3.4-5 HEL30AF BEST RANGE RATE DIFFERENCE PROFILE TO BE COMPARED WITH RANGE ACCELERATION PROFILE OF FIGURE 3.4-6

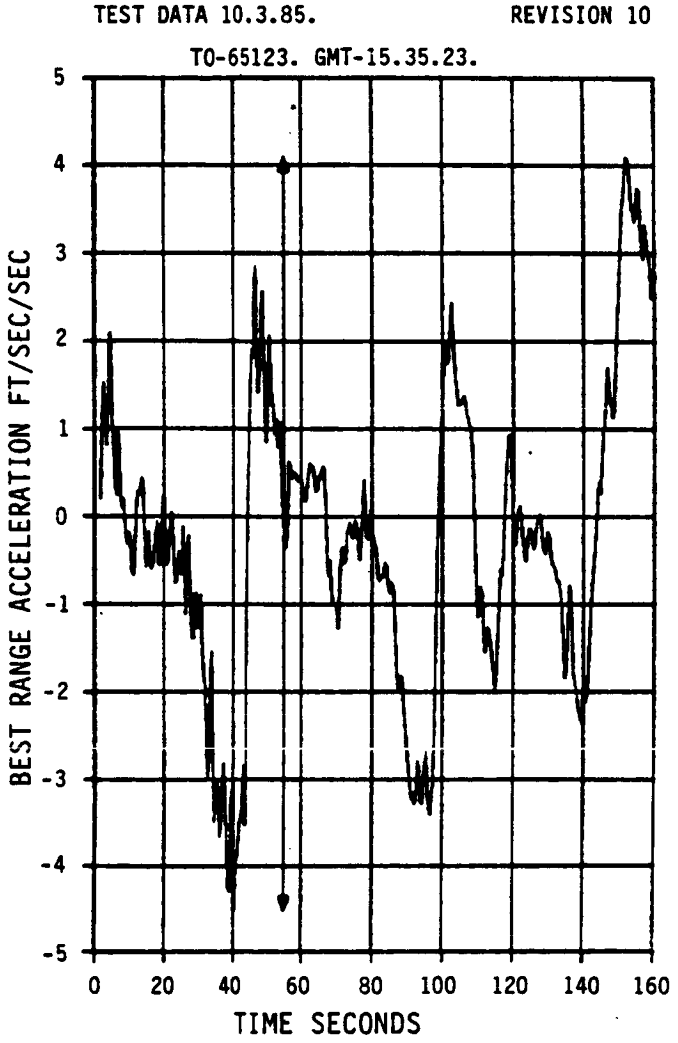


FIGURE 3.4-6 HEL30AF BEST RANGE ACCELERATION PROFILE TO BE COMPARED WITH THE RANGE RATE DIFFERENCE PROFILE OF FIGURE 3.4-5

found to be too high by a factor of about 4. The range rate difference data referenced to the CINES (Figure 3.4-7) shows the scale factor to be reduced to 3, but this is still a significant discrepancy. To further probe this problem, the BEST profile for HEL30AF was used to drive the simulator. The simulation generated range rate was differenced with the BEST range rate data to produce the profile shown in Figure 3.4-8. This data gives the expected theoretical result. At the writing of this report, the source of the discrepancy in the data has not been resolved.

Analysis of the HEL30AG and HL546AE profiles gave similar results. Both profiles show high correlation between the BEST range rate difference data and the BEST range acceleration data. Also a scale factor error was found to be present in both cases. However the scale factor appeared to be closer to 2 rather than 3 or 4 as in the HEL30AF case.

3.4.2 GDOP Effects

3.4.2.1 A Qualitative Description of GDOP

Geometric Dilution of Precision (GDOP) is the name applied to the inaccuracies induced in a set of target measurements caused by the placement of the system sensors relative to the target and the random errors in the individual sensor measurements. A complete development of the theory of the TMR GDOP effects on range and range rate measurements is given in Appendix D.

One of the most significant facts that surfaced during the GDOP development can be described as follows. First, notice that the three TMR radars and the Ku-Band Radar lie approximately in a plane and the TMR radars surround the Ku-Band Radar (see Figure D-1). Also, observe that the TMR radars only supply target range and range rate measurements. Now, if the target is at low altitude over the brass cap or directly over the Ku-Band Radar, the TMR radars cannot measure the vertical component (or the Z-component) of the target velocity. Furthermore, any random errors in TMR measurements will translate into significant errors in the Z-component of velocity estimated by TMR. In this configuration, the Z-component translates to range rate as measured by the Ku-Band Radar. Hence, there is significant

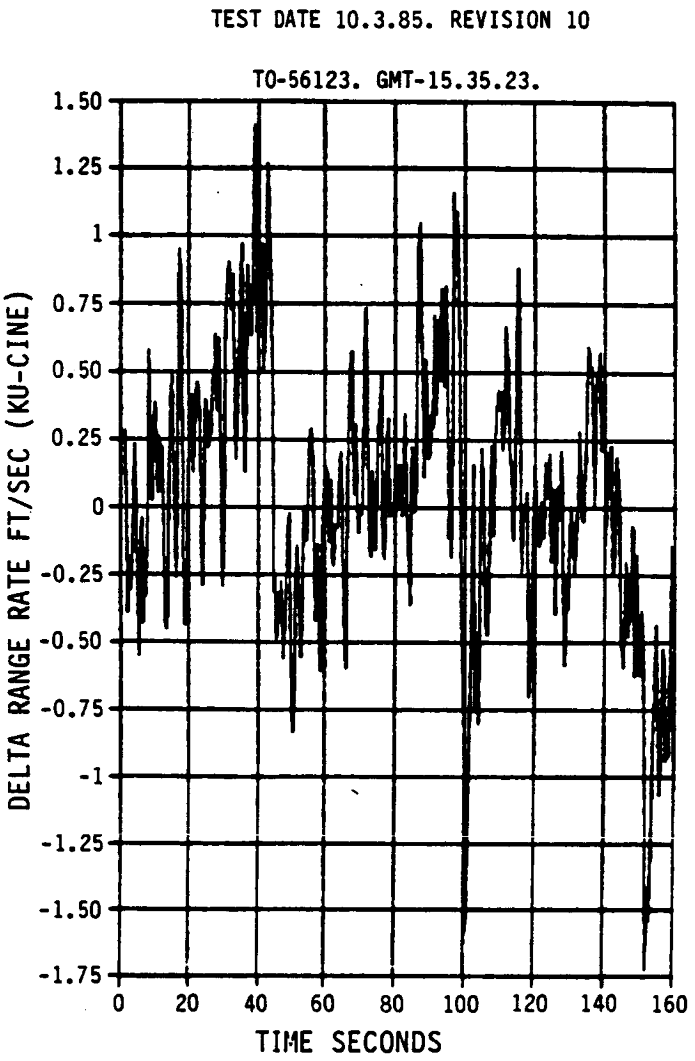
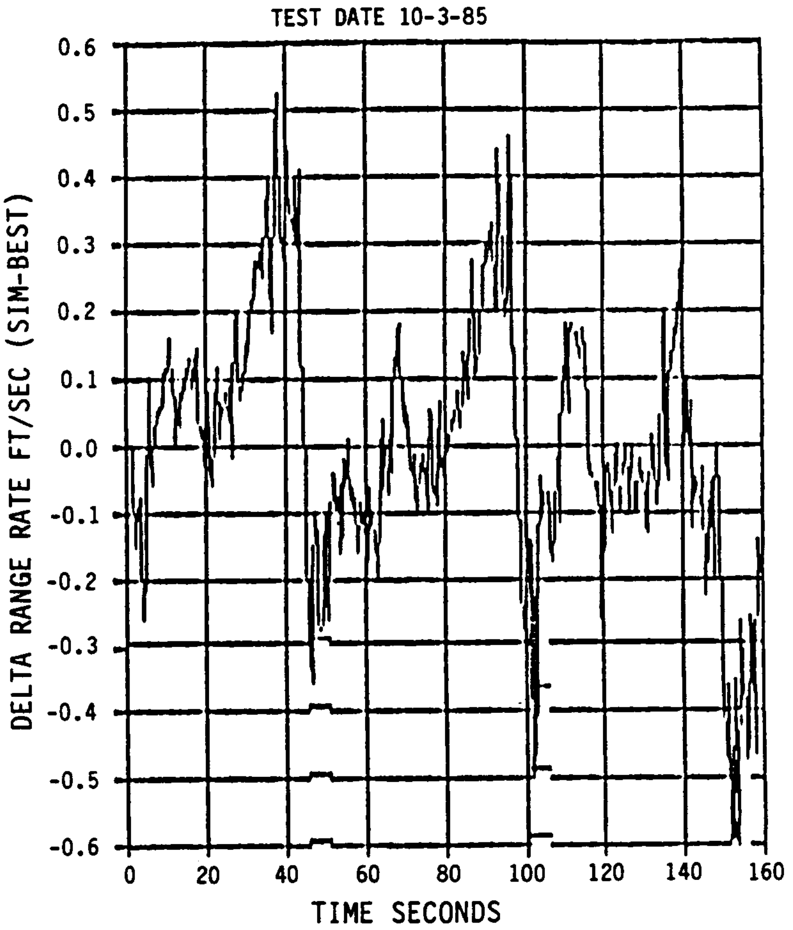


FIGURE 3.4-7 HEL30AF CINE RANGE RATE DIFFERENCE PROFILE TO BE COMPARED WITH RANGE ACCELERATION PROFILE OF FIGURE 3.4-6



MEAN= -0.001 STANDARD DEVIATION= 0.185

FIGURE 3.4-8 SIMULATION GENERATED HEL30AF RANGE RATE DATA REFERENCED TO THE HEL30AF BEST RANGE RATE DATA

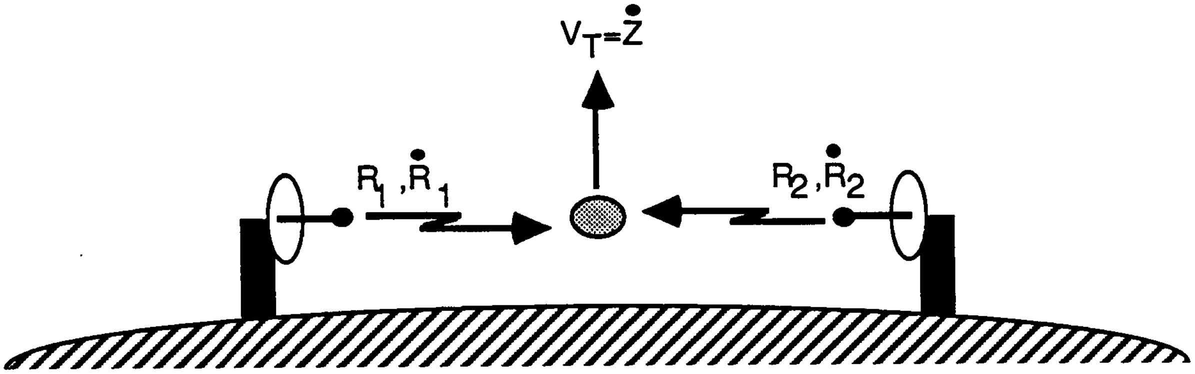
GDOP error in the TMR range rate measurement. The situation described above is illustrated for a two dimensional case in Figure 3.4-9.

The general result of the above qualitative discussion is that any flight profile that puts the target at low altitude, directly over the Ku-Band Radar should have significant error in the TMR range rate measurements. Thus, we should expect TMR GDOP problems with the SAT1, SAT2, and SAT3 data. Also, GDOP problems should be found early in the flight for the BAL and GEM series of tests. To a lesser extent, one should expect GDOP problems with the H30SK series. Even though this flight profile is offset from the brass cap, it is still at a relatively low altitude.

Although there was not time to perform a GDOP analysis of the CINE sensor system, it is appropriate at this point to make some qualitative observations about the CINE GDOP performance in the situation described above. The CINE system develops the target position and velocity using azimuth, elevation, azimuth rate and elevation rate from each of 5 cinetheodolites. When the target is directly over the brass cap each individual cine should have reasonably good knowledge of the target vertical velocity component. Thus, contrary to the TMR system, the CINE system should experience very little problem with GDOP in the range rate measurement for the profiles cited above.

3.4.2.2 GDOP Analysis of SORTE Range Rate Data

The preliminary step in the analysis was to compute the standard deviation of the range rate error produced by GDOP at each point in the flight profile. Then the mean and standard deviation of this GDOP error profile was computed. In the preliminary analysis, the mean of the GDOP error profile was used to screen all of the test cases. If the mean of the GDOP error profile was greater than 0.25 feet/sec, then the test case was examined in further detail. Table 3.4-3 summarizes those cases with significant GDOP error problems that were analyzed in more detail. Results of those analyses are discussed below.



NOTE: NEITHER SENSOR'S MEASUREMENTS CONTAIN INFORMATION ABOUT $\dot{Z}$

FIGURE 3.4-9 ILLUSTRATION OF A SEVERE GDOP VELOCITY SITUATION

TABLE 3.4-3 TEST CASES WHERE GDOP PRODUCED SIGNIFICANT RANGE RATE ERROR

PROFILE	GDOP MEAN* RANGE RATE ERROR	DELTA RANGE RATE STANDARD DEVIATION	
		BEST	CINE
BAL1	1.14	1.38	ND
BAL2	2.95	3.08	ND
BAL6	1.07	1.38	ND
BAL7	1.03	2.90	ND
SAT1	--	2.33	ND
SAT2	1.76	3.06	1.41
SAT3	2.97	6.78	1.85
H30SKAE	1.43	1.47	0.32
H30SKAF	1.37	1.72	0.33
H30SKAG	1.90	0.84	0.83
H30SKAH	1.40	2.21	1.14
H30SKAI	1.17	1.06	0.49
HEL30AF	0.35	0.75	0.50
HEL30AI	0.38	0.67	0.36
HEL30AJ	0.38	0.76	0.38

* This is the mean of the GDOP range rate error standard deviation profile.

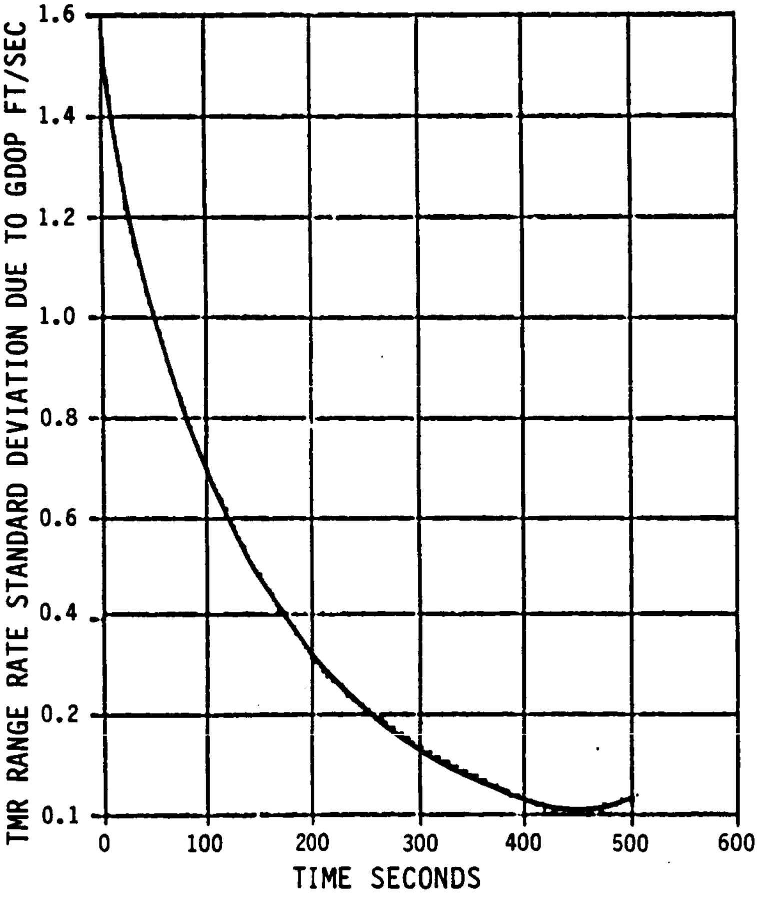
GDOP induced range rate errors are very similar for the BAL and GEM series of tests. Figure 3.4-10 and 3.4-11 give typical examples of the GDOP range rate error for the GEM and the BAL tests, respectively. Both tests have similar shaped GDOP profiles; GDOP range error is large at the beginning of the flight and tapers off rapidly after 100 seconds or so. This behavior correlates perfectly with the qualitative description of GDOP given in Section 3.4.2.1. For these test cases, a helium-filled gemsphere is released at the brass cap and allowed to free-fly. Hence, early in the flight, the target is at very low altitude, e.g. 1000 to 2000 feet. But the balloon rises rapidly to several thousand feet in altitude. Based on the qualitative discussion of Section 3.4.2.1, one would expect large GDOP range rate error at low altitude or early in the flight and small GDOP range rate error at high altitude or late in the flight. The behaviors of the GDOP computation shown in the two figures correlates perfectly with the intuitive explanation.

As further proof that the GDOP computation is correct, a range rate difference profile referenced to the TMR data was computed for the BAL7 profile and is plotted in Figure 3.4-12. A comparison of this profile with the BAL7 GDOP computation given in Figure 3.4-11, indicates good agreement in shape and magnitude between the two profiles. It shows that GDOP dominates over the first 200 seconds and that a different source range acceleration due to target rotation (as discussed in Section 3.4.1.2), dominates over the remainder of the flight.

One final observation about the GDOP calculations for BAL7 and GEM2 is warranted. The difference in magnitudes at the start of the profiles is due to the difference in altitude for initial tracking. As one would expect the initial altitude of BAL7 is much lower than for GEM2.

In the SAT series of tests, the helium-filled GEMsphere was tethered. For the SAT1 and SAT2 tests the range, and approximately the altitude, of the balloon was about 2500 feet above the Ku-Band Radar for the entire test. For the SAT3 test the initial range was 2500 feet and the balloon was reeled into 1200 feet final range. The range of the balloon in the SAT4 test was 10,000 to 12,000 feet. The SAT data of Table 3.4-3

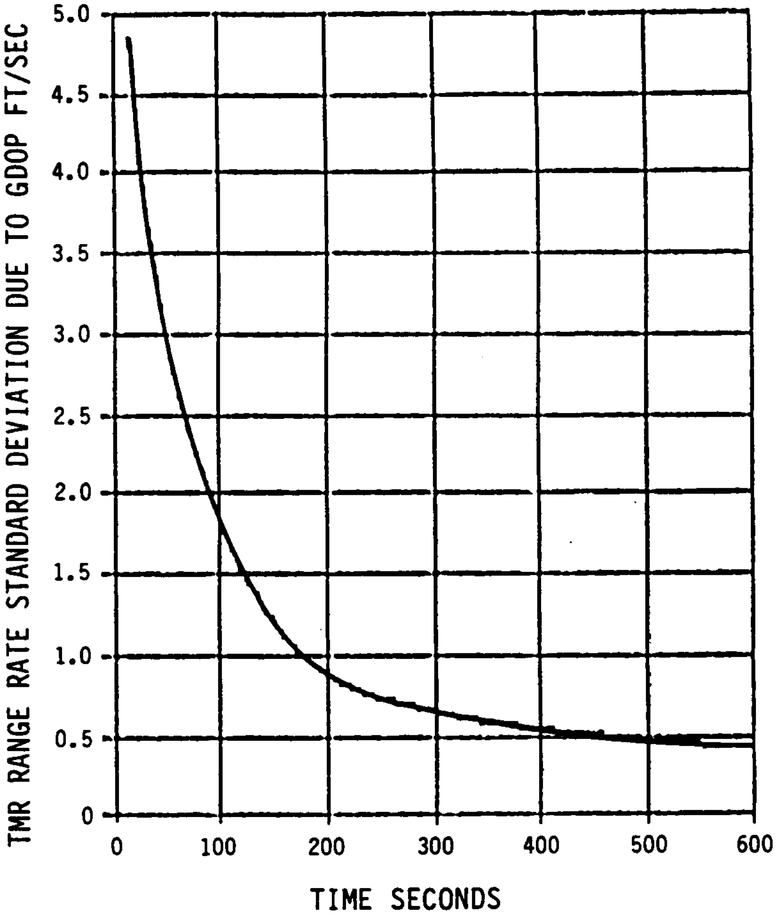
TEST DATE 10-16-85



MEAN= 0.355

FIGURE 3.4-10 GDOP-INDUCED RANGE RATE ERROR STANDARD DEVIATION PROFILE FOR GEM2

TEST DATE 11-4-85

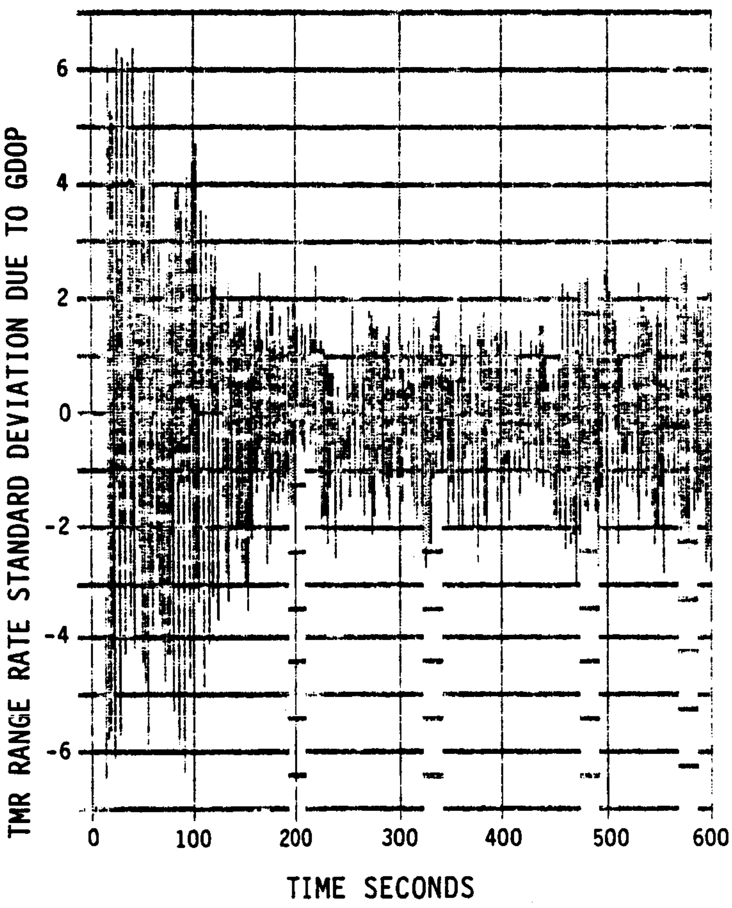


MEAN= 1.03

FIGURE 3.4-11 GDOP-INDUCED RANGE RATE ERROR STANDARD DEVIATION PROFILE FOR BAL7

ORIGINAL PAGE IS
OF POOR QUALITY

TEST DATE 11-4-85



MEAN=.011

STANDARD DEVIATION= 1.577

FIGURE 3.4-12 BAL7 TMR RANGE RATE DIFFERENCE PROFILE AFTER COMPENSATION FOR TARGET ROTATION EFFECTS

C-3

correlates with these test descriptions, since we expect the test run with the lowest average altitude to have the worst average GDOP range rate error and worst difference range rate standard deviation. Figure 3.4-13 gives the GDOP range rate error standard deviation profile for the SAT2 test run.

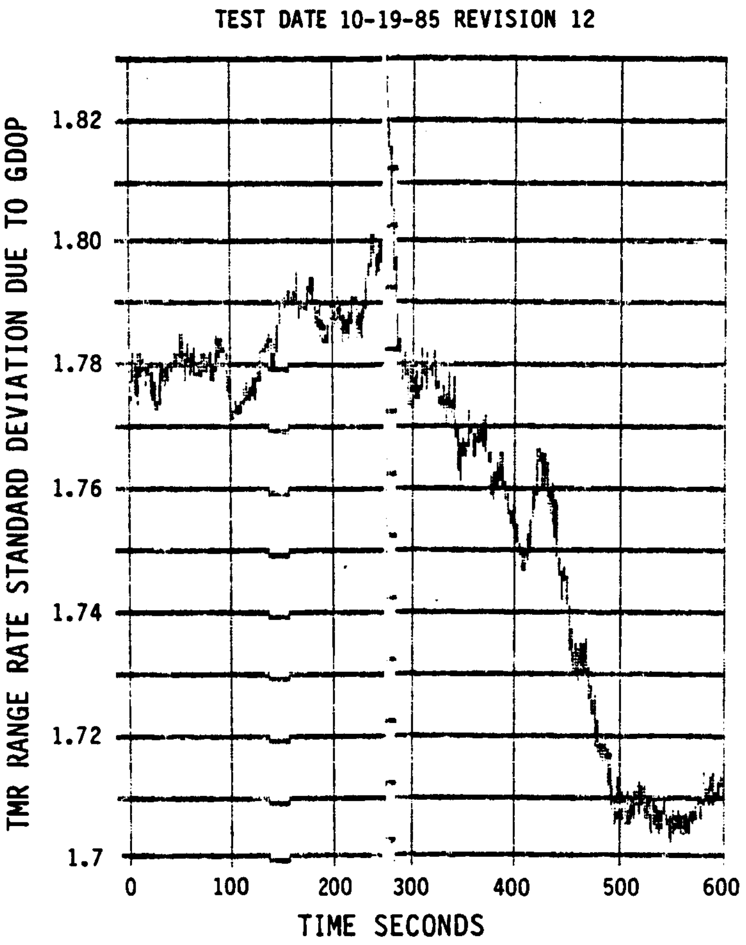
Notice that GDOP range rate error does not appear to significantly affect the CINE data. That this is true can be seen by comparing the difference range rate data referenced to the BEST solution against the difference range rate data referenced to the CINE solution. In both the SAT2 and SAT3 runs, the standard deviation of the BEST data is 2-3 times greater than the CINE data. As was conjectured earlier, the CINE system is not susceptible to low altitude GDOP errors because the sensors in the system measure θ and $\dot{\theta}$, rather than R and $\dot{R}$ which is measured by the TMR sensors.

In the H30SK series of runs, the target, a UH-1H helicopter, flew a 30 degree glide slope toward the radar, starting at a range of 4000 feet and altitude of 1700 feet and finishing at a range of 2000 feet and an altitude of 1500 feet. Figure 3.4-14 gives the GDOP range rate error standard deviation profile for a typical run (H30SKAG). This shows the anticipated behavior: GDOP increases with time because the altitude decreases with time.

A comparison of the CINE range rate difference data and the BEST range rate difference data for this series of test shows that GDOP range rate error is much less significant for the CINE system of sensors. This result is identical to the SAT series of tests and therefore similar comments apply.

The final tests shown in Table 3.4-3 is the HEL30 series. In these tests the helicopter flies toward the radar from a range of 12000 feet into 7000 feet. The starting altitude is 6000 feet and the final altitude is 5000 feet. Since these tests were at a higher altitude and further range than any of the previous sets of tests, one would expect the GDOP range rate error to be smaller than the other cases. This is verified by the data of Table 3.4-3. A GDOP range rate standard deviation plot for the HEL30AF profile is provided in Figure 3.4-15. This data confirms that the GDOP range rate error increases as the altitude decreases.

ORIGINAL PAGE IS
OF POOR QUALITY



MEAN= 1.75

FIGURE 3.4-13 GDOP-INDUCED RANGE RATE ERROR STANDARD DEVIATION PROFILE FOR SAT2

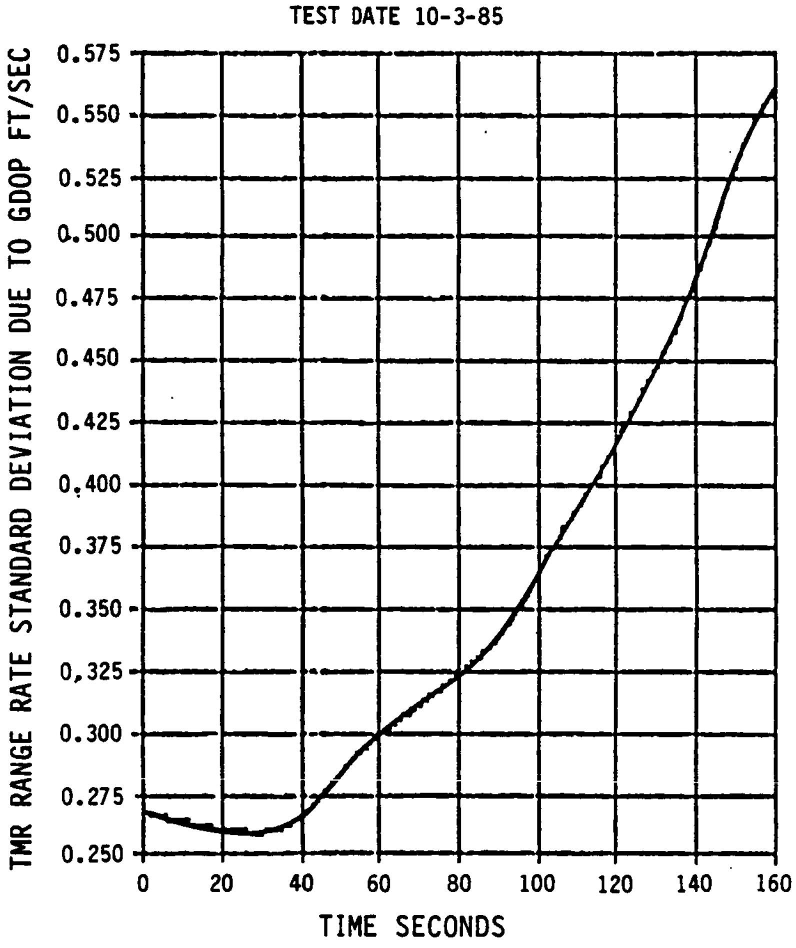
The HL- and HJ- series of tests consisted of tracking a helicopter at long range and high altitude for a duration of 10 minutes. All of the previous discussion on GDOP error would lead us to conclude that these tests, because of high altitude and long range, would have insignificant GDOP range rate error. The GDOP calculations given in Figures 3.4-16 and 3.4-17 support this conclusion for the HJ- and HL- series, respectively.

3.4.3 Target Rotation Effects

An examination of the range rate difference data referenced to the TMR for the GEM and BAL series of tests (see Figure 3.4-2) reveals peak-to-peak oscillations of 10 feet/second. Further investigations showed that the Ku-Band Radar range rate profile had peak-to-peak oscillations of 10 feet/second with a period of 4 seconds. This probing of the data also indicated that the TMR range rate profile also had oscillations with the same peak-to-peak value and period.

3.4.3.1 Evidence Supporting the Target Spin Theory

What was the source of these oscillations? It is conjectured that the source of these range rate oscillations was rotation of the GEMsphere with both the Ku-Band Radar and the TMR radars tracking slowly back and forth across the spinning balloon. There are two facts that lend support to this conjecture. First, examination of the SMMS rendezvous data from flight 41-C reveals a similar oscillation in range rate. In this case it was confirmed through visual observation that the SMMS was in fact rotating about its axis. Computation of the SMMS rotation speed from the peak-to-peak velocity value compared quite well with the rotation speed estimated from the visual observations. The second fact is that both the TMR radars and the Ku-Band Radar produced identical oscillatory patterns. Since these radars operate at widely different RF (2 GHz for the TMR and 14 GHz for the Ku-Band Radar) and the signal processing and waveforms are different, the observed effect must be generated by some mechanism that is independent of the radars. This leaves only the target and its dynamics. The only dynamics that would produce an oscillation in range rate is a spinning of the target.



MEAN= .354

FIGURE 3.4-15 GDOP-INDUCED RANGE RATE ERROR STANDARD DEVIATION PROFILE FOR HEL30AF

The HL- and HJ- series of tests consisted of tracking a helicopter at long range and high altitude for a duration of 10 minutes. All of the previous discussion on GDOP error would lead us to conclude that these tests, because of high altitude and long range, would have insignificant GDOP range rate error. The GDOP calculations given in Figures 3.4-16 and 3.4-17 support this conclusion for the HJ- and HL- series, respectively.

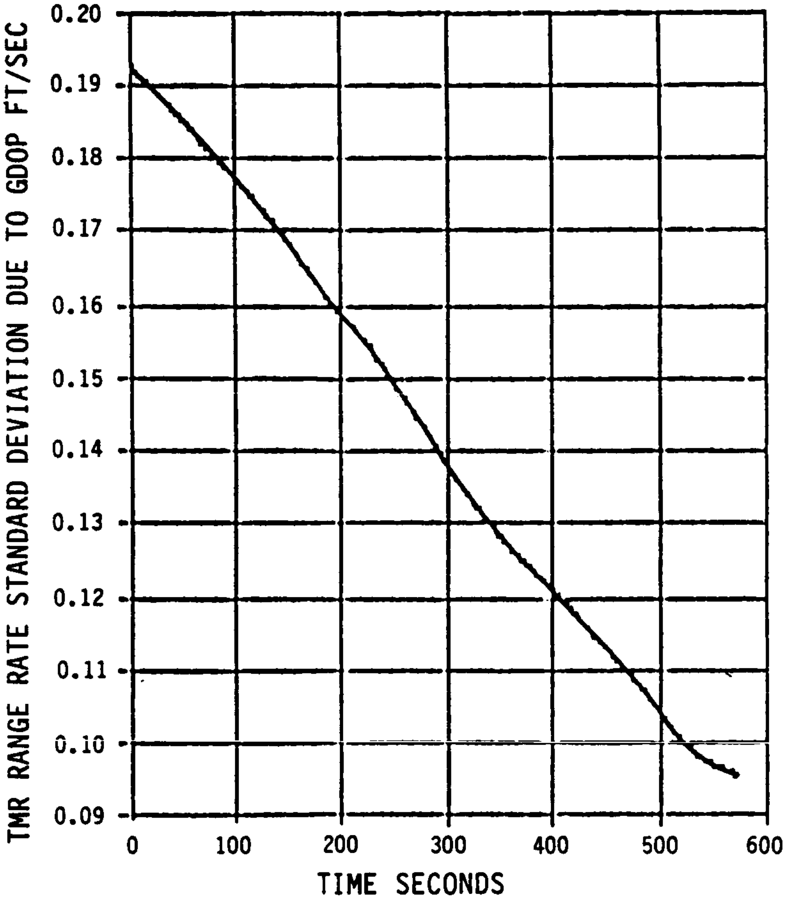
3.4.3 Target Rotation Effects

An examination of the range rate difference data referenced to the TMR for the GEM and BAL series of tests (see Figure 3.4-2) reveals peak-to-peak oscillations of 10 feet/second. Further investigations showed that the Ku-Band Radar range rate profile had peak-to-peak oscillations of 10 feet/second with a period of 4 seconds. This probing of the data also indicated that the TMR range rate profile also had oscillations with the same peak-to-peak value and period.

3.4.3.1 Evidence Supporting the Target Spin Theory

What was the source of these oscillations? It is conjectured that the source of these range rate oscillations was rotation of the GEMsphere with both the Ku-Band Radar and the TMR radars tracking slowly back and forth across the spinning balloon. There are two facts that lend support to this conjecture. First, examination of the SMMS rendezvous data from flight 41-C reveals a similar oscillation in range rate. In this case it was confirmed through visual observation that the SMMS was in fact rotating about its axis. Computation of the SMMS rotation speed from the peak-to-peak velocity value compared quite well with the rotation speed estimated from the visual observations. The second fact is that both the TMR radars and the Ku-Band Radar produced identical oscillatory patterns. Since these radars operate at widely different RF (2 GHz for the TMR and 14 GHz for the Ku-Band Radar) and the signal processing and waveforms are different, the observed effect must be generated by some mechanism that is independent of the radars. This leaves only the target and its dynamics. The only dynamics that would produce an oscillation in range rate is a spinning of the target.

TEST DATE 10-5-85



MEAN= 0.142

FIGURE 3.4-16 GDOP-INDUCED RANGE RATE ERROR STANDARD DEVIATION PROFILE FOR HJ146AD

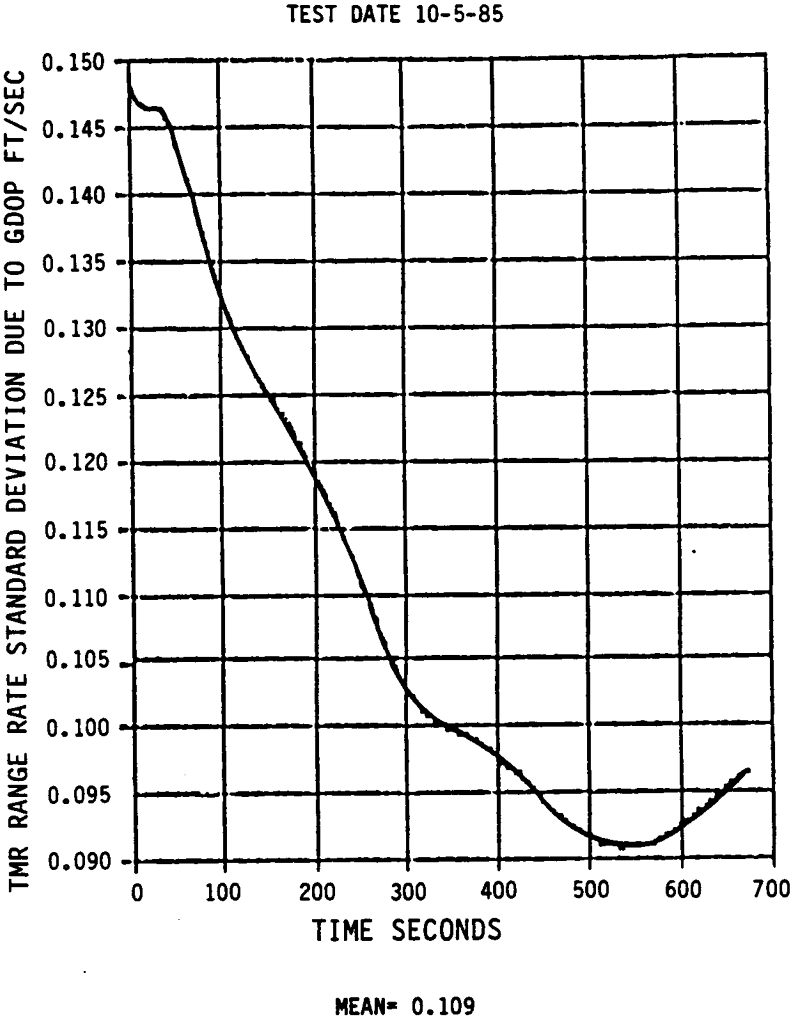


FIGURE 3.4-17 GDOP-INDUCED RANGE RATE ERROR STANDARD DEVIATION PROFILE FOR HL246AE

A natural question to ask is: does the spin rate computed from the oscillatory range rate data correspond to a reasonable value for a free-flying GEMsphere? This spin rate is computed from the following expression

$$\theta = \frac{\Delta V}{R} \quad \text{(cycle/}2\pi\text{ radians)} \tag{3-6}$$

where R = Radius of the GEMsphere

ΔV = 1/2 the peak-to-peak range rate oscillation

Now, using R=3 feet and ΔV = 5 feet/second, it is found that the balloon is rotating at a speed 0.27 cycles/second or one revolution every 3.77 seconds. This rotation rate certainly seems reasonable, especially if there is any air turbulence to generate the tumbling or spinning effect.

3.4.3.2 Modified Analysis of the Difference Range Rate Data

If one accepts the conclusion that target spin produced the oscillation in the range rate data, then GEM and BAL range rate data must be re-evaluated using the following technique. First, observe that both TMR radar and Ku-Band Radar boresights oscillated back and forth over the target with a period of about 4 seconds. This period was identical for both the Ku-Band Radar and the TMR system as shown in Figure 3.4-18. However, a closer look at that data reveals that the oscillations are out of phase which is not surprising. This effect was denoted as "phase skewing" in Table 3.4-1 and this nomenclature will be retained in the sequel.

Now, to analyze the true radar performance, these target spinning effects must be removed. This is done by shifting the TMR range rate profile until the oscillations in this profile align with the Ku-Band Radar range rate profile oscillations. The aligned profiles are then differenced and the statistics of the resulting data are computed. The result of this process for the BAL7 profile is illustrated in Figure 3.4-3. The features found in that profile were then easily explained.

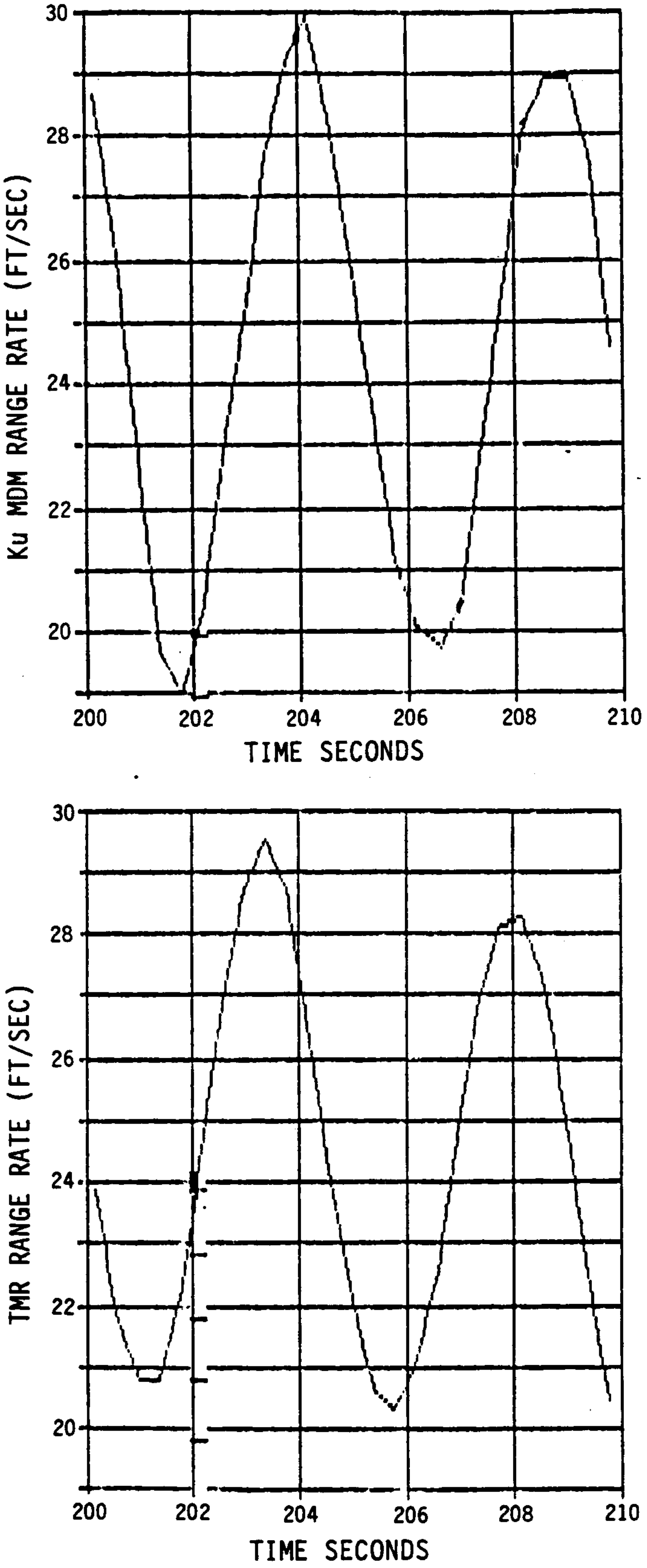


FIGURE 3.4-18 ILLUSTRATION OF THE PHASE DIFFERENCE BETWEEN THE KUBAND RADAR RANGE RATE AND THE TMR RANGE RATE

3.4.4 Low SNR Effects

Up to this point in the discussion, most of the range rate data problems have been reasonably well explained by the error sources discussed in the previous three subsections. However, there remain some test runs with apparent range rate problems that need to be addressed. These cases are part of the HL- series of tests. Table 3.4-4 summarizes those HL- runs that have significant problems in the BEST or TMR range rate difference data. These errors cannot be legitimately explained by target acceleration, rotation, or GDOP. What, then, can be the source? After some investigation, the following theory was developed. Examination of the HJ- series revealed the difference range rate statistics were much better than the HL. But both the HL- and HJ-series used a helicopter as a target and the flights were at about the same range and altitude. What was different between the flight trajectories of the two series? The HL series trajectory was a circular arc with Ku-Band Radar at the center, while the HJ series trajectory was more on a line directly toward the radar. These two trajectories are illustrated in Figure 3.4-19. This means that the helicopter was broadside to the radar in the HL series, but was nose-on to the radar in the HJ series. From the photograph of 3.1-3, one can see that the target enhancement devices (two Luneberg lenses mounted on the underside of the helicopter pointing forward) would help in the nose-on view, but would not provide much assistance in the broadside view.

It was learned during the System Design Verification Tests (SDVT) of the Ku-Band Radar that an UH-1H helicopter had a -5 to 5 dBSM RCS when viewed from broadside. It was also found during these tests that an SNR at the doppler filter output (denoted as SNR_D) of less than 10 dB caused visible degradation in the range rate performance and that the system breaks track for SNR_D less than 0 DB.

Using the information cited above one can compute the SNR_D for the HL series of runs from the expression

$$SNR_D = 183.6 - 40 \log R \text{ (FT)} + 10 \log RCS \; (M^2) + G \qquad (3\text{-}7)$$

where R = Range in feet
RCS = Radar cross section in square meters
G = Gain of the SNR through the digital processor

TABLE 3.4-4 SUMMARY OF THE HL-SERIES WITH PROBLEMS IN THE BEST OR TMR RANGE RATE DIFFERENCE DATA

PROFILE	RANGE RATE DIFFERENCE STD. DEV., FT/SEC	REFERENCE	COMMENTS
HL146AE	0.44	TMR	Strong RCS here
HL246AD	0.48	BEST	Lens fading in and out
HL246AE	0.52	BEST	Lens fading in and out
HL346AD	0.66	BEST	Large gaps where target fades
HL346AE	0.51	BEST	Lens fading in and out
HL346AF	0.56	TMR	No RSS here - RCS value?
HL446AC	0.41	BEST	RCS between 0 and 10dBSM
HL446AD	0.54	BEST	Large gaps where target fades
HL446AE	0.51	TMR	Target 0 dBSM
HL546AC	1.34	TMR	No RSS here - very small RCS
HL546AE	0.67	BEST	Large gaps where target fades
HL546AF	0.54	TMR	Lens fading in and out
HL546AG	0.46	TMR	Strong RCS here

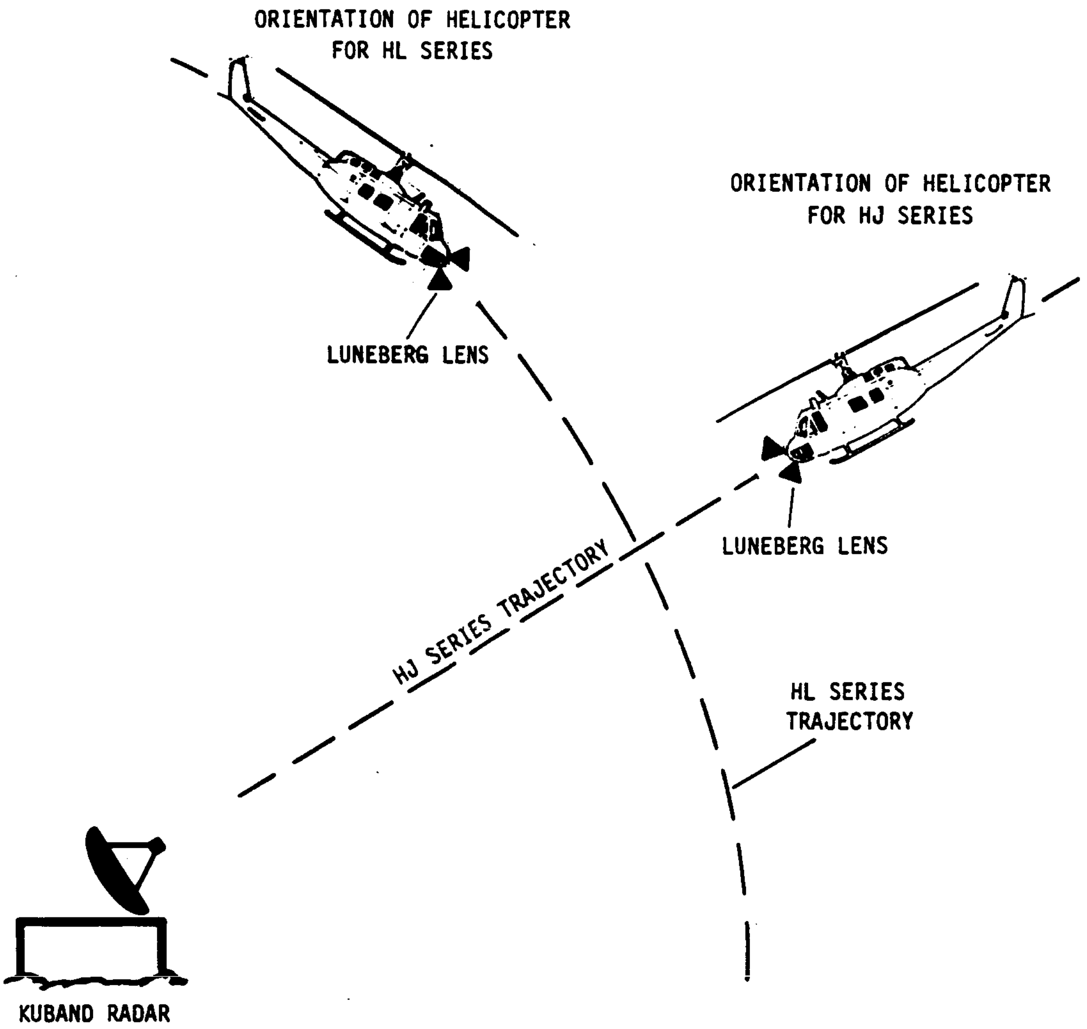


FIGURE 3.4-19 FLIGHT GEOMETRIES FOR HJ AND HL SERIES OF EXPERIMENTS SHOWING ORIENTATION OF LUNEBERG LENS WITH RESPECT TO KUBAND RADAR

Assuming a target range of 47000 feet and a gain of 32, equation 3-7 reduces to

$$\text{SNR}_D = 11.76 + \text{RCS (dBSM)} \qquad (3\text{-}8)$$

Now, if the broadside RCS of a UH-1H helicopter fluctuates between -5 and 5 dBSM, then the SNR_D fluctuates between 6.8 dB and 16.8 dB From this calculation and the system test observations provided above, one can see that it is possible for the range rate estimate of the Ku-Band Radar to be corrupted by internal noise due to a weak return signal.

As further evidence to corroborate the effects of a weak target return signal on the range rate difference data performance, Figure 3.4-20 compares the target RCS profile against the range rate difference profile. One can see that there is a high correlation between the RCS strength and the range rate random error behavior. It is conjectured that the gaps in RCS are due to the lens moving out of view of the radar.

3.5 ROLL AND PITCH ANGLE DATA ANALYSIS

The first pass through the roll and pitch angle data in the analysis procedure is summarized in Table 3.5-1. The first thing that is apparent in this data is that the number of failing cases is lopsided toward the BEST/TMR cases, and that there are virtually no CINE failures. Based on the analysis of the range and range rate data presented in the previous sections, one immediately suspects that GDOP-induced error plays a major role in most of these failures. To support this conjecture, most of the failures would have to be in those flights at low altitudes and very nearly over the PEARL site brass cap. This would principally include the following family of profiles: SAT, BAL, GEM, H30SK and, to a lesser extent, HEL30. Table 3.5-2 provides a breakdown of the failures by flight series. This data shows that the majority of the angle specification failures occur for the GEM through the H30SK series of flights. The failures listed for the HL- and HJ- series of flights will be shown to be caused by other error sources as well; namely, (1) angle acceleration and (2) weak target return signal strength.

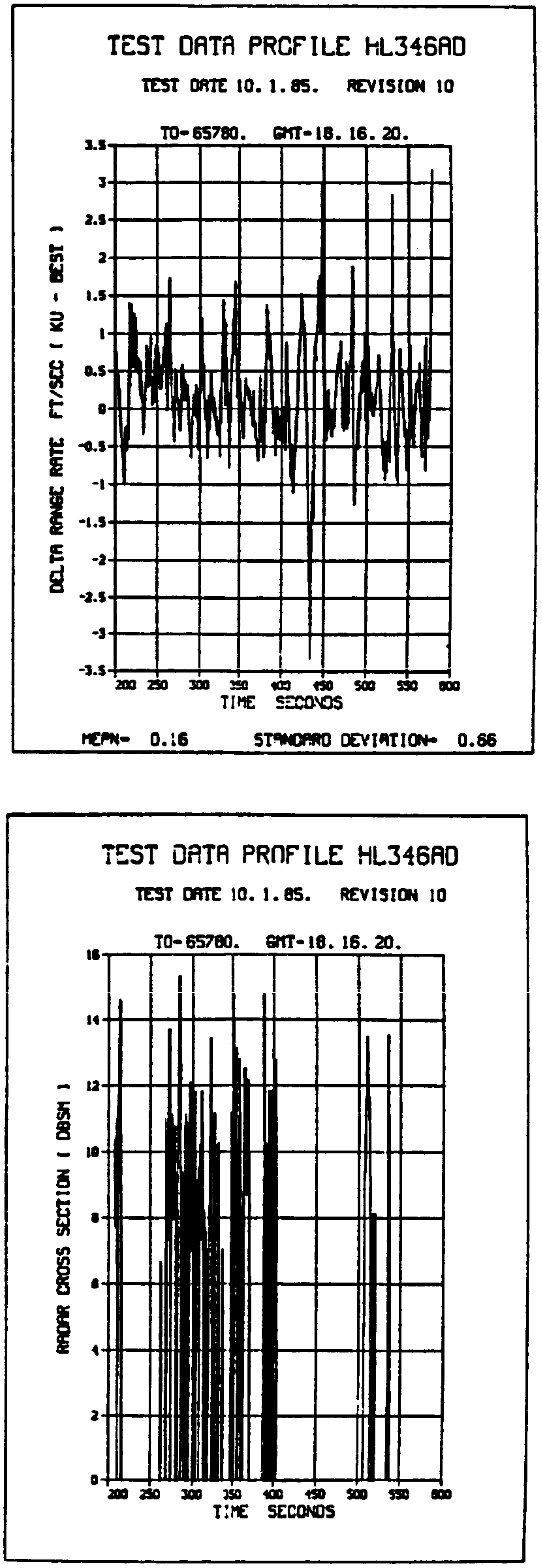


FIGURE 3.4-20 COMPARISON OF THE RADAR CROSS SECTION PROFILE AND THE RANGE RATE DIFFERENCE PROFILE FOR HL346AD

TABLE 3.5-1 SUMMARY OF INITIAL ROLL AND PITCH ANGLE PERFORMANCE ASSESSMENT

		BEST/TMR		CINE		COMBINED
PARAMETER	SPEC	NUMBER FAILING	PERCENT	NUMBER FAILING	PERCENT	TOTAL PERCENT
Roll Angle Mean	0.667 deg	5	8.0	1	1.6	9.6
STD DEV	0.153 deg	23	37.0	4	6.4	43.4
Pitch Angle Mean	0.667 deg	8	12.9	1	1.6	14.5
STD DEV	0.153 deg	11	17.7	1	1.6	19.3

*The data in this table is based on a combined total of 62 sets of data.

TABLE 3.5-2 CATEGORIZATION OF ROLL AND PITCH ANGLE FAILURES BY FLIGHT SERIES

		NO. OF PITCH FAILURES		NO. OF ROLL FAILURES	
SERIES	NUMBER IN SERIES	MEAN	STD DEV	MEAN	STD DEV
GEM	2	0	2	0	2
BAL	5	3	2	1	4
SAT	4	2	3	1	4
H30SK	5	2	2	1	4
HEL30	4	0	1	0	3
HL146 TO 546	13	1	1	2	4
HJ146	3	0	0	0	2

The second observation about the data of Tables 3.5-1 and 3.5-2 is that the roll angle standard deviation failures outnumber the pitch angle standard deviation failures two to one. This also is suspected to be related to GDOP. A quantitative analysis is being done concurrently with the writing of this report to confirm this conjecture.

3.5.1 Description of Angle Error Sources

Before launching into a description of the roll angle and pitch angle analysis, it is worthwhile to list some of the sources that induce error in the angle data. These sources are:

- GDOP
- Coordinate Transformation Inaccuracy
- Angle Acceleration
- Weak Target Return Signal (low SNR)

In the present set of tests, GDOP is the primary source causing failure. The other three sources are equally weighted and are a distant second. A short description of each of these errors sources follows.

GDOP. As discussed in the previous subsections, when the target is at low altitude over the PEARL site brass cap, the TMR sensor system develops a very poor estimate of the target's brass cap Z-coordinate and Z-velocity. This translates into poor range and range rate when the target is over the brass cap. Also observe that this same poor estimate of the target position is folded into the calculation of the target's roll and pitch angle. But, because this calculation is a nonlinear transformation, it is hard to guess the effects of the brass cap Z-component errors on roll and pitch. For reference, the following expression for roll and pitch are provided.

$$\text{(3-9)} \qquad \begin{aligned} &\text{Roll Angle} = \text{ARCTAN } (Y_B/Z_B) \\ &\text{Pitch Angle} = \text{ARCTAN } (X_B/(Y_B^2+Z_B^2)^{1/2}) \end{aligned}$$

where (X_B, Y_B, Z_B) is the target position in the shuttle body coordinate system. The position in body coordinates is obtained from the position in brass cap coordinates through the transformation:

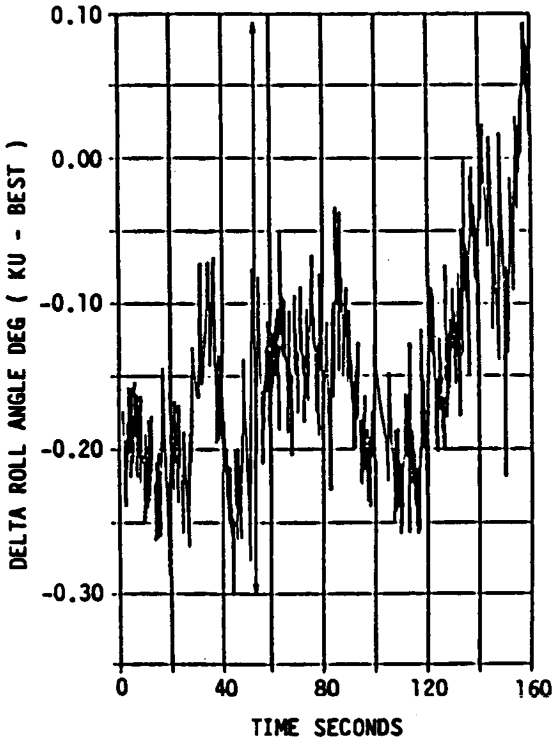
(3-10) $$(X_B, Y_B, Z_B) = T_{BP} (X_P, Y_P, Z_P)$$

where T_{BP} is the transformation matrix and (Xp, Yp, Zp) is the target position in PEARL site brass cap coordinates. The elements of T_{BP} are fixed by the orientation of the radar relative to the brass cap at WSMR. Zp is the brass cap component of importance in this analysis. For the errors along this axis are large due to the geometry of the TMR radars.

Coordinate Transformation Inaccuracies. Prior to the discovery of GDOP as the principal error source in the angle data analysis, a significant amount of time was spent analyzing the effects of errors in the coordinate transformation, T_{BP}, on the angle difference data. These errors take the form of inaccurate estimate of the four angles that compose this transformation: (1) the lower azimuth angle rotation about the brass cap Z-axis, (2) the elevation angle rotation about the new y-axis, (3) the upper azimuth angle rotation about the new Z-axis, and (4) another rotation about Z which transfers the data from the radar frame (Reference 9) to the shuttle body coordinate system. Nominal values for these angles are 30 degrees for the lower azimuth, 30 degrees for elevation, 0 degrees for upper azimuth, and 24.5 degrees for the final rotation. If any of these measured angles are in error, this produces a misalignment between the desired and the actual coordinate system. Appendix E provides a detailed analysis of the effect of misalignment on the computed roll angles.

The results of the analysis was that small errors in any of these angles can produce significant bias in the angle difference data. Consider an example. The value of the lower azimuth angle was changed from 30 to 30.5 degrees and the angle difference data for HEL30AF was recomputed. Figure 3.5-1 compares the original roll angle difference data against the modified difference data. Clearly, the bias has been reduced in the modified data case.

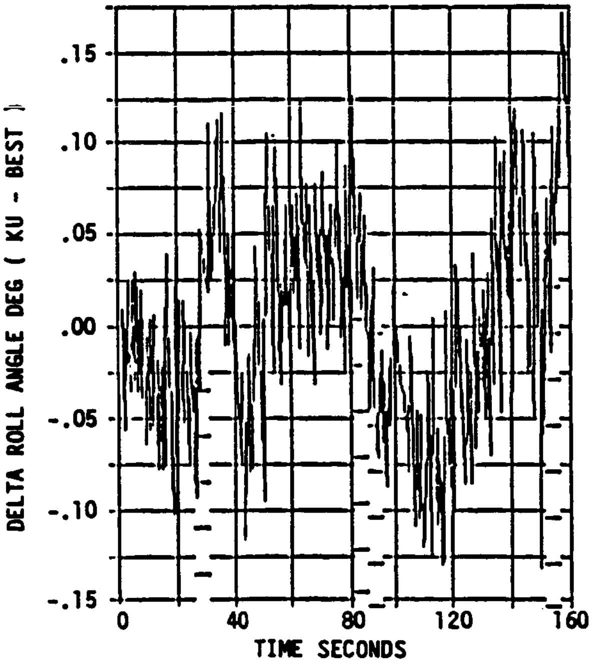
The detailed analysis of angle transformation error effects also showed that the odd-shaped trends found in the angle difference data could not be explained by this error source alone. Hence, other sources were pursued.



MEAN = -0.15 deg.

STANDARD
DEVIATION = 0.07 deg.

LOWER
AZIMUTH = 30.0 deg.



MEAN = 0.02 deg.

STANDARD
DEVIATION = 0.056 deg.

LOWER
AZIMUTH = 30.5 deg.

FIGURE 3.5-1 ILLUSTRATION OF THE EFFECT OF A CHANGE IN LOWER AZIMUTH ANGLE ON THE ROLL ANGLE DIFFERENCE DATA FOR HEL30AF

Angle Acceleration. The present configuration of the angle tracking loop can produce an asymptotic bias in angle in the presence of angle acceleration. This loop can be modelled as a second order loop with the following transfer function:

$$\hat{\theta}(s)/\theta(s) = (\omega_n^2 + \omega_n^2 Ts)/(s^2 + \omega_n^2 Ts + \omega_n^2) \tag{3-11}$$

where $T = 2/\omega_n$ for the design of this particular system. If the target is accelerated at rate, a, (which means $\theta = (a/2)t^2$), then using the final value therein from control theory one can compute the asymptotic bias of the loop from the relation:

$$\text{Angle Bias} = a/\omega_n^2 \tag{3-12}$$

where ω_n is the natural radian frequency of the loop and a is the angle acceleration. Consider an example. Let $\omega_n = 0.754$ Hz and $a = 0.04$ deg./sec^2, then the asymptotic angle bias is 0.07 degrees.

Weak Target Return Signal. This error source is due to a low SNR at the doppler filter output caused by a weak target return signal. This just means that the thermal noise from the receiver is beginning to compete with the desired signal. This will corrupt the angle discriminant which, in turn, corrupts the performance of the angle tracking loop. Unfortunately, an SNR_D threshold where the angle tracking begins to degrade rapidly is not known. However, it is guessed that SNR_D less than 7-8 dB will induce significant degradation in angle tracking performance. What does this mean in terms of a target RCS? Using equation 3-7 and a range of 45000 feet, an SNR_D less than 8 dB implies an RCS of less than -4.5 dBSM. RCS and ranges of these values are found in some of the HL- and HJ- series. Hence, the cause of poor angle tracking performance in these cases is suspected to be weak target signal returns.

3.5.2 Discussion of SORTE Angle Difference Data Problems

BAL Series. According to the flight log given in Appendix G only one radar (R394) was operating during the BAL series of tests. Hence, there is no true TMR solution available and therefore TMR GDOP-induced angle errors cannot exist for this case. The reason for the errors in these cases are not understood at this time.

Figure 3.5-2 gives the difference pitch angle profiles for BAL6 and BAL7. Observe that the error is very large early in the profile (or low altitude) and tapers off rapidly as the gemspheres gain altitude. This shape profile looks suspiciously like a GDOP-induced error. Thus, it is not clear that only one TMR radar was working in this case. This problem probably can be resolved through the official WSMR test logs.

GEM Series. Figure 3.5-3 gives the TMR pitch and roll angle difference data for GEM2 and Figure 3.5-4 gives a similar plot for GEM3. These difference profiles have the same shape as the corresponding range and range rate profiles. It is conjectured that GDOP is the dominant error source early in the flight (through the first 150 seconds). In the latter portion of the flight, both roll and pitch level off to a constant bias term. The source of this bias error is probably due to error in the coordinate system transformation as discussed in Section 3.5.1. The angle accelerations involved are at least an order of magnitude too small to produce the bias indicated in the figures.

The initial pitch and roll error for GEM3 is slightly worse than the corresponding values for GEM2. In addition, the GEM3 profile starts at an altitude of approximately 1600 feet, while GEM2's initial altitude is about 2000 feet. These facts are consistent with the earlier description of GDOP-induced error as a function of altitude.

Concurrent with the writing of this report, an effort is under way to quantitatively verify the theory described above.

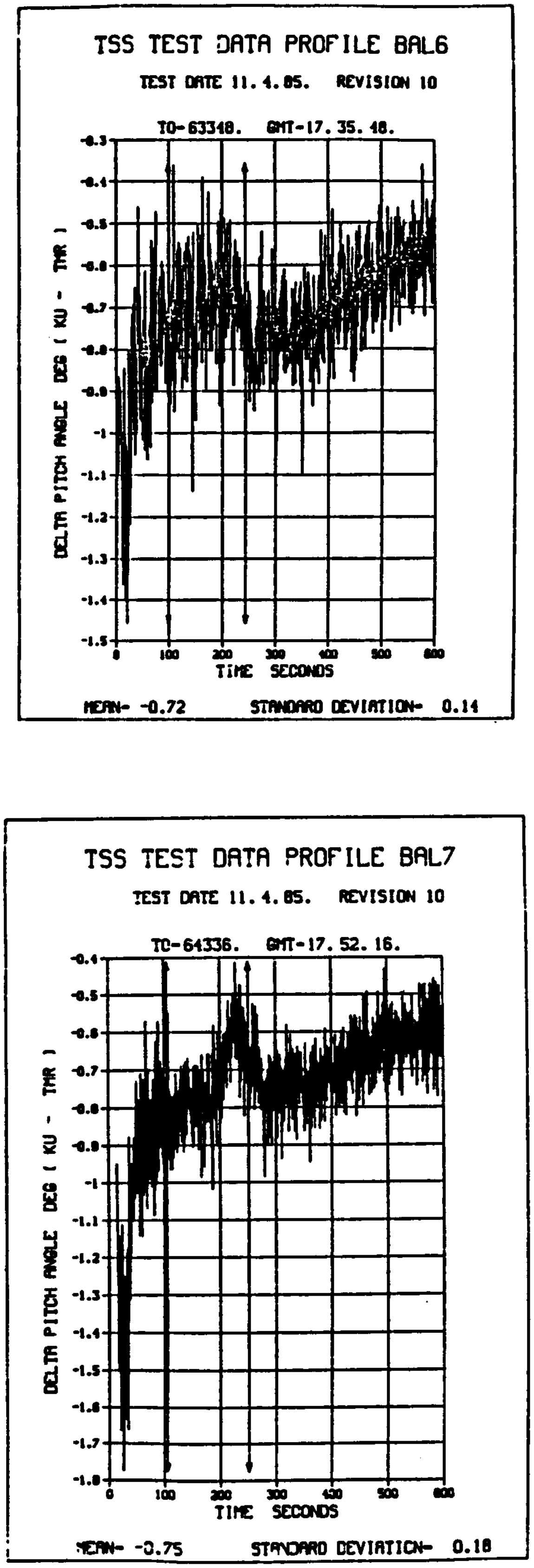


FIGURE 3.5-2 ILLUSTRATION OF THE PITCH ANGLE DIFFERENCE DATA FOR THE BAL6 AND BAL7 PROFILES

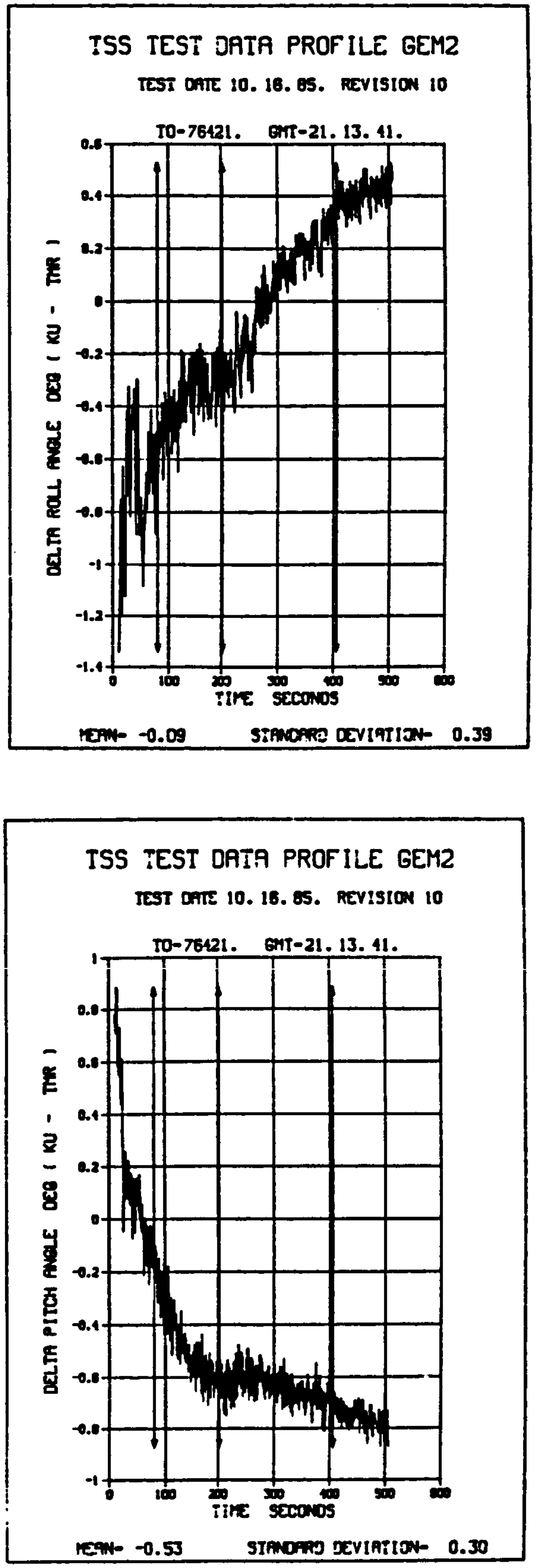


FIGURE 3.5-3 TMR ROLL AND PITCH ANGLE DIFFERENCE DATA FOR THE GEM2 PROFILE

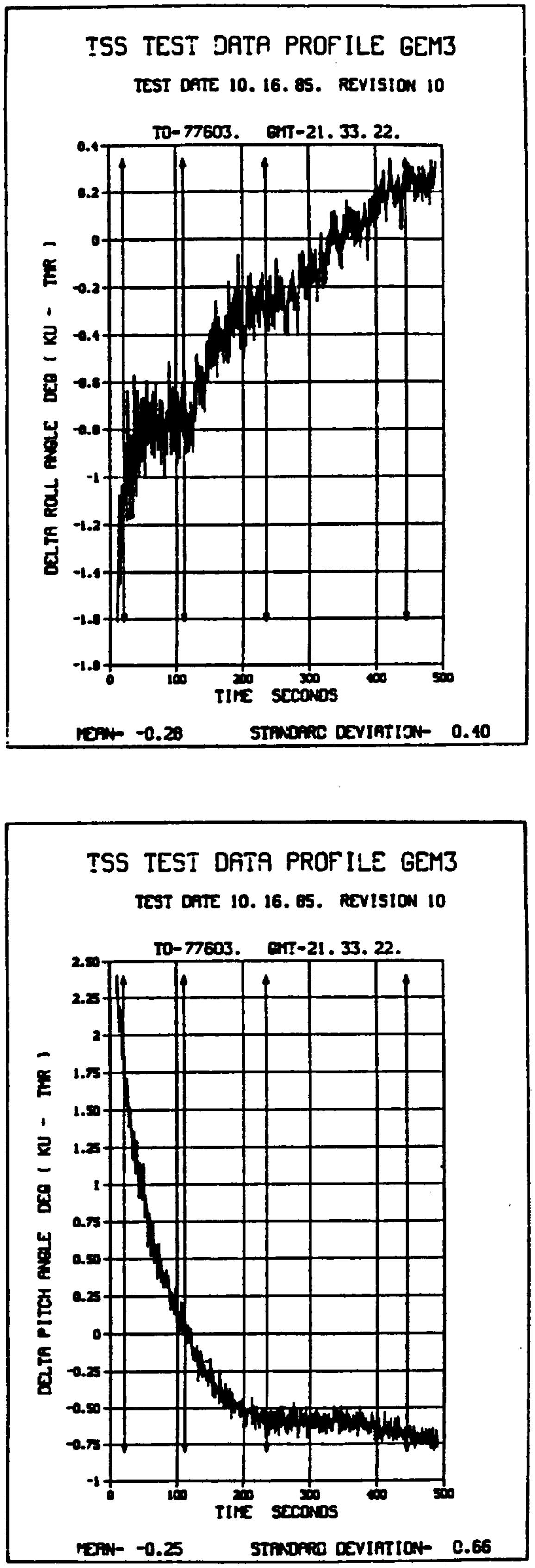


FIGURE 3.5-4 TMR ROLL AND PITCH ANGLE DIFFERENCE DATA FOR THE GEM2 PROFILE

H30SK Series. Of this series of tests, H30SKAI gave the best results, i.e. the fewest specification failures. The major difference between this profile and the others is that the target flew to a final range of 3300 feet, rather than 2000 feet. Assumption of GDOP as the primary error source would fit this failure pattern quite well.

Of the remaining flights in this series, H30SKAH gave the worst results in roll and pitch angle difference data. Since the flight paths were quite similar, it is hard to decide just what the difference might be. Close examination reveals that the GDOP error, which is considered the principal source here, is quite sensitive to the ground track at these altitudes and ranges. In particular, Figure 3.5-5 shows that the pitch angle difference data and range difference data are both highly correlated with the Y- Brass cap coordinate. This lends support to the idea that errors are heavily position dependent.

Thus far, the H30SK series is the first series discussed where the CINE data is available as a reference. There are two observations we can make about this data and its relation to the TMR data. Firstly, the CINE roll and pitch angle differences for H30SKAH flight are well-behaved as shown in Figure 3.5-6. This is in direct contrast to the BEST data for the same flight. But remember the arguments from a previous section. While GDOP is a major problem for the TMR for targets at low altitude, directly over the brass cap, it is not a problem for the CINE sensor system. Hence, the data of Figure 3.5-6 does not conflict with the previously discussed data, but instead supports the conclusions of that discussion.

The second observation concerns the mean of the CINE data. The data of Figure 3.5-6 shows that there is a significant bias in the pitch angle (0.5 - 0.6 degrees) and a bias of approximately 0.2 - 0.3 degrees in range. These biases are consistent for all of the H30SK flights, including H30SKAI. The principal source of this error is believed to be errors in the angles of the brass cap-to-shuttle body coordinate transformation matrix. This data lends support to the GEM data analysis as well. The GEM angle difference data (see Figures 3.4-3 and 3.4-4) decayed to a fixed bias level. The H30SK CINE supports the argument that this bias is the result of transformation error and not a residue of GDOP.

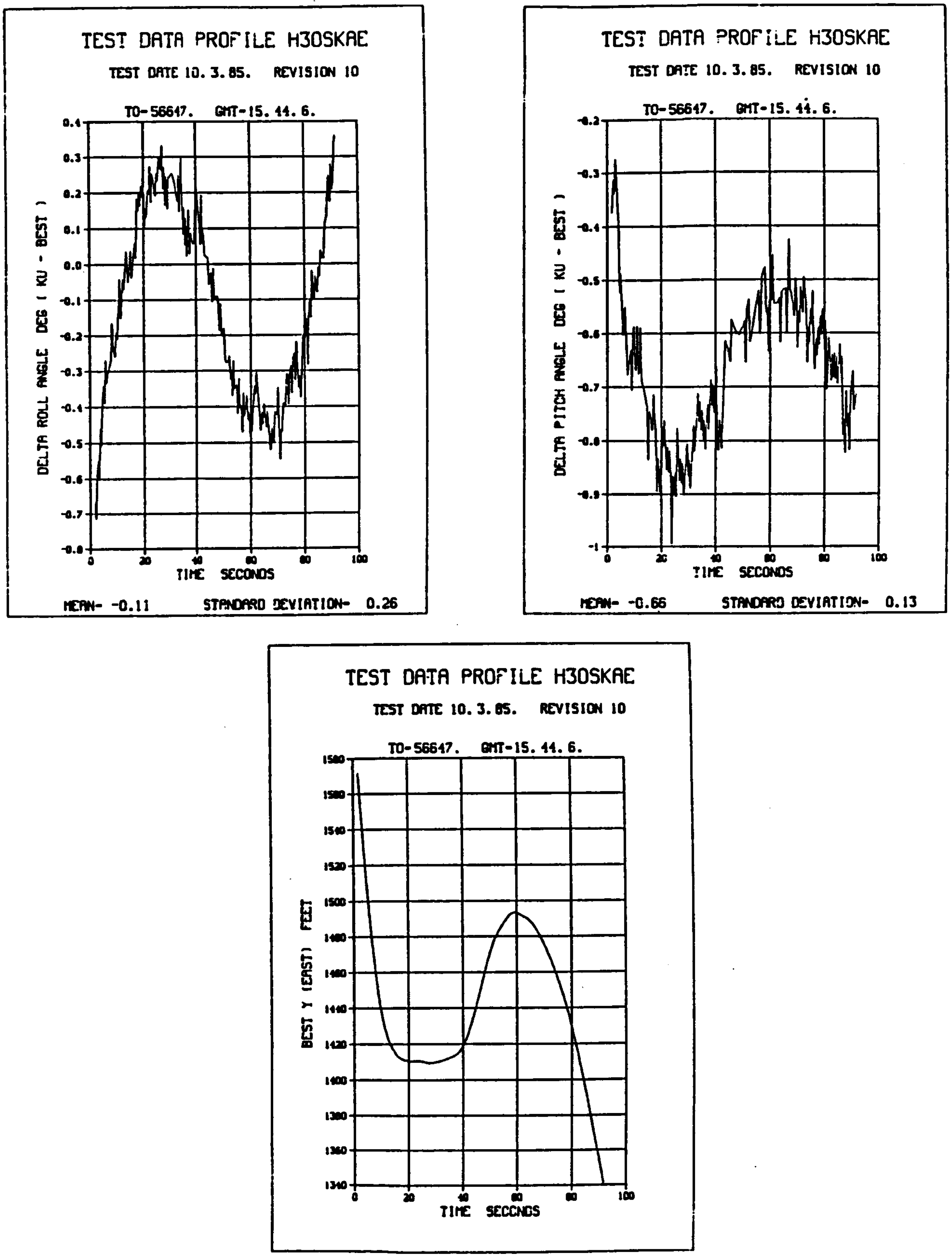
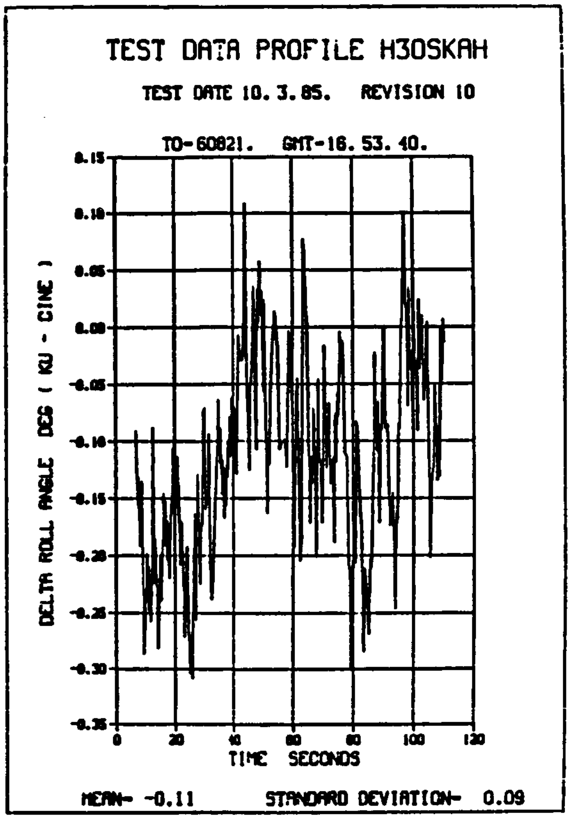


FIGURE 3.5-5 ILLUSTRATION OF HIGH CORRELATION BETWEEN Y-BRASS CAP COORDINATE AND THE ANGLE DIFFERENCE DATA

ORIGINAL PAGE IS OF POOR QUALITY



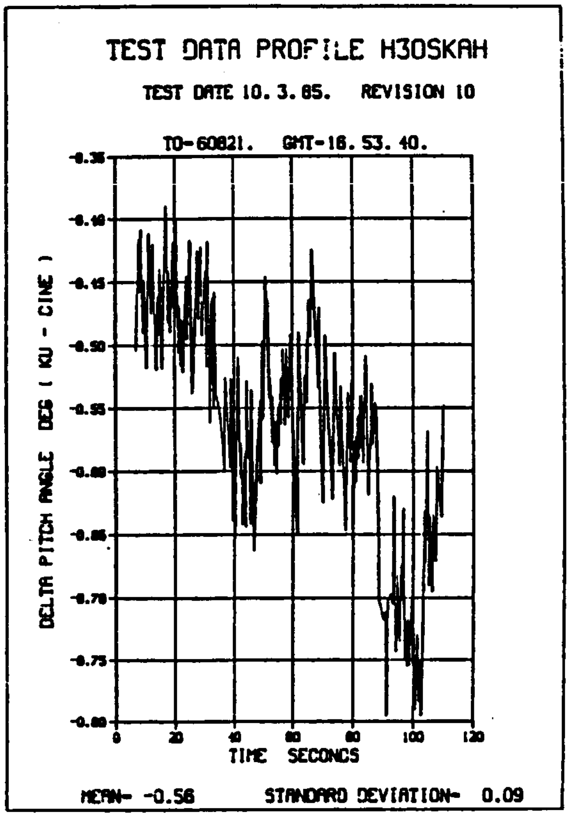


FIGURE 3.5-6 CINE ROLL AND PITCH ANGLE DIFFERENCE DATA FOR H30SKAH TO BE COMPARED WITH THE BEST DATA OF FIGURE 3.5-5

HEL30 Series. Of the four flights in this series, HEL30AJ has the worst performance. The roll and pitch angle difference data provided in Figure 3.5-7 shows that while GDOP does affect the error in the first 300 seconds, the error rapidly increases in the last 100 seconds. As noted in previous discussions, GDOP not only increases with decreasing altitude but is very sensitive to the X-Y ground track. Figure 3.5-8 shows the X-Y ground track. The last 100 seconds of this profile correlates well with the roll and pitch data because it shows the target flying directly toward the brass cap.

Since all of the HEL30 flight profiles were quite similar, one wonders why the errors vary significantly from flight-to-flight. A closer examination of the data revealed that in both HEL30AJ and HEL30AI the final altitude was the lowest (3200' to 4000') and the errors in these two cases were the worst. On the other hand, HEL30AF and HEL30AG both had a final altitude of 5000 feet and both had significantly better angle difference data performance.

At this point it would be best to have quantitative calculations to support these conclusions. Unfortunately, this work is being done in parallel with the final report.

Finally, to add further support to the conclusion that the error shown in Figure 3.5-7 is a function of the TMR radar, Figure 3.5-9 gives the CINE pitch angle difference data for HEL30AJ. This data clearly shows there is not a problem with the Ku-Band roll and pitch angle estimates.

HL- and HJ- Series. Table 3.5-3 summarizes the cases that failed in the HL- and HJ- series. An analysis of the individual cases generated the following observations.

The roll angle difference data of HJ146AC showed a high correlation with the -Z (or altitude) profile as shown in Figure 3.5-10. Since the CINE result showed no problem in roll angle, GDOP is suspected. Although this is somewhat surprising at this range. Also observe that a weak target return signal was not a problem in this case as the random component had peak-to-peak fluctuations of 0.1 degrees. (The data set for HJ146AE was missing.)

ORIGINAL PAGE IS
OF POOR QUALITY

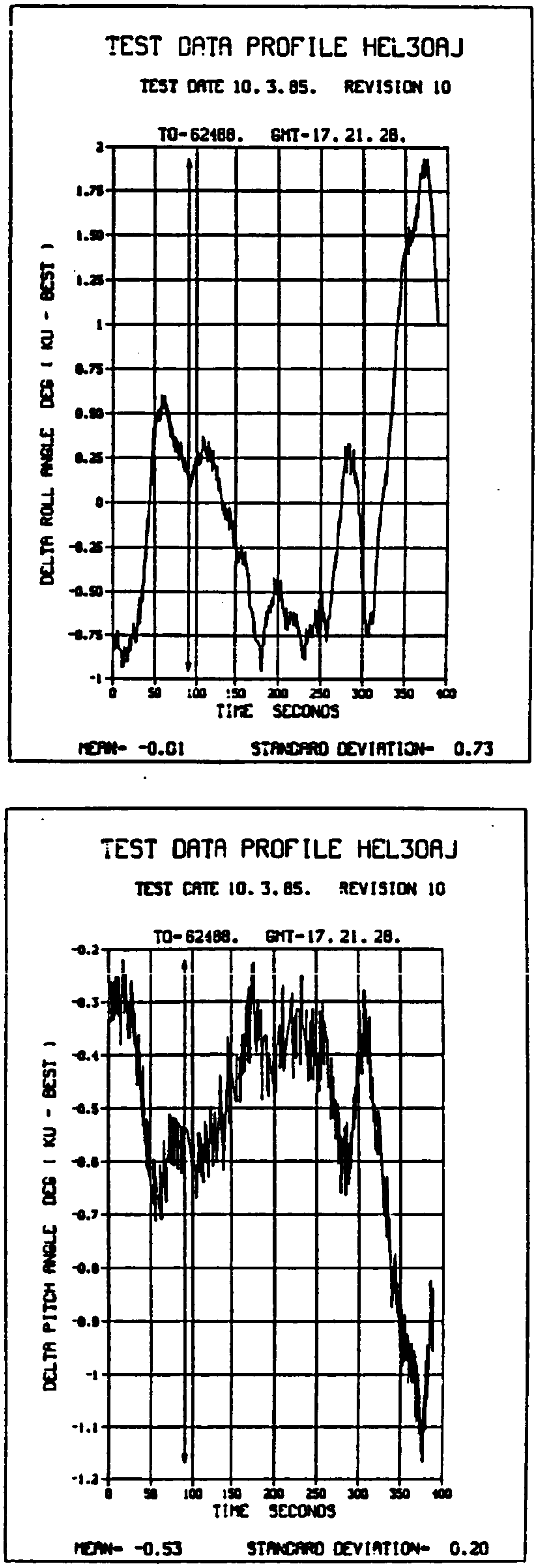


FIGURE 3.5-7 BEST ROLL AND PITCH ANGLE DIFFERENCE DATA FOR HEL30AJ

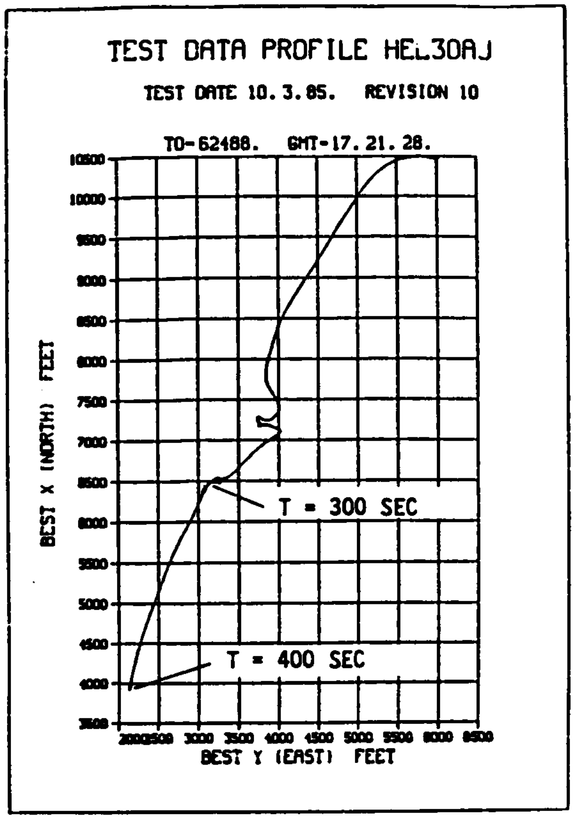


FIGURE 3.5-8 ILLUSTRATION OF X-Y GROUND TRACK FOR HEL30AJ TO BE COMPARED WITH FIGURE 3.5-7

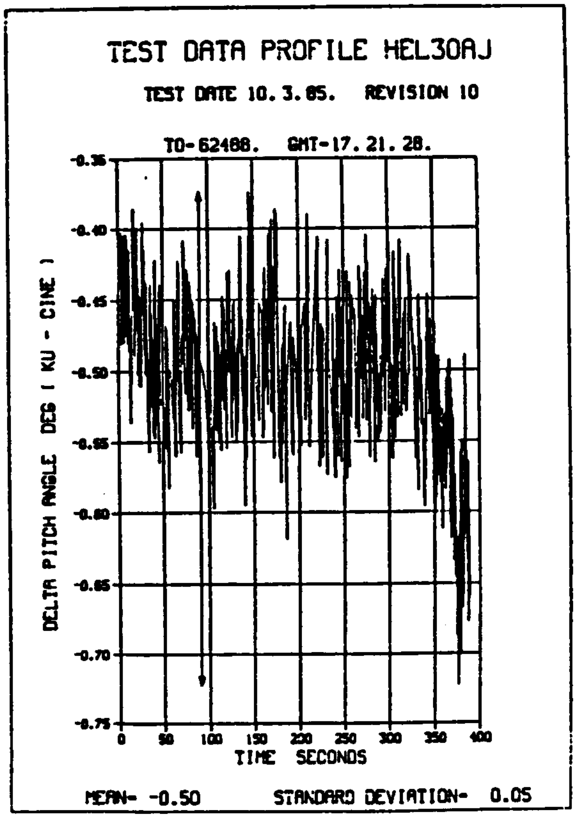


FIGURE 3.5-9 CINE PITCH ANGLE DIFFERENCE DATA TO BE COMPARED WITH FIGURE 3.5-7

ORIGINAL PAGE IS
OF POOR QUALITY

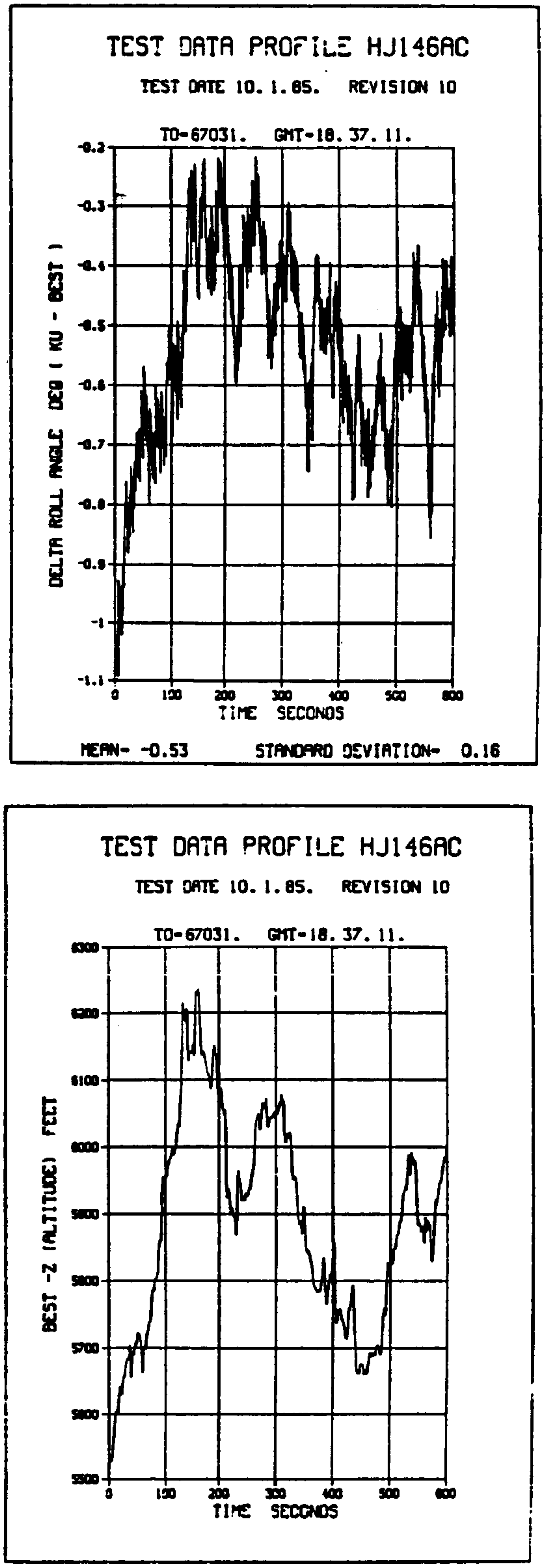


FIGURE 3.5-10 ILLUSTRATION OF CORRELATION BETWEEN THE ROLL ANGLE DIFFERENCE DATA AND THE BEST ALTITUDE PROFILE

TABLE 3.5-3 SUMMARY OF ANGLE DIFFERENCE DATA FAILURES

	TMR/BEST		CINE	
PROFILE	MEAN	STD. DEV.	MEAN	STD. DEV.
HJ146AC	--	0.1573R	--	--
HJ146AE	--	0.2364R	--	0.2359R
HL346AE	--	0.1870R	--	0.1622R
HL446AC	--	0.2439R	--	--
HL446AE	--	0.1972R	--	0.1968R
HL546AC	-0.6859R	2.8925R	ND	ND
		0.9770P		

P = Pitch R = Roll ND = No Data

The problem in the HL346AE data appears to be related to angle acceleration. The roll angle difference data of Figure 3.5-11 shows a significant increase in error around 200 seconds. The size of the change in bias would indicate an angle acceleration (or deceleration) with a magnitude of about 0.01 deg/sec^2. The CINE roll angle difference data shows the same major feature.

The trend in the HL446AC BEST roll angle difference data is apparently caused by GDOP, since it appears to be highly correlated with the altitude data as shown in Figure 3.5-12. If the problems were due to coordinate transformation, then the trends would have been found in the CINE data. Also a weak target return signal strength is not a problem as indicated by the peak-to-peak random fluctuations.

The problem with the HL446AE appears to be a glitch of about 2 degrees between 200 and 300 seconds into the flight. This glitch shows up in roll and pitch in both the CINE and the TMR data. The conjecture in this case is angle acceleration. The magnitude of the acceleration is on the order 0.04 deg/sec^2. Weak target return signal strength does not appear to be a problem in this case.

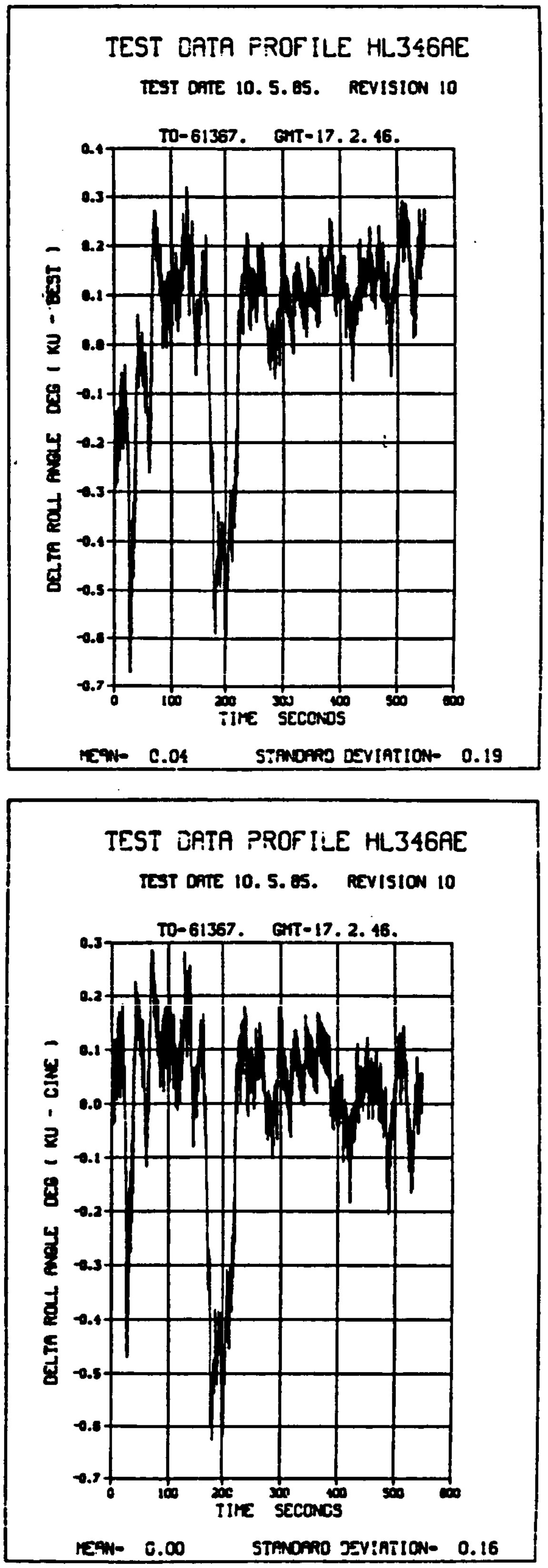
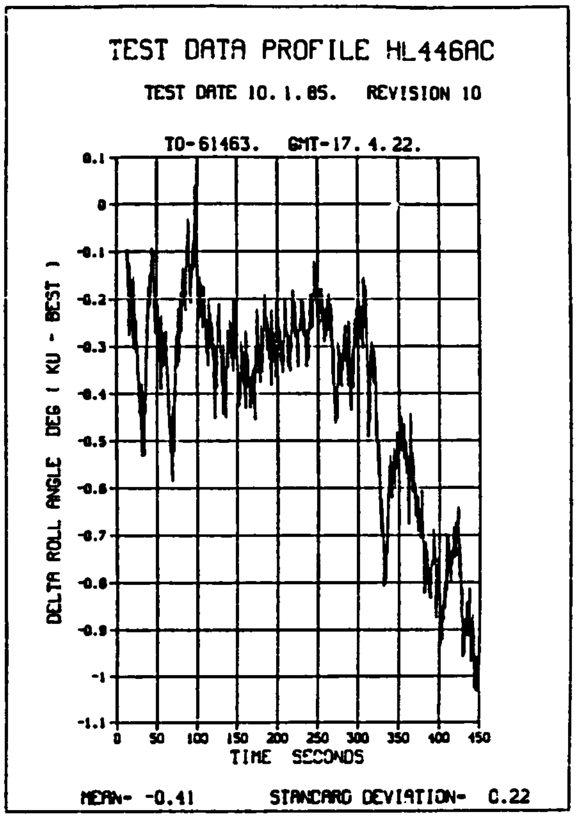


FIGURE 3.5-11 BEST AND CINE ROLL ANGLE DIFFERENCE DATA. THE NEGATIVE-GOING GLITCH IS DUE TO ANGLE ACCELERATION



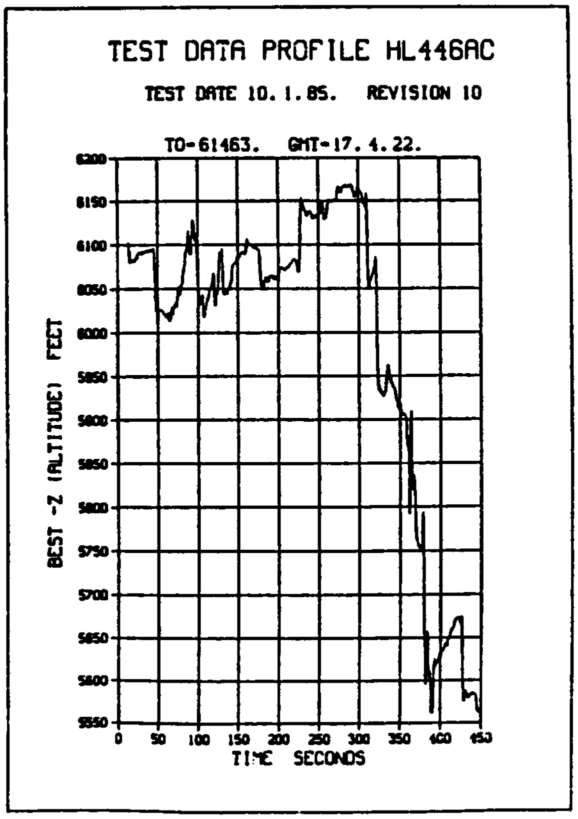


FIGURE 3.5-12 ILLUSTRATION OF CORRELATION BETWEEN ROLL ANGLE DIFFERENCE DATA AND THE BEST ALTITUDE PROFILE

The HL546AC roll and pitch angle difference data are definitely corrupted by GDOP due to the target being at low altitude. Figure 3.5-13 compares the roll angle difference data with the altitude profile. High correlation between these two profiles is evident. Based on this data and the data from HJ146AC and HL446AC, GDOP appears to become a major factor for altitudes less than 5000 feet.

3.5.2.1 Explanation of GDOP-Induced Error in Angle at Long Range

When the analysis of the data was first started, it was thought that angle data failures in the long range cases, i.e. the HL- and HJ- series, would be for reasons other than GDOP just as in the range and range rate data analysis. However, the situation in this case is very different. An explanation of the difference follows.

In both the HJ- and HL- flight configurations, the roll and pitch angle calculations include the Z-component (or altitude component) of target position in the brass cap coordinate system. As explained in an earlier section, any time the target is very nearly in the plane containing the TMR radars, the error in the out-of-plane coordinate (or the Z-component) is extremely large. This is because the TMR radars measure range, so they can only achieve accurate X-Y target position components when the target is near the plane of radars.

Now, the CINEs do not have a problem measuring the Z-component in the HJ- and HL- case for two reasons. Firstly, the five CINEs were chosen to surround the target flight path as shown in Figure 3.1-5, so they will not have trouble with a long range target. Secondly, they do not have trouble with measuring a target's Z-component when the target is at low altitude near the plane of the CINEs.

From the argument of the previous paragraph, it can be concluded that the CINE Z-component (or altitude) data profile can be used as a reference to determine the error in the BEST Z-component (or altitude) profile. Figure 3.5-14 compares the CINE altitude profile to the BEST altitude profile for the HJ146AC flight. This comparison clearly shows the

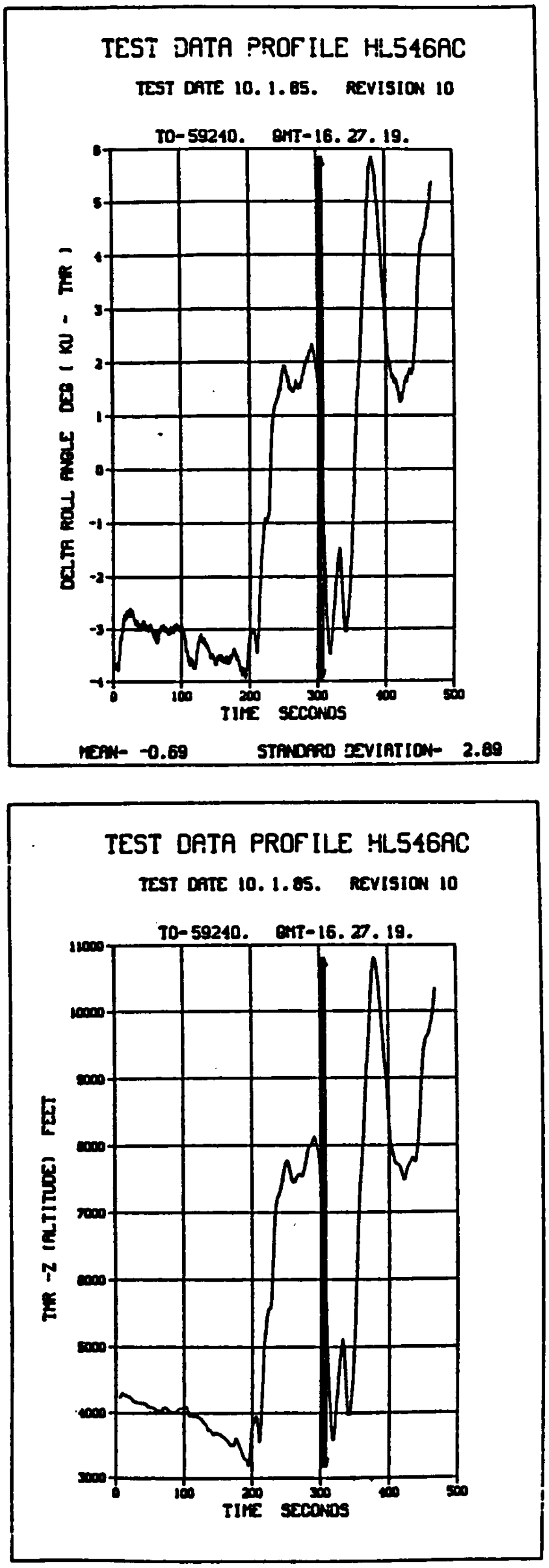


FIGURE 3.5-13 ILLUSTRATION OF CORRELATION BETWEEN ROLL ANGLE DIFFERENCE DATA AND BEST ALTITUDE PROFILE

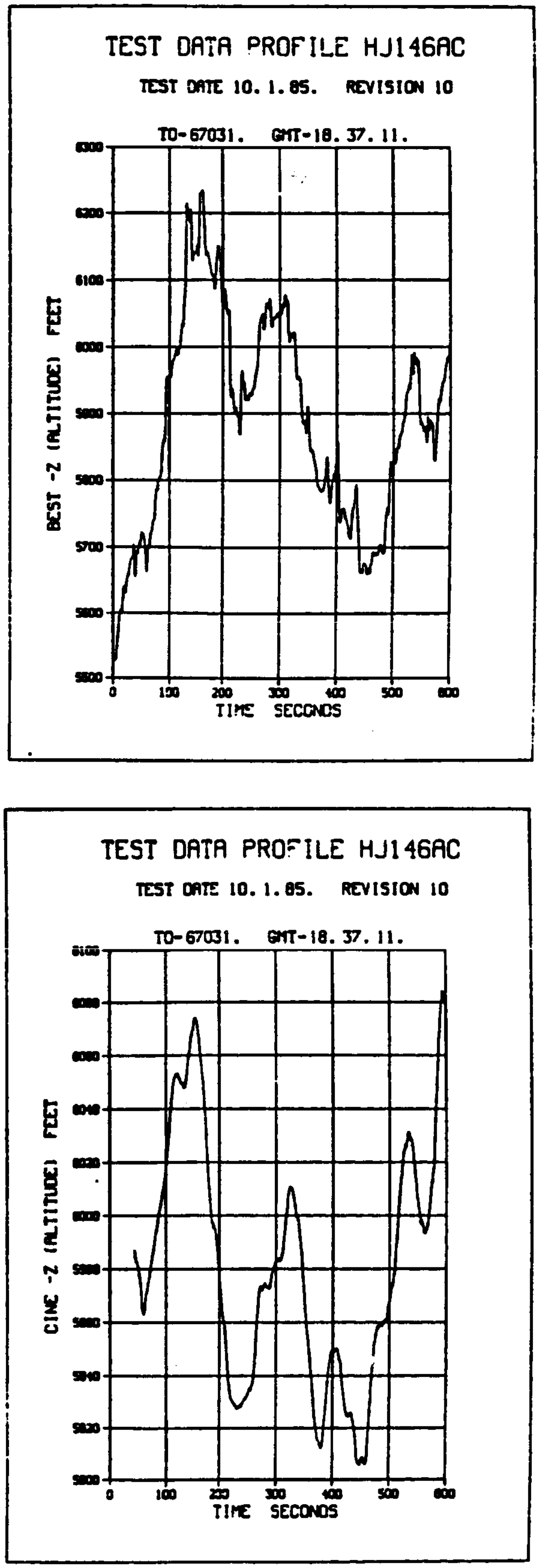


FIGURE 3.5-14 A COMPARISON OF THE CINE ALTITUDE AND THE BEST ALTITUDE FOR THE HJ146AC PROFILE

BEST altitude errors of 300 feet or more, especially at the lower altitudes. Now, how does this affect the roll and pitch angle accuracy? This is hard to answer for the present flight geometry. So let's simplify the situation. Assume that the pitch angle is zero and that the error is entirely in the roll angle. This situation is depicted in Figure 3.5-15.

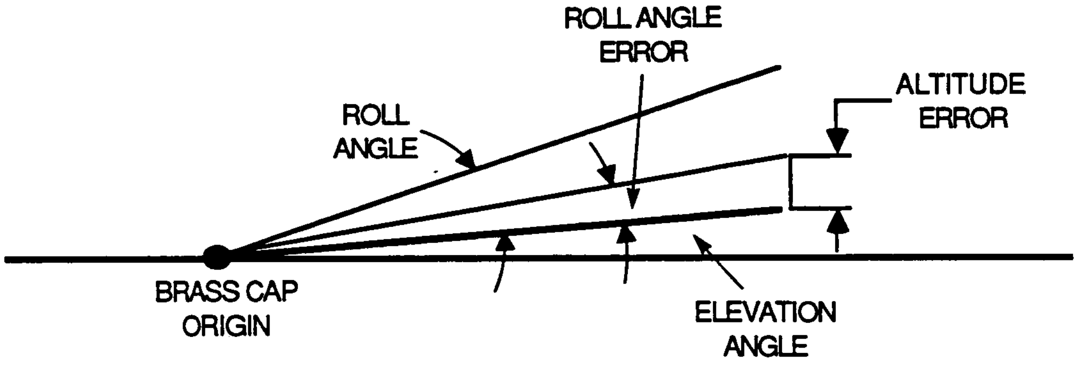


FIGURE 3.5-15 ILLUSTRATION OF EFFECT OF ALTITUDE ERROR ON ROLL ANGLE ESTIMATE

The error in roll angle can be calculated as follows:

Roll Angle Error = Z cos(E)/R

where Z = Altitude Error = 500 feet

E = Elevation Angle = 6.3 degrees

R = Range = 45,000 feet

Using the above values for Z, E, and R, the roll angle error is 0.63 degrees. This magnitude of the error fits with the data presented for the HL- and HJ- series tests.

3.6 ILOS ROLL AND PITCH ANGLE DATA ANALYSIS

Table 3.6-1 summarizes the results of the preliminary analysis of the ILOS roll and pitch range difference data. As in the range rate difference case, the number of failures in angle rate was quite alarming. Furthermore, it contradicts the flight rendezvous data. These data indicated a problem with the random component inside 1.9 nautical miles or in the widest tracker bandwidth case. For ranges greater than 1.9 nautical miles, the angle rate random component was well within specification.

TABLE 3.6-1 SUMMARY OF INITIAL ILOS ROLL AND PITCH RATE PERFORMANCE ASSESSMENT

		BEST/TMR		CINE		COMBINED
PARAMETER	SPEC	NUMBER FAILING	PERCENT	NUMBER FAILING	PERCENT	TOTAL PERCENT
Roll Angle						
Mean	0.0027 deg/sec	33	53.2	25	40.3	93.5
STD DEV	0.0027 deg/sec	36	58.0	26	42.0	100.
Pitch Angle						
Mean	0.0027 deg/sec	36	58.0	26	42.0	100.
STD DEV	0.0027 deg/sec	36	58.0	26	42.0	100.

*There are a total of 62 difference data sets.

A second pass through the data showed that the only two cases passing the mean specification were SAT1 and SAT2. As it turns out, these two tests were the only two tests where the target remained stationary with respect to angular motion for the entire test. So, the preliminary indication was that there was something wrong in those cases with angular motion, which does not occurin the cases with no angular motion. An extensive analysis was undertaken to determine the nature of this problem. Results of that analysis are described below.

3.6.1 Preliminary Analysis Results

The analysis was started by looking at the angle rate difference data sets. These looked awful! In some cases, the difference data sets looked like scaled down copies of the Ku MDM angle rate profiles. It was clear that these difference data sets would be of little value in resolving the problem. The real break in this case came when it was decided to compare the Ku MDM angle rate profile with the corresponding Ku MDM angle profile. This comparison of the pitch and pitch rate for H30SKAF is provided in Figure 3.6-1. A similar comparison for roll and roll rate is given in Figure 3.6-2. Now, since the earth's rotation rate is quite small, the ILOS angle rate integrated over a fixed interval should be equal to the total angle change over that same interval. Let's apply this rule to the data of Figures 3.6-1 and 3.6-2. Consider the time interval from 40 to 60 seconds. The average value of the roll rate is about -0.5 degrees/second and the average value of the pitch rate is about -0.35 degrees/seconds. Integrating over 20 seconds gives a total change of -10 degrees in roll and -7 degrees in pitch. Now, examining Figures 3.6-1 and 3.6-2 to determine the total angle change from the roll and pitch data, it is seen that the roll angle changes -5 degrees and the pitch angle changes about -3.5 degrees over the same 20 second interval. This tells us that either the angle or the angle rate is off by a factor of 2. But, since the angle data analysis showed no such problem, it can be assumed that the scale factor problem is in the angle rate data.

At this point, several questions come to mind. What is the value of the scale factor? Is it a constant? What is the source of the error? The answer to the first two questions were easy. Additional analysis of the same data showed that scale factor was about 2 for the entire interval. Analysis of other data sets showed the same factor. The only exceptions to this rule were the tests conducted after the k_5 gain in the servo was increased by a factor of 4. In that case, the scale factor was 0.5. (This problem is addressed at the end of this section.)

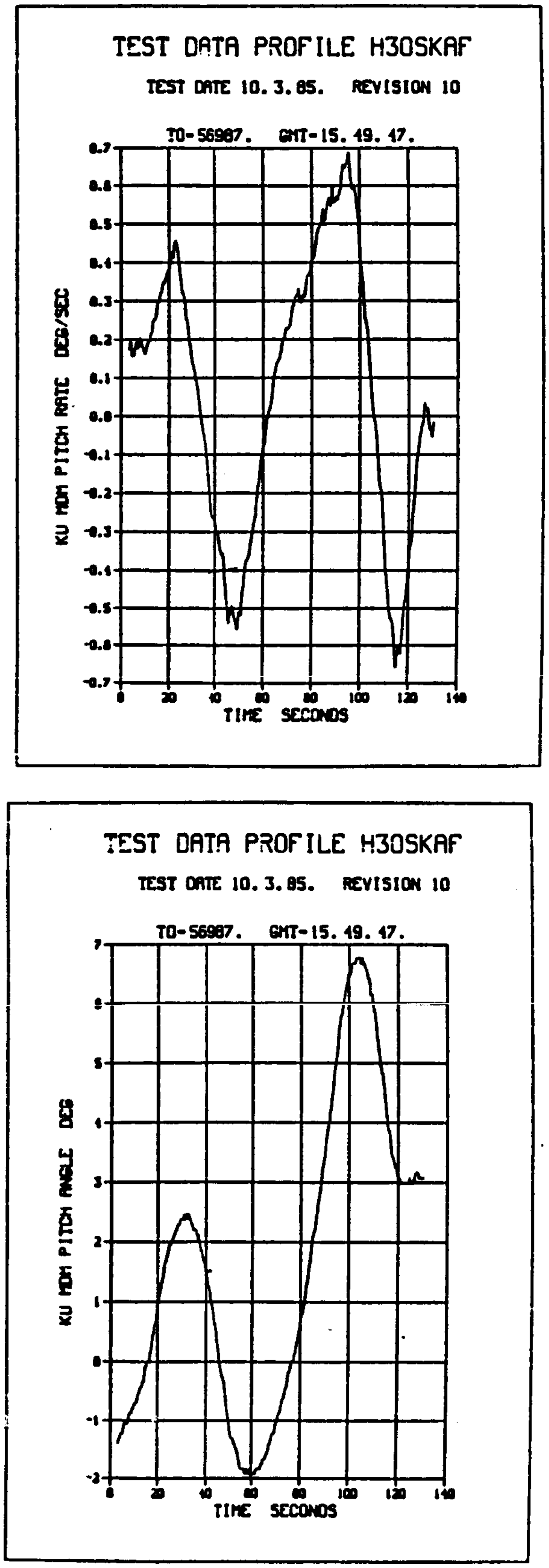


FIGURE 3.6-1 A COMPARISON OF THE KU MDM PITCH ANGLE AND ILOS PITCH RATE PROFILES FOR H30SKAF

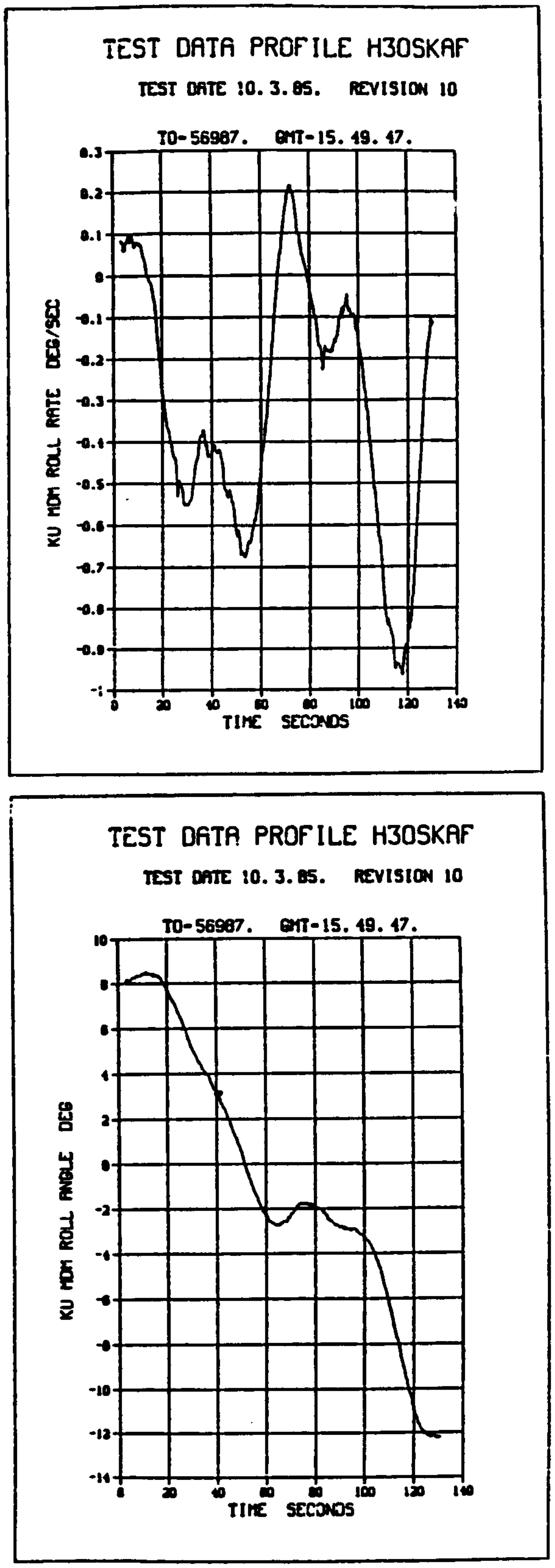


FIGURE 3.6-2 A COMPARISON OF THE KU MDM ROLL ANGLE AND ILOS ROLL RATE PROFILES FOR H30SKAF

What is the source of this scale factor error? There are two places where this scale factor can corrupt the angle rate: (1) in the Ku-Band Radar itself, after the azimuth and elevation angle rates are converted to roll and pitch angle rates in the EA-1 microprocessor, and (2) in the data processing sequence developed for the SORTE program. In either case, a factor of 2 seems quite reasonable since that represents a slip of a single bit in the binary representation of the angle rate value. At the writing of this final report, both possibilities are being pursued.

Regardless of what the error source turns out to be, the Ku MDM angle rate data will be scaled down by a factor of 2. The scaled data will then be analyzed for other problems that were masked by the scale factor problem.

3.6.2 Description of Angle Rate Error Sources

There are several sources that can corrupt the angle rate data besides the scale factor problem. Among these are:

o GDOP
o Angle Acceleration
o Weak Target Return Signals

A discussion of each of these is provided below.

GDOP. This error source will have the same affect on the angle rate as on the angle. However, we are not interested in wrestling with GDOP problems in the present analysis. Therefore, only the CINE reference data will be used in this analysis, since this system as configured is immune to GDOP.

Angle Acceleration. As will be demonstrated in the next section, this was the primary error source in the data examined, once GDOP was removed. The effect of angle acceleration on the ILOS angle rate tracking loop is identical to the acceleration effects on the angle tracking loop described in equations 3-11 and 3-12. That is, prolonged angle acceleration produces an asymptotic bias in the ILOS angle rate estimate. This can be ascertained from the following arguments.

Figure 3.6-3 illustrates the analog second order loop which is used to represent the ILOS angle rate tracking loop in the following analysis. The transfer function for this loop can be expressed as

(3-13) $$\hat{\dot{\theta}}(s)/\ \theta(s) = s\ w_n^2/(s^2 + w_n^2 Ts + w_n^2)$$

Since the loop is critically damped, then $T = 2/w_n$ where w_n is the natural frequency of the loop, and T is the loop settling time. To determine the response to angle acceleration we set

(3-14) $\theta(t) = At^2/2$ (angle position)

or $\dot{\theta}(t) = At$ (angle rate)

or $\ddot{\theta}(t) = A$ (angle acceleration)

where A is the angle acceleration and the $\dot{\theta}$ notation represents the derivative of the variable with respect to time. The Laplace transform of this quantity is

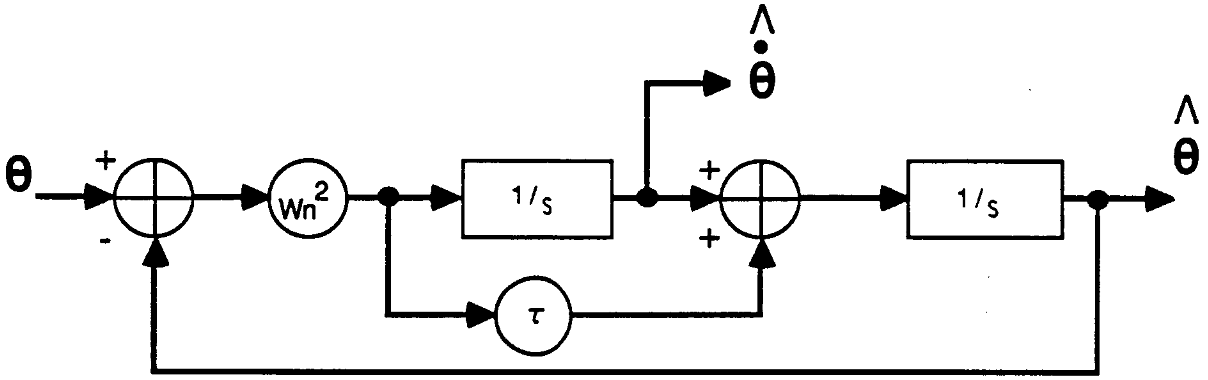
(3-15) $$\theta(s) = A/s^3$$

Using equations 3-13 and 3-15, the Laplace domain representation of the tracking loop response is

(3-16) $$\hat{\dot{\theta}}(s) = (A/s^2)(w_n^2/(s^2+2w_n s+w_n^2))$$

The inverse Laplace transform of 3-16 is

(3-17) $$\hat{\dot{\theta}}(t) = At(1+\exp(-w_n t) - (2A/w_n)(1-\exp(-w_n t))$$



NOTE: FOR A CRITICAL DAMPED LOOP $\tau = {}^{2}/Wn$

FIGURE 3.6-3 SECOND ORDER ANALOG MODEL OF THE ANGLE AND ANGLE RATE TRACKING LOOPS

To obtain the error in the angle rate estimate, the true angle rate =At is subtracted from equation 3-17. This gives

(3-18) $$\Delta\dot{\theta}(t) = -At \exp(-w_n t) + (2A/w_n)(1-\exp(-w_n t))$$

The asymptotic value is obtained by allowing t to approach infinity, which gives

(3-19) $$\Delta\dot{\theta} = 2A/w_n$$

Now, what sort of angle rate error does this expression produce for the Ku-Band angle rate tracking loop parameters? In the widest bandwidth case, $w_n=2\pi$ (0.12). If we consider an angle acceleration of 0.04 degrees per sec^2, this gives an angle rate bias of 0.11 degrees/sec. This is a significant amount of bias. For reference, Figure 3.6-4 shows the response of the angle and angle rate loops in the presence of a 0.04 deg/sec^2 constant acceleration.

Weak Target Return Signal. A weak target return signal produces a low SNR at the doppler filter output, which, in turn, produces noisy angle discriminants. These noisy angle discriminants get injected into the angle rate tracking loop filter which smooths the noise on the output angle rate estimate. The target return signal is usually the weakest at long range (greater than 40000 feet) where the angle rate tracker bandwidth is the narrowest. Now, the SNR_D threshold required to produce out-of-spec performance is estimated to be 7-8 dB as in the angle tracking case. In the present set of tests, this condition will only be achieved in some of the long range tests, e.g. some of the HL- and HJ- series tests. In general, weak target return signals should not be a problem.

TABLE 3.6-2 COMPARISON OF SCALED AND UNSCALED ROLL RATE DIFFERENCE DATA STATISTICS (Page 1 of 2)

PROFILE	MEAN		STANDARD DEVIATION	
	OLD	NEW	OLD	NEW
SAT2.CIN	1.40 E-3	-1.81 E-3	4.02 E-2	6.00 E-2
SAT2.BST	9.00 E-4	-7.55 E-5	4.95 E-2	7.68 E-2
SAT3.CIN	1.00 E-1	2.67 E-4	1.41 E-1	5.74 E-2
SAT3.BST	8.51 E-2	2.90 E-4	4.83 E-1	1.15 E-1
SAT4.CIN	-1.98 E-2	-3.38 E-3	3.71 E-1	2.52 E-2
SAT4.BST	-1.93 E-2	-4.00 E-3	3.82 E-2	2.89 E-2
H30SKAE.BST	2.33 E-1	1.38 E-2	1.10 E-1	4.92 E-2
H30SKAE.CIN	2.39 E-1	5.19 E-3	1.04 E-1	2.74 E-2
H30SKAF.BST	4.07 E-2	-1.16 E-2	1.91 E-1	8.45 E-2
H30SKAF.CIN	3.44 E-2	8.76 E-4	1.88 E-3	5.72 E-2
H30SKAG.BST	9.00 E-4	3.32 E-2	7.88 E-2	3.12 E-2
H30SKAG.CIN	5.33 E-2	9.13 E-3	1.05 E-1	6.47 E-2
H30SKAH.BST	6.41 E-2	6.49 E-3	1.44 E-1	7.48 E-2
H30SKAH.CIN	7.26 E-2	-1.25 E-3	1.41 E-1	6.31 E-2
H30SKAI.BST	1.02 E-1	2.35 E-3	2.16 E-2	4.33 E-2
H30SKAI.CIN	6.08 E-2	-4.16 E-3	8.88 E-2	4.27 E-2
HEL30AF.BST	1.04 E-1	5.39 E-3	9.13 E-2	1.99 E-2
HEL30AF.CIN	1.04 E-1	4.20 E-3	9.14 E-2	2.21 E-2
HEL30AG.BST	5.30 E-3	4.27 E-4	5.04 E-2	2.09 E-2
HEL30AG.CIN	6.00 E-3	-6.49 E-4	5.02 E-2	2.01 E-2
HEL30AI.BST	4.30 E-3	-6.15 E-3	4.07 E-2	2.13 E-2
HEL30AI.CIN	5.10 E-3	-7.81 E-3	4.09 E-2	1.41 E-2
HEL30AJ.BST	1.17 E-2	-1.21 E-3	5.91 E-2	3.59 E-2
HEL30AJ.CIN	1.31 E-2	-5.62 E-3	5.66 E-2	1.28 E-2

TABLE 3.6-2 COMPARISON OF SCALED AND UNSCALED ROLL RATE DIFFERENCE DATA STATISTICS (Page 2 of 2)

PROFILE	MEAN		STANDARD DEVIATION	
	OLD	NEW	OLD	NEW
HL246AD.BST	7.64 E-2	-6.35 E-3	1.15 E-2	8.87 E-3
HL246AD.CIN	7.57 E-2	-6.83 E-3	8.60 E-3	4.59 E-3
HL246AE.BST	6.30 E-2	-4.06 E-2	1.35 E-2	8.69 E-3
HL246AE.CIN	6.32 E-2	-4.53 E-3	1.35 E-2	4.55 E-3
HL346AD.BST	6.20 E-2	-4.29 E-3	1.14 E-2	1.11 E-2
HL346AD.CIN	6.34 E-2	-4.39 E-3	9.70 E-3	7.04 E-3
HL346AE.BST	7.55 E-2	-4.26 E-3	1.54 E-2	1.04 E-2
HL346AE.CIN	7.60 E-2	-5.01 E-3	1.52 E-2	8.20 E-3
HL446AC.BST	7.06 E-2	-5.95 E-3	1.05 E-2	1.65 E-2
HL446AC.CIN	6.90 E-2	-5.18 E-3	8.70 E-3	5.17 E-3
HL446AD.BST	6.08 E-2	-4.93 E-3	1.88 E-2	8.18 E-3
HL446AD.CIN	6.09 E-2	-5.05 E-3	1.87 E-2	4.92 E-3
HL546AE.BST	9.06 E-2	-4.36 E-3	1.30 E-2	8.55 E-3
HL546AE.CIN	9.07 E-2	-4.58 E-3	1.26 E-2	5.36 E-3
HL546AC.TMR	7.52 E-2	1.17 E-2	4.40 E-2	1.21 E-1
HL546AF.TMR	7.51 E-2	-5.05 E-3	1.07 E-2	6.87 E-3
HJ146AC.BST	6.00 E-2	-3.41 E-3	1.67 E-2	1.21 E-2
HJ146AC.CIN	6.34 E-2	-3.09 E-3	1.36 E-2	5.78 E-3
HJ146AD.BST	5.88 E-2	-5.04 E-3	1.43 E-2	7.74 E-3
HJ146AD.CIN	5.89 E-2	-5.19 E-3	1.41 E-2	6.61 E-3

TABLE 3.6-3 COMPARISON OF SCALED AND UNSCALED PITCH RATE DIFFERENCE DATA STATISTICS (Page 1 of 2)

PROFILE	MEAN		STANDARD DEVIATION	
	OLD	NEW	OLD	NEW
SAT2.CIN	2.49 E-2	-1.27 E-3	1.18 E-1	4.05 E-2
SAT2.BST	2.32 E-2	-5.67 E-4	1.04 E-1	2.91 E-2
SAT3.CIN	-7.40 E-3	-1.84 E-2	1.39 E-1	4.75 E-1
SAT3.BST	-8.20 E-3	-5.27 E-3	9.35 E-2	8.53 E-2
SAT4.CIN	3.13 E-2	-4.41 E-3	5.34 E-2	1.98 E-2
SAT4.BST	2.96 E-2	-4.46 E-3	5.15 E-2	1.91 E-2
H30SKAE.BST	-1.48 E-1	-2.07 E-2	6.11 E-2	3.65 E-2
H30SKAE.CIN	-1.65 E-1	-1.71 E-2	7.93 E-2	2.52 E-2
H30SKAF.BST	-1.67 E-1	-6.38 E-3	1.80 E-1	1.55 E-2
H30SKAF.CIN	-1.56 E-1	-1.10 E-3	1.69 E-1	5.58 E-2
H30SKAG.BST	-3.42 E-1	2.53 E-2	4.28 E-2	2.84 E-2
H30SKAG.CIN	-3.82 E-1	3.99 E-2	8.38 E-2	3.14 E-2
H30SKAH.BST	-1.54 E-1	-9.67 E-3	1.39 E-1	3.86 E-2
H30SKAH.CIN	-1.63 E-1	-5.25 E-3	1.28 E-1	3.17 E-2
H30SKAI.BST	1.84 E-2	-3.93 E-3	5.54 E-2	1.56 E-2
H30SKAI.CIN	-1.14 E-1	5.88 E-3	1.37 E-1	5.14 E-2
HEL30AF.BST	-6.73 E-3	-6.38 E-3	8.26 E-2	1.55 E-2
HEL30AF.CIN	-6.77 E-2	-5.95 E-3	8.47 E-2	1.61 E-2
HEL30AG.BST	-6.87 E-2	-2.77 E-3	9.36 E-2	1.31 E-2
HEL30AG.CIN	-7.14 E-2	-2.58 E-3	9.32 E-2	1.32 E-2
HEL30AI.BST	-3.16 E-2	-1.62 E-3	5.11 E-2	1.23 E-2
HEL30AI.CIN	-3.35 E-2	2.25 E-3	4.99 E-2	1.24 E-2
HEL30AJ.BST	-3.73 E-2	-5.70 E-4	7.69 E-2	1.52 E-2
HEL30AJ.CIN	-4.20 E-2	4.65 E-4	6.45 E-2	1.08 E-2

TABLE 3.6-3 COMPARISON OF SCALED AND UNSCALED PITCH RATE DIFFERENCE DATA STATISTICS (Page 2 of 2)

PROFILE	MEAN		STANDARD DEVIATION	
	OLD	NEW	OLD	NEW
HL246AD.BST	2.66 E-2	-2.26 E-3	1.76 E-2	5.19 E-3
HL246AD.CIN	2.35 E-2	-1.49 E-3	1.09 E-2	3.28 E-3
HL246AE.BST	2.32 E-2	-3.59 E-4	1.40 E-2	6.45 E-3
HL246AE.CIN	2.27 E-2	-1.71 E-4	1.18 E-2	5.99 E-3
HL346AD.BST	2.43 E-2	-3.04 E-3	1.55 E-2	7.50 E-3
HL346AD.CIN	2.24 E-2	-2.63 E-3	1.42 E-2	4.36 E-3
HL346AE.BST	2.66 E-2	4.61 E-4	1.86 E-2	7.63 E-2
HL346AE.CIN	2.59 E-2	9.35 E-4	1.72 E-2	6.96 E-3
HL446AC.BST	2.43 E-2	2.93 E-3	1.94 E-2	6.66 E-3
HL446AC.CIN	2.59 E-2	-3.43 E-3	1.13 E-2	4.07 E-3
HL446AD.BST	1.86 E-2	-1.06 E-4	1.23 E-2	7.96 E-3
HL446AD.CIN	1.85 E-2	-4.82 E-5	1.04 E-2	7.55 E-3
HL546AE.BST	3.15 E-2	-1.88 E-3	1.42 E-2	6.81 E-3
HL546AE.CIN	3.14 E-2	-1.76 E-3	1.23 E-2	6.09 E-3
HL546AC.TMR	4.31 E-2	-7.54 E-3	1.22 E-2	4.29 E-2
HL546AF.TMR	2.28 E-2	-7.91 E-5	1.16 E-2	5.39 E-3
HJ146AC.BST	2.02 E-2	-4.08 E-3	1.46 E-2	7.87 E-3
HJ146AC.CIN	2.10 E-2	-2.21 E-3	9.80 E-3	8.73 E-3
HJ146AD.BST	1.37 E-2	3.74 E-3	1.31 E-2	7.11 E-3
HJ146AD.CIN	1.35 E-2	5.29 E-4	1.25 E-2	6.94 E-3

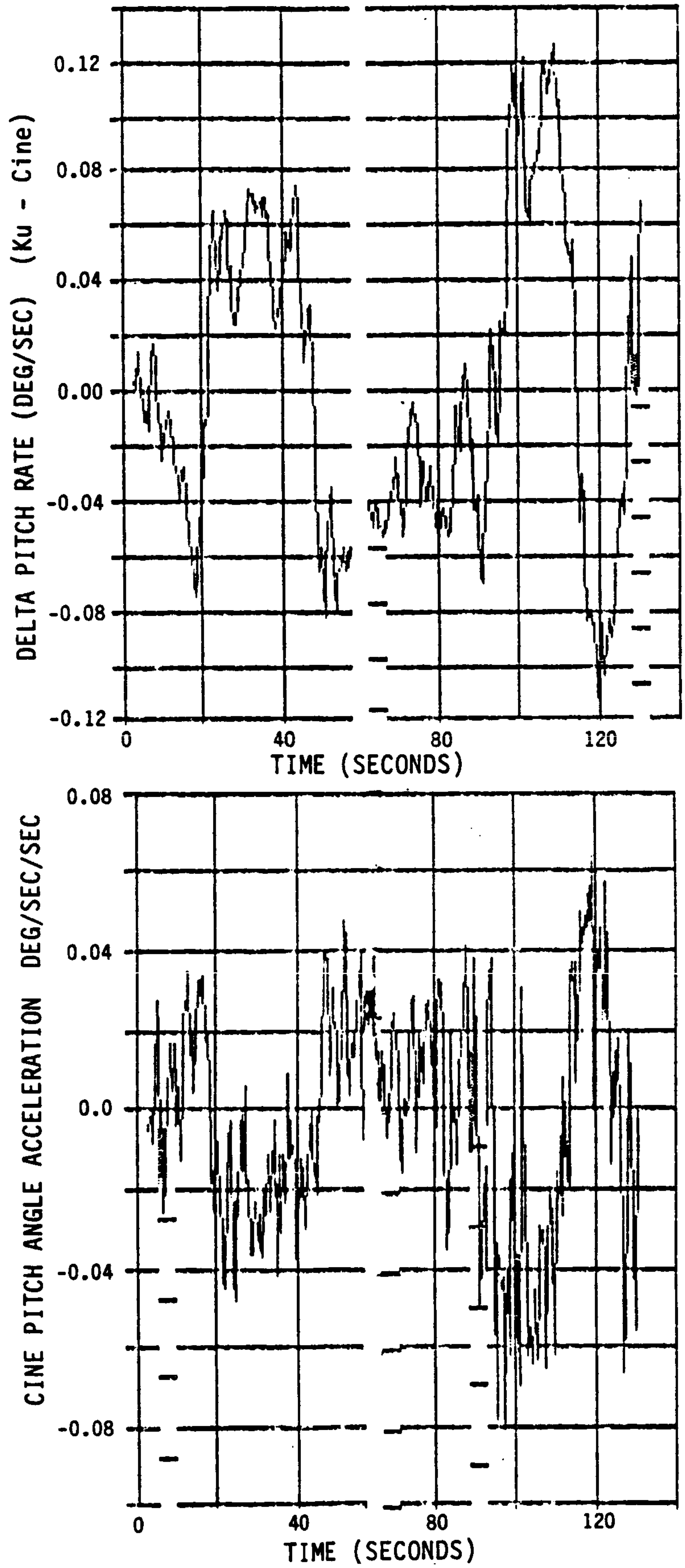


FIGURE 3.6-5 A COMPARISON OF THE CINE PITCH ANGLE ACCELERATION PROFILE AND THE CINE PITCH RATE DIFFERENCE DATA PROFILE FOR H30SKAf

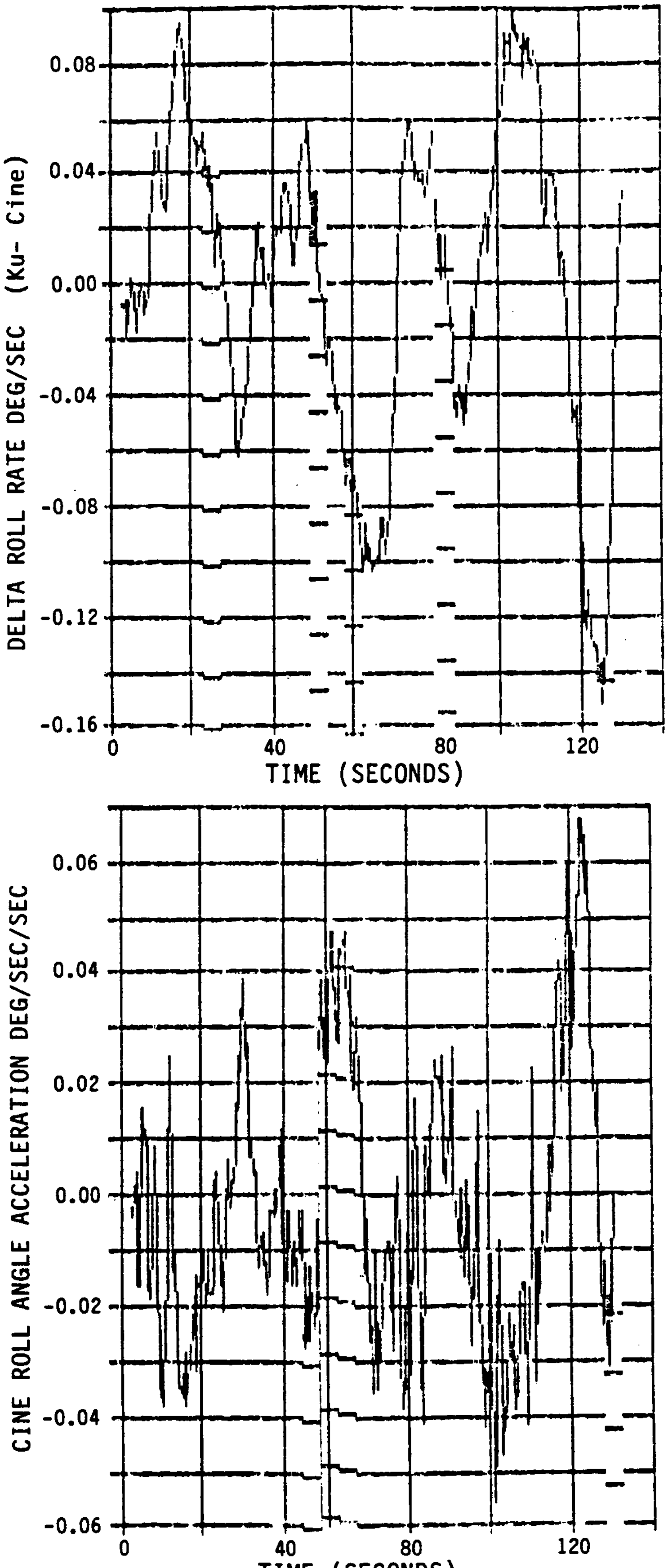


FIGURE 3.6-6 A COMPARISON OF THE CINE ROLL ANGLE ACCELERATION PROFILE AND THE CINE ROLL RATE DIFFERENCE DATA PROFILE FOR H30SKAF

performed with the widest angle tracking noise bandwidth. However, the HEL30 and H30SK series were performed at ranges of 2000 to 12000 offset in X and Y from the radar while SAT tests were over a range interval of 2500 feet to 1200 feet directly over the radar. The difference here is that slight wind disturbances in the SAT test configuration translate into reasonably large angle accelerations that produce momentary biases. These biases, in turn, produce large standard deviations in the difference data. This phenomenon will be examined in detail in the next subseciton.

3.6.3.1 Acceleration Effects. In this case it turns out that acceleration is the primary source of error in the angle rate difference data once the scale factor of two has been removed. Figure 3.6-5 compares the pitch angle acceleration profile against the pitch rate difference data profile. Figure 3.6-6 gives a similar comparison for roll angle. Observe that the angle acceleration profile shape and the angle rate difference profile shape are highly correlated for both roll and pitch. Next, it will be demonstrated that not only are the shapes highly correlated, but that they are related by the expression given in equation 3-18 or 3-19.

Consider the interval 20 to 40 seconds in the pitch data of Figure 3.6-5. The average pitch angle acceleration during this period is -0.02 degrees/sec/sec. Using equation 3-19, the corresponding average pitch angle rate bias error is computed as 0.053 degrees/sec. This value agrees quite well with the pitch rate difference data in the same 20 to 40 second time interval.

Consider a similar calculation for the roll rate for the time interval 100 to 110 seconds. The average acceleration in this case is about -0.04 degrees/sec/sec and the computed roll rate bias error is 0.11 degrees/second. The average roll rate error taken for the same interval from the roll rate difference profile is about 0.09 degrees/second. Hence, the calculated data and the measured data agree reasonable well.

The conclusion from the above discussion is that the primary error source in the H30SKAF case is angle acceleration. Although the other flights must be evaluated on a case-by-case basis to determine the dominant error source, one can draw an additional conclusion from the above data analysis. It was observed that very small angle accelerations, i.e. acceleration less than 0.04 deg/sec/sec produced angle rate biases of 0.11 deg/sec which is 40 times the specification on the standard deviation. Based on these numbers it is reasonable to conclude that the primary error source in the other tests will be angle acceleration as well. At shorter ranges, the same wind turbulence will cause larger angle rate errors and at longer ranges the reverse will be true.

Another important conclusion that can be drawn from this comparison is that the model shown in Figure 3.6-4 is an accurate representation of the angle and angle rate tracking loop. Furthermore, it shows that the actual bandwidths of these tracking loops (which is related to the natural frequency f_n of the loop) matches the intended design bandwidth values. This is verified by the matching of the angle acceleration and the angle rate difference data through the relation 3-19 and the matching of the angle acceleration and the angle difference data through the equation 3-12. Both of these expressions contain w_n which is the natural radiam frequency of the loop.

Reflecting upon the comments above, it may well be that the problems with the angle rate tracker at close range during a space flight rendezvous are related to very slight angle accelerations of the target. A target acceleration of 0.01 deg/sec/sec causes an angle rate bias that is 10 times greater than the standard deviation specification. It is not known whether 0.01 deg/sec/sec angle acceleration is typically encountered in the shuttle-satellite rendezvous. However, it is recommended that radar data from some typical rendezvous be analyzed for acceleration bias problems. If this turns out to be the problem, it casts a new light on potential solutions to the angle rate tracking loop.

Before leaving this subsection, there is some additional evidence that lends additional support to the angle acceleration theory. The intent here is to demonstrate that the bias found in the pitch and roll angle difference data is consistent with the magnitude of angle acceleration given in the plots of Figures 3.6-5 and 3.6-6. Pitch angle difference data and roll

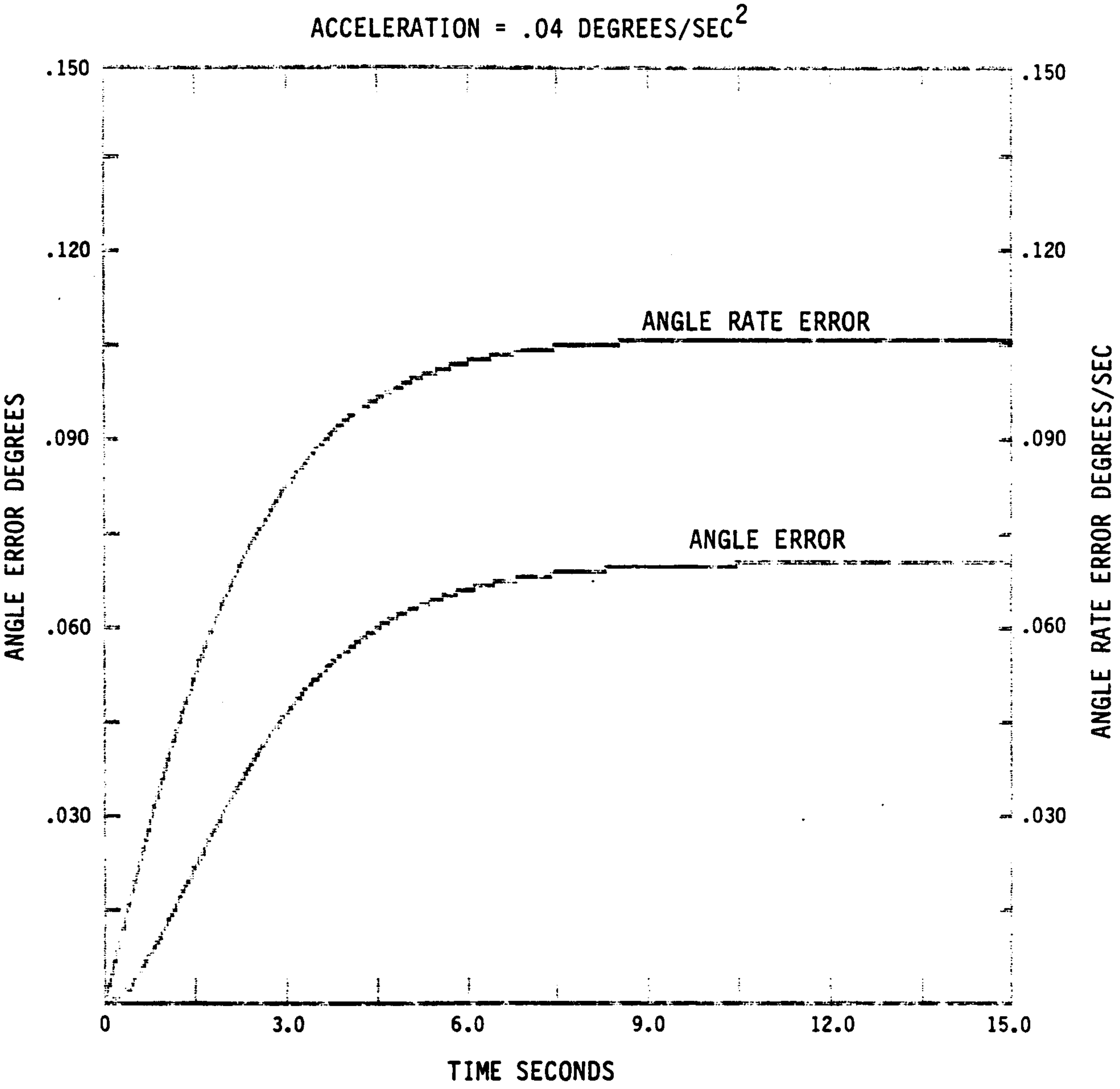
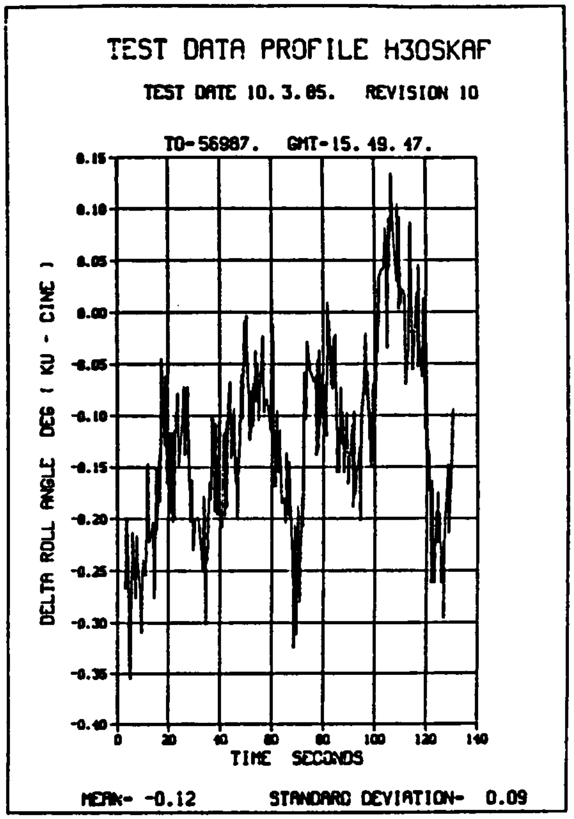


FIGURE 3.6-4 ANGLE AND ANGLE RATE ERROR DUE TO AN ACCELERATION OF 0.04 DEGREES/SEC



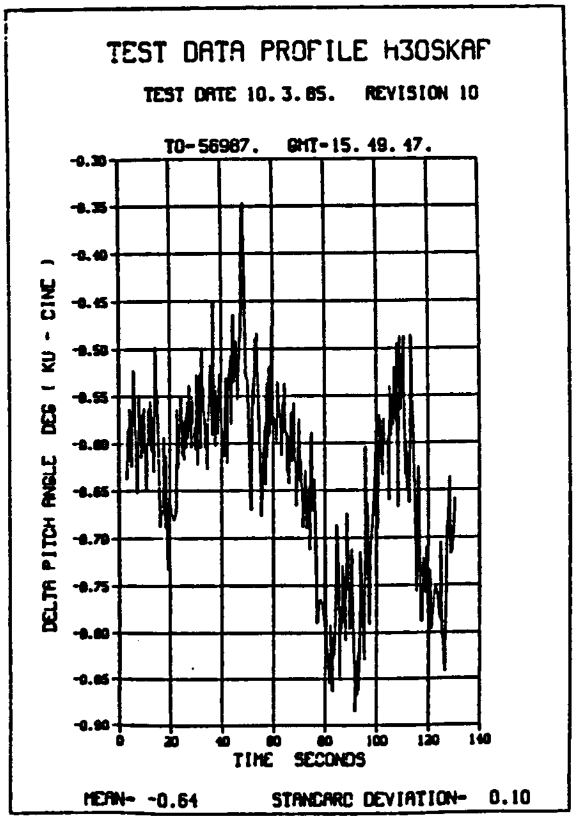
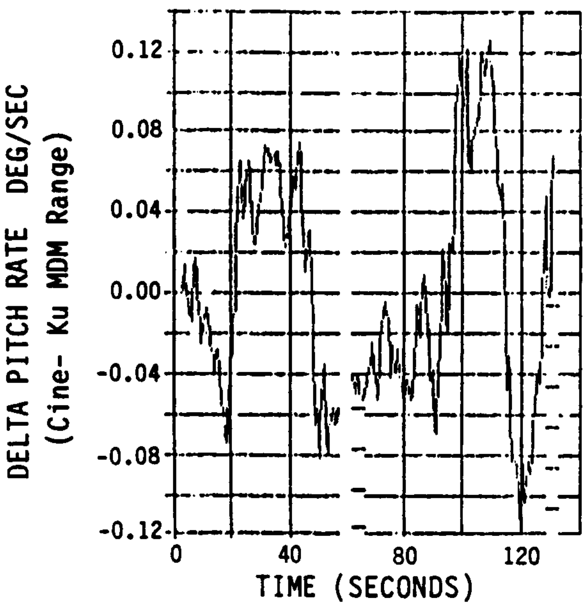


FIGURE 3.6-7 CINE ROLL AND PITCH ANGLE DIFFERENCE DATA PROFILE FOR H30SKAF



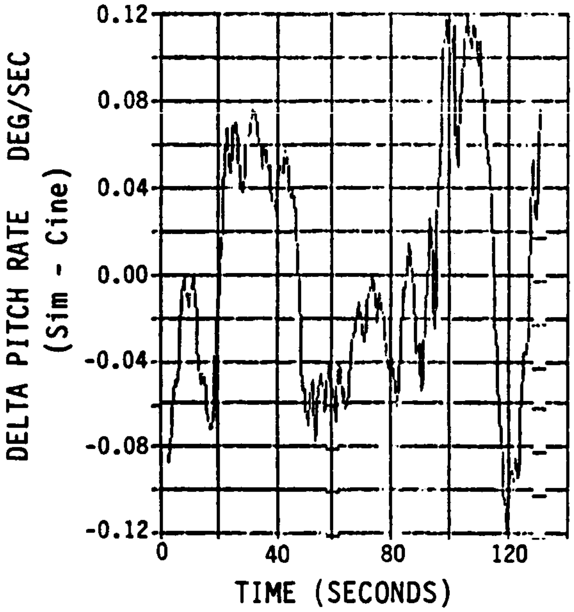


FIGURE 3.6-8 A COMPARISON OF THE CINE KU PITCH RATE DIFFERENCE DATA AND THE CINE SIM PITCH RATE DIFFERENCE DATA FOR H30SKAF

3.6.3 SORTE Angle Rate Data Analysis

Since the scale factor problem was discovered near the end of the contract performance period, only limited analysis of the angle rate data could be done. This analysis consists of (1) recomputing all of the means and standard deviations of the roll and pitch angle rate data, excluding the November 4, 1985 flights due to the servo gain change, and (2) performing an in-depth analysis of a single flight (H30SKAF).

Table 3.6-2 compares the mean and standard deviation of the roll rate difference data, generated from the rescaled roll rate data, to the mean and standard deviation of the original roll rate difference data. Table 3.6-3 gives a similar comparison for the pitch rate data. Observe that both the means and the standard deviations of these difference data improve by at least a factor of two in every case. A comparison with the specification reveals that many of the mean values are within specification and most of the rest of the mean values are very close to the spec limit.

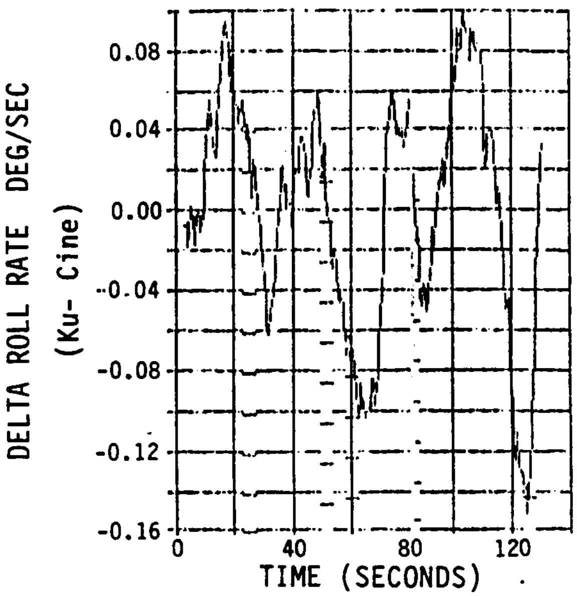
There are some general observations that can be made concerning the recomputed standard deviation values. Firstly, every value of standard deviation is still outside the specification limit and in only a few the values are just slightly outside the limit. This fact is still alarming. However, analysis of a sample case will demonstrate the source of error for many of these cases. Secondly, a comparison of the standard deviations for the various flight series is illuminating and encouraging. The performance of these flight series can order from best to worst as follows:

Best Performance:	HL- and HJ- series
Intermediate Performance:	H30SK- and HEL30- series
Worst Performance:	SAT series

This ordering is quite reasonable. The HL- and HJ- series should give the best performance for two reasons: (1) they were performed at long range with the narrowest angle tracking noise bandwidth and (2) at long range the angle accelerations are reduced. The HEL30- H30SK-, and SAT- series all were angle difference data for H30SKAF are provided in Figure 3.6-7 for reference. Consider the pitch angle acceleration data time interval of 115 seconds to 125 seconds. In this interval the average acceleration is about 0.05 deg/sec/sec. Using equation 3-12, the angle bias error is computed as 0.088 degrees. If this is added to the pitch angle difference mean shown in Figure 3.6-7, then the total predicted angle error is -0.728 degrees. The pitch angle difference data of Figure 3.6-7 shows an average error of about -0.75 degrees for the same time interval. Hence, the pitch angle acceleration profile agrees with the pitch angle difference profile as well as the pitch angle rate difference profile.

Let's also do a calculation for the roll angle. Consider the time interval 120 to 125 seconds. The average roll angle acceleration is 0.05 deg/sec/sec and the calculated roll angle bias error is 0.088 degrees. Adding this to the mean error of the roll angle difference profile, the total computed average roll angle error for the time interval is 0.208 degrees. A review of the measured roll angle difference data for the same time period shows an average roll angle bias error of -0.22 to -0.225 degrees. Again the roll angle acceleration profile agrees with both the roll rate difference profile and the roll angle difference profile. Hence, the data seems consistent among the three variables for both roll and pitch.

Simulation Verification. As further proof that the scale factor of 2 should be removed and that acceleration is the major contributor to the angle rate errors, the H30SKAF CINE profile was injected into the final version of the simulation and angle rate and angle difference data was generated. Figure 3.6-8 compares the Ku-Band pitch angle rate difference data to the simulation pitch angle rate difference data (both sets are referenced to the CINE data), for the H30SKAF profile. Figure 3.6-9 gives a similar set for the roll angle rate difference data. These comparisons show that the simulation accurately reflects the angle rate response of the Ku-Band (at least for the present flight profile). It shows that the acceleration errors appear in the simulation response and are of the same magnitude. It also shows that there is no scale factor problem between the simulation and the modified measured angle rate data.



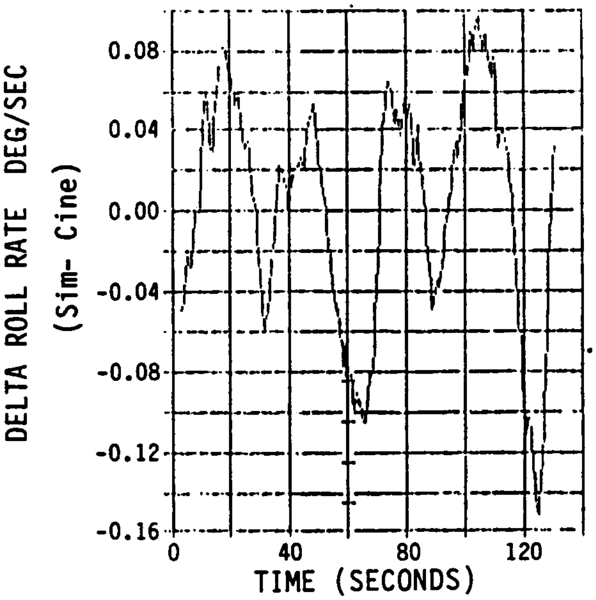


FIGURE 3.6-9 A COMPARISON OF THE CINE KU ROLL RATE DIFFERENCE DATA AND THE CINE SIM ROLL RATE DIFFERENCE DATA FOR H30SKAF

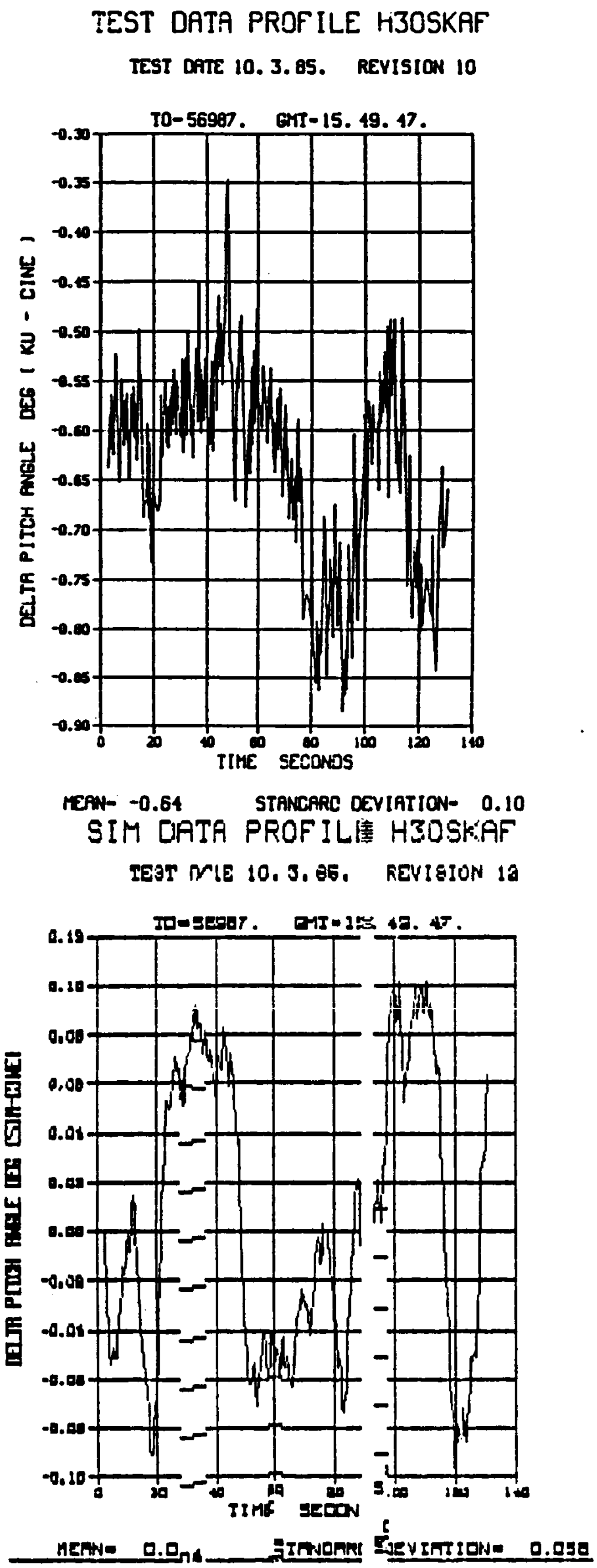


FIGURE 3.6-10 A COMPARISON OF THE CINE KU-BAND PITCH ANGLE DIFFERENCE DATA AND THE CINE SIM PITCH ANGLE DIFFERENCE DATA FOR H30SKAF

Figure 3.6-10 compares the CINE Ku pitch angle difference data and the CINE sim pitch angle difference data for the H30SKAF flight profile. A similar comparison for the roll angle difference data is provided in Figure 3.6-11. A first impression is that the data does not match as well as the angle rate comparison. However, it should be noted that the sim data is not quantized to 0.1 degrees as it is in the Ku-Band Radar. A quantized version of the simulation data would probably show a better fit.

3.6.4 A Discussion of the Servo Experiment

Once the scale factor problem in the angle rate was discovered in one of the sets of data, all of the data sets were scrutinized to determine whether the scale factor was the same for all cases. LEMSCO personnel discovered that the data from the 4 November 1985 flights had a scale factor of 0.5, rather than a factor of 2. All of these flight tests were flown with an increase by a factor of 4 in the k_5 gain of the angle tracking loop. Thus, it became clear that the output angle rate is scaled inversely with the change in the k_5 gain. What follows is a derivation of this fact.

Figure 3.6-12 gives an equivalent second order analog model representation of the angle rate tracking loop modified to include the k_5 gain (compare this configuration with the model in Figure 3.6-4).

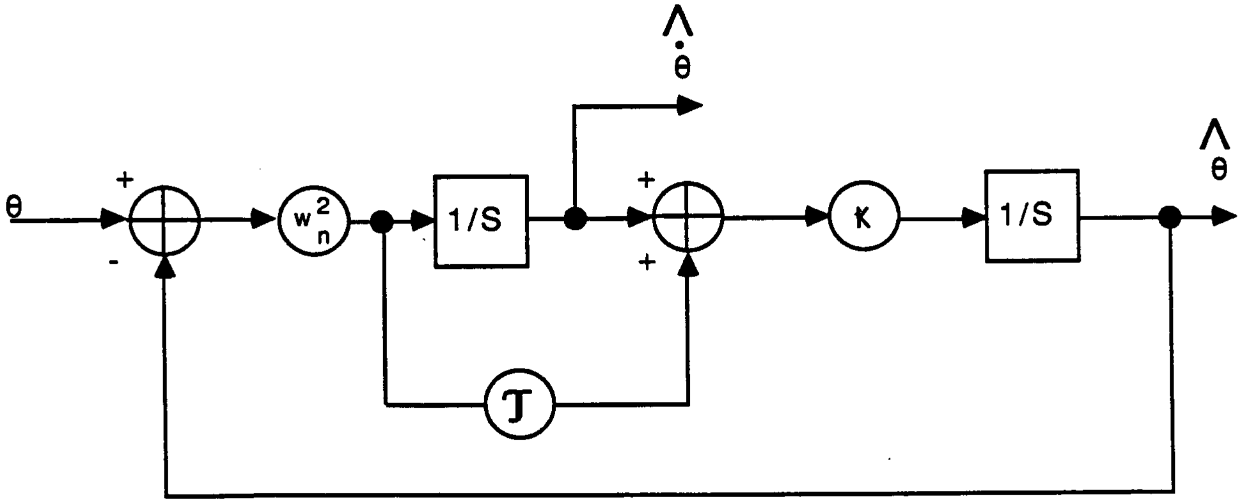


FIGURE 3.6-12 SECOND ORDER ANALOG MODEL OF THE MODIFIED ANGLE RATE TRACKING LOOP

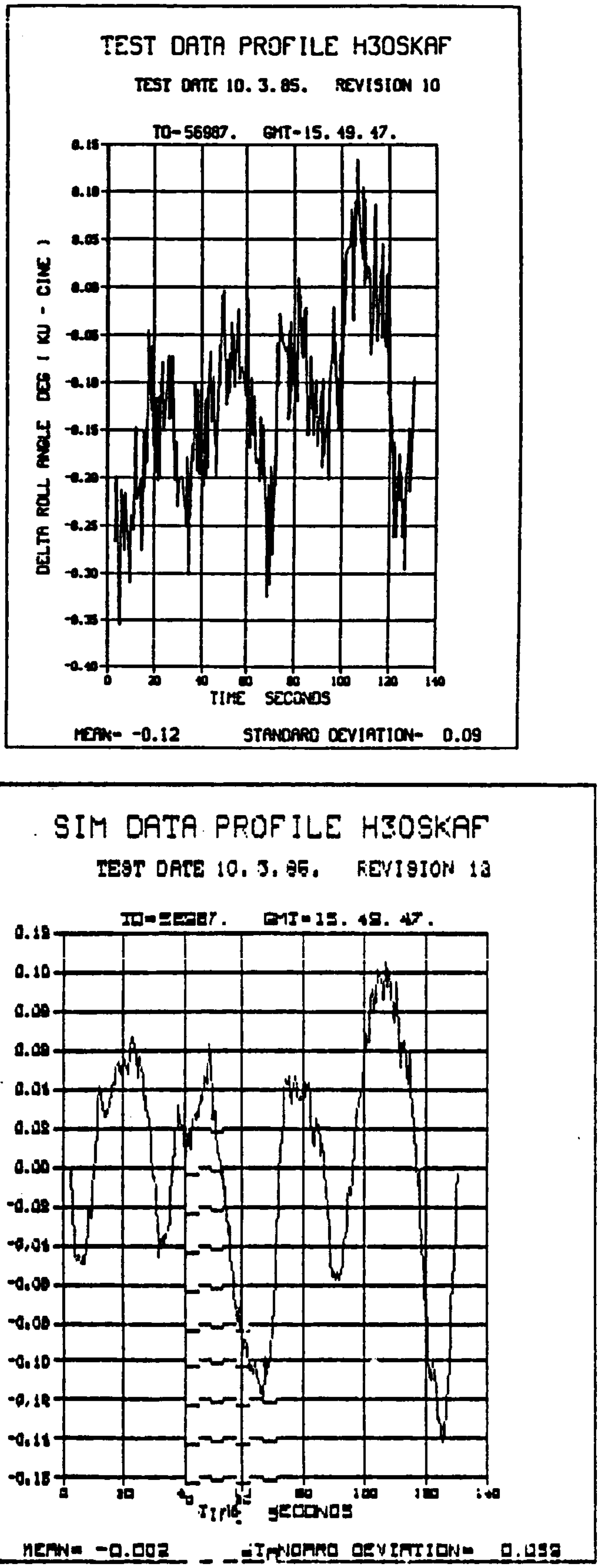


FIGURE 3.6-11 A COMPARISON OF THE CINE KU-BAND ROLL ANGLE DIFFERENCE DATA AND THE CINE SIM PITCH ANGLE DIFFERENCE DATA FOR H30SKAF

The transfer function for this loop is given by the expression

$$\hat{\dot{\theta}}(s)/\,\theta(s) = sw_n{}^2/(s^2+2Kw_ns+Kw_n{}^2) \tag{3-20}$$

where w_n has been defined previously and K is associated with the k_5 constant and represents the gain inserted into the Ku-Band Radar servo electronics during the SORTE program at WSMR. For the normal operational design of the radar K = 1, but in the SORTE experiments the gain K was raised to a value of 4.

It is now demonstrated that the angle rate output is divided down by a factor of K. This fact is most easily demonstrated by choosing a particular input. The input chosen is a ramp in angle or a step in angle rate.

$$\theta(t) = At \quad \text{(angle position)} \tag{3-21}$$

$$\dot{\theta}(t) = A \quad \text{(angle rate)}$$

Where A is the value of the angle rate and should be the value output by the tracking loop. The s-domain representation of the ramp is

$$\theta(s) = A/s^2$$

and the s-domain response to this input is

$$\hat{\dot{\theta}}(s) = (A/s^2)((\ sw_n^2\ )/(s^2 +2\ Kw_nD+Kw_n^2\ )) \tag{3-22}$$

To determine the steady-state response to this input, apply the final value theorem from control theory. The final value theorem is

$$\hat{\dot{\theta}} = \lim_{s \to o} s\ (\ \hat{\dot{\theta}}\ (s)) \tag{3-23}$$

and the result is

$$\hat{\dot{\theta}} = A/K. \tag{3-24}$$

This shows that the output of the loop is the true angle rate divided by K. So when K is 1 as in the operational design, angle rate meters give a true indication of the target's ILOS angle rate. However, when K is different from 1 then the angle rate meters give a scaled version of the true angle rate.

The above result is consistent with the data from the SORTE program servo experiments (disregarding the scale factor of 2). The implication is, that to determine the true angle rate noise performance with the increased gain, one must scale the angle rate data by the gain K prior to determining the noise properties. When this is done with the present data it is found that the noise performance does not improve, but degrades, when increasing the k_5 gain.

If the problem with the angle rate loop in flight is too much random noise, then the noise bandwidth of the loop should be decreased. Since the noice bandwith is proportional to k_5, the value of this gain should be decreased to reduce the noise bandwidth. However, if the problem with the angle rate performance in flight is related to biases induced by fluctuating angle acceleration, then the solution to the problem is much more difficult. The bias in angle rate due to acceleration can be shown to be proportional to τ (see Figure 3.6-12); therefore, to decrease the bias due to acceleration, τ must somehow be reduced. A change in the gain k_5 will not change the bias in angle rate.

The real solution to the angle acceleration problem is to use a third order loop (commonly called an alpha-beta-gamma filter). This type of loop will not suffer from angle rate bias in the presence of a constant angle acceleration.

(THIS PAGE INTENTIONALLY LEFT BLANK)

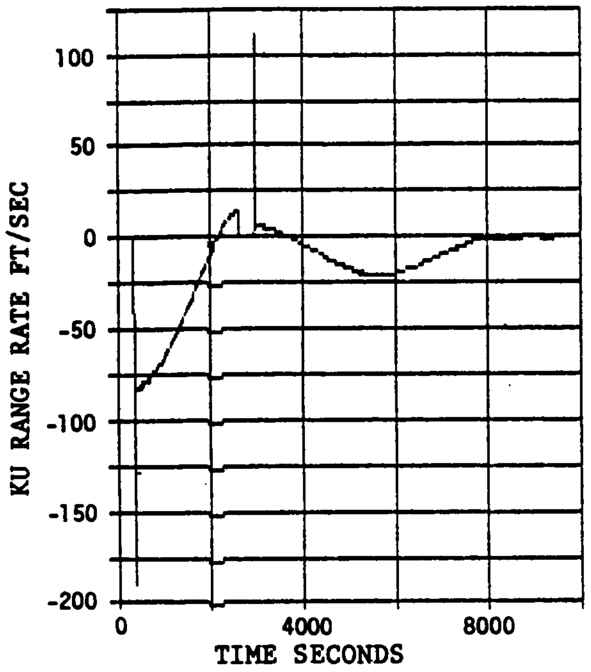
4.0 PALAPA MISSION DATA ANALYSIS

The purpose of this section is to provide analysis of some space flight rendezvous data. The particular set of data supplied for this exercise was the radar data from the space shuttle rendezvous with the Palapa B Satellite during mission 51A in November 1984. The summary of the analysis is done on three different levels. Firstly, a general qualitative discussion is presented to point out all significant features in the data. Secondly, some limited quantitative analysis of the data is given to provide the reader with a feel for the radar errors encountered in an operational environment. Thirdly, the results of injecting the smoothed Palapa profile into the simulation are compared to the actual data. In addition, the validity of this simulation technique is discussed.

4.1 QUALITATIVE DISCUSSION OF THE DATA

Excluding the Radar Signal Strength (RSS), there are six basic target parameters that the Ku-Band Radar tracks during a rendezvous: range, range rate, roll and pitch angles and inertial line of sight (ILOS) roll rate and pitch rate. Figure 4.1-1 shows the range and range rate data for the entire rendezvous which was approximately 9000 seconds in duration. Figure 4.1-2 gives a similar plot for the roll and pitch angle data, and Figure 4.1-3 gives the data for ILOS roll and pitch rate. Some general qualitative observations about these data follow.

The range and range rate data of Figure 4.1-1 looks very well behaved (at least on the scale shown in the figure). It will be shown in the next section that the random component of these data are well within specification for these time intervals, corresponding to three different range tracker bandwidths. Also, observe that the glitches in the data in the intervals 0 to 400 seconds and 2500 to 3000 seconds are not caused by the radar, but instead are missing data due to data link drop-out or some other communication link problem.



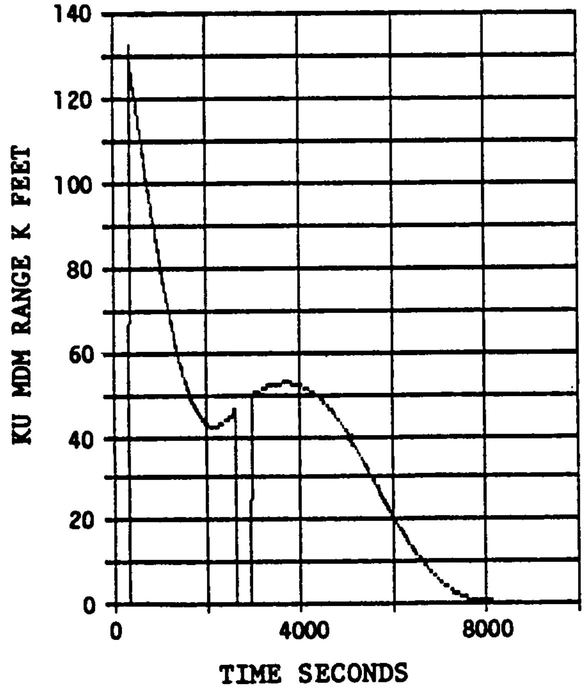
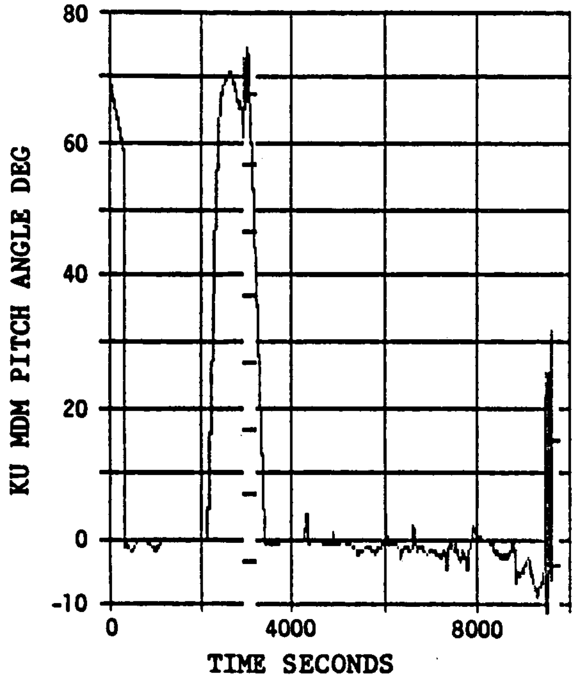


FIGURE 4.1-1 KU-BAND RADAR RANGE AND RANGE RATE PROFILES FOR THE RENDEZVOUS WITH THE PALAPA SATELLITE



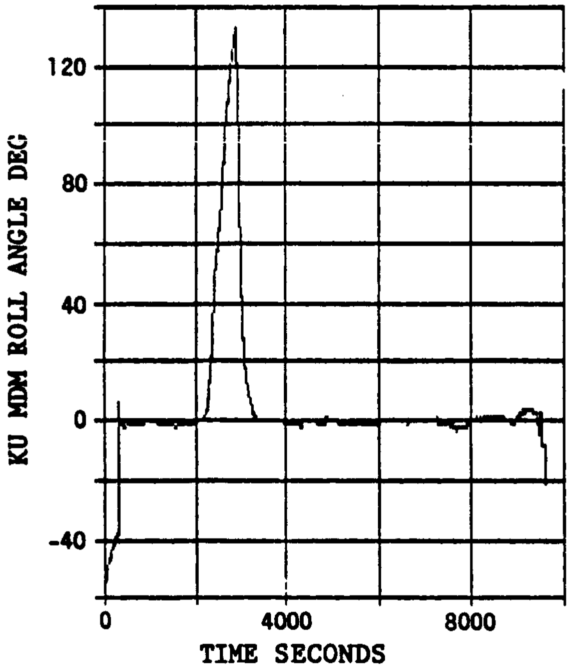
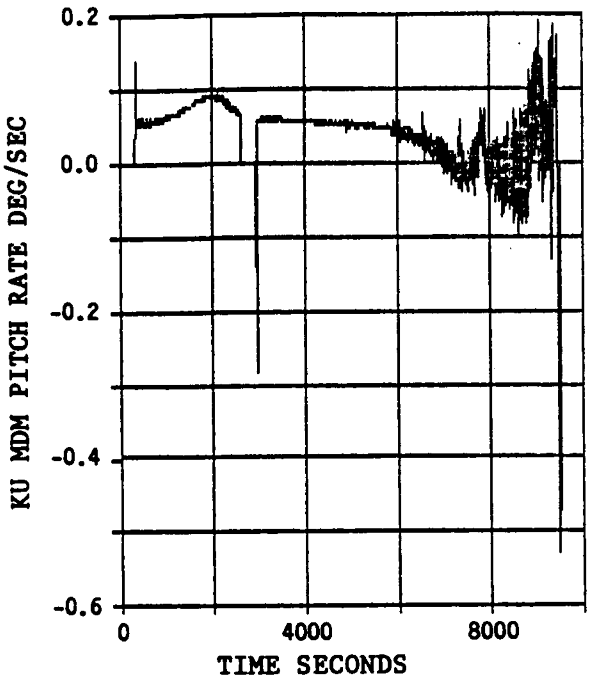


FIGURE 4.1-2 KU-BAND RADAR PITCH AND ROLL ANGLE PROFILES FOR THE RENDEZVOUS WITH THE PALAPA SATELLITE



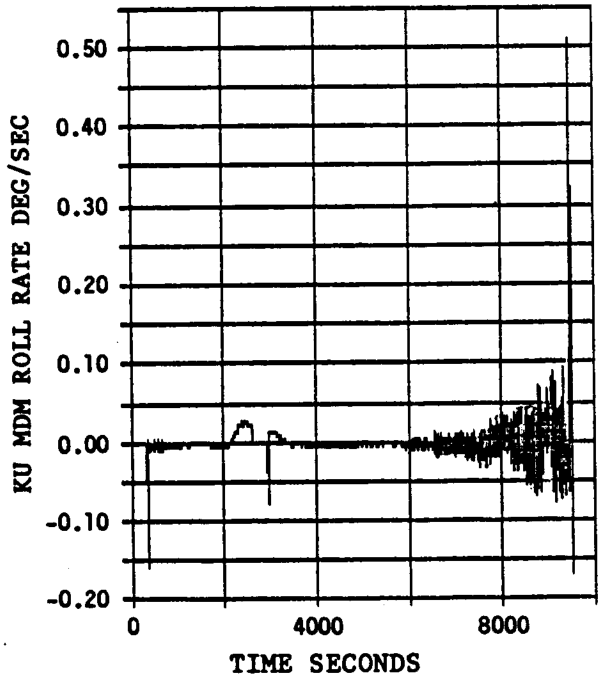


FIGURE 4.1-3 KU-BAND RADAR ILOS PITCH AND ROLL RATE PROFILES FOR THE RENDEZVOUS WITH THE PALAPA SATELLITE

Except for a few time intervals, the roll and pitch angle data is very near zero for the entire rendezvous. This just means that the shuttle and the target are coming together along the -Z axis of the Shuttle Body Coordinate System. The most prominent features in these data is the large angular change in the data over the time interval 2200 seconds to 3200 seconds. This corresponds to an intentional change in the Orbiter's attitude and preparation for what is known as a TI burn. This injects the Orbiter into the final phase of the rendezvous. Also notice that there is some nonzero angular positions in the time after 5500 seconds. During this time, the Orbiter is performing several small "hops" to move toward the target. In summary, the data is well-behaved and, as shown in the next section, the random component is well within specification for both the roll and pitch angle in all three bandwidth intervals.

The ILOS roll and pitch rate data of Figure 4.1-3 has some interesting features. First, the glitch in the data over the interval 2500 to 3000 seconds is caused by data link drop-out as in the range and range rate case. The hump in the roll rate data from 2000 to 3500 seconds is associated with the TI burn maneuver, but the mechanism producing it cannot be stated for certain. It could be caused by true target inertial angle rate or, it could be that the body rate during the maneuver was not compensated perfectly. Similar comments apply to the pitch rate data over this time interval.

The next significant feature that can be picked up from these data are the bandwidth switch points, especially in the roll rate data. These switch points are marked by a noticeable step increase in the "random" component of the data. The first switch point (which is the hardest to see on the scale of the data) occurs at range 23,030 feet and approximately 6000 seconds. The second switch point is quite prominent and occurs at a 11,510 feet and approximately 6500 seconds. For a long time, this increase in the random component was solely due to the increase in tracker noise bandwidth when the bandwith is switched. However, based on the analysis of the SORTE test data, it is now felt that a significant part is due to very slight inertial angle accelerations and the angle rate biases induced by these accelerations. Also observe that these angle rates can also be produced by beam wander on the target, especially for ranges less than 1000 feet.

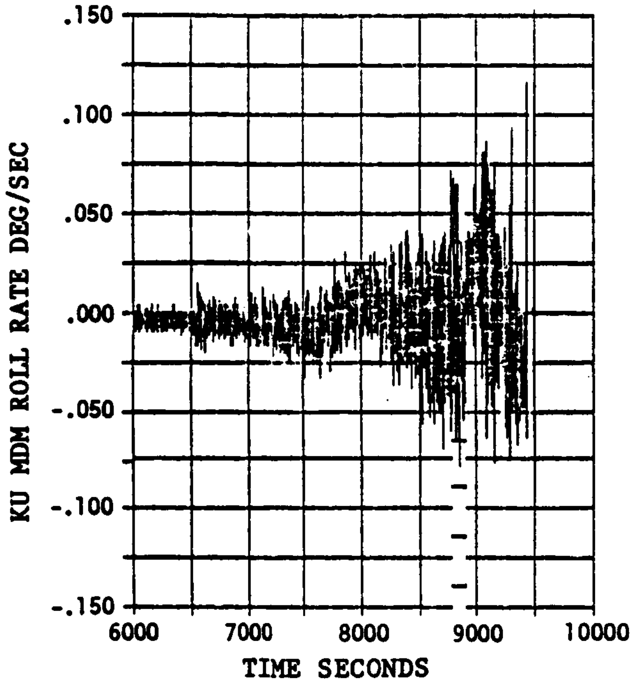
Another feature of the data is that the envelope of the random component in roll and pitch rate appears to grow from time 6500 seconds to 9500 seconds as the range decreases into 100 feet. Figure 4.1-4 gives an expanded view of this envelope for roll and pitch rates. This observation supports the statements of the preceding paragraph. If the fluctuation in the data were caused by thermal noise, then the random component would certainly not grow with decreasing range and increasing target signal strength. On the other hand, problems with actual inertial cross line-of-sight movement (producing angle acceleration) would increase with decreasing range, and problems with beam wander on the target would also increase with decreasing range. Neither of these problems can be controlled with adjustments in the angle rate tracking loop parameters.

Figure 4.1-5 gives an even more expanded view of the roll rate data for the time interval 8000 to 8100 seconds. A qualitative observation about this data is that it appears to have a less random or more deterministic character to it. It is more oscillatory in nature. (Spectral analysis of the data would verify this statement.)

The final observation concerns the pitch rate data. The significant bias seen in the data is due to the orbital rate. That is, the shuttle orbits the earth approximately every 90 minutes. This produces a rate of 0.067 degrees per second and corresponds perfectly to the pitch rate bias. This is reasonable since pitch is the angular movement in the plane of the orbit due to the attitude of the shuttle during the rendezvous.

4.2 SOME SIMPLE QUANTITATIVE DATA ANALYSIS

To perform an accurate quantitative analysis of the Ku-Band Radar requires accurate reference data. That is, data generated by an independent sensor or set of sensors whose measurement accuracies are as good or better than the Ku-Band Radar. The purpose of the SORTE program was to provide such a reference and, from this data, develop some quantitative estimates of radar performance. However, the SORTE program experiments could not exactly duplicate space flight conditions. Hence, a quantitative analysis of the Palapa flight data was undertaken using a psuedo-reference. The psuedo



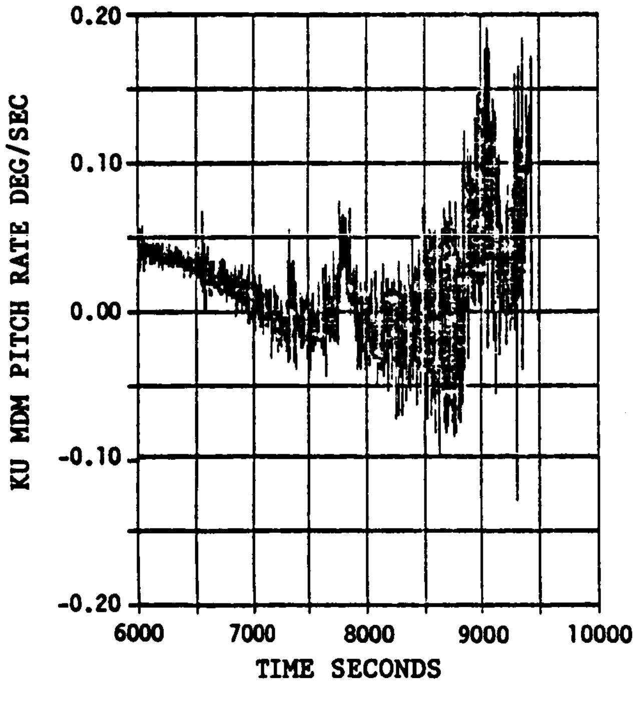


FIGURE 4.1-4 EXPANDED VIEW OF ROLL AND PITCH RATE PROFILES FOR THE PALAPA RENDEZVOUS

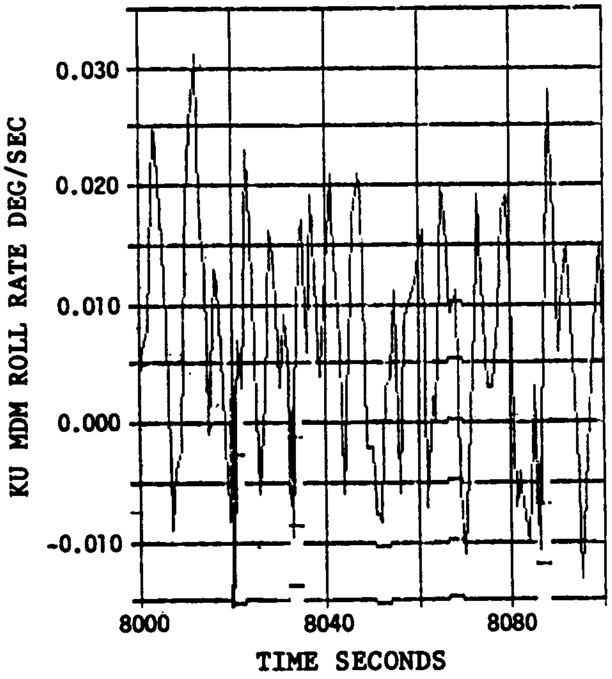


FIGURE 4.1-5 EXPANDED VIEW OF ROLL RATE DATA FOR THE PALAPA RENDEZVOUS ILLUSTRATING THE FINE STRUCTURE OF THE DATA

reference generation and the dangers associated with it are discussed in the next subsection. Results of the data analysis are provided in the subsection following the reference discussion.

4.2.1 Reference Data Generation

Assumptions. Generation of the reference data set was accomplished by making the following assumptions. First, it is assumed that the average of the radar parameter estimates over short intervals (10-50 seconds) are bias-free and represent the target's true parameter average in that interval. There is one significant drawback in this assumption: prolonged range and angle acceleration produce significant biases in range, range rate, angle and angle rate. To alleviate this problem to some extent, an analysis of the parameter bias error was ignored in the present exercise.

The second assumption is that the fluctuations in the data over small intervals is due to radar thermal or quantization noise. Hence, these features were eliminated when forming the reference. The danger in doing this was not discovered until after the fact, during SORTE data analysis. These so-called random fluctuations, especially in the angle rates, may be induced by the shuttle - Palapa rendezvous dynamics. In the case of the angle rate, for example, significant short-term angle rate bias could be induced by slight angle accelerations due to flight control adjustments by the shuttle pilot. These short-term biases on a larger time scale appear to have a random nature and were removed for the data analysis reported below. It is now believed that the discrepancy between the simulation angle rate and the flight angle rate data is due to removing true fluctuations in the angle rate data.

Method. Ths basic method for developing a data reference was to smooth the radar flight data using a short-term averaging technique. The technique was moving window averaging and can be represented by the following expression:

$$(4\text{-}1) \qquad P_R(n) = \sum_{j = -(N-1)/2}^{(N-1)/2} P(n+j)/N$$

where P_R is the nth reference value for the parameter P

$P(n)$ is the nth radar estimate for the parameter P

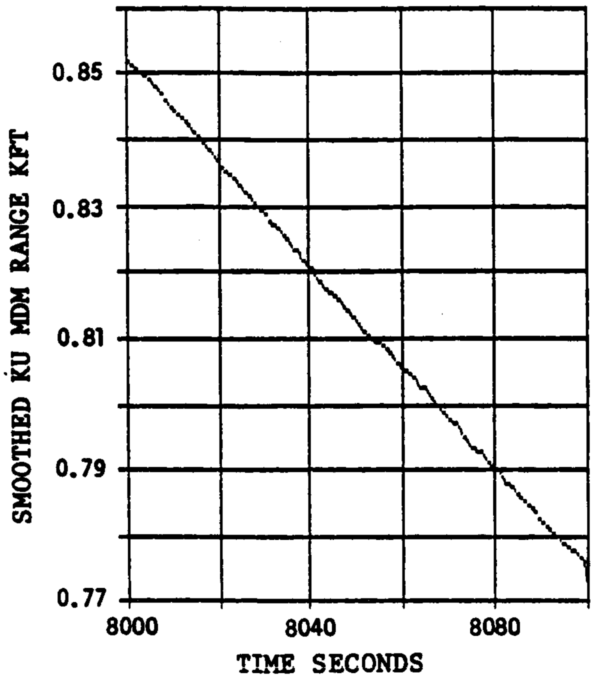
N is the length of the window (it is taken as odd in this analysis).

The value of N used for range, range rate, and roll and pitch angle was 13 samples (at 1 second per sample), and a value of 51 samples was used for ILOS roll and pitch rate. The larger value for the angle rate was to suppress the more severe fluctuations in that data. Figures 4.2-1 and 4.2-2 compare the smoothed and unsmoothed range and range rate data for a window of length 13. Figure 4.2-3 compares the smoothed an unsmoothed pitch angle data. The "steps" in the unsmoothed pitch angle data is due to 0.1 degree quantization of the roll and pitch angle data prior to transmission over the MDM to the shuttle general purpose computer (GPC). These "steps" are eliminated in the smoothed data, as they should be.

Figure 4.2-4 gives the smoothed and unsmoothed ILOS roll rate. In this case, a window of length 51 was used to heavily smooth the "noisy" angle rate. As discussed above, this was probably a mistake, since these fluctuations may have been induced by actual shuttle motion and/or beam wander. However, the validity of this statement cannot be established without a true reference.

4.2.2 Data Analysis Results

Table 4.2-1 summarizes the results of the Palapa rendezvous radar data analysis. This analysis computes the standard deviation of the random component only. Furthermore, the analysis is done for three distinct time intervals corresponding to the three different tracking bandwidths. (It is a fact that the range and angle trackers both have three different bandwidth values and that these values are switched at the same points in range. Also, the bandwidth values of both trackers increase with decreasing range intervals.)



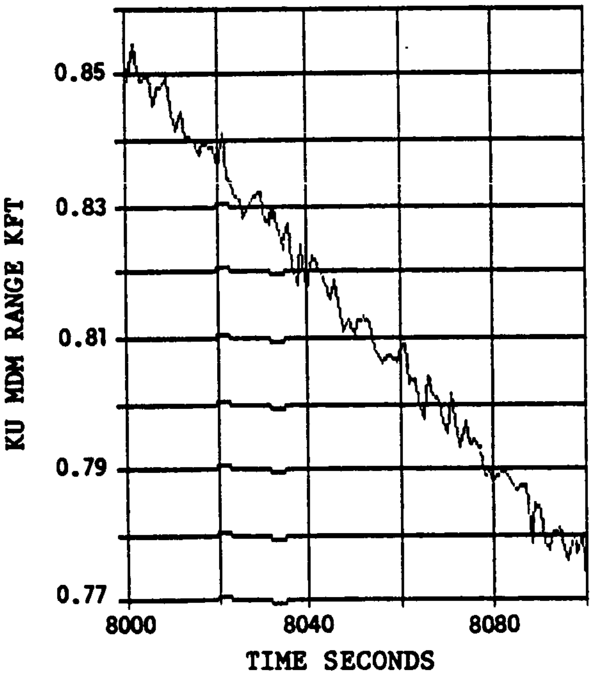
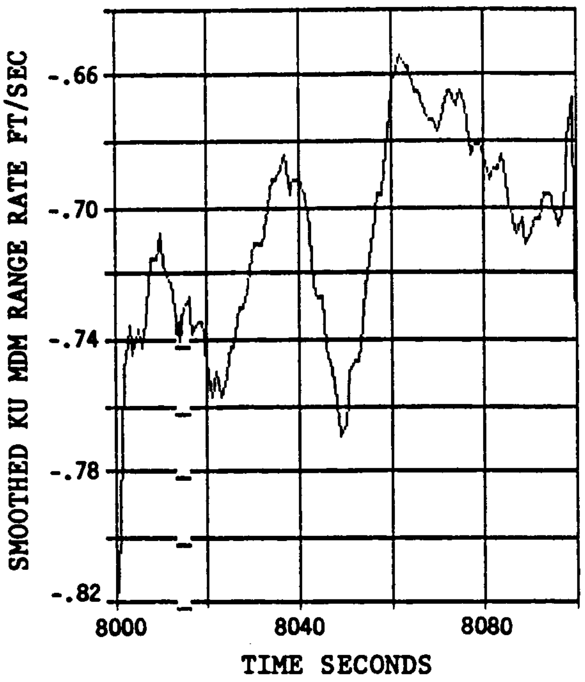


FIGURE 4.2-1 SMOOTHED AND UNSMOOTHED KU MDM RANGE DATA. A 13 SAMPLE WINDOW WAS USED FOR SMOOTHING.



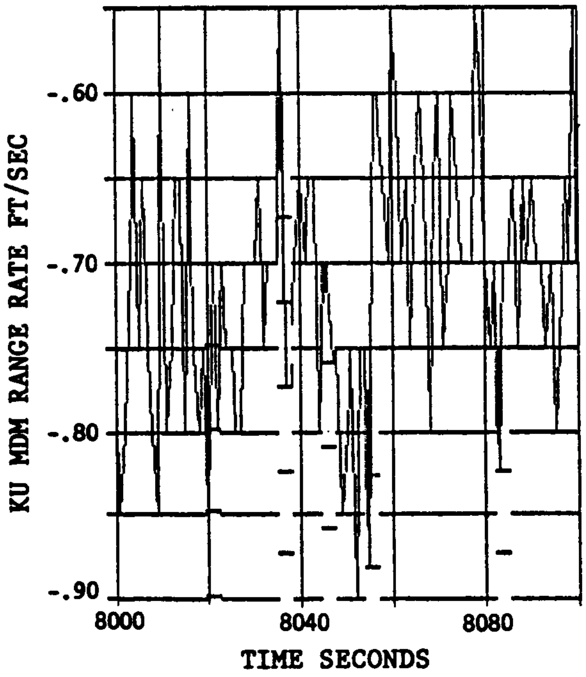
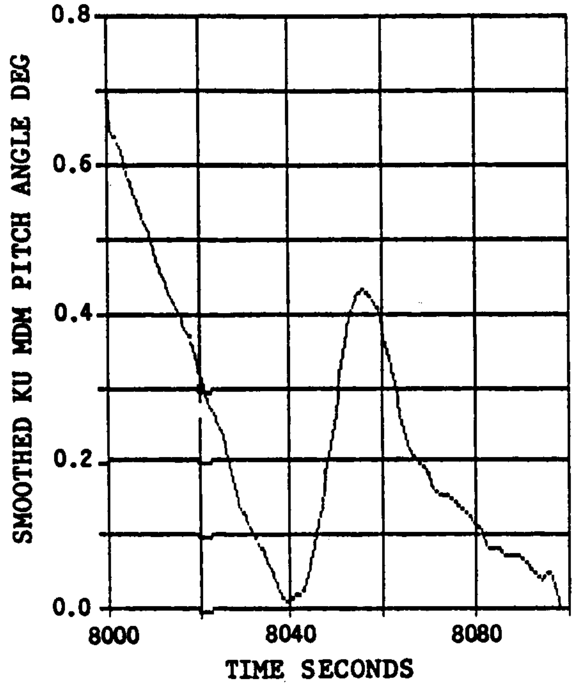


FIGURE 4.2-2 SMOOTHED AND UNSMOOTHED KU MDM RANGE RATE DATA. A 13 SAMPLE WINDOW WAS USED FOR SMOOTHING.



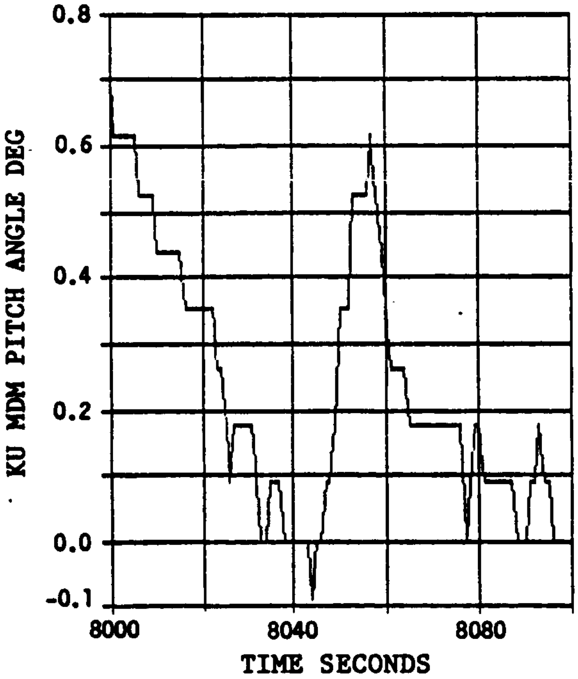
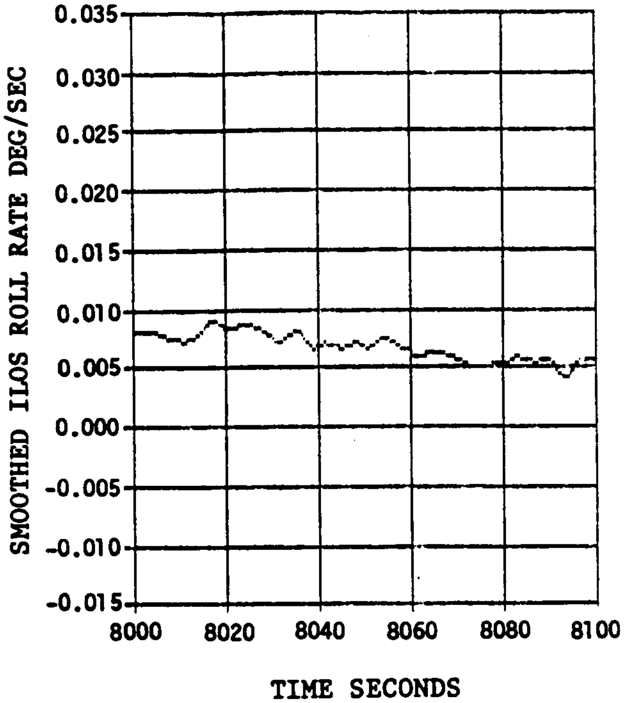


FIGURE 4.2-3 SMOOTHED AND UNSMOOTHED KU MDM PITCH ANGLE DATA. A 13 SAMPLE WINDOW WAS USED FOR SMOOTHING.



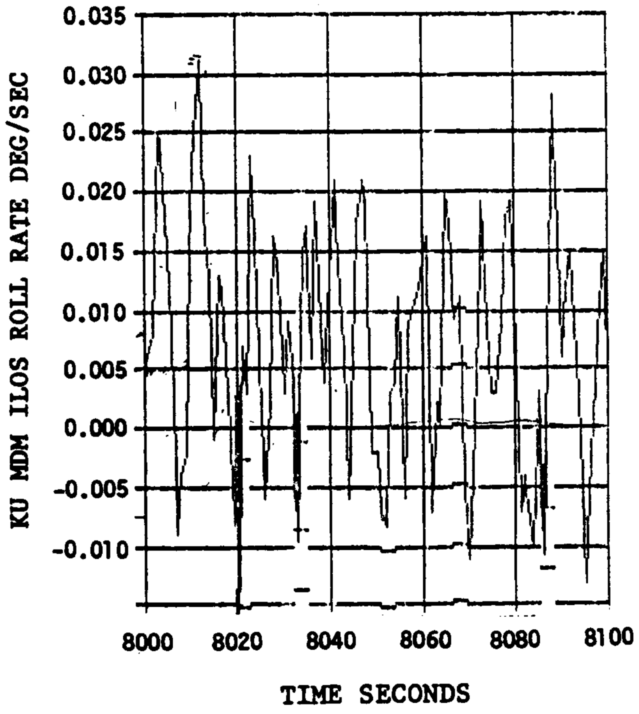


FIGURE 4.2-4 SMOOTHED AND UNSMOOTHED KU MDM ILOS ROLL RATE DATA. A 51 SAMPLE WINDOW WAS USED FOR SMOOTHING.

TABLE 4.2-1 SUMMARY OF RANDOM COMPONENT ANALYSIS OF THE KU-BAND RADAR DATA FROM THE PALAPA SATELLITE RENDEZVOUS OF MISSION 51A

TIME INTERVAL,SEC		4855 - 5890	5890 - 6530	6530 - 6993
RANGE INTERVAL,FT		43520 - 23040	23040 - 11520	11520 - 5760
	SPECIFICATION STD DEV	STD DEV	STD DEV	STD DEV
Range,Ft	26.7 Ft*	20.45	10.97	5.3
Range Rate,Ft	0.333 Ft/Sec	0.119	0.088	0.076
Roll Angle,Deg	0.153 deg	0.037	0.026	0.031
Pitch Angle,Deg	0.153 deg	0.034	0.056	0.052
ILOS Roll Rate, Deg/Sec	2.7 E-3 deg/ sec	8.9 E-4	2.9 E-3	4.7 E-3
ILOS Pitch Rate, Deg/Sec	2.7 E-3 deg/ sec	1.4 E-3	4.4 E-3	6.8 E-3

*The three sigma range specification is 1 percent of range for ranges greater than 8000 feet.

The data of Table 4.2-1 shows that range, range rate and angle are well within their respective specifications for all three range intervals. On the other hand, the angle rate data is within specification for the narrowest bandwidth, but is out of specification for the other bandwidths. Please observe that these data can neither be considered as best-case or worst-case random component analysis. Short-term range accelerations will induce short-term bias in the range and range rate that have been removed with the present smoothing technique. Now, these short-term biases can add to the standard deviation of the random component. However, a calculation of range bias generated by typical acceleration shows that this problem does not add significantly to the range data. Hence, the range data analysis of Table 4.2-1 is an accurate reflection of the radar range performance in flight.

On the other hand, if there is appreciable change in range acceleration, e.g., greater than 10 feet/sec/sec, the bias profile of the range rate data will be affected significantly. A changing range acceleration over a given bandwidth interval produces a changing range rate bias over the corresponding interval (as shown in the SORTE data). This changing bias could add significantly to the random component, putting it out of specification. However, in the range interval of most importance, e.g., ranges less than 5 nautical miles, the deceleration is of very small magnitude. Hence, the range rate data analysis of Table 4.2-1 is an accurate reflection of radar performance under space flight rendezvous conditions.

Angle acceleration will also induce bias in the angle and angle rate data. Hence, a varying bias due to a varying angle accelertaion could induce addition error in the random component. Calculation of this error in Section 3.5 for the angle tracker shows that the bias, under heavy angle acceleration, does not influence the random component significantly. Thus, the angle data analysis of Table 4.2-1 gives representative performance in a space operations environment. Observe that the angle data standard deviation is better than specification by a factor of 5.

At close range, it is hard to decide whether the fluctuations seen in the radar angle rate data are caused by true target shuttle motion, or beam wander, or by radar noise. If the randomness is based on radar noise, then the data of Table 4.2-1 is representative of radar performance and is out of specification. If the fluctuations in the data are non-noise related, as the SORTE data indicates, then the data of Table 4.2-1 is a worst-case result, and it maybe that the angle rate is really within specification once the proper reference is applied.

4.3 SIMULATION RESULTS

4.3.1 Reference Generation

To generate simulation data for the Palapa Satellite rendezvous, a reference flight trajectory had to be developed. This development can be described as follows.

The required inputs to the simulation are the target's position and velocity vectors in shuttle body coordinates and the shuttle angular velocity vector, W_B, in shuttle body coordinates. The target's position and velocity vectors can be obtained from the smoothed range, range rate, and roll angle and pitch angle data described in Section 4.2.1. To obtain the shuttle angular velocity vector requires some additional thought.

The radar data can provide us with two of the three components of the shuttle body angular velocity vector components in body coordinates. These are the X-component and the Y-component. The Z-component representing vehicle cannot be obtained from the data and is assumed zero. The X-component is determined by computing the roll rate from first differences of the roll angle data and subtracting the smooth ILOS roll rate value. The Y-component is determined in a similar fashion using the smoothed pitch angle and smoothed ILOS pitch rate information. Mathematically, this can be expressed as

$$\begin{aligned} \text{(4-2)} \qquad EWB_1(N) &= (SRANG(N)-SRANG(N-1))/\Delta T-SRRTE(N) \\ EWB_2(N) &= (SPANG(N)-SPANG(N-1))/\Delta T-SPRTE(N) \end{aligned}$$

where EWB_1, EWB_2 = X- and Y- component of the shuttle body angular velocity vector

SRANG(N) = Nth value of smoothed roll angle

SPANG(N) = Nth value of smoothed pitch angle

SRRTE(N) = Nth value of smoothed roll rate

SPRTE(N) = Nth value of smoothed pitch rate

ΔT = Sampling period (1 second)

4.3.2 Simulation Performance Against Palapa Reference

Table 4.3-1 summarizes the results of injecting the Palapa reference data into the Ku-Band Radar simulation program and computing statistics over the same range intervals as in the flight data analysis of Section 4.2. The simulation outputs were differenced with their corresponding reference data, and the mean and standard deviation were computed. Comparing this data against the specification yields the following observations. The range standard deviations are within specification, while the mean is not. However, the reason the mean is not in specification is due to artificially setting a bias in the program code. This bias value should probably be changed. The range rate data is within the mean and standard deviation specification for all cases. The same is true for roll angle and pitch angle. However, the ILOS roll and pitch rate standard deviations are slightly out-of-specification in all three range intervals. The reason for this will be drawn into focus in the next section where the simulation and flight difference data statistics are compared.

TABLE 4.3-1 **PERFORMANCE OF THE KU BAND RADAR SIMULATION MODEL USING THE PALAPA SATELLITE RENDEZVOUS OF MISSION 51A AS THE INPUT TRAJECTORY**

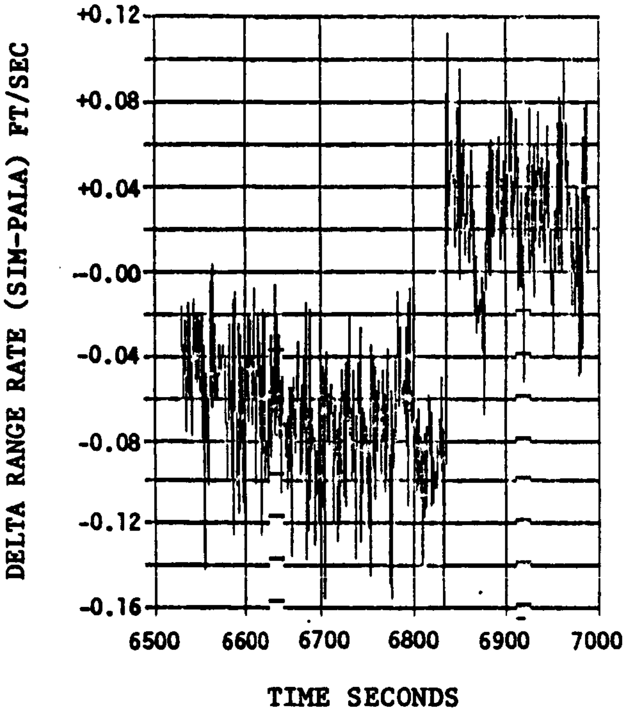
TIME INTERVAL,SEC	4855 - 5890		5890 - 6530		6530 - 6993	
RANGE INTERVAL,FT	43520 - 23040		23040 - 11520		11520 - 5760	
	MEAN	STD DEV	MEAN	STD DEV	MEAN	STD DEV
Range,Ft	99.2	8.57	99.2	5.37	99.6	3.1
Range Rate,Ft/Sec	-0.04	0.06	0.0	0.044	-0.04	0.055
Roll Angle,Deg	0.015	0.044	0.029	0.034	-0.023	0.054
Pitch Angle,Deg	0.066	0.036	0.064	0.041	0.059	0.042
ILOS Roll Rate, Deg/Sec	3.59 E-4	1.02 E-4	3.11 E-4	8.12 E-4	2.25 E-4	1.86 E-3
ILOS Pitch Rate, Deg/Sec	-1.22 E-3	4.24 E-3	-1.01 E-3	4.10 E-3	-8.36 E-4	3.26 E-3

4.3.3 Comparison with Flight Data Performance

The purpose of this subsection is to compare the flight difference data to the simulation difference data. First, the statistics of these two data sets are compared. The range standard deviations compare quite well. The simulation seems to give more optimistic estimates here. However, the simulation shows a decreasing trend in sigma as the absolute range decreases, just as the flight data does. The simulation range rate standard deviation compares well with the corresponding flight data. Again, the simulation shows more optimistic performance than the flight data and the comparison becomes closer at close range. The differences in the range rate performance are not serious enough to question the fidelity of the simulation in this area. The roll and pitch angle standard deviations for the flight and simulation data are both excellent for all three range intervals.

A comparison of the angle rate data statistics shows some inconsistency from range interval-to-range interval. In both roll and pitch rate, the flight data showed the random component progressively getting worse as the range decreases. The simulation data on the other hand seems to fluctuate as the range decreases. This seems confusing! Let's try to make some sense of it by considering the closest range interval. In this case, roll rate flight data is 2.5 times worse than the simulation, and the pitch rate flight data is 3 times worse than the simulation data. As discussed earlier, it is felt that the source of this error was use of the wrong reference for the flight data analysis. That is, the reference was wrong because the apparent randomness in the angle rate was removed with heavy smoothing to form the reference. Based on the analysis of the SORTE angle rate data, it is now felt this "randomness" is, in fact, part of the rendezvous dynamics or, at very close range (less than 2000 feet), beam wander on the target. Another fact that heavily supports this conclusion, is that a comparison of the SORTE flight data and corresponding simulation data showed excellent agreement (see Figures 3.6-8 and 3.6-9). In this case there was a very accurate reference to inject into the simulation. It is recommended that significantly less smoothing be used in the generation of the angle range data reference.

Another method of analysis is to compare the difference data profiles of the flight and simulation data. Figures 4.3-1 and 4.3-2 makes this comparison for range rate and roll rate in the time interval 6500 to 7000 feet (or range interval 11500 to 5700 feet). The reason for the discontinuous jump of 0.12 feet/sec in the simulation range rate data is not known at the present. Otherwise, the data confirms the discussion given above.



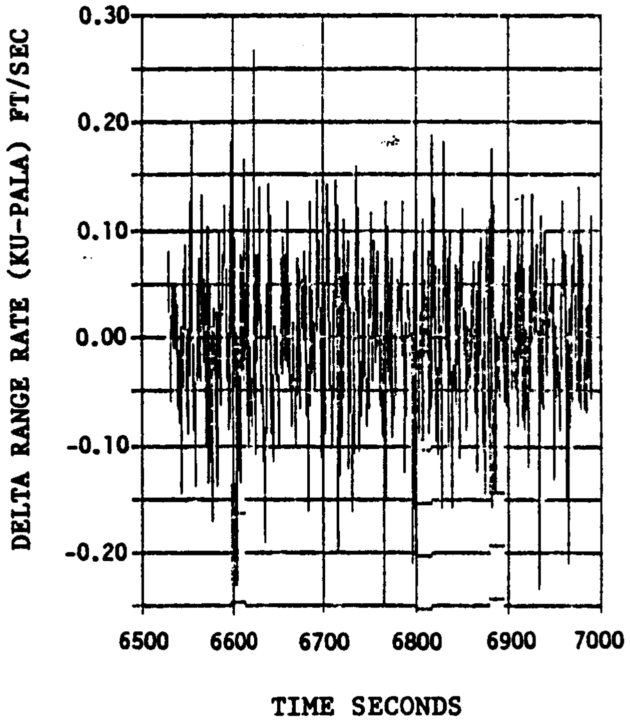
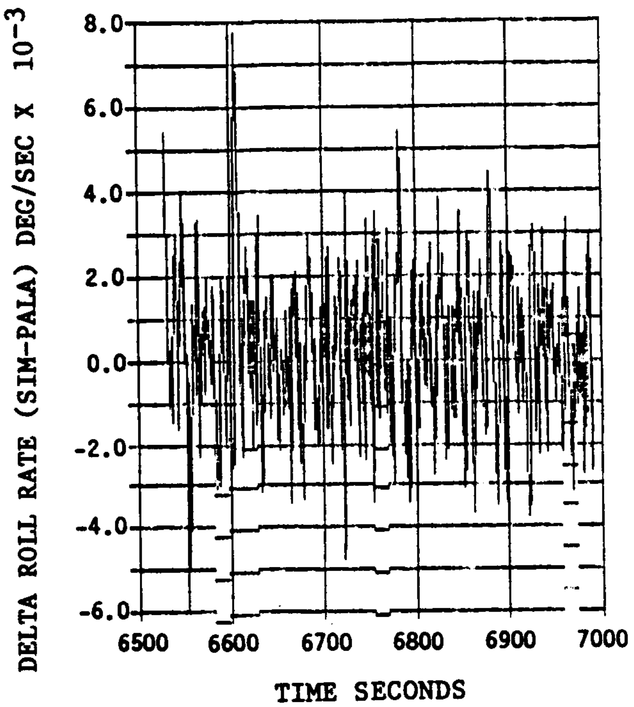


FIGURE 4.3-1 A COMPARISON OF THE KU-BAND RADAR AND THE SIMULATION RANGE RATE DIFFERENCE DATA FOR THE PALAPA SATELLITE RENDEZVOUS



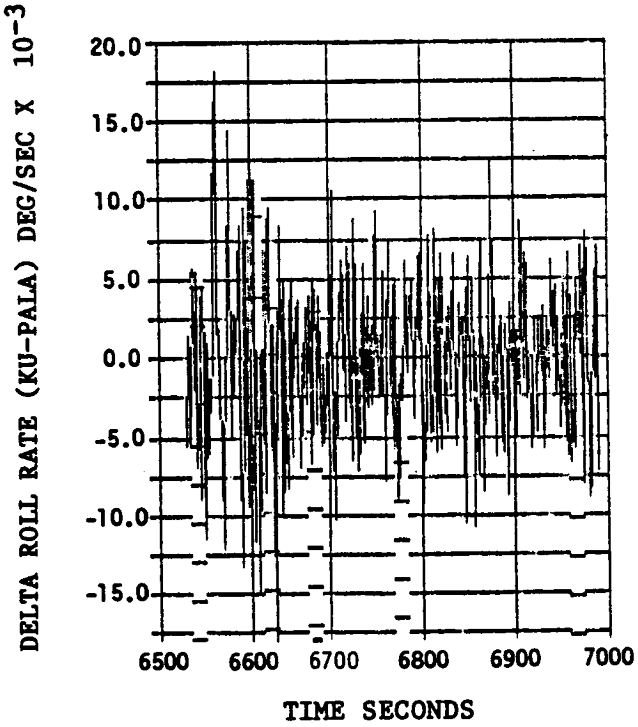


FIGURE 4.3-2 A COMPARISON OF THE KU-BAND RADAR AND THE SIMULATION ILOS ROLL RATE DIFFERENCE DATA FOR THE PALAPA SATELLITE RENDEZVOUS

5.0 REFERENCES

1. Shuttle Orbiter Radar Test and Evaluation, Job Order No. 16-659, LEMSCO Inc., Houston, TX, April 1986

2. Integrated Communications and Radar Equipment Ku-Band Procurement Specification - Revision C, MC409-0025, Rockwell International 7 May 1981

3. Final Report: Ku-Band Rendezvous Radar Performance Computer Simulation Model Contract No. NAS9-15840, Hughes Aircraft Co. RSG, El Segundo, CA, July 1980

4. Final Report: Ku-Band Rendezvous Radar Performance Computer Simulation Model, Contract No. NAS9-15840, Hughes Aircraft Co RSG, El Segundo, CA, June 1984

5. "Ku-Band Angle and Angle-Rate Measurement Accuracy", S. C. Iglehart, Reference HS237-1411, Hughes Aircraft Co RSG, El Segundo, CA, 3 Jan 1978

6. "Effects of Sum and Difference Channel Phase Imbalance on the Ku-Band Radar Angle Acquisition and Tracing Performance", IOC HS237-5494, F. P. Langley III, Hughes Aircraft Co., 7 Aug 1984

7. "Re-evaluation of Ku-Band Radar Angle Rate Tracing Performance", IDC HS237-5493, H. G. Magnusson, Hughes Aircraft Co., 27 June 1984

8. Target Motion Resolution Capabilities, WSMR Instrumentation Directorate Technical Report STEWS-ID-84-1, Elwin C. Nunn and Pamela J. Smith, WSMR, Nov. 1984

9. "SORTE Mathematics", Bill Culpepper, LEMSCO Internal Memo, 28 Mar. 1983

10. Proposal for the Development of Ku-Band Rendezvous Radar Tracking and Acquisition Simulation Programs, Hughes Aircraft Co. SCG, El Segundo, CA, May 1985

(THIS PAGE INTENTIONALLY LEFT BLANK)

APPENDIX A

SOURCE CODE LISTING OF BASELINE PROGRAM

This appendix is a listing of the baseline program which was obtained from JSC at the beginning of the contract. The program is available on the Building 44 VAX and resides in the KUBAND.HOWARD directory. The name of the source program is HACSIM.

```
       COMMON /TARGET/ITARG,SRCS
       COMMON /ACTDAT/R,ARDOT,SPANG,SRANG,SPRTE,SRRTE,AL,BT,SALF,SBTA,
     1ER(3),EV(3),ERTO(3),AZRATE,ELRATE,AZRTE,ELRTE
       COMMON /TERM/ITERM
       COMMON /OUTPUT/MSWF,MTF,MSF,SSRNG,SSRDOT,SSPANG,SSRANG,SSPRTE,
     2                 SSRRTE,SSRSS,MADVF,MRDVF,MARDVF,MRRDVF
     3 ,SSALP,SSBET
       COMMON /SYSDAT/TS,DUM2(14)
C       TEST DATA FROM WS32TDATA1
        CHARACTER*9 FPRO(18)
        CHARACTER*32 IXT,IYT(22),LPRO(18)
        DATA IXT/'TIME  SECONDS$'/
        DATA IYT(1)/'RANGE  FEET$'/
        DATA IYT(2)/'RANGE RATE  FT/SEC$'/
        DATA IYT(3)/'ROLL ANGLE  DEG$'/
        DATA IYT(4)/'PITCH ANGLE  DEG$'/
        DATA IYT(5)/'ROLL RATE  DEG/SEC$'/
        DATA IYT(6)/'PITCH RATE  DEG/SEC$'/
        DATA IYT(7)/'ALPHA  DEG$'/
        DATA IYT(8)/'BETA  DEG$'/
        DATA IYT(9)/'AZ RATE  DEG/SEC$'/
        DATA IYT(10)/'EL RATE  DEG/SEC$'/
        DATA IYT(11)/' X (NORTH)  FEET$'/
        DATA IYT(12)/' Y (EAST)  FEET$'/
        DATA IYT(13)/'-Z (ALTITUDE)  FEET$'/
        DATA IYT(14)/' ELEVATION ANGLE DEG$'/
        DATA IYT(15)/'DELTA RANGE  FEET$'/
        DATA IYT(16)/'DELTA RANGE RATE  FT/SEC$'/
        DATA IYT(17)/'DELTA ROLL ANGLE  DEG$'/
        DATA IYT(18)/'DELTA PITCH ANGLE  DEG$'/
        DATA IYT(19)/'DELTA ROLL RATE  DEG/SEC$'/
        DATA IYT(20)/'DELTA PITCH RATE  DEG/SEC$'/
        DATA IYT(21)/'DELTA ALPHA  DEG$'/
        DATA IYT(22)/'DELTA BETA  DEG$'/
        DATA LPRO(1)/' SIMULATION PROFILE HJ146$'/
        DATA LPRO(2)/' SIMULATION PROFILE HL146$'/
        DATA LPRO(3)/' SIMULATION PROFILE HL246$'/
        DATA LPRO(4)/' SIMULATION PROFILE HL346$'/
        DATA LPRO(5)/' SIMULATION PROFILE HL446$'/
        DATA LPRO(6)/' SIMULATION PROFILE HL546$'/
        DATA LPRO(7)/' SIMULATION PROFILE BJ146$'/
        DATA LPRO(8)/' SIMULATION PROFILE BL146$'/
        DATA LPRO(9)/' SIMULATION PROFILE BL246$'/
        DATA LPRO(10)/' SIMULATION PROFILE BL346$'/
        DATA LPRO(11)/' SIMULATION PROFILE BL446$'/
        DATA LPRO(12)/' SIMULATION PROFILE BL546$'/
        DATA LPRO(13)/' SIMULATION PROFILE C6P48$'/
        DATA LPRO(14)/' SIMULATION PROFILE C6M48$'/
        DATA LPRO(15)/' SIMULATION PROFILE C6P30$'/
        DATA LPRO(16)/' SIMULATION PROFILE C6M30$'/
```

```
      DATA LPRO(17)/' SIMULATION PROFILE CLP16$'/
      DATA LPRO(18)/' SIMULATION PROFILE CLM16$'/
      DIMENSION RID(120)
      DATA FPRO(1)/'HJ146.JSC'/
      DATA FPRO(2)/'HL146.BIN'/
      DATA FPRO(3)/'HL246.BIN'/
      DATA FPRO(4)/'HL346.BIN'/
      DATA FPRO(5)/'HL446.BIN'/
      DATA FPRO(6)/'HL546.BIN'/
      DATA FPRO(7)/'BJ146.BIN'/
      DATA FPRO(8)/'BL146.BIN'/
      DATA FPRO(9)/'BL246.BIN'/
      DATA FPRO(10)/'BL346.BIN'/
      DATA FPRO(11)/'BL446.BIN'/
      DATA FPRO(12)/'BL546.BIN'/
      DATA FPRO(13)/'C6P48.BIN'/
      DATA FPRO(14)/'C6M48.BIN'/
      DATA FPRO(15)/'C6P30.BIN'/
      DATA FPRO(16)/'C6M30.BIN'/
      DATA FPRO(17)/'CLP16.BIN'/
      DATA FPRO(18)/'CLM16.BIN'/
      CHARACTER*9 UNIT7
      BYTE IC(120)
      COMMON /TMR/X,Y,Z,VX,VY,VZ,
     1              DLP(3),DEL(3),DUE(3),
     2              DSU(3),THAZL1,THEL1,THAZU1
     COMMON /INPUT/RO(3),VO(3),EWB(3)
      DIMENSION TP(2001),D(2001,22)
C
      WRITE (6,*)'1 : TEK'
      WRITE (6,*)'2 : VT125'
      WRITE (6,*)'3 : VT240'
      WRITE (6,*)'4 : PC'
      READ (5,*)ITERM
      WRITE(6,*)'PROFILE NUMBER   PROFILE'
      DO L=1,18
      WRITE(6,200)L,LPRO(L)
200   FORMAT(7X,I2,9X,A32)
      ENDDO
      WRITE(6,*)'INPUT PROFILE NUMBER'
      READ(5,*)ITAPE
C     WRITE(6,*) 'ENTER NAME OF BINARY INPUT FILE'
C     READ(5,1001)UNIT7
C1001 FORMAT(A24)
      UNIT7=FPRO(ITAPE)
      OPEN(UNIT=4,FORM='UNFORMATTED',STATUS='OLD',
     2     FILE=UNIT7)
C
C     READ(4)IC
C     WRITE(6,150)(IC(I),I=1,30)
150   FORMAT(60A2)
      IFTRK=0
      WRITE(6,*)' INPUT 1 IF YOU WANT TO FILTER USING TRACK FLAG'
      READ(5,*)IFTRK
      WRITE(6,*)'INPUT RSC IN SQUARE METERS'
      READ (5,*)RCSM
      SRCS=RCSM*3.28*3.28
      SRCS=SQRT(SRCS)
      WRITE(6,*)'SRCS=',SRCS
      TOUT=0
      THAZL1=30.
      THEL1=30.
      THAZU1=0.
      DLP(1)=-0.2347
```

```
        DLP(2)=0.05
        DLP(3)=-9.748
        DEL(1)=-0.192738
        DEL(2)=-0.055573
        DEL(3)=-3.299135
        DUE(1)=0.88
        DUE(2)=0.55
        DUE(3)=-0.39988
        DSU(1)=1.67
        DSU(2)=0.73
        DSU(3)=-5.46
        WRITE(6,*)' INPUT 1 FOR SCREEN OUTPUT'
        READ(5,*)TOUT
        J=0
        READ(4,END=99)T,X,Y,Z,VX,VY,VZ
        READ(4,END=99)T1,X,Y,Z,VX,VY,VZ
        TS=T1-T
        WRITE(6,*)' TS= ',TS
1       CONTINUE
        READ(4,END=99)T,X,Y,Z,VX,VY,VZ
C DATA IN METERS
        CALL TMR2KU
        IF(TOUT.EQ.1)THEN
        WRITE(6,100)T,SSRNG,SSSRRDOT,SSPANG,SRANG,SSPRTE,SRRTE,SALF,SBTA,
     1   AZRATE,ELRATE,AZRTE,ELRTE
100     FORMAT(' ',2F9.1,9F9.3)
        ENDIF
        CALL EXEC
        IF(IFTRK.EQ.1.AND.MTF.EQ.0)GO TO 1
        J=J+1
        IF(J.EQ.2001)GO TO 99
        TP(J)=T
        D(J,1)=SSRNG
        D(J,2)=SSRDOT
        D(J,4)=SSPANG
        D(J,3)=SSRANG
        D(J,5)=SSRRTE
        D(J,6)=SSPRTE
        D(J,7)=SSALP
        D(J,8)=SSBET
        D(J,9)=AZRTE
        D(J,10)=ELRTE
        D(J,11)=X
        D(J,12)=Y
        D(J,13)=-Z
        D(J,14)=ATAND(-Z/(X*X+Y*Y))
        D(J,15)=SSRNG-R
        D(J,16)=SSRDOT-ARDOT
        D(J,17)=SSRANG-SRANG
        D(J,18)=SSPANG-SPANG
        D(J,19)=SSRRTE-SRTE
        D(J,20)=SSPRTE-SPRTE
        D(J,21)=SSALP-SALF
        D(J,22)=SSBET-SBTA
        GO TO 1
99      CONTINUE
        IXD=0
94      CONTINUE
        WRITE(6,*)'RCS IN METERS=',RCSM
        WRITE(6,*)'PARA          AXES TITLE'
        DO I=1,22
       WRITE(6,68)I,IYT(I)
68      FORMAT(1X,I4,10X,A32)
        ENDDO
```

```
WRITE(6,*)'INPUT IXD,IYD  IXD=0 FOR TIME'
READ(5,*)IXD,IYD
CALL SORT(TP,D,J,ITAPE,IXD,IYD)
GO TO 94
END
SUBROUTINE SORT(T,D,J,ITAPE,IXD,IYD)
DIMENSION D(2001,22),X(2001),Y(2001),T(2001)
CHARACTER*32 IXT,IYT(22),LPRO(18)
DIMENSION ITILT(8),IXL(8),IYL(8)
DATA IXT/'TIME  SECONDS$'/
DATA IYT(1)/'RANGE  FEET$'/
DATA IYT(2)/'RANGE RATE  FT/SEC$'/
DATA IYT(3)/'ROLL ANGLE  DEG$'/
DATA IYT(4)/'PITCH ANGLE  DEG$'/
DATA IYT(5)/'ROLL RATE  DEG/SEC$'/
DATA IYT(6)/'PITCH RATE  DEG/SEC$'/
DATA IYT(7)/'ALPHA  DEG$'/
DATA IYT(8)/'BETA  DEG$'/
DATA IYT(9)/'AZ RATE  DEG/SEC$'/
DATA IYT(10)/'EL RATE  DEG/SEC$'/
DATA IYT(11)/' X (NORTH)  FEET$'/
DATA IYT(12)/' Y (EAST)  FEET$'/
DATA IYT(13)/'-Z (ALTITUDE)  FEET$'/
DATA IYT(14)/' ELEVATION ANGLE DEG$'/
DATA IYT(15)/'DELTA RANGE  FEET$'/
DATA IYT(16)/'DELTA RANGE RATE  FT/SEC$'/
DATA IYT(17)/'DELTA ROLL ANGLE  DEG$'/
DATA IYT(18)/'DELTA PITCH ANGLE  DEG$'/
DATA IYT(19)/'DELTA ROLL RATE  DEG/SEC$'/
DATA IYT(20)/'DELTA PITCH RATE  DEG/SEC$'/
DATA IYT(21)/'DELTA ALPHA  DEG$'/
DATA IYT(22)/'DELTA BETA  DEG$'/
DATA LPRO(1)/' SIMULATION PROFILE HJ146$'/
DATA LPRO(2)/' SIMULATION PROFILE HL146$'/
DATA LPRO(3)/' SIMULATION PROFILE HL246$'/
DATA LPRO(4)/' SIMULATION PROFILE HL346$'/
DATA LPRO(5)/' SIMULATION PROFILE HL446$'/
DATA LPRO(6)/' SIMULATION PROFILE HL546$'/
DATA LPRO(7)/' SIMULATION PROFILE BJ146$'/
DATA LPRO(8)/' SIMULATION PROFILE BL146$'/
DATA LPRO(9)/' SIMULATION PROFILE BL246$'/
DATA LPRO(10)/' SIMULATION PROFILE BL346$'/
DATA LPRO(11)/' SIMULATION PROFILE BL446$'/
DATA LPRO(12)/' SIMULATION PROFILE BL546$'/
DATA LPRO(13)/' SIMULATION PROFILE C6P48$'/
DATA LPRO(14)/' SIMULATION PROFILE C6M48$'/
DATA LPRO(15)/' SIMULATION PROFILE C6P30$'/
DATA LPRO(16)/' SIMULATION PROFILE C6M30$'/
DATA LPRO(17)/' SIMULATION PROFILE CLP16$'/
DATA LPRO(18)/' SIMULATION PROFILE CLM16$'/
JPRO=ITAPE
CALL FIXIT(ITILT,LPRO(JPRO))
IF(IXD.EQ.0)THEN
DO I=1,J
X(I)=T(I)
Y(I)=D(I,IYD)
ENDDO
CALL FIXIT(IXL,IXT)
CALL FIXIT(IYL,IYT(IYD))
ELSE
DO I=1,J
X(I)=D(I,IXD)
Y(I)=D(I,IYD)
ENDDO
```

```
      CALL FIXIT(IXL,IYT(IXD))
      CALL FIXIT(IYL,IYT(IYD))
      ENDIF
      CALL PLOTIT(ITILT,IXL,IYL,X,Y,J)
      RETURN
      END
      SUBROUTINE FIXIT(IOUT,IN)
      DIMENSION IOUT(8)
      CHARACTER*4 ITEMP(8)
      CHARACTER*32 IN
      ITEMP(1)=(IN(1:4))
      ITEMP(2)=(IN(5:8))
      ITEMP(3)=(IN(9:12))
      ITEMP(4)=(IN(13:16))
      ITEMP(5)=(IN(17:20))
      ITEMP(6)=(IN(21:24))
      ITEMP(7)=(IN(25:28))
      ITEMP(8)=(IN(29:32))
      ENCODE(32,999,IOUT)(ITEMP(I),I=1,8)
999   FORMAT(8A4)
      RETURN
      END
      SUBROUTINE PLOTIT(ITILT,IXL,IYL,X,Y,J)
      COMMON /TERM/ITERM
      DIMENSION ITILT(8),IXL(8),IYL(8)
      DIMENSION X(1),Y(1)
      BYTE CR(2)
      COMMON/TMR/A,B,C,D,E,F,G(3),AH(3),AI(3),AJ(3),THAZL1,THEL1,THAZU1
      CR(1)=27
      CR(2)=12
      XMAX=X(1)
      XMIN=X(1)
      YMAX=Y(1)
      YMIN=Y(1)
      DO I=1,J
      IF(X(I).GT.XMAX) XMAX=X(I)
      IF(X(I).LT.XMIN) XMIN=X(I)
      IF(Y(I).GT.YMAX) YMAX=Y(I)
      IF(Y(I).LT.YMIN) YMIN=Y(I)
      END DO
      IF(XMAX.EQ.XMIN)XMAX=XMIN*1.1
      IF(YMAX.EQ.YMIN)YMAX=YMIN*1.1
      IF (ITERM.EQ.1) CALL TEKALL(4114,480,0,1,0)
      IF (ITERM.EQ.2) CALL REGIS (1,0)
      IF (ITERM.EQ.3) CALL PVT240
      CALL BGNPL(-1)
      CALL FLATBD
      CALL PAGE(14.,18.)
      CALL HEIGHT(.3)
      CALL TITLE(ITILT,100,IXL,100,IYL,100,9.0,13.5)
      I100=100
C     CALL MESSAG('LOWER AZIMUTH=$',I100,1.7,13.)
C     CALL REALNO(THAZL1,2,'ABUT','ABUT')
C     CALL MESSAG('UPPER AZIMUTH=$',I100,1.7,12.5)
C     CALL REALNO(THAZU1,2,'ABUT','ABUT')
C     CALL MESSAG('ELEVATION=$',I100,1.7,12.)
C     CALL REALNO(THEL1,2,'ABUT','ABUT')
C     CALL BLNK1(1.5,7.5,11.9,13.5,4)
C     CALL HEADIN(ITILT,-100,-8,4)
C     CALL HEADIN('LOWER AZIMUTH=$',100,4,4)
C     CALL REALNO(THAZL1,2,'ABUT','ABUT')
C     CALL HEADIN('UPPER AZIMUTH=$',100,4,4)
C     CALL REALNO(THAZU1,2,'ABUT','ABUT')
C     CALL HEADIN('ELEVATION=$',100,4,4)
```

```
C       CALL REALNO(THEL1,2,'ABUT','ABUT')
        CALL YAXANG(0.)
        CALL GRAF(XMIN,'SCALE',XMAX,YMIN,'SCALE',YMAX)
        CALL CURVE(X,Y,J,0)
C       KK=J/30
C       K=0
C       DO I=1,KK
C       K=30+K
C       CALL RLINT(K,X(K),Y(K))
C       ENDDO

        CALL GRID(1,1)
        CALL HEIGHT(.1)
        CALL RESET('HEIGHT')
888     FORMAT('+',2A1)
        CALL DONEPL
C MICKEY MOUSE FIX
        IMM=1
        IF(IMM.EQ.0)THEN
        REWIND (5)
        READ(5,192)IC
192     FORMAT(A1)
       WRITE(6,888)CR
        ENDIF
        RETURN
        END
        SUBROUTINE TMR2KU
C **  MODED JWG 2/8/85
C **

C **   INPUT VIA COMMON VIA X,Y,Z,VX,VY,VZ
C **  OUTPUT VIA COMMON /ACTDAT/
C **
C *** WHITE SANDS TO KU-BAND RADAR PARAMETER CONVERSION ***
C
C
C                 ******* COMMENTARY *******
C
C
C                       ** PURPOSE **
C THIS SOFTWARE TAKES THE POSITION AND VELOCITY OF A TARGET REFERENCED
C TO THE PEARL SITE SURVEY CAP AND CALCULATES THE VALUES OF THE KU-BAND
C RADAR PARAMETERS AS SEEN AT THE KU-BAND RADAR GIMBAL AXES INTERSECTION.
C THESE CALCULATIONS INVOLVE COORDINATE ROTATIONS THROUGH A THREE-AXIS
C POSITIONER AND FOUR TRANSLATIONS FROM THE PEARL CAP TO THE RADAR GIMBAL
C AXES INTERSECTION.
C
C THESE CALCULATIONS ARE TO BE DONE BY WSMR DATA REDUCTION USING THE WSMR
C RANGE REFERENCE ESTIMATIONS OF TARGET LOCATION WITH TIME.  COMPARISON
C CAN BE MADE DIRECTLY WITH THE KU-BAND OUTPUTS FOR THE SAME TIME VALUES.
C
C                   ** INPUTS & CONSTANTS **
C
C  WSMR PROVIDED INPUTS:
C   WSMR WILL PROVIDE TARGET POSITION - X, Y, Z - AND VELOCITY - VX, VY,
C   VZ AS INPUTS TO THIS PROGRAM.
C   UNITS ARE FEET AND FEET/SECOND.
C   THE COORDINATE SYSTEM IS:
C       ORIGIN = PEARL SURVEY CAP
C       X-AXIS IS POSITIVE TOWARD THE NORTH
C       Y-AXIS IS POSITIVE TOWARD THE EAST
C       NEGATIVE Z-AXIS IS UPWARD ALONG THE LOCAL VERTICAL.
C
C  CONSTANTS PROVIDED BY SIMULATION TEST TAPE:
```

```
C    FOR ANY GIVEN TEST THE FOLLOWING PARAMETERS WILL BE DEFINED ON THE
C    SIMULATION MAGNETIC DATA TAPE AND WILL REMAIN CONSTANT FOR THAT TEST:
C     DSU(I) I=1,3  IS THE LOCATION OF THE KU-BAND RADAR GIMBAL AXES IN
C                   UPPER AZIMUTH COORDINATES.
C     THAZL1        IS THE LOWER AZIMUTH AXIS ROTATION ANGLE IN DEGREES.
C     THEL1         IS THE ELEVATION AXIS ROTATION ANGLE IN DEGREES.
C     THAZU1        IS THE UPPER AZIMUTH AXIS ROTATION ANGLE IN DEGREES.
C
C   ONE TIME INPUT CONSTANTS:
C    THE FOLLOWING PARAMETERS WILL BE MEASURED AFTER INSTALLATION OF THE
C    ANTENNA PEDESTAL AT THE PEARL SITE.  THEIR VALUES SHOULD NOT CHANGE.
C    THEY ARE CURRENTLY DEFINED AS ZERO IN THIS SOFTWARE.
C      DLP(I) I=1,3  LOCATION OF THE LOWER AZIMUTH ORIGIN IN PEARL
C                    COORDINATES.
C      DEL(I) I=1,3  LOCATION OF THE ELEVATION ORIGIN IN LOWER AZIMUTH
C                    COORDINATES.
C      DUE(I) I=1,3  LOCATION OF THE UPPER AZIMUTH ORIGIN IN ELEVATION
C                    COORDINATES.
C
C
C                  ** SOFTWARE OUTPUTS **
C   THIS SOFTWARE PRODUCES THE FOLLOWING OUTPUTS REFERENCED TO THE
C   RADAR GIMBAL AXES INTERSECTION.
C
C    R = RANGE (FT)
C    ARDOT = RANGE RATE (FT/SEC)
C    SRANG = ROLL ANGLE (DEG)
C    SPANG = PITCH ANGLE (DEG)
C    SRRTE = INERTIAL ROLL RATE (DEG/SEC)
C    SPRTE = INERTIAL PITCH RATE (DEG/SEC)
C    SALF = ALPHA ANGLE (DEG)
C    SBTA = BETA ANGLE (DEG)
C    AZRTE = AZIMUTH ANGLE RATE (DEG/SEC)
C    ELRTE = ELEVATION ANGLE RATE (DEG/SEC)
C
C                  ** EXAMPLE **
C   AN EXAMPLE CASE IS INCLUDED IN THE CODE.  IF THIS SOURCE IS COMPILED,
C   LINKED, AND EXECUTED, OUTPUTS WILL GO TO UNIT 6.  THEIR VALUES SHOULD
C   BE:
C     R = 43760.6016                ARDOT = -9.87364578
C     SRANG = 25.2644920            SPANG = 28.2407990
C     SRRTE = -.926818550E-01       SPRTE = .688237743E-02
C     SALF = -36.1578255            SBTA = 9.27430439
C     AZRTE = .302744657E-01        ELRTE = -.105446391
C
C
C
        COMMON /TMR/X,Y,Z,VX,VY,VZ,
     1            DLP(3),DEL(3),DUE(3),
     2            DSU(3),THAZL1,THEL1,THAZU1
       COMMON /INPUT/RO(3),VO(3),EWB(3)
       COMMON /ACTDAT/R,ARDOT,SPANG,SRANG,SPRTE,SRRTE,AL,BT,SALF,SBTA,
     1ER(3),EV(3),ERTO(3),AZRATE,ELRATE,AZRTE,ELRTE
C      DIMENSION DLP(3),DEL(3),DUE(3),DSU(3)
       DIMENSION AZL(3,3),ELV(3,3),AZU(3,3)
       DIMENSION DPT(3),DLT(3),DET(3),DUT(3),DST(3)
       DIMENSION DLAZ(3),DELV(3),DAZU(3)
       DIMENSION VPT(3),VLAZ(3),VELV(3),VST(3)
       DATA DEGRAD/57.275/,PI/3.14159/
C   THE EWB PARAMETERS ARE ALWAYS DEFINED AS 0.0
       EWB(1)=0.0
       EWB(2)=0.0
       EWB(3)=0.0
C EXAMPLE CASE VALUES:
```

```
C      X=39417.2812
C      Y=16164.0078
C      Z=-9999.65820
C      VX=-41.1736259
C      VY=73.6755753
C      VZ=.166666671E-02
C      THAZL2=45.0
C      THEL2=-45.0
C      THAZU2=0.0
C
C
C  ** INPUTS **
C  WSMR WILL NORMALLY PROVIDE X,Y,Z,VX,VY,VZ.  REF IS PEARL SURVEY POINT.
C THIS IS PROVIDED VIA COMMON TMR BLOCK
       DPT(1)=X
       DPT(2)=Y
       DPT(3)=Z
       VPT(1)=VX
       VPT(2)=VY
       VPT(3)=VZ
C
C
C  ** CONSTANTS **
C DLP(I); DEL(I); AND DUE(I) WILL BE PROVIDED ONE TIME AFTER INSTALLATION
C OF THE ANTENNA PEDESTAL
C THIS IS PROVIDED VIA COMMON TMR BLOCK
C      DLP(1)=0.0
C      DLP(2)=0.0
C      DLP(3)=0.0
C      DEL(1)=0.0
C      DEL(2)=0.0
C      DEL(3)=0.0
C      DUE(1)=0.0
C      DUE(2)=0.0
C      DUE(3)=0.0
C
C
C ** CONSTANTS FROM SIMULATION DATA TAPE **
C THIS IS PROVIDED VIA COMMON TMR BLOCK
C      DSU(1)=0.0
C      DSU(2)=0.0
C      DSU(3)=0.0
C      THAZL1=0.0
C      THEL1=0.0
C      THAZU1=0.0
C
C EXAMPLE ANGLE VALUES ARE EQUATED HERE.
C      THAZL1=THAZL2
C      THEL1=THEL2
C      THAZU1=THAZU2
C  CONVERT TO RADIANS
       THAZL=THAZL1/DEGRAD
       THEL=THEL1/DEGRAD
       THAZU=THAZU1/DEGRAD
C SET UP THE ROTATIONAL MATRICES
       CALL AZGEN(AZL,THAZL)
       CALL ELGEN(ELV,THEL)
       CALL AZGEN(AZU,THAZU)
C CONVERT TARGET IN PEARL TO TARGET AT GIMBALS
       DO 11 I=1,3
11     DLT(I)=DPT(I)-DLP(I)
       CALL MULT31(AZL,DLT,DLAZ)
       DO 21 I=1,3
21     DET(I)=DLAZ(I)-DEL(I)
```

```
      CALL MULT31(ELV,DET,DELV)
      DO 31 I=1,3
31    DUT(I)=DELV(I)-DUE(I)
      CALL MULT31(AZU,DUT,DAZU)
      DO 41 I=1,3
41    DST(I)=DAZU(I)-DSU(I)
C THESE ARE THE THREE TARGET COORDINATES IN RADAR GIMBAL REFERENCE.
      RO(1)=DST(1)
      RO(2)=DST(2)
      RO(3)=DST(3)
C CONVERT TO VELOCITIES REFERENCED TO GIMBALS
      CALL MULT31(AZL,VPT,VLAZ)
      CALL MULT31(ELV,VLAZ,VELV)
      CALL MULT31(AZU,VELV,VST)
C THESE ARE VELOCITIES IN GIMBAL REFERENCE.
      VO(1)=VST(1)
      VO(2)=VST(2)
      VO(3)=VST(3)
C
C       RO(I) VO(I) I=1,3 SHUTTLE BODY POS AND VEL VECTOR
C
C CALCULATE THE KU-BAND RADAR PARAMETERS BASED ON THE INPUTS.
        C23=COSD(23.)
        S23=SIND(23.)
        X1=RO(2)*C23-RO(3)*S23
        Y1=-RO(2)*S23-RO(3)*C23
        Z1=-RO(1)
        RO(1)=X1
        RO(2)=Y1
        RO(3)=Z1
        VX=VO(2)*C23-VO(3)*S23
        VY=-VO(2)*S23-VO(3)*C23
        VZ=-VO(1)
        VO(1)=VX
        VO(2)=VY
        VO(3)=VZ
       CALL ACT
       SRRTE=SRRTE*(DEGRAD/1000.)
       SPRTE=SPRTE*(DEGRAD/1000.)
       SALF=AL*DEGRAD
       SBTA=BT*DEGRAD
       AZRTE=AZRATE*DEGRAD
       ELRTE=ELRATE*DEGRAD
C THE EXAMPLE CASE RESULTS ARE:
C      WRITE(6,*)R,ARDOT
C      WRITE(6,*)SRANG,SPANG
C      WRITE(6,*)SRRTE,SPRTE
C      WRITE(6,*)SALF,SBTA
C      WRITE(6,*)AZRTE,ELRTE
        RETURN
       END
       SUBROUTINE AZGEN(AZ,ANGAZ)
C THIS SUBROUTINE PRODUCES A 3X3 MATRIX, AZ, FOR
C AN AZIMUTH TABLE ROTATION OF ANGAZ RADIANS.
       DIMENSION AZ(3,3)
       DO 10 I=1,3
       DO 10 J=1,3
10     AZ(I,J)=0.0
       AZ(1,1)=COS(ANGAZ)
       AZ(1,2)=SIN(ANGAZ)
       AZ(2,1)=-SIN(ANGAZ)
       AZ(2,2)=COS(ANGAZ)
       AZ(3,3)=1.0
       RETURN
      END
      SUBROUTINE ELGEN(EL,ANGEL)
      DIMENSION EL(3,3)
      DO 10 I=1,3
      DO 10 J=1,3
10    EL(I,J)=0.0
      EL(1,1)=COS(ANGEL)
      EL(1,3)=-SIN(ANGEL)
      EL(2,2)=1.0
      EL(3,1)=SIN(ANGEL)
      EL(3,3)=COS(ANGEL)
      RETURN
      END
C   SUBROUTINE ACT
C                                                                       00015100
C                                                                       00015110
C *************************************************************         00015120
C * THIS SUBROUTINE INITIALIZES THE ANGLE TRACKING LOOPS, THE   *       00015130
C * RANGE TRACKING LOOP, AND THE VELOCITY PROCESSOR -- STEADY   *       00015140
C * STATE CONDITIONS ARE ASSUMED.                               *       00015150
C *************************************************************         00015160
C                                                                       00015170
C                                                                       00015180
      SUBROUTINE ACT
      COMMON /ACTDAT/R,ARDOT,SPANG,SRANG,SPRTE,SRRTE,AL,BT,SALF,SBTA
     2,ER(3),EV(3),ERTO(3),AZRATE,ELRATE,AZRTE,ELRTE
      COMMON /INPUT/ ERT(3),EVT(3),EWB(3),DUM(18)                       00015210
      COMMON /SYSDAT/TSAM,DR(3),CP,SP,PSI,PSBIAS,DUM2(7),TRB(3,3)       00015250
      DIMENSION       FLTWID(3),RI(10)
      DIMENSION TX1(3,3),TX2(3,3),TX3(3,3),TBL(3,3)
      DATA PI/3.141592653/
      DATA IONE/0/
      IF(IONE.EQ.0)CALL DATA
      IONE=1
C                                                                       00015560
C STEP 1-1: COMPUTE INITIAL INNER AND OUTER GIMBAL POSITIONS.           00015570
C   (NOTE: TRANSFORM CONSISTS OF TRANSLATION PLUS ROTATION.)            00015580
C   PERFORM TRANSLATION -- SHIFT TO RADAR FRAME ORIGIN.                 00015590
      DO 1 I=1,3                                                        00015600
    1 ERTO(I)=ERT(I)-DR(I)                                              00015610
C   TRANSFORM TARGET POSITION FROM BODY TO RADAR FRAME.                 00015640
      CALL MULT31(TRB,ERTO,ER)                                          00015650
C   TRANSFORM TARGET VELOCITY FROM BODY TO RADAR FRAME.                 00015660
      CALL MULT31(TRB,EVT,EV)                                           00015670
      SQ=SQRT(ER(2)*ER(2)+ER(3)*ER(3))                                  00015680
C   COMPUTE INNER(BETA) GIMBAL POSITION -- BT.                          00015690
      IF(ER(1).EQ.0.0.AND.SQ.EQ.0.0) STOP                               00015700
      BT=-ATAN2(ER(1),SQ)                                               00015710
      ER2=-ER(2)                                                        00015720
      ER3=-ER(3)                                                        00015730
C   COMPUTE OUTER(ALPHA) GIMBAL POSITION -- AL.                         00015740
      IF(ER2.EQ.0.0.AND.ER3.EQ.0.0) GO TO 8                             00015750
      AL=-ATAN2(ER2,ER3)                                                00015760
      GO TO 9                                                           00015770
    8 IF(ER(1).GT.0.0) AL=PI/2.                                         00015780
      IF(ER(1).LT.0.0) AL=-PI/2.                                        00015790
      IF(ER(1).EQ.0.0) STOP                                             00015800
C                                                                       00015810
C STEP 1-2: COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH AND             00015820
C           ELEVATION RATES.                                            00015830
C   PRELIMINARY TRIGONOMETRIC COMPUTATIONS.                             00015840
    9 CA=COS(AL)                                                        00015850
      SA=SIN(AL)                                                        00015860
      CB=COS(BT)                                                        00015870
      SB=SIN(BT)                                                        00015880
C    TRANSFORM BODY ANGULAR VELOCITY VECTOR FROM BODY TO OUTER          00015890
C    GIMBAL(G) REFERENCE FRAME.                                         00015900
      WGX=CP*EWB(1)+SP*EWB(2)                                           00015910
      WGY=CA*(-SP*EWB(1)+CP*EWB(2))+SA*EWB(3)                           00015920
      WGZ=-SA*(-SP*EWB(1)+CP*EWB(2))+CA*EWB(3)                          00015930
C    COMPUTE THE RANGE TO TARGET.                                       00015940
      R=SQRT(ER(1)*ER(1)+ER(2)*ER(2)+ER(3)*ER(3))                       00015950
      YZR=SQRT(ER(2)*ER(2)+ER(3)*ER(3))
C    COMPUTE RANGE RATE TO TARGET
      ARDOT=(ER(1)*EV(1)+ER(2)*EV(2)+ER(3)*EV(3))/R
C    COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH RATE(AZRATE).          00015960
      VGY=CA*EV(2)+SA*EV(3)                                             00015970
      AZRATE=VGY/R+(CB*WGX-SB*WGZ)                                      000159
C    COMPUTE INITIAL TARGET INERTIAL LOS ELEVATION RATE(ELRATE).        00015990
      ELRATE=-(CB*EV(1)-SB*(-SA*EV(2)+CA*EV(3)))/R+WGY                  00016000
C ****************************************                              00026710
C * STEP 1: UPDATE ROUGH RANGE RATE ESTIMATE *                          00026720
C ****************************************                              00026730
C                                                                       00025530
C ***************************************************************       00025540
C * STEP 1: UPDATE ANTENNA LOS-TO-BODY TRANSFORMATION (NOTE: TRANS- *   00025550
C *         FORMATION INCLUDES GIMBAL BIAS ERRORS AND RADAR YAW     *   00025560
C *         ANGLE ERROR WRT BODY FRAME).                            *   00025570
C ***************************************************************       00025580
      CALL GAMMA(TX1,-(BT+BTBIAS))                                      00025590
      CALL THETA(TX2,-(AL+ALBIAS))                                      00025600
      CALL MULT33(TX2,TX1,TX3)                                          00025610
      CALL PHI(TX2,-PSI)                                                00025620
      CALL MULT33(TX2,TX3,TBL)                                          00025630
C                                                                       00025640
C **********************************************************            00026140
C * STEP 6: TRANSFORM TARGET ANGLES AND INERTIAL ANGLE RATES TO *       00026150
C *         BODY FRAME FOR USE IN DISPLAYS AND G AND N.         *       00026160
C **********************************************************            00026170
C NOTE: TRANSFORMATION TBL INCLUDES GIMBAL BIAS ERRORS AND RADAR YAW    00026180
C       ANGLE ERROR WRT BODY FRAME.                                     00026190
C UPDATE TARGET INERTIAL PITCH RATE IN ORBITER BODY COORDINATES         00026200
C FOR DISPLAY.                                                          00026210
      SPRTE=-1000.*(TBL(2,1)*AZRATE+TBL(2,2)*ELRATE)                    00026220
C UPDATE TARGET INERTIAL ROLL RATE IN ORBITER BODY COORDINATES          00026230
C FOR DISPLAY.                                                          00026240
      SRRTE=-1000.*(TBL(1,1)*AZRATE+TBL(1,2)*ELRATE)                    00026250
C UPDATE ANTENNA PITCH ANGLE IN ORBITER BODY COORDINATES FOR DISPLAY.   00026260
      SPANG=-ASIN(TBL(1,3))*57.29576                                    00026270
C UPDATE ANTENNA IN ORBITER BODY COORDINATES FOR DISPLAY.               00026280
      IF(TBL(2,3).EQ.0.0.AND.TBL(3,3).EQ.0.0) GO TO 5                   00026290
      SRANG=-ATAN2(-TBL(2,3).TBL(3,3))*57.29576                         00026300
      GO TO 7                                                           00026310
    5 IF(TBL(1,3).GT.0.0) SRANG=-90.0                                   00026320
      IF(TBL(1,3).LT.0.0) SRANG=90.0                                    00026330
      IF(TBL(1,3).EQ.0.0) STOP                                          00026340
C RESOLVE POSSIBLE ANGLE AMBIGUITIES, VIZ., -90.<SPANG<90. AND          00026350
C -180.<SRANG<180.                                                      00026360
    7 IF(SPANG.LE.90.) GO TO 10                                         00026370
      SPANG=-(180.-ABS(SPANG))*(SPANG/ABS(SPANG))                       00026380
      SRANG=(180.-ABS(SRANG))*(SRANG/ABS(SRANG))                        00026390
   10 CONTINUE                                                          00026400
      RETURN                                                            00026510
      END                                                               00026520
C                                                                       00029600
C                                                                       00029610
C *************************************************************         00029620
C * THIS SUBROUTINE INITIALIZES ALL DATA REQUIRED BY THE SEARCH, *      00029630
```

```
C  * ACQUISITION, AND TRACK SUBPROGRAMS.                               *
C  ********************************************************************
C
C
C
C  ********************************************************************
C  * THIS SUBROUTINE INITIALIZES ALL DATA REQUIRED BY THE SEARCH,     *
C  * ACQUISITION, AND TRACK SUBPROGRAMS.                               *
C  ********************************************************************
C
C
      SUBROUTINE DATA
      REAL IDUM1
      COMMON /RTDAT/IDUM1(2),RBIAS,DUM1(9)
      COMMON /SYSDAT/TSAM,DR(3),CP,SP,PSI,PSBIAS,ALBIAS,BTBIAS,GP,GA,
     2               TGTSIG,GPS,GAS,TRB(3,3)
      COMMON /NOISE/NS1,NS2,NN(10),GAUSS(320)
      DIMENSION A(3,3),B(3,3),C(3,3)
      REAL LT,KTS
C  *********************
C  * SYSTEM PARAMETERS *
C  *********************
      PI=3.1415926
      PII=PI/180.
C  RADAR FRAME YAW ANGLE IN BODY COORDINATES (DEGREES).
      PSI=PII*67.0
      CP=COS(PSI)
      SP=SIN(PSI)
C  RADAR LOCATION OFFSET FROM ORBITER C.G. IN BODY COORD. (FEET)
C **** VALUES MODIFIED MAR 24 83 PER FM8 MEMO ***********
      DR(1)=45.738
      DR(2)=11.130
      DR(3)=-5.79
C  RANGE BIAS ERROR IS COMPUTED IN SUBROUTINE RTRACK AS
C  FUNCTION OF RANGE
C  ALPHA GIMBAL BIAS.
      ALBIAS=0.0
C  BETA GIMBAL BIAS.
      BTBIAS=0.0
C
C  RADAR PLATFORM ORIENTATION ERRORS WITH RESPECT TO BODY FRAME.
C
C  YAW ANGLE ERROR.
      PSBIAS=PII*0.1
C
C  ROLL ANGLE ERROR.
      RLBIAS=PII*0.25
C  PITCH ANGLE ERROR.
      PTBIAS=PII*0.25
C
C NBIAS=0 FOR N0 BIAS AND RADAR AT ORGIN
C
       NBIAS=1
       IF(NBIAS.NE.0)GO TO 700
701     FORMAT(' ALL ANGLE  BIAS SET TO ZERO RADAR AT ORGIN')
       DO 4 I=1,3
 4      DR(I)=0.0
C        PSI=0.0
        PSBIAS=0.0
        RLBIAS=0.0
        PTBIAS=0.0
 700        CONTINUE
C
C  COMPUTE MATRIX OF TRANSFORMATION FROM BODY FRAME TO RADAR FRAME.
```

```
      CALL PHI(B,PSI+PSBIAS)
      CALL THETA(A,RLBIAS)
      CALL MULT33(A,B,C)
      CALL GAMMA(A,PTBIAS)
      CALL MULT33(A,C,TRB)
C  **************************
C  * SYSTEM SAMPLE INTERVAL *
C  **************************
C
C  ************************
C  * COMPUTE SNR CONSTANT *
C  ************************
C  EQUIVALENT ONE-SIDED NOISE POWER SPECTRAL DENSITY (MW/KHZ)
      KTS=-137.5
      KTS=10.**(0.1*KTS)
C  SYSTEM LOSSES ON TRANSMIT (DB).
      LT=2.5
      LT=10.**(0.1*LT)
C  ONE-WAY ANTENNA GAIN (DB).
      G=37.7
      G=10.**(0.1*G)
      ALMBDA=0.070845
C  CONSTANT FOR PASSIVE TRACKING SNR COMPUTATION.
      GP=4.*(G**2)*(ALMBDA**2)/((4.*PI)**3*LT*KTS)
C  BEACON PARAMETER (DBM)
      BCN=44.0
      BCN=10.**(0.1*BCN)
C  CONSTANT FOR ACTIVE TRACKING SNR COMPUTATION.
      GA=4.*G*ALMBDA**2*BCN/((4.*PI)**2*KTS)
C  CONSTANT FOR PASSIVE MODE VIDEO SNR COMPUTATION (DB).
      GPS=183.9
C  CONSTANT FOR ACTIVE MODE VIDEO SNR COMPUTATION (DB).
      GAS=146.9
C
C  *********************************
C  * RANDOM NUMBER GENERATOR SEEDS *
C  *********************************
      NS1=48
      NS2=135
      NN(1)=0
C  INITIALIZE NOISE SEQUENCE.
      DO 2 I=1,320
    2 GAUSS(I)=ANORM(NS1,NS2)
     IF(ITEST.EQ.2)GO TO6341
     ITEST=2
C    WRITE(6,592)
 592 FORMAT(1H1,' RANDOM NUMBER INITIALIZATION')
C    WRITE(6,593)(GAUSS(I),I=1,320)
 593 FORMAT(8F8.4)
C    WRITE(6,592)
6341     CONTINUE
C
C  ****************************
C  * DEFINE TARGET PARAMETERS *
C  ****************************
C  TARGET SEARCH CROSS-SECTION ( FIXED TEMPORARILY).
      TGTSIG=10.0
      RETURN
      END
       SUBROUTINE SETIT
       COMMON /TARGET/ITARG,SRCS
       COMMON /LEN1/ANGOFF
      COMMON /SATDAT/RADAR(3),KTAR,R(70,3),SIG(70),ROLD,
     1 ICLOSE,ICOLD,JHOT(60)
```

```
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,ISRCHC,ISRCHG,IAZS,IELS,ISLR,
     2        EDRNG,EDPA,EDRA
      COMMON /ICNTL/IOLDPW,IOLDMD,IOLDSM,ISHOLD,KMSCLK,KWMUP,KSNCLK,
     2             KSNMAX,KACCLK,MTP,MZ1,MZ0,MSS,MTKINT,MRNG,MSAM,MPRF,
     3              MBKTRK,MBTSUM,MBT(8)
      COMMON /OUTPUT/MSWF,MTF,MSF,SRNG,SRDOT,SPANG,SRANG,SPRTE,
     2               SRRTE,SRSS,MADVF,MRDVF,MARDVF,MRRDVF
      COMMON /INPUT/ERTO(3),EVTO(3),EWB(3),TBT(3,3),TBTD(3,3)
      COMMON /ATDAT/DUM1(10),PREF,RREF
      COMMON /SYSDAT/TS,DUM2(14)
      COMMON /CGMAIN/RO(3),VO(3),AO(3)
      COMMON /DSCRM/DUM3(6),SIGBAR,SNRD,SIGDB
      COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD
C   ITARG = 0   POINT TARGET RCS OF POINT TARGET
C               SRCS IS VARIABLE NAME OF RCS VALUE
C               SRCS = 3.27 IS IMSQ TARGET.
C    SRCS=3.27
       DO I=1,3
       DO J=1,3
       EWB(I)=0.
       TBT(I,J)=0
       IF(I.EQ.J)TBT(I,J)=1.
       TBTD(I,J)=0.
       ENDDO
       ENDDO
       KOLD=-1
      CALL SYSINT
      IPWR=3
      IMODE=2
      IASM=1
      ITXP=1
      ISRCH=0
       IAZS=0
       IELS=0
       ISLR=0
       ISRCHG=0
      EDRNG=500.0
      EDRA=0.0
      EDPA=0.0
      PII=3.14159265/180.
      EDPA=EDPA*PII
      EDRA=EDRA*PII
      MTF=0
      MTP=1
      MTP=1
       RETURN
       END
C
C
C   **************************************************************
C   * THIS FUNCTION GENERATES A RANDOM NUMBER FROM A GAUSSIAN PDF *
C   * WITH ZERO MEAN AND UNIT VARIANCE.                           *
C   **************************************************************
C
C
      FUNCTION ANORM(K1,K2)
      Y1=RNDU(K1)
      Y2=RNDU(K2)
      TPI=6.2831852
      ANORM=SQRT(-2.*ALOG(Y1))*COS(TPI*Y2)
      RETURN
      END
C
C
```

```
C  ********************************************************     00025250
C  * THIS SUBROUTINE UPDATES AZ AND EL INERTIAL LOS RATES, THE *  00025260
C  * ALPHA AND BETA GIMBAL RATES, THE ALPHA AND BETA GIMBAL    *  00025270
C  * POSITIONS, AND THE TARGET PITCH AND ROLL ANGLES FOR THE   *  00025280
C  * DISPLAY.                                                  *  00025290
C  ********************************************************     00025300
C                                                                 00025310
C                                                                 00025320
      SUBROUTINE ATRACK                                           00025330
      REAL INTT,IAZDSC,IELDSC                                     00025335
      COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)                    00025350
      COMMON /INPUT/DUM(6),EWB(3),DUM2(18)                        00025360
      COMMON /OUTPUT/I1DUM(3),D1DUM(2),SPANG,SRANG,SPRTE,SRRTE,SRSS,  00025370
     2              IDUM1(4),SSALP,SSBET                                  00025380
      COMMON /ICNTL/I2DUM(14),MRNG,MSAM,MPRF,IDUM2(11)            00025390
      COMMON /SYSDAT/TSAM,DR(3),CP,SP,PSI,PSBIAS,ALBIAS,BTBIAS,   00025400
     2              DUM4(5)                                       00025410
      COMMON /ATDAT/CA,SA,CB,SB,AZRATE,ELRATE,ALRATE,BTRATE,AL,BT,  00025420
     2             DUM3(4)                                        00025430
      COMMON /DSCRM/AZDISC,ELDISC,DUM1(7)                         00025440
      DIMENSION AT1(10,2),AT2(10,2),TX1(3,3),TX2(3,3),TX3(3,3),TBL(3,3)00025450
     DIMENSION TDC(3)
      DATA AT1/9*1.5529E-3,2.0106E-4,6*3.9750E-3,1.5529E-3,       00025460
     2         3*2.0106E-4/,AT2/9*6.5907E-3,2.3725E-3,            00025470
     3         6*1.0546E-2,6.5907E-3,3*2.3725E-3/                 00025480
     DATA TDC/0.05122118,0.1195161,0.2561557/
C  DEFINITION:  AT1=KEQ=(WN**2)/(4.*DIFFERENCE PATTERN SLOPE) WHERE   00025490
C                WN IS NATURAL FREQUENCY OF THE LOOP.             00025500
C  DEFINITION:  AT2=KEQ*TAU WHERE TAU IS PROPORTIONAL TO STEP RESPONSE 00025510
C                CONVERGENCE TIME.                                00025520
C                                                                 00026700
     TCON=TSAM/TDC(MPRF)
C  ****************************************                       00026710
C  * STEP 1: UPDATE ROUGH RANGE RATE ESTIMATE *                   00026720
C  ****************************************                       00026730
C                                                                 00025530
C  ***********************************************************    00025540
C  * STEP 1: UPDATE ANTENNA LOS-TO-BODY TRANSFORMATION (NOTE: TRANS- *  00025550
C  *         FORMATION INCLUDES GIMBAL BIAS ERRORS AND RADAR YAW    *  00025560
C  *         ANGLE ERROR WRT BODY FRAME).                           *  00025570
C  ***********************************************************    00025580
      CALL GAMMA(TX1,-(BT+BTBIAS))                                00025590
      CALL THETA(TX2,-(AL+ALBIAS))                                00025600
      CALL MULT33(TX2,TX1,TX3)                                    00025610
      CALL PHI(TX2,-PSI)                                          00025620
      CALL MULT33(TX2,TX3,TBL)                                    00025630
C                                                                 00025640
C  **********************************************************     00025650
C  * STEP 2: UPDATE ESTIMATED TARGET INERTIAL AZIMUTH AND ELEVATION *  00025660
C  *         RATES IN ANTENNA LOS FRAME.                          *  00025670
C  **********************************************************     00025680
C                                                                 00025690
C  QUANTIZE THE ANGLE DISCRIMINANTS TO 3/16 DB.                   00025700
      IAZDSC=INTT(5.333333*AZDISC*TCON+0.5)/TCON
      IELDSC=INTT(5.333333*ELDISC*TCON+0.5)/TCON                  00025720
      IF(IELDSC.GT.255)IELDSC=255
      IF(IAZDSC.GT.255)IAZDSC=255
      IF(IELDSC.LT.-256)IELDSC=-256
      IF(IAZDSC.LT.-256)IAZDSC=-256
      ADSC=0.0431*IAZDSC                                          00025730
      EDSC=0.0431*IELDSC                                          00025740
C  UPDATE ESTIMATED TARGET INERTIAL AZIMUTH RATE.                 00025790
      AZRATE=AZRATE+TSAM*AT1(MRNG,IMODE)*ADSC                     00025800
C  UPDATE ESTIMATED TARGET INERTIAL ELEVATION RATE.               00025810
```

```
      ELRATE=ELRATE+TSAM*AT1(MRNG,IMODE)*EDSC
C
C  ************************************************
C  * STEP 3: UPDATE INNER AND OUTER GIMBAL RATES. *
C  ************************************************
C  COMPUTE REQUIRED COMPONENTS OF ORBITER ANGULAR VELOCITY VECTOR IN
C  OUTER GIMBAL FRAME.
      WGX=CP*EWB(1)+SP*EWB(2)
      WGY=CA*(-SP*EWB(1)+CP*EWB(2))+SA*EWB(3)
      WGZ=-SA*(-SP*EWB(1)+CP*EWB(2))+CA*EWB(3)
C  OUTER GIMBAL RATE.
      IF(ABS(CB).LT.1.0E-6) GO TO 2
      ALRATE=(AZRATE+AT2(MRNG,IMODE)*ADSC+WGZ*SB)/CB-WGX
      GO TO 4
    2 ALRATE=0.
    4 CONTINUE
C  INNER GIMBAL RATE.
      BTRATE=(ELRATE+AT2(MRNG,IMODE)*EDSC)-WGY
C
C  ****************************************************
C  * STEP 4: UPDATE INNER AND OUTER GIMBAL POSITIONS. *
C  ****************************************************
C  OUTER GIMBAL POSITION (ALPHA ANGLE)
      AL=AL+TSAM*ALRATE
C  INNER GIMBAL POSITION (BETA ANGLE)
      BT=BT+TSAM*BTRATE
C
C ADD ALPHA AND BETA TO OUTPUT IN DEG
            SSALP=AL*57.29576
            SSBET=BT*57.29576
C  ***************************************************************
C  * STEP 6: TRANSFORM TARGET ANGLES AND INERTIAL ANGLE RATES TO *
C  *         BODY FRAME FOR USE IN DISPLAYS AND G AND N.         *
C  ***************************************************************
C  NOTE: TRANSFORMATION TBL INCLUDES GIMBAL BIAS ERRORS AND RADAR YAW
C        ANGLE ERROR WRT BODY FRAME.
C  UPDATE TARGET INERTIAL PITCH RATE IN ORBITER BODY COORDINATES
C  FOR DISPLAY.
      SPRTE=-1000.*(TBL(2,1)*AZRATE+TBL(2,2)*ELRATE)
C  UPDATE TARGET INERTIAL ROLL RATE IN ORBITER BODY COORDINATES
C  FOR DISPLAY.
      SRRTE=-1000.*(TBL(1,1)*AZRATE+TBL(1,2)*ELRATE)
C  UPDATE ANTENNA PITCH ANGLE IN ORBITER BODY COORDINATES FOR DISPLAY.
      SPANG=-ASIN(TBL(1,3))*57.29576
C  UPDATE ANTENNA IN ORBITER BODY COORDINATES FOR DISPLAY.
      IF(TBL(2,3).EQ.0.0.AND.TBL(3,3).EQ.0.0) GO TO 5
      SRANG=-ATAN2(-TBL(2,3),TBL(3,3))*57.29576
      GO TO 7
    5 IF(TBL(1,3).GT.0.0) SRANG=-90.0
      IF(TBL(1,3).LT.0.0) SRANG=90.0
      IF(TBL(1,3).EQ.0.0) STOP
C  RESOLVE POSSIBLE ANGLE AMBIGUITIES, VIZ., -90.<SPANG<90. AND
C  -180.<SRANG<180.
    7 IF(SPANG.LE.90.) GO TO 10
      SPANG=-(180.-ABS(SPANG))*(SPANG/ABS(SPANG))
      SRANG=(180.-ABS(SRANG))*(SRANG/ABS(SRANG))
   10 CONTINUE
C
C  NOTE: DEBUGGING PRINT STATEMENTS.
C     WRITE(6,899)
  899 FORMAT(/' ATRACK DEBUGGGING DATA')
C     WRITE(6,900) ALRATE,BTRATE,AZRATE,ELRATE,SRRTE,SPRTE
C     WRITE(6,901) TBL(1,1),TBL(1,2),TBL(2,1),TBL(2,2)
C     WRITE(6,902) AZDISC,ELDISC,ADSC,EDSC
```

```
  900  FORMAT(' ALR,BTR,AZR,ELR,SRR,SPR=',6F10.2)
  901  FORMAT(' TBL 2X2 =',4F10.4)
  902  FORMAT(' AZD,ELD,AD,ED =',4F10.4)
       RETURN
       END
C
C   ********************************************************
C   * THIS SUBROUTINE IMPLEMENTS THE BREAK-TRACK ALGORITHM *
C   ********************************************************
C
C
       SUBROUTINE BRKTRK
       REAL IVMAX,THRSHC,THRSHO,IVDISC,INTT,IODISC
       COMMON /ICNTL/IDUM2(17),MBKTRK,MBTSUM,MBT(8)
       COMMON /DSCRM/DUM(3),VDISC,DUM1,ODISC,DUM2(3)
       DATA IVMAX,THRSHC,THRSHO/51.,14.,-11./
C
C   **************************************************
C   * STEP 1: DETERMINE STATUS OF L-H DISCRETE (FTH) *
C   **************************************************
C
C   STEP 1-1: QUANTIZE THE VELOCITY DISCRIMINANT TO 3/16 DB STEPS.
        IVDISC=INTT(VDISC*5.333333+0.5)
C
C   STEP 1-2: DETERMINE STATUS OF L-H DISCRETE.
        IFTH=0
        IF(ABS(IVDISC).GE.IVMAX) IFTH=1
C
C   *******************************************************
C   * STEP 2: DETERMINE STATUS OF ON-TARGET DISCRETE (OT) *
C   *******************************************************
C
C   STEP 2-1: QUANTIZE THE O-DISCRIMINANT TO 3/16 DB STEPS.
        IODISC=INTT(ODISC*5.333333+0.5)
C
C   STEP 2-2: DETERMINE STATUS OF ON-TARGET DISCRIMINANT.
        IOT=0
        IF(IODISC.GE.THRSHC) IOT=1
C
C   *****************************************************************
C   * STEP 3: DETERMINE STATUS OF ADJACENT ON-TARGET DISCRETE (AOT) *
C   *****************************************************************
        IAOT=0
        IF(IODISC.LE.THRSHO) IAOT=1
C
C   ****************************************************************
C   * STEP 4: COMBINE ABOVE DISCRETES TO DETERMINE STATUS OF ● NO- *
C   *         TARGET● DISCRETE (NOTARG).                           *
C   ****************************************************************
C   DEFINITION: THE NO-TARGET DISCRETE IS HIGH (OR 1) IF THE DISCRETES
C               FTH, OT, AND AOT ARE ALL LOW (OR 0).
        NOTARG=(1-IFTH)*(1-IOT)*(1-IAOT)
C
C   *********************************************************
C   * STEP 5: DETERMINE STATUS OF BREAK-TRACK FLAG (MBKTRK) *
C   *********************************************************
C   DEFINITION: BREAK-TRACK SHALL BE DECLARED IF NOTARG=1 FOR AT
C               LEAST 5 OF THE MOST RECENT 8 DATA CYCLES.
C
C   STEP 5-1: UPDATE MOVING WINDOW-OF-8 SUM (MBTSUM).
        MBTSUM=MBTSUM+(NOTARG-MBT(1))
C
C   STEP 5-2: UPDATE STORAGE REGISTERS.
        DO 10 I=1,7
```

```
   10 MBT(I)=MBT(I+1)
      MBT(8)=NOTARG
C
C  STEP 5-3: DETERMINE STATUS OF BREAK-TRACK FLAG (1=BREAK-TRACK).
      MBKTRK=MBTSUM/5
C
C  NOTE: DEBUGGING PRINT STATEMENTS.
C     WRITE(6,900) IODISC,THRSHO,THRSHC,IVDISC,IVMAX,MBTSUM
  900 FORMAT(' OD,THO,THC,VD,THV,SUM =',6I8)
      RETURN
      END
C
C
C  *****************************************************
C  * THIS SUBROUTINE CONTAINS THE CFAR DETECTION MODEL *
C  *****************************************************
C
C
      SUBROUTINE CFAR
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),EDRNG,DUMC(2)
      COMMON /OUTPUT/MSWF,MTF,MSF,DUM1(7),IDUM1(4)
      COMMON /ICNTL/IDUM2(8),KACCLK,MTP,IDUM3(4),MRNG,MSAM,MPRF
      COMMON /TGTDAT/NT,DUM3(500),RO(3),ROU(3),CGRNGE,CGVEL
      COMMON /DETDAT/SIGMA,CGANG
      DIMENSION RI(6),PW(6),NP(6),FW(3),TPRI(3),TS(2),P(41)
      DATA NRI,NSRCH/6,37 /,C,ALMDA/983.5,0.070845/,RI/2552.,5772.,
     2 11544.,23089.,43747.,57722./,PW/0.122,4.15,8.3,16.6,33.2,66.4/,
     3 NP/1,2,4,8,16,32/,FW/7.7215,3.3090,0.2969/,TS/0.122,2.075/,
     4 TPRI/143.5,334.7,3731.1/
      DATA P/6*0.0,.001,.003,2*.004,.008,.012,.015,.043,.053,.076,.107,
     2 .147,.193,.244,.312,.363,.444,.514,.590..644,.706,.765,.815,.861,
     3 .882,.918,.937,.955,.966,.976,.980,.989,.991,.997,.996/
      PI=3.14159265 .
C
C  ******************************************************************
C  * STEP 1: SET INTERNAL CONTROLS BASED UPON SYSTEM OPERATING MODE *
C  ******************************************************************
C
C  STEP 1-1: GPC MODES OR AUTO/MANUAL MODES"
      IF(IASM.GE.3) GO TO 15
C
C  STEP 1-2: SET INTERNAL CONTROLS FOR APPROPRIATE MODE.
C
C  CONTROL SETTINGS FOR GPC MODES.
C
C  DETERMINE RANGE INTERVAL.
      DO 5 I=1,NRI
      MRNG=I
      IF(RI(I).GT.EDRNG) GO TO 10
    5 CONTINUE
C
C  SET SAMPLE RATE
   10 MSAM=2
C
C  DETERMINE PRF
      MPRF=1
      IF(EDRNG.GE.RI(6)) MPRF=2
      GO TO 20
C
C  CONTROL SETTINGS FOR AUTO/MANUAL MODES.
C
C  SET RANGE INTERVAL.
   15 MRNG=6
C
```

```
C  SET SAMPLE RATE.
      MSAM=2
C
C  SET PRF.
      MPRF=1
C
C  ****************************************************
C  * STEP 2: COMPUTE NOMINAL SNR AT VIDEO FILTER OUTPUT *
C  ****************************************************
   20 SNR=SNRV(SIGMA,CGRNGE)
C
C  ****************************************************
C  * STEP 3: IF NOT SCANNING ADD BEAMSHAPE LOSS TO SNRV *
C  ****************************************************
C
C  STEP 3-1: CHECK SCAN FLAG.
      IF(MSF.EQ.1) GO TO 25
C
C  STEP 3-2: COMPUTE BEAMSHAPE LOSS ---- BASED UPON C.G. POSITION OFF
C            BORESIGHT.
      BETA2=SPAT(CGANG)**2
C
C  STEP 3-3: ADD BEAMSHAPE LOSS TO NOMINAL SNRV, I.E. COMPUTE ACTUAL
C            SNRV.
      SNR=SNR*BETA2
C
C  *****************************************************************
C  * STEP 4: COMPUTE NET PROCESSOR GAIN AND COMBINE WITH SNRV TO FORM *
C  *         SNRD.                                                  *
C  *****************************************************************
C
C  STEP 4-1: COMPUTE RANGE GATE LOSS (RGL) ---- DIFFERS FOR GPC AND
C            AUTO/MANUAL MODES.
C
C  COMPUTE EQUIVALENT RANGE OF XMIT PULSEWIDTH.
   25 CTD2=C*PW(MRNG)/2.
C
C  DETERMINE OPERATING MODE
      IF(IASM.GE.3) GO TO 30
C
C  COMPUTE RGL FOR GPC MODES.
      DEL=ABS(EDRNG-CGRNGE)/CTD2
      IF(DEL.GE.1.5) RGL=0.0
      IF(DEL.GE.0.5.AND.DEL.LT.1.5) RGL=.6666666*(1.5-DEL)**2
      IF(DEL.LT.0.5) RGL=.6666666
      GO TO 35
C
C  COMPUTE RGL FOR AUTO/MANUAL MODES
   30 DEL=ABS(CGRNGE)/CTD2
      DEL1=DEL-INT(DEL)
      IF(DEL.LE.1.0) RGL=DEL*DEL
      IF(DEL.GT.1.0.AND.DEL.LT.4.5.AND.DEL1.LT.0.5)
     2 RGL=(1.0-DEL1)**2
      IF(DEL.GT.1.0.AND.DEL.LT.4.5.AND.DEL1.GE.0.5)
     2 RGL=DEL1*DEL1
C
C  STEP 4-2: COMPUTE NET PRESUM GAIN ---- SAME FOR ALL PASSIVE ANTENNA
C            STEERING MODES.
C
C  COMPUTE DOPPLER FREQUENCY ASSOCIATED WITH TARGET RADIAL VELOCITY
   35 FDOP=-2.*CGVEL/ALMDA*1.0E-06
C
C  COMPUTE ARGUMENT ASSOCIATED WITH TARGET VELOCITY
      ARG=PI*FDOP*TS(MSAM)
```

```
C
C     COMPUTE NET PRESUM GAIN
          PSG=SUM(ARG,NP(MRNG))
C
C     STEP 4-3: COMPUTE NET DOPPLER FILTER GAIN ---- SAME FOR ALL PASSIVE
C               ANTENNA STEERING MODES.
C
C     COMPUTE NUMBER OF DOPPLER FILTER NEAREST TARGET.
          MFIL=MOD(INT(CGVEL/FW(MPRF)+320.5),32)
C
C     COMPUTE ARGUMENT ASSOCIATED WITH TARGET DOPPLER
          ARG=PI*(FLOAT(MFIL)/32.+FDOP*TPRI(MPRF))
C
C     COMPUTE NET DOPPLER FILTER GAIN
          DFG=SUM(ARG,16)
C
C     STEP 4-4: COMPUTE NET PROCESSOR GAIN.
          NPG=RGL*PSG*DFG
C
C     STEP 4-5: COMPUTE SNR AT DOPPLER FILTER OUTPUT
          SNR=SNR*NPG
C
C     *************************************************************
C     * STEP 5: DETERMINE PROBABILITY OF DETECTION BASED UPON SNR *
C     *************************************************************
C
C     STEP 5-1: DETERMINE INDEX TO ACCESS APPROPRIATE CURVE
          IF(IASM.GE.3) GO TO 40
          NCRV=1
          GO TO 45
   40     NCRV=3
C
C     ADJUST INDEX FOR SCANNING
   45     NCRV=NCRV+MSF
C
C     STEP 5-2: CONVERT SNR TO DB.
          IF(SNR.LE.1.0E-08) GO TO 50
          SNR=10.*ALOG10(SNR)
          GO TO 55
   50     SNR=-100.
C
C     STEP 5-3: SNR OUTSIDE (0 DB, +20 DB) INTERVAL" ---- IF SO, SET
C               OUTCOME APPROPRIATELY AND SKIP REMAINING STEPS.
C
C     IF SNRD < 0. DB ---- DECLARE A MISS.
   55     IF(SNR.LE.0.) GO TO 60
C
C     IF SNRD > 20. DB ---- DECLARE A HIT.
          IF(SNR.GT.20.) GO TO 65
C
C     STEP 5-4: COMPUTE INDEX FOR LOOKUP TABLE AND FACTORS FOR LINEAR
C               INTERPOLATION.
          SCALE=(SNR+0.)*2.+1.0000001
          ISNR=INT(SCALE)
          REMAIN=SCALE-FLOAT(ISNR)
C
C     STEP 5-5: DETERMINE PD USING TABLE AND LINEAR (IN DB) INTERPOLATION.
          PROB=P(ISNR)+REMAIN*(P(ISNR+1)-P(ISNR))
C
C     **************************************************
C     * STEP 6: DETERMINE OUTCOME OF DETECTION ATTEMPT *
C     **************************************************
C
          X=RNDU(NSRCH)
```

```
      IF(X.LE.PROB) GO TO 65
C
C ************************************************************
C * STEP 7: SET CONTROLS BASED UPON OUTCOME OF DETECTION ATTEMPT *
C ************************************************************
C
C STEP 7-1: IF NO DETECTION ---- SET TARGET PRESENT FLAG LOW.
   60 MTP=0
      RETURN
C
C STEP 7-2: IF DETECTION SUCCESSFUL ---- SET TARGET PRESENT FLAG
C           HIGH AND INITIALIZE ACQUISITION CLOCK.
   65 MTP=1
      KACCLK=0
      RETURN
      END
C
C
C *****************************************************
C * THIS SUBROUTINE UPDATES ALL RADAR INTERNAL CONTROLS. *
C *****************************************************
C
C
      SUBROUTINE CNTRLS
      REAL INTT,NFIL,IRNG,IRDOT
      COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)
      COMMON /OUTPUT/IDUM0(3),SRNG,SRDOT,DUM2(5),IDUM(4)
      COMMON /ICNTL/I1DUM(14),MRNG,MSAM,MPRF,IDUM1(10),MPFOLD
      COMMON /RTDAT/IRDOT,IRNG,RBIAS,VEST(4),MDF(5)
      DIMENSION RI(10),FW(3)
C RI(4) CHANGED TO 2560 FROM 2552
      DATA RI/120.,240.,780.,2560.,5772.,11544.,23089.,43747.,
     2        57722.,1.8228E+6/
      DATA FW/7.7215,3.3090,0.2969/,NRI/10/
C
C *************************************
C * STEP 1: SET RANGE INTERVAL PARAMETER *
C *************************************
      XRNG=IRNG*0.3125
      DO 60 I=1,NRI
      IF(XRNG.LE.RI(I)) GO TO 70
   60 CONTINUE
   70 MRNG=I
      IF(MRNG.GT.NRI) STOP
C
C ***********************************
C * STEP 2: SET SAMPLE RATE PARAMETER *
C ***********************************
      IF(IMODE.GE.2)  GO TO 74
      IF(MRNG.GT.9) GO TO 72
      MSAM=1
      GO TO 80
72    MSAM=2
      GO TO 80
74    IF(MRNG.GT.4) GO TO 76
      MSAM=1
      GO TO 80
76    MSAM=2
C
C ***************************
C * STEP 3: SET PRF PARAMETER *
C ***************************
C
C STEP 3-1: DETERMINE IF IN ACTIVE OR PASSIVE MODE.
```

```
80     IF(IMODE.GE.2) GO TO 84
C
C  STEP 3-2: DETERMINE CORRECT PRF FOR GIVEN OPERATING MODE.
       IF(MRNG.GT.9) GO TO 82
       MPRF=1
       GO TO 90
82     MPRF=3
       GO TO 90
84     IF(MRNG.GT.9) GO TO 86
       MPRF=1
       GO TO 90
86     MPRF=2
90     CONTINUE
C
C  STEP 3-3: IF PRF HAS CHANGED FROM PREVIOUS DATA CYCLE, THEN
C            RESET THE 5 DOPPLER TRACKING FILTERS ACCORDINGLY.
       IF(MPFOLD.EQ.MPRF) GO TO 96
       NFIL=INTT((-SRDOT/FW(MPRF))+0.5)+31998.
       XX=AMOD(NFIL,32.)
       MDF(1)=INT(XX)
       DO 95 I=1,4
   95  MDF(I+1)=MOD(MDF(1)+I,32)
   96  MPFOLD=MPRF
C
C  NOTE: DEBUGGING PRINT STATEMENTS.
C      WRITE(6,999) MPRF,MPFOLD,MDF(1)
  999  FORMAT(' MPRF,MPFOLD,MDF1 =',3I8)
       RETURN
       END
C
C
C  *********************************************************************
C  * THIS SUBROUTINE PERFORMS THE TARGET DETECTION FUNCTION FOR ACTIVE *
C  * AND PASSIVE MODES AND ALL ANTENNA STEERING MODES.                  *
C  *********************************************************************
C
C
       SUBROUTINE DETECT
       COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),EDRNG,DUMC(2)
       COMMON /ICNTL/IDUM2(9),MTP,IDUM3(17)
       COMMON /SYSDAT/DUM2(12),TGTSIG,GPS,GAS
       COMMON /TGTDAT/NT,DUM3(500),RO(3),ROU(3),CGRNGE,CGVEL
       COMMON /DETDAT/SIGMA,CGANG
C
C  ***********************************************
C  * STEP 1: COMPUTE TARGET PARAMETERS WRT RADAR *
C  ***********************************************
C
C  STEP 1-1: TRANSFORM TARGET C.G. POSITION AND VELOCITY TO LOS FRAME.
       CALL TRNSFM
       CALL PVTRAN
C
C  STEP 1-2: COMPUTE TARGET C.G. ANGLE OFF-BORESIGHT (NON-SCANNING).
CCCCCCCCCCCCCCCCCCCCCCC MOD MAR 24 1983 CCCCCCCCCCCCCCCCCCCCCC
       CGANG=ACOS(-ROU(3))
C
C  STEP 1-3: DETERMINE TARGET CROSS-SECTION.
       SIGMA=TGTSIG
C
C  ****************************************************
C  * STEP 2: PRELIMINARY DETECTION MODE DETERMINATION *
C  ****************************************************
C
C  STEP 2-1: DETERMINE WHETHER ACTIVE OR PASSIVE.
```

```
      IF(IMODE.EQ.1) GO TO 5
C
C  STEP 2-2: GPC MODES OR AUTO/MANUAL MODES"
      IF(IASM.GE.3) GO TO 10
      GO TO 15
C
C  ***************************************
C  * STEP 3: ACTIVE MODE DETECTION PROCESS *
C  ***************************************
C
    5 CALL SINGLE
      RETURN
C
C  ****************************************************
C  * STEP 4: PASSIVE AUTO/MANUAL MODE DETECTION PROCESS *
C  ****************************************************
C
C  STEP 4-1: CHECK SHORT RANGE FIRST ---- CALL SINGLE-HIT DETECTION
C            MODEL.
   10 CALL SINGLE
C
C  STEP 4-2: CHECK FOR SUCCESS IN SINGLE-HIT DETECTION ---- IF NOT SUC
C            CESSFUL, THEN TRY LONG RANGE SEARCH.
      IF(MTP.EQ.0) CALL CFAR
      RETURN
C
C  *********************************************
C  * STEP 5: PASSIVE GPC MODES DETECTION PROCESS *
C  *********************************************
C
C  STEP 5-1: CHECK DESIGNATED RANGE.
   15 IF(EDRNG.GT.2552.) GO TO 20
C
C  STEP 5-2: IF DESIGNATED RANGE < 0.42 NM ---- USE SINGLE-HIT
C            DETECTION MODEL.
      CALL SINGLE
      RETURN
C
C  STEP 5-3: IF DESIGNATED RANGE > 0.42 NM ---- USE CFAR DETECTION MODEL.
   20 CALL CFAR
      RETURN
      END
C
C
C  ****************************************************************
C  * THIS SUBROUTINE ADDS THE EQUIVALENT NOISE TO THE ANGLE, RANGE, *
C  * VELOCITY AND ON-TARGET DISCRIMINANT COMPONENTS AND THEN COM-   *
C  * PUTES THE ANGLE, RANGE, VELOCITY, AND ON-TARGET DISCRIMINANTS. *
C  ****************************************************************
C
C
      SUBROUTINE DISCRM
      REAL LATE,MEAN
      COMMON /OUTPUT/MSWF,MTF,MSF,SRNG,SRDOT,SPANG,SRANG,SPRTE,
     2               SRRTE,SRSS,MADVF,MRDVF,MARDVF,MRRDVF
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)
      COMMON /ICNTL/I3DUM(14),MRNG,MSAM,MPRF,IDUM4(10)
      COMMON /SYSDAT/TSAM,DR(3),CP,SP,PSI,PSBIAS,ALBIAS,BTBIAS,GP.GA,
     2               DUMS(3)
      COMMON /TGTDAT/NT,DUM5(506),CGRNGE,CGVEL
      COMMON /DSCRM/AZDISC,ELDISC,RDISC,VDISC,RRTE,ODISC,SIGBR1,SNRD,
     2               SIGDB
      COMMON /SIGDAT/SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE,DF1,DF5,
     2               DF2,DF4,SIGBAR
```

```
      COMMON /NOISE/NS1,NS2,NN(10),GAUSS(320)                           00022900
      COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD                           00022910
      DIMENSION NFREQ(2),PDIA(2),PDIR(2),PDIV(2),PS(10,2),BN(2),PT(3)   00022920
    2  ,TDC(3)
      DATA NFREQ/1,5/,BN/9772.4,616.6/,PS/9*1.,2.,5*1.,2.,4.,8.,8.,16./00022930
    2        ,PDIA,PDIR,PDIV/1.4142,3.1623,2.0,4.4721,2.8284,6.3246/,   00022940
    3         PT/42658.,3125.,195.3/,QNV/.04166666/                     00022950
      DATA TDC/0.05122118,0.1195161,0.2561557/
C                                                                       00022970
C  NOTE: DEBUGGING PRINT STATEMENTS.                                    00022980
C     WRITE(6,900) SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE                       00022990
C     WRITE(6,901) DF1,DF5,DF2,DF4,SIGBAR                               00023000
  900 FORMAT(' SPZ,SMZ,SPL,SML,E,L =',6F10.2)                           00023010
  901 FORMAT(' DF1,DF5,DF2,DF4,SIG =',5F10.2)                           00023020
C                                                                       00023030
C  ***************************************************************     00023040
C  * STEP 1: COMPUTE CONSTANT USED IN SIGNAL SCALING AND COMPUTATION *  00023050
C  *         OF NOISE STATISTICS.                                  *    00023060
C  ***************************************************************     00023070
C
C
      TCON=(TSAM/TDC(MPRF))**.5
C                                                                       00023080
C  STEP 1-1: COMPUTE CONSTANT (NOTE: IT IS DIFFERENT FOR ACTIVE AND     00023090
C            PASSIVE MODES).                                            00023100
      IF(IMODE.EQ.2) GO TO 5                                            00023120
C  NOTE: THIS IS THE CONSTANT USED IN ACTIVE MODE.                      00023130
      YY=GA*PS(MRNG,IMODE)/(CGRNGE**2*BN(MSAM))                         00023140
      S1=YY/FLOAT(NFREQ(IMODE))                                         00023150
      GO TO 10                                                          00023160
C  NOTE: THIS IS THE CONSTANT USED IN PASSIVE MODE.                     00023170
CCCCCCCCCCCCCCCCCCCCCCCCCCC MODS 2-15-83 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
5       CONTINUE
         PTFIX=PT(ITXP)
         IF(SRNG.LT.640.)PTFIX=4.2
         ISTS7=0
         IF(ISTS7.EQ.1)PTFIX=4.2
C
      YY=GP*PS(MRNG,IMODE)*PTFIX   /(CGRNGE**4*BN(MSAM))                00023180
      S1=YY/FLOAT(NFREQ(IMODE))                                         00023190
C                                                                       00023200
C .STEP 1-2: COMPUTE PEAK SIGNAL POWER TO AVERAGE THERMAL NOISE POWER   00023210
C            AT DOPPLER FILTER OUTPUT.                                  00023220
   10 SNRDT=YY*SIGBAR                                                   00023230
C      WRITE(6,221)YY,SIGBAR
C  221     FORMAT('YY,SIGBAR =',F14.5)
      SNRDTD=10.*ALOG10(SNRDT)                                          00023240
      SIGDB=10.*ALOG10(SIGBAR)                                          00023250
      SIGBR1=SIGBAR                                                     00023260
C222  WRITE(6,990) SNRDTD,SIGDB                                         00023262
  990 FORMAT(' SNRDTD,SIGDB =',2F14.2)                                  00023264
C                                                                       00023270
C  STEP 1-3: COMPUTE PEAK SIGNAL POWER TO TOTAL (THERMAL PLUS           00023280
C            QUANTIZATION) NOISE POWER AT THE DOPPLER FILTER OUTPUT.    00023290
      CALL SATNSE(SNF)                                                  00023292
      XX=SNF*AGCO                                                       00023294
      XX=XX/(XX+QNV)                                                    00023296
      S1=S1*XX                                                          00023300
      YY=YY*XX                                                          00023310
      SNRD=YY*SIGBAR                                                    00023320
      SNRD=10.*ALOG10(SNRD)                                             00023330
C                                                                       00023340
C  STEP 1-4: UPDATE NOISE SEQUENCE.                                     00023350
```

```
      NN(1)=MOD(NN(1)+1,320)+1
      DO 15 I=2,10
   15 NN(I)=MOD(NN(I-1)+29,320)+1
      ID1=NN(1)
      GAUSS(ID1)=ANORM(NS1,NS2)
C
C  *******************************************************
C  * STEP 2: COMPUTE ANGLE DISCRIMINANT (INCLUDES NOISE) *
C  *******************************************************
C
C  STEP 2-1: CHECK ANTENNA STEERING MODE ---- SKIP STEP 2 IF IN
C            GPC-DES OR MANUAL.
CCCCCCCCCCCCCCCCCCCC MOD FEB 16 1983 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 20
C
C  STEP 2-2: COMPUTE ANGLE DISCRIMINANT COMPONENT SCALE FACTOR.
      ASCALE=S1*PDIA(IMODE)
C
C  STEP 2-3: COMPUTE STATISTICS OF ADDITIVE NOISE FOR ANGLE
C            DISCRIMINANT COMPONENTS.
      MEAN=PDIA(IMODE)
      VARPAZ=SQRT(2.*S1*SPAZ+1.)
      VARMAZ=SQRT(2.*S1*SMAZ+1.)
      VARPEL=SQRT(2.*S1*SPEL+1.)
      VARMEL=SQRT(2.*S1*SMEL+1.)
C
C  STEP 2-4: ADD EQUIVALENT NOISE TO ANGLE DISCRIMINANT COMPONENT
C            SIGNALS.
      ID6=NN(6)
      SPAZ=ABS(ASCALE*SPAZ+MEAN+VARPAZ*GAUSS(ID1))
      SMAZ=ABS(ASCALE*SMAZ+MEAN+VARMAZ*GAUSS(ID6))
      ID2=NN(2)
      ID7=NN(7)
      SPEL=ABS(ASCALE*SPEL+MEAN+VARPEL*GAUSS(ID2))
      SMEL=ABS(ASCALE*SMEL+MEAN+VARMEL*GAUSS(ID7))
C
C  STEP 2-5: COMPUTE AZ AND EL DISCRIMINANT COMPONENTS.
      AZDISC=10.*ALOG10(SPAZ/SMAZ)
      ELDISC=10.*ALOG10(SPEL/SMEL)
C      AZDISC=0.
C      ELDISC=0.
C
C  *******************************************************
C  * STEP 3: COMPUTE RANGE DISCRIMINANT (INCLUDES NOISE) *
C  *******************************************************
C
C  STEP 3-1: COMPUTE RANGE DISCRIMINANT COMPONENT SCALE FACTOR.
   20 RSCALE=S1*PDIR(IMODE)
C
C  STEP 3-2: COMPUTE STATISTICS OF ADDITIVE NOISE FOR RANGE
C            DISCRIMINANT.
      MEAN=PDIR(IMODE)
      VARELY=SQRT(2.*S1*EARLY+1.)*TCON
      VARLTE=SQRT(2.*S1*LATE+1.)*TCON
C
C  STEP 3-3: ADD EQUIVALENT NOISE TO RANGE DISCRIMINANT COMPONENT
C            SIGNALS.
      ID3=NN(3)
      ID8=NN(8)
      EARLY=ABS(RSCALE*EARLY+MEAN+VARELY*GAUSS(ID3))
      LATE=ABS(RSCALE*LATE+MEAN+VARLTE*GAUSS(ID8))
C
C  STEP 3-4: COMPUTE RANGE DISCRIMINANT.
      RDISC=10.*ALOG10(LATE/EARLY)
```

```
C
C   ***********************************************************
C   * STEP 4: COMPUTE VELOCITY DISCRIMINANT (INCLUDES NOISE) *
C   ***********************************************************
C
C   STEP 4-1: COMPUTE VELOCITY DISCRIMINANT COMPONENT SCALE FACTOR.
      VSCALE=S1*PDIV(IMODE)
C
C   STEP 4-2: COMPUTE STATISTICS OF ADDITIVE NOISE FOR VELOCITY
C             DISCRIMINANT COMPONENTS.
      MEAN=PDIV(IMODE)
      VARDF2=SQRT(2.*S1*DF2+1.)
      VARDF4=SQRT(2.*S1*DF4+1.)
C
C   STEP 4-3: ADD EQUIVALENT NOISE TO VELOCITY DISCRIMINANT
C             COMPONENT SIGNALS.
      ID4=NN(4)
      ID9=NN(9)
      DF2=ABS(VSCALE*DF2+MEAN+VARDF2*GAUSS(ID4))
      DF4=ABS(VSCALE*DF4+MEAN+VARDF4*GAUSS(ID9))
C
C   STEP 4-4: COMPUTE VELOCITY DISCRIMINANT.
      VDISC=10.*ALOG10(DF2/DF4)
C
C   ****************************************************************
C   * STEP 5: COMPUTE ON-TARGET DISCRIMINANT -- USED FOR BREAK- *
C   *         TRACK AND VELOCITY DATA INVALID DETERMINATION      *
C   ****************************************************************
C
C   STEP 5-1: COMPUTE STATISTICS OF ADDITIVE NOISE FOR OUTER DOPPLER
C             FILTER SIGNALS.
      VARDF1=SQRT(2.*S1*DF1+1.)
      VARDF5=SQRT(2.*S1*DF5+1.)
C
C   STEP 5-2: ADD EQUIVALENT NOISE TO OUTER DOPPLER FILTER SIGNALS.
      ID5=NN(5)
      ID10=NN(10)
      DF1=ABS(VSCALE*DF1+MEAN+VARDF1*GAUSS(ID5))
      DF5=ABS(VSCALE*DF5+MEAN+VARDF5*GAUSS(ID10))
C
C   STEP 5-3: COMPUTE ON-TARGET DISCRIMINANT.
C             NOTE: THE FACTOR OF SQRT(2.) IS DUE TO THE METHOD OF
C                   NORMALIZATION OF DISCRIMINANT COMPONENTS.
      ODISC=10.*ALOG10((EARLY+LATE)*SQRT(2.)/(DF1+DF5))
C
C   NOTE: DEBUGGING PRINT STATEMENTS.
C     WRITE(6,902) AZDISC,ELDISC,RDISC,VDISC,ODISC
C     WRITE(6,903) SNRD,SIGDB,SIGBAR
C     WRITE(6,904) SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE
C     WRITE(6,905) DF1,DF5,DF2,DF4,SIGBAR
 902  FORMAT(/' AZD,ELD,RD,VD,OD =',5F14.6)
 903  FORMAT(' SNRD,SIGDB,SIGBAR =',3F14.6)
 904  FORMAT(' SPZ,SMZ,SPL,SML,E,L+NOISE =',6F10.2)
 905  FORMAT(' DF1,DF5,DF2,DF4,SIG+NOISE =',5F10.2)
      RETURN
      END
C
C
C   ****************************************************************
C   * THIS FUNCTION COMPUTES THE DOPPLER FILTER OUTPUT AMPLITUDE *
C   * AND PHASE FOR AN INPUT SIGNAL OF FREQUENCY X.              *
C   ****************************************************************
C
C
```

```
      COMPLEX FUNCTION DOPFIL(X)                                        00031230
      COMPLEX DENOM,NUMER                                               00031240
      DENOM=1.-CEXP(CMPLX(0.,X))                                        00031250
      DENOM=16.*DENOM                                                   00031260
C  CHECK FOR DENOMINATOR EQUAL TO ZERO.                                 00031270
      XX=CABS(DENOM)                                                    00031280
      IF(XX.GT.1.0E-06) GO TO 10                                        00031290
      DOPFIL=(1.0,0.0)                                                  00031300
      RETURN                                                            00031310
   10 NUMER=1.-CEXP(CMPLX(0.,16.*X))                                    00031320
      DOPFIL=NUMER/DENOM                                                00031330
      RETURN                                                            00031340
      END                                                               00031350
C                                                                       00030650
C                                                                       00030660
C  ******************************************************************** 00030670
C  * THIS FUNCTION GIVES THE ANTENNA DIFFERENCE PATTERN WEIGTHING OF  * 00030680
C  * THE RADAR SIGNAL FOR THE GIVEN ANGLE(IN RADIANS) OFF BORESIGHT.  * 00030690
C  * NOTE: THIS PATTERN IS THE DERIVATIVE OF THE SUM PATTERN          * 00030700
C  ******************************************************************** 00030710
C                                                                       00030720
C                                                                       00030730
      FUNCTION DPAT(X)                                                  00030740
      IF(ABS(X).GT.1.E-4) GO TO 10                                      00030750
      DPAT=-0.6228*X                                                    00030760
      RETURN                                                            00030770
10    Y=93.80*X                                                         00030780
      DPAT=1.1465*(Y*COS(Y)-SIN(Y))/(Y*Y)                               00030790
      RETURN                                                            00030800
      END                                                               00030810
C                                                                       00003000
C                                                                       00003010
C  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00003020
C  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00003030
C  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00003040
C  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00003050
C  $$                                                                 $$00003060
C  $$                                                                 $$00003070
C  $$    PREPARE TO RECEIVE THE PRODUCT YOU'VE BEEN WAITING FOR ...   $$00003080
C  $$                                                                 $$00003090
C  $$        THE ACTUAL SES SPACE SHUTTLE RADAR SIMULATION @@@        $$00003100
C  $$                                                                 $$00003110
C  $$                                                                 $$00003120
C  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00003130
C  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00003140
C  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00003150
C  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00003160
C                                                                       00003170
C                                                                       00003180
C                                                                       00003190
C                                                                       00003200
C                                                                       00003210
C                                                                       00003220
C  *********************************************************            00003230
C  *  EXECUTIVE PROGRAM: INTERFACE WITH PARENT SIMULATION *             00003240
C  *********************************************************            00003250
C                                                                       00003260
C                                                                       00003270
      SUBROUTINE EXEC                                                   00003280
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)                00003290
      COMMON /OUTPUT/MSWF,MTF,MSF,DUM(7),IDUM2(4)                       00003300
      COMMON /ICNTL/IOLDPW,IOLDMD,IOLDSM,ISHOLD,KMSCLK,KWMUP,IDUM1(3),  00003310
     2              MTP,IDUM5(17)                                       00003320
      DATA DATINT/1.0/                                                  00003330
      KWMUP=1
C
C ************************************************
C * STEP 0: INITIALIZE ALL TARGET AND SYSTEM DATA *
C ************************************************
      IF(DATINT.NE.1.0) GO TO 1
       CALL SETIT
      CALL DATA
C     CALL SYSINT
      IOLDPW=IPWR
      DATINT=0.0
    1 II=1
      IF(II.EQ.1) GO TO 30
C
C ************************************
C * STEP 1: CHECK SYSTEM POWER SWITCH *
C ************************************
      IF(IPWR.GT.1) GO TO 5
C IF POWER OFF ---INITIALIZE ALL SYSTEM FLAGS AND CLOCKS.
      KMSCLK=0
      CALL SYSINT
      RETURN
C IF POWER ON --- UPDATE MASTER CLOCK AND DETERMINE OPERATING MODE.
    5 KMSCLK=KMSCLK+1
C
C ***********************************
C * STEP 2: CHECK SYSTEM MODE SWITCH *
C ***********************************
      IF(IMODE.LT.3) GO TO 7
C IF SYSTEM IN COMM(IMODE=3) --- INITIALIZE ALL SYSTEM FLAGS.
      CALL SYSINT
      RETURN
C IF SYSTEM IN RADAR MODE --- CHECK FOR CHANGE IN MODE (I.E. ACTIVE-TO
C -PASSIVE OR PASSIVE-TO-ACTIVE).
    7 IF(IMODE.EQ.IOLDMD) GO TO 10
C IF RADAR MODE CHANGE --- RESET SYSTEM TO SEARCH.
      CALL SYSINT
C UPDATE STATUS OF IOLDMD.
   10 IOLDMD=IMODE
C
C **********************************************
C * STEP 3: DETERMINE WHETHER SYSTEM IN STANDBY *
C **********************************************
      IF(IPWR.GT.2) GO TO 15
      CALL SYSINT
      RETURN
C
C ***************************************************
C * STEP 4: DETERMINE WHETHER WARMUP PERIOD EXCEEDED *
C ***************************************************
   15 IF(KMSCLK.GT.KWMUP) GO TO 20
C IF NOT EXCEEDED --- INITIALIZE ALL SYSTEM FLAGS AND RETURN.
      CALL SYSINT
      RETURN
C IF EXCEEDED --- CONTINUE SYSTEM OPERATING MODE DETERMINATION.
C
C ************************************************************
C * STEP 5: DETERMINE IF THERE HAS BEEN AN ANTENNA STEERING MODE *
C *       CHANGE                                               *
C ************************************************************
   20 IF(IASM.EQ.IOLDSM) GO TO 25
C IF CHANGE HAS OCCURRED --- RESET ALL FLAGS AND GO TO NEW MODE.
      CALL SYSINT
   25 IOLDSM=IASM
```

```
C                                                                       00003970
C ************************************************************         00003980
C * STEP 5: DETERMINE WHETHER SYSTEM IS IN SEARCH AND ACQUISITION *     00003990
C *         OR TRACK MODE.                                        *     00004000
C ************************************************************         00004010
      IF(MTF.EQ.1.OR.MTP.EQ.1) GO TO 30                                 00004020
C IF TRACK FLAG DOWN --- GO TO SEARCH MODE.                             00004030
      CALL SEARCH                                                       00004040
      RETURN                                                            00004050
C IF TRACK FLAG IS UP --- GO TO  TRACK MODE.                            00004060
   30 CALL TRACK                                                        00004070
      RETURN                                                            00004080
      END                                                               00004090
C                                                                       00032440
C                                                                       00032450
C ************************************************************         00032460
C * THIS SUBROUTINE GENERATES A (3X3) MATRIX TGA THAT PRODUCES *        00032470
C * A ROTATION OF GA RADIANS ABOUT THE Y-AXIS.                 *        00032480
C ************************************************************         00032490
C                                                                       00032500
C                                                                       00032510
      SUBROUTINE GAMMA(TGA,GA)                                          00032520
      DIMENSION TGA(3,3)                                                00032530
      DO 10 I=1,3                                                       00032540
      DO 10 J=1,3                                                       00032550
   10 TGA(I,J)=0.0                                                      00032560
      TGA(2,2)=1.0                                                      00032570
      TGA(1,1)=COS(GA)                                                  00032580
      TGA(1,3)=-SIN(GA)                                                 00032590
      TGA(3,3)=TGA(1,1)                                                 00032600
      TGA(3,1)=-TGA(1,3)                                                00032610
      RETURN                                                            00032620
      END                                                               00032630
C                                                                       00031360
C                                                                       00031370
C ***************************************************************       00031380
C * THIS FUNCTION CHECKS FOR NEGATIVE ARGUMENT FOR INT FUNCTION *       00031390
C * AND CORRECTS THE QUANTIZATION PROCEDURE.                    *       00031400
C ***************************************************************       00031410
C                                                                       00031420
C                                                                       00031430
      REAL FUNCTION INTT(Y)                                             00031440
      X=Y                                                               00031450
      IF(X.LT.0.0) X=X-1.0                                              00031460
      INTT=AINT(X)                                                      00031470
      RETURN                                                            00031480
      END                                                               00031490
C                                                                       00031880
C                                                                       00031890
C ***************************************************************       00031900
C * THIS SUBROUTINE MULTIPLIES THE (3X3) MATRIX A AND THE (3X1) *       00031910
C * VECTOR B TO OBTAIN THE (3X1) VECTOR C.                      *       00031920
C ***************************************************************       00031930
C                                                                       00031940
C                                                                       00031950
      SUBROUTINE MULT31(A,B,C)                                          00031960
      DIMENSION A(3,3),B(3),C(3)                                        00031970
      DO 10 I=1,3                                                       00031980
      C(I)=0.0                                                          00031990
      DO 10 J=1,3                                                       00032000
10    C(I) = C(I)+A(I,J)*B(J)                                           00032010
      RETURN                                                            00032020
      END                                                               00032030
C                                                                       00031710
```

```
C                                                                       00031720
C  ***************************************************************      00031730
C  * THIS SUBROUTINE MULTIPLIES THE (3X3) MATRIX A AND THE (3X3) *      00031740
C  * MATRIX B TO OBTAIN THE (3X3) MATRIX C.                      *      00031750
C  ***************************************************************      00031760
C                                                                       00031770
C                                                                       00031780
      SUBROUTINE MULT33(A,B,C)                                          00031790
      DIMENSION A(3,3), B(3,3), C(3,3)                                  00031800
      DO 10 I=1,3                                                       00031810
      DO 10 J=1,3                                                       00031820
      C(I,J)=0.0                                                        00031830
      DO 10 K=1,3                                                       00031840
10    C(I,J) = C(I,J)+A(I,K)*B(K,J)                                     00031850
      RETURN                                                            00031860
      END                                                               00031870
C                                                                       00032240
C                                                                       00032250
C  **************************************************************       00032260
C  * THIS SUBROUTINE GENERATES A (3X3) MATRIX TPH THAT PRODUCES *       00032270
C  * A ROTATION OF PH RADIANS ABOUT THE Z-AXIS.                 *       00032280
C  **************************************************************       00032290
C                                                                       00032300
C                                                                       00032310
      SUBROUTINE PHI(TPH,PH)                                            00032320
      DIMENSION TPH(3,3)                                                00032330
      DO 10 I=1,3                                                       00032340
      DO 10 J=1,3                                                       00032350
10    TPH(I,J)=0.0                                                      00032360
      TPH(3,3)=1.                                                       00032370
      TPH(1,1)=COS(PH)                                                  00032380
      TPH(2,2)=TPH(1,1)                                                 00032390
      TPH(1,2)=SIN(PH)                                                  00032400
      TPH(2,1)=-TPH(1,2)                                                00032410
      RETURN                                                            00032420
      END                                                               00032430
C                                                                       00031500
C                                                                       00031510
C  *****************************************************************    00031520
C  * THIS SUBROUTINE GENERATES A (3X3) MATRIX TPHD THAT REPRESENTS *    00031530
C  * THE DERIVATIVE OF A MATRIX THAT REPRESENTS UNIFORM ROTATION   *    00031540
C  * ABOUT THE Z-AXIS. THE ROTATION SPEED IS W AND THE ANGLE AT    *    00031550
C  * WHICH THE DERIV. IS TAKEN IS PH.                              *    00031560
C  *****************************************************************    00031570
C                                                                       00031580
C                                                                       00031590
      SUBROUTINE PHID(TPHD,PH,W)                                        00031600
      DIMENSION TPHD(3,3)                                               00031610
      DO 10 I=1,3                                                       00031620
      TPHD(3,I)=0.0                                                     00031630
10    TPHD(I,3)=0.0                                                     00031640
      TPHD(1,1)=-W*SIN(PH)                                              00031650
      TPHD(2,2)=TPHD(1,1)                                               00031660
      TPHD(1,2)=W*COS(PH)                                               00031670
      TPHD(2,1)=-TPHD(1,2)                                              00031680
      RETURN                                                            00031690
      END                                                               00031700
C                                                                       00010980
C                                                                       00010990
C  ***************************************************************      00011000
C  * THIS SUBROUTINE UPDATES THE POSITION OF THE ANTENNA GIMBALS *      00011010
C  ***************************************************************      00011020
C                                                                       00011030
C                                                                       00011040
```

```
      SUBROUTINE POINT
      COMMON /OUTPUT/IDUM1(3),DUM4(2),SPANG,SRANG,DUM5(3),IDUM2(4)
      COMMON /SYSDAT/TS,DUM(3),CG,SG,DUM2(9)
      COMMON /ATDAT/DUM1(4),SALRTE,SBTRTE,DUM3(2),AL,BT,PREF,RREF,
     2              AREF,BREF
      DATA AK/2.0/,TAU/1.414/,PI/3.141592653/
C
C   ************************************
C   * STEP 1: PRELIMINARY COMPUTATIONS *
C   ************************************
      CR=COS(-RREF)
      SR=SIN(-RREF)
      CP=COS(-PREF)
      SP=SIN(-PREF)
C
C   **************************************************************
C   * STEP 2: COMPUTE ANTENNA REFERENCE ROLL/PITCH ANGLES IN THE *
C   *         RADAR FRAME.                                       *
C   **************************************************************
      XX=CG*SP-SG*SR*CP
      YY=SG*SP+CG*SR*CP
      ZZ=CR*CP
      IF(YY.EQ.0.0.AND.ZZ.EQ.0.0) GO TO 1
      AREF=ATAN2(YY,ZZ)
      GO TO 2
    1 IF(XX.GT.0.0) AREF=-PI/2.
      IF(XX.LT.0.0) AREF=PI/2.
    2 BREF=ASIN(XX)
C
C   *********************************************************
C   * STEP 3: UPDATE OUTER (ALPHA) GIMBAL RATE AND POSITION *
C   *********************************************************
C   COMPUTE ALPHA LOOP POSITION ERROR.
      ERRA=AREF-AL
C   UPDATE SMOOTHED ALPHA GIMBAL RATE ESTIMATE.
      SALRTE=SALRTE+TS*AK*ERRA
C   UPDATE ALPHA GIMBAL RATE.
      ALRATE=AK*TAU*ERRA+SALRTE
C   CHECK FOR ALPHA GIMBAL RATE LIMITING.
      IF(ABS(ALRATE).GT.56.) ALRATE=56.*ALRATE/ABS(ALRATE)
C   UPDATE ALPHA GIMBAL POSITION.
      AL=AL+TS*ALRATE
C
C   ********************************************************
C   * STEP 4: UPDATE INNER (BETA) GIMBAL RATE AND POSITION *
C   ********************************************************
C   COMPUTE BETA LOOP POSITION ERROR.
      ERRB=BREF-BT
C   UPDATE SMOOTHED BETA GIMBAL RATE ESTIMATE.
      SBTRTE=SBTRTE+TS*AK*ERRB
C   UPDATE BETA GIMBAL RATE.
      BTRATE=AK*TAU*ERRB+SBTRTE
C   CHECK FOR BETA GIMBAL RATE LIMITING.
      IF(ABS(BTRATE).GT.56.) BTRATE=56.*BTRATE/ABS(BTRATE)
C   UPDATE BETA GIMBAL POSITION.
      BT=BT+TS*BTRATE
C
C   ********************************************
C   * STEP 5 :  ANTENNA IN OBSCURATION REGION" *
C   ********************************************
C     CALL SCNWRN
C
C   ***************************************************************
C   * STEP 6: COMPUTE ANTENNA ROLL/PITCH ANGLES IN THE BODY FRAME *
```

```
C   ************************************************************
         CA=COS(AL)
         SA=SIN(AL)
         CB=COS(BT)
         SB=SIN(BT)
         XX=CA*SB+SG*SA*CB
         YY=-SG*SB+CG*SA*CB
         ZZ=CA*CB
         IF(YY.EQ.0.0.AND.ZZ.EQ.0.0) GO TO 3
         SRANG=-57.29576*ATAN2(YY,ZZ)
         GO TO 4
     3   IF(XX.GT.0.0) SRANG=+90.0
         IF(XX.LT.0.0) SRANG=-90.0
     4   SPANG=-57.29576*ASIN(XX)
C   RESOLVE POSSIBLE ANGLE AMBIGUITIES, VIZ., -90.<SPANG<90. AND
C   -180.<SRANG<180.
         IF(SPANG.LE.90.) GO TO 10
         SPANG=-(180.-ABS(SPANG))*(SPANG/ABS(SPANG))
         SRANG=(180.-ABS(SRANG))*(SRANG/ABS(SRANG))
    10   RETURN
         END
C
C
C   ***********************************************************
C   * THIS SUBROUTINE COMPUTES TARGET C.G. POSITION AND VELOCITY *
C   * WRT ANTENNA LOS COORDINATES AND INDIVIDUAL SCATTERER POSI- *
C   * TIONS AND VELOCITIES WRT ANTENNA LOS COORDINATES.          *
C   ***********************************************************
C
C
         SUBROUTINE PVTRAN
          COMMON /TEST1/RA(3)
         COMMON /CNTL/IPWR,IMODE
         COMMON /INPUT/ERT(3),EVT(3),DUM(21)
         COMMON /OUTPUT/MSWF,MTF,MSF,DUMO(7),IDUMO(4)
         COMMON /ICNTL/IDUM6(9),MTP,IDUM7(3),MTKINT
         COMMON /SYSDAT/TSAM,DR(3),DUM2(11)
         COMMON /TGTDAT/NT,RAU(3,100),RANGE(100),RADVEL(100),RO(3),
      2            ROU(3),CGRNGE,CGVEL
         COMMON /SATDAT/RADAR(3),N20,RT(70,3),SIG(70),ROLD,ICLOSE,ICLOLD
         COMMON /XFORMS/TLB(3,3),TLBD(3,3),TLT(3,3),TLTD(3,3)
         COMMON /TARGET/ ITARG,SRCS
         DIMENSION ROR(3),ROD(3),V1(3),RL(3),RAD(3),RLD(3),XRT(3)
C   ***********************************************************
C   * STEP 1: COMPUTE TARGET C.G. POSITION IN ANTENNA LOS FRAME *
C   ***********************************************************
C
C   STEP 1-1: ADD RADAR OFFSET IN ORBITER BODY FRAME.
         DO 5 I=1,3
     5   ROR(I)=ERT(I)-DR(I)
C
C   STEP 1-2: TRANSFORM TARGET C.G. POSITION FROM BODY FRAME TO
C             ANTENNA LOS FRAME.
         CALL MULT31(TLB,ROR,RO)
C
C   STEP 1-3: COMPUTE RANGE OF TARGET C.G. WRT RADAR.
         CGRNGE=SQRT(RO(1)*RO(1)+RO(2)*RO(2)+RO(3)*RO(3))
C
C   STEP 1-4: COMPUTE UNIT VECTOR IN DIRECTION OF TARGET C.G. WRT
C             ANTENNA LOS FRAME.
         DO 10 I=1,3
    10   ROU(I)=RO(!)/CGRNGE
C
C   ************************************************************
```

```
C  * STEP 2: COMPUTE TARGET C.G. RADIAL VELOCITY WRT ANTENNA LOS *
C  *         FRAME (OR RADAR).                                    *
C  ****************************************************************
C
C  STEP 2-1: COMPUTE TARGET C.G. VELOCITY COMPONENTS WRT ANTENNA
C            LOS FRAME.
      CALL MULT31(TLBD,ROR,V1)
      CALL MULT31(TLB,EVT,ROD)
      DO 15 I=1,3
   15 ROD(I)=ROD(I)+V1(I)
C
C  STEP 2-2: COMPUTE TARGET C.G. RADIAL VELOCITY WRT ANTENNA LOS.
      CGVEL=0.0
      DO 20 I=1,3
   20 CGVEL=CGVEL+ROD(I)*ROU(I)
C
C  ***************************************************************
C  * STEP 3: COMPUTE TARGET SCATTERING CHARACTERISTICS -- = OF  *
C  *         ILLUMINATED POINTS, THE POINT LOCATIONS, AND THE   *
C  *         RCS FOR EACH POINT.                                 *
C  ***************************************************************
C
C  STEP 3-1: IF IN ACTIVE MODE, SEARCH MODE, OR TRACKER INITIALIZATION
C            -- ASSUME SINGLE SCATTERER LOCATED AT TARGET FRAME ORIGIN.
C
C ITARG=0 POINT TARGET
C ITARG=1 SPAS
C ITARG=2 SMM
      IF(ITARG.EQ.0) GO TO 24
C  CHECK CONDITION.
      IF(IMODE.NE.1.AND.MTKINT.NE.0.AND.MTP.NE.0) GO TO 30
C  IF ABOVE CONDITION TRUE -- THEN SET PARAMETERS AS FOLLOWS AND DO
C  NOT CALL TARGET MODEL.
   24 NT=1
      SIG(1)=SRCS
      DO 25 I=1,3
   25 RT(1,I)=0.0
C
C  STEP 3-2: COMPUTE LOCATION OF RADAR IN TARGET FRAME.
   30 DO 35 I=1,3
      RADAR(I)=0.0
      DO 35 J=1,3
   35 RADAR(I)=RADAR(I)-TLT(J,I)*RO(J)
      IF(ITARG.EQ.0)GO TO 40
C
C  STEP 3-3: COMPUTE TARGET SCATTERING CHARACTERISTICS.
     IF(ITARG.EQ.2)CALL SMM
      IF(ITARG.EQ.1)CALL SPAS
      NT=N20
C
   40 DO 70 K=1,NT
C
C  ****************************************************************
C  * STEP 4: COMPUTE KTH SCATTERER POSITION, RANGE, AND DIRECTION *
C  *         VECTOR WRT ANTENNA LOS FRAME (OR RADAR).             *
C  ****************************************************************
C
C  STEP 4-1: COMPUTE KTH SCATTERER POSITION WRT ANTENNA LOS FRAME.
      DO 45 J=1,3
      RL(J)=0.0
      DO 45 I=1,3
   45 RL(J)=RL(J)+TLT(J,I)*RT(K,I)
      DO 50 I=1,3
   50 RA(I)=RO(I)+RL(I)
```

```
C
C  STEP 4-2: COMPUTE RANGE OF KTH SCATTERER WRT RADAR.
      RANGE(K)=SQRT(RA(1)*RA(1)+ RA(2)*RA(2)+RA(3)*RA(3))
C
C  STEP 4-3: COMPUTE UNIT VECTOR IN DIRECTION OF KTH SCATTERER WRT
C            ANTENNA LOS FRAME.
      DO 55 I=1,3
   55 RAU(I,K)=RA(I)/RANGE(K)
C
C  ******************************************************
C  * STEP 5: COMPUTE KTH SCATTERER RADIAL VELOCITY WRT RADAR *
C  ******************************************************
C
C  STEP 5-1: COMPUTE KTH SCATTERER VELOCITY COMPONENTS WRT ANTENNA
C            LOS FRAME.
      DO 58 I=1,3
   58 XRT(I)=RT(K,I)
      CALL MULT31(TLTD,XRT,RLD)
      DO 60 I=1,3
   60 RAD(I)=ROD(I)+RLD(I)
C
C  STEP 5-2: COMPUTE KTH SCATTERER RADIAL VELOCITY WRT TO RADAR.
      RADVEL(K)=0.0
      DO 65 I=1,3
   65 RADVEL(K)=RADVEL(K)+RAD(I)*RAU(I,K)
   70 CONTINUE
C
C  NOTE: DEBUGGING PRINT STATEMENTS.
C     WRITE(6,900) RO(1),RO(2),RO(3),CGRNGE,CGVEL
C     WRITE(6,901) RAU(1,1),RAU(2,1),RAU(3,1),RANGE(1),RADVEL(1)
C     WRITE(6,902)
C     WRITE(6,903)(I,(RT(I,J),J=1,3),SIG(I),I=1,N20)
  900 FORMAT(//' RO1,RO2,RO3,CGR,CGV =',5F10.2)
  901 FORMAT(' RAU1,RAU2,RAU3,R,V =',5F10.2)
  902 FORMAT(' SPAS RCS DATA:',/,
     1 /,9X,'I',4X,'R(I,1)',4X,'R(I,2)',4X,'R(I,3)',9X,'SIG(I)',/)
  903 FORMAT(I10,3F10.2,F15.1)
      RETURN
      END
C
C
C  ***********************************************************
C  * THIS FUNCTION GENERATES A RANDOM NUMBER FROM A UNIFORM ●0,1● *
C  * DISTRIBUTION                                             *
C  ***********************************************************
      FUNCTION RNDU(IRAN)
      DATA MU/524287/,IETA/997/
      IF(IRAN.EQ.0)  GO TO 10
      IRAN=IETA*IRAN
      IKEEP=IRAN/MU
      IRAN=IRAN-IKEEP*MU
      XRAN=IRAN
      XRAN=XRAN/MU
      RNDU=XRAN
   10 RETURN
      END
C
C
C  ************************************************************
C  * THIS SUBROUTINE COMPUTES THE RADAR SIGNAL STRENGTH AND UPDATES *
C  * THE AGC SETTING.                                               *
C  ************************************************************
C
C
```

```
      SUBROUTINE RSS                                                    00029300
      COMMON /CNTL/IPWR,IMODE,IDUM1(7),DUM1(3)                          00029310
      COMMON /ICNTL/IDUM2(14),MRNG,IDUM6(12)                            00029320
      COMMON /OUTPUT/IDUM7(3),DUM3(6),SRSS,IDUM4(4)                     00029330
      COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD                           00029340
      DIMENSION PS(10,2)                                                00029350
      DATA PS/9*1.,2.,5*1.,2.,4.,8.,8.,16./,QNV/0.04166666/             00029360
C                                                                       00029370
C  *****************************                                        00029380
C  * STEP 1: UPDATE SYSTEM AGC *                                        00029390
C  *****************************                                        00029400
C                                                                       00029410
C STEP 1-1: COMPUTE AGC ERROR AND CHECK LIMITS.                         00029420
      AGCERR=4.*PS(MRNG,IMODE)/(AGCO*(SNRDT+1.0)+QNV)                   00029430
      IF(AGCERR.GT.10.) AGCERR=10.0                                     00029440
      IF(AGCERR.LT.0.1) AGCERR=0.1                                      00029450
C                                                                       00029460
C STEP 1-2: COMPUTE NEW AGC VALUE AND CHECK LIMITS.                     00029470
      AGCO=AGCERR*AGCO                                                  00029480
      IF(AGCO.GT.1.0) AGCO=1.0                                          00029490
      AGCODB=10.*ALOG10(AGCO)                                           00029500
C                                                                       00029510
C  **********************************************                       00029520
C  * STEP 2: UPDATE RADAR SIGNAL STRENGTH VALUE *                       00029530
C  **********************************************                       00029540
      IF(AGCO.LT.1.0E-15) AGCO=1.0E-15                                  00029550
      SRSS=1./AGCO                                                      00029560
      SRSS=10.*ALOG10(SRSS)                                             00029570
      RETURN                                                            00029580
      END                                                               00029590
C                                                                       00026530
C                                                                       00026540
C  ***********************************************************          00026550
C  * THIS SUBROUTINE UPDATES RANGE AND RANGE RATE ESTIMATES. *          00026560
C  ***********************************************************          00026570
C                                                                       00026580
C                                                                       00026590
      SUBROUTINE RTRACK                                                 00026600
      REAL INTT,IRDISC,IRNG,IRDOT                                       00026605
      COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)                          00026610
      COMMON /OUTPUT/IDUM0(3),SRNG,SRDOT,DUM2(5),IDUM(4)                00026620
      COMMON /ICNTL/I1DUM(14),MRNG,MSAM,MPRF,IDUM1(10),MPFOLD           00026630
      COMMON /SYSDAT/TSAM,DUMS(14)                                      00026640
      COMMON /RTDAT/IRDOT,IRNG,RBIAS,VEST(4),MDF(5)                     00026650
      COMMON /DSCRM/DUM(2),RDSC,VDISC,RRTE,ODISC,DUM3(3)                00026660
      DIMENSION RT1(10,2),RT2(10,2),TDC(3),RGBIAS(2)                    00026670
      DATA RT1/9*0.125,0.25,4*0.125,2.,1.,2.,2*0.5,0.25/,RT2/9*0.5,     00026680
     2         4.0,4*0.5,8.,8.,4*16./                                   00026690
     DATA TDC/0.05122118,0.1195161,0.2561557/
      DATA RGBIAS/32.3,94.7                        /
C                                                                       00026700
C  ********************************************                         00026710
C  * STEP 1: UPDATE ROUGH RANGE RATE ESTIMATE *                         00026720
C  ********************************************                         00026730
C                                                                       00026740
C STEP 1-1: INTEGERIZE RANGE DISCRIMINANT AND CHECK FOR SATURATION.     00026750
      RDISC=5.333333*RDSC                                               00026760
      TCON=TSAM/TDC(MPRF)
      IRDISC=INTT(RDISC*TCON+0.5)/TCON                                  00026770
      IF(IRDISC.GT.127.) IRDISC=127.                                    00026780
      IF(IRDISC.LT.-128.) IRDISC=-128.                                  00026790
C                                                                       00026800
C STEP 1-2: COMPUTE ROUGH RANGE RATE PREDICTION FROM ALPHA-BETA         00026810
C            TRACKING EQUATIONS.                                        00026820
```

```
C DEFINITION:  RT1(MRNG,IMODE) CORRESPONDS TO BETA IN ALPHA-BETA TRACK.
      RR1=IRDISC*RT1(MRNG,IMODE)
      IRDOT=IRDOT+INTT(RR1+0.5)
C
C ********************************
C * STEP 2: UPDATE RANGE ESTIMATE *
C ********************************
C
C STEP 2-1: UPDATE RANGE ESTIMATE USING ALPHA-BETA TRACKER EQUATIONS.
C DEFINITION: RT2 CORRESPONDS TO ALPHA IN ALPHA-BETA TRACKER.
      R1=IRDISC*RT2(MRNG,IMODE)
      IRNG=IRNG+IRDOT+INTT(R1+0.5)
C
C STEP 2-2: CONVERT RANGE ESTIMATE (IRNG) TO FEET USING THE FACT THAT
C           THE LSB OF IRNG REPRESENTS 5/16 FEET.
      RNG=0.3125*IRNG
C
C STEP 2-3: ADD FIXED BIAS TO FINAL RANGE ESTIMATE.
      SRNG=RNG+RGBIAS(MSAM)
C
C FORCE BREAK TRACK IF RANGE LESS THAN 100 FT
C
      IF(SRNG.LT.100.)CALL SYSINT
C
      RETURN
      END
C
C
C ***********************************************************
C * THIS SUBROUTINE DETERMINES WHETHER THE SIGNAL PLUS NOISE *
C * IS SATURATING THE A/D --- IF SO,THEN THE SNR AT DOPPLER  *
C * FILTER OUTPUT IS LIMITED TO THE VALUE THAT JUST SATUR-   *
C * ATES THE A/D.                                            *
C ***********************************************************
C
C
      SUBROUTINE SATNSE(SNF)
      COMMON /CNTL/IPWR,IMODE
      COMMON /ICNTL/IDUM(14),MRNG
      COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD
      DIMENSION PS(10,2)
      DATA PS/9*10.0,2.,5*1.,2.,4.,8.,8.,16./
      SNF=1.
      X=AGCO*(SNRDT/(4.*PS(MRNG,IMODE))+1.0)
      X=12.25/X
      IF(X.GT.1) RETURN
      SNF=X
      RETURN
      END
C
C
C ******************************************************************
C * THIS SUBROUTINE  CYCLES THRU THE LOGIC FOR ANY SCAN GENERATION. *
C ******************************************************************
C
C
      SUBROUTINE SCAN
      COMMON /CNTL/IDUM(4),ISRCHC,ISRCHG,IDUMC(3),EDRNG,DUMC(2)
      COMMON /OUTPUT/MSWF,MTF,MSF,DUM1(7),IDUM2(4)
      COMMON /ICNTL/IDUM3(6),KSNCLK,IDUM4(2),MTP,IDUM5(17),MSWTCH,
     2             KSN,IAROLD,ITROLD
      COMMON /SYSDAT/TSAM,DUMS(14)
      COMMON /TGTDAT/NT,DUM2(503),ROU(3),DUM3(2)
      COMMON /ATDAT/DUM4(8),AL,BT,DUM5(2),AREF,BREF
      DIMENSION TIMINT(31),ANGINT(31),RSW(10),TSW(10)                   00012820
      DATA TIMINT/.7,1.4,1.9,2.6,3.4,4.3,5.1,6.,7.,8.,9.1,10.4,11.8,    00012830
     1 13.3,14.9,16.9,18.9,21.1,23.4,25.9,28.6,31.5,33.5,36.6,39.8,     00012840
     2 43.2,46.8,50.5,54.3,58.4,60.0/                                   00012850
      DATA ANGINT/0.,.7,1.5,2.,2.7,3.6,4.4,5.2,6.1,7.,7.9,8.8,9.8,10.9,00012860
     1 11.9,13.0,14.2,15.3,16.5,17.6,18.8,19.9,21.1,22.2,23.4,24.5,     00012870
     2 25.6,26.7,27.8,28.9,30./                                         00012880
      DATA TSW/60.0,54.3,43.2,33.5,28.6,21.1,14.9,11.8,8.0,6.0/,        00012890
     2 RSW/48609.2,55900.6,62584.3,71698.6,91142.5,151903.8,            00012900
     3 243046.0,394949.8,881041.8,1822845.0/                            00012910
      PII=180./3.141592653                                              00012920
C                                                                       00012930
C  ******************************************************************* 00012940
C  * STEP 1: DETERMINE WHETHER TO PERFORM SCAN INITIALIZATION(MSF=0) * 00012950
C  *         OR SCAN UPDATE(MSF=1).                                  * 00012960
C  ******************************************************************* 00012970
      IF(MSF.EQ.1) GO TO 15                                             00012980
C                                                                       00012990
C  ***************************************                             00013000
C  * STEP 2: PERFORM SCAN INITIALIZATION *                             00013010
C  ***************************************                             00013020
C  INITIALIZE ALL FLAGS.                                                00013030
      MSF=1                                                             00013040
C  INITIALIZE RING MONITORS.                                            00013050
      IAROLD=0                                                          00013060
      ITROLD=10                                                         00013070
C  INITIALIZE SCAN CLOCK.                                               00013080
      KSNCLK=0                                                          00013090
C  INITIALIZE SCAN TIME PARAMETER.                                      00013100
      KSN=0                                                             00013110
C                                                                       00013120
C  DETERMINE SWITCH POINT PARAMETER.                                    00013130
      DO 5 I=1,10                                                       00013140
      IF(EDRNG.LT.RSW(I)) GO TO 10                                      00013150
    5 CONTINUE                                                          00013160
   10 MSWTCH=I                                                          00013170
C                                                                       00013180
C  ******************************                                      00013190
C  * STEP 3: UPDATE SCAN CLOCKS *                                      00013200
C  ******************************                                      00013210
C                                                                       00013220
C  STEP 3-1: UPDATE SCAN CLOCK (TRACKS TOTAL ELAPSED TIME FROM SCAN     00013230
C            INITIATION).                                               00013240
   15 KSNCLK=KSNCLK+1                                                   00013250
      T=FLOAT(KSNCLK)*TSAM                                              00013260
C                                                                       00013270
C  STEP 3-2: UPDATE SCAN TIME PARAMETER (USED TO DETERMINE BORESIGHT    00013280
C            POSITION IN SCAN PATTERN).                                 00013290
      IF(T.LE.TSW(MSWTCH)) KSN=KSN+1                                    00013300
      IF(T.GT.TSW(MSWTCH)) KSN=KSN-1                                    00013310
      TSN=FLOAT(KSN)*TSAM                                               00013320
C                                                                       00013330
C  ***********************************************************         00013340
C  * STEP 4: DETERMINE ANTENNA POSITION TO NEAREST SCAN RING *         00013350
C  ***********************************************************         00013360
      DO 20 I=1,31                                                      00013370
      IF(TSN.LT.TIMINT(I)) GO TO 25                                     00013380
   20 CONTINUE                                                          00013390
   25 IARNG=I                                                           00013400
C                                                                       00013410
C  ***********************************************************         00013420
C  * STEP 5: DETERMINE TARGET POSITION IN SCAN PATTERN (SCAN *         00013430
C  *         RING NUMBER FOR TARGET)                         *         00013440
C  ***********************************************************         00013450
```

```
C
C  STEP 5-1: DETERMINE TARGET POSITION EXACTLY.
       ALOLD=AL
       BTOLD=BT
       AL=AREF
       BT=BREF
       CALL TRNSFM
       CALL PVTRAN
       AL=ALOLD
       BT=BTOLD
C
C  STEP 5-2: DETERMINE TARGET SCAN RING NUMBER.
C
C  DETERMINE TARGET ANGLE OFF SCAN DESIGNATES (DEGREES).
CCCCCCCCCCCCCCCCCCCC MOD MAR 24 1983 CCCCCCCCCCCCCCCCCCCC
       CGANG=ACOS(-ROU(3))*PII
C
C  DETERMINE TARGET SCAN RING NUMBER.
       DO 30 I=1,31
       IF(CGANG.LT.ANGINT(I)) GO TO 35
   30  CONTINUE
   35  ITRNG=I
       IF(CGANG.GT.30.) ITRNG=32
C
C  ****************************************************
C  * STEP 6: DETERMINE IF A DETECTION SHOULD BE ATTEMPTED *
C  ****************************************************
C
C  STEP 6-1: CHECK CONDITION.
       IF(IARNG.EQ.ITRNG.AND.IAROLD.NE.ITROLD) CALL DETECT
C
C  STEP 6-2: UPDATE RING NUMBER MONITOR.
       IAROLD=IARNG
       ITROLD=ITRNG
C
C  **********************************************
C  * STEP 7: CHECK FOR SCAN TERMINATION CONDITIONS *
C  **********************************************
C
C  STEP 7-1: CHECK ALL POSSIBLE TERMINATION CONDITIONS.
C
C  CONDITION = 1: T > 60. SECONDS"
       IF(T.GE.60.) GO TO 40
C
C  CONDITION = 2: NEXT SCAN TIME PARAMETER < 0. "
       ITEMP=KSN-1
       IF(ITEMP.LT.0) GO TO 40
C
C  CONDITION = 3: DETECT A TARGET"
       IF(MTP.EQ.0) RETURN
C
C  STEP 7-2: PERFORM SCAN TERMINATION STEPS —— IF TERMINATION COND
C              ITION OBTAINED.
   40  MSF=0
       KSNCLK=0
       KSN=0
       ISRCHG=0
       ISRCHC=0
       RETURN
       END
C
C
C  **************************************************************
C  * THIS SUBROUTINE DETERMINES WHETHER  THE ANTENNA IS IN THE OB- *
```

```
C  * SCURATION ZONE AND SETS THE SCAN WARNING FLAG APPROPRIATELY.  *
C  ****************************************************************
C
C
      SUBROUTINE SCNWRN
      COMMON /OUTPUT/MSWF,IDUMO(2),DUMO(7),IDUMO1(4)
      COMMON /ATDAT/DUM(8),A,B,DUMA(4)
      DIMENSION ICLEAR(36,72)
      DATA ICLEAR /17*1,13*0,6*1,18*1,12*0,6*1,18*1,12*0,6*1,
     1 18*1,12*0,6*1,19*1,11*0,6*1,19*1,11*0,6*1,19*1,11*0,6*1,
     2 19*1,11*0,6*1,19*1,11*0,6*1,19*1,11*0,6*1,20*1,10*0,6*1,
     3 20*1,10*0,6*1,20*1,10*0,6*1,20*1,10*0,6*1,20*1,10*0,
     4 6*1,20*1,10*0,6*1,19*1,11*0,6*1,18*1,12*0,6*1,17*1,13*0,
     5 6*1,16*1,14*0,6*1,15*1,15*0,6*1,14*1,16*0,6*1,14*1,16*0,
     6 6*1,13*1,17*0,6*1,12*1,18*0,6*1,11*1,19*0,6*1,10*1,20*0,6*1,
     7 9*1,21*0,6*1,9*1,21*0,6*1,8*1,22*0,6*1,4*1,0,3*1,22*0,6*1,
     8 4*1,26*0,6*1,4*1,26*0,6*1,4*1,26*0,6*1,4*1,26*0,6*1,4*1,26*0,6*1,
     9 4*1,26*0,6*1,4*1,26*0,6*1,4*1,26*0,6*1,4*1,26*0,6*1,4*1,26*0,6*1,
     A 4*1,26*0,6*1,4*1,26*0.6*1,4*1,26*0,6*1,4*1,26*0,6*1,4*1,26*0,6*1,
     B 4*1,26*0,6*1,4*1,26*0,6*1,4*1,26*0,6*1,4*1,7*0,2*1,17*0,6*1,
     C 4*1,7*0,2*1,17*0,6*1,4*1,6*0,3*1,17*0,6*1,4*1,5*0,4*1,17*0,6*1,
     D 4*1,5*0,6*1,15*0,6*1,4*1,0,12*1,13*0,6*1,19*1,11*0,6*1,
     E 21*1,9*0,6*1,24*1,6*0,6*1,26*1,4*0,
     F 6*1,27*1,3*0,6*1,28*1,2*0,6*1,29*1,0,6*1,29*1,0,6*1,28*1,
     G 2*0,6*1,27*06,3*0,6*1,26*1,4*0,6*1,25*1,5*0,6*1,23*1,7*0,6*1,
     H 23*1,7*0,6*1,22*1,8*0,6*1,19*1,11*0,6*1,18*1,12*0,6*1/
C
      ALPHA=A*57.3
      BETA=B*57.3
      IF(ABS(BETA).LE.90.) GO TO 1
      BETA=-(180-ABS(B))*(B/ABS(B))
      ALPHA=(180-ABS(A))*(A/ABS(A))
    1 CONTINUE
      IA=INT((ALPHA+180.)/5.+1.)
      IB=INT((90-BETA)/5.+1.)
      MSWF=ICLEAR(IB,IA)
      RETURN
      END
C
C
C  ************************************************************
C  * THIS SUBROUTINE COMPUTES THE RESPONSE TO ALL DISPLAYS AND *
C  * CONTROLS WHEN THE RADAR IS IN ANY OF THE SEARCH MODES.    *
C  ************************************************************
C
C
      SUBROUTINE SEARCH
      COMMON /CNTL/IDUM(3),IASM,ISRCHC,ISRCHG,IAZS,IELS,ISLR,EDRNG,
     2              EDPA,EDRA
      COMMON /OUTPUT/MSWF,MTF,MSF,SRNG,SRDOT,SPANG,SRANG,SPRTE,
     2               SRRTE,SRSS,IDUM2(4)
      COMMON /ICNTL/IOLDPW,IOLDMD,IOLDSM,ISHOLD,KMSCLK,KWMUP,KSNCLK,
     2              KSNMAX,KACCLK,MTP,MZ1,MZ0,MSS,MTKINT,MRNG,MSAM,MPRF,
     3              IDUM1(10)
      COMMON /SYSDAT/TS,DUMS(14)
      COMMON /ATDAT/DUM2(10),PREF,RREF,DUMA(2)
      DIMENSION SLWRTE(2)
      DATA SLWRTE/6.9814E-3,3.4907E-1/
C
C  **********************************
C  * DETERMINE ANTENNA STEERING MODE. *
C  **********************************
      GO TO (10,20,30,40),IASM
C
```

```
C
C  ******************************************************************
C  **************** GPC-ACQ SEARCH AND ACQUISITION MODE. ***************
C  ******************************************************************
C
C
C  ***********************************************************
C  * STEP 1: DETERMINE WHETHER SEQUENCING THRU POINT OR SCAN *
C  ***********************************************************
   10  IF(MSF.EQ.1) GO TO 14
       IF(MZ1.EQ.1.AND.ISRCHG.EQ.1) GO TO 14
C
C  ********************************************
C  * STEP 2: PERFORM GIMBAL POINTING SEQUENCE *
C  ********************************************
C
C  STEP 2-1: UPDATE ROLL/PITCH REFERENCES
       IF(ISHOLD.EQ.1.AND. ISRCHG.EQ.1) GO TO 12
       RREF=EDRA
       PREF=EDPA
   12  ISHOLD=ISRCHG
C
C  STEP 2-2: UPDATE POSITION OF GIMBALS.
       CALL POINT
C
C  STEP 2-3: DETERMINE WHETHER BORESIGHT IN ZONE I AND/OR ZONE O AND
C            TAKE APPROPRIATE ACTION.
       CALL ZONECK
C  IF NOT IN ZONE O, THEN DETECTION IS NOT ALLOWED.
       IF(MZ0.EQ.0) RETURN
C
C  ******************************************************
C  * STEP 3: CHECK FOR TARGET DETECTION —— IF IN ZONE O *
C  ******************************************************
C
       CALL DETECT
       RETURN
C
C  *********************************
C  * STEP 4: PERFORM SCAN SEQUENCE *
C  *********************************
   14  CALL SCAN
       RETURN
C
C
C  ******************************************************************
C  **************** GPC-DES SEARCH AND ACQUISITION MODE ***************
C  ******************************************************************
C
C  ********************************************
C  * STEP1 : PERFORM GIMBAL POINTING SEQUENCE *
C  ********************************************
C
C  STEP 1-1: UPDATE ROLL/PITCH REFERENCE ANGLES.
   20  PREF=EDPA
       RREF=EDRA
C
C  STEP 1-2: UPDATE POSITION OF GIMBALS.
       CALL POINT
C
C  STEP I-3: DETERMINE WHETHER BORESIGHT IN ZONE 1 AND/OR ZONE O AND
C            TAKE APPROPRIATE ACTIN.
       CALL ZONECK
C  IF BORESIGHT NOT IN ZONE O, THEN TARGET DETECTION NOT ALLOWED.
```

```
       IF(MZ0.EQ.0) RETURN
C
C  *********************************************************
C  * STEP 2: CHECK FOR TARGET DETECTION —— IF IN ZONE O.  *
C  *********************************************************
C
       CALL DETECT
       RETURN
C
C
C  ***********************************************************************
C  ****************** AUTO SEARCH AND ACQUISITION MODE *****************
C  ***********************************************************************
C
C
C  ************************************************************
C  * STEP 1: DETERMINE WHETHER SEQUENCING THRU POINT OR SCAN *
C  ************************************************************
   30  IF(ISRCHC.EQ.1) GO TO 32
C
C  *********************************************
C  * STEP 2: PERFORM GIMBAL POINTING SEQUENCE *
C  *********************************************
C
C  STEP 2-1: UPDATE ROLL/PITCH REFERENCE ANGLES.
       PREF=PREF+FLOAT(IELS)*SLWRTE(ISLR+1)*TS
       RREF=RREF+FLOAT(IAZS)*SLWRTE(ISLR+1)*TS
C
C  STEP 2-2: UPDATE POSITION OF GIMBALS.
       CALL POINT
C
C  STEP 2-3: DETERMINE SLEW RATE AND TAKE APPROPRIATE ACTION.
C              IF SLEW RATE IS GREATER THAN 0.4 DEG/SEC, THEN TARGET DET-
       IF(ISLR.GT.0) RETURN
C
C  ******************************************************************
C  * STEP 3: CHECK FOR TARGET DETECTION —— IF SLEW RATE <0.4 DEG *
C  *        PER SECOND.                                            *
C  ******************************************************************
       CALL DETECT
       RETURN
C
C  **********************************
C  * STEP 4: PERFORM SCAN SEQUENCE *
C  **********************************
   32  CALL SCAN
       RETURN
C
C
C  ***********************************************************************
C  **************** MANUAL SEARCH AND ACQUISITION MODE ****************
C  ***********************************************************************
C
C
C  ************************************
C  * STEP 1: UPDATE ANTENNA POSITION *
C  ************************************
C
C  STEP 1-1: UPDATE ROLL/PITCH REFERENCE ANGLES.
   40  PREF=PREF+FLOAT(IELS)*SLWRTE(ISLR+1)*TS
       RREF=RREF+FLOAT(IAZS)*SLWRTE(ISLR+1)*TS
C
C  STEP 1-2: UPDATE POSITION OF GIMBALS.
       CALL POINT
```

```
C                                                                       00006550
C  STEP 1-3: DETERMINE SLEW RATE AND TAKE APPROPRIATE ACTION.           00006560
C            IF SLEW RATE IS GREATER THAN 0.4 DEG/SEC, THEN TARGET DET-00006570
C            ECTION IS NOT ALLOWED.                                     00006580
      IF(ISLR.GT.0) RETURN                                              00006590
C                                                                       00006600
C  ***************************************************************      00006610
C  * STEP 2: CHECK FOR TARGET DETECTION --- IF SLEW RATE <0.4 DEG *     00006620
C  *         PER SECOND.                                          *     00006630
C  ***************************************************************      00006640
      CALL DETECT                                                       00006650
      RETURN                                                            00006660
      END                                                               00006670
C                                                                       00020030
C                                                                       00020040
C  ******************************************************************   00020050
C  * THIS SUBROUTINE GENERATES THE NOISE-FREE ANGLE, RANGE, VELOCITY *  00020060
C  * AND 0N-TARGET DISCRIMINANT COMPONENTS.                          *  00020070
C  ******************************************************************   00020080
C                                                                       00020090
C                                                                       00020100
      SUBROUTINE SIGNAL                                                 00020110
      REAL IRDOT,IRNG                                                   00020115
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)                00020120
      COMMON /OUTPUT/I1DUM(3),SRNG,DUM1(6),IDUM2(4)                     00020130
      COMMON /ICNTL/IDUM5(13),MTKINT,MRNG,MSAM,MPRF,MBKTRK,MBTSUM,      00020140
     2              MBT(8)                                              00020150
      COMMON /TGTDAT/NT,RAU(3,100),RANGE(100),RADVEL(100),RO(3),        00020160
     2               ROU(3),CGRNGE,CGVEL                                00020170
      COMMON /SATDAT/RADAR(3),N20,RT(70,3),SIG(70)                      00020180
      COMMON /RTDAT/IRDOT,IRNG,DUM2(5),MDF(5)                           00020190
      COMMON /SIGDAT/SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE,DF1,DF5,            00020200
     2               DF2,DF4,SIGBAR                                     00020210
      COMMON /XFORMS/TLB(3,3),TLBD(3,3),TLT(3,3),TLTD(3,3)              00020220
      COMPLEX CSUM,CDIFAZ,CDIFEL,CEARLY,CLATE,CDF1,CDF5,CDF2,CDF4,      00020230
     2        DFWTS,PHASE,PHASE1,DOPFIL                                 00020240
      DIMENSION CTP(10,2),DFWTS(5,100),ALAM(5),ALAMD(3),NFREQ(2)        00020250
      DATA CTP/9*.03318,9.799E-4,4*.03318,1.9599E-3,9.8E-4,4.9E-4,      00020270
     2         2*2.45E-4,1.225E-4/                                      00020280
      DATA NFREQ/1,5/,ALAM/177.3733,176.0447,178.7149,176.7089,         00020290
     2     178.0393/,ALAMD/1.272461E-2,2.969089E-2,3.309023E-1/         00020300
      REAL LATE                                                         00020310
C                                                                       00020320
C  *****************************************************************    00020330
C  *  STEP 1: PRELIMINARY COMPUTATIONS AND PARAMETER INITIALIZATION *   00020340
C  *****************************************************************    00020350
C                                                                       00020360
C  STEP 1-1: INITIALIZE DISCRIMINANT COMPONENTS (NOTE: THESE ARE THE    00020370
C            COMPONENT SIGNALS AFTER SQUARE-LAW DETECTION).             00020380
      SPAZ=0.0                                                          00020390
      SMAZ=0.0                                                          00020400
      SPEL=0.0                                                          00020410
      SMEL=0.0                                                          00020420
      EARLY=0.0                                                         00020430
      LATE=0.0                                                          00020440
      DF1=0.0                                                           00020450
      DF5=0.0                                                           00020460
      DF2=0.0                                                           00020470
      DF4=0.0                                                           00020480
      SIGBAR=0.0                                                        00020490
C                                                                       00020500
      NFMAX=NFREQ(IMODE)                                                00020510
      DO 55 I=1,NFMAX                                                   00020520
C                                                                       00020530
```

```
C  STEP 1-2: INITIALIZE COMPLEX DISCRIMINANT COMPONENTS BEFORE EACH     00020540
C            XMIT FREQUENCY (NOTE: THESE ARE THE COMPONENT SIGNALS      00020550
C            BEFORE SQUARE-LAW DETECTION).                              00020560
      CSUM=(0.,0.)                                                      00020570
      CDIFAZ=(0.,0.)                                                    00020580
      CDIFEL=(0.,0.)                                                    00020590
      CEARLY=(0.,0.)                                                    00020600
      CLATE=(0.,0.)                                                     00020610
      CDF1=(0.,0.)                                                      00020620
      CDF5=(0.,0.)                                                      00020630
      CDF2=(0.,0.)                                                      00020640
      CDF4=(0.,0.)                                                      00020650
      DO 45 K=1,NT                                                      00020660
C                                                                       00020670
      IF(I.GT.1) GO TO 35                                               00020680
C                                                                       00020690
C  ***********************************************************          00020700
C  * STEP 2: COMPUTE SUM CHANNEL MULTIPLICATION FACTOR FOR KTH *          00020710
C  *         SCATTERER.                                        *          00020720
C  ***********************************************************          00020730
C                                                                       00020740
C  STEP 2-1: COMPUTE SUM PATTERN ANGLE.                                 00020750
      PSI=ACOS(ABS(RAU(3,K)))                                           00020760
C                                                                       00020770
C  STEP 2-2: COMPUTE ANTENNA SUM PATTERN MULTIPLICATION FACTOR.         00020780
      X=SPAT(PSI)                                                       00020790
C                                                                       00020800
C  STEP 2-3: COMPUTE SUM CHANNEL MULTIPLICATION FACTOR.                 00020810
      XX=SIG(K)*X                                                       00020820
C  NOTE: IF IN ACTIVE MODE SET XX=1.0.                                  00020830
      IF(IMODE.EQ.1) XX=1.0                                             00020840
      S=XX*X                                                            00020850
C                                                                       00020860
C  STEP 2-4: CHECK ANTENNA STEERING MODE (IF IN GPC-DES OR MANUAL       00020870
C                --- SKIP STEP 4).                                      00020880
      IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 20                               00020890
C                                                                       00020900
C  **************************************************************       00020910
C  * STEP 3: COMPUTE AZ AND EL DIFFERENCE CHANNEL MULTIPLICATION *       00020920
C  *         FACTORS FOR KTH SCATTERER.                           *       00020930
C  **************************************************************       00020940
C                                                                       00020950
C  STEP 3-1: COMPUTE AZ AND EL DIFFERENCE PATTERN ANGLES.               00020960
      DELAZ=-ASIN(RAU(2,K))                                             00020970
      DELEL=ASIN(RAU(1,K))                                              00020980
C                                                                       00020990
C  STEP 3-2: COMPUTE AZ AND EL DIFFERENCE PATTERN MULTIPLICATION        00021000
C            FACTORS.                                                   00021010
      Y=DPAT(DELAZ)                                                     00021020
      Z=DPAT(DELEL)                                                     00021030
C                                                                       00021040
C  STEP 3-3: COMPUTE AZ AND EL DIFFERENCE CHANNEL MULTIPLICATION        00021050
C             FACTORS (INCLUDE RCS AND SUM PATTERN WEIGHTINGS).         00021060
      DAZ=XX*Y                                                          00021070
      DEL=XX*Z                                                          00021080
C                                                                       00021090
C  ********************************************************             00021100
C  * STEP 4: COMPUTE RANGE GATE WEIGHTING FOR KTH SCATTERER *             00021110
C  ********************************************************             00021120
C  DEFINITION:  CTP=4./(C*PULSEWIDTH) WHERE C IS SPEED OF LIGHT.        00021130
C                                                                       00021140
C  STEP 4-1: COMPUTE RANGE GATE LOCATION WRT RANGE GATE CENTER.         00021150
CCCCCCCCCCCCCCCCCC MOD MAR 24 1983 CCCCCCCCCCCCCCCCCC
   20  CONTINUE
```

```
      SRNGX=10.*AINT(0.03125*IRNG)
      DELX=CTP(MRNG,IMODE)*(RANGE(K)-SRNGX)
C
C  STEP 4-2: COMPUTE EARLY AND LATE RANGE GATE WEIGHTINGS FOR
C            KTH SCATTERER.
      II=INT((DELX+7.)/2.)
      IF(II.LE.1) II=1
      IF(II.GE.5) II=5
      GO TO (21,22,23,24,21),II
   21 RGE=0.0
      RGL=0.0
      GO TO 25
   22 RGE=3.+DELX
      RGL=0.0
      GO TO 25
   23 RGE=1.-DELX
      RGL=1.+DELX
      GO TO 25
   24 RGE=0.0
      RGL=3.-DELX
C
C  STEP 4-3: COMPUTE RANGE GATE WEIGHT FOR NON-RANGE DISCRIMINANT
C            COMPONENTS.
   25 RGWGT=0.5*(RGL+RGE)
C
C  STEP 4-4: APPLY RANGE GATE WEIGHTING TO SUM AND DIFFERENCE
C            CHANNEL MULTIPLICATION FACTORS.
      RGE=S*RGE
      RGL=S*RGL
      S=S*RGWGT
      DAZ=DAZ*RGWGT
      DEL=DEL*RGWGT
C
C  ****************************************************************
C  * STEP 5: COMPUTE DOPPLER FILTER PHASE SHIFT AND WEIGHTING FOR KTH *
C  *         SCATTERER. NOTE: THIS CALCULATION IS INDEPENDENT OF XMIT *
C  *         FREQUENCY AND ASSUMES NO ACCELERATION OVER DATA CYCLE.   *
C  ****************************************************************
C
C  DEFINITION:  ALAMD(MPRF)=2.*PI/(PRF*LAMBDA)
C  DEFINITION:  THE CONSTANT 0.196348=PI/16.
C
C  STEP 5-2: COMPUTE DOPPLER FREQUENCY CORRESPONDING TO RADIAL VELOCITY
C            OF KTH SCATTERER.
      FDT=-2.*ALAMD(MPRF)*RADVEL(K)
C
C  STEP 5-3: COMPUTE DOPPLER FILTER WEIGHTING FOR EACH OF FIVE DOPPLER
C            TRACKING FILTERS.
      DO 30 J=1,5
      ARG=0.196348*MDF(J)-FDT
   30 DFWTS(J,K)=DOPFIL(ARG)
C
C  ****************************************************************
C  * STEP 6: COMPUTE PHASE FACTOR ASSOCIATED WITH KTH SCATTERER RANGE *
C  *         (NOTE: PHASE IS REFERENCD TO PHASE ASSOCIATED WITH RANGE *
C  *         OF TARGET C.G.)                                          *
C  ****************************************************************
C
C  DEFINITION: RANGE(K) IS RANGE OF KTH SCATTERER TO ANTENNA PHASE CENTR
C  DEFINITION: ALAM=4.*PI/LAMBDA WHERE LAMBDA IS XMIT FREQUENCY.
C
C  STEP 6-1: COMPUTE PHASE REFERENCED TO TARGET C.G.
   35 DELPSI=ALAM(I)*(RANGE(K)-CGRNGE)
C
```

```
C  STEP 6-2: COMPUTE PHASE FACTOR, I.E. EXP(J*DELPHI).
      PHASE=CEXP(CMPLX(0.,DELPSI))
      PHASE1=PHASE
C
C  STEP 6-3: COMBINE RANGE PHASE FACTOR AND DOPPLER FILTER =3
C            WEIGHT AND PHASE FACTOR.
      PHASE=PHASE*DFWTS(3,K)
C
C  ****************************************************************
C  * STEP 7: ADD (VECTORIALLY) KTH SCATTERER CONTRIBUTION TO EACH *
C  *         DISCRIMINANT'S COMPONENT SIGNALS.                    *
C  ****************************************************************
C
C  STEP 7-1: ADD KTH SCATTERER CONTRIBUTION TO SUM CHANNEL SIGNAL.
      CSUM=CSUM+S*PHASE
C
C  STEP 7-2: CHECK ANTENNA STEERING MODE ---- SKIP STEP 8-3 IF IN
C            GPC-DES OR MANUAL MODE.
      IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 40
C
C  STEP 7-3: ADD KTH SCATTERER CONTRIBUTION TO AZ AND EL DIFFERENCE
C            CHANNELS SIGNALS.
      CDIFAZ=CDIFAZ+DAZ*PHASE
      CDIFEL=CDIFEL+DEL*PHASE
C
C  STEP 7-4: ADD KTH SCATTERER CONTRIBUTION TO RANGE DISCRIMINANT
C            COMPONENT SIGNALS.
   40 CEARLY=CEARLY+RGE*PHASE
      CLATE=CLATE+RGL*PHASE
C
C  STEP 7-5: ADD KTH SCATTERER CONTRIBUTION TO VELOCITY DISCRIMINANT
C            COMPONENT SIGNALS.
      PHASE1=PHASE1*S
      CDF2=CDF2+PHASE1*DFWTS(2,K)
      CDF4=CDF4+PHASE1*DFWTS(4,K)
C
C  STEP 7-6: ADD KTH SCATTERER CONTRIBUTION TO ON-TARGET DISCRIMINANT
C            COMPONENT SIGNALS.
      CDF1=CDF1+PHASE1*DFWTS(1,K)
      CDF5=CDF5+PHASE1*DFWTS(5,K)
   45 CONTINUE
C
C  *****************************************************************
C  * STEP 8: FORM NOISE-FREE ANGLE, RANGE, VELOCITY, AND ON-TARGET *
C  *         DISCRIMINANT COMPONENTS AT ITH FREQUENCY AND SQUARE   *
C  *         LAW DETECT THESE COMPONENTS.                          *
C  *****************************************************************
C
C  STEP 8-1: CHECK ANTENNA STEERING MODE ----  SKIP STEPS 9-2 AND 9-3
C            IF IN GPC-DES OR MANUAL.
      IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 50
C
C  STEP 8-2: COMPUTE AZ DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
      SPAZ=SPAZ+CABS(CSUM+CDIFAZ)**2
      SMAZ=SMAZ+CABS(CSUM-CDIFAZ)**2
C
C  STEP 8-3: COMPUTE EL DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
      SPEL=SPEL+CABS(CSUM+CDIFEL)**2
      SMEL=SMEL+CABS(CSUM-CDIFEL)**2
C
C  STEP 8-4: COMPUTE RANGE DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT
   50 EARLY=EARLY+CABS(CEARLY)**2
      LATE=LATE+CABS(CLATE)**2
C
```

```
C  STEP 8-5: COMPUTE VELOCITY DISCRIMINANT COMPONENTS AND SQUARE-LAW
C           DETECT.
      DF2=DF2+CABS(CDF2)**2
      DF4=DF4+CABS(CDF4)**2
C
C  STEP 8-6: COMPUTE ON-TARGET DISCRIMINANT COMPONENTS AND SQUARE-LAW
C           DETECT.
      DF1=DF1+CABS(CDF1)**2
      DF5=DF5+CABS(CDF5)**2
C
C  ***************************************************************
C  * STEP 9: COMPUTE EFFECTIVE CROSS-SECTION AVERAGED OVER PROPER *
C  *         NUMBER OF TRANSMIT FREQUENCIES.                      *
C  ***************************************************************
      SIGBAR=SIGBAR+CABS(CSUM)**2
   55 CONTINUE
      SIGBAR=SIGBAR/FLOAT(NFREQ(IMODE))
C
C   NOTE: DEBUGGING PRINT STATEMENTS
C     WRITE(6,900) (I,SIG(I), I=1,NT)
  900 FORMAT(' I,SIG =',I8,F14.4)
C     WRITE(6,902) NT,S,DAZ,DEL,RGE,RGL,RGWGT,MDF(3)
C     WRITE(6,901) DFWTS(1,K),DFWTS(2,K),DFWTS(3,1),DFWTS(4,1),
C    2 DFWTS(5,1)
  902 FORMAT(' NT,S,DAZ,DEL,RGE,RGL,RGWGT,F3 =',I5,6F10.2,I5)
  901 FORMAT(' DF WTS =',10F12.4)
      RETURN
      END
C
C
C  ****************************************************
C  * THIS SUBROUTINE CONTAINS SINGLE-HIT DETECTION MODEL *
C  ****************************************************
C
C
      SUBROUTINE SINGLE
      DIMENSION P(41)
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUM(5),DUMC(3)
      COMMON /OUTPUT/MSWF,MTF,MSF,DUM(7),IDUM1(4)
      COMMON /ICNTL/IDUM2(8),KACCLK,MTP,IDUM3(5),MSAM,IDUM4(11)
      COMMON /TGTDAT/NT,DUM1(500),RO(3),ROU(3),CGRNGE,CGVEL
      COMMON /DETDAT/SIGMA,CGANG
      DATA NSRCH/105/
      DATA P/6*0.0,.001,.003,2*.004,.008,.012,.015,.043,.053,.076,.107,
     2 .147,.193,.244,.312,.363,.444,.514,.590,.644,.706,.765,.815,.861,
     3 .882,.918,.937,.955,.966,.976,.980,.989,.991,.997,.996/
C
C  ***************************************************
C  * STEP 1: COMPUTE NOMINAL SNR AT VIDEO FILTER OUTPUT *
C  ***************************************************
C
C  STEP 1-1: SET SAMPLE RATE TO OBTAIN CORRECT NOISE BW IN SNRV COMP.
      MSAM=1
      IF (IMODE.EQ.1) MSAM=2
C
C  STEP 1-2: COMPUTE NOMINAL SNRV.
      SNR=SNRV(SIGMA,CGRNGE)
C
C  ***************************************************
C  * STEP 2: IF NOT SCANNING ADD BEAMSHAPE LOSS TO SNRV *
C  ***************************************************
C
C  STEP 2-1: CHECK SCAN FLAG.
      IF(MSF.EQ.1) GO TO 1
```

```
C
C  STEP 2-2: COMPUTE BEAMSHAPE LOSS ——— BASED UPON C.G. POSITION
C            OFF BORESIGHT.
      BETA2=SPAT(CGANG)**2
C
C  STEP 2-3: ADD BEAMSHAPE LOSS TO NOMINALV, I.E. COMPUTE ACTUAL SNR
C            SNRV.
      SNR=SNR*BETA2
C
C  *************************************************************
C  * STEP 3: DETERMINE PROBABILITY OF DETECTION, PD, BASED UPON SNR *
C  *************************************************************
C
C  STEP 3-1: DETERMINE INDEX TO ACCESS APPROPRIATE PD VERSUS SNR
C            CURVE.
    1 IF(IMODE.EQ.2) GO TO 5
      NCRV=1
      GO TO 15
    5 IF(IASM.LT.3) GO TO 10
      NCRV=3
      GO TO 15
   10 NCRV=5
C
C  ADJUST INDEX FOR SCANNING.
   15 NCRV=NCRV+MSF
C
C
C  STEP 3-2: CONVERT SNRV TO DB.
      IF(SNR.LT.1.E-08) GO TO 20
      SNR=10.*ALOG10(SNR)
      GO TO 25
   20 SNR=-100.
C
C  STEP 3-3: SNR OUTSIDE (-30 DB, 0 DB) INTERVAL" ——— IF SO, SET
C            OUTCOME APPROPRIATELY AND SKIP REMAINING STEPS.
C
C  IF SNR < -25 DB THEN SET PD=0.0 (DECLARE A MISS).
   25 IF(SNR.LT.-25.) GO TO 30
C
C  IF SNR > -5 DB THEN SET PD=1.0 (DECLARE A HIT).
      IF(SNR.GT.-5.0) GO TO 35
C
C  STEP 3-4: COMPUTE INDEX FOR LOOKUP TABLE AND FACTORS FOR LINEAR
C            INTERPOLATION.
      SCALE=(SNR+25.)*2.+1.000001
      ISNR=INT(SCALE)
      REMAIN=SCALE-FLOAT(ISNR)
C
C  STEP 3-5: DETERMINE PD USING TABLE AND LINEAR (IN DB) INTERPOLATION.
      PROB=P(ISNR)+REMAIN*(P(ISNR+1)-P(ISNR))
C
C  ***********************************************
C  * STEP 4: DETERMINE OUTCOME OF DETECTION ATTEMPT *
C  ***********************************************
C
      X=RNDU(NSRCH)
      IF(X.LE.PROB) GO TO 35
C
C  ************************************************************
C  * STEP 5: SET CONTROLS BASED UPON OUTCOME OF DETECTION ATTEMPT *
C  ************************************************************
C
C  STEP 5-1: IF NO DETECTION ——— SET TARGET PRESENT FLAG LOW.
   30 MTP=0
```

```
      RETURN                                                            00008440
C                                                                       00008450
C  STEP 5-2: IF DETECTION SUCCESSFUL --- SET TARGET PRESENT FLAG        00008460
C            HIGH AND INITIALIZE ACQUISITION CLOCK.                     00008470
   35 MTP=1                                                             00008480
      KACCLK=0                                                          00008490
      RETURN                                                            00008500
      END                                                               00008510
C                                                                       00010640
C                                                                       00010650
C  *******************************************************************  00010660
C  * THIS FUNCTION COMPUTES THE @NOMINAL@ SNR AT THE VIDEO OUTPUT *     00010670
C  * --- IT ASSUMES NO BEAMSHAPE OR SCAN LOSS.                     *    00010680
C  *******************************************************************  00010690
C                                                                       00010700
C                                                                       00010710
      FUNCTION SNRV(SIGMA,RANGE)                                        00010720
      COMMON /CNTL/IPWR,IMODE,ITXP,IDUMC(6),DUMC(3)                     00010730
      COMMON /ICNTL/IDUM(12),MSS,MTKINT,MRNG,MSAM,MPRF,IDUM2(10)        00010740
      COMMON /SYSDAT/DUM(12),TGTSIG,GPS,GAS                             00010750
      DIMENSION PT(4),BN(2)                                             00010760
CCCCCCCCCCCCCCCCC MOD MAR 24 1983 CCCCCCCCCCCCCCCCCCC
      DATA PT/46.3,34.9,23.,6.2/, BN/69.9,57.9/                            00010770
C                                                                       00010780
C  **********************************************                       00010790
C  * DETERMINE WHETHER ACTIVE OR PASSIVE MODE *                         00010800
C  **********************************************                       00010810
      IF(IMODE.EQ.1) GO TO 10                                           00010820
C                                                                       00010830
C  ****************************************                             00010840
C  * PASSIVE MODE VIDEO SNR CALCULATION *                               00010850
C  ****************************************                             00010860
       IF((SRNG.LT.640.).OR.(ISTS7.EQ.1))ITXP=4
      SNRV=GPS+PT(ITXP)+10.*ALOG10(SIGMA)-BN(MSAM)-40.*ALOG10(RANGE)    00010870
      SNRV=10.**(0.1*SNRV)                                              00010880
      RETURN                                                            00010890
C                                                                       00010900
C  ***************************************                              00010910
C  * ACTIVE MODE VIDEO SNR CALCULATION *                                00010920
C  ***************************************                              00010930
   10 SNRV=GAS-20.*ALOG10(RANGE)                                        00010940
      SNRV=10.**(0.1*SNRV)                                              00010950
      RETURN                                                            00010960
      END                                                               00010970
C MODIFIED FOR LENS
C                                                                       00032640
C                                                                       00032650
C  ************************************************************         00032660
C  * THIS SUBROUTINE MODELS THE SPAS SPACECRAFT SCATTERING *            00032670
C  * PROPERTIES.                                            *           00032680
C  ************************************************************         00032690
C                                                                       00032700
C  SES SPAS MODEL AS OF JULY 7,1981.
C
C                                                                       00032710
      SUBROUTINE SPAS                                                   00032720
      COMMON /SATDAT/RADAR(3),KTAR,R(70,3),SIG(70),ROLD,ICLOSE,ICLOLD
    1 ,JHOT(60)
      DIMENSION SIGMA(61),TARG(61,3),PHIMIN(61,3),PHIMAX(61,3)
      DIMENSION OFFSET(61)          ,PHI(61,3)
      DIMENSION VECT(3),COSPHI(61,3)
      DIMENSION ALPH(24,3),V(24,3),NORMAL(24),DIM(24,3),WRAN(24,3)
      DIMENSION WSCALE(24,3),DPHI(24),PHIOLD(24),VOLD(24,3),KSEED(24,3)
      DIMENSION TTRAN(3)
```

```
C
C
C   ************************************************************
C   * DATA DEFINITION:  INCLUDES SCATTERER LOCATION IN TARGET FRAME, *
C   *                   MAXIMUM SCATTERER RCS VALUE, ANGULAR EXTENT  *
C   *                   OF NONZERO RCS, AND OTHER MISCELLANEOUS DATA *
C   *                   REQUIRED BY THE ROUTINE.                     *
C   ************************************************************
C
C
C   SEED FOR RANDOM NUMBER GENERATOR
      DATA KSEED/45,678,908,607,5678,897,345,7777,67,4,
    1 560,809,444,888,999,555,222,70,80,8000,
    2 5,15,25,35,45,55,65,75,85,95,
    3 7,17,27,37,47,57,67,77,87,97,
    4 9876,984,6666,2398,76,412,7589,409,899,561,
    5 205,3895,9457,9643,937,656,453,980,567,2154,
    6 801,88,99,31,85,106,4,9,3,987,
    7 888,999/
C
C   DATA DESCRIBING DIMENSIONS OF WIDE-ANGLE SCATTERERS
C   DEFINITION: DIM=2*D/LAMBDA (UNITLESS)
C   DEFINITION: WSCALE=SQRT(D**2/(12*NF)) (UNITS=FEET, NF== OF FREQ)
      DATA DIM /72*64.8/
      DATA WSCALE /72*0.2965/
C
C   FOR EACH DIFFUSE SCATTERER, SPECIFY NORMAL COMPONENT
      DATA NORMAL /10*1,2*2,12*3/
C
C   SQUARE ROOT OF RCS VALUES ( FEET).
      DATA SIGMA/24*.05,3*2.6,2*61.,1200.,1.25,0.17,25.7,110.,90.,
    2 100.,850.,1200.,1117.,0.4,80.,100.,900.,85.,750.,850.,920.,
    3 730.,6*0.03,1250.,1130.,1400.,900.,1000.,1150.,32.39/
C
C   COORDINATES OF SCATTERERS IN SPAS FRAME (FEET)
      DATA TARG /4*.12,6*-.7,8*-.35,.37,4*-.35,.37,3*.24,2*.37,
    2   .66,3*-.35,3*.12,3*-.3,5*-.35,4*.37,6*.24,6*.7,0.0,
    3 1.75,-1.05,-1.75,.35,1.75,1.05,.35,-.35,-1.05,-1.75,2.15,
    4 -2.15,1.75,1.05,.35,-.35,-1.05,-1.75,.35,1.05,.35,-.35,
    5 -1.75,.35,-.83,-1.05,-1.27,1.05,-.35,.35,3*-1.05,1.9,-1.05,
    6 -1.8,2.0,-2.0,.0,1.75,1.05,.35,-.35,-1.75,2*1.05,2*-.35,
    7 2*-.83,2*-1.05,2*-1.27,1.75,1.05,.35,-.35,-1.05,-1.75,0.0,
    8 12*.0,7*.48,5*-.48,3*.15,3*.0,3*-.8,3*.0,3*.67,-.86,
    9 4*-.48,.425,-.425,.425,-.425,-.02,.3,-.02,.3,-.02,.3,
    A 6*0.0,2.38/
C
C   MINIMUM SUBTENDED ANGLE
      DATA PHIMIN /4*.0,6*90.,14*0.,16*0.,4*88.5,4*88.0,6*0.0,
    2 6*177.9,0.,
    3 11*.0,90.,12*.0,50.,35.,30.,.0,45.0,3*.0,10.0,4*.0,177.4,
    4 89.7,.0,4*88.5,4*88.0,12*.0,48.,
    5 19*0.,5*90.,3*85.9,3*88.5,156.,90.,87.7,3*88.5,2*87.4,.0,
    6 90.,4*178.5,0.,178.,0.,178.,90.,0.,90.,0.,90.,0.,6*88.5,
    7  48.0/
C
C   MAXIMUM SUBTENDED ANGLE
      DATA PHIMAX /4*90.,20*180.,5*90.,2.1,3*180.,3*2.1,4*180.,
    2 4*91.5,4*92.,6*90.,6*180.,48.,
    3 10*180.,90.,13*180.,4*150.,155.,135.,2*180.,145.,3*180.,
    4 2.6;180.,90.3,180.,4*91.5,4*92.,6*180.,6*180.,138.,
    5 12*180.,7*90.,5*180.,3*94.1,3*91.5,180.,156.,92.3,3*91.5,2*92.6,
    6 125.,5*180.,2.,180.,2.,2*180.,90.,180.,90.,180.,90.,6*91.5,138./
C
C   RADII OF THE SCATTERERS (FEET)
```

```
         DATA OFFSET /24*.0,3*.1,2*.29,.0,2*.35,.315,5*.0,.24,.35,8*0.,
       2 6*.1,6*.0,0.0/
C
C  MISCELLANEOUS DATA.
         DATA NTAR/61/,KWIDE/24/,PI/3.141592653/
         DATA TTRAN/3*0.0/,INIT1/1/
C
C  *************************************************************
C  * STEP 0: TRANSLATE POINT TARGETS BY TARGET FRAME OFFSET (TTRAN) *
C  *************************************************************
         IF(INIT1.NE.1) GO TO 2
C
C  RANDOMIZE DIFFUSE SCATTERER RCS VALUES.
C
          ISEED=100
          DO 107 I=1,1000
107       X=RNDU(ISEED)
          DO 108 I=1,KWIDE
          X=RNDU(ISEED)
C        CHANCE MADE 9-11-81
108       SIGMA(I)=SIGMA(I)+(X*0.005)-0.0025
C
C  CONVERT TARGET DATA APPROPRIATELY.
C
          FTM=0.3048
          DO 101 I=1,NTAR
101       SIGMA(I)=SQRT(SIGMA(I))/FTM
          DO 102 J=1,NTAR
          DO 102 I=1,3
102       TARG(J,I)=TARG(J,I)/FTM
          DO 103 J=1,NTAR
          DO 103 I=1,3
          PHIMIN(J,I)=COS(PHIMIN(J,I)*PI/180.)
103       PHIMAX(J,I)=COS(PHIMAX(J,I)*PI/180.)
          DO 105 I=1;NTAR
105       OFFSET(I)=OFFSET(I)/FTM
C
         DO 1 K=1,NTAR
         DO 1 I=1,3
       1 TARG(K,I)=TARG(K,I)+TTRAN(I)
         INIT1=0
C
C  ***********************************************************
C  * STEP 1: DETERMINE WHICH SCATTERER ARE ILLUMINATED AND HAVE A *
C  *          NONZERO RCS IN THE DIRECTION OF THE RADAR.         *
C  ***********************************************************
C
C  STEP 1-1: PERFORM REQUIRED INITIALIZATIONS.
    2  CONTINUE
       NWIDE=0
       KTAR=0
C-
C  STEP 1-2: COMPUTE UNIT VECTOR IN DIRECTION OF RADAR FOR
C             ITH SCATTERING CENTER.
       DO 15 I=1,NTAR
       DO 5 J=1,3
       VECT(J)=RADAR(J)-TARG(I,J)
    5  CONTINUE
       VNORM=SQRT(VECT(1)**2+VECT(2)**2+VECT(3)**2)
       DO 10 J=1,3
      IF(ABS(VECT(J)).GT.ABS(VNORM))WRITE(6,*)'VECT GREATER THAN VNORM'
       COSPHI(I,J)=VECT(J)/VNORM
C
C  STEP 1-3: DETERMINE WHETHER ITH SCATTERER HAS A NONZERO RCS IN THE
```

```
C              DIRECTION OF THE RADAR.
         IF(COSPHI(I,J).LT.PHIMAX(I,J).OR.COSPHI(I,J).GT.PHIMIN(I,J))
     2 GO TO 15
   10  CONTINUE
C
C  STEP 1-4: IF ITH SCATTERER RCS IS NONZERO THEN ADD TO VECTOR OF
C            ILLUMINATED SCATTERERS.
       KTAR=KTAR+1
       JHOT(KTAR)=I
       SIG(KTAR)=SIGMA(I)
       IF(I.LE.KWIDE) NWIDE=NWIDE+1
   15  CONTINUE
C
C  *******************************************************************
C  * STEP 2: COMPUTE LOCATION OF SPECULAR POINTS THAT ARE ILLUMINATED *
C  *******************************************************************
       DO 20 K=1,KTAR
       I=JHOT(K)
       DO 20 J=1,3
       R(K,J)=TARG(I,J)+OFFSET(I)*COSPHI(I,J)
   20  CONTINUE
C
C  ****************************************************************
C  * STEP 3: COMPUTE SQUARE ROOT OF RCS FOR ALL ILLUMINATED WIDE *
C  *         ANGLE SCATTERERS (REPRESENTING DIFFUSE SCATTERING    *
C  *         AREAS).                                              *
C  ****************************************************************
       DO 22 K=1,NWIDE
       I=JHOT(K)
       IQ=NORMAL(I)
   22  SIG(K)=SQRT(ABS(COSPHI(I,IQ       )))*SIGMA(I)
C
C  ****************************************
C  * STEP 4: CHECK FOR SHORT RANGE CONDITION *
C  ****************************************
C
C  STEP 4-1: DETERMINE RANGE TO RADAR IN TARGET FRAME.
   24  RANGE=SQRT(RADAR(1)**2+RADAR(2)**2+RADAR(3)**2)
C
C  STEP 4-2: SET HYSTERESIS LOOP MONITORING VARIABLE.
       IF((ROLD.LT..01.OR.RANGE-ROLD.LE.0.).AND.RANGE.LE.270.) ICLOSE=1
       IF(RANGE-ROLD.GT.0..AND.RANGE.GT.50.) ICLOSE=0
C
C  STEP 4-3: CHECK MONITORING VARIABLE TO DETERMINE IF SHORT RANGE
C            CONDITION EXISTS.
       IF(ICLOSE.EQ.0.OR.NWIDE.EQ.0) GO TO 55
C
C  ********************************************************
C  * STEP 5: PROCEDURE FOR UPDATING OF DIFFUSE SCATTERING    *
C  *         CENTER LOCATION -- SHORT RANGE CONDITION ONLY. *
C  ********************************************************
C
C  STEP 5-1: IF FIRST TIME THRU -- PERFORM INITIALIZATION OF
C            DIFFERENCE EQUATIONS FOR ALL DIFFUSE SCATTERERS.
       IF(ICLOLD.EQ.1) GO TO 35
       DO 30 I=1,KWIDE
      IQ=NORMAL(I)
       PHIOLD(I)=ACOS(COSPHI(I,IQ       ))
       DO 25 J=1,3
       IF(J.EQ.IQ       ) GO TO 25
       V(I,J)=WSCALE(I,J)*(RNDU(KSEED(I,J))-.5)
       VOLD(I,J)=V(I,J)
       R(I,J)=R(I,J)+V(I,J)
   25  CONTINUE
   30  CONTINUE
       GO TO 55
C
C  STEP 5-2: UPDATE ANGULAR INCREMENT FOR EACH DIFFUSE SCATTERER
C          ---- CHANGE IN ANGLE FROM SAMPLE-TO-SAMPLE.
   35  DO 40 I=1,KWIDE
      IQ=NORMAL(I)
       PHI(I,IQ       )=ACOS(COSPHI(I,IQ       ))
       DPHI(I)=(PHI(I,IQ       )-PHIOLD(I))
       PHIOLD(I)=PHI(I,IQ       )
   40  CONTINUE
C
C
C  STEP 5-3: UPDATE SCATTERER LOCATION FOR ALL ILLUMINATED DIFFUSE
C          SCATTERER ---- UPDATE DIFFERENCE EQUATIONS.
       DO 50 K=1,NWIDE
       I=JHOT(K)
       DO 45 J=1,3
      IQ=NORMAL(I)
       IF(J.EQ.IQ       ) GO TO 45
       ALPH(I,J)=EXP(-DIM(I,J)*ABS(DPHI(I)*COSPHI(I,IQ       )))
       WRAN(I,J)=SQRT(1.-ALPH(I,J)**2)*WSCALE(I,J)*(RNDU(KSEED(I,J))-.5)
       V(I,J)=ALPH(I,J)*VOLD(I,J)+WRAN(I,J)
       VOLD(I,J)=V(I,J)
       R(K,J)=R(K,J)+V(I,J)
   45  CONTINUE
   50  CONTINUE
  55   CONTINUE
C
C  *************************************************************
C  * STEP 6: UPDATE PARAMETERS USED TO MONITOR TARGET POSITION *
C  *          ON SHORT RANGE HYSTERESIS CURVE.                 *
C  *************************************************************
       ROLD=RANGE
       ICLOLD=ICLOSE
C
C
C      WRITE(6,908) KTAR,NWIDE,ICLOSE,ROLD
  908  FORMAT(/' TT,WT,IC,R =',3I8,F12.4)
C  *******************************************************************
C  * NOTE: THE FOLLOWING STATEMENTS ARE PRINT STATEMENTS USED IN THE *
C  *       DEBUGGING PROCESS.                                        *
C  *******************************************************************
C
C
C  NOTE: DEBUGGING PRINT STATEMENTS.
C  PRINT LOCATION OF RADAR IN TARGET FRAME.
C      WRITE(6,900) RADAR
C
C  PRINT TABULAR LISTING OF ALL DATA ASSOCIATED WITH SPAS SCATTERERS.
C      WRITE(6,901)(I,SIGMA(I),TARG(I,1),TARG(I,2),TARG(I,3),OFFSET(I)
C    8 ,PHIMIN(I,1),
C    1 PHIMAX(I,1),PHIMIN(I,2),PHIMAX(I,2),PHIMIN(I,3),PHIMAX(I,3),
C    2 I=1,NTAR)
C
C  PRINT TOTAL = OF SCATTERERS AND = OF DIFFUSE SCATTERERS.
C      WRITE(6,902) KTAR,NWIDE
C
C  PRINT INFORMATION ASSOCIATED WITH ILLUMINATED SCATTERERS.
C      WRITE(6,903)
C      WRITE(6,904) (I,JHOT(I),SIG(I),(R(I,J),J=1,3),
C    1 I=1,KTAR)
C
C  PRINT DATA ASSOCIATED WITH DIFFUSE SCATTERER DIFFERENCE EQUATION.
```

```
C       WRITE(6,905)I,PHIOLD(I),
C     1 (V(I,L),L=1,3),(R(I,L),L=1,3)
C      IQ=NORMAL(I)
C       WRITE(6,906) I,PHI(I,IQ        ),PHIOLD(I),DPHI(I)
C       WRITE(6,907)K,I,(VOLD(I,J),J=1,3),(ALPH(I,J),J=1,3),
C     1 (WRAN(I,J),J=1,3),(V(I,J),J=1,3),(R(I,J),J=1,3)
C
C  ALL PRINT FORMAT STATEMENTS.
  900   FORMAT(' IN FEET, RADAR = (',F8.1,',',F8.1,',',F8.1,')')
  901   FORMAT(I12,F10.2,3F8.3,F12.3,4X,2F8.2,4X,2F8.2,4X,2F8.2)
  902   FORMAT(' TOTAL = OF TARGETS = ',I3,'      OF THESE, = MARKOV = ',
     1  I2)
  903   FORMAT(//,9X,'I',3X,'JHOT(I)',7X,'RCS',5X,'PHI-X',5X,'PHI-Y',
     1  5X,'PHI-Z',/)
  904   FORMAT(2I10,4F10.3)
  905   FORMAT(I3,F15.3,2(5X,3F10.3))
  906   FORMAT(' I,PHI,PHIOLD,DPHI',/,I3,3F10.3)
  907   FORMAT(2I3,5(2X,3F7.3))
        RETURN
        END
C
C
C  ***********************************************************
C  * THIS FUNCTION GIVES THE ANTENNA SUM PATTERN WEIGHTING OF THE *
C  * RADAR SIGNAL FOR THE GIVEN ANGLE(IN RADIANS) OFF BORESIGHT   *
C  ***********************************************************
C
C
        FUNCTION SPAT(X)
C  NOTE: THE FOLLOWING VALUE OF B GIVES THE SUM PATTERN A SINGLE-SIDED
C          3 DB BEAMWIDTH OF 0.85 DEGREES.
        Y=93.80*X
        TEMP=ABS(Y)
        IF(TEMP.GT.1.0E-06) GO TO 10
        SPAT=1.0
        RETURN
   10   SPAT=SIN(Y)/Y
        RETURN
        END
C
C
C  ***************************************************************
C  * THIS FUNCTION COMPUTES THE EXPRESSION (SIN(NX)**2/(N SIN(X)**2)) *
C  ***************************************************************
C
C
        FUNCTION SUM(X,N)
        Y=SIN(X)**2
        IF(Y.GT.1.0E-08) GO TO 10
        SUM=N
        RETURN
   10   SUM=SIN(N*X)**2/(N*Y)
        RETURN
        END
C
C
C  ***************************************************************
C  * THIS SUBROUTINE RESETS THE SYSTEM UNDER THE FOLLOWING CONDITIONS *
C  * (1) BREAK-TRACK (TO SEARCH), (2) PASSIVE/ACTIVE MODE CHANGE (TO  *
C  * SEARCH), AND (3) SYSTEM IN STANDBY (TO IDLE).                    *
C  ***************************************************************
C
C
        SUBROUTINE SYSINT
```

```
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)
      COMMON /OUTPUT/MSWF,MTF,MSF,SRNG,SRDOT,SPANG,SRANG,SPRTE,SRRTE,
     2               SSRS,MADVF,MRDVF,MARDVF,MRRDVF
      COMMON /ICNTL/IOLDPW,IOLDMD,IOLDSM,ISHOLD,KMSCLK,KWMUP,KSNCLK,
     2             KSNMAX,KACCLK,MTP,MZ1,MZ0,MSS,MTKINT,MRNG,MSAM,MPRF,
     3             MBKTRK,MBTSUM,MBT(8)
      COMMON /ATDAT/DUM1(4),ALRATE,BTRATE,DUM2(2),AL,BT,PREF,RREF
C
C  ***************************************************
C  * STEP 1: INITIALIZE ALL INTERNAL FLAGS AND CONTROLS *
C  ****************************************************
      IOLDMD=IMODE
      IOLDSM=IASM
      ISHOLD=0
      MTP=1
      MZ1=0
      MZ0=0
      MSS=0
      MTKINT=0
C
C  ****************************************
C  * STEP 2: INITIALIZE ALL INTERNAL CLOCKS *
C  ****************************************
      KACCLK=0
      KSNCLK=0
C
C  **************************************
C  * STEP 3: INITIALIZE ALL DISPLAY FLAGS *
C  **************************************
      MSWF=0
      MSF=0
      MTF=0
      MADVF=0
      MRDVF=0
      MRRDVF=0
      MARDVF=0
C
C  ***************************************
C  * STEP 4: INITIALIZE ALL DISPLAY METERS *
C  ***************************************
      SRNG=0.0
      SRDOT=0.0
      SPRTE=0.0
      SRRTE=0.0
      SRSS=0.0
C
C  *******************************************
C  * STEP 5: INITIALIZE GIMBAL POINTING LOOP *
C  *******************************************
      PII=3.14159265/180.
      ALRATE=0.0
      BTRATE=0.0
      IF(IPWR.NE.1.AND.KMSCLK.NE.1) GO TO 5
C
C  STEP 5-1: IF SYSTEM POWER OFF THEN ALIGN BORESIGHT WITH ZENITH.
      PREF=0.0
      RREF=0.0
      AL=0.0
      BT=0.0
      SPANG=0.0
      SRANG=0.0
      IOLDPW=IPWR
      RETURN
    5 IF(IPWR.GT.2) GO TO 15
```

```
C                                                                       00004840
C  STEP 5-2: IF SYSTEM IN STANDBY THEN HOLD GIMBALS AT POSITION WHEN    00004850
C            STANDBY ENTERED AND ZERO DISPLAYS.                         00004860
      IF(IOLDPW.EQ.IPWR) GO TO 10                                       00004870
      PREF=PII*SPANG                                                    00004880
      RREF=PII*SRANG                                                    00004890
   10 SPANG=0.0                                                         00004900
      SRANG=0.0                                                         00004910
      IOLDPW=IPWR                                                       00004920
      RETURN                                                            00004930
C                                                                       00004940
C  STEP 5-3: PREPARE GIMBAL LOOP FOR ENTRY INTO ANY OF SEARCH MODES.    00004950
   15 PREF=PII*SPANG                                                    00004960
      RREF=PII*SRANG                                                    00004970
      IOLDPW=IPWR                                                       00004980
      RETURN                                                            00004990
      END                                                               00005000
C                                                                       00017190
C                                                                       00017200
C  ****************************************************                 00017210
C  * THIS SUBROUTINE UPDATES THE DATA VALID FLAG STATUS *               00017220
C  ****************************************************                 00017230
C                                                                       00017240
C                                                                       00017250
      SUBROUTINE TGTACQ                                                 00017260
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)                00017270
      COMMON /OUTPUT/MSWF,MTF,MSF,DUM1(7),MADVF,MRDVF,MARDVF,MRRDVF     00017280
      COMMON /ICNTL/IDUM3(8),KACCLK,MTP,MZ1,MZ0,MSS,MTKINT,             00017290
     2              MRNG,IDUM4(12)                                      00017300
      COMMON /SYSDAT/TS,DUMS(14)                                        00017310
      DIMENSION ADV(10,2),RDV(10,2),ARDV(10,2)                          00017320
      DATA ADV/9*1.02,5.12,8*1.02,2*2.33/                               00017330
      DATA RDV/9*6.15,28.69,8*6.97,2*29.76/                             00017340
      DATA ARDV/9*8.2,28.69,7*8.2,26.23,2*29.76/                        00017350
C                                                                       00017360
C  ************************************                                 00017370
C  * STEP 1: UPDATE ACQUISITION CLOCK *                                 00017380
C  ************************************                                 00017390
      KACCLK=KACCLK+1                                                   00017400
      ACCLK=KACCLK*TS                                                   00017410
C                                                                       00017420
C  ******************************************************************  00017430
C  * STEP 2: PERFORM ANGLE DATA VALID TEST --- GPC-ACQ + AUTO ONLY *    00017440
C  ******************************************************************  00017450
      IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 10                               00017460
      IF(ACCLK.LT.ADV(MRNG,IMODE))  GO TO 10                            00017470
      MADVF=1                                                           00017480
C                                                                       00017490
C  *******************************************************             00017500
C  * STEP 3: PERFORM RANGE AND RANGE RATE DATA VALID TEST *             00017510
C  *******************************************************             00017520
   10 IF(ACCLK.LT.RDV(MRNG,IMODE))  GO TO 15                            00017530
      MRDVF=1                                                           00017540
      MRRDVF=1                                                          00017550
C                                                                       00017560
C  IF GPC-DES OR MANUAL INITIALIZE RADAR TRACKING PARAMETERS.           00017570
CCCCCCCCCCCCCCCCCCC MOD MAR 24 1983 CCCCCCCCCCCCCCCCCCCCCCCC
   15 IF((IASM.EQ.2.OR.IASM.EQ.4).AND.MRDVF.EQ.1) GO TO 20                  00017580
C                                                                       00017590
C  ******************************************************************  00017600
C  * STEP 4: PERFORM ANGLE RATE DATA VALID TEST --- GPC-ACQ + AUTO *    00017610
C  *         MODES ONLY.                                            *   00017620
C  ******************************************************************  00017630
      IF(ACCLK.LT.ARDV(MRNG,IMODE)) RETURN                              00017640
      MARDVF=1
C
C **************************************************************
C * STEP 5: PERFORM STEADY STATE RADAR TRACKING INITIALIZATION *
C **************************************************************
   20 KACCLK=0
      MTF=1
      RETURN
      END
C
C
C **************************************************************
C * THIS SUBROUTINE GENERATES A (3X3) MATRIX TTH THAT PRODUCES *
C * A ROTATION OF TH RADIANS ABOUT THE X-AXIS.                 *
C **************************************************************
C
C
      SUBROUTINE THETA(TTH,TH)
      DIMENSION TTH(3,3)
      DO 10 I=1,3
      DO 10 J=1,3
10    TTH(I,J)=0.0
      TTH(1,1)=1.0
      TTH(2,2)=COS(TH)
      TTH(3,3)=TTH(2,2)
      TTH(2,3)=SIN(TH)
      TTH(3,2)=-TTH(2,3)
      RETURN
      END
C
C
C **************************************************************
C * THIS SUBROUTINE INITIALIZES THE ANGLE TRACKING LOOPS, THE  *
C * RANGE TRACKING LOOP, AND THE VELOCITY PROCESSOR -- STEADY *
C * STATE CONDITIONS ARE ASSUMED.                              *
C **************************************************************
C
C
      SUBROUTINE TKINIT
      REAL INTT,IRNG,IRDOT,IVR
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)
      COMMON /INPUT/ ERT(3),EVT(3),EWB(3),DUM(18)
      COMMON /OUTPUT/ I3DUM(3),SRNG,DUM1(6),IDUM1(4)
      COMMON /ICNTL/I1DUM(13),MTKINT,MRNG,MSAM,MPRF,MBKTRK,MBTSUM,
     2           MBT(8),MPFOLD
      COMMON /SYSDAT/TSAM,DR(3),CP,SP,PSI,PSBIAS,DUM2(7),TRB(3,3)
      COMMON /TGTDAT/NT,DUM5(500),RO(3),ROU(3),CGRNGE,CGVEL
      COMMON /SATDAT/RADAR(3),KTAR,RT(70,3),SIG(70),ROLD,ICLOSE,ICLOLD
      COMMON /ATDAT/CA,SA,CB,SB,AZRATE,ELRATE,ALRATE,BTRATE,AL,BT,
     2           DUM3(2)
      COMMON /RTDAT/IRDOT,IRNG,RBIAS,VEST(4),MDF(5)
      COMMON /XFORMS/ TLB(3,3),TLBD(3,3),TLT(3,3),TLTD(3,3)
      COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD
      DIMENSION        ER(3),EV(3),ERTO(3),FLTWID(3),RI(10)
      DATA FLTWID/7.7215,3.3090,0.2969/
      DATA RI/120.,240.,780.,2552.,5772.,11544.,23089.,43747.,
     2        57722.,1.8228E+6/,NRI/10/,PI/3.141592653/
C
C *******************************************
C * STEP 0: INITIALIZE BREAK-TRACK ALGORITHM *
C *******************************************
C
C STEP 0-1: INITIALIZE MOVING WINDOW-OF-8 REGISTERS.
      DO 3 I=1,8
```

```
    3 MBT(I)=0
C
C STEP 0-2: INITIALIZE SUM REGISTER.
      MBTSUM=0
C
C STEP 0-3: SET BREAK-TRACK FLAG TO LOW (OR 0) STATE.
      MBKTRK=0
C
C ****************************************
C * STEP 1: INITIALIZE ANGLE TRACKING LOOP *
C ****************************************
      IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 5
C
C STEP 1-1: COMPUTE INITIAL INNER AND OUTER GIMBAL POSITIONS.
C   (NOTE: TRANSFORM CONSISTS OF TRANSLATION PLUS ROTATION.)
C   PERFORM TRANSLATION -- SHIFT TO RADAR FRAME ORIGIN.
      DO 1 I=1,3
    1 ERTO(I)=ERT(I)-DR(I)
C   TRANSFORM TARGET POSITION FROM BODY TO RADAR FRAME.
      CALL MULT31(TRB,ERTO,ER)
C   TRANSFORM TARGET VELOCITY FROM BODY TO RADAR FRAME.
      CALL MULT31(TRB,EVT,EV)
      SQ=SQRT(ER(2)*ER(2)+ER(3)*ER(3))
C   COMPUTE INNER(BETA) GIMBAL POSITION -- BT.
      IF(ER(1).EQ.0.0.AND.SQ.EQ.0.0) STOP
      BT=-ATAN2(ER(1),SQ)
      ER2=-ER(2)
      ER3=-ER(3)
C   COMPUTE OUTER(ALPHA) GIMBAL POSITION -- AL.
      IF(ER2.EQ.0.0.AND.ER3.EQ.0.0) GO TO 8
      AL=-ATAN2(ER2,ER3)
      GO TO 9
    8 IF(ER(1).GT.0.0) AL=PI/2.
      IF(ER(1).LT.0.0) AL=-PI/2.
      IF(ER(1).EQ.0.0) STOP
C
C STEP 1-2: COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH AND
C           ELEVATION RATES.
C   PRELIMINARY TRIGONOMETRIC COMPUTATIONS.
    9 CA=COS(AL)
      SA=SIN(AL)
      CB=COS(BT)
      SB=SIN(BT)
C   TRANSFORM BODY ANGULAR VELOCITY VECTOR FROM BODY TO OUTER
C   GIMBAL(G) REFERENCE FRAME.
      WGX=CP*EWB(1)+SP*EWB(2)
      WGY=CA*(-SP*EWB(1)+CP*EWB(2))+SA*EWB(3)
      WGZ=-SA*(-SP*EWB(1)+CP*EWB(2))+CA*EWB(3)
C   COMPUTE THE RANGE TO TARGET.
      R=SQRT(ER(1)*ER(1)+ER(2)*ER(2)+ER(3)*ER(3))
C   COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH RATE(AZRATE).
      VGY=CA*EV(2)+SA*EV(3)
      AZRATE=VGY/R+(CB*WGX-SB*WGZ)
C   COMPUTE INITIAL TARGET INERTIAL LOS ELEVATION RATE(ELRATE).
      ELRATE=-(CB*EV(1)-SB*(-SA*EV(2)+CA*EV(3)))/R+WGY
C
C STEP 1-3: COMPUTE INITIAL INNER AND OUTER GIMBAL RATES.
C   COMPUTE INITIAL OUTER GIMBAL RATE(ALRATE).
      RCB=R*CB
      IF(ABS(RCB).LT.1.0E-6) GO TO 2
      ALRATE=VGY/RCB
      GO TO 4
    2 ALRATE=0.
    4 CONTINUE
```

```
C   COMPUTE INITIAL INNER GIMBAL RATE(BTRATE).
      BTRATE=ELRATE-WGY
C
C *****************************************
C * STEP 2: INITIALIZE RANGE TRACKING LOOP *
C *****************************************
C
C STEP 2-1: TRANSFORM TARGET C.G. POSITION AND C.G. VELOCITY FROM
C           BODY TO ANTENNA LOS FRAME.
    5 CALL TRNSFM
      CALL PVTRAN
C
C STEP 2-2: INITIALIZE THE RANGE ESTIMATE REGISTER.
      SRNG=CGRNGE
      IRNG=INTT(SRNG*3.2+0.5)
C
C STEP 2-3: INITIALIZE THE RANGE RATE ESTIMATE REGISTER.
      IRDOT=INTT(CGVEL*TSAM*3.2+0.5)
C
C *************************************************************
C * STEP 3: SET OPERATING PARAMETERS BASED UPON INITIAL RANGE *
C *         AND SYSTEM MODE.                                  *
C *************************************************************
C
C STEP 3-1: DETERMINE CORRECT RANGE INTERVAL.
      DO 30 I=1,NRI
      MRNG=I
      IF(RI(I) .GT. SRNG) GO TO 40
30    CONTINUE
C
C STEP 3-2: DETERMINE CORRECT SAMPLE RATE.
40    IF(IMODE.GE.2)  GO TO 44
      IF(MRNG.GT.9) GO TO 42
      MSAM=1
      GO TO 50
42    MSAM=2
      GO TO 50
44    IF(MRNG.GT.4) GO TO 46
      MSAM=1
      GO TO 50
46    MSAM=2
C
C STEP 3-3: DETERMINE CORRECT PRF.
50    IF(IMODE.GE.2) GO TO 54
      IF(MRNG.GT.9) GO TO 52
      MPRF=1
      GO TO 60
52    MPRF=3
      GO TO 60
54    IF(MRNG.GT.9) GO TO 56
      MPRF=1
      GO TO 60
56    MPRF=2
60    CONTINUE
C
C STEP 3-4: SET PRF TRANSITION FLAG.
      MPFOLD=MPRF
C
C *****************************************
C * STEP 4: INITIALIZE VELOCITY PROCESSOR *
C *****************************************
C
C STEP 4-1: INITIALIZE MOVING WINDOW VELOCITY AVERAGING.
      DO 10 I=1,4
```

```
10    VEST(I)=CGVEL*20.
C
C  STEP 4-2: SET INITIAL POSITION OF 5 DOPPLER FILTERS.
      VR=-CGVEL/FLTWID(MPRF)
      IVR=INTT(VR+0.5)+16000.
      XX=AMOD(IVR,32.)
      MDF(3)=INT(XX)
      DO 20 I=1,5
      MD=MDF(3)+I-3+160
20    MDF(I)=MOD(MD,32)
C
C  ****************************
C  * STEP 5: INITIALIZE AGC LOOP *
C  ****************************
      AGCO=1.0
      ITXP=1
C
C  ************************************************************
C  * STEP 6: SET TRACK INDICATOR TO ALLOW OPERATION OF TRAQK LOOP *
C  ************************************************************
      MTKINT=1
C
      ROLD=0.
      ICLOSE=0
      ICLOLD=0
C
C  NOTE: DEBUGGING PRINT STATEMENTS.
C     WRITE(6,899)
C     WRITE(6,900) AZRATE,ELRATE,ALRATE,BTRATE,AL,BT
C     WRITE(6,901)
C     WRITE(6,902) IRNG,IRDOT,SRNG
C     WRITE(6,903)
C     WRITE(6,904) (VEST(I),I=1,4),(MDF(J),J=1,5)
C     WRITE(6,905)
C     WRITE(6,906) IMODE,MRNG,MSAM,MPRF
  899 FORMAT(//' TRACKER INITIALIZATION:'/' ATRACK: AZRATE',
     2 ',ELRATE,ALRATE,BTRATE,AL,BT')
  900 FORMAT(6F14.6)
  901 FORMAT(' RTRACK: IRNG,IRDOT,SRNG')
  902 FORMAT(2I8,F14.6)
  903 FORMAT(' VTRACK: VEST,MDF')
  904 FORMAT(4F14.6,5I8)
  905 FORMAT(' CNTL: IMODE,MRNG,MSAM,MPRF')
  906 FORMAT(4I8//)
      RETURN
      END
C
C
C  ************************************************************
C  * THIS SUBROUTINE SIMULATES THE TRACKING MODES OF THE KU-BAND *
C  * RADAR.                                                     *
C  ************************************************************
C
C
      SUBROUTINE TRACK
      COMMON /CNTL/IDUM(3),IASM,ISRCHC,ISRCHG,IAZS,IELS,ISLR,EDRNG,
     2             EDPA,EDRA
      COMMON /OUTPUT/MSWF,MTF,MSF,DUMO(7),IDUMO(4)
      COMMON /ICNTL/IIDUM(13),MTKINT,MRNG,MSAM,MPRF,MBKTRK,IDUM2(9)
      COMMON /SYSDAT/TSAM,DUM2(14)
      COMMON /ATDAT/DUM1(10),PREF,RREF,DUMA(2)
      DIMENSION SLWRTE(2)
      DATA SLWRTE/6.9814E-3,3.4907E-1/
C
```

```
C     ***************************************************************
C     * STEP 1: INITIALIZE TRACK MODE --- INITIALIZE ALL TRACK LOOPS *
C     *         AND UPDATE STATUS OF DATA VALID FLAGS.               *
C     ***************************************************************
C
C     STEP 1-1: IF TRACK LOOPS INITIALIZED(MTKINT=1) SKIP STEP 1-2 AND IF
C               ALL DATA VALID FLAGS ARE UP(MTF=1) SKIP STEP 1-2 AND 1-3.
          IF(MTF.EQ.1) GO TO 6
          IF(MTKINT.NE.0) GO TO 5
C
C     STEP 1-1: INITIALIZE RANGE,ANGLE,AND VELOCITY TRACK LOOPS --- ASSUMES
C                STEADY STATE TRACKING OF TARGET C.G.
          CALL TKINIT
C
C     STEP 2-1: UPDATE DATA VALID FLAG STATUS --- ONLY WHEN ENTERING
C               TRACK FROM SEARCH.
        5 CALL TGTACQ
C
C     ***********************************************
C     * STEP 2: PERFORM TRACKING LOOP UPDATE PROCEDURE *
C     ***********************************************
C
C     STEP 2-1: UPDATE TRANSFORMATION MATRICES AND MATRICE RATES.
        6 CALL TRNSFM
C
C     STEP 2-2: TRANSFORM TARGET POSITION AND VELOCITY COMPONENTS FROM
C               ORBITER BODY FRAME-TO-ANTENNA LOS FRAME.
          CALL PVTRAN
C
C     STEP 2-3: GENERATE NOISE-FREE TARGET RETURN SIGNAL AND PROCESS
C               SIGNAL TO PRODUCE NOISE-FREE DISCRIMINANT COMPONENTS.
          CALL SIGNAL
C
C     STEP 2-4: ADD EQUIVALENT NOISE TO DISCRIMINANT COMPONENTS AND FORM
C               ALL REQUIRED DISCRIMINANTS.
          CALL DISCRM
C
C     STEP 2-5: UPDATE STATUS OF BREAK-TRACK FLAG.
          CALL BRKTRK
C
C     STEP 2-6: CHECK STATUS OF BREAK-TRACK FLAG --- IF BREAK-TRACK FLAG
C               UP (MBKTRK=1) RESET SYSTEM AND RETURN TP SEARCH.
          IF(MBKTRK.NE.1) GO TO 7
          CALL SYSINT
          RETURN
C
C     STEP 2-7: DETERMINE RADAR SIGNAL STRENGTH (FOR DISPLAY METER)
C               AND UPDATE AGC VALUE.
        7 CALL RSS
C
C     STEP 2-8: UPDATE ANTENNA GIMBAL POSITIONS AND RATES AND TARGET
C               ANGLES AND ANGLE RATES FOR DISPLAY (GPC-ACQ AND AUTO
C               MODES ONLY.)
          IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 10
C
C     STEP 2-8A: IF IN GPC-ACQ OR AUTO MODE USE RADAR ESTIMATED TARGET
C                ANGLES AS GIMBAL TRACK SERVO INPUT.
          CALL ATRACK
          GO TO 15
       10 IF(IASM.EQ.4) GO TO 12
C
C     STEP 2-8B: IF IN GPC-DES MODE USE GPC-SUPPLIED ANGLE DESIGNATES AS
C                GIMBAL TRACK SERVO INPUT.
          PREF=EDPA
```

```
      RREF=EDRA
      CALL POINT
      GO TO 15
C
C  STEP 2-8C: IF IN MANUAL MODE USE CREW-SUPPLIED SLEW RATES TO DETER
C             MINE GIMBAL TRACK SERVO INPUT.
   12 PREF=PREF+FLOAT(IELS)*SLWRTE(ISLR+1)*TSAM
      RREF=RREF+FLOAT(IAZS)*SLWRTE(ISLR+1)*TSAM
      CALL POINT
C
C  STEP 2-9: UPDATE THE RANGE AND RANGE RATE ESTIMATES.
   15 CALL RTRACK
C
C  STEP 2-10: UPDATE ACCURATE VELOCITY ESTIMATE USING VELOCITY
C            PROCESSOR.
      CALL VELPRO
C
C  STEP 2-11: UPDATE ALL RADAR INTERNAL CONTROLS.
      CALL CNTRLS
   20 RETURN
      END
C
C
C  *******************************************************
C  * THIS SUBROUTINE UPDATES ALL REQUIRED TRANSFORMATION *
C  * MATRICES AND TRANSFORMATION MATRIX RATES.           *
C  *******************************************************
C
C
      SUBROUTINE TRNSFM
      COMMON /INPUT/DUM(9),TBT(3,3),TBTD(3,3)
      COMMON /SYSDAT/DUM2(4),CP,SP,DUM4(9),TRB(3,3)
      COMMON /ATDAT/CA,SA,CB,SB,DUM1(2),ALRATE,BTRATE,AL,BT,DUM3(4)
      COMMON /XFORMS/TLB(3,3),TLBD(3,3),TLT(3,3),TLTD(3,3)
      DIMENSION TLR(3,3)
C
C  ****************************************
C  * STEP 1: UPDATE TRANSFORMATION MATRICES *
C  ****************************************
C
C  STEP 1-1: PRELIMINARY COMPUTATIONS.
      CB=COS(BT)
      SB=SIN(BT)
      CA=COS(AL)
      SA=SIN(AL)
C
C  STEP 1-2: COMPUTE TRANSFORMATION MATRIX TLB (BODY-TO-LOS FRAME).
      TLR(1,1)=CB
      TLR(1,2)=SB*SA
      TLR(1,3)=-SB*CA
      TLR(2,1)=0.0
      TLR(2,2)=CA
      TLR(2,3)=SA
      TLR(3,1)=SB
      TLR(3,2)=-CB*SA
      TLR(3,3)=CB*CA
      CALL MULT33(TLR,TRB,TLB)
C
C  STEP 1-3: COMPUTE TRANSFORMATION MATRIX TLT (TARGET-TO-LOS FRAME).
      CALL MULT33(TLB,TBT,TLT)
C
C  *********************************************
C  * STEP 2: UPDATE TRANSFORMATION MATRIX RATES *
C  *********************************************
```

```
C
C  STEP 2-1: COMPUTE TLB-DOT.
      TLBD(1,1)=-BTRATE*TLB(3,1)+ALRATE*SB*TLB(2,1)
      TLBD(1,2)=-BTRATE*TLB(3,2)+ALRATE*SB*TLB(2,2)
      TLBD(1,3)=-BTRATE*TLB(3,3)+ALRATE*SB*TLB(2,3)
      TLBD(2,1)=ALRATE*SP*TLB(2,3)
      TLBD(2,2)=-ALRATE*CP*TLB(2,3)
      TLBD(2,3)=ALRATE*CA
      TLBD(3,1)=BTRATE*TLB(1,1)-ALRATE*CB*TLB(2,1)
      TLBD(3,2)=BTRATE*TLB(1,2)-ALRATE*CB*TLB(2,2)
      TLBD(3,3)=BTRATE*TLB(1,3)-ALRATE*CB*TLB(2,3)
C
C  STEP 2-2: COMPUTE TLT-DOT.
      DO 20 I=1,3
      DO 20 J=1,3
      TLTD(I,J)=0.0
      DO 20 K=1,3
   20 TLTD(I,J)=TLTD(I,J)+TLBD(I,K)*TBT(K,J)+TLB(I,K)*TBTD(K,J)
      RETURN
      END
C
C
C  *****************************************************************
C  * THIS SUBROUTINE COMPUTES AN ACCURATE, SMOOTHED VELOCITY USING *
C  * THE KU-BAND RADAR VELOCITY PROCESSOR ALGORITHM.               *
C  *****************************************************************
C
C
      SUBROUTINE VELPRO
      REAL IRDOT,IRNG,INTT,IVEL,IVDISC,IFVEL,IRVEL,IR1,IR2,IR3,
     2     IF3,IDELTA
      COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)
      COMMON /OUTPUT/IDUM0(3),SRNG,SRDOT,DUM2(5),IDUM(4)
      COMMON /ICNTL/I1DUM(14),MRNG,MSAM,MPRF,IDUM1(10),MPFOLD
      COMMON /SYSDAT/TSAM,DUMS(14)
      COMMON /RTDAT/IRDOT,IRNG,RBIAS,VEST(4),MDF(5)
      COMMON /DSCRM/DUM(2),RDISC,VDSC,RRTE,ODISC,DUM3(3)
      DIMENSION IPROM(128),VT1(3),VT2(3),MW(4,3)
      DATA IPROM/127,127,125,124,122,121,120,118,117,116,114,113,
     2 111,110,109,107,106,105,103,102,101,99,98,97,95,94,93,92,90,
     3 89,88,87,85,84,83,82,81,79,78,77,76,75,73,72,71,70,69,68,67,
     4 66,65,64,63,62,61,60,59,58,57,56,55,54,53,52,51,50,49,49,48,
     5 47,46,45,44,44,43,42,41,41,40,39,38,38,37,36,36,35,34,34,33,
     6 32,32,31,31,30,30,29,28,28,27,27,26,26,25,25,24,24,23,23,22,
     7 22,22,21,21,20,20,19,19,19,18,18,17,17,17,16,16,16,15,15/
      DATA VT1/1.012592E-2,2.362726E-2,2.633237E-1/,VT2/1.204935,
     2        0.5163982,0.04633489/
      DATA MW/1,2,3,4,1,1,2,2,1,1,1,1/
C
C  ************************************************
C  * STEP 1: GENERATE AMBIGUOUS VELOCITY ESTIMATE *
C  ************************************************
C
C  STEP 1-1: INTEGERIZE VELOCITY DISCRIMINANT AND CHECK FOR SATURATION.
      VDISC=5.333333*VDSC
      IVDISC=INTT(VDISC+0.5)
      IF(IVDISC.LT.-128.) IVDISC=-128.
      IF(IVDISC.GT.127.) IVDISC=127.
C
C  STEP 1-2: COMPUTE INTEGRAL FILTER NUMBER PORTION OF AMBIGUOUS
C         VELOCITY ESTIMATE.
      INTEG=MDF(2)
      IF(IVDISC.LT.0.) INTEG=MOD(INTEG+1,32)
C
```

```
C  STEP 1-3: COMPUTE FRACTIONAL FILTER PORTION OF AMBIGUOUS VELOCITY      00027450
C                ESTIMATE.                                                 00027460
C  ESTIMATE.                                                               00027470
       IV1=INT(ABS(IVDISC))+1                                              00027480
       IFRAC=IPROM(IV1)                                                    00027490
       IF(IVDISC.LT.0.) IFRAC=127-IFRAC                                    00027500
C                                                                          00027510
C  STEP 1-4: COMPUTE AMBIGUOUS VELOCITY ESTIMATE --- COMBINE INTEGRAL      00027520
C                AND FRACTIONAL PARTS. NOTE: LSB IS 1/128 OF FILTER WIDTH. 00027530
C  FRACTIONAL PARTS. NOTE: LSB IS 1/128 OF A FILTER WIDTH.                 00027540
       IFVEL=FLOAT(IFRAC+128*INTEG)                                        00027550
C                                                                          00027560
C  ***************************************                                 00027570
C  * STEP 2: SCALE ROUGH VELOCITY ESTIMATE *                               00027580
C  ***************************************                                 00027590
C                                                                          00027600
C  STEP 2-1: SCALE LSB OF ROUGH RANGE RATE ESTIMATE TO 4 TIMES A DOPPLER00027610
C                WIDTH.                                                    00027620
C  DEFINITION:  VT1(MPRF)=(RANGE LSB)/((MAX. UNAMBIGUOUS VELOCITY)/8)      00027630
C                   OR VT1(MPRF)=5./(PRF*LAMBDA)                           00027640
       R1=IRDOT*VT1(MPRF)/TSAM                                             00027650
       IR1=AINT(R1)                                                        00027660
C                                                                          00027670
C  STEP 2-2: PERFORM SOME REQUIRED AUXILIARY CALCULATIONS.                 00027680
       R2=IR1/8.                                                           00027690
       IR2=AINT(R2)                                                        00027700
       IRVEL=IR2*4096.                                                     00027710
C                                                                          00027720
C  ****************************                                            00027730
C  * STEP 3: RESOLVE AMBIGUITY *                                           00027740
C  ****************************                                            00027750
C                                                                          00027760
C  STEP 3-1: COMPUTE 3 MSB'S OF AMBIGUOUS VELOCITY ESTIMATE.               00027770
       IF3=AINT(IFVEL/512.)                                                00027780
C                                                                          00027790
C  STEP 3-2: COMPUTE 3 LSB'S OF SCALED ROUGH RANGE RATE ESTIMATE.          00027800
       IR3=ABS(IR1-8.*IR2)                                                 00027810
       IF(R1.LE.0.)GO TO 10
       IRVEL=IRVEL+4096.                                                   00027830
       IR3=7.-IR3                                                          00027840
   10  CONTINUE                                                            00027850
C                                                                          00027860
C  STEP 3-3: COMPARE 3 MSB'S AND 3 LSB'S AND INCREMENT NUMBER OF           00027870
C                AMBIGUOUS FILTER BANK WIDTHS APPROPRIATELY.               00027880
       IDELTA=IR3-IF3                                                      00027890
       IF(IDELTA.GE.4.) IRVEL=IRVEL-4096.                                  00027900
       IF(IDELTA.LE.-4.) IRVEL=IRVEL+4096.                                 00027910
C                                                                          00027920
C  *************************************************                       00027930
C  * STEP 4: COMPUTE UNAMBIGUOUS VELOCITY ESTIMATE. *                      00027940
C  *************************************************                       00027950
C                                                                          00027960
C  STEP 4-1: ADD NUMBER OF AMBIGUOUS FILTER BANK WIDTHS TO ESTIMATE        00027970
C                OF FRACTIONAL FILTER BANK WIDTH. NOTE: LSB OF RESULTANT   00027980
C                ESTIMATE REPRESENTS 1/4096 OF A FILTER BANK WIDTH.        00027990
       IVEL=INTT(IRVEL-IFVEL)                                              00028000
C                                                                          00028010
C  STEP 4-2: SCALE LSB OF RESULTANT ESTIMATE TO 0.05 FEET/SEC.             00028020
C  DEFINITION:  VT2(MPRF)=((FILTER SEPARATION)/128.)/(VELOCITY LSB)        00028030
C                   OR VT2(MPRF)=(PRF*LAMBDA)/(0.05*8196).                 00028040
       IVEL=INTT(IVEL*VT2(MPRF)+0.5)                                       00028050
C                                                                          00028060
C  *************************************************                       00028070
C  * STEP 5: COMPUTE SMOOTHED UNAMBIGUOUS VELOCITY *                       00028080
```

```
C  ***********************************************                          00028090
C                                                                           00028100
C  STEP 5-1: UPDATE REGISTERS OF MOVING WINDOW AVERAGER.                    00028110
      DO 20 I=1,3                                                           00028120
   20 VEST(5-I)=VEST(4-I)                                                   00028130
      VEST(1)=IVEL                                                          00028140
C                                                                           00028150
C  STEP 5-2: COMPUTE MOVING WINDOW AVERAGE AND SCALE ANSWER INTO            00028160
C            FEET/SEC FROM UNITS OF 0.05 FEET/SEC.                          00028170
      M=MPRF                                                                00028178
      M1=MW(1,M)
      M2=MW(2,M)
      M3=MW(3,M)
      M4=MW(4,M)
      SRDOT=0.0125*(VEST(M1      )+VEST(M2      )+VEST(M3      )+           00028180
    2              VEST(M4      ))                                          00028182
C                                                                           00028190
C  ***********************************                                      00028200
C  * STEP 6: RESET DOPPLER FILTER BANK *                                    00028210
C  ***********************************                                      00028220
C                                                                           00028230
C  STEP 6-1: USE ON-TARGET DISCRIMINANT AND VELOCITY DISCRIMINANT TO        00028240
C            DETERMINE UPDATE OF FILTER BANK POSITION.                      00028250
C            THE FOLLOWING RULES ARE USED:                                  00028260
C                                                                           00028270
C            CASE 1: ODISC>0. AND -51.<IVDISC<51. IMPLIES NO CHANGE.        00028280
C                                                                           00028290
C            CASE 2: ODISC>0. AND IVDISC>51. IMPLIES SHIFT -1.              00028300
C                                                                           00028310
C            CASE 3: ODISC>0. AND IVDISC<-51. IMPLIES SHIFT +1.             00028320
C                                                                           00028330
C            CASE 4: ODISC<0. AND IVDISC>0. IMPLIES SHIFT -2.               00028340
C                                                                           00028350
C            CASE 5: ODISC<0. AND IVDISC<0. IMPLIES SHIFT +2.               00028360
      IF(ODISC.GE.0.) GO TO 30                                              00028370
      IF(IVDISC.LT.0.) MDF(1)=MOD(MDF(1)+2,32)                              00028380
      IF(IVDISC.GE.0.) MDF(1)=MOD(MDF(1)+30,32)                             00028390
      GO TO 40                                                              00028400
   30 IF(IVDISC.GT.51.) MDF(1)=MOD(MDF(1)+31,32)                            00028410
      IF(IVDISC.LT.-51.) MDF(1)=MOD(MDF(1)+1,32)                            00028420
C                                                                           00028430
C  STEP 6-2: RESET REMAINING FILTERS IN THE BANK-OF-5.                      00028440
   40 DO 50 I=1,4                                                           00028450
   50 MDF(I+1)=MOD(MDF(1)+I,32)                                             00028460
      RETURN                                                                00028470
      END                                                                   00028480
C                                                                           00012320
C                                                                           00012330
C  ****************************************************************         00012340
C  * THIS SUBROUTINE DETERMINES WHETHER ANTENNA IS IN ZONE 1 AND/OR *       00012350
C  * ZONE 0 (FOR GPC-ACQ AND GPC-DES POINTING MODES ONLY).          *       00012360
C  ****************************************************************         00012370
C                                                                           00012380
C                                                                           00012390
      SUBROUTINE ZONECK                                                     00012400
      COMMON /CNTL/IDUMC(9),EDRNG,EDPA,EDRA                                 00012410
      COMMON /OUTPUT/IDUM1(3),DUM1(2),SPANG,SRANG,DUM3(3),IDUM3(4)          00012420
      COMMON /ICNTL/IDUM2(10),MZ1,MZ0,IDUM4(15)                             00012430
      MZ0=0                                                                 00012440
      MZ1=1                                                                 00012450
      PII=3.141592653/180.                                                  00012460
      RB=-PII*SRANG                                                         00012470
      PB=-PII*SPANG                                                         00012480
      P=-EDPA                                                               00012490
      R=-EDRA
      CPB=COS(PB)
      SPB=SIN(PB)
      CRB=COS(RB)
      SRB=SIN(RB)
      CP=COS(P)
      SP=SIN(P)
      CR=COS(R)
      SR=SIN(R)
      ANGDIF=ACOS(SPB*CRB*SP*CR+SRB*SR+CPB*CRB*CP*CR)/PII
      ANGDIF=ABS(ANGDIF)
      IF(ANGDIF.GT.3.0) RETURN
      MZ0=1
      IF(ANGDIF.GT.0.3) RETURN
      MZ1=1
      RETURN
      END
C
C  SES SMM MODEL AS OF JANUARY 13,1982
C
      SUBROUTINE SMM
C  II.     DIMENSION ARRAYS & DATA STATEMENTS
C     A) DIMENSION STATEMENTS
C       REAL KSEED
      COMMON /SATDAT/RADAR(3),KTAR,R(70,3),SIG(70),ROLD,ICLOSE,ICLOLD
      DIMENSION SIGMA(49),TARG(49,3),PHIMIN(49,3),PHIMAX(49,3)
      DIMENSION OFFSET(49),JHOT(49),JHOT20(49),PHI(49),FG(3)
      DIMENSION VECT(3),COSPHI(49,3),COSPHN(49),ORIENT(49,3)
      DIMENSION ALPH(19,3),V(19,3),DIM(19,3),WRAN(19,3),SDMAX(19,3)
      DIMENSION WSCALE(19,3),DPHI(19),PHIOLD(19),VOLD(19,3),KSEED(19,3)
      DIMENSION TTRAN(3),ABG(19,3),TMAX(49),PL(49),SDMIN(19,3)
C
C     B) DATA STATEMENTS
C
C       1. KSEED- SEEDS FOR RANDOM NUMBER GENERATOR "ZUDU".
      DATA KSEED/45,678,908,607,5678,897,345,7777,67,4,
     1 560,809,444,888,999,555,222,70,80,8000,
     2 5,15,25,35,45,55,65,75,85,95,
     3 7,17,27,37,47,57,67,77,87,97,
     4 9876,984,6666,2398,76,412,7589,409,899,561,
     5 205,3895,9457,9643,937,656,453/
C
C       2. DIM- THE GENERAL SIZE OF EACH DIFFUSE SCATTERER.
      DATA DIM /57*64.8/
C
C       3. WSCALE- WEIGHTING ASSIGNED TO EACH SIDE OF A DIFFUSE
C                 SCATTERER.
      DATA WSCALE/8*10.84,5.9386,2*5.6804,5.9386,5.6804,4*11.1026,
     1 2*6.7958,
     2 2*6.9068,2*2.7111,2*3.6148,2*2.5174,4.3894,2*5.8095,4.3894,
     3 5.8095,4*17.8803,2*6.7958,19*0./
C
C       4. ORIENT- THE i,j,k COMPONENTS OF THE NORMAL VECTOR OF EACH
C                 TARGET.
C         a) i COMPONENT
      DATA ORIENT/13*0.,.9976,-.9976,.9976,-.9976,1.,-1.,
     1 23*0.,.9976,-.9976,.9976,-.9976,1.,2*-1.,
C         b) j COMPONENT
     2 1.,-1.,2*.6428,2*0.,-.6494,-.6361,1.,.4924,.8704,.6428,-1.,.0637,
     3 2*-.0637,.0637,2*0.,1.,-1.,2*.6428,.9272,.5150,.2924,2*0.,-.6494,
     4 -.6361,2*0.,2*1.,.4924,.8704,.4924,.866,-.8660,-1.,0.,-.6428,
     5 .0637,2*-.0637,.0637,3*0.,
C         c) k COMPONENT
     6 2*0.,-.766,.766,1.,-1.,-.7604,.7716,0.,-.8704,.4924,.766,0.,
     7 .0284,2*-.0284,.0284,4*0.,-.766,.766,.3746,.8572,.9563,1.,-1.,
     8 -.7604,.7716,2*0.,2*0.,-.8704,.4924,.8704,-.5,.5,0.,1.,.766,
     9 .0284,2*-.0284,.0284,3*0./
C
C         5. ABG- ARRAY OF TRANSFORMATION ANGLES(RAD), ALPHA, BETA,
C                 GAMMA, FOR DIFFUSE SCATTERERS.
C            a) ALPHA
       DATA ABG/4*3.141593,2*1.570796,2*0.,4*3.141593,0.,1.634563,
     1 -1.50703,1.50703,4.648623,1.570796,4.712389,
C            b) BETA
     2 2*1.570796,2.443392,.6982,0.,3.141593,2.434725,.689444,
     3 1.570796,2.626811,1.055951,.6982,1.570796,1.542392,
     4 2*1.5992,1.542392,2*1.570796,
C            c) GAMMA
     5 4*3.141593,2*1.570796,2*0.,4*3.141593,0.,2*2.723729,.4178642,
     6 2.723729,2*1.570796/
C
C         6. SIGMA- THE CALCULATED RCS FOR EACH TARGET IN M**2.
       DATA SIGMA/2*.1,2*.0154,2*.0274,2*.0133,.0121,2*.0194,.0121,
     2 .0194,4*.7026,2*.0606,2*2419.,373.,7.25,21.84,11.14,18.83,
     3 2*663.,2*321.,2*3.63,.92,.97,470.,82.13,470.,2*83.,470.,83.,
     4 6.34,4*16995.,2*146615.,.3322/
C
C         7. TARG- TARGET POSITION (IN X,Y,Z COORDINATES) RELATIVE TO
C                  THE COORDINATE AXIS OF SMM.
C            a) X COORDINATE
       DATA TARG /9*1.394,4*-.774,.270,.231,.270,.231,2.491,-1.497,
     2 3*1.394,.542,3*1.626,4*1.394,2*0.,-.413,-1.149,8*-.774,.270,
     3 .231,.270,.231,2.491,2*-1.497,
C            b) Y COORDINATE
     4 .862,-.862,2*.555,2*0.,2*.555,.748,.439,1.097,-.3614,-.955,
     5 2*2.233,2*-2.233,2*0.,.826,-.826,2*.555,.658,.568,.439,2*0.,
     6 2*.555,2*0.,2*.748,.439,.865,1.097,.865,-.207,-.955,-.684,
     7 -.3614,2*2.233,2*-2.233,3*0.,
C            c) Z COORDINATE
     8 2*0.,-.929,.929,1.058,-1.058,-.878,.878,0.,-.774,.852,.645,
     9 0.,2*.620,2*-.620,4*0.,-.929,.929,.826,.930,.994,1.058,
     A -1.058,-.878,.878,4*0.,-.774,-.258,.852,.272,.903,0.,.581,
     B .645,2*.620,2*-.620,3*0./
C
C         8. PHIMIN- MINIMUM ANGLE OF DEVIATION FROM SMM COORDINATES
C                    RELATIVE TO TARGET NORMAL.
C            a) MINIMUM ANGLE SUBTENDED IN X-DIRECTION
       DATA PHIMIN /13*0.,2.5,174.5,2.5,174.5,0.,90.,11*88.5,
     2 2*89.2,10*88.5,2.5,174.5,2.5,174.5,0.,2*178.5,
C            b) MINIMUM ANGLE SUBTENDED IN Y-DIRECTION
     3 0.,90.,2*48.5,2*0.,129.,128.,0.,59.,149.,128.5,90.,22.5,
     4 2*154.5,22.5,2*0.,0.,178.5,2*48.5,20.5,57.5,71.5,2*88.5,129.,
     5 128.,0.,90.,2*0.,59.,2*149.,2*148.5,178.5,88.5,128.5,22.5,
     6 2*154.5,22.5,3*88.5,
C            c) MINIMUM ANGLE SUBTENDED IN Z-DIRECTION
     7 2*0.,138.5,38.5,0.,90.,138.,38.,0.,149.,59.,38.5,0.,64.5,
     8 2*112.5,64.5,2*0.,2*88.5,138.5,38.5,66.5,29.5,15.5,0.,178.5,
     9 138.,38.,2*0.,2*88.5,149.,2*58.,118.5,58.5,88.5,0.,
     A 38.5,64.5,2*112.5,64.5,3*88.5/
C
C         9. PHIMAX- MAXIMUM ANGLE OF DEVIATION FROM SMM COORDINATES
C                    RELATIVE TO TARGET NORMAL.
C            a) MAXIMUM ANGLE SUBTENDED IN X-DIRECTION
       DATA PHIMAX /13*180.,5.5,177.5,5.5,177.5,90.,180.,11*91.5,
     2 2*90.8,10*91.5,5.5,177.5,5.5,177.5,1.5,2*180.,
C            b) MAXIMUM ANGLE SUBTENDED IN Y-DIRECTION
     3 90.,180.,2*51.5,2*180.,132.,131.,1.5,62.,152.,131.5,180.,25.5,
     4 2*157.5,25.5,2*180.,1.5,180.,2*51.5,23.5,60.5,74.5,2*91.5,132.,
     5 131.,90.,180.,2*1.5,62.,2*152.,2*151.5,180.,91.5,131.5,25.5,
     6 2*157.5,25.5,3*91.5,
C          c) MAXIMUM ANGLE SUBTENDED IN Z-DIRECTION
     7 2*180.,141.5,41.5,90.,180.,141.,41.,180.,152.,62.,41.5,180.,67.5,
     8 2*115.5,67.5,2*180.,2*91.5,141.5,41.5,69.5,32.5,18.5,1.5,180.,
     9 141.,41.,2*180.,2*91.5,152.,2*62.,121.5,61.5,91.5,1.5,41.5,67.5,
     A 2*115.5,67.5,3*91.5/
C
C        10. OFFSET- POSITION OF TARGET SPECULAR PT. RELATIVE TO TARGET
C                    COORDINATES.
       DATA OFFSET /17*0.,2*0.,11*0.,.7486,.8,14*0.,2*0.,.6518/
C
C        11. MISCELLANIOUS
       DATA PL/ 30*1.,2*0.,16*1.,0./
       DATA TMAX/19*90.,11*1.5,2*0.,16*1.5,0./
       DATA NTAR/49/,KWIDE/19/,PI/3.141592653/
       DATA TTRAN/3*0.0/,INIT1/1/
       IF(INIT1.NE.1) GO TO 2
C
C        12. SDMIN- MINIMUM ANGLE OF VIEW; TARGET SHADOWING.
C          a) X-COORDINATE
      DATA SDMIN/2*-0.6828,-1.,-0.7467,2*-1.,-0.7467,12*-1.,
C          b) Y-COORDINATE
     1      19*-1.,
C          c) Z-COORDINATE
     2      19*-1./
C
C         13. SDMAX- MAXIMUM ANGLE OF VIEW; TARGET SHADOWING.
C          a) X-COORDINATE
      DATA SDMAX/8*1.,0.4218,3*1.,0.4218,0.5037,0.6046,0.5037,0.6046,
     1     2*1.,
C          b) Y-COORDINATE
     2      19*1.,
C          c) Z-COORDINATE
     3      19*1./
C
C
C III.      RANDOMIZE DIFFUSE SCATTERER RCS VALUES.
C
        ISEED1=100
        ISEED2=83
        DO 107 I=1,1000
107     X=RNDU(ISEED1,ISEED2)
        DO 108 I=1,KWIDE
        X=RNDU(ISEED1,ISEED2)
108     SIGMA(I)=SIGMA(I)*2.*X
C
C  IV.      CONVERT TARGET DATA APPROPRIATELY.
C
        FTM=0.3048
        DO 101 I=1,NTAR
101     SIGMA(I)=SQRT(SIGMA(I))/FTM
        DO 102 J=1,NTAR
        DO 102 I=1,3
102     TARG(J,I)=TARG(J,I)/FTM
        DO 103 J=1,NTAR
        TMAX(J)=COS( TMAX(J)*PI/180.)
        DO 103 I=1,3
        PHIMIN(J,I)=COS(PHIMIN(J,I)*PI/180.)
103     PHIMAX(J,I)=COS(PHIMAX(J,I)*PI/180.)
        DO 105 I=1,NTAR
105     OFFSET(I)=OFFSET(I)/FTM
C
C   V.      INITIALIZATION OF TARGET POSITION & COUNTING PARAMETERS
```

```
C       NWIDE & KTAR.
C
        DO 1 K=1,NTAR
        DO 1 I=1,3
    1   TARG(K,I)=TARG(K,I)+TTRAN(I)
         INIT1=0
    2   CONTINUE
        NWIDE=0
        KTAR=0
C
C  VI.       DETERMINE WHICH TARGETS ARE ILLUMINATED.
C
C       WRITE(2,500)
  500       FORMAT(1X,'TARGET #',2X,'COSPHN')
        DO 15 I=1,NTAR
C
C       A) DETERMINE THE POSITION OF THE RADAR RELATIVE TO
C          TARGET SPECULAR POINT.
C
C         1. "VECT"- POSITION VECTOR
        DO 5 J=1,3
        VECT(J)=RADAR(J)-TARG(I,J)
    5   CONTINUE
C         2. VNORM- MAGNITUDE OF "VECT".
        VNORM=SQRT(VECT(1)**2+VECT(2)**2+VECT(3)**2)
C
C       B) DETERMINE THE COSINE OF THE ANGLE BETWEEN THE
C          RADAR POSITION RELATIVE TO THE TARGET SPECULAR PT. &
C          TARGET NORMAL.
C
C         1. CALCULATE THE ANGLE BY EMPLOYING THE DOT PRODUCT
C            OF THE TWO VECTORS: "COSPHI" & "ORIENT".
        DP=0.
        DO 7 J=1,3
C         2. COSPHI- UNIT VECTOR OF "VECT"; REPRESENTATIVE OF THE
C            COSINE OF THE ANGLE BETWEEN "VECT" & SMM COORDINATE AXIS.
        COSPHI(I,J)=VECT(J)/VNORM
    7   DP=DP+COSPHI(I,J)*ORIENT(I,J)
C         3. COSPHN- COSINE OF THE ANGLE; RESULT OF THE DOT PRODUCT.
        COSPHN(I)=DP
C
C       C) TEST OF ILLUMINATION- TWO METHODS: COMPARE COSPHN W/TMAX
C          OR COMPARE COMPONENTS OF COSPHI W/PHIMIN & PHIMAX.
C
C         1. PL- A FLAG: 0 INDICATES METHOD 1 & 1 INDICATES METHOD 2.
        IF(PL(I).EQ.0.)GO TO 9
C         2. METHOD 1
        IF(COSPHN(I).LT.TMAX(I))GO TO 15
        GO TO 11
C         3. METHOD 2
    9   DO 10 J=1,3
        IF(COSPHI(I,J).LT.PHIMAX(I,J).OR.COSPHI(I,J).GT.PHIMIN(I,J))
     2  GO TO 15
   10   CONTINUE
C
C       D) TARGET SHADOWING
C
C         1. TEST FIRST 19 TARGETS ONLY.
   11      IF(I.GT.19)GO TO 13
C         2. FIND SHADOWING VECTOR BY TRANSFORMATION OF COSPHI
C            FROM SMMS TO TARGET COORDINATES.
       F1=COSPHI(1,1)*COS(ABG(I,1))+COSPHI(I,2)*SIN(ABG(I,1))
       F2=COSPHI(I,2)*COS(ABG(I,1))-COSPHI(I,1)*SIN(ABG(I,1))
       F3=COSPHI(I,3)
```

```
      FB2=F2*COS(ABG(I,2))+F3*SIN(ABG(I,2))
      FB3=F3*COS(ABG(I,2))-F2*SIN(ABG(I,2))
      FG(1)=F1*COS(ABG(I,3))+FB2*SIN(ABG(I,3))
      FG(2)=FB2*COS(ABG(I,3))-F1*SIN(ABG(I,3))
      FG(3)=FB3
C         3. TEST FOR TARGET SHADOWING.
      DO 12 J=1,3
      IF(FG(J).GT.SDMAX(I,J).OR.FG(J).LT.SDMIN(I,J))GO TO 15
   12    CONTINUE
C
C       E) COUNT NUMBER OF ILLUMINATED TARGETS.
C
C         1. KTAR- # OF TARGETS ILLUMINATED
   13  KTAR=KTAR+1
C         2. JHOT- TARGET IDENTIFICATION NUMBER
       JHOT(KTAR)=I
       SIG(KTAR)=SIGMA(I)
C         3. NWIDE- # OF DIFFUSE SCATTERERS
       IF(I.LE.KWIDE) NWIDE=NWIDE+1
C      WRITE(2,100)I,COSPHN(I)
  100     FORMAT(1X,I3,7X,F6.3)
   15  CONTINUE
C
C VII.      UPDATE RANGE OF RADAR RELATIVE TO EACH TARGETS SPECULAR PT.
C
C       A) RANGE UPDATE
C
       DO 20 K=1,KTAR
       I=JHOT(K)
       DO 20 J=1,3
       R(K,J)=TARG(I,J)+OFFSET(I)*COSPHI(I,J)
   20  CONTINUE
      IEE=1
      IF (IEE.EQ.0)GO TO 24
C
C       B) RE-EVALUATE RCS FOR DIFFUSE SCATTERERS
C
       DO 22 K=1,NWIDE
       I=JHOT(K)
       SIG(K)=SQRT(ABS(COSPHN(I)))*SIGMA(I)
   22  CONTINUE
   24  RANGE=SQRT(RADAR(1)**2+RADAR(2)**2+RADAR(3)**2)
C
C       C) TEST FOR CLOSE RANGE
C
       IF((ROLD.LT..01.OR.RANGE-ROLD.LE.0.).AND.RANGE.LE.270.) ICLOSE=1
       IF(RANGE-ROLD.GT.0..AND.RANGE.GT.300.) ICLOSE=0
C       ICLOSE=0
       IF(ICLOSE.EQ.0.OR.NWIDE.EQ.0) GO TO 55
       IF(ICLOLD.EQ.1) GO TO 35
C
C       D) RANGE UPDATE FOR DIFFUSE SCATTERERS
C
C         1. PERFORMS INITIALIZATION OF DIFFERENCE EQUATIONS
C            FOR ALL DIFFUSE SCATTERERS.
C
       DO 30 I=1,KWIDE
       IF(COSPHN(I).GT.1.)COSPHN(I)=1.
       PHIOLD(I)=ACOS(COSPHN(I))
C          a) "V"- WANDERING VECTOR
       DO 25 J=1,3
       V(I,J)=WSCALE(I,J)*(ZUDU(KSEED(I,J))-.5)
       VOLD(I,J)=V(I,J)
   25  CONTINUE
```

```
C
C         b) TRANSFORMATION OF "V" FROM TARGET COORDINATES TO
C            SMMS COORDINATES.
      TGAM1=V(I,1)*COS(ABG(I,3))-V(I,2)*SIN(ABG(I,3))
      TGAM2=V(I,1)*SIN(ABG(I,3))+V(I,2)*COS(ABG(I,3))
      TBETA2=COS(ABG(I,2))*TGAM2-SIN(ABG(I,2))*V(I,3)
      TBETA3=SIN(ABG(I,2))*TGAM2+COS(ABG(I,2))*V(I,3)
      V(I,1)=COS(ABG(I,1))*TGAM1-SIN(ABG(I,1))*TBETA2
      V(I,2)=SIN(ABG(I,1))*TGAM1+COS(ABG(I,1))*TBETA2
      V(I,3)=TBETA3
      DO 26 J=1,3
      R(I,J)=R(I,J)+V(I,J)
   26 CONTINUE
   30 CONTINUE
      GO TO 55
C
C        2. UPDATES THE ANGLE BETWEEN THE RADAR VECTOR & THE
C           TARGET NORMAL.
   35 DO 40 I=1,KWIDE
      PHI(I)=ACOS(COSPHN(I))
      DPHI(I)=(PHI(I)-PHIOLD(I))
      PHIOLD(I)=PHI(I)
   40 CONTINUE
C
C        3. UPDATES THE RANGE COMPONENTS DUE TO RADAR BEAM
C           DEFLECTION OVER THE SURFACE OF THE DIFFUSE SCATTERER.
C           THE TRANSFORMATION PERFORMS THE SAME FUNCTION DESCRIBED
C           PREVIOUSLY.
      DO 50 K=1,NWIDE
      I=JHOT(K)
      DO 45 J=1,3
      ALPH(I,J)=EXP(-DIM(I,J)*ABS(DPHI(I)*COSPHN(I)))
      WRAN(I,J)=SQRT(1.-ALPH(I,J)**2)*WSCALE(I,J)*(ZUDU(KSEED(I,J))-.5)
      V(I,J)=ALPH(I,J)*VOLD(I,J)+WRAN(I,J)
      VOLD(I,J)=V(I,J)
   45 CONTINUE
      TGAM1=V(I,1)*COS(ABG(I,3))-V(I,2)*SIN(ABG(I,3))
      TGAM2=V(I,1)*SIN(ABG(I,3))+V(I,2)*COS(ABG(I,3))
      TBETA2=COS(ABG(I,2))*TGAM2-SIN(ABG(I,2))*V(I,3)
      TBETA3=SIN(ABG(I,2))*TGAM2+COS(ABG(I,2))*V(I,3)
      V(I,1)=COS(ABG(I,1))*TGAM1-SIN(ABG(I,1))*TBETA2
      V(I,2)=SIN(ABG(I,1))*TGAM1+COS(ABG(I,1))*TBETA2
      V(I,3)=TBETA3
      DO 46 J=1,3
      R(K,J)=R(K,J)+V(I,J)
   46 CONTINUE
   50 CONTINUE
   55 CONTINUE
      ROLD=RANGE
      ICLOLD=ICLOSE
      RETURN
      END
C
C
C
      FUNCTION ZUDU(KSEED)
C     THIS SUBROUTINE GENERATES RANDOM NUMBERS.
      DATA MU/524287/,XMU/524287./,IETA/997/
      IF(KSEED) 20.10,20
   20 CONTINUE
      KSEED=IETA*KSEED
      IKEEP=KSEED/MU
      KSEED=KSEED-IKEEP*MU
      XRAN=KSEED
      XRAN=XRAN/MU
      ZUDU=XRAN
10    RETURN
      END
```

APPENDIX B

SOURCE CODE LISTING OF FINAL DELIVERABLE PROGRAM

This appendix is a listing of the final simulation program delivered at the end of the contract. The program has been installed on the Building 44 VAX system at JSC under the Ku-Band account in the KUBAND.HOWARD.MARK directory. The name of the source program is FINSIM1.

```
C                  **************************************************
C
C                       MODIFIED 01/27/86 TO COMPUTE AND
C                       PLOT REF. RANGE ACCELERATION.
C
C                  *************************************************
C
C********************************************************************************************
C        MDMIN - KUBAND DATA : SSRNG, SSRDOT, SSRANG, SSPANG, SSRRTE, SSPRTE,
C                              SSALP, SSBET
C
C        WHITE SANDS - REF DATA : X, Y, Z, VX, VY, VZ
C
C        REF -> TMR2KU -> ACT : R, ARDOT, SPANG, SRANG, SRRTE, SPRTE,
C                               SALF, SBTA, SAZRTE, SELRTE
C
C        REF -> TMR2KU -> SIM : HRNG, HRDOT, HRANG, HPANG, HRRTE, HPRTE,
C                               HALP, HBET, HELRTE, HALRTE
C
C *******************************************************************************************
      COMMON /TARGET/ITARG,SRCS
      COMMON /ACTDAT/R,ARDOT,SPANG,SRANG,SPRTE,SRRTE,AL,BT,SALF,SBTA
     1  ,ER(3),EV(3),ERTO(3),AZRATE,ELRATE,SAZRTE,SELRTE
     2  ,AX,AY,AZ,AAX,AAY,AAZ,RACCEL
      COMMON /TERM/ITERM,XMO,XDAY,XYR,TBIAS,XJMO,XJDAY,XJYR
      COMMON /OUTPUT/MSWF,MTF,MSF,HRNG,HRDOT,HPANG,HRANG,HPRTE
     2                ,HRRTE,HRSS,MADVF,MRDVF,MARDVF,MRRDVF
     3                ,HALP,HBET
      COMMON /SYSDAT/TS,DUM2(14)
       COMMON /TMR/X,Y,Z,VX,VY,VZ,
     1               DLP(3),DEL(3),DUE(3),
     2               DSU(3),THAZL1,THEL1,THAZU1,A23
      COMMON /INPUT/RO(3),VO(3),ĖWB(3)
      COMMON /ICNTL/IDUM(16),MPRF
       CHARACTER ANS,REPLY
       CHARACTER*11 FPRO(57)
       CHARACTER*40 IXT,LPRO(57)
       CHARACTER*80 COMMENT
       CHARACTER*11 UNIT7
       INTEGER IREF
       INTEGER*2 IS1,IS2
       DİMENSION TP(2001),D(2001,43)
       DIMENSION ITILT(10)
       DIMENSION RNEW(3),ROLD(3),VNEW(3),VOLD(3)
       BYTE IC(120)
C
C      TEST DATA FROM WS32TDATA1
c
       DATA LPRO(1)/' SIM DATA PROFILE HL146AB$'/
       DATA LPRO(2)/' SIM DATA PROFILE HL246AB$'/
DATA LPRO(3)/' SIM DATA PROFILE HJ146AB$'/
DATA LPRO(4)/' SIM DATA PROFILE HEL30AB$'/
DATA LPRO(5)/' SIM DATA PROFILE H30SKAB$'/
DATA LPRO(6)/' SIM DATA PROFILE H30SKAC$'/
DATA LPRO(7)/' SIM DATA PROFILE HEL30AC$'/
DATA LPRO(8)/' SIM DATA PROFILE HEL30AD$'/
DATA LPRO(9)/' SIM DATA PROFILE HL246AC$'/
DATA LPRO(10)/' SIM DATA PROFILE HL346AB$'/
DATA LPRO(11)/' SIM DATA PROFILE HL446AB$'/
DATA LPRO(12)/' SIM DATA PROFILE HL546AB$'/
DATA LPRO(13)/' SIM DATA PROFILE HL546AC$'/
DATA LPRO(14)/' SIM DATA PROFILE HL246AD$'/
DATA LPRO(15)/' SIM DATA PROFILE HL446AC$'/
DATA LPRO(16)/' SIM DATA PROFILE HL146AC$'/
DATA LPRO(17)/' SIM DATA PROFILE HL346AD$'/
DATA LPRO(18)/' SIM DATA PROFILE HJ146AC$'/
DATA LPRO(19)/' SIM DATA PROFILE HEL30AE$'/
DATA LPRO(20)/' SIM DATA PROFILE HEL30AF$'/
DATA LPRO(21)/' SIM DATA PROFILE H30SKAD$'/
DATA LPRO(22)/' SIM DATA PROFILE H30SKAE$'/
DATA LPRO(23)/' SIM DATA PROFILE H30SKAF$'/
DATA LPRO(24)/' SIM DATA PROFILE HEL30AG$'/
DATA LPRO(25)/' SIM DATA PROFILE HEL30AH$'/
DATA LPRO(26)/' SIM DATA PROFILE H30SKAG$'/
DATA LPRO(27)/' SIM DATA PROFILE H30SKAH$'/
DATA LPRO(28)/' SIM DATA PROFILE H30SKAI$'/
DATA LPRO(29)/' SIM DATA PROFILE HEL30AI$'/
DATA LPRO(30)/' SIM DATA PROFILE HEL30AJ$'/
DATA LPRO(31)/' SIM DATA PROFILE HL546AE$'/
DATA LPRO(32)/' SIM DATA PROFILE HL246AE$'/
DATA LPRO(33)/' SIM DATA PROFILE HL446AD$'/
DATA LPRO(34)/' SIM DATA PROFILE HL146AD$'/
DATA LPRO(35)/' SIM DATA PROFILE HL346AE$'/
DATA LPRO(36)/' SIM DATA PROFILE HJ146AD$'/
DATA LPRO(37)/' SIM DATA PROFILE HL546AF$'/
DATA LPRO(38)/'TSS SIM DATA PROFILE GEM1$'/
DATA LPRO(39)/'TSS SIM DATA PROFILE GEM2$'/
DATA LPRO(40)/'TSS SIM DATA PROFILE GEM3$'/
DATA LPRO(41)/'TSS SIM DATA PROFILE SAT1$'/
DATA LPRO(42)/'TSS SIM DATA PROFILE SAT2$'/
DATA LPRO(43)/'TSS SIM DATA PROFILE SAT3$'/
DATA LPRO(44)/'TSS SIM DATA PROFILE SAT4$'/
DATA LPRO(45)/'TSS SIM DATA PROFILE SAT6$'/
DATA LPRO(46)/'TSS SIM DATA PROFILE SAT8$'/
DATA LPRO(47)/'TSS SIM DATA PROFILE BAL1$'/
DATA LPRO(48)/'TSS SIM DATA PROFILE BAL2$'/
DATA LPRO(49)/'TSS SIM DATA PROFILE BAL5$'/
DATA LPRO(50)/'TSS SIM DATA PROFILE BAL6$'/
DATA LPRO(51)/'TSS SIM DATA PROFILE BAL7$'/
DATA LPRO(52)/' SIM DATA PROFILE HL546AG$'/
DATA LPRO(53)/' SIM DATA PROFILE HL246AF$'/
DATA LPRO(54)/' SIM DATA PROFILE HL446AE$'/
DATA LPRO(55)/' SIM DATA PROFILE HL146AE$'/
DATA LPRO(56)/' SIM DATA PROFILE HL346AF$'/
DATA LPRO(57)/' SIM DATA PROFILE HJ146AE$'/
DATA FPRO(1)/'HL146AB.XXX'/
DATA FPRO(2)/'HL246AB.XXX'/
DATA FPRO(3)/'HJ146AB.XXX'/
DATA FPRO(4)/'HEL30AB.XXX'/
DATA FPRO(5)/'H30SKAB.XXX'/
DATA FPRO(6)/'H30SKAC.XXX'/
DATA FPRO(7)/'HEL30AC.XXX'/
DATA FPRO(8)/'HEL30AD.XXX'/
DATA FPRO(9)/'HL246AC.XXX'/
```

```
      DATA FPRO(10)/'HL346AB.XXX'/
      DATA FPRO(11)/'HL446AB.XXX'/
      DATA FPRO(12)/'HL546AB.XXX'/
      DATA FPRO(13)/'HL546AC.XXX'/
      DATA FPRO(14)/'HL246AD.XXX'/
      DATA FPRO(15)/'HL446AC.XXX'/
      DATA FPRO(16)/'HL146AC.XXX'/
      DATA FPRO(17)/'HL346AD.XXX'/
      DATA FPRO(18)/'HJ146AC.XXX'/
      DATA FPRO(19)/'HEL30AE.XXX'/
      DATA FPRO(20)/'HEL30AF.XXX'/
      DATA FPRO(21)/'H30SKAD.XXX'/
      DATA FPRO(22)/'H30SKAE.XXX'/
      DATA FPRO(23)/'H30SKAF.XXX'/
      DATA FPRO(24)/'HEL30AG.XXX'/
      DATA FPRO(25)/'HEL30AH.XXX'/
      DATA FPRO(26)/'H30SKAG.XXX'/
      DATA FPRO(27)/'H30SKAH.XXX'/
      DATA FPRO(28)/'H30SKAI.XXX'/
      DATA FPRO(29)/'HEL30AI.XXX'/
      DATA FPRO(30)/'HEL30AJ.XXX'/
      DATA FPRO(31)/'HL546AE.XXX'/
      DATA FPRO(32)/'HL246AE.XXX'/
      DATA FPRO(33)/'HL446AD.XXX'/
      DATA FPRO(34)/'HL146AD.XXX'/
      DATA FPRO(35)/'HL346AE.XXX'/
      DATA FPRO(36)/'HJ146AD.XXX'/
      DATA FPRO(37)/'HL546AF.XXX'/
      DATA FPRO(38)/'GEM1.XXX'/
      DATA FPRO(39)/'GEM2.XXX'/
      DATA FPRO(40)/'GEM3.XXX'/
      DATA FPRO(41)/'SAT1.XXX'/
      DATA FPRO(42)/'SAT2.XXX'/
      DATA FPRO(43)/'SAT3.XXX'/
      DATA FPRO(44)/'SAT4.XXX'/
      DATA FPRO(45)/'SAT6.XXX'/
      DATA FPRO(46)/'SAT8.XXX'/
      DATA FPRO(47)/'BAL1.XXX'/
      DATA FPRO(48)/'BAL2.XXX'/
      DATA FPRO(49)/'BAL5.XXX'/
      DATA FPRO(50)/'BAL6.XXX'/
      DATA FPRO(51)/'BAL7.XXX'/
      DATA FPRO(52)/'HL546AG.XXX'/
      DATA FPRO(53)/'HL246AF.XXX'/
      DATA FPRO(54)/'HL446AE.XXX'/
      DATA FPRO(55)/'HL146AE.XXX'/
      DATA FPRO(56)/'HL346AF.XXX'/
      DATA FPRO(57)/'HJ146AE.XXX'/
C
C ******************************************************************************
C
C     SIMULATION FILE MODIFICATION
C
      A23=24.5
      TS=0.051
      WRITE (6,*) '  INPUT RCS IN SQUARE METERS   '
      READ (5,*)RCSM
      SRCS=RCSM*3.28*3.28
      SRCS=SQRT(SRCS)
      ITARG=0
C
      WRITE (6,*)'1 : TEK'
      WRITE (6,*)'2 : VT125'
      WRITE (6,*)'3 : VT240'
```

```
        WRITE (6,*)'4 : PC'
        READ (5,*)ITERM
C
        WRITE (6,*)'ENTER :  1 IF YOU ARE PROCESSING TMR DATA'
        WRITE (6,*)'         2 IF YOU ARE PROCESSING CINE DATA'
        WRITE (6,*)'         3 IF YOU ARE PROCESSING BEST DATA'
        READ (5,*)IREF
C
        WRITE(6,*)'ENTER TIME INTERVAL  ( 0,0 FOR THE WHOLE INTERVAL )'
        READ(5,*)STIME,STTIME
        IF (STTIME.EQ.0)STTIME=999
C
        WRITE (6,*)'DO YOU WANT TO FILTER THE DATA ? (Y/N)'
        READ (5,2322)ANS
 2322   FORMAT(A)
        WRITE(6,*)'PROFILE NUMBER   PROFILE'
        DO L=1,19
        WRITE(6,200)L,LPRO(L)
200     FORMAT(7X,I2,9X,A32)
        ENDDO
        WRITE (6,*)'ENTER C TO CONTINUE, Q TO QUIT :'
        READ (5,101) REPLY
101     FORMAT (A)
        IF (REPLY.EQ.'C') THEN
          DO L=20,38
            WRITE(6,200)L,LPRO(L)
          ENDDO
          WRITE (6,*)'ENTER C TO CONTINUE, Q TO QUIT :'
          READ (5,101) REPLY
          IF (REPLY.EQ.'C') THEN
            DO L=39,57
              WRITE(6,200)L,LPRO(L)
            ENDDO
          ENDIF
        ENDIF
        WRITE(6,*)'INPUT PROFILE NUMBER'
        READ(5,*)ITAPE
        UNIT7=FPRO(ITAPE)
        CALL FIXIT(ITILT,LPRO(ITAPE))
        IF (ITAPE.LT.38.AND.ITAPE.GT.51)GO TO 39
        IF (ITAPE.GE.38.AND.ITAPE.LE.51)GO TO 49
C
 39     IF (IREF.EQ.1) THEN
          UNIT7(9:11)='JST'
        ELSE IF (IREF.EQ.2) THEN
          UNIT7(9:11)='JSC'
        ELSE
          UNIT7(9:11)='BST'
        ENDIF
        GO TO 59
 49     IF (IREF.EQ.1) THEN
          UNIT7(6:8)='JST'
        ELSE IF (IREF.EQ.2) THEN
          UNIT7(6:8)='JSC'
        ELSE
          UNIT7(6:8)='BST'
        ENDIF
 59     OPEN(UNIT=4,FORM='UNFORMATTED',STATUS='OLD',
     2       FILE=UNIT7)
C
        TOUT=0.
        THAZL1=30.
        THEL1=30.
        THAZU1=0.
```

```
      DLP(1)=-0.2347
      DLP(2)=0.05
      DLP(3)=-9.748
      DEL(1)=-0.192738
      DEL(2)=-0.055573
      DEL(3)=-3.299135
      DUE(1)=0.88
      DUE(2)=0.55
      DUE(3)=-0.39988
      DSU(1)=1.67
      DSU(2)=0.73
      DSU(3)=-5.46
C     WRITE(6,*)' INPUT 1 FOR SCREEN OUTPUT'
C     READ(5,*)TOUT
      J=0
C READ START TIME
      READ(4)TBIAS,GMTIME,XMO,XDAY,XYR
      ILOOP=1
1     CONTINUE
      READ(4,END=99)T,SSRNG,SSRDOT,SSRANG,SSPANG,SSRRTE,SSPRTE
     1 ,X,Y,Z,VX,VY,VZ,AX,AY,AZ,IS1,IS2,RSS,RFPWR,AERR,BERR,ALFX,
     1 BETY,SCRR,SCPR
      IF (T.LT.STIME) GOTO 1
      IJJ=2**13
      ITF=IAND(IS2,IJJ)
      IF (ITF.NE.IJJ.AND.ANS.EQ.'Y') GO TO 1
      CALL RPAB(SSRANG,SSPANG,SSALP,SSBET)
      CALL TMR2KU
      DO I=1,3
        RNEW(I)=RO(I)
        VNEW(I)=VO(I)
      END DO
      IF(ILOOP.NE.1) GO TO 7
  6   CALL EXEC
      IF(MPRF.EQ.1) THEN
          TS=.051
      ELSE
          TS=.119
      END IF
      IF(ILOOP.EQ.1)THEN
         T1=T
         ILOOP=0
         GO TO 196
      END IF
  7   CONTINUE
      T1=T1+TS
      IF(T1.GT.T)THEN
         T1=T1-TS
         GO TO 196
      END IF
      DO I=1,3
         RO(I)=(RNEW(I)-ROLD(I))*(T1-T2)/(T-T2)+ROLD(I)
         VO(I)=(VNEW(I)-VOLD(I))*(T1-T2)/(T-T2)+VOLD(I)
      END DO
      GO TO 6
  196 CONTINUE
      T2=T
      DO I=1,3
         ROLD(I)=RNEW(I)
         VOLD(I)=VNEW(I)
      END DO
      HRRTE=HRRTE*180./(3.14159*1000.)
      HPRTE=HPRTE*180./(3.14159*1000.)
      J=J+1
```

```
        IF(J.EQ.2001)GO TO 99
        IF(T.GE.STTIME)GO TO 99
        TP(J)=T
        D(J,1)=SSRNG
        D(J,2)=SSRDOT
        D(J,3)=SSRANG
        D(J,4)=SSPANG
        D(J,5)=SSRRTE
        D(J,6)=SSPRTE
        D(J,7)=SSALP
        D(J,8)=SSBET
        D(J,9)=HRNG
        D(J,10)=HRDOT
        D(J,11)=RO(1)
        D(J,12)=RO(2)
        D(J,13)=-RO(3)
        D(J,14)=ATAND(-RO(3)/SQRT(RO(1)*RO(1)+RO(2)*RO(2)))
        D(J,15)=SSRNG-R
        D(J,16)=SSRDOT-ARDOT
        D(J,17)=SSRANG-SRANG
        D(J,18)=SSPANG-SPANG
        D(J,19)=SSRRTE-SRRTE
        D(J,20)=SSPRTE-SPRTE
        D(J,21)=SSALP-SALF
        D(J,22)=SSBET-SBTA
        D(J,23)=SAZRTE
        D(J,24)=SELRTE
        D(J,25)=RSS
        D(J,26)=RFPWR
        D(J,27)=AERR
        D(J,28)=BERR
        D(J,29)=ALFX
        D(J,30)=BETY
        D(J,31)=SCRR
        D(J,32)=SCPR
        IF (HRSS.LE.0) THEN
          D(J,33)=0
        ELSE
          D(J,33)=(32*HRSS)-181.+(40*ALOG10(HRNG))
        ENDIF
        D(J,34)=RACCEL
        D(J,35)=HRNG-R
        D(J,36)=HRDOT-ARDOT
        D(J,37)=HRANG-SRANG
        D(J,38)=HPANG-SPANG
        D(J,39)=HRRTE-SRRTE
        D(J,40)=HPRTE-SPRTE
        D(J,41)=HALP-SALF
        D(J,42)=HBET-SBTA
        D(J,43)=HRSS/32.
        IF(J.GT.2000)THEN
        WRITE(6,*)' MORE THAN 2000 POINTS'
        STOP
        ENDIF
        GO TO 1
99      CONTINUE
        J=J-1
        IXD=0
94      CONTINUE
        CALL SORT(TP,D,J,ITILT,IXD,IYD,GMTIME,IREF)
        GO TO 94
        END
C ****************************************************************************
        SUBROUTINE SORT(T,D,J,ITILT,IXD,IYD,GMTIME,IREF)
```

```
      DIMENSION D(2001,43),X(2001),Y(2001),T(2001)
      CHARACTER*40 IXT,IYT(43),PRONAME
      CHARACTER*4 REFF
      DIMENSION ITILT(10),IXL(10),IYL(10)
      DATA IXT/'TIME  SECONDS$'/
      DATA IYT(1)/'KU MDM RANGE  FEET$'/
      DATA IYT(2)/'KU MDM RANGE RATE  FT/SEC$'/
      DATA IYT(3)/'KU MDM ROLL ANGLE  DEG$'/
      DATA IYT(4)/'KU MDM PITCH ANGLE  DEG$'/
      DATA IYT(5)/'KU MDM ROLL RATE  DEG/SEC$'/
      DATA IYT(6)/'KU MDM PITCH RATE  DEG/SEC$'/
      DATA IYT(7)/'KU MDM ALPHA  DEG$'/
      DATA IYT(8)/'KU MDM BETA  DEG$'/
      DATA IYT(9)/'SIM RANGE FEET$'/
      DATA IYT(10)/'SIM RANGE RATE FT/SEC$'/
      DATA IYT(11)/'WSMR X (NORTH)  FEET$'/
      DATA IYT(12)/'WSMR Y (EAST)  FEET$'/
      DATA IYT(13)/'WSMR  -Z (ALTITUDE)  FEET$'/
      DATA IYT(14)/'WSMR ELEVATION ANGLE DEG$'/
      DATA IYT(15)/'DELTA RANGE  FEET ( KU - WSMR )$'/
      DATA IYT(16)/'DELTA RANGE RATE  FT/SEC ( KU - WSMR )$'/
      DATA IYT(17)/'DELTA ROLL ANGLE  DEG ( KU -  WSMR )$'/
      DATA IYT(18)/'DELTA PITCH ANGLE  DEG ( KU -  WSMR )$'/
      DATA IYT(19)/'DELTA ROLL RATE  DEG/SEC ( KU -  WSMR)$'/
      DATA IYT(20)/'DELTA PITCH RATE  DEG/SEC ( KU -  WSMR )$'/
      DATA IYT(21)/'DELTA ALPHA  DEG ( KU - WSMR )$'/
      DATA IYT(22)/'DELTA BETA  DEG ( KU -  WSMR )$'/
      DATA IYT(23)/'WSMR AZ RATE  DEG/SEC$'/
      DATA IYT(24)/'WSMR EL RATE  DEG/SEC$'/
      DATA IYT(25)/'KU SCANNER RSS ( VOLTS )$'/
      DATA IYT(26)/'KU SCANNER RF POWER ( VOLTS )$'/
      DATA IYT(27)/'KU SCANNER ALPHA ERROR ( VOLTS )$'/
      DATA IYT(28)/'KU SCANNER BETA ERROR ( VOLTS )$'/
      DATA IYT(29)/'KU SCANNER ALPHA X ( VOLTS )$'/
      DATA IYT(30)/'KU SCANNER BETA Y ( VOLTS )$'/
      DATA IYT(31)/'KU SCANNER ROLL RATE ( VOLTS )$'/
      DATA IYT(32)/'KU SCANNER PITCH RATE ( VOLTS )$'/
      DATA IYT(33)/'SIM RADAR CROSS SECTION ( DBSM )$'/
      DATA IYT(34)/'WSMR RANGE ACCELERATION  FT/SEC/SEC$'/
      DATA IYT(35)/'DELTA RANGE FEET (SIM-WSMR)$'/
      DATA IYT(36)/'DELTA RANGE RATE FT/SEC (SIM-WSMR)$'/
      DATA IYT(37)/'DELTA ROLL ANGLE DEG (SIM-WSMR)$'/
      DATA IYT(38)/'DELTA PITCH ANGLE DEG (SIM-WSMR)$'/
      DATA IYT(39)/'DELTA ROLL RATE DEG/SEC (SIM-WSMR)$'/
      DATA IYT(40)/'DELTA PITCH RATE DEG/SEC (SIM-WSMR)$'/
      DATA IYT(41)/'DELTA ALPHA DEG (SIM-WSMR)$'/
      DATA IYT(42)/'DELTA BETA DEG (SIM-WSMR)$'/
      DATA IYT(43)/'SIM RADAR SIGNAL STRENGTH$'/
      IFLAG=1
      IF (IREF.EQ.1)THEN
        REFF=' TMR'
      ELSE IF (IREF.EQ.2) THEN
        REFF='CINE'
      ELSE
        REFF='BEST'
      ENDIF
      DO I=1,43
        L=INDEX(IYT(I),'WSMR')
        IF (L .GT. 0) THEN
          IYT(I)(L:L+3) = REFF
        ENDIF
      ENDDO
1     CONTINUE
      DO I=1,43
```

```
        WRITE(6,68)I,IYT(I)
68      FORMAT(1X,I4,10X,A40)
        ENDDO
        WRITE(6,*)'INPUT IXD,IYD  IXD=0 FOR TIME'
        IF (IFLAG.EQ.0)THEN
          IFLAG=1
          IXD=0
          IYD=1
          GO TO 731
        ENDIF
        READ(5,*)IXD,IYD
731     IF(IXD.EQ.0)THEN
        DO I=1,J
        X(I)=T(I)
        Y(I)=D(I,IYD)
        ENDDO
        CALL FIXIT(IXL,IXT)
        CALL FIXIT(IYL,IYT(IYD))
        ELSE
        DO I=1,J
        X(I)=D(I,IXD)
        Y(I)=D(I,IYD)
        ENDDO
        CALL FIXIT(IXL,IYT(IXD))
        CALL FIXIT(IYL,IYT(IYD))
        ENDIF
        CALL PLOTIT(ITILT,IXL,IYL,X,Y,J,GMTIME,IYD,IXD)
        GO TO 1
  2     CONTINUE
        RETURN
        END
C ******************************************************************************
        SUBROUTINE FIXIT(IOUT,IN)
        DIMENSION IOUT(10)
        CHARACTER*4 ITEMP(10)
        CHARACTER*40 IN
        ITEMP(1)=(IN(1:4))
        ITEMP(2)=(IN(5:8))
        ITEMP(3)=(IN(9:12))
        ITEMP(4)=(IN(13:16))
        ITEMP(5)=(IN(17:20))
        ITEMP(6)=(IN(21:24))
        ITEMP(7)=(IN(25:28))
        ITEMP(8)=(IN(29:32))
        ITEMP(9)=(IN(33:36))
        ITEMP(10)=(IN(37:40))
        ENCODE(40,999,IOUT)(ITEMP(I),I=1,10)
999     FORMAT(10A4)
        RETURN
        END
C ******************************************************************************
        SUBROUTINE PLOTIT(ITILT,IXL,IYL,X,Y,J,GMTIME,IYD,IXD)
        COMMON /TERM/ITERM,XMO,XDAY,XYR,TBIAS,XJMO,XJDAY,XJYR
         COMMON/TMR/A,B,C,D,E,F,G(3),AH(3),AI(3),AJ(3),THAZL1,THEL1,THAZU1
        DOUBLE PRECISION SIG,AVG
        BYTE CR(2)
        DIMENSION ITILT(8),IXL(8),IYL(8)
        DIMENSION X(1),Y(1),TINL(30)
        WRITE(6,*)'  1 FOR MEAN AND STANDARD DEVIATION OF Y'
        READ(5,*)ISTA
        NSC=0
        XMAX=X(1)
        XMIN=X(1)
        YMAX=Y(1)
```

```
      YMIN=Y(1)
      GMHOUR1=GMTIME/60./60.
      GMHOUR=INT(GMHOUR1)
      GMMIN1=(GMHOUR1-GMHOUR)*60.
      GMMIN=INT(GMMIN1)
      GMSEC=INT((GMMIN1-GMMIN)*60.)
      DO I=1,J
      IF(X(I).GT.XMAX) XMAX=X(I)
      IF(X(I).LT.XMIN) XMIN=X(I)
      IF(Y(I).GT.YMAX) YMAX=Y(I)
      IF(Y(I).LT.YMIN) YMIN=Y(I)
      END DO
      IF(XMAX.EQ.XMIN)XMAX=XMIN*1.1
      IF(YMAX.EQ.YMIN)YMAX=YMIN*1.1
      IF(YMAX.EQ.YMIN)YMAX=0.1
2     CONTINUE
      YMAX1=YMAX
      YMIN1=YMIN
      IF (ITERM.EQ.1) CALL TEKALL(4114,480,0,1,0)
      IF (ITERM.EQ.2) CALL REGIS (1,0)
      IF (ITERM.EQ.3) CALL PVT240
      IF (IYD.EQ.1)CALL RINTL(X,Y,J,TINL,NTINL)
      CALL BGNPL(-1)
      CALL FLATBD
      CALL PAGE(14.,20.)
      CALL AREA2D (9.0,14.0)
      CALL HEIGHT(.45)
C     CALL TITLE(ITILT,100,IXL,100,IYL,100,9.0,13.5)
      CALL MESSAG(ITILT,100,-0.6,16.5)
      CALL RESET ('HEIGHT')
      CALL HEIGHT (.3)
      I100=100
C     0.6 WAS SUBTRACTED TO CENTER AND 1 INCHE WERE ADDED IN HEIGHT
      CALL MESSAG('TEST DATE$',I100,0.7,15.5)
      IF (XMO.GE.10)THEN
        CALL REALNO(XMO,0,3.0,15.5)
      ELSE
        CALL REALNO(XMO,0,3.3,15.5)
      ENDIF
      CALL REALNO(XDAY,0,3.9,15.5)
      IF (XDAY.GE.10) THEN
        CALL REALNO(XYR,0,4.8,15.5)
      ELSE
        CALL REALNO(XYR,0,4.5,15.5)
      ENDIF
      CALL MESSAG(' REVISION 12$',I100,6.0,15.5)
C     POSITION CHANGED FROM 13.7 TO 14.2
C     X-POSITION MOVED FORWARD BY 1.2
      CALL MESSAG('T0=          GMT=$',I100,1.2,14.2)
      CALL REALNO(GMTIME,0,1.8,14.2)
      CALL REALNO(GMHOUR,0,5.1,14.2)
      CALL REALNO(GMMIN,0,6.0,14.2)
      CALL REALNO(GMSEC,0,6.9,14.2)
      IF(ISTA.EQ.1)THEN
      AVG=0
      SIG=0
      DO I=1,J
      AVG=AVG+Y(I)
      SIG=SIG+Y(I)**2
      END DO
      AVG=AVG/J
      SIG=SQRT( SIG/J -AVG*AVG)
      CALL MESSAG('MEAN= $',I100,-0.9,-2.0)
      CALL REALNO(AVG,3,'ABUT','ABUT')
```

```
        CALL MESSAG('  STANDARD DEVIATION= $',I100,3.3,-2.0)
        CALL REALNO(SIG,3,'ABUT','ABUT')
        ENDIF
        CALL XNAME(IXL,100)
        CALL YNAME(IYL,100)
        CALL INTAXS
        CALL YAXANG(0.)
        IF(NSC.EQ.0)THEN
        CALL GRAF(XMIN,'SCALE',XMAX,YMIN,'SCALE',YMAX)
        ENDIF
        IF(NSC.EQ.1)THEN
        CALL GRAF(XMIN,'SCALE',XMAX,YMIN,'SCALE',YMAX)
        ENDIF
         IF (NTINL.NE.0.AND.IXD.EQ.0)THEN
          DO K=1,NTINL
               IVEC=1302
            CALL RLVEC (TINL(K),YMIN1,TINL(K),YMAX1,IVEC)
          ENDDO
         ENDIF
        CALL CURVE(X,Y,J,0)
        CALL GRID(1,1)
        CALL HEIGHT(.1)
        CALL RESET('HEIGHT')
        CALL DONEPL
        CR(1)=27
        CR(2)=12
        WRITE(6,888)CR
888     FORMAT('+',2A1)
        WRITE(6,*)'  INPUT 1 TO CHANGE SCALE OF Y AXIS'
        READ(5,*)NSC
        IF(NSC.EQ.1)THEN
        WRITE(6,*)'YMAX=',YMAX,' YMIN=',YMIN
        WRITE(6,*)' NEW YMAX'
        READ(5,*)YMAX
        WRITE(6,*)'NEW YMIN'
        READ(5,*)YMIN
        GO TO 2
        ENDIF
        RETURN
        END
C **********************************************************************
        SUBROUTINE RPAB(ROLLQ,PITCHQ,ALPHA,BETA)
       DEGRAD=57.29576
       PSI=67./DEGRAD
       PIT=PITCHQ/DEGRAD
       ROL=ROLLQ/DEGRAD
       XB=SIN(PIT)
       YB=-(SIN(ROL))*SQRT(1.0-XB*XB)
       Z=SQRT(1.0-XB*XB-YB*YB)
       IF(ROLLQ.LE.90.0.AND.ROLLQ.GE.-90.0)Z=-Z
       XR=XB*COS(PSI)+YB*SIN(PSI)
       YR=YB*COS(PSI)-XB*SIN(PSI)
       YRZR=SQRT(YR*YR+Z*Z)
       ALF=ASIN(YR/YRZR)
       BTA=ASIN(-XR/SQRT(XR*XR+YR*YR+Z*Z))
       ALPHA=ALF*DEGRAD
       BETA=BTA*DEGRAD
       IF(Z.GE.0.0.AND.YR.LE.0.0)ALPHA=-(180.0+ALPHA)
       IF(Z.GE.0.0.AND.YR.GT.0.0)ALPHA=(180.0-ALPHA)
        RETURN
       END
C ****************************************************************************
        SUBROUTINE RINTL(T,R,N,TI,J)
        DIMENSION RI(5),R(1),DS(5),TI(30),T(1)
```

```
      DATA RI/2550.,5750.,11510.,23030.,43510./
      RMAX=R(1)
      RMIN=R(1)
      DO 1 I=1,N
      RMAX=AMAX1(RMAX,R(I))
      RMIN=AMIN1(RMIN,R(I))
1     CONTINUE
      MRMAX=1
      MRMIN=1
      DO 2 I=1,5
      IF(RMAX.GT.RI(I))MRMAX=I
      IF(RMIN.GT.RI(I))MRMIN=I
2     CONTINUE
      J=0
      IF(MRMAX.EQ.MRMIN)RETURN
      J=0
      DO 3 L=1,5
      DS(L)=R(1)-RI(L)
3     CONTINUE
      DO 4 I=1,N
      DO 5 L=1,5
      IF( (R(I)-RI(L)) * DS(L) .LT. 0 )THEN
      J=J+1
      TI(J)=T(I)
      DS(L)=R(I)-RI(L)
      ENDIF
5     CONTINUE
4     CONTINUE
      RETURN
      END
C *************************************************************************
C **  MODED JWG 2/8/85
C **
C **  INPUT VIA COMMON VIA X,Y,Z,VX,VY,VZ,AX,AY,AZ
C **  OUTPUT VIA COMMON /ACTDAT/
C **
C *** WHITE SANDS TO KU-BAND RADAR PARAMETER CONVERSION ***
C
C
C                  ******* COMMENTARY *******
C
C
C                         ** PURPOSE **
C THIS SOFTWARE TAKES THE POSITION AND VELOCITY OF A TARGET REFERENCED
C TO THE PEARL SITE SURVEY CAP AND CALCULATES THE VALUES OF THE KU-BAND
C RADAR PARAMETERS AS SEEN AT THE KU-BAND RADAR GIMBAL AXES INTERSECTION.
C THESE CALCULATIONS INVOLVE COORDINATE ROTATIONS THROUGH A THREE-AXIS
C POSITIONER AND FOUR TRANSLATIONS FROM THE PEARL CAP TO THE RADAR GIMBAL
C AXES INTERSECTION.
C
C THESE CALCULATIONS ARE TO BE DONE BY WSMR DATA REDUCTION USING THE WSMR
C RANGE REFERENCE ESTIMATIONS OF TARGET LOCATION WITH TIME.  COMPARISON
C CAN BE MADE DIRECTLY WITH THE KU-BAND OUTPUTS FOR THE SAME TIME VALUES.
C
C                    ** INPUTS & CONSTANTS **
C
C  WSMR PROVIDED INPUTS:
C   WSMR WILL PROVIDE TARGET POSITION - X, Y, Z - AND VELOCITY - VX, VY,
C   VZ AS INPUTS TO THIS PROGRAM.
C   UNITS ARE FEET AND FEET/SECOND.
C   THE COORDINATE SYSTEM IS:
C       ORIGIN = PEARL SURVEY CAP
C       X-AXIS IS POSITIVE TOWARD THE NORTH
C       Y-AXIS IS POSITIVE TOWARD THE EAST
```

```
C        NEGATIVE Z-AXIS IS UPWARD ALONG THE LOCAL VERTICAL.
C
C   CONSTANTS PROVIDED BY SIMULATION TEST TAPE:
C    FOR ANY GIVEN TEST THE FOLLOWING PARAMETERS WILL BE DEFINED ON THE
C    SIMULATION MAGNETIC DATA TAPE AND WILL REMAIN CONSTANT FOR THAT TEST:
C     DSU(I) I=1,3  IS THE LOCATION OF THE KU-BAND RADAR GIMBAL AXES IN
C                   UPPER AZIMUTH COORDINATES.
C     THAZL1        IS THE LOWER AZIMUTH AXIS ROTATION ANGLE IN DEGREES.
C     THEL1         IS THE ELEVATION AXIS ROTATION ANGLE IN DEGREES.
C     THAZU1        IS THE UPPER AZIMUTH AXIS ROTATION ANGLE IN DEGREES.
C
C   ONE TIME INPUT CONSTANTS:
C    THE FOLLOWING PARAMETERS WILL BE MEASURED AFTER INSTALLATION OF THE
C    ANTENNA PEDESTAL AT THE PEARL SITE.  THEIR VALUES SHOULD NOT CHANGE.
C    THEY ARE CURRENTLY DEFINED AS ZERO IN THIS SOFTWARE.
C      DLP(I) I=1,3  LOCATION OF THE LOWER AZIMUTH ORIGIN IN PEARL
C                    COORDINATES.
C      DEL(I) I=1,3  LOCATION OF THE ELEVATION ORIGIN IN LOWER AZIMUTH
C                    COORDINATES.
C      DUE(I) I=1,3  LOCATION OF THE UPPER AZIMUTH ORIGIN IN ELEVATION
C                    COORDINATES.
C
C
C                 ** SOFTWARE OUTPUTS **
C   THIS SOFTWARE PRODUCES THE FOLLOWING OUTPUTS REFERENCED TO THE
C   RADAR GIMBAL AXES INTERSECTION.
C
C    R = RANGE (FT)
C    ARDOT = RANGE RATE (FT/SEC)
C    SRANG = ROLL ANGLE (DEG)
C    SPANG = PITCH ANGLE (DEG)
C    SRRTE = INERTIAL ROLL RATE (DEG/SEC)
C    SPRTE = INERTIAL PITCH RATE (DEG/SEC)
C    SALP = ALPHA ANGLE (DEG)
C    SBTA = BETA ANGLE (DEG)
C    AZRTE = AZIMUTH ANGLE RATE (DEG/SEC)
C    ELRTE = ELEVATION ANGLE RATE (DEG/SEC)
C
C                 ** EXAMPLE **
C   AN EXAMPLE CASE IS INCLUDED IN THE CODE.  IF THIS SOURCE IS COMPILED,
C   LINKED, AND EXECUTED, OUTPUTS WILL GO TO UNIT 6.  THEIR VALUES SHOULD
C   BE:
C     R = 43760.6016                ARDOT = -9.87364578
C     SRANG = 25.2644920            SPANG = 28.2407990
C     SRRTE = -.926818550E-01       SPRTE = .688237743E-02
C     SALF = -36.1578255            SBTA = 9.27430439
C     AZRTE = .302744657E-01        ELRTE = -.105446391
C
C
C
        SUBROUTINE TMR2KU
        COMMON /TMR/X,Y,Z,VX,VY,VZ,
     1            DLP(3),DEL(3),DUE(3),
     2            DSU(3),THAZL1,THEL1,THAZU1,A23
       COMMON /INPUT/RO(3),VO(3),EWB(3)
       COMMON /ACTDAT/R,ARDOT,SPANG,SRANG,SPRTE,SRRTE,AL,BT,SALF,SBTA,
     1ER(3),EV(3),ERTO(3),AZRATE,ELRATE,AZRTE,ELRTE
     2,AX,AY,AZ,AAX,AAY,AAZ,RACCEL
C      DIMENSION DLP(3),DEL(3),DUE(3),DSU(3)
       DIMENSION AZL(3,3),ELV(3,3),AZU(3,3)
       DIMENSION DPT(3),DLT(3),DET(3),DUT(3),DST(3)
       DIMENSION DLAZ(3),DELV(3),DAZU(3)
       DIMENSION VPT(3),VLAZ(3),VELV(3),VST(3)
       DIMENSION APT(3),ALAZ(3),AELV(3),AST(3)
```

```
      DATA DEGRAD/57.275/,PI/3.14159/
C  THE EWB PARAMETERS ARE ALWAYS DEFINED AS 0.0
      EWB(1)=0.0
      EWB(2)=0.0
      EWB(3)=0.0
C EXAMPLE CASE VALUES:
C      X=39417.2812
C      Y=16164.0078
C      Z=-9999.65820
C      VX=-41.1736259
C      VY=73.6755753
C      VZ=.166666671E-02
C      THAZL2=45.0
C      THEL2=-45.0
C      THAZU2=0.0
C
C
C  ** INPUTS **
C  WSMR WILL NORMALLY PROVIDE X,Y,Z,VX,VY,VZ.  REF IS PEARL SURVEY POINT.
C THIS IS PROVIDED VIA COMMON TMR BLOCK
      DPT(1)=X
      DPT(2)=Y
      DPT(3)=Z
      VPT(1)=VX
      VPT(2)=VY
      VPT(3)=VZ
      APT(1)=AX
      APT(2)=AY
      APT(3)=AZ
C
C
C  ** CONSTANTS **
C DLP(I); DEL(I); AND DUE(I) WILL BE PROVIDED ONE TIME AFTER INSTALLATION
C OF THE ANTENNA PEDESTAL
C THIS IS PROVIDED VIA COMMON TMR BLOCK
C      DLP(1)=0.0
C      DLP(2)=0.0
C      DLP(3)=0.0
C      DEL(1)=0.0
C      DEL(2)=0.0
C      DEL(3)=0.0
C      DUE(1)=0.0
C      DUE(2)=0.0
C      DUE(3)=0.0
C
C
C ** CONSTANTS FROM SIMULATION DATA TAPE **
C THIS IS PROVIDED VIA COMMON TMR BLOCK
C      DSU(1)=0.0
C      DSU(2)=0.0
C      DSU(3)=0.0
C      THAZL1=0.0
C      THEL1=0.0
C      THAZU1=0.0
C
C EXAMPLE ANGLE VALUES ARE EQUATED HERE.
C      THAZL1=THAZL2
C      THEL1=THEL2
C      THAZU1=THAZU2
C  CONVERT TO RADIANS
      THAZL=THAZL1/DEGRAD
      THEL=THEL1/DEGRAD
      THAZU=THAZU1/DEGRAD
C SET UP THE ROTATIONAL MATRICES
```

```
      CALL AZGEN(AZL,THAZL)
      CALL ELGEN(ELV,THEL)
      CALL AZGEN(AZU,THAZU)
C CONVERT TARGET IN PEARL TO TARGET AT GIMBALS
      DO 11 I=1,3
11    DLT(I)=DPT(I)-DLP(I)
      CALL MULT31(AZL,DLT,DLAZ)
      DO 21 I=1,3
21    DET(I)=DLAZ(I)-DEL(I)
      CALL MULT31(ELV,DET,DELV)
      DO 31 I=1,3
31    DUT(I)=DELV(I)-DUE(I)
      CALL MULT31(AZU,DUT,DAZU)
      DO 41 I=1,3
41    DST(I)=DAZU(I)-DSU(I)
C THESE ARE THE THREE TARGET COORDINATES IN RADAR GIMBAL REFERENCE.
      RO(1)=DST(1)
      RO(2)=DST(2)
      RO(3)=DST(3)
C CONVERT TO VELOCITIES REFERENCED TO GIMBALS
      CALL MULT31(AZL,VPT,VLAZ)
      CALL MULT31(ELV,VLAZ,VELV)
      CALL MULT31(AZU,VELV,VST)
C CONVERT TO ACCELATIONS REFERENCED TO GIMBALS
      CALL MULT31(AZL,APT,ALAZ)
      CALL MULT31(ELV,ALAZ,AELV)
      CALL MULT31(AZU,AELV,AST)
C THESE ARE VELOCITIES IN GIMBAL REFERENCE.
      VO(1)=VST(1)
      VO(2)=VST(2)
      VO(3)=VST(3)
C
C       RO(I) VO(I) I=1,3 SHUTTLE BODY POS AND VEL VECTOR
C
C CALCULATE THE KU-BAND RADAR PARAMETERS BASED ON THE INPUTS.
       C23=COSD(A23)
       S23=SIND(A23)
       X1=RO(2)*C23-RO(3)*S23
       Y1=-RO(2)*S23-RO(3)*C23
       Z1=-RO(1)
       RO(1)=X1
       RO(2)=Y1
       RO(3)=Z1
       VX=VO(2)*C23-VO(3)*S23
       VY=-VO(2)*S23-VO(3)*C23
       VZ=-VO(1)
       VO(1)=VX
       VO(2)=VY
       VO(3)=VZ
       AAX=AST(2)*C23-AST(3)*S23
       AAY=-AST(2)*S23-AST(3)*C23
       AAZ=-AST(1)
      CALL ACT
      SRRTE=SRRTE*(DEGRAD/1000.)
      SPRTE=SPRTE*(DEGRAD/1000.)
      SALF=AL*DEGRAD
      SBTA=BT*DEGRAD
      AZRTE=AZRATE*DEGRAD
      ELRTE=ELRATE*DEGRAD
       RETURN
      END
C *******************************************************************************
      SUBROUTINE AZGEN(AZ,ANGAZ)
C THIS SUBROUTINE PRODUCES A 3X3 MATRIX, AZ, FOR
```

```
C AN AZIMUTH TABLE ROTATION OF ANGAZ RADIANS.
      DIMENSION AZ(3,3)
      DO 10 I=1,3
      DO 10 J=1,3
10    AZ(I,J)=0.0
      AZ(1,1)=COS(ANGAZ)
      AZ(1,2)=SIN(ANGAZ)
      AZ(2,1)=-SIN(ANGAZ)
      AZ(2,2)=COS(ANGAZ)
      AZ(3,3)=1.0
      RETURN
      END
C ***********************************************************************
      SUBROUTINE ELGEN(EL,ANGEL)
      DIMENSION EL(3,3)
      DO 10 I=1,3
      DO 10 J=1,3
10    EL(I,J)=0.0
      EL(1,1)=COS(ANGEL)
      EL(1,3)=-SIN(ANGEL)
      EL(2,2)=1.0
      EL(3,1)=SIN(ANGEL)
      EL(3,3)=COS(ANGEL)
      RETURN
      END
C ***********************************************************************
C   SUBROUTINE ACT
C
C
C  *********************************************************
C  * THIS SUBROUTINE INITIALIZES THE ANGLE TRACKING LOOPS, THE  *
C  * RANGE TRACKING LOOP, AND THE VELOCITY PROCESSOR -- STEADY *
C  * STATE CONDITIONS ARE ASSUMED.                            *
C  *********************************************************
C
C
C ***********************************************************************
      SUBROUTINE ACT
      COMMON /ACTDAT/R,ARDOT,SPANG,SRANG,SPRTE,SRRTE,AL,BT,SALF,SBTA
     2,ER(3),EV(3),ERTO(3),AZRATE,ELRATE,AZRTE,ELRTE
     3,AX,AY,AZ,AAX,AAY,AAZ,RACCEL
      COMMON /INPUT/ ERT(3),EVT(3),EWB(3),DUM(18)
      COMMON /SYSDAT/TSAM,DR(3),CP,SP,PSI,PSBIAS,DUM2(7),TRB(3,3)
      DIMENSION         FLTWID(3),RI(10)
      DIMENSION TX1(3,3),TX2(3,3),TX3(3,3),TBL(3,3)
      DATA PI/3.141592653/
      DATA IONE/0/
      IF(IONE.EQ.0)CALL DATA
      IONE=1
C
C  STEP 1-1: COMPUTE INITIAL INNER AND OUTER GIMBAL POSITIONS.
C    (NOTE: TRANSFORM CONSISTS OF TRANSLATION PLUS ROTATION.)
C    PERFORM TRANSLATION -- SHIFT TO RADAR FRAME ORIGIN.
      DO 1 I=1,3
    1 ERTO(I)=ERT(I)-DR(I)
C    TRANSFORM TARGET POSITION FROM BODY TO RADAR FRAME.
      CALL MULT31(TRB,ERTO,ER)
C    TRANSFORM TARGET VELOCITY FROM BODY TO RADAR FRAME.
      CALL MULT31(TRB,EVT,EV)
      SQ=SQRT(ER(2)*ER(2)+ER(3)*ER(3))
C    COMPUTE INNER(BETA) GIMBAL POSITION -- BT.
      IF(ER(1).EQ.0.0.AND.SQ.EQ.0.0) STOP
      BT=-ATAN2(ER(1),SQ)
      ER2=-ER(2)
      ER3=-ER(3)
C    COMPUTE OUTER(ALPHA) GIMBAL POSITION --- AL.
      IF(ER2.EQ.0.0.AND.ER3.EQ.0.0) GO TO 8
      AL=-ATAN2(ER2,ER3)
      GO TO 9
    8 IF(ER(1).GT.0.0) AL=PI/2.
      IF(ER(1).LT.0.0) AL=-PI/2.
      IF(ER(1).EQ.0.0) STOP
C
C  STEP 1-2: COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH AND
C            ELEVATION RATES.
C    PRELIMINARY TRIGONOMETRIC COMPUTATIONS.
    9 CA=COS(AL)
      SA=SIN(AL)
      CB=COS(BT)
      SB=SIN(BT)
C    TRANSFORM BODY ANGULAR VELOCITY VECTOR FROM BODY TO OUTER
C    GIMBAL(G) REFERENCE FRAME.
      WGX=CP*EWB(1)+SP*EWB(2)
      WGY=CA*(-SP*EWB(1)+CP*EWB(2))+SA*EWB(3)
      WGZ=-SA*(-SP*EWB(1)+CP*EWB(2))+CA*EWB(3)
C    COMPUTE THE RANGE TO TARGET.
      R=SQRT(ER(1)*ER(1)+ER(2)*ER(2)+ER(3)*ER(3))
      YZR=SQRT(ER(2)*ER(2)+ER(3)*ER(3))
C    COMPUTE RANGE RATE TO TARGET
      ARDOT=(ER(1)*EV(1)+ER(2)*EV(2)+ER(3)*EV(3))/R
C    COMPUTE RANGE ACCELERATION TO TARGET.
      VSQ=EV(1)**2+EV(2)**2+EV(2)**2
      RACCEL=(VSQ+ER(1)*AAX+ER(2)*AAY+ER(3)*AAZ-ARDOT**2)/R
C    COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH RATE(AZRATE).
      VGY=CA*EV(2)+SA*EV(3)
      AZRATE=VGY/R+(CB*WGX-SB*WGZ)
C    COMPUTE INITIAL TARGET INERTIAL LOS ELEVATION RATE(ELRATE).
      ELRATE=-(CB*EV(1)-SB*(-SA*EV(2)+CA*EV(3)))/R+WGY
C  ****************************************
C  * STEP 1: UPDATE ROUGH RANGE RATE ESTIMATE *
C  ****************************************
C
C  ****************************************************************
C  * STEP 1: UPDATE ANTENNA LOS-TO-BODY TRANSFORMATION (NOTE: TRANS- *
C  *         FORMATION INCLUDES GIMBAL BIAS ERRORS AND RADAR YAW     *
C  *         ANGLE ERROR WRT BODY FRAME).                            *
C  ****************************************************************
      CALL GAMMA(TX1,-(BT+BTBIAS))
      CALL THETA(TX2,-(AL+ALBIAS))
      CALL MULT33(TX2,TX1,TX3)
      CALL PHI(TX2,-PSI)
      CALL MULT33(TX2,TX3,TBL)
C
C  ***********************************************************
C  * STEP 6: TRANSFORM TARGET ANGLES AND INERTIAL ANGLE RATES TO *
C  *         BODY FRAME FOR USE IN DISPLAYS AND G AND N.         *
C  ***********************************************************
C  NOTE: TRANSFORMATION TBL INCLUDES GIMBAL BIAS ERRORS AND RADAR YAW
C        ANGLE ERROR WRT BODY FRAME.
C  UPDATE TARGET INERTIAL PITCH RATE IN ORBITER BODY COORDINATES
C  FOR DISPLAY.
      SPRTE=-1000.*(TBL(2,1)*AZRATE+TBL(2,2)*ELRATE)
C  UPDATE TARGET INERTIAL ROLL RATE IN ORBITER BODY COORDINATES
C  FOR DISPLAY.
      SRRTE=-1000.*(TBL(1,1)*AZRATE+TBL(1,2)*ELRATE)
C  UPDATE ANTENNA PITCH ANGLE IN ORBITER BODY COORDINATES FOR DISPLAY.
      SPANG=-ASIN(TBL(1,3))*57.29576
C  UPDATE ANTENNA IN ORBITER BODY COORDINATES FOR DISPLAY.
```

```
      IF(TBL(2,3).EQ.0.0.AND.TBL(3,3).EQ.0.0) GO TO 5                   00026290
      SRANG=-ATAN2(-TBL(2,3),TBL(3,3))*57.29576                          00026300
      GO TO 7                                                            00026310
    5 IF(TBL(1,3).GT.0.0) SRANG=-90.0                                    00026320
      IF(TBL(1,3).LT.0.0) SRANG=90.0                                     00026330
      IF(TBL(1,3).EQ.0.0) STOP                                           00026340
C RESOLVE POSSIBLE ANGLE AMBIGUITIES, VIZ., -90.<SPANG<90. AND           00026350
C -180.<SRANG<180.                                                       00026360
    7 IF(SPANG.LE.90.) GO TO 10                                          00026370
      SPANG=-(180.-ABS(SPANG))*(SPANG/ABS(SPANG))                        00026380
      SRANG=(180.-ABS(SRANG))*(SRANG/ABS(SRANG))                         00026390
   10 CONTINUE                                                           00026400
      RETURN                                                             00026510
      END                                                                00026520
C                                                                        00029600
C                                                                        00029610
C  *************************************************************         00029620
C  * THIS SUBROUTINE INITIALIZES ALL DATA REQUIRED BY THE SEARCH, *      00029630
C  * ACQUISITION, AND TRACK SUBPROGRAMS.                          *      00029640
C  *************************************************************         00029650
C                                                                        00029660
C                                                                        00029670
C                                                                        00029610
C  *************************************************************         00029620
C  * THIS SUBROUTINE INITIALIZES ALL DATA REQUIRED BY THE SEARCH, *      00029630
C  * ACQUISITION, AND TRACK SUBPROGRAMS.                          *      00029640
C  *************************************************************         00029650
C                                                                        00029660
C                                                                        00029670
      SUBROUTINE DATA                                                    00029680
      REAL IDUM1                                                         00029685
      COMMON /RTDAT/IDUM1(2),RBIAS,DUM1(9)                               00029690
      COMMON /SYSDAT/TSAM,DR(3),CP,SP,PSI,PSBIAS,ALBIAS,BTBIAS,GP,GA,    00029700
     2               TGTSIG,GPS,GAS,TRB(3,3)                             00029710
      COMMON /NOISE/NS1,NS2,NN(10),GAUSS(320)                            00029720
      DIMENSION A(3,3),B(3,3),C(3,3)                                     00029725
      REAL LT,KTS                                                        00029730
C  *******************                                                   00029740
C  * SYSTEM PARAMETERS *                                                 00029750
C  *******************                                                   00029760
      PI=3.1415926                                                       00029770
      PII=PI/180.                                                        00029780
C  RADAR FRAME YAW ANGLE IN BODY COORDINATES (DEGREES).                  00029790
      PSI=PII*67.0                                                       00029800
      CP=COS(PSI)                                                        00029820
      SP=SIN(PSI)                                                        00029830
C  RADAR LOCATION OFFSET FROM ORBITER C.G. IN BODY COORD. (FEET)         00029840
C **** VALUES MODIFIED MAR 24 83 PER FM8 MEMO ***********
      DR(1)=0.0
      DR(2)=11.130
      DR(3)=-5.79
C  RANGE BIAS ERROR IS COMPUTED IN SUBROUTINE RTRACK AS
C  FUNCTION OF RANGE
C  ALPHA GIMBAL BIAS.                                                    00029920
      ALBIAS=0.0                                                         00029930
C  BETA GIMBAL BIAS.                                                     00029940
      BTBIAS=0.0                                                         00029950
C                                                                        00029952
C  RADAR PLATFORM ORIENTATION ERRORS WITH RESPECT TO BODY FRAME.         00029954
C                                                                        00029956
C  YAW ANGLE ERROR.                                                      00029958
      PSBIAS=PII*0.0
C                                                                        00029962
C  ROLL ANGLE ERROR.                                                     00029964
```

```
      RLBIAS=PII*0.0
C  PITCH ANGLE ERROR.
      PTBIAS=PII*0.0
C
C NBIAS=0 FOR N0 BIAS AND RADAR AT ORGIN
C
       NBIAS=0
       IF(NBIAS.NE.0)GO TO 700
701     FORMAT(' ALL ANGLE  BIAS SET 10 ZERO RADAR AT ORGIN')
       DO 4 I=1,3
 4      DR(I)=0.0
C        PSI=0.0
        PSBIAS=0.0
        RLBIAS=0.0
        PTBIAS=0.0
 700        CONTINUE
C
C  COMPUTE MATRIX OF TRANSFORMATION FROM BODY FRAME TO RADAR FRAME.
      CALL PHI(B,PSI+PSBIAS)
      CALL THETA(A,RLBIAS)
      CALL MULT33(A,B,C)
      CALL GAMMA(A,PTBIAS)
      CALL MULT33(A,C,TRB)
C  *************************
C  * SYSTEM SAMPLE INTERVAL *
C  *************************
C
C  ***********************
C  * COMPUTE SNR CONSTANT *
C  ***********************
C  EQUIVALENT ONE-SIDED NOISE POWER SPECTRAL DENSITY (MW/KHZ)
      KTS=-137.5
      KTS=10.**(0.1*KTS)
C  SYSTEM LOSSES ON TRANSMIT (DB).
      LT=2.5
      LT=10.**(0.1*LT)
C  ONE-WAY ANTENNA GAIN (DB).
      G=37.7
      G=10.**(0.1*G)
      ALMBDA=0.070845
C  CONSTANT FOR PASSIVE TRACKING SNR COMPUTATION.
      GP=4.*(G**2)*(ALMBDA**2)/((4.*PI)**3*LT*KTS)
C  BEACON PARAMETER (DBM)
      BCN=44.0
      BCN=10.**(0.1*BCN)
C  CONSTANT FOR ACTIVE TRACKING SNR COMPUTATION.
      GA=4.*G*ALMBDA**2*BCN/((4.*PI)**2*KTS)
C  CONSTANT FOR PASSIVE MODE VIDEO SNR COMPUTATION (DB).
      GPS=183.9
C  CONSTANT FOR ACTIVE MODE VIDEO SNR COMPUTATION (DB).
      GAS=146.9
C
C  ********************************
C  * RANDOM NUMBER GENERATOR SEEDS *
C  ********************************
      NS1=48
      NS2=135
      NN(1)=0
C  INITIALIZE NOISE SEQUENCE.
      DO 2 I=1,320
    2 GAUSS(I)=ANORM(NS1,NS2)
     IF(ITEST.EQ.2)GO TO6341
     ITEST=2
C    WRITE(6,592)
```

```
 592  FORMAT(1H1,' RANDOM NUMBER INITIALIZATION')
C     WRITE(6,593)(GAUSS(I),I=1,320)
 593  FORMAT(8F8.4)
C     WRITE(6,592)
6341      CONTINUE
C                                                                       00030380
C   ***************************                                         00030390
C   * DEFINE TARGET PARAMETERS *                                        00030400
C   ***************************                                         00030410
C  TARGET SEARCH CROSS-SECTION ( FIXED TEMPORARILY).                    00030420
       TGTSIG=10.0                                                      00030430
       RETURN                                                           00030440
       END                                                              00030450
        SUBROUTINE SETIT
        COMMON /TARGET/ITARG,SRCS
        COMMON /LEN1/ANGOFF
       COMMON /SATDAT/RADAR(3),KTAR,R(70,3),SIG(70),ROLD,
     1 ICLOSE,ICOLD,JHOT(60)
       COMMON /CNTL/IPWR,IMODE,ITXP,IASM,ISRCHC,ISRCHG,IAZS,IELS,ISLR,
     2         EDRNG,EDPA,EDRA
       COMMON /ICNTL/IOLDPW,IOLDMD,IOLDSM,ISHOLD,KMSCLK,KWMUP,KSNCLK,
     2              KSNMAX,KACCLK,MTP,MZ1,MZ0,MSS,MTKINT,MRNG,MSAM,MPRF,
     3               MBKTRK,MBTSUM,MBT(8)
       COMMON /OUTPUT/MSWF,MTF,MSF,SRNG,SRDOT,SPANG,SRANG,SPRTE,
     2                SRRTE,SRSS,MADVF,MRDVF,MARDVF,MRRDVF
       COMMON /INPUT/ERTO(3),EVTO(3),EWB(3),TBT(3,3),TBTD(3,3)
       COMMON /ATDAT/DUM1(10),PREF,RREF
       COMMON /SYSDAT/TS,DUM2(14)
       COMMON /CGMAIN/RO(3),VO(3),AO(3)
       COMMON /DSCRM/DUM3(6),SIGBAR,SNRD,SIGDB
       COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD
C  ITARG = 0   POINT TARGET RCS OF POINT TARGET
C              SRCS IS VARIABLE NAME OF RCS VALUE
C              SRCS = 3.27 IS IMSQ TARGET.
C     SRCS=3.27
        DO I=1,3
        DO J=1,3
        EWB(I)=0.
        TBT(I,J)=0
        IF(I.EQ.J)TBT(I,J)=1.
        TBTD(I,J)=0.
        ENDDO
        ENDDO
        KOLD=-1
       CALL SYSINT
       IPWR=3
       IMODE=2
       IASM=1
       ITXP=1
       ISRCH=0
        IAZS=0
        IELS=0
        ISLR=0
        ISRCHG=0
       EDRNG=500.0
       EDRA=0.0
       EDPA=0.0
       PII=3.14159265/180.
       EDPA=EDPA*PII
       EDRA=EDRA*PII
       MTF=0
       MTP=1
       MTP=1
        RETURN
       END
C                                                                       00030820
C                                                                       00030830
C    ***************************************************************    00030840
C    * THIS FUNCTION GENERATES A RANDOM NUMBER FROM A GAUSSIAN PDF *    00030850
C    * WITH ZERO MEAN AND UNIT VARIANCE.                           *    00030860
C    ***************************************************************    00030870
C                                                                       00030880
C                                                                       00030890
      FUNCTION ANORM(K1,K2)                                             00030900
      Y1=RNDU(K1)                                                       00030910
      Y2=RNDU(K2)                                                       00030920
      TPI=6.2831852                                                     00030930
      ANORM=SQRT(-2.*ALOG(Y1))*COS(TPI*Y2)                              00030940
      RETURN                                                            00030950
      END                                                               00030960
C                                                                       00025230
C                                                                       00025240
C    *************************************************************      00025250
C    * THIS SUBROUTINE UPDATES AZ AND EL INERTIAL LOS RATES, THE *      00025260
C    * ALPHA AND BETA GIMBAL RATES, THE ALPHA AND BETA GIMBAL    *      00025270
C    * POSITIONS, AND THE TARGET PITCH AND ROLL ANGLES FOR THE   *      00025280
C    * DISPLAY.                                                  *      00025290
C    *************************************************************      00025300
C                                                                       00025310
C                                                                       00025320
      SUBROUTINE ATRACK                                                 00025330
      REAL INTT,K4,K5,K6                                                         00025335
      INTEGER AT1A(10,2),AT1E(10,2),AT2A(10,2),AT2E(10,2)
      COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)                          00025350
      COMMON /INPUT/DUM(6),EWB(3),DUM2(18)                              00025360
      COMMON /OUTPUT/I1DUM(3),D1DUM(2),SPANG,SRANG,SPRTE,SRRTE,SRSS,    00025370
     2              IDUM1(4),SSALP,SSBET                                            00025380
      COMMON /ICNTL/I2DUM(14),MRNG,MSAM,MPRF,IDUM2(11)                  00025390
      COMMON /SYSDAT/TSAM,DR(3),CP,SP,PSI,PSBIAS,ALBIAS,BTBIAS,         00025400
     2              DUM4(5)                                             00025410
      COMMON /ATDAT/CA,SA,CB,SB,AZRATE,ELRATE,ALRATE,BTRATE,AL,BT,      00025420
     2             DUM3(4)                                              00025430
      COMMON /DSCRM/AZDISC,ELDISC,DUM1(7)                               00025440
      DIMENSION TX1(3,3),TX2(3,3),TX3(3,3),TBL(3,3)                     00025450
     DIMENSION TDC(3)
C ********************************************************
C
C   ATRACK MODIFIED JAN 28 1986 BY M. MEYER
C   MODIFICATIONS TO SUBROUTINE ATRACK WERE IMPLEMENTED
C   TO UPDATE THE LOOP CONSTANTS AND MORE ACCURATELY
C   SIMULATE THE ACTUAL SIGNAL PROCESSING PERFORMED
C   BY THE RADAR
C
C *******************************************************
C
C-------- NEW LOOP CONSTANTS JAN 28 1986--------------------
C
      DATA AT1A/9*5,1,6*13,5,3*1/
      DATA AT1E/9*6,1,6*16,6,2*1,2/
      DATA AT2A/9*407,149,6*662,407,3*149/
      DATA AT2E/9*532,195,6*866,532,3*195/
      DATA k6/3.60E-5/,K4/.0048876/,K5/.236/,DTOR/.0174533/
C
     DATA TDC/0.05122118,0.1195161,0.2561557/
C   DEFINITION.  AT1=KEQ=(WN**2)/(4.*DIFFERENCE PATTERN SLOPE) WHERE    00025490
C                   WN IS NATURAL FREQUENCY OF THE LOOP.                00025500
C   DEFINITION:  AT2=KEQ*TAU WHERE TAU IS PROPORTIONAL TO STEP RESPONSE 00025510
C                   CONVERGENCE TIME.                                   00025520
```

```
C                                                                       00026700
      TCON=TSAM/TDC(MPRF)
C  ********************************************                         00026710
C  * STEP 1: UPDATE ROUGH RANGE RATE ESTIMATE *                         00026720
C  ********************************************                         00026730
C                                                                       00025530
C  ******************************************************************** 00025540
C  * STEP 1: UPDATE ANTENNA LOS-TO-BODY TRANSFORMATION (NOTE: TRANS-  * 00025550
C  *         FORMATION INCLUDES GIMBAL BIAS ERRORS AND RADAR YAW      * 00025560
C  *         ANGLE ERROR WRT BODY FRAME).                             * 00025570
C  ******************************************************************** 00025580
       CALL GAMMA(TX1,-(BT+BTBIAS))                                     00025590
       CALL THETA(TX2,-(AL+ALBIAS))                                     00025600
       CALL MULT33(TX2,TX1,TX3)                                         00025610
       CALL PHI(TX2,-PSI)                                               00025620
       CALL MULT33(TX2,TX3,TBL)                                         00025630
C                                                                       00025640
C  *******************************************************************  00025650
C  * STEP 2: UPDATE ESTIMATED TARGET INERTIAL AZIMUTH AND ELEVATION *   00025660
C  *         RATES IN ANTENNA LOS FRAME.                             *  00025670
C  *******************************************************************  00025680
C                                                                       00025690
C  QUANTIZE THE ANGLE DISCRIMINANTS TO 3/16 DB.                         00025700
       IAZDSC=INTT(5.333333*AZDISC*TCON+0.5)/TCON
       IELDSC=INTT(5.333333*ELDISC*TCON+0.5)/TCON                       00025720
       IF(IELDSC.GT.255)IELDSC=255
       IF(IAZDSC.GT.255)IAZDSC=255
       IF(IELDSC.LT.-256)IELDSC=-256
       IF(IAZDSC.LT.-256)IAZDSC=-256
C
C ------- NEW CODE AS OF JAN 28 1986-------------
C
C  UPDATE ESTIMATED TARGET INERTIAL AZIMUTH RATE.                       00025790
       IAZRATE=KSAT(IAZRATE+AT1A(MRNG,IMODE)*IAZDSC)                    00025800
C  UPDATE ESTIMATED TARGET INERTIAL ELEVATION RATE.                     00025810
       IELRATE=KSAT(IELRATE+AT1E(MRNG,IMODE)*IELDSC)                    00025820
C
       AZRATE=K6*DTOR*FLOAT(IAZRATE)
       ELRATE=K6*DTOR*FLOAT(IELRATE)
C
       IALRATE=KSAT(IAZRATE+AT2A(MRNG,IMODE)*IAZDSC)
       IBTRATE=KSAT(IELRATE+AT2E(MRNG,IMODE)*IELDSC)
C
       IF(IALRATE.GT.0) THEN
           ALRATE=K4*K5*DTOR*FLOAT(IALRATE/32)
       ELSE
           ALRATE=K4*K5*DTOR*FLOAT((IALRATE-31)/32)
       END IF
C
       IF(IBTRATE.GT.0) THEN
           BTRATE=K4*K5*DTOR*FLOAT(IBTRATE/32)
       ELSE
           BTRATE=K4*K5*DTOR*FLOAT((IBTRATE-31)/32)
       END IF
C
C  ************************************************                     00025840
C  * STEP 3· UPDATE INNER AND OUTER GIMBAL RATES. *                     00025850
C  ************************************************                     00025860
C  COMPUTE REQUIRED COMPONENTS OF ORBITER ANGULAR VELOCITY VECTOR IN    00025870
C  OUTER GIMBAL FRAME.                                                  00025880
       WGX=CP*EWB(1)+SP*EWB(2)                                          00025890
       WGY=CA*(-SP*EWB(1)+CP*EWB(2))+SA*EWB(3)                          00025900
       WGZ=-SA*(-SP*EWB(1)+CP*EWB(2))+CA*EWB(3)                         00025910
C  OUTER GIMBAL RATE.                                                   00025920
```

```
        IF(ABS(CB).LT.1.0E-6) GO TO 2                                     00025930
        ALRATE=(ALRATE+WGZ*SB)/CB-WGX
        GO TO 4                                                           00025950
     2  ALRATE=0.                                                         00025960
     4  CONTINUE                                                          00025970
C   INNER GIMBAL RATE.                                                    00025980
        BTRATE=BTRATE-WGY
C
C ---- END OF JAN 28 1986 MODIFICATIONS-------------------------
C                                                                         00026000
C   ***************************************************                   00026010
C   * STEP 4: UPDATE INNER AND OUTER GIMBAL POSITIONS. *                  00026020
C   ***************************************************                   00026030
C   OUTER GIMBAL POSITION (ALPHA ANGLE)                                   00026040
        AL=AL+TSAM*ALRATE                                                 00026050
C   INNER GIMBAL POSITION (BETA ANGLE)                                    00026060
        BT=BT+TSAM*BTRATE                                                 00026070
C                                                                         00026130
C  ADD ALPHA AND BETA TO OUTPUT IN DEG
             SSALP=AL*57.29576
             SSBET=BT*57.29576
C   *********************************************************             00026140
C   * STEP 6: TRANSFORM TARGET ANGLES AND INERTIAL ANGLE RATES TO *       00026150
C   *         BODY FRAME FOR USE IN DISPLAYS AND G AND N.         *       00026160
C   *********************************************************             00026170
C   NOTE: TRANSFORMATION TBL INCLUDES GIMBAL BIAS ERRORS AND RADAR YAW    00026180
C         ANGLE ERROR WRT BODY FRAME.                                     00026190
C   UPDATE TARGET INERTIAL PITCH RATE IN ORBITER BODY COORDINATES         00026200
C   FOR DISPLAY.                                                          00026210
        SPRTE=-1000.*(TBL(2,1)*AZRATE+TBL(2,2)*ELRATE)                    00026220
C   UPDATE TARGET INERTIAL ROLL RATE IN ORBITER BODY COORDINATES          00026230
C   FOR DISPLAY.                                                          00026240
        SRRTE=-1000.*(TBL(1,1)*AZRATE+TBL(1,2)*ELRATE)                    00026250
C   UPDATE ANTENNA PITCH ANGLE IN ORBITER BODY COORDINATES FOR DISPLAY.   00026260
        SPANG=-ASIN(TBL(1,3))*57.29576                                    00026270
C   UPDATE ANTENNA IN ORBITER BODY COORDINATES FOR DISPLAY.               00026280
        IF(TBL(2,3).EQ.0.0.AND.TBL(3,3).EQ.0.0) GO TO 5                   00026290
        SRANG=-ATAN2(-TBL(2,3),TBL(3,3))*57.29576                         00026300
        GO TO 7                                                           00026310
     5  IF(TBL(1,3).GT.0.0) SRANG=-90.0                                   00026320
        IF(TBL(1,3).LT.0.0) SRANG=90.0                                    00026330
        IF(TBL(1,3).EQ.0.0) STOP                                          00026340
C   RESOLVE POSSIBLE ANGLE AMBIGUITIES, VIZ., -90.<SPANG<90. AND          00026350
C   -180.<SRANG<180.                                                      00026360
     7  IF(SPANG.LE.90.) GO TO 10                                         00026370
        SPANG=-(180.-ABS(SPANG))*(SPANG/ABS(SPANG))                       00026380
        SRANG=(180.-ABS(SRANG))*(SRANG/ABS(SRANG))                        00026390
    10  CONTINUE                                                          00026400
C                                                                         00026410
C   NOTE: DEBUGGING PRINT STATEMENTS.                                     00026420
C       WRITE(6,899)                                                      00026430
  899   FORMAT(/' ATRACK DEBUGGGING DATA')                                00026440
C       WRITE(6,900) ALRATE,BTRATE,AZRATE,ELRATE,SRRTE,SPRTE              00026450
C       WRITE(6,901) TBL(1,1),TBL(1,2),TBL(2,1),TBL(2,2)                  00026460
C       WRITE(6,902) AZDISC,ELDISC,IAZDSC,IELDSC                          00026470
  900   FORMAT(' ALR,BTR,AZR,ELR,SRR,SPR=',6F14.9)                        00026480
  901   FORMAT(' TBL 2X2 =',4F10.4)                                       00026490
  902   FORMAT(' AZD,ELD,AD,ED =',2F10.4,2I9)                                 00026500
        RETURN                                                            00026510
        END                                                               00026520
C
C   ******************************************************
C   * INTEGER FUNCTION KSAT JAN 28 1986                  *
C   ******************************************************
```

```
C
C  THIS FUNCTION CHECKS ATRACK LOOP FOR SATURATION
C
      INTEGER FUNCTION KSAT(K)
C
      IF(K.GE.0) THEN
           KSAT=JMIN0(K,2**15)
      ELSE
           KSAT=JMAX0(K,-2**15)
      END IF
      RETURN
      END
C                                                                       00024530
C  ***************************************************                  00024540
C  * THIS SUBROUTINE IMPLEMENTS THE BREAK-TRACK ALGORITHM *             00024550
C  ***************************************************                  00024560
C                                                                       00024570
C                                                                       00024580
      SUBROUTINE BRKTRK                                                 00024590
      REAL IVMAX,THRSHC,THRSHO,IVDISC,INTT,IODISC                       00024595
      COMMON /ICNTL/IDUM2(17),MBKTRK,MBTSUM,MBT(8)                      00024600
      COMMON /DSCRM/DUM(3),VDISC,DUM1,ODISC,DUM2(3)                     00024610
      DATA IVMAX,THRSHC,THRSHO/51.,14.,-11./                            00024630
C                                                                       00024640
C  **********************************************                       00024650
C  * STEP 1: DETERMINE STATUS OF L-H DISCRETE (FTH) *                   00024660
C  **********************************************                       00024670
C                                                                       00024680
C  STEP 1-1: QUANTIZE THE VELOCITY DISCRIMINANT TO 3/16 DB STEPS.       00024690
      IVDISC=INTT(VDISC*5.333333+0.5)                                   00024700
C                                                                       00024710
C  STEP 1-2: DETERMINE STATUS OF L-H DISCRETE.                          00024720
      IFTH=0                                                            00024730
      IF(ABS(IVDISC).GE.IVMAX) IFTH=1                                   00024740
C                                                                       00024750
C  ***************************************************                  00024760
C  * STEP 2: DETERMINE STATUS OF ON-TARGET DISCRETE (OT) *              00024770
C  ***************************************************                  00024780
C                                                                       00024790
C  STEP 2-1: QUANTIZE THE O-DISCRIMINANT TO 3/16 DB STEPS.              00024800
      IODISC=INTT(ODISC*5.333333+0.5)                                   00024810
C                                                                       00024820
C  STEP 2-2: DETERMINE STATUS OF ON-TARGET DISCRIMINANT.                00024830
      IOT=0                                                             00024840
      IF(IODISC.GE.THRSHC) IOT=1                                        00024850
C                                                                       00024860
C  ************************************************************         00024870
C  * STEP 3: DETERMINE STATUS OF ADJACENT ON-TARGET DISCRETE (AOT) *    00024880
C  ************************************************************         00024890
      IAOT=0                                                            00024900
      IF(IODISC.LE.THRSHO) IAOT=1                                       00024910
C                                                                       00024920
C  ***********************************************************          00024930
C  * STEP 4: COMBINE ABOVE DISCRETES TO DETERMINE STATUS OF @ NO- *     00024940
C  *         TARGET@ DISCRETE (NOTARG).                         *       00024950
C  ***********************************************************          00024960
C  DEFINITION: THE NO-TARGET DISCRETE IS HIGH (OR 1) IF THE DISCRETES   00024970
C              FTH, OT, AND AOT ARE ALL LOW (OR 0).                     00024980
      NOTARG=(1-IFTH)*(1-IOT)*(1-IAOT)                                  00024990
C                                                                       00025000
C  *****************************************************                00025010
C  * STEP 5: DETERMINE STATUS OF BREAK-TRACK FLAG (MBKTRK) *            00025020
C  *****************************************************                00025030
C  DEFINITION: BREAK-TRACK SHALL BE DECLARED IF NOTARG=1 FOR AT         00025040
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C                LEAST 5 OF THE MOST RECENT 8 DATA CYCLES.              00025050
C                                                                       00025060
C  STEP 5-1: UPDATE MOVING WINDOW-OF-8 SUM (MBTSUM).                    00025070
      MBTSUM=MBTSUM+(NOTARG-MBT(1))                                     00025080
C                                                                       00025090
C  STEP 5-2: UPDATE STORAGE REGISTERS.                                  00025100
      DO 10 I=1,7                                                       00025110
   10 MBT(I)=MBT(I+1)                                                   00025120
      MBT(8)=NOTARG                                                     00025130
C                                                                       00025140
C  STEP 5-3: DETERMINE STATUS OF BREAK-TRACK FLAG (1=BREAK-TRACK).      00025150
      MBKTRK=MBTSUM/5                                                   00025160
C                                                                       00025170
C  NOTE: DEBUGGING PRINT STATEMENTS.                                    00025180
C     WRITE(6,900) IODISC,THRSHO,THRSHC,IVDISC,IVMAX,MBTSUM             00025190
  900 FORMAT(' OD,THO,THC,VD,THV,SUM =',6I8)                            00025200
      RETURN                                                            00025210
      END                                                               00025220
C                                                                       00008520
C                                                                       00008530
C  *****************************************************                00008540
C  * THIS SUBROUTINE CONTAINS THE CFAR DETECTION MODEL *                00008550
C  *****************************************************                00008560
C                                                                       00008570
C                                                                       00008580
      SUBROUTINE CFAR                                                   00008590
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),EDRNG,DUMC(2)          00008600
      COMMON /OUTPUT/MSWF,MTF,MSF,DUM1(7),IDUM1(4)                      00008610
      COMMON /ICNTL/IDUM2(8),KACCLK,MTP,IDUM3(4),MRNG,MSAM,MPRF         00008620
      COMMON /TGTDAT/NT,DUM3(500),RO(3),ROU(3),CGRNGE,CGVEL             00008630
      COMMON /DETDAT/SIGMA,CGANG                                        00008640
      DIMENSION RI(6),PW(6),NP(6),FW(3),TPRI(3),TS(2),P(41)             00008650
      DATA NRI,NSRCH/6,37 /,C,ALMDA/983.5,0.070845/,RI/2552.,5772.,     00008660
     2 11544.,23089.,43747.,57722./,PW/0.122,4.15,8.3,16.6,33.2,66.4/,  00008670
     3 NP/1,2,4,8,16,32/,FW/7.7215,3.3090,0.2969/,TS/0.122,2.075/,      00008680
     4 TPRI/143.5,334.7,3731.1/                                         00008690
      DATA P/6*0.0,.001,.003,2*.004,.008,.012,.015,.043,.053,.076,.107,00008700
     2 .147,.193,.244,.312,.363,.444,.514,.590,.644,.706,.765,.815,.861,00008710
     3 .882,.918,.937,.955,.966,.976,.980,.989,.991,.997,.996/          00008720
      PI=3.14159265                                                     00008730
C                                                                       00008740
C  ******************************************************************   00008750
C  * STEP 1: SET INTERNAL CONTROLS BASED UPON SYSTEM OPERATING MODE *   00008760
C  ******************************************************************   00008770
C                                                                       00008780
C  STEP 1-1: GPC MODES OR AUTO/MANUAL MODES"                            00008790
      IF(IASM.GE.3) GO TO 15                                            00008800
C                                                                       00008810
C  STEP 1-2: SET INTERNAL CONTROLS FOR APPROPRIATE MODE.                00008820
C                                                                       00008830
C  CONTROL SETTINGS FOR GPC MODES.                                      00008840
C                                                                       00008850
C  DETERMINE RANGE INTERVAL.                                            00008860
      DO 5 I=1,NRI                                                      00008870
      MRNG=I                                                            00008880
      IF(RI(I).GT.EDRNG) GO TO 10                                       00008890
    5 CONTINUE                                                          00008900
C                                                                       00008910
C  SET SAMPLE RATE                                                      00008920
   10 MSAM=2                                                            00008930
C                                                                       00008940
C  DETERMINF PRF                                                        00008950
      MPRF=1                                                            00008960
      IF(EDRNG.GE.RI(6)) MPRF=2                                         00008970
      GO TO 20
C
C  CONTROL SETTINGS FOR AUTO/MANUAL MODES.
C
C  SET RANGE INTERVAL.
   15 MRNG=6
C
C  SET SAMPLE RATE.
      MSAM=2
C
C  SET PRF.
      MPRF=1
C
C  ****************************************************
C  * STEP 2: COMPUTE NOMINAL SNR AT VIDEO FILTER OUTPUT *
C  ****************************************************
   20 SNR=SNRV(SIGMA,CGRNGE)
C
C  ****************************************************
C  * STEP 3: IF NOT SCANNING ADD BEAMSHAPE LOSS TO SNRV *
C  ****************************************************
C
C  STEP 3-1: CHECK SCAN FLAG.
      IF(MSF.EQ.1) GO TO 25
C
C  STEP 3-2: COMPUTE BEAMSHAPE LOSS --- BASED UPON C.G. POSITION OFF
C            BORESIGHT.
      BETA2=SPAT(CGANG)**2
C
C  STEP 3-3: ADD BEAMSHAPE LOSS TO NOMINAL SNRV, I.E. COMPUTE ACTUAL
C            SNRV.
      SNR=SNR*BETA2
C
C  *****************************************************************
C  * STEP 4: COMPUTE NET PROCESSOR GAIN AND COMBINE WITH SNRV TO FORM *
C  *         SNRD.                                                  *
C  *****************************************************************
C
C  STEP 4-1: COMPUTE RANGE GATE LOSS (RGL) --- DIFFERS FOR GPC AND
C            AUTO/MANUAL MODES.
C
C  COMPUTE EQUIVALENT RANGE OF XMIT PULSEWIDTH.
   25 CTD2=C*PW(MRNG)/2.
C
C  DETERMINE OPERATING MODE
      IF(IASM.GE.3) GO TO 30
C
C  COMPUTE RGL FOR GPC MODES.
      DEL=ABS(EDRNG-CGRNGE)/CTD2
      IF(DEL.GE.1.5) RGL=0.0
      IF(DEL.GE.0.5.AND.DEL.LT.1.5) RGL=.6666666*(1.5-DEL)**2
      IF(DEL.LT.0.5) RGL=.6666666
      GO TO 35
C
C  COMPUTE RGL FOR AUTO/MANUAL MODES
   30 DEL=ABS(CGRNGE)/CTD2
      DEL1=DEL-INT(DEL)
      IF(DEL.LE.1.0) RGL=DEL*DEL
      IF(DEL.GT.1.0.AND.DEL.LT.4.5.AND.DEL1.LT.0.5)
     2 RGL=(1.0-DEL1)**2
      IF(DEL.GT.1.0.AND.DEL.LT.4.5.AND.DEL1.GE.0.5)
     2 RGL=DEL1*DEL1
C
C  STEP 4-2: COMPUTE NET PRESUM GAIN --- SAME FOR ALL PASSIVE ANTENNA
```

```
C                STEERING MODES.
C
C     COMPUTE DOPPLER FREQUENCY ASSOCIATED WITH TARGET RADIAL VELOCITY
 35       FDOP=-2.*CGVEL/ALMDA*1.0E-06
C
C     COMPUTE ARGUMENT ASSOCIATED WITH TARGET VELOCITY
          ARG=PI*FDOP*TS(MSAM)
C
C     COMPUTE NET PRESUM GAIN
          PSG=SUM(ARG,NP(MRNG))
C
C     STEP 4-3: COMPUTE NET DOPPLER FILTER GAIN -- SAME FOR ALL PASSIVE
C               ANTENNA STEERING MODES.
C
C     COMPUTE NUMBER OF DOPPLER FILTER NEAREST TARGET.
          MFIL=MOD(INT(CGVEL/FW(MPRF)+320.5),32)
C
C     COMPUTE ARGUMENT ASSOCIATED WITH TARGET DOPPLER
          ARG=PI*(FLOAT(MFIL)/32.+FDOP*TPRI(MPRF))
C
C     COMPUTE NET DOPPLER FILTER GAIN
          DFG=SUM(ARG,16)
C
C     STEP 4-4: COMPUTE NET PROCESSOR GAIN.
          NPG=RGL*PSG*DFG
C
C     STEP 4-5: COMPUTE SNR AT DOPPLER FILTER OUTPUT
          SNR=SNR*NPG
C
C     **********************************************************
C     * STEP 5: DETERMINE PROBABILITY OF DETECTION BASED UPON SNR *
C     **********************************************************
C
C     STEP 5-1: DETERMINE INDEX TO ACCESS APPROPRIATE CURVE
          IF(IASM.GE.3) GO TO 40
          NCRV=1
          GO TO 45
 40       NCRV=3
C
C     ADJUST INDEX FOR SCANNING
 45       NCRV=NCRV+MSF
C
C     STEP 5-2: CONVERT SNR TO DB.
          IF(SNR.LE.1.0E-08) GO TO 50
          SNR=10.*ALOG10(SNR)
          GO TO 55
 50       SNR=-100.
C
C     STEP 5-3: SNR OUTSIDE (0 DB, +20 DB) INTERVAL" -- IF SO, SET
C               OUTCOME APPROPRIATELY AND SKIP REMAINING STEPS.
C
C     IF SNRD < 0. DB -- DECLARE A MISS.
 55       IF(SNR.LE.0.) GO TO 60
C
C     IF SNRD > 20. DB -- DECLARE A HIT.
          IF(SNR.GT.20.) GO TO 65
C
C     STEP 5-4: COMPUTE INDEX FOR LOOKUP TABLE AND FACTORS FOR LINEAR
C               INTERPOLATION.
          SCALE=(SNR+0.)*2.+1.0000001
          ISNR=INT(SCALE)
          REMAIN=SCALE-FLOAT(ISNR)
C
C     STEP 5-5: DETERMINE PD USING TABLE AND LINEAR (IN DB) INTERPOLATION.
```

```
      PROB=P(ISNR)+REMAIN*(P(ISNR+1)-P(ISNR))
C
C  **************************************************
C  * STEP 6: DETERMINE OUTCOME OF DETECTION ATTEMPT *
C  **************************************************
C
      X=RNDU(NSRCH)
      IF(X.LE.PROB) GO TO 65
C
C  *****************************************************************
C  * STEP 7: SET CONTROLS BASED UPON OUTCOME OF DETECTION ATTEMPT *
C  *****************************************************************
C
C  STEP 7-1: IF NO DETECTION ---- SET TARGET PRESENT FLAG LOW.
   60 MTP=0
      RETURN
C
C  STEP 7-2: IF DETECTION SUCCESSFUL ---- SET TARGET PRESENT FLAG
C            HIGH AND INITIALIZE ACQUISITION CLOCK.
   65 MTP=1
      KACCLK=0
      RETURN
      END
C
C
C  ********************************************************
C  * THIS SUBROUTINE UPDATES ALL RADAR INTERNAL CONTROLS. *
C  ********************************************************
C
C
      SUBROUTINE CNTRLS
      REAL INTT,NFIL,IRNG,IRDOT
      COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)
      COMMON /OUTPUT/IDUM0(3),SRNG,SRDOT,DUM2(5),IDUM(4)
      COMMON /ICNTL/I1DUM(14),MRNG,MSAM,MPRF,IDUM1(10),MPFOLD
      COMMON /RTDAT/IRDOT,IRNG,RBIAS,VEST(4),MDF(5)
      DIMENSION RI(10),FW(3)
C RI(4) CHANGED TO 2560 FROM 2552
      DATA RI/120.,640.,1520.,2560.,5760.,11520.,23040.,43520.,
     2        49920.,1.8228E+6/
      DATA FW/7.7215,3.3090,0.2969/,NRI/10/
C
C
C  ***********************************************************
C     IMPLEMENTATION OF HYSTERISIS FOR THE SAMPLING RATE
C     CHANGE AND FOR THE PRF CHANGE ALONG WITH CHANGES IN
C     RI(RANGE INTERVAL) WAS COMPLETED FEB 6,1986 BY M. MEYER
C  ***********************************************************
C
C  ****************************************
C  * STEP 1: SET RANGE INTERVAL PARAMETER *
C  ****************************************
      XRNG=IRNG*0.3125
      DO 60 I=1,NRI
      IF(XRNG.LE.RI(I)) GO TO 70
   60 CONTINUE
   70 MRNG=I
      IF(MRNG.GT.NRI) STOP
C
C  *************************************
C  * STEP 2: SET SAMPLE RATE PARAMETER *
C  *************************************
      IF(IMODE.GE.2)  GO TO 74
      IF(MRNG.GT.9) GO TO 72
      MSAM=1
      GO TO 80
72    MSAM=2
      GO TO 80
C***** MODIFIED FEB 6 1986 BY M. MEYER*****************
74    IF(MSAM.EQ.1)THEN
         IF(XRNG.GT.3200.)THEN
            MSAM=2
         ELSE
            MSAM=1
C***** MODIFIED FEB 17,1986 BY M. MEYER ***************
C***** GUARANTEES THE CORRECT LOOP BANDWIDTHS**********
C***** FOR THE HYSTERISIS LOOP************************
C
            IF(XRNG.GT.2560) MRNG=4
C
C*****************************************************
         END IF
      ELSE
         IF(XRNG.GT.2560.)THEN
            MSAM=2
         ELSE
            MSAM=1
         END IF
      END IF
C
C  ***************************
C  * STEP 3: SET PRF PARAMETER *
C  ***************************
C
C  STEP 3-1: DETERMINE IF IN ACTIVE OR PASSIVE MODE.
80    IF(IMODE.GE.2) GO TO 84
C
C  STEP 3-2: DETERMINE CORRECT PRF FOR GIVEN OPERATING MODE.
      IF(MRNG.GT.9) GO TO 82
      MPRF=1
      GO TO 90
82    MPRF=3
      GO TO 90
C ****** MODIFIED FEB 6 1986 BY M. MEYER ***********
84    IF(MPRF.EQ.1)THEN
         IF(XRNG.GT.49920.)THEN
            MPRF=2
         ELSE
            MPRF=1
         END IF
      ELSE
         IF(XRNG.GT.43520.)THEN
           MPRF=2
C***** MODIFIED FEB 17, 1986 BY M. MEYER*************
C***** GUARANTEES THE CORRECT CONSTANTS ************
C***** FOR THE LOW PRF******************************
C
           MRNG=10
C
C**************************************************
         ELSE
            MPRF=1
         END IF
      END IF
90    CONTINUE
C
C  STEP 3-3: IF PRF HAS CHANGED FROM PREVIOUS DATA CYCLE, THEN
C            RESET THE 5 DOPPLER TRACKING FILTERS ACCORDINGLY.
```

```
      IF(MPFOLD.EQ.MPRF) GO TO 96                                       00029100
      NFIL=INTT((-SRDOT/FW(MPRF))+0.5)+31998.                           00029110
      XX=AMOD(NFIL,32.)                                                 00029115
      MDF(1)=INT(XX)                                                    00029120
      DO 95 I=1,4                                                       00029130
   95 MDF(I+1)=MOD(MDF(1)+I,32)                                         00029140
   96 MPFOLD=MPRF                                                       00029150
C                                                                       00029160
C  NOTE: DEBUGGING PRINT STATEMENTS.                                    00029170
C     WRITE(6,999) MPRF,MPFOLD,MDF(1)                                   00029180
  999 FORMAT(' MPRF,MPFOLD,MDF1 =',3I8)                                 00029190
      RETURN                                                            00029200
      END                                                               00029210
C                                                                       00006680
C                                                                       00006690
C  *********************************************************************00006700
C  * THIS SUBROUTINE PERFORMS THE TARGET DETECTION FUNCTION FOR ACTIVE *00006710
C  * AND PASSIVE MODES AND ALL ANTENNA STEERING MODES.                 *00006720
C  *********************************************************************00006730
C                                                                       00006740
C                                                                       00006750
      SUBROUTINE DETECT                                                 00006760
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),EDRNG,DUMC(2)          00006770
      COMMON /ICNTL/IDUM2(9),MTP,IDUM3(17)                              00006780
      COMMON /SYSDAT/DUM2(12),TGTSIG,GPS,GAS                            00006790
      COMMON /TGTDAT/NT,DUM3(500),RO(3),ROU(3),CGRNGE,CGVEL             00006800
      COMMON /DETDAT/SIGMA,CGANG                                        00006810
C                                                                       00006820
C  ***********************************************                      00006830
C  * STEP 1: COMPUTE TARGET PARAMETERS WRT RADAR *                      00006840
C  ***********************************************                      00006850
C                                                                       00006860
C  STEP 1-1: TRANSFORM TARGET C.G. POSITION AND VELOCITY TO LOS FRAME.  00006870
      CALL TRNSFM                                                       00006880
      CALL PVTRAN                                                       00006890
C                                                                       00006900
C  STEP 1-2: COMPUTE TARGET C.G. ANGLE OFF-BORESIGHT (NON-SCANNING).    00006910
CCCCCCCCCCCCCCCCCCCCCCCC MOD MAR 24 1983 CCCCCCCCCCCCCCCCCCCCCC
      CGANG=ACOS(-ROU(3))                                                00006920
C                                                                       00006930
C  STEP 1-3: DETERMINE TARGET CROSS-SECTION.                            00006940
      SIGMA=TGTSIG                                                      00006950
C                                                                       00006960
C  ****************************************************                 00006970
C  * STEP 2: PRELIMINARY DETECTION MODE DETERMINATION *                 00006980
C  ****************************************************                 00006990
C                                                                       00007000
C  STEP 2-1: DETERMINE WHETHER ACTIVE OR PASSIVE.                       00007010
      IF(IMODE.EQ.1) GO TO 5                                            00007020
C                                                                       00007030
C  STEP 2-2: GPC MODES OR AUTO/MANUAL MODES"                            00007040
      IF(IASM.GE.3) GO TO 10                                            00007050
      GO TO 15                                                          00007060
C                                                                       00007070
C  *****************************************                            00007080
C  * STEP 3: ACTIVE MODE DETECTION PROCESS *                            00007090
C  *****************************************                            00007100
C                                                                       00007110
    5 CALL SINGLE                                                       00007120
      RETURN                                                            00007130
C                                                                       00007140
C  ******************************************************               00007150
C  * STEP 4: PASSIVE AUTO/MANUAL MODE DETECTION PROCESS *               00007160
C  ******************************************************               00007170
```

```
C
C  STEP 4-1: CHECK SHORT RANGE FIRST -- CALL SINGLE-HIT DETECTION
C            MODEL.
 10   CALL SINGLE
C
C  STEP 4-2: CHECK FOR SUCCESS IN SINGLE-HIT DETECTION -- IF NOT SUC
C            CESSFUL, THEN TRY LONG RANGE SEARCH.
      IF(MTP.EQ.0) CALL CFAR
      RETURN
C
C  ********************************************
C  * STEP 5: PASSIVE GPC MODES DETECTION PROCESS *
C  ********************************************
C
C  STEP 5-1: CHECK DESIGNATED RANGE.
 15   IF(EDRNG.GT.2552.) GO TO 20
C
C  STEP 5-2: IF DESIGNATED RANGE < 0.42 NM -- USE SINGLE-HIT
C            DETECTION MODEL.
      CALL SINGLE
      RETURN
C
C  STEP 5-3: IF DESIGNATED RANGE > 0.42 NM -- USE CFAR DETECTION MODEL.
 20   CALL CFAR
      RETURN
      END
C
C
C  ****************************************************************
C  * THIS SUBROUTINE ADDS THE EQUIVALENT NOISE TO THE ANGLE, RANGE, *
C  * VELOCITY AND ON-TARGET DISCRIMINANT COMPONENTS AND THEN COM-   *
C  * PUTES THE ANGLE, RANGE, VELOCITY, AND ON-TARGET DISCRIMINANTS. *
C  ****************************************************************
C
C
      SUBROUTINE DISCRM
      REAL LATE,MEAN
      COMMON /OUTPUT/MSWF,MTF,MSF,SRNG,SRDOT,SPANG,SRANG,SPRTE,
     2               SRRTE,SRSS,MADVF,MRDVF,MARDVF,MRRDVF
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)
      COMMON /ICNTL/I3DUM(14),MRNG,MSAM,MPRF,IDUM4(10)
      COMMON /SYSDAT/TSAM,DR(3),CP,SP,PSI,PSBIAS,ALBIAS,BTBIAS,GP,GA,
     2               DUMS(3)
      COMMON /TGTDAT/NT,DUM5(506),CGRNGE,CGVEL
      COMMON /DSCRM/AZDISC,ELDISC,RDISC,VDISC,RRTE,ODISC,SIGBR1,SNRD,
     2              SIGDB
      COMMON /SIGDAT/SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE,DF1,DF5,
     2               DF2,DF4,SIGBAR
      COMMON /NOISE/NS1,NS2,NN(10),GAUSS(320)
      COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD
      DIMENSION NFREQ(2),PDIA(2),PDIR(2),PDIV(2),PS(10,2),BN(2),PT(3)
     2 ,TDC(3)
      DIMENSION QNV(2)
C
C ----------PS AND QNV CONSTANT CHANGES FEB 17,1986 BY M. MEYER----------
C
      DATA NFREQ/1,5/,BN/9772.4,616.6/
      DATA PS/9*4.,2.,5*4.,2.,4.,8.,8.,16./
     2     ,PDIA,PDIR,PDIV/1.4142,3.1623,2.0,4.4721,2.8284,6.3246/,
     3      PT/42658.,3125.,195.3/
      DATA QNV/.00067,.011/
      DATA TDC/0.05122118,0.1195161,0.2561557/
C
C  NOTE: DEBUGGING PRINT STATEMENTS.
```

```
C      WRITE(6,900) SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE
C      WRITE(6,901) DF1,DF5,DF2,DF4,SIGBAR
  900  FORMAT(' SPZ,SMZ,SPL,SML,E,L =',6F10.2)
  901  FORMAT(' DF1,DF5,DF2,DF4,SIG =',5F10.2)
C
C  **************************************************************
C  * STEP 1: COMPUTE CONSTANT USED IN SIGNAL SCALING AND COMPUTATION *
C  *          OF NOISE STATISTICS.                                  *
C  ********************** ***************************************
C
C
       TCON=(TSAM/TDC(MPRF))**.5
C
C  STEP 1-1: COMPUTE CONSTANT (NOTE: IT IS DIFFERENT FOR ACTIVE AND
C             PASSIVE MODES).
       IF(IMODE.EQ.2) GO TO 5
C  NOTE: THIS IS THE CONSTANT USED IN ACTIVE MODE.
       YY=GA*PS(MRNG,IMODE)/(CGRNGE**2*BN(MSAM))
       S1=YY/FLOAT(NFREQ(IMODE))
       GO TO 10
C  NOTE: THIS IS THE CONSTANT USED IN PASSIVE MODE.
CCCCCCCCCCCCCCCCCCCCCCCCCCC MODS 2-15-83 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
5        CONTINUE
          PTFIX=PT(ITXP)
          IF(SRNG.LT.640.)PTFIX=4.2
          ISTS7=0
          IF(ISTS7.EQ.1)PTFIX=4.2
C
       YY=GP*PS(MRNG,IMODE)*PTFIX   /(CGRNGE**4*BN(MSAM))
       S1=YY/FLOAT(NFREQ(IMODE))
C
C  STEP 1-2: COMPUTE PEAK SIGNAL POWER TO AVERAGE THERMAL NOISE POWER
C             AT DOPPLER FILTER OUTPUT.
   10  SNRDT=YY*SIGBAR
C          WRITE(6,221)YY,SIGBAR
  221     FORMAT('YY,SIGBAR =',2F14.5)
       SNRDTD=10.*ALOG10(SNRDT)
       SIGDB=10.*ALOG10(SIGBAR)
       SIGBR1=SIGBAR
C222   WRITE(6,990) SNRDTD,SIGDB
  990  FORMAT(' SNRDTD,SIGDB =',2F14.2)
C
C  STEP 1-3: COMPUTE PEAK SIGNAL POWER TO TOTAL (THERMAL PLUS
C             QUANTIZATION) NOISE POWER AT THE DOPPLER FILTER OUTPUT.
       CALL SATNSE(SNF)
       XX=SNF*AGCO
       XX=XX/(XX+QNV(MSAM))
       S1=S1*XX
       YY=YY*XX
       SNRD=YY*SIGBAR
       SNRD=10.*ALOG10(SNRD)
C
C  STEP 1-4: UPDATE NOISE SEQUENCE.
       NN(1)=MOD(NN(1)+1,320)+1
       DO 15 I=2,10
   15  NN(I)=MOD(NN(I-1)+29,320)+1
       ID1=NN(1)
       GAUSS(ID1)=ANORM(NS1,NS2)
C
C  ****************************************************
C  * STEP 2: COMPUTE ANGLE DISCRIMINANT (INCLUDES NOISE) *
C  ****************************************************
C
```

```
C  STEP 2-1: CHECK ANTENNA STEERING MODE —— SKIP STEP 2 IF IN
C            GPC-DES OR MANUAL.
CCCCCCCCCCCCCCCCCCCCCC MOD FEB 16 1983 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 20
C
C  STEP 2-2: COMPUTE ANGLE DISCRIMINANT COMPONENT SCALE FACTOR.
      ASCALE=S1*PDIA(IMODE)
C
C  STEP 2-3: COMPUTE STATISTICS OF ADDITIVE NOISE FOR ANGLE
C            DISCRIMINANT COMPONENTS.
      MEAN=PDIA(IMODE)
      VARPAZ=SQRT(2.*S1*SPAZ+1.)
      VARMAZ=SQRT(2.*S1*SMAZ+1.)
      VARPEL=SQRT(2.*S1*SPEL+1.)
      VARMEL=SQRT(2.*S1*SMEL+1.)
C
C  STEP 2-4: ADD EQUIVALENT NOISE TO ANGLE DISCRIMINANT COMPONENT
C            SIGNALS.
      ID6=NN(6)
      SPAZ=ABS(ASCALE*SPAZ+MEAN+VARPAZ*GAUSS(ID1))
      SMAZ=ABS(ASCALE*SMAZ+MEAN+VARMAZ*GAUSS(ID6))
      ID2=NN(2)
      ID7=NN(7)
      SPEL=ABS(ASCALE*SPEL+MEAN+VARPEL*GAUSS(ID2))
      SMEL=ABS(ASCALE*SMEL+MEAN+VARMEL*GAUSS(ID7))
C
C  STEP 2-5: COMPUTE AZ AND EL DISCRIMINANT COMPONENTS.
      AZDISC=10.*ALOG10(SPAZ/SMAZ)
      ELDISC=10.*ALOG10(SPEL/SMEL)
C      AZDISC=0.
C      ELDISC=0.
C
C  ******************************************************
C  * STEP 3: COMPUTE RANGE DISCRIMINANT (INCLUDES NOISE) *
C  ******************************************************
C
C  STEP 3-1: COMPUTE RANGE DISCRIMINANT COMPONENT SCALE FACTOR.
   20 RSCALE=S1*PDIR(IMODE)
C
C  STEP 3-2: COMPUTE STATISTICS OF ADDITIVE NOISE FOR RANGE
C            DISCRIMINANT.
      MEAN=PDIR(IMODE)
      VARELY=SQRT(2.*S1*EARLY+1.)*TCON
      VARLTE=SQRT(2.*S1*LATE+1.)*TCON
C
C  STEP 3-3: ADD EQUIVALENT NOISE TO RANGE DISCRIMINANT COMPONENT
C            SIGNALS.
      ID3=NN(3)
      ID8=NN(8)
      EARLY=ABS(RSCALE*EARLY+MEAN+VARELY*GAUSS(ID3))
      LATE=ABS(RSCALE*LATE+MEAN+VARLTE*GAUSS(ID8))
C
C  STEP 3-4: COMPUTE RANGE DISCRIMINANT.
      RDISC=10.*ALOG10(LATE/EARLY)
C
C  *********************************************************
C  * STEP 4· COMPUTE VELOCITY DISCRIMINANT (INCLUDES NOISE) *
C  *********************************************************
C
C  STEP 4-1: COMPUTE VELOCITY DISCRIMINANT COMPONENT SCALE FACTOR.
      VSCALE=S1*PDIV(IMODE)
C
C  STEP 4-2: COMPUTE STATISTICS OF ADDITIVE NOISE FOR VELOCITY
C             DISCRIMINANT COMPONENTS.
```

```
      MEAN=PDIV(IMODE)
      VARDF2=SQRT(2.*S1*DF2+1.)
      VARDF4=SQRT(2.*S1*DF4+1.)
C
C  STEP 4-3: ADD EQUIVALENT NOISE TO VELOCITY DISCRIMINANT
C              COMPONENT SIGNALS.
      ID4=NN(4)
      ID9=NN(9)
      DF2=ABS(VSCALE*DF2+MEAN+VARDF2*GAUSS(ID4))
      DF4=ABS(VSCALE*DF4+MEAN+VARDF4*GAUSS(ID9))
C
C  STEP 4-4: COMPUTE VELOCITY DISCRIMINANT.
      VDISC=10.*ALOG10(DF2/DF4)
C
C  ************************************************************
C  * STEP 5: COMPUTE ON-TARGET DISCRIMINANT --- USED FOR BREAK- *
C  *         TRACK AND VELOCITY DATA INVALID DETERMINATION      *
C  ************************************************************
C
C  STEP 5-1: COMPUTE STATISTICS OF ADDITIVE NOISE FOR OUTER DOPPLER
C            FILTER SIGNALS.
      VARDF1=SQRT(2.*S1*DF1+1.)
      VARDF5=SQRT(2.*S1*DF5+1.)
C
C  STEP 5-2: ADD EQUIVALENT NOISE TO OUTER DOPPLER FILTER SIGNALS.
      ID5=NN(5)
      ID10=NN(10)
      DF1=ABS(VSCALE*DF1+MEAN+VARDF1*GAUSS(ID5))
      DF5=ABS(VSCALE*DF5+MEAN+VARDF5*GAUSS(ID10))
C
C  STEP 5-3: COMPUTE ON-TARGET DISCRIMINANT.
C            NOTE: THE FACTOR OF SQRT(2.) IS DUE TO THE METHOD OF
C                  NORMALIZATION OF DISCRIMINANT COMPONENTS.
      ODISC=10.*ALOG10((EARLY+LATE)*SQRT(2.)/(DF1+DF5))
C
C  NOTE: DEBUGGING PRINT STATEMENTS.
C     WRITE(6,902) AZDISC,ELDISC,RDISC,VDISC,ODISC
C     WRITE(6,903) SNRD,SIGDB,SIGBAR
C     WRITE(6,904) SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE
C     WRITE(6,905) DF1,DF5,DF2,DF4,SIGBAR
  902 FORMAT(/' AZD,ELD,RD,VD,OD =',5F14.6)
  903 FORMAT(' SNRD,SIGDB,SIGBAR =',3F14.6)
  904 FORMAT(' SPZ,SMZ,SPL,SML,E,L+NOISE =',6F10.2)
  905 FORMAT(' DF1,DF5,DF2,DF4,SIG+NOISE =',5F10.2)
      RETURN
      END
C
C
C  ************************************************************
C  * THIS FUNCTION COMPUTES THE DOPPLER FILTER OUTPUT AMPLITUDE *
C  * AND PHASE FOR AN INPUT SIGNAL OF FREQUENCY X.              *
C  ************************************************************
C
C
      COMPLEX FUNCTION DOPFIL(X)
      COMPLEX DENOM,NUMER
      DENOM=1.-CEXP(CMPLX(0.,X))
      DENOM=16.*DENOM
C  CHECK FOR DENOMINATOR EQUAL TO ZERO.
      XX=CABS(DENOM)
      IF(XX.GT.1.0E-06) GO TO 10
      DOPFIL=(1.0,0.0)
      RETURN
   10 NUMER=1.-CEXP(CMPLX(0.,16.*X))
```

```
      DOPFIL=NUMER/DENOM                                                00031330
      RETURN                                                            00031340
      END                                                               00031350
C                                                                       00030650
C                                                                       00030660
C  *******************************************************************  00030670
C  * THIS FUNCTION GIVES THE ANTENNA DIFFERENCE PATTERN WEIGTHING OF *  00030680
C  * THE RADAR SIGNAL FOR THE GIVEN ANGLE(IN RADIANS) OFF BORESIGHT. *  00030690
C  * NOTE: THIS PATTERN IS THE DERIVATIVE OF THE SUM PATTERN         *  00030700
C  *******************************************************************  00030710
C                                                                       00030720
C                                                                       00030730
      FUNCTION DPAT(X)                                                  00030740
      IF(ABS(X).GT.1.E-4) GO TO 10                                      00030750
      DPAT=-0.6228*X                                                    00030760
      RETURN                                                            00030770
10    Y=93.80*X                                                         00030780
      DPAT=1.1465*(Y*COS(Y)-SIN(Y))/(Y*Y)                               00030790
      RETURN                                                            00030800
      END                                                               00030810
C                                                                       00003000
C                                                                       00003010
C  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00003020
C  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00003030
C  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00003040
C  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00003050
C  $$                                                                 $$00003060
C  $$                                                                 $$00003070
C  $$    PREPARE TO RECEIVE THE PRODUCT YOU'VE BEEN WAITING FOR ...   $$00003080
C  $$                                                                 $$00003090
C  $$        THE ACTUAL SES SPACE SHUTTLE RADAR SIMULATION ***        $$00003100
C  $$                                                                 $$00003110
C  $$                                                                 $$00003120
C  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00003130
C  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00003140
C  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00003150
C  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00003160
C                                                                       00003170
C                                                                       00003180
C                                                                       00003190
C                                                                       00003200
C                                                                       00003210
C                                                                       00003220
C  ********************************************************             00003230
C  *  EXECUTIVE PROGRAM: INTERFACE WITH PARENT SIMULATION *             00003240
C  ********************************************************             00003250
C                                                                       00003260
C                                                                       00003270
      SUBROUTINE EXEC                                                   00003280
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)                00003290
      COMMON /OUTPUT/MSWF,MTF,MSF,DUM(7),IDUM2(4)                       00003300
      COMMON /ICNTL/IOLDPW,IOLDMD,IOLDSM,ISHOLD,KMSCLK,KWMUP,IDUM1(3),  00003310
     2              MTP,IDUM5(17)                                       00003320
      DATA DATINT/1.0/                                                  00003330
      KWMUP=1                                                           00003340
C                                                                       00003350
C  *************************************************                    00003360
C  * STEP 0: INITIALIZE ALL TARGET AND SYSTEM DATA *                    00003370
C  *************************************************                    00003380
      IF(DATINT.NE.1.0) GO TO 1                                         00003390
       CALL SETIT
      CALL DATA                                                         00003400
C     CALL SYSINT                                                       00003410
      IOLDPW=IPWR                                                       00003420
```

C-5

```
      DATINT=0.0
    1 II=1
      IF(II.EQ.1) GO TO 30
C
C ************************************
C * STEP 1: CHECK SYSTEM POWER SWITCH *
C ************************************
      IF(IPWR.GT.1) GO TO 5
C IF POWER OFF ---INITIALIZE ALL SYSTEM FLAGS AND CLOCKS.
      KMSCLK=0
      CALL SYSINT
      RETURN
C IF POWER ON --- UPDATE MASTER CLOCK AND DETERMINE OPERATING MODE.
    5 KMSCLK=KMSCLK+1
C
C ************************************
C * STEP 2: CHECK SYSTEM MODE SWITCH *
C ************************************
      IF(IMODE.LT.3) GO TO 7
C IF SYSTEM IN COMM(IMODE=3) --- INITIALIZE ALL SYSTEM FLAGS.
      CALL SYSINT
      RETURN
C IF SYSTEM IN RADAR MODE --- CHECK FOR CHANGE IN MODE (I.E. ACTIVE-TO
C -PASSIVE OR PASSIVE-TO-ACTIVE).
    7 IF(IMODE.EQ.IOLDMD) GO TO 10
C IF RADAR MODE CHANGE --- RESET SYSTEM TO SEARCH.
      CALL SYSINT
C UPDATE STATUS OF IOLDMD.
   10 IOLDMD=IMODE
C
C *********************************************
C * STEP 3: DETERMINE WHETHER SYSTEM IN STANDBY *
C *********************************************
      IF(IPWR.GT.2) GO TO 15
      CALL SYSINT
      RETURN
C
C **************************************************
C * STEP 4: DETERMINE WHETHER WARMUP PERIOD EXCEEDED *
C *************************************************
   15 IF(KMSCLK.GT.KWMUP) GO TO 20
C IF NOT EXCEEDED --- INITIALIZE ALL SYSTEM FLAGS AND RETURN.
      CALL SYSINT
      RETURN
C IF EXCEEDED --- CONTINUE SYSTEM OPERATING MODE DETERMINATION.
C
C **************************************************************
C * STEP 5: DETERMINE IF THERE HAS BEEN AN ANTENNA STEERING MODE *
C *        CHANGE                                                *
C **************************************************************
   20 IF(IASM.EQ.IOLDSM) GO TO 25
C IF CHANGE HAS OCCURRED --- RESET ALL FLAGS AND GO TO NEW MODE.
      CALL SYSINT
   25 IOLDSM=IASM
C
C ***************************************************************
C * STEP 5: DETERMINE WHETHER SYSTEM IS IN SEARCH AND ACQUISITION *
C *        OR TRACK MODE.                                         *
C ***************************************************************
      IF(MTF.EQ.1.OR.MTP.EQ.1) GO TO 30
C IF TRACK FLAG DOWN --- GO TO SEARCH MODE.
      CALL SEARCH
      RETURN
C IF TRACK FLAG IS UP --- GO TO  TRACK MODE.
```

```
   30  CALL TRACK                                                        00004070
       RETURN                                                            00004080
       END                                                               00004090
C                                                                        00032440
C                                                                        00032450
C  **************************************************************       00032460
C  * THIS SUBROUTINE GENERATES A (3X3) MATRIX TGA THAT PRODUCES *       00032470
C  * A ROTATION OF GA RADIANS ABOUT THE Y-AXIS.                 *       00032480
C  **************************************************************       00032490
C                                                                        00032500
C                                                                        00032510
       SUBROUTINE GAMMA(TGA,GA)                                          00032520
       DIMENSION TGA(3,3)                                                00032530
       DO 10 I=1,3                                                       00032540
       DO 10 J=1,3                                                       00032550
   10  TGA(I,J)=0.0                                                      00032560
       TGA(2,2)=1.0                                                      00032570
       TGA(1,1)=COS(GA)                                                  00032580
       TGA(1,3)=-SIN(GA)                                                 00032590
       TGA(3,3)=TGA(1,1)                                                 00032600
       TGA(3,1)=-TGA(1,3)                                                00032610
       RETURN                                                            00032620
       END                                                               00032630
C                                                                        00031360
C                                                                        00031370
C  ***************************************************************      00031380
C  * THIS FUNCTION CHECKS FOR NEGATIVE ARGUMENT FOR INT FUNCTION *      00031390
C  * AND CORRECTS THE QUANTIZATION PROCEDURE.                    *      00031400
C  ***************************************************************      00031410
C                                                                        00031420
C                                                                        00031430
       REAL FUNCTION INTT(Y)                                             00031440
       X=Y                                                               00031450
       IF(X.LT.0.0) X=X-1.0                                              00031460
       INTT=AINT(X)                                                      00031470
       RETURN                                                            00031480
       END                                                               00031490
C                                                                        00031880
C                                                                        00031890
C  ***************************************************************      00031900
C  * THIS SUBROUTINE MULTIPLIES THE (3X3) MATRIX A AND THE (3X1) *      00031910
C  * VECTOR B TO OBTAIN THE (3X1) VECTOR C.                      *      00031920
C  ***************************************************************      00031930
C                                                                        00031940
C                                                                        00031950
       SUBROUTINE MULT31(A,B,C)                                          00031960
       DIMENSION A(3,3),B(3),C(3)                                        00031970
       DO 10 I=1,3                                                       00031980
       C(I)=0.0                                                          00031990
       DO 10 J=1,3                                                       00032000
10     C(I) = C(I)+A(I,J)*B(J)                                           00032010
       RETURN                                                            00032020
       END                                                               00032030
C                                                                        00031710
C                                                                        00031720
C  ***************************************************************      00031730
C  * THIS SUBROUTINE MULTIPLIES THE (3X3) MATRIX A AND THE (3X3) *      00031740
C  * MATRIX B TO OBTAIN THE (3X3) MATRIX C.                      *      00031750
C  ***************************************************************      00031760
C                                                                        00031770
C                                                                        00031780
       SUBROUTINE MULT33(A,B,C)                                          00031790
       DIMENSION A(3,3), B(3,3), C(3,3)                                  00031800
       DO 10 I=1,3                                                       00031810
```

```
      DO 10 J=1,3                                                          00031820
      C(I,J)=0.0                                                           00031830
      DO 10 K=1,3                                                          00031840
10    C(I,J) = C(I,J)+A(I,K)*B(K,J)                                        00031850
      RETURN                                                               00031860
      END                                                                  00031870
C                                                                          00032240
C                                                                          00032250
C  **************************************************************          00032260
C  * THIS SUBROUTINE GENERATES A (3X3) MATRIX TPH THAT PRODUCES *          00032270
C  * A ROTATION OF PH RADIANS ABOUT THE Z-AXIS.                 *          00032280
C  **************************************************************          00032290
C                                                                          00032300
C                                                                          00032310
      SUBROUTINE PHI(TPH,PH)                                               00032320
      DIMENSION TPH(3,3)                                                   00032330
      DO 10 I=1,3                                                          00032340
      DO 10 J=1,3                                                          00032350
10    TPH(I,J)=0.0                                                         00032360
      TPH(3,3)=1.                                                          00032370
      TPH(1,1)=COS(PH)                                                     00032380
      TPH(2,2)=TPH(1,1)                                                    00032390
      TPH(1,2)=SIN(PH)                                                     00032400
      TPH(2,1)=-TPH(1,2)                                                   00032410
      RETURN                                                               00032420
      END                                                                  00032430
C                                                                          00031500
C                                                                          00031510
C  *****************************************************************       00031520
C  * THIS SUBROUTINE GENERATES A (3X3) MATRIX TPHD THAT REPRESENTS *       00031530
C  * THE DERIVATIVE OF A MATRIX THAT REPRESENTS UNIFORM ROTATION   *       00031540
C  * ABOUT THE Z-AXIS. THE ROTATION SPEED IS W AND THE ANGLE AT    *       00031550
C  * WHICH THE DERIV. IS TAKEN IS PH.                              *       00031560
C  *****************************************************************       00031570
C                                                                          00031580
C                                                                          00031590
      SUBROUTINE PHID(TPHD,PH,W)                                           00031600
      DIMENSION TPHD(3,3)                                                  00031610
      DO 10 I=1,3                                                          00031620
      TPHD(3,I)=0.0                                                        00031630
10    TPHD(I,3)=0.0                                                        00031640
      TPHD(1,1)=-W*SIN(PH)                                                 00031650
      TPHD(2,2)=TPHD(1,1)                                                  00031660
      TPHD(1,2)=W*COS(PH)                                                  00031670
      TPHD(2,1)=-TPHD(1,2)                                                 00031680
      RETURN                                                               00031690
      END                                                                  00031700
C                                                                          00010980
C                                                                          00010990
C  ***************************************************************         00011000
C  * THIS SUBROUTINE UPDATES THE POSITION OF THE ANTENNA GIMBALS *         00011010
C  ***************************************************************         00011020
C                                                                          00011030
C                                                                          00011040
      SUBROUTINE POINT                                                     00011050
      COMMON /OUTPUT/IDUM1(3),DUM4(2),SPANG,SRANG,DUM5(3),IDUM2(4)         00011060
      COMMON /SYSDAT/TS,DUM(3),CG,SG,DUM2(9)                               00011070
      COMMON /ATDAT/DUM1(4),SALRTE,SBTRTE,DUM3(2),AL,BT,PREF,RREF,         00011080
     2              AREF,BREF                                              00011090
      DATA AK/2.0/,TAU/1.414/,PI/3.141592653/                              00011100
C                                                                          00011110
C  ************************************                                    00011120
C  * STEP 1: PRELIMINARY COMPUTATIONS *                                    00011130
C  ************************************                                    00011140
```

```
      CR=COS(-RREF)
      SR=SIN(-RREF)
      CP=COS(-PREF)
      SP=SIN(-PREF)
C
C **************************************************************
C * STEP 2: COMPUTE ANTENNA REFERENCE ROLL/PITCH ANGLES IN THE *
C *         RADAR FRAME.                                       *
C **************************************************************
      XX=CG*SP-SG*SR*CP
      YY=SG*SP+CG*SR*CP
      ZZ=CR*CP
      IF(YY.EQ.0.0.AND.ZZ.EQ.0.0) GO TO 1
      AREF=ATAN2(YY,ZZ)
      GO TO 2
    1 IF(XX.GT.0.0) AREF=-PI/2.
      IF(XX.LT.0.0) AREF=PI/2.
    2 BREF=ASIN(XX)
C
C *********************************************************
C * STEP 3: UPDATE OUTER (ALPHA) GIMBAL RATE AND POSITION *
C *********************************************************
C COMPUTE ALPHA LOOP POSITION ERROR.
      ERRA=AREF-AL
C UPDATE SMOOTHED ALPHA GIMBAL RATE ESTIMATE.
      SALRTE=SALRTE+TS*AK*ERRA
C UPDATE ALPHA GIMBAL RATE.
      ALRATE=AK*TAU*ERRA+SALRTE
C CHECK FOR ALPHA GIMBAL RATE LIMITING.
      IF(ABS(ALRATE).GT.56.) ALRATE=56.*ALRATE/ABS(ALRATE)
C UPDATE ALPHA GIMBAL POSITION.
      AL=AL+TS*ALRATE
C
C ********************************************************
C * STEP 4: UPDATE INNER (BETA) GIMBAL RATE AND POSITION *
C ********************************************************
C COMPUTE BETA LOOP POSITION ERROR.
      ERRB=BREF-BT
C UPDATE SMOOTHED BETA GIMBAL RATE ESTIMATE.
      SBTRTE=SBTRTE+TS*AK*ERRB
C UPDATE BETA GIMBAL RATE.
      BTRATE=AK*TAU*ERRB+SBTRTE
C CHECK FOR BETA GIMBAL RATE LIMITING.
      IF(ABS(BTRATE).GT.56.) BTRATE=56.*BTRATE/ABS(BTRATE)
C UPDATE BETA GIMBAL POSITION.
      BT=BT+TS*BTRATE
C
C *******************************************
C * STEP 5 :  ANTENNA IN OBSCURATION REGION" *
C *******************************************
C     CALL SCNWRN
C
C ***************************************************************
C * STEP 6: COMPUTE ANTENNA ROLL/PITCH ANGLES IN THE BODY FRAME *
C ***************************************************************
      CA=COS(AL)
      SA=SIN(AL)
      CB=COS(BT)
      SB=SIN(BT)
      XX=CA*SB+SG*SA*CB
      YY=-SG*SB+CG*SA*CB
      ZZ=CA*CB
      IF(YY.EQ.0.0.AND.ZZ.EQ.0.0) GO TO 3
      SRANG=-57.29576*ATAN2(YY,ZZ)
      GO TO 4
    3 IF(XX.GT.0.0) SRANG=+90.0
      IF(XX.LT.0.0) SRANG=-90.0
    4 SPANG=-57.29576*ASIN(XX)
C RESOLVE POSSIBLE ANGLE AMBIGUITIES, VIZ., -90.<SPANG<90. AND
C -180.<SRANG<180.
      IF(SPANG.LE.90.) GO TO 10
      SPANG=-(180.-ABS(SPANG))*(SPANG/ABS(SPANG))
      SRANG=(180.-ABS(SRANG))*(SRANG/ABS(SRANG))
   10 RETURN
      END
C
C
C     ***********************************************************
C     * THIS SUBROUTINE COMPUTES TARGET C.G. POSITION AND VELOCITY *
C     * WRT ANTENNA LOS COORDINATES AND INDIVIDUAL SCATTERER POSI- *
C     * TIONS AND VELOCITIES WRT ANTENNA LOS COORDINATES.          *
C     ***********************************************************
C
C
      SUBROUTINE PVTRAN
       COMMON /TEST1/RA(3)
      COMMON /CNTL/IPWR,IMODE
      COMMON /INPUT/ERT(3),EVT(3),DUM(21)
      COMMON /OUTPUT/MSWF,MTF,MSF,DUMO(7),IDUMO(4)
      COMMON /ICNTL/IDUM6(9),MTP,IDUM7(3),MTKINT
      COMMON /SYSDAT/TSAM,DR(3),DUM2(11)
      COMMON /TGTDAT/NT,RAU(3,100),RANGE(100),RADVEL(100),RO(3),
     2              ROU(3),CGRNGE,CGVEL
      COMMON /SATDAT/RADAR(3),N20,RT(70,3),SIG(70),ROLD,ICLOSE,ICLOLD
      COMMON /XFORMS/TLB(3,3),TLBD(3,3),TLT(3,3),TLTD(3,3)
      COMMON /TARGET/ ITARG,SRCS
      DIMENSION ROR(3),ROD(3),V1(3),RL(3),RAD(3),RLD(3),XRT(3)
C     **********************************************************
C     * STEP 1: COMPUTE TARGET C.G. POSITION IN ANTENNA LOS FRAME *
C     **********************************************************
C
C     STEP 1-1: ADD RADAR OFFSET IN ORBITER BODY FRAME.
      DO 5 I=1,3
    5 ROR(I)=ERT(I)-DR(I)
C
C     STEP 1-2: TRANSFORM TARGET C.G. POSITION FROM BODY FRAME TO
C               ANTENNA LOS FRAME.
      CALL MULT31(TLB,ROR,RO)
C
C     STEP 1-3: COMPUTE RANGE OF TARGET C.G. WRT RADAR.
      CGRNGE=SQRT(RO(1)*RO(1)+RO(2)*RO(2)+RO(3)*RO(3))
C
C     STEP 1-4: COMPUTE UNIT VECTOR IN DIRECTION OF TARGET C.G. WRT
C               ANTENNA LOS FRAME.
      DO 10 I=1,3
   10 ROU(I)=RO(I)/CGRNGE
C
C     ************************************************************
C     * STEP 2: COMPUTE TARGET C.G. RADIAL VELOCITY WRT ANTENNA LOS *
C     *         FRAME (OR RADAR).                                   *
C     ************************************************************
C
C     STEP 2-1: COMPUTE TARGET C.G. VELOCITY COMPONENTS WRT ANTENNA
C               LOS FRAME.
      CALL MULT31(TLBD,ROR,V1)
      CALL MULT31(TLB,EVT,ROD)
      DO 15 I=1,3
   15 ROD(I)=ROD(I)+V1(I)
```

```
C
C  STEP 2-2: COMPUTE TARGET C.G. RADIAL VELOCITY WRT ANTENNA LOS.
      CGVEL=0.0
      DO 20 I=1,3
   20 CGVEL=CGVEL+ROD(I)*ROU(I)
C
C  ***********************************************************
C  * STEP 3: COMPUTE TARGET SCATTERING CHARACTERISTICS -- = OF *
C  *         ILLUMINATED POINTS, THE POINT LOCATIONS, AND THE   *
C  *         RCS FOR EACH POINT.                                *
C  ***********************************************************
C
C  STEP 3-1: IF IN ACTIVE MODE, SEARCH MODE, OR TRACKER INITIALIZATION
C            -- ASSUME SINGLE SCATTERER LOCATED AT TARGET FRAME ORIGIN.
C
C ITARG=0 POINT TARGET
C ITARG=1 SPAS
C ITARG=2 SMM
      IF(ITARG.EQ.0) GO TO 24
C  CHECK CONDITION.
      IF(IMODE.NE.1.AND.MTKINT.NE.0.AND.MTP.NE.0) GO TO 30
C  IF ABOVE CONDITION TRUE -- THEN SET PARAMETERS AS FOLLOWS AND DO
C  NOT CALL TARGET MODEL.
   24 NT=1
      SIG(1)=SRCS
      DO 25 I=1,3
   25 RT(1,I)=0.0
C
C  STEP 3-2: COMPUTE LOCATION OF RADAR IN TARGET FRAME.
   30 DO 35 I=1,3
      RADAR(I)=0.0
      DO 35 J=1,3
   35 RADAR(I)=RADAR(I)-TLT(J,I)*RO(J)
      IF(ITARG.EQ.0)GO TO 40
C
C  STEP 3-3: COMPUTE TARGET SCATTERING CHARACTERISTICS.
     IF(ITARG.EQ.2)CALL SMM
      IF(ITARG.EQ.1)CALL SPAS
      NT=N20
C
   40 DO 70 K=1,NT
C
C  *************************************************************
C  * STEP 4: COMPUTE KTH SCATTERER POSITION, RANGE, AND DIRECTION *
C  *         VECTOR WRT ANTENNA LOS FRAME (OR RADAR).              *
C  *************************************************************
C
C  STEP 4-1: COMPUTE KTH SCATTERER POSITION WRT ANTENNA LOS FRAME.
      DO 45 J=1,3
      RL(J)=0.0
      DO 45 I=1,3
   45 RL(J)=RL(J)+TLT(J,I)*RT(K,I)
      DO 50 I=1,3
   50 RA(I)=RO(I)+RL(I)
C
C  STEP 4-2: COMPUTE RANGE OF KTH SCATTERER WRT RADAR.
      RANGE(K)=SQRT(RA(1)*RA(1)+ RA(2)*RA(2)+RA(3)*RA(3))
C
C  STEP 4-3: COMPUTE UNIT VECTOR IN DIRECTION OF KTH SCATTERER WRT
C            ANTENNA LOS FRAME.
      DO 55 I=1,3
   55 RAU(I,K)=RA(I)/RANGE(K)
C
C  **********************************************************
```

```
C  * STEP 5: COMPUTE KTH SCATTERER RADIAL VELOCITY WRT RADAR *
C  **********************************************************
C
C  STEP 5-1: COMPUTE KTH SCATTERER VELOCITY COMPONENTS WRT ANTENNA
C            LOS FRAME.
       DO 58 I=1,3
   58  XRT(I)=RT(K,I)
       CALL MULT31(TLTD,XRT,RLD)
       DO 60 I=1,3
   60  RAD(I)=ROD(I)+RLD(I)
C
C  STEP 5-2: COMPUTE KTH SCATTERER RADIAL VELOCITY WRT TO RADAR.
       RADVEL(K)=0.0
       DO 65 I=1,3
   65  RADVEL(K)=RADVEL(K)+RAD(I)*RAU(I,K)
   70  CONTINUE
C
C  NOTE: DEBUGGING PRINT STATEMENTS.
C      WRITE(6,900) RO(1),RO(2),RO(3),CGRNGE,CGVEL
C      WRITE(6,901) RAU(1,1),RAU(2,1),RAU(3,1),RANGE(1),RADVEL(1)
C      WRITE(6,902)
C      WRITE(6,903)(I,(RT(I,J),J=1,3),SIG(I),I=1,N20)
  900  FORMAT(//' RO1,RO2,RO3,CGR,CGV =',5F10.2)
  901  FORMAT(' RAU1,RAU2,RAU3,R,V =',5F10.2)
  902  FORMAT(' SPAS RCS DATA:',/,
     1 /,9X,'I',4X,'R(I,1)',4X,'R(I,2)',4X,'R(I,3)',9X,'SIG(I)',/)
  903  FORMAT(I10,3F10.2,F15.1)
       RETURN
       END
C
C
C    *************************************************************
C    * THIS FUNCTION GENERATES A RANDOM NUMBER FROM A UNIFORM Ø0,1Ø *
C    * DISTRIBUTION.                                              *
C    *************************************************************
       FUNCTION RNDU(IRAN)
       DATA MU/524287/,IETA/997/
       IF(IRAN.EQ.0)  GO TO 10
       IRAN=IETA*IRAN
       IKEEP=IRAN/MU
       IRAN=IRAN-IKEEP*MU
       XRAN=IRAN
       XRAN=XRAN/MU
       RNDU=XRAN
   10  RETURN
       END
C
C
C    ***************************************************************
C    * THIS SUBROUTINE COMPUTES THE RADAR SIGNAL STRENGTH AND UPDATES *
C    * THE AGC SETTING.                                               *
C    ***************************************************************
C
C
       SUBROUTINE RSS
       COMMON /CNTL/IPWR,IMODE,IDUM1(7),DUM1(3)
       COMMON /ICNTL/IDUM2(14),MRNG,MSAM,IDUM6(11)
       COMMON /OUTPUT/IDUM7(3),DUM3(6),SRSS,IDUM4(4)
       COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD
       DIMENSION PS(10,2),QNV(2),A1(2)
       DATA PS/9*4.,2.,5*4.,2.,4.,8.,8.,16./
       DATA QNV/.00067,.011/,A1/.0321,.51/
C ***************************************************
C SUBROUITNE RSS HAS BEEN UPDATED TO CORRESPOND TO THE
```

```
C DERIVATION OF AGCERR PRESENTED IN THE FINAL REPORT ON
C KUBAND COMPUTER SIMULATION. M. MEYER FEB 17, 1986
C ******************************************************
C
C  *****************************
C  * STEP 1: UPDATE SYSTEM AGC *
C  *****************************
C
C  STEP 1-1: COMPUTE AGC ERROR AND CHECK LIMITS.
C --------UPDATED FEB 17, 1986------------------------------------
      AGCERR=A1(MSAM)*4.*PS(MRNG,IMODE)/(AGCO*(SNRDT+1.0)+QNV(MSAM))
      IF(AGCERR.GT.10.) AGCERR=10.0
      IF(AGCERR.LT.0.1) AGCERR=0.1
C
C  STEP 1-2: COMPUTE NEW AGC VALUE AND CHECK LIMITS.
      AGCO=AGCERR*AGCO
C --------UPDATED FEB 17, 1986-----------------------------------
      IF(AGCO.GT.0.25) AGCO=0.25
      AGCODB=10.*ALOG10(AGCO)
C
C  **********************************************
C  * STEP 2: UPDATE RADAR SIGNAL STRENGTH VALUE *
C  **********************************************
      IF(AGCO.LT.1.0E-15) AGCO=1.0E-15
      SRSS=1./AGCO
C --------UPDATED FEB 17, 1986------------------------------------
      SRSS=10.*ALOG10(SRSS)-6.0
      RETURN
      END
C
C
C  ************************************************************
C  * THIS SUBROUTINE UPDATES RANGE AND RANGE RATE ESTIMATES. *
C  ************************************************************
C
C
      SUBROUTINE RTRACK
      REAL INTT,IRDISC,IRNG,IRDOT
      COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)
      COMMON /OUTPUT/IDUM0(3),SRNG,SRDOT,DUM2(5),IDUM(4)
      COMMON /ICNTL/I1DUM(14),MRNG,MSAM,MPRF,IDUM1(10),MPFOLD
      COMMON /SYSDAT/TSAM,DUMS(14)
      COMMON /RTDAT/IRDOT,IRNG,RBIAS,VEST(4),MDF(5)
      COMMON /DSCRM/DUM(2),RDSC,VDISC,RRTE,ODISC,DUM3(3)
      DIMENSION RT1(10,2),RT2(10,2),TDC(3),RGBIAS(2)
      DATA RT1/9*0.125,0.25,4*0.125,2.,1.,2.,2*0.5,0.25/,RT2/9*0.5,
     2         4.0,4*0.5,8.,8.,4*16./
     DATA TDC/0.05122118,0.1195161,0.2561557/
      DATA RGBIAS/32.3,94.7                        /
C
C  *********************************************
C  * STEP 1: UPDATE ROUGH RANGE RATE ESTIMATE *
C  *********************************************
C
C  STEP 1-1: INTEGERIZE RANGE DISCRIMINANT AND CHECK FOR SATURATION.
      RDISC=5.333333*RDSC
      TCON=TSAM/TDC(MPRF)
      IRDISC=INTT(RDISC*TCON+0.5)/TCON
      IF(IRDISC.GT.127.) IRDISC=127.
      IF(IRDISC.LT.-128.) IRDISC=-128.
C
C  STEP 1-2: COMPUTE ROUGH RANGE RATE PREDICTION FROM ALPHA-BETA
C            TRACKING EQUATIONS.
C  DEFINITION:  RT1(MRNG,IMODE) CORRESPONDS TO BETA IN ALPHA-BETA TRACK.
```

```
      RR1=IRDISC*RT1(MRNG,IMODE)
      IRDOT=IRDOT+INTT(RR1+0.5)
C
C  ********************************
C  * STEP 2: UPDATE RANGE ESTIMATE *
C  ********************************
C
C  STEP 2-1: UPDATE RANGE ESTIMATE USING ALPHA-BETA TRACKER EQUATIONS.
C  DEFINITION: RT2 CORRESPONDS TO ALPHA IN ALPHA-BETA TRACKER.
      R1=IRDISC*RT2(MRNG,IMODE)
      IRNG=IRNG+IRDOT+INTT(R1+0.5)
C
C  STEP 2-2: CONVERT RANGE ESTIMATE (IRNG) TO FEET USING THE FACT THAT
C            THE LSB OF IRNG REPRESENTS 5/16 FEET.
      RNG=0.3125*IRNG
C
C  STEP 2-3: ADD FIXED BIAS TO FINAL RANGE ESTIMATE.
      SRNG=RNG+RGBIAS(MSAM)
C
C  FORCE BREAK TRACK IF RANGE LESS THAN 100 FT
C
      IF(SRNG.LT.100.)CALL SYSINT
C
      RETURN
      END
C
C
C  ********************************************************
C  * THIS SUBROUTINE DETERMINES WHETHER THE SIGNAL PLUS NOISE *
C  * IS SATURATING THE A/D -- IF SO,THEN THE SNR AT DOPPLER   *
C  * FILTER OUTPUT IS LIMITED TO THE VALUE THAT JUST SATUR-   *
C  * ATES THE A/D.                                            *
C  ********************************************************
C
C
      SUBROUTINE SATNSE(SNF)
      COMMON /CNTL/IPWR,IMODE
      COMMON /ICNTL/IDUM(14),MRNG
      COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD
      DIMENSION PS(10,2)
C
C ------ PS VALUES WERE UPDATED FEB 17,1986 BY M. MEYER----------
C
      DATA PS/9*4.0,2.,5*4.,2.,4.,8.,8.,16./
      SNF=1.
      X=AGCO*(SNRDT/(4.*PS(MRNG,IMODE))+1.0)
C  **************************************************
C  X=12.25/X WAS REPLACED BY X=6.25/X TO MORE ACCURATELY
C  REFLECT A/D SATURATION BY M. MEYER FEB 17, 1986
C  **************************************************
      X=6.25/X
      IF(X.GT.1) RETURN
      SNF=X
      RETURN
      END
C
C
C  ****************************************************************
C  * THIS SUBROUTINE  CYCLES THRU THE LOGIC FOR ANY SCAN GENERATION. *
C  ****************************************************************
C
C
      SUBROUTINE SCAN
      COMMON /CNTL/IDUM(4),ISRCHC,ISRCHG,IDUMC(3),EDRNG,DUMC(2)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      COMMON /OUTPUT/MSWF,MTF,MSF,DUM1(7),IDUM2(4)
      COMMON /ICNTL/IDUM3(6),KSNCLK,IDUM4(2),MTP,IDUM5(17),MSWTCH,
     2              KSN,IAROLD,ITROLD
      COMMON /SYSDAT/TSAM,DUMS(14)
      COMMON /TGTDAT/NT,DUM2(503),ROU(3),DUM3(2)
      COMMON /ATDAT/DUM4(8),AL,BT,DUM5(2),AREF,BREF
      DIMENSION TIMINT(31),ANGINT(31),RSW(10),TSW(10)
      DATA TIMINT/.7,1.4,1.9,2.6,3.4,4.3,5.1,6.,7.,8.,9.1,10.4,11.8,
     1 13.3,14.9,16.9,18.9,21.1,23.4,25.9,28.6,31.5,33.5,36.6,39.8,
     2 43.2,46.8,50.5,54.3,58.4,60.0/
      DATA ANGINT/0.,.7,1.5,2.,2.7,3.6,4.4,5.2,6.1,7.,7.9,8.8,9.8,10.9,
     1 11.9,13.0,14.2,15.3,16.5,17.6,18.8,19.9,21.1,22.2,23.4,24.5,
     2 25.6,26.7,27.8,28.9,30./
      DATA TSW/60.0,54.3,43.2,33.5,28.6,21.1,14.9,11.8,8.0,6.0/,
     2 RSW/48609.2,55900.6,62584.3,71698.6,91142.5,151903.8,
     3 243046.0,394949.8,881041.8,1822845.0/
      PII=180./3.141592653
C
C  *******************************************************************
C  * STEP 1: DETERMINE WHETHER TO PERFORM SCAN INITIALIZATION(MSF=0) *
C  *          OR SCAN UPDATE(MSF=1).                                 *
C  *******************************************************************
      IF(MSF.EQ.1) GO TO 15
C
C  ***************************************
C  * STEP 2: PERFORM SCAN INITIALIZATION *
C  ***************************************
C  INITIALIZE ALL FLAGS.
      MSF=1
C  INITIALIZE RING MONITORS.
      IAROLD=0
      ITROLD=10
C  INITIALIZE SCAN CLOCK.
      KSNCLK=0
C  INITIALIZE SCAN TIME PARAMETER.
      KSN=0
C
C  DETERMINE SWITCH POINT PARAMETER.
      DO 5 I=1,10
      IF(EDRNG.LT.RSW(I)) GO TO 10
    5 CONTINUE
   10 MSWTCH=I
C
C  ******************************
C  * STEP 3: UPDATE SCAN CLOCKS *
C  ******************************
C
C  STEP 3-1: UPDATE SCAN CLOCK (TRACKS TOTAL ELAPSED TIME FROM SCAN
C            INITIATION).
   15 KSNCLK=KSNCLK+1
      T=FLOAT(KSNCLK)*TSAM
C
C  STEP 3-2: UPDATE SCAN TIME PARAMETER (USED TO DETERMINE BORESIGHT
C            POSITION IN SCAN PATTERN).
      IF(T.LE.TSW(MSWTCH)) KSN=KSN+1
      IF(T.GT.TSW(MSWTCH)) KSN=KSN-1
      TSN=FLOAT(KSN)*TSAM
C
C  ***********************************************************
C  * STEP 4: DETERMINE ANTENNA POSITION TO NEAREST SCAN RING *
C  ***********************************************************
      DO 20 I=1,31
      IF(TSN.LT.TIMINT(I)) GO TO 25
   20 CONTINUE
   25 IARNG=I
C
C  ******************************************************
C  * STEP 5: DETERMINE TARGET POSITION IN SCAN PATTERN (SCAN *
C  *         RING NUMBER FOR TARGET)                        *
C  ******************************************************
C
C  STEP 5-1: DETERMINE TARGET POSITION EXACTLY.
      ALOLD=AL
      BTOLD=BT
      AL=AREF
      BT=BREF
      CALL TRNSFM
      CALL PVTRAN
      AL=ALOLD
      BT=BTOLD
C
C  STEP 5-2: DETERMINE TARGET SCAN RING NUMBER.
C
C  DETERMINE TARGET ANGLE OFF SCAN DESIGNATES (DEGREES).
CCCCCCCCCCCCCCCCCCCC MOD MAR 24 1983 CCCCCCCCCCCCCCCCCCCCC
      CGANG=ACOS(-ROU(3))*PII
C
C  DETERMINE TARGET SCAN RING NUMBER.
      DO 30 I=1,31
      IF(CGANG.LT.ANGINT(I)) GO TO 35
   30 CONTINUE
   35 ITRNG=I
      IF(CGANG.GT.30.) ITRNG=32
C
C  ***************************************************
C  * STEP 6: DETERMINE IF A DETECTION SHOULD BE ATTEMPTED *
C  ***************************************************
C
C  STEP 6-1: CHECK CONDITION.
      IF(IARNG.EQ.ITRNG.AND.IAROLD.NE.ITROLD) CALL DETECT
C
C  STEP 6-2: UPDATE RING NUMBER MONITOR.
      IAROLD=IARNG
      ITROLD=ITRNG
C
C  *********************************************
C  * STEP 7: CHECK FOR SCAN TERMINATION CONDITIONS *
C  *********************************************
C
C  STEP 7-1: CHECK ALL POSSIBLE TERMINATION CONDITIONS.
C
C  CONDITION = 1: T > 60. SECONDS"
      IF(T.GE.60.) GO TO 40
C
C  CONDITION = 2: NEXT SCAN TIME PARAMETER < 0. "
      ITEMP=KSN-1
      IF(ITEMP.LT.0) GO TO 40
C
C  CONDITION = 3: DETECT A TARGET"
      IF(MTP.EQ.0) RETURN
C
C  STEP 7-2: PERFORM SCAN TERMINATION STEPS -- IF TERMINATION COND
C            ITION OBTAINED.
   40 MSF=0
      KSNCLK=0
      KSN=0
      ISRCHG=0
      ISRCHC=0
      RETURN                                                            00014030
      END                                                               00014040
C                                                                       00011900
C                                                                       00011910
C  ************************************************************        00011920
C  * THIS SUBROUTINE DETERMINES WHETHER  THE ANTENNA IS IN THE OB- *   00011930
C  * SCURATION ZONE AND SETS THE SCAN WARNING FLAG APPROPRIATELY.  *   00011940
C  ************************************************************        00011950
C                                                                       00011960
C                                                                       00011970
      SUBROUTINE SCNWRN                                                 00011980
      COMMON /OUTPUT/MSWF,IDUMO(2),DUMO(7),IDUMO1(4)                    00011990
      COMMON /ATDAT/DUM(8),A,B,DUMA(4)                                  00012000
      DIMENSION ICLEAR(36,72)                                           00012010
      DATA ICLEAR /17*1,13*0,6*1,18*1,12*0,6*1,18*1,12*0,6*1,          00012020
     1 18*1,12*0,6*1,19*1,11*0,6*1,19*1,11*0,6*1,19*1,11*0,6*1,         00012030
     2 19*1,11*0,6*1,19*1,11*0,6*1,19*1,11*0,6*1,20*1,10*0,6*1,         00012040
     3 20*1,10*0,6*1,20*1,10*0,6*1,20*1,10*0,6*1,20*1,10*0,             00012050
     4 6*1,20*1,10*0,6*1,19*1,11*0,6*1,18*1,12*0,6*1,17*1,13*0,         00012060
     5 6*1,16*1,14*0,6*1,15*1,15*0,6*1,14*1,16*0,6*1,14*1,16*0,         00012070
     6 6*1,13*1,17*0,6*1,12*1,18*0,6*1,11*1,19*0,6*1,10*1,20*0,6*1,     00012080
     7 9*1,21*0,6*1,9*1,21*0,6*1,8*1,22*0,6*1,4*1,0,3*1,22*0,6*1,       00012090
     8 4*1,26*0,6*1,4*1,26*0,6*1,4*1,26*0,6*1,4*1,26*0,6*1,4*1,26*0,6*1,00012100
     9 4*1,26*0,6*1,4*1,26*0,6*1,4*1,26*0,6*1,4*1,26*0,6*1,4*1,26*0,6*1,00012110
     A 4*1,26*0,6*1,4*1,26*0,6*1,4*1,26*0,6*1,4*1,26*0,6*1,4*1,26*0,6*1,00012120
     B 4*1,26*0,6*1,4*1,26*0,6*1,4*1,26*0,6*1,4*1,7*0,2*1,17*0,6*1,     00012130
     C 4*1,7*0,2*1,17*0,6*1,4*1,6*0,3*1,17*0,6*1,4*1,5*0,4*1,17*0,6*1,  00012140
     D 4*1,5*0,6*1,15*0,6*1,4*1,0,12*1,13*0,6*1,19*1,11*0,6*1,          00012150
     E 21*1,9*0,6*1,24*1,6*0,6*1,26*1,4*0,                              00012160
     F 6*1,27*1,3*0,6*1,28*1,2*0,6*1,29*1,0,6*1,29*1,0,6*1,28*1,        00012170
     G 2*0,6*1,27*06,3*0,6*1,26*1,4*0,6*1,25*1,5*0,6*1,23*1,7*0,6*1,    00012180
     H 23*1,7*0,6*1,22*1,8*0,6*1,19*1,11*0,6*1,18*1,12*0,6*1/           00012190
C                                                                       00012200
      ALPHA=A*57.3                                                          00012210
      BETA=B*57.3                                                           00012220
      IF(ABS(BETA).LE.90.) GO TO 1                                      00012230
      BETA=-(180-ABS(B))*(B/ABS(B))                                     00012240
      ALPHA=(180-ABS(A))*(A/ABS(A))                                     00012250
    1 CONTINUE                                                          00012260
      IA=INT((ALPHA+180.)/5.+1.)                                        00012270
      IB=INT((90-BETA)/5.+1.)                                           00012280
      MSWF=ICLEAR(IB,IA)                                                00012290
      RETURN                                                            00012300
      END                                                               00012310
C                                                                       00005010
C                                                                       00005020
C  ********************************************************            00005030
C  * THIS SUBROUTINE COMPUTES THE RESPONSE TO ALL DISPLAYS AND *       00005040
C  * CONTROLS WHEN THE RADAR IS IN ANY OF THE SEARCH MODES.    *       00005050
C  ********************************************************            00005060
C                                                                       00005070
C                                                                       00005080
      SUBROUTINE SEARCH                                                 00005090
      COMMON /CNTL/IDUM(3),IASM,ISRCHC,ISRCHG,IAZS,IELS,ISLR,EDRNG,     00005100
     2             EDPA,EDRA                                            00005110
      COMMON /OUTPUT/MSWF,MTF,MSF,SRNG,SRDOT,SPANG,SRANG,SPRTE,         00005120
     2               SRRTE,SRSS,IDUM2(4)                                00005130
      COMMON /ICNTL/IOLDPW,IOLDMD,IOLDSM,ISHOLD,KMSCLK,KWMUP,KSNCLK,    00005140
     2             KSNMAX,KACCLK,MTP,MZ1,MZ0,MSS,MTKINT,MRNG,MSAM,MPRF,00005150
     3             IDUM1(10)                                            00005160
      COMMON /SYSDAT/TS,DUMS(14)                                        00005170
      COMMON /ATDAT/DUM2(10),PREF,RREF,DUMA(2)                          00005180
      DIMENSION SLWRTE(2)                                               00005190
      DATA SLWRTE/6.9814E-3,3.4907E-1/                                  00005200
```

```
C
C  ************************************
C  * DETERMINE ANTENNA STEERING MODE. *
C  ************************************
       GO TO (10,20,30,40),IASM
C
C
C  ***********************************************************************
C  *************** GPC-ACQ SEARCH AND ACQUISITION MODE. ***************
C  ***********************************************************************
C
C
C  ************************************************************
C  * STEP 1: DETERMINE WHETHER SEQUENCING THRU POINT OR SCAN *
C  ************************************************************
   10  IF(MSF.EQ.1) GO TO 14
       IF(MZ1.EQ.1.AND.ISRCHG.EQ.1) GO TO 14
C
C  *********************************************
C  * STEP 2: PERFORM GIMBAL POINTING SEQUENCE *
C  *********************************************
C
C  STEP 2-1: UPDATE ROLL/PITCH REFERENCES
       IF(ISHOLD.EQ.1.AND. ISRCHG.EQ.1) GO TO 12
       RREF=EDRA
       PREF=EDPA
   12  ISHOLD=ISRCHG
C
C  STEP 2-2: UPDATE POSITION OF GIMBALS.
       CALL POINT
C
C  STEP 2-3: DETERMINE WHETHER BORESIGHT IN ZONE I AND/OR ZONE O AND
C            TAKE APPROPRIATE ACTION.
       CALL ZONECK
C  IF NOT IN ZONE O, THEN DETECTION IS NOT ALLOWED.
       IF(MZ0.EQ.0) RETURN
C
C  ********************************************************
C  * STEP 3: CHECK FOR TARGET DETECTION -- IF IN ZONE O *
C  ********************************************************
C
       CALL DETECT
       RETURN
C
C  ********************************
C  * STEP 4: PERFORM SCAN SEQUENCE *
C  ********************************
   14  CALL SCAN
       RETURN
C
C
C  ***********************************************************************
C  *************** GPC-DES SEARCH AND ACQUISITION MODE ****************
C  ***********************************************************************
C
C  *********************************************
C  * STEP1 : PERFORM GIMBAL POINTING SEQUENCE *
C  *********************************************
C
C  STEP 1-1: UPDATE ROLL/PITCH REFERENCE ANGLES.
   20  PREF=EDPA
       RREF=EDRA
C
C  STEP 1-2: UPDATE POSITION OF GIMBALS.
```

```
      CALL POINT                                                        00005850
C                                                                       00005860
C  STEP I-3: DETERMINE WHETHER BORESIGHT IN ZONE 1 AND/OR ZONE O AND    00005870
C            TAKE APPROPRIATE ACTIN.                                    00005880
      CALL ZONECK                                                       00005890
C  IF BORESIGHT NOT IN ZONE O, THEN TARGET DETECTION NOT ALLOWED.       00005900
      IF(MZ0.EQ.0) RETURN                                               00005910
C                                                                       00005920
C  ********************************************************             00005930
C  * STEP 2: CHECK FOR TARGET DETECTION --- IF IN ZONE O. *             00005940
C  ********************************************************             00005950
C                                                                       00005960
      CALL DETECT                                                       00005970
      RETURN                                                            00005980
C                                                                       00005990
C                                                                       00006000
C  *********************************************************************00006010
C  ***************** AUTO SEARCH AND ACQUISITION MODE *****************00006020
C  *********************************************************************00006030
C                                                                       00006040
C                                                                       00006050
C  ***********************************************************          00006060
C  * STEP 1: DETERMINE WHETHER SEQUENCING THRU POINT OR SCAN *          00006070
C  ***********************************************************          00006080
   30  IF(ISRCHC.EQ.1) GO TO 32                                         00006090
C                                                                       00006100
C  ********************************************                         00006110
C  * STEP 2: PERFORM GIMBAL POINTING SEQUENCE *                         00006120
C  ********************************************                         00006130
C                                                                       00006140
C  STEP 2-1: UPDATE ROLL/PITCH REFERENCE ANGLES.                        00006150
       PREF=PREF+FLOAT(IELS)*SLWRTE(ISLR+1)*TS                          00006160
       RREF=RREF+FLOAT(IAZS)*SLWRTE(ISLR+1)*TS                          00006170
C                                                                       00006180
C  STEP 2-2: UPDATE POSITION OF GIMBALS.                                00006190
       CALL POINT                                                       00006200
C                                                                       00006210
C  STEP 2-3: DETERMINE SLEW RATE AND TAKE APPROPRIATE ACTION.           00006220
C             IF SLEW RATE IS GREATER THAN 0.4 DEG/SEC, THEN TARGET DET-00006230
       IF(ISLR.GT.0) RETURN                                             00006240
C                                                                       00006250
C  ****************************************************************     00006260
C  * STEP 3: CHECK FOR TARGET DETECTION --- IF SLEW RATE <0.4 DEG *     00006270
C  *         PER SECOND.                                          *     00006280
C  ****************************************************************     00006290
       CALL DETECT                                                      00006300
       RETURN                                                           00006310
C                                                                       00006320
C  **********************************                                   00006330
C  * STEP 4: PERFORM SCAN SEQUENCE *                                    00006340
C  **********************************                                   00006350
   32  CALL SCAN                                                        00006360
       RETURN                                                           00006370
C                                                                       00006380
C                                                                       00006390
C  *********************************************************************00006400
C  **************** MANUAL SEARCH AND ACQUISITION MODE ****************00006410
C  *********************************************************************00006420
C                                                                       00006430
C                                                                       00006440
C  ************************************                                 00006450
C  * STEP 1: UPDATE ANTENNA POSITION *                                  00006460
C  ************************************                                 00006470
C                                                                       00006480
```

```
C  STEP 1-1: UPDATE ROLL/PITCH REFERENCE ANGLES.
   40  PREF=PREF+FLOAT(IELS)*SLWRTE(ISLR+1)*TS
       RREF=RREF+FLOAT(IAZS)*SLWRTE(ISLR+1)*TS
C
C  STEP 1-2: UPDATE POSITION OF GIMBALS.
       CALL POINT
C
C  STEP 1-3: DETERMINE SLEW RATE AND TAKE APPROPRIATE ACTION.
C              IF SLEW RATE IS GREATER THAN 0.4 DEG/SEC, THEN TARGET DET-
C              ECTION IS NOT ALLOWED.
       IF(ISLR.GT.0) RETURN
C
C  *************************************************************
C  * STEP 2: CHECK FOR TARGET DETECTION --- IF SLEW RATE <0.4 DEG *
C  *         PER SECOND.                                        *
C  *************************************************************
       CALL DETECT
       RETURN
       END
C
C
C  ****************************************************************
C  * THIS SUBROUTINE GENERATES THE NOISE-FREE ANGLE, RANGE, VELOCITY *
C  * AND ON-TARGET DISCRIMINANT COMPONENTS.                          *
C  ****************************************************************
C
C
       SUBROUTINE SIGNAL
       REAL IRDOT,IRNG
       COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)
       COMMON /OUTPUT/I1DUM(3),SRNG,DUM1(6),IDUM2(4)
       COMMON /ICNTL/IDUM5(13),MTKINT,MRNG,MSAM,MPRF,MBKTRK,MBTSUM,
     2             MBT(8)
       COMMON /TGTDAT/NT,RAU(3,100),RANGE(100),RADVEL(100),RO(3),
     2               ROU(3),CGRNGE,CGVEL
       COMMON /SATDAT/RADAR(3),N20,RT(70,3),SIG(70)
       COMMON /RTDAT/IRDOT,IRNG,DUM2(5),MDF(5)
       COMMON /SIGDAT/SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE,DF1,DF5,
     2               DF2,DF4,SIGBAR
       COMMON /XFORMS/TLB(3,3),TLBD(3,3),TLT(3,3),TLTD(3,3)
       COMMON /SUDIPH/ X,Y,Z,PAZ,PEL
       COMPLEX CSUM,CDIFAZ,CDIFEL,CEARLY,CLATE,CDF1,CDF5,CDF2,CDF4,
     2         DFWTS,PHASE,PHASE1,DOPFIL
       DIMENSION CTP(10,2),DFWTS(5,100),ALAM(5),ALAMD(3),NFREQ(2)
       DATA CTP/9*.03318,9.799E-4,4*.03318,1.9599E-3,9.8E-4,4.9E-4,
     2          2*2.45E-4,1.225E-4/
       DATA NFREQ/1,5/,ALAM/177.3733,176.0447,178.7149,176.7089,
     2      178.0393/,ALAMD/1.272461E-2,2.969089E-2,3.309023E-1/
       REAL LATE
       COMPLEX DAZ,DEL
       DATA ILOOP/1/
C
C  ***************************************************************
C
C  MODIFIED JAN 10 1986 BY M. MEYER
C  MODIFICATIONS TO SUBROUTINE SIGNAL INCLUDE
C  CALCULATION OF THE AZIMUTH AND ELEVATION ANGLES
C  USE OF MEASURED ANTENNA PATTERNS INSTEAD
C  OF FUNCIONS SPAT AND DPAT AND A
C  FACTOR IN THE DIFFERENCE CHANNELS SIGNAL
C  WHICH ACCOUNTS FOR THE FINITE WIDTH PHASE
C  TRANSITION IN THE REAL PHASE PATTERNS.
C
C  ***************************************************************
```

```
C
C  *****************************************************************
C  * STEP 0: READ IN ANTENNA PATTERNTERNS AND SET PHASE BALANCE    *
C  *****************************************************************
C
      IF (ILOOP.NE.1) GO TO 11
          CALL READPAT
          PBAL=0.
          ILOOP=0
11    CONTINUE
C
C  *****************************************************************
C  * STEP 1: PRELIMINARY COMPUTATIONS AND PARAMETER INITIALIZATION *
C  *****************************************************************
C
C  STEP 1-1: INITIALIZE DISCRIMINANT COMPONENTS (NOTE: THESE ARE THE
C            COMPONENT SIGNALS AFTER SQUARE-LAW DETECTION).
      SPAZ=0.0
      SMAZ=0.0
      SPEL=0.0
      SMEL=0.0
      EARLY=0.0
      LATE=0.0
      DF1=0.0
      DF5=0.0
      DF2=0.0
      DF4=0.0
      SIGBAR=0.0
C
      NFMAX=NFREQ(IMODE)
      DO 55 I=1,NFMAX
C
C  STEP 1-2: INITIALIZE COMPLEX DISCRIMINANT COMPONENTS BEFORE EACH
C            XMIT FREQUENCY (NOTE: THESE ARE THE COMPONENT SIGNALS
C            BEFORE SQUARE-LAW DETECTION).
      CSUM=(0.,0.)
      CDIFAZ=(0.,0.)
      CDIFEL=(0.,0.)
      CEARLY=(0.,0.)
      CLATE=(0.,0.)
      CDF1=(0.,0.)
      CDF5=(0.,0.)
      CDF2=(0.,0.)
      CDF4=(0.,0.)
      DO 45 K=1,NT
C
      IF(I.GT.1) GO TO 35
C
C  *************************************************************
C  * STEP 2: COMPUTE SUM CHANNEL MULTIPLICATION FACTOR FOR KTH *
C  *         SCATTERER.                                        *
C  *************************************************************
C
C  STEP 2-1: COMPUTE AZIMUTH AND ELEVATION ANGLE.
      AZ=ATAN2D(RAU(2,K),ABS(RAU(3,K)))
      EL=-ATAN2D(RAU(1,K),ABS(RAU(3,K)))
C  STEP 2-2: COMPUTE ANTENNA SUM, DIFFERENCE AND PHASE FACTORS
      CALL INTERP(AZ,EL)
C
C  STEP 2-3: COMPUTE SUM CHANNEL MULTIPLICATION FACTOR.
      XX=SIG(K)*X
C  NOTE: IF IN ACTIVE MODE SET XX=1.0.
      IF(IMODE.EQ.1) XX=1.0
      S=XX*X
```

```
C
C  STEP 2-4: CHECK ANTENNA STEERING MODE (IF IN GPC-DES OR MANUAL
C          --- SKIP STEP 4).
      IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 20
C
C  ***************************************************************
C  * STEP 3: COMPUTE AZ AND EL DIFFERENCE CHANNEL MULTIPLICATION *
C  *         FACTORS FOR KTH SCATTERER.                          *
C  ***************************************************************
C
C
C  STEP 3-3: COMPUTE AZ AND EL DIFFERENCE CHANNEL MULTIPLICATION
C              FACTORS (INCLUDE RCS AND SUM PATTERN WEIGHTINGS).
C              AND PHASE DIFFERENCE AND BALANCE WEIGHTINGS
      DAZ=XX*Y*CMPLX(COSD(PAZ+PBAL),SIND(PAZ+PBAL))
      DEL=XX*Z*CMPLX(COSD(PEL+PBAL),SIND(PEL+PBAL))
C
C  **********************************************************
C  * STEP 4: COMPUTE RANGE GATE WEIGHTING FOR KTH SCATTERER *
C  **********************************************************
C  DEFINITION:  CTP=4./(C*PULSEWIDTH) WHERE C IS SPEED OF LIGHT.
C
C  STEP 4-1: COMPUTE RANGE GATE LOCATION WRT RANGE GATE CENTER.
CCCCCCCCCCCCCCCCCCC MOD MAR 24 1983 CCCCCCCCCCCCCCCCCC
   20   CONTINUE
      SRNGX=10.*AINT(0.03125*IRNG)
      DELX=CTP(MRNG,IMODE)*(RANGE(K)-SRNGX)
C
C  STEP 4-2: COMPUTE EARLY AND LATE RANGE GATE WEIGHTINGS FOR
C            KTH SCATTERER.
      II=INT((DELX+7.)/2.)
      IF(II.LE.1) II=1
      IF(II.GE.5) II=5
      GO TO (21,22,23,24,21),II
   21 RGE=1.0E-4
      RGL=1.0E-4
      GO TO 25
   22 RGE=3.+DELX
      RGL=0.0
      GO TO 25
   23 RGE=1.-DELX
      RGL=1.+DELX
      GO TO 25
   24 RGE=0.0
      RGL=3.-DELX
C
C  STEP 4-3: COMPUTE RANGE GATE WEIGHT FOR NON-RANGE DISCRIMINANT
C            COMPONENTS.
   25 RGWGT=0.5*(RGL+RGE)
C
C  STEP 4-4: APPLY RANGE GATE WEIGHTING TO SUM AND DIFFERENCE
C            CHANNEL MULTIPLICATION FACTORS.
      RGE=S*RGE
      RGL=S*RGL
      S=S*RGWGT
      DAZ=DAZ*RGWGT
      DEL=DEL*RGWGT
C
C  ********************************************************************
C  * STEP 5: COMPUTE DOPPLER FILTER PHASE SHIFT AND WEIGHTING FOR KTH *
C  *         SCATTERER. NOTE: THIS CALCULATION IS INDEPENDENT OF XMIT *
C  *         FREQUENCY AND ASSUMES NO ACCELERATION OVER DATA CYCLE.   *
C  ********************************************************************
C
```

```
C DEFINITION:  ALAMD(MPRF)=2.*PI/(PRF*LAMBDA)
C DEFINITION:  THE CONSTANT 0.196348=PI/16.
C
C STEP 5-2: COMPUTE DOPPLER FREQUENCY CORRESPONDING TO RADIAL VELOCITY
C           OF KTH SCATTERER.
      FDT=-2.*ALAMD(MPRF)*RADVEL(K)
C
C STEP 5-3: COMPUTE DOPPLER FILTER WEIGHTING FOR EACH OF FIVE DOPPLER
C           TRACKING FILTERS.
      DO 30 J=1,5
      ARG=0.196348*MDF(J)-FDT
  30  DFWTS(J,K)=DOPFIL(ARG)
C
C ********************************************************************
C * STEP 6: COMPUTE PHASE FACTOR ASSOCIATED WITH KTH SCATTERER RANGE *
C *         (NOTE: PHASE IS REFERENCD TO PHASE ASSOCIATED WITH RANGE *
C *         OF TARGET C.G.)                                          *
C ********************************************************************
C
C DEFINITION: RANGE(K) IS RANGE OF KTH SCATTERER TO ANTENNA PHASE CENTR
C DEFINITION: ALAM=4.*PI/LAMBDA WHERE LAMBDA IS XMIT FREQUENCY.
C
C STEP 6-1: COMPUTE PHASE REFERENCED TO TARGET C.G.
  35  DELPSI=ALAM(I)*(RANGE(K)-CGRNGE)
C
C STEP 6-2: COMPUTE PHASE FACTOR, I.E. EXP(J*DELPHI).
      PHASE=CEXP(CMPLX(0.,DELPSI))
      PHASE1=PHASE
C
C STEP 6-3: COMBINE RANGE PHASE FACTOR AND DOPPLER FILTER =3
C           WEIGHT AND PHASE FACTOR.
      PHASE=PHASE*DFWTS(3,K)
C
C ****************************************************************
C * STEP 7: ADD (VECTORIALLY) KTH SCATTERER CONTRIBUTION TO EACH *
C *         DISCRIMINANT'S COMPONENT SIGNALS.                    *
C ****************************************************************
C
C STEP 7-1: ADD KTH SCATTERER CONTRIBUTION TO SUM CHANNEL SIGNAL.
      CSUM=CSUM+S*PHASE
C
C STEP 7-2: CHECK ANTENNA STEERING MODE --- SKIP STEP 8-3 IF IN
C           GPC-DES OR MANUAL MODE.
      IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 40
C
C STEP 7-3: ADD KTH SCATTERER CONTRIBUTION TO AZ AND EL DIFFERENCE
C           CHANNELS SIGNALS.
      CDIFAZ=CDIFAZ+DAZ*PHASE
      CDIFEL=CDIFEL+DEL*PHASE
C
C STEP 7-4: ADD KTH SCATTERER CONTRIBUTION TO RANGE DISCRIMINANT
C           COMPONENT SIGNALS.
  40  CEARLY=CEARLY+RGE*PHASE
      CLATE=CLATE+RGL*PHASE
C
C STEP 7-5: ADD KTH SCATTERER CONTRIBUTION TO VELOCITY DISCRIMINANT
C           COMPONENT SIGNALS.
      PHASE1=PHASE1*S
      CDF2=CDF2+PHASE1*DFWTS(2,K)
      CDF4=CDF4+PHASE1*DFWTS(4,K)
C
C STEP 7-6: ADD KTH SCATTERER CONTRIBUTION TO ON-TARGET DISCRIMINANT
C           COMPONENT SIGNALS.
      CDF1=CDF1+PHASE1*DFWTS(1,K)
      CDF5=CDF5+PHASE1*DFWTS(5,K)
   45 CONTINUE
C
C ***************************************************************
C * STEP 8: FORM NOISE-FREE ANGLE, RANGE, VELOCITY, AND ON-TARGET *
C *         DISCRIMINANT COMPONENTS AT ITH FREQUENCY AND SQUARE   *
C *         LAW DETECT THESE COMPONENTS.                          *
C ***************************************************************
C
C STEP 8-1: CHECK ANTENNA STEERING MODE ---  SKIP STEPS 9-2 AND 9-3
C           IF IN GPC-DES OR MANUAL.
      IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 50
C
C STEP 8-2: COMPUTE AZ DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
      SPAZ=SPAZ+CABS(CSUM+CDIFAZ)**2
      SMAZ=SMAZ+CABS(CSUM-CDIFAZ)**2
C
C STEP 8-3: COMPUTE EL DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
      SPEL=SPEL+CABS(CSUM+CDIFEL)**2
      SMEL=SMEL+CABS(CSUM-CDIFEL)**2
C
C STEP 8-4: COMPUTE RANGE DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT
   50 EARLY=EARLY+CABS(CEARLY)**2
      LATE=LATE+CABS(CLATE)**2
C
C STEP 8-5: COMPUTE VELOCITY DISCRIMINANT COMPONENTS AND SQUARE-LAW
C           DETECT.
      DF2=DF2+CABS(CDF2)**2
      DF4=DF4+CABS(CDF4)**2
C
C STEP 8-6: COMPUTE ON-TARGET DISCRIMINANT COMPONENTS AND SQUARE-LAW
C           DETECT.
      DF1=DF1+CABS(CDF1)**2
      DF5=DF5+CABS(CDF5)**2
C
C **************************************************************
C * STEP 9: COMPUTE EFFECTIVE CROSS-SECTION AVERAGED OVER PROPER *
C *         NUMBER OF TRANSMIT FREQUENCIES.                      *
C **************************************************************
      SIGBAR=SIGBAR+CABS(CSUM)**2
   55 CONTINUE
      SIGBAR=SIGBAR/FLOAT(NFREQ(IMODE))
C
C  NOTE: DEBUGGING PRINT STATEMENTS
C     WRITE(6,900) (I,SIG(I), I=1,NT)
  900 FORMAT(' I,SIG =',I8,F14.4)
C     WRITE(6,902) NT,S,DAZ,DEL,RGE,RGL,RGWGT,MDF(3)
C      WRITE(6,901) DFWTS(1,K),DFWTS(2,K),DFWTS(3,1),DFWTS(4,1),
C    2 DFWTS(5,1)
  902 FORMAT(' NT,S,DAZ,DEL,RGE,RGL,RGWGT,F3 =',I5,6F10.2,I5)
  901 FORMAT(' DF WTS =',10F12.4)
      RETURN
      END
C
C
C ******************************************************
C * THIS SUBROUTINE CONTAINS SINGLE-HIT DETECTION MODEL *
C ******************************************************
C
C
      SUBROUTINE SINGLE
      DIMENSION P(41)
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUM(5),DUMC(3)
      COMMON /OUTPUT/MSWF,MTF,MSF,DUM(7),IDUM1(4)
      COMMON /ICNTL/IDUM2(8),KACCLK,MTP,IDUM3(5),MSAM,IDUM4(11)
      COMMON /TGTDAT/NT,DUM1(500),RO(3),ROU(3),CGRNGE,CGVEL
      COMMON /DETDAT/SIGMA,CGANG
      DATA NSRCH/105/
      DATA P/6*0.0,.001,.003,2*.004,.008,.012,.015,.043,.053,.076,.107,
     2 .147,.193,.244,.312,.363,.444,.514,.590,.644,.706,.765,.815,.861,
     3 .882,.918,.937,.955,.966,.976,.980,.989,.991,.997,.996/
C
C  ******************************************************
C  * STEP 1: COMPUTE NOMINAL SNR AT VIDEO FILTER OUTPUT *
C  ******************************************************
C
C  STEP 1-1: SET SAMPLE RATE TO OBTAIN CORRECT NOISE BW IN SNRV COMP.
      MSAM=1
      IF (IMODE.EQ.1) MSAM=2
C
C  STEP 1-2: COMPUTE NOMINAL SNRV.
      SNR=SNRV(SIGMA,CGRNGE)
C
C  ******************************************************
C  * STEP 2: IF NOT SCANNING ADD BEAMSHAPE LOSS TO SNRV *
C  ******************************************************
C
C  STEP 2-1: CHECK SCAN FLAG.
      IF(MSF.EQ.1) GO TO 1
C
C  STEP 2-2: COMPUTE BEAMSHAPE LOSS --- BASED UPON C.G. POSITION
C            OFF BORESIGHT.
      BETA2=SPAT(CGANG)**2
C
C  STEP 2-3: ADD BEAMSHAPE LOSS TO NOMINALV, I.E. COMPUTE ACTUAL SNR
C            SNRV.
      SNR=SNR*BETA2
C
C  ******************************************************************
C  * STEP 3: DETERMINE PROBABILITY OF DETECTION, PD, BASED UPON SNR *
C  ******************************************************************
C
C  STEP 3-1: DETERMINE INDEX TO ACCESS APPROPRIATE PD VERSUS SNR
C            CURVE.
    1 IF(IMODE.EQ.2) GO TO 5
      NCRV=1
      GO TO 15
    5 IF(IASM.LT.3) GO TO 10
      NCRV=3
      GO TO 15
   10 NCRV=5
C
C  ADJUST INDEX FOR SCANNING.
   15 NCRV=NCRV+MSF
C
C
C  STEP 3-2: CONVERT SNRV TO DB.
      IF(SNR.LT.1.E-08) GO TO 20
      SNR=10 *ALOG10(SNR)
      GO TO 25
   20 SNR=-100.
C
C  STEP 3-3: SNR OUTSIDE (-30 DB, 0 DB) INTERVAL" --- IF SO, SET
C            OUTCOME APPROPRIATELY AND SKIP REMAINING STEPS.
C
C  IF SNR < -25 DB THEN SET PD=0.0 (DECLARE A MISS).
   25 IF(SNR.LT.-25.) GO TO 30
C
```

```
C  IF SNR > -5 DB THEN SET PD=1.0 (DECLARE A HIT).                       00008190
      IF(SNR.GT.-5.0) GO TO 35                                           00008200
C                                                                        00008210
C  STEP 3-4: COMPUTE INDEX FOR LOOKUP TABLE AND FACTORS FOR LINEAR       00008220
C            INTERPOLATION.                                              00008230
      SCALE=(SNR+25.)*2.+1.000001                                        00008240
      ISNR=INT(SCALE)                                                    00008250
      REMAIN=SCALE-FLOAT(ISNR)                                           00008260
C                                                                        00008270
C  STEP 3-5: DETERMINE PD USING TABLE AND LINEAR (IN DB) INTERPOLATION.  00008280
      PROB=P(ISNR)+REMAIN*(P(ISNR+1)-P(ISNR))                            00008290
C                                                                        00008300
C  ***************************************************                   00008310
C  * STEP 4: DETERMINE OUTCOME OF DETECTION ATTEMPT *                    00008320
C  ***************************************************                   00008330
C                                                                        00008340
      X=RNDU(NSRCH)                                                      00008350
      IF(X.LE.PROB) GO TO 35                                             00008360
C                                                                        00008370
C  *****************************************************************     00008380
C  * STEP 5: SET CONTROLS BASED UPON OUTCOME OF DETECTION ATTEMPT *      00008390
C  *****************************************************************     00008400
C                                                                        00008410
C  STEP 5-1: IF NO DETECTION ---- SET TARGET PRESENT FLAG LOW.           00008420
   30 MTP=0                                                              00008430
      RETURN                                                             00008440
C                                                                        00008450
C  STEP 5-2: IF DETECTION SUCCESSFUL ---- SET TARGET PRESENT FLAG        00008460
C            HIGH AND INITIALIZE ACQUISITION CLOCK.                      00008470
   35 MTP=1                                                              00008480
      KACCLK=0                                                           00008490
      RETURN                                                             00008500
      END                                                                00008510
C                                                                        00010640
C                                                                        00010650
C  ******************************************************************    00010660
C  * THIS FUNCTION COMPUTES THE ●NOMINAL● SNR AT THE VIDEO OUTPUT *      00010670
C  * ---- IT ASSUMES NO BEAMSHAPE OR SCAN LOSS.                     *    00010680
C  ******************************************************************    00010690
C                                                                        00010700
C                                                                        00010710
      FUNCTION SNRV(SIGMA,RANGE)                                         00010720
      COMMON /CNTL/IPWR,IMODE,ITXP,IDUMC(6),DUMC(3)                      00010730
      COMMON /ICNTL/IDUM(12),MSS,MTKINT,MRNG,MSAM,MPRF,IDUM2(10)         00010740
      COMMON /SYSDAT/DUM(12),TGTSIG,GPS,GAS                              00010750
      DIMENSION PT(4),BN(2)                                              00010760
CCCCCCCCCCCCCCCCC MOD MAR 24 1983 CCCCCCCCCCCCCCCCCCC
      DATA PT/46.3,34.9,23.,6.2/, BN/69.9,57.9/                                  00010770
C                                                                        00010780
C  ********************************************                          00010790
C  * DETERMINE WHETHER ACTIVE OR PASSIVE MODE *                          00010800
C  ********************************************                          00010810
      IF(IMODE.EQ.1) GO TO 10                                            00010820
C                                                                        00010830
C  **************************************                                00010840
C  * PASSIVE MODE VIDEO SNR CALCULATION *                                00010850
C  **************************************                                00010860
       IF((SRNG.LT.640.).OR.(ISTS7.EQ.1))ITXP=4
      SNRV=GPS+PT(ITXP)+10.*ALOG10(SIGMA)-BN(MSAM)-40.*ALOG10(RANGE)     00010870
      SNRV=10.**(0.1*SNRV)                                               00010880
      RETURN                                                             00010890
C                                                                        00010900
C  *************************************                                 00010910
C  * ACTIVE MODE VIDEO SNR CALCULATION *                                 00010920
```

```
C   ***********************************                                 00010930
   10  SNRV=GAS-20.*ALOG10(RANGE)                                       00010940
       SNRV=10.**(0.1*SNRV)                                             00010950
       RETURN                                                           00010960
       END                                                              00010970
C MODIFIED FOR LENS
C                                                                       00032640
C                                                                       00032650
C   *****************************************************               00032660
C   * THIS SUBROUTINE MODELS THE SPAS SPACECRAFT SCATTERING *           00032670
C   * PROPERTIES.                                           *           00032680
C   *****************************************************               00032690
C                                                                       00032700
C   SES SPAS MODEL AS OF JULY 7,1981.
C
C                                                                       00032710
       SUBROUTINE SPAS                                                  00032720
       COMMON /SATDAT/RADAR(3),KTAR,R(70,3),SIG(70),ROLD,ICLOSE,ICLOLD
     1 ,JHOT(60)
       DIMENSION SIGMA(61),TARG(61,3),PHIMIN(61,3),PHIMAX(61,3)
       DIMENSION OFFSET(61)          ,PHI(61,3)
       DIMENSION VECT(3),COSPHI(61,3)
       DIMENSION ALPH(24,3),V(24,3),NORMAL(24),DIM(24,3),WRAN(24,3)
       DIMENSION WSCALE(24,3),DPHI(24),PHIOLD(24),VOLD(24,3),KSEED(24,3)
       DIMENSION TTRAN(3)
C                                                                       00032800
C                                                                       00032810
C   ***************************************************************     00032820
C   * DATA DEFINITION: INCLUDES SCATTERER LOCATION IN TARGET FRAME, *   00032830
C   *                  MAXIMUM SCATTERER RCS VALUE, ANGULAR EXTENT  *   00032840
C   *                  OF NONZERO RCS, AND OTHER MISCELLANEOUS DATA *   00032850
C   *                  REQUIRED BY THE ROUTINE.                     *   00032860
C   ***************************************************************     00032870
C                                                                       00032880
C                                                                       00032890
C   SEED FOR RANDOM NUMBER GENERATOR                                    00032900
       DATA KSEED/45,678,908,607,5678,897,345,7777,67,4,
     1 560,809,444,888,999,555,222,70,80,8000,
     2 5,15,25,35,45,55,65,75,85,95,
     3 7,17,27,37,47,57,67,77,87,97,
     4 9876,984,6666,2398,76,412,7589,409,899,561,
     5 205,3895,9457,9643,937,656,453,980,567,2154,
     6 801,88,99,31,85,106,4,9,3,987,
     7 888,999/
C                                                                       00032970
C   DATA DESCRIBING DIMENSIONS OF WIDE-ANGLE SCATTERERS                 00032980
C   DEFINITION: DIM=2*D/LAMBDA (UNITLESS)                               00032990
C   DEFINITION: WSCALE=SQRT(D**2/(12*NF)) (UNITS=FEET, NF== OF FREQ)    00033000
       DATA DIM /72*64.8/
       DATA WSCALE /72*0.2965/
C
C   FOR EACH DIFFUSE SCATTERER, SPECIFY NORMAL COMPONENT
       DATA NORMAL /10*1,2*2,12*3/
C
C   SQUARE ROOT OF RCS VALUES ( FEET).
       DATA SIGMA/24*.05,3*2.6,2*61.,1200.,1.25,0.17,25.7,110.,90.,
     2 100.,850.,1200.,1117.,0.4,80.,100.,900.,85.,750.,850.,920.,
     3 730.,6*0.03,1250.,1130.,1400.,900.,1000.,1150.,32.39/
C
C   COORDINATES OF SCATTERERS IN SPAS FRAME (FEET)
       DATA TARG /4*.12,6*-.7,8*-.35,.37,4*-.35,.37,3*.24,2*.37,
     2  .66,3*-.35,3*.12,3*-.3,5*-.35,4*.37,6*.24,6*.7,0.0,
     3 1.75,-1.05,-1.75,.35,1.75,1.05,.35,-.35,-1.05,-1.75,2.15,
     4 -2.15,1.75,1.05,.35,-.35,-1.05,-1.75,.35,1.05,.35,-.35,
     5 -1.75,.35,-.83,-1.05,-1.27,1.05,-.35,.35,3*-1.05,1.9,-1.05,
     6 -1.8,2.0,-2.0,.0,1.75,1.05,.35,-.35,-1.75,2*1.05,2*-.35,
     7 2*-.83,2*-1.05,2*-1.27,1.75,1.05,.35,-.35,-1.05,-1.75,0.0,
     8 12*.0,7*.48,5*-.48,3*.15,3*.0,3*-.8,3*.0,3*.67,-.86,
     9 4*-.48,.425,-.425,.425,-.425,-.02,.3,-.02,.3,-.02,.3,
     A 6*0.0,2.38/
C
C  MINIMUM SUBTENDED ANGLE
       DATA PHIMIN /4*.0,6*90.,14*0.,16*0.,4*88.5,4*88.0,6*0.0,
     2 6*177.9,0.,
     3 11*.0,90.,12*.0,50.,35.,30.,.0,45.0,3*.0,10.0,4*.0,177.4,
     4 89.7,.0,4*88.5,4*88.0,12*.0,48.,
     5 19*0.,5*90.,3*85.9,3*88.5,156.,90.,87.7,3*88.5,2*87.4,.0,
     6 90.,4*178.5,0.,178.,0.,178.,90.,0.,90.,0.,90.,0.,6*88.5,
     7  48.0/
C
C  MAXIMUM SUBTENDED ANGLE
       DATA PHIMAX /4*90.,20*180.,5*90.,2.1,3*180.,3*2.1,4*180.,
     2 4*91.5,4*92.,6*90.,6*180.,48.,
     3 10*180.,90.,13*180.,4*150.,155.,135.,2*180.,145.,3*180.,
     4 2.6,180.,90.3,180.,4*91.5,4*92.,6*180.,6*180.,138.,
     5 12*180.,7*90.,5*180.,3*94.1,3*91.5,180.,156.,92.3,3*91.5,2*92.6,
     6 125.,5*180.,2.,180.,2.,2*180.,90.,180.,90.,180.,90.,6*91.5,138./
C                                                                       00033580
C  RADII OF THE SCATTERERS (FEET)                                       00033590
       DATA OFFSET /24*.0,3*.1,2*.29,.0,2*.35,.315,5*.0,.24,.35,8*0.,
     2 6*.1,6*.0,0.0/
C                                                                       00033620
C  MISCELLANEOUS DATA.                                                  00033630
       DATA NTAR/61/,KWIDE/24/,PI/3.141592653/
       DATA TTRAN/3*0.0/,INIT1/1/
C                                                                       00033660
C  ******************************************************************  00033670
C  * STEP 0: TRANSLATE POINT TARGETS BY TARGET FRAME OFFSET (TTRAN) *  00033680
C  ******************************************************************  00033690
       IF(INIT1.NE.1) GO TO 2                                           00033700
C
C  RANDOMIZE DIFFUSE SCATTERER RCS VALUES.
C
        ISEED=100
        DO 107 I=1,1000
107     X=RNDU(ISEED)
        DO 108 I=1,KWIDE
        X=RNDU(ISEED)
C      CHANCE MADE 9-11-81
108     SIGMA(I)=SIGMA(I)+(X*0.005)-0.0025
C
C  CONVERT TARGET DATA APPROPRIATELY.
C
        FTM=0.3048
        DO 101 I=1,NTAR
101     SIGMA(I)=SQRT(SIGMA(I))/FTM
        DO 102 J=1,NTAR
        DO 102 I=1,3
102     TARG(J,I)=TARG(J,I)/FTM
        DO 103 J=1,NTAR
        DO 103 I=1,3
        PHIMIN(J,I)=COS(PHIMIN(J,I)*PI/180.)
103     PHIMAX(J,I)=COS(PHIMAX(J,I)*PI/180.)
        DO 105 I=1,NTAR
105     OFFSET(I)=OFFSET(I)/FTM
C
       DO 1 K=1,NTAR                                                    00033710
       DO 1 I=1,3                                                       00033720
```

```
    1 TARG(K,I)=TARG(K,I)+TTRAN(I)
      INIT1=0
C
C ***************************************************************
C * STEP 1: DETERMINE WHICH SCATTERER ARE ILLUMINATED AND HAVE A *
C *         NONZERO RCS IN THE DIRECTION OF THE RADAR.           *
C ***************************************************************
C
C STEP 1-1: PERFORM REQUIRED INITIALIZATIONS.
    2 CONTINUE
      NWIDE=0
      KTAR=0
C
C STEP 1-2: COMPUTE UNIT VECTOR IN DIRECTION OF RADAR FOR
C           ITH SCATTERING CENTER.
      DO 15 I=1,NTAR
      DO 5 J=1,3
      VECT(J)=RADAR(J)-TARG(I,J)
    5 CONTINUE
      VNORM=SQRT(VECT(1)**2+VECT(2)**2+VECT(3)**2)
      DO 10 J=1,3
     IF(ABS(VECT(J)).GT.ABS(VNORM))WRITE(6,*)'VECT GREATER THAN VNORM'
      COSPHI(I,J)=VECT(J)/VNORM
C
C STEP 1-3: DETERMINE WHETHER ITH SCATTERER HAS A NONZERO RCS IN THE
C           DIRECTION OF THE RADAR.
      IF(COSPHI(I,J).LT.PHIMAX(I,J).OR.COSPHI(I,J).GT.PHIMIN(I,J))
     2 GO TO 15
   10 CONTINUE
C
C STEP 1-4: IF ITH SCATTERER RCS IS NONZERO THEN ADD TO VECTOR OF
C           ILLUMINATED SCATTERERS.
      KTAR=KTAR+1
      JHOT(KTAR)=I
      SIG(KTAR)=SIGMA(I)
      IF(I.LE.KWIDE) NWIDE=NWIDE+1
   15 CONTINUE
C
C *******************************************************************
C * STEP 2: COMPUTE LOCATION OF SPECULAR POINTS THAT ARE ILLUMINATED *
C *******************************************************************
      DO 20 K=1,KTAR
      I=JHOT(K)
      DO 20 J=1,3
      R(K,J)=TARG(I,J)+OFFSET(I)*COSPHI(I,J)
   20 CONTINUE
C
C **************************************************************
C * STEP 3: COMPUTE SQUARE ROOT OF RCS FOR ALL ILLUMINATED WIDE *
C *         ANGLE SCATTERERS (REPRESENTING DIFFUSE SCATTERING   *
C *         AREAS).                                             *
C **************************************************************
      DO 22 K=1,NWIDE
      I=JHOT(K)
      IQ=NORMAL(I)
   22 SIG(K)=SQRT(ABS(COSPHI(I,IQ       )))*SIGMA(I)
C
C ****************************************
C * STEP 4: CHECK FOR SHORT RANGE CONDITION *
C ****************************************
C
C STEP 4-1: DETERMINE RANGE TO RADAR IN TARGET FRAME.
   24 RANGE=SQRT(RADAR(1)**2+RADAR(2)**2+RADAR(3)**2)
C
```

```
C  STEP 4-2: SET HYSTERESIS LOOP MONITORING VARIABLE.
       IF((ROLD.LT..01.OR.RANGE-ROLD.LE.0.).AND.RANGE.LE.270.) ICLOSE=1
       IF(RANGE-ROLD.GT.0..AND.RANGE.GT.50.) ICLOSE=0
C
C  STEP 4-3: CHECK MONITORING VARIABLE TO DETERMINE IF SHORT RANGE
C            CONDITION EXISTS.
       IF(ICLOSE.EQ.0.OR.NWIDE.EQ.0) GO TO 55
C
C  *********************************************************
C  * STEP 5: PROCEDURE FOR UPDATING OF DIFFUSE SCATTERING    *
C  *         CENTER LOCATION ---- SHORT RANGE CONDITION ONLY. *
C  *********************************************************
C
C  STEP 5-1: IF FIRST TIME THRU ---- PERFORM INITIALIZATION OF
C            DIFFERENCE EQUATIONS FOR ALL DIFFUSE SCATTERERS.
       IF(ICLOLD.EQ.1) GO TO 35
       DO 30 I=1,KWIDE
      IQ=NORMAL(I)
       PHIOLD(I)=ACOS(COSPHI(I,IQ        ))
       DO 25 J=1,3
       IF(J.EQ.IQ        ) GO TO 25
       V(I,J)=WSCALE(I,J)*(RNDU(KSEED(I,J))-.5)
       VOLD(I,J)=V(I,J)
       R(I,J)=R(I,J)+V(I,J)
   25  CONTINUE
   30  CONTINUE
       GO TO 55
C
C  STEP 5-2: UPDATE ANGULAR INCREMENT FOR EACH DIFFUSE SCATTERER
C            ---- CHANGE IN ANGLE FROM SAMPLE-TO-SAMPLE.
   35  DO 40 I=1,KWIDE
      IQ=NORMAL(I)
       PHI(I,IQ        )=ACOS(COSPHI(I,IQ        ))
       DPHI(I)=(PHI(I,IQ        )-PHIOLD(I))
       PHIOLD(I)=PHI(I,IQ        )
   40  CONTINUE
C
C
C  STEP 5-3: UPDATE SCATTERER LOCATION FOR ALL ILLUMINATED DIFFUSE
C            SCATTERER ---- UPDATE DIFFERENCE EQUATIONS.
       DO 50 K=1,NWIDE
       I=JHOT(K)
       DO 45 J=1,3
      IQ=NORMAL(I)
       IF(J.EQ.IQ        ) GO TO 45
       ALPH(I,J)=EXP(-DIM(I,J)*ABS(DPHI(I)*COSPHI(I,IQ        )))
       WRAN(I,J)=SQRT(1.-ALPH(I,J)**2)*WSCALE(I,J)*(RNDU(KSEED(I,J))-.5)
       V(I,J)=ALPH(I,J)*VOLD(I,J)+WRAN(I,J)
       VOLD(I,J)=V(I,J)
       R(K,J)=R(K,J)+V(I,J)
   45  CONTINUE
   50  CONTINUE
  55   CONTINUE .
C
C  ***********************************************************
C  * STEP 6: UPDATE PARAMETERS USED TO MONITOR TARGET POSITION *
C  *         ON SHORT RANGE HYSTERESIS CURVE.                  *
C  ***********************************************************
       ROLD=RANGE
       ICLOLD=ICLOSE
C
C
C      WRITE(6,908) KTAR,NWIDE,ICLOSE,ROLD
  908  FORMAT(/' TT,WT,IC,R =',3I8,F12.4)
```

```
C  ****************************************************************   00034950
C  * NOTE: THE FOLLOWING STATEMENTS ARE PRINT STATEMENTS USED IN THE *  00034960
C  *        DEBUGGING PROCESS.                                       *  00034970
C  ****************************************************************   00034980
C                                                                     00034990
C                                                                     00035000
C  NOTE: DEBUGGING PRINT STATEMENTS.                                  00035010
C  PRINT LOCATION OF RADAR IN TARGET FRAME.                           00035020
C      WRITE(6,900) RADAR                                             00035030
C                                                                     00035040
C  PRINT TABULAR LISTING OF ALL DATA ASSOCIATED WITH SPAS SCATTERERS. 00035050
C      WRITE(6,901)(I,SIGMA(I),TARG(I,1),TARG(I,2),TARG(I,3),OFFSET(I) 00035060
C    8 ,PHIMIN(I,1),                                                  00035070
C    1 PHIMAX(I,1),PHIMIN(I,2),PHIMAX(I,2),PHIMIN(I,3),PHIMAX(I,3),   00035080
C    2 I=1,NTAR)                                                      00035090
C                                                                     00035100
C  PRINT TOTAL = OF SCATTERERS AND = OF DIFFUSE SCATTERERS.           00035110
C      WRITE(6,902) KTAR,NWIDE                                        00035120
C                                                                     00035130
C  PRINT INFORMATION ASSOCIATED WITH ILLUMINATED SCATTERERS.          00035140
C      WRITE(6,903)                                                   00035150
C      WRITE(6,904) (I,JHOT(I),SIG(I),(R(I,J),J=1,3),                 00035160
C    1 I=1,KTAR)                                                      00035170
C                                                                     00035180
C  PRINT DATA ASSOCIATED WITH DIFFUSE SCATTERER DIFFERENCE EQUATION.  00035190
C      WRITE(6,905)I,PHIOLD(I),                                       00035200
C    1 (V(I,L),L=1,3),(R(I,L),L=1,3)                                  00035210
C     IQ=NORMAL(I)
C      WRITE(6,906) I,PHI(I,IQ        ),PHIOLD(I),DPHI(I)             00035220
C      WRITE(6,907)K,I,(VOLD(I,J),J=1,3),(ALPH(I,J),J=1,3),           00035230
C    1 (WRAN(I,J),J=1,3),(V(I,J),J=1,3),(R(I,J),J=1,3)                00035240
C                                                                     00035250
C  ALL PRINT FORMAT STATEMENTS.                                       00035260
  900  FORMAT(' IN FEET, RADAR = (',F8.1,',',F8.1,',',F8.1,')')       00035270
  901  FORMAT(I12,F10.2,3F8.3,F12.3,4X,2F8.2,4X,2F8.2,4X,2F8.2)       00035280
  902  FORMAT(' TOTAL = OF TARGETS = ',I3,'      OF THESE, = MARKOV = ',00035290
     1 I2)                                                            00035300
  903  FORMAT(//,9X,'I',3X,'JHOT(I)',7X,'RCS',5X,'PHI-X',5X,'PHI-Y',  00035310
     1 5X,'PHI-Z',/)                                                  00035320
  904  FORMAT(2I10,4F10.3)                                            00035330
  905  FORMAT(I3,F15.3,2(5X,3F10.3))                                  00035340
  906  FORMAT(' I,PHI,PHIOLD,DPHI',/,I3,3F10.3)                       00035350
  907  FORMAT(2I3,5(2X,3F7.3))                                        00035360
       RETURN                                                         00035370
       END                                                            00035380
C                                                                     00030460
C                                                                     00030470
C  **************************************************************     00030480
C  * THIS FUNCTION GIVES THE ANTENNA SUM PATTERN WEIGHTING OF THE *   00030490
C  * RADAR SIGNAL FOR THE GIVEN ANGLE(IN RADIANS) OFF BORESIGHT   *   00030500
C  **************************************************************     00030510
C                                                                     00030520
C                                                                     00030530
       FUNCTION SPAT(X)                                               00030540
C  NOTE: THE FOLLOWING VALUE OF B GIVES THE SUM PATTERN A SINGLE-SIDED 00030550
C         3 DB BEAMWIDTH OF 0.85 DEGREES.                             00030560
       Y=93.80*X                                                      00030570
       TEMP=ABS(Y)                                                    00030580
       IF(TEMP.GT.1.0E-06) GO TO 10                                   00030590
       SPAT=1.0                                                       00030600
       RETURN                                                         00030610
   10  SPAT=SIN(Y)/Y                                                  00030620
       RETURN                                                         00030630
       END                                                            00030640
```

```
C                                                                       00010490
C                                                                       00010500
C  ******************************************************************** 00010510
C  * THIS FUNCTION COMPUTES THE EXPRESSION (SIN(NX)**2/(N SIN(X)**2)) * 00010520
C  ******************************************************************** 00010530
C                                                                       00010540
C                                                                       00010550
      FUNCTION SUM(X,N)                                                 00010560
      Y=SIN(X)**2                                                       00010570
      IF(Y.GT.1.0E-08) GO TO 10                                         00010580
      SUM=N                                                             00010590
      RETURN                                                            00010600
   10 SUM=SIN(N*X)**2/(N*Y)                                             00010610
      RETURN                                                            00010620
      END                                                               00010630
C                                                                       00004100
C                                                                       00004110
C  ********************************************************************* 00004120
C  * THIS SUBROUTINE RESETS THE SYSTEM UNDER THE FOLLOWING CONDITIONS * 00004130
C  * (1) BREAK-TRACK (TO SEARCH), (2) PASSIVE/ACTIVE MODE CHANGE (TO  * 00004140
C  * SEARCH), AND (3) SYSTEM IN STANDBY (TO IDLE).                    * 00004150
C  ********************************************************************* 00004160
C                                                                       00004170
C                                                                       00004180
      SUBROUTINE SYSINT                                                 00004190
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)                00004200
      COMMON /OUTPUT/MSWF,MTF,MSF,SRNG,SRDOT,SPANG,SRANG,SPRTE,SRRTE,   00004210
     2               SSRS,MADVF,MRDVF,MARDVF,MRRDVF                     00004220
      COMMON /ICNTL/IOLDPW,IOLDMD,IOLDSM,ISHOLD,KMSCLK,KWMUP,KSNCLK,    00004230
     2             KSNMAX,KACCLK,MTP,MZ1,MZ0,MSS,MTKINT,MRNG,MSAM,MPRF,00004240
     3             MBKTRK,MBTSUM,MBT(8)                                 00004250
      COMMON /ATDAT/DUM1(4),ALRATE,BTRATE,DUM2(2),AL,BT,PREF,RREF       00004260
C                                                                       00004270
C  *****************************************************                00004280
C  * STEP 1: INITIALIZE ALL INTERNAL FLAGS AND CONTROLS *               00004290
C  ******************************************************               00004300
      IOLDMD=IMODE                                                      00004310
      IOLDSM=IASM                                                       00004320
      ISHOLD=0                                                          00004330
      MTP=1                                                             00004340
      MZ1=0                                                             00004350
      MZ0=0                                                             00004360
      MSS=0                                                             00004370
      MTKINT=0                                                          00004380
C                                                                       00004390
C  ******************************************                           00004400
C  * STEP 2: INITIALIZE ALL INTERNAL CLOCKS *                           00004410
C  ******************************************                           00004420
      KACCLK=0                                                          00004430
      KSNCLK=0                                                          00004440
C                                                                       00004450
C  ****************************************                             00004460
C  * STEP 3: INITIALIZE ALL DISPLAY FLAGS *                             00004470
C  ****************************************                             00004480
      MSWF=0                                                            00004490
      MSF=0                                                             00004500
      MTF=0                                                             00004510
      MADVF=0                                                           00004520
      MRDVF=0                                                           00004530
      MRRDVF=0                                                          00004540
      MARDVF=0                                                          00004550
C                                                                       00004560
C  *****************************************                            00004570
C  * STEP 4: INITIALIZE ALL DISPLAY METERS *                            00004580
```

```
C  ***************************************
      SRNG=0.0
      SRDOT=0.0
      SPRTE=0.0
      SRRTE=0.0
      SRSS=0.0
C
C  *******************************************
C  * STEP 5: INITIALIZE GIMBAL POINTING LOOP *
C  *******************************************
      PII=3.14159265/180.
      ALRATE=0.0
      BTRATE=0.0
      IF(IPWR.NE.1.AND.KMSCLK.NE.1) GO TO 5
C
C  STEP 5-1: IF SYSTEM POWER OFF THEN ALIGN BORESIGHT WITH ZENITH.
      PREF=0.0
      RREF=0.0
      AL=0.0
      BT=0.0
      SPANG=0.0
      SRANG=0.0
      IOLDPW=IPWR
      RETURN
    5 IF(IPWR.GT.2) GO TO 15
C
C  STEP 5-2: IF SYSTEM IN STANDBY THEN HOLD GIMBALS AT POSITION WHEN
C            STANDBY ENTERED AND ZERO DISPLAYS.
      IF(IOLDPW.EQ.IPWR) GO TO 10
      PREF=PII*SPANG
      RREF=PII*SRANG
   10 SPANG=0.0
      SRANG=0.0
      IOLDPW=IPWR
      RETURN
C
C  STEP 5-3: PREPARE GIMBAL LOOP FOR ENTRY INTO ANY OF SEARCH MODES.
   15 PREF=PII*SPANG
      RREF=PII*SRANG
      IOLDPW=IPWR
      RETURN
      END
C
C
C  ******************************************************
C  * THIS SUBROUTINE UPDATES THE DATA VALID FLAG STATUS *
C  ******************************************************
C
C
      SUBROUTINE TGTACQ
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)
      COMMON /OUTPUT/MSWF,MTF,MSF,DUM1(7),MADVF,MRDVF,MARDVF,MRRDVF
      COMMON /ICNTL/IDUM3(8),KACCLK,MTP,MZ1,MZ0,MSS,MTKINT,
     2              MRNG,IDUM4(12)
      COMMON /SYSDAT/TS,DUMS(14)
      DIMENSION ADV(10,2),RDV(10,2),ARDV(10,2)
      DATA ADV/9*1.02,5.12,8*1.02,2*2.33/
      DATA RDV/9*6.15,28.69,8*6.97,2*29.76/
      DATA ARDV/9*8.2,28.69,7*8.2,26.23,2*29.76/
C
C  ************************************
C  * STEP 1: UPDATE ACQUISITION CLOCK *
C  ************************************
      KACCLK=KACCLK+1
      ACCLK=KACCLK*TS
C
C  *****************************************************************
C  * STEP 2: PERFORM ANGLE DATA VALID TEST —— GPC-ACQ + AUTO ONLY *
C  *****************************************************************
      IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 10
      IF(ACCLK.LT.ADV(MRNG,IMODE))  GO TO 10
      MADVF=1
C
C  *********************************************************
C  * STEP 3: PERFORM RANGE AND RANGE RATE DATA VALID TEST *
C  *********************************************************
   10 IF(ACCLK.LT.RDV(MRNG,IMODE))  GO TO 15
      MRDVF=1
      MRRDVF=1
C
C  IF GPC-DES OR MANUAL INITIALIZE RADAR TRACKING PARAMETERS.
CCCCCCCCCCCCCCCCCCC MOD MAR 24 1983 CCCCCCCCCCCCCCCCCCCCCCC
   15 IF((IASM.EQ.2.OR.IASM.EQ.4).AND.MRDVF.EQ.1) GO TO 20
C
C  *****************************************************************
C  * STEP 4: PERFORM ANGLE RATE DATA VALID TEST —— GPC-ACQ + AUTO *
C  *         MODES ONLY.                                           *
C  *****************************************************************
      IF(ACCLK.LT.ARDV(MRNG,IMODE)) RETURN
      MARDVF=1
C
C  **************************************************************
C  * STEP 5: PERFORM STEADY STATE RADAR TRACKING INITIALIZATION *
C  **************************************************************
   20 KACCLK=0
      MTF=1
      RETURN
      END
C
C
C  **************************************************************
C  * THIS SUBROUTINE GENERATES A (3X3) MATRIX TTH THAT PRODUCES *
C  * A ROTATION OF TH RADIANS ABOUT THE X-AXIS.                 *
C  **************************************************************
C
C
      SUBROUTINE THETA(TTH,TH)
      DIMENSION TTH(3,3)
      DO 10 I=1,3
      DO 10 J=1,3
10    TTH(I,J)=0.0
      TTH(1,1)=1.0
      TTH(2,2)=COS(TH)
      TTH(3,3)=TTH(2,2)
      TTH(2,3)=SIN(TH)
      TTH(3,2)=-TTH(2,3)
      RETURN
      END
C
C
C  **************************************************************
C  * THIS SUBROUTINE INITIALIZES THE ANGLE TRACKING LOOPS, THE  *
C  * RANGE TRACKING LOOP, AND THE VELOCITY PROCESSOR —— STEADY *
C  * STATE CONDITIONS ARE ASSUMED.                              *
C  **************************************************************
C
C
      SUBROUTINE TKINIT
      REAL INTT,IRNG,IRDOT,IVR                                          00015195
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)                00015200
      COMMON /INPUT/ ERT(3),EVT(3),EWB(3),DUM(18)                       00015210
      COMMON /OUTPUT/ I3DUM(3),SRNG,DUM1(6),IDUM1(4)                    00015220
      COMMON /ICNTL/I1DUM(13),MTKINT,MRNG,MSAM,MPRF,MBKTRK,MBTSUM,      00015230
     2              MBT(8),MPFOLD                                       00015240
      COMMON /SYSDAT/TSAM,DR(3),CP,SP,PSI,PSBIAS,DUM2(7),TRB(3,3)       00015250
      COMMON /TGTDAT/NT,DUM5(500),RO(3),ROU(3),CGRNGE,CGVEL             00015260
      COMMON /SATDAT/RADAR(3),KTAR,RT(70,3),SIG(70),ROLD,ICLOSE,ICLOLD  00015270
      COMMON /ATDAT/CA,SA,CB,SB,AZRATE,ELRATE,ALRATE,BTRATE,AL,BT,      00015280
     2              DUM3(2)                                             00015290
      COMMON /RTDAT/IRDOT,IRNG,RBIAS,VEST(4),MDF(5)                     00015300
      COMMON /XFORMS/ TLB(3,3),TLBD(3,3),TLT(3,3),TLTD(3,3)             00015310
      COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD                           00015320
      DIMENSION          ER(3),EV(3),ERTO(3),FLTWID(3),RI(10)           00015330
      DATA FLTWID/7.7215,3.3090,0.2969/                                 00015340
C ****RI DATA STATEMENT UPDATED FEB 6,1986 BY M. MEYER ************
      DATA RI/120.,640.,1520.,2560.,5760.,11520.,23040.,43520.,         00015350
     2        49920.,1.8228E+6/,NRI/10/,PI/3.141592653/                 00015360
C                                                                       00015370
C    ******************************************                         00015380
C    * STEP 0: INITIALIZE BREAK-TRACK ALGORITHM *                       00015390
C    ******************************************                         00015400
C                                                                       00015410
C    STEP 0-1: INITIALIZE MOVING WINDOW-OF-8 REGISTERS.                 00015420
      DO 3 I=1,8                                                        00015430
    3 MBT(I)=0                                                          00015440
C                                                                       00015450
C    STEP 0-2: INITIALIZE SUM REGISTER.                                 00015460
      MBTSUM=0                                                          00015470
C                                                                       00015480
C    STEP 0-3: SET BREAK-TRACK FLAG TO LOW (OR 0) STATE.                00015490
      MBKTRK=0                                                          00015500
C                                                                       00015510
C    *****************************************                          00015520
C    * STEP 1: INITIALIZE ANGLE TRACKING LOOP *                         00015530
C    *****************************************                          00015540
      IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 5                                00015550
C                                                                       00015560
C    STEP 1-1: COMPUTE INITIAL INNER AND OUTER GIMBAL POSITIONS.        00015570
C      (NOTE: TRANSFORM CONSISTS OF TRANSLATION PLUS ROTATION.)         00015580
C      PERFORM TRANSLATION ---- SHIFT TO RADAR FRAME ORIGIN.            00015590
      DO 1 I=1,3                                                        00015600
    1 ERTO(I)=ERT(I)-DR(I)                                              00015610
C     TRANSFORM TARGET POSITION FROM BODY TO RADAR FRAME.               00015640
      CALL MULT31(TRB,ERTO,ER)                                          00015650
C     TRANSFORM TARGET VELOCITY FROM BODY TO RADAR FRAME.               00015660
      CALL MULT31(TRB,EVT,EV)                                           00015670
      SQ=SQRT(ER(2)*ER(2)+ER(3)*ER(3))                                  00015680
C     COMPUTE INNER(BETA) GIMBAL POSITION ---- BT.                      00015690
      IF(ER(1).EQ.0.0.AND.SQ.EQ.0.0) STOP                               00015700
      BT=-ATAN2(ER(1),SQ)                                               00015710
      ER2=-ER(2)                                                        00015720
      ER3=-ER(3)                                                        00015730
C     COMPUTE OUTER(ALPHA) GIMBAL POSITION ---- AL.                     00015740
      IF(ER2.EQ.0.0.AND.ER3.EQ.0.0) GO TO 8                             00015750
      AL=-ATAN2(ER2,ER3)                                                00015760
      GO TO 9                                                           00015770
    8 IF(ER(1).GT.0.0) AL=PI/2.                                         00015780
      IF(ER(1).LT.0.0) AL=-PI/2.                                        00015790
      IF(ER(1).EQ.0.0) STOP                                             00015800
C                                                                       00015810
C    STEP 1-2: COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH AND          00015820
C              ELEVATION RATES.                                         00015830
```

```
C     PRELIMINARY TRIGONOMETRIC COMPUTATIONS.
    9   CA=COS(AL)
        SA=SIN(AL)
        CB=COS(BT)
        SB=SIN(BT)
C     TRANSFORM BODY ANGULAR VELOCITY VECTOR FROM BODY TO OUTER
C     GIMBAL(G) REFERENCE FRAME.
        WGX=CP*EWB(1)+SP*EWB(2)
        WGY=CA*(-SP*EWB(1)+CP*EWB(2))+SA*EWB(3)
        WGZ=-SA*(-SP*EWB(1)+CP*EWB(2))+CA*EWB(3)
C     COMPUTE THE RANGE TO TARGET.
        R=SQRT(ER(1)*ER(1)+ER(2)*ER(2)+ER(3)*ER(3))
C     COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH RATE(AZRATE).
        VGY=CA*EV(2)+SA*EV(3)
        AZRATE=VGY/R+(CB*WGX-SB*WGZ)
C     COMPUTE INITIAL TARGET INERTIAL LOS ELEVATION RATE(ELRATE).
        ELRATE=-(CB*EV(1)-SB*(-SA*EV(2)+CA*EV(3)))/R+WGY
C
C   STEP 1-3: COMPUTAE INITIAL INNER AND OUTER GIMBAL RATES.
C     COMPUTE INITIAL OUTER GIMBAL RATE(ALRATE).
        RCB=R*CB
        IF(ABS(RCB).LT.1.0E-6) GO TO 2
        ALRATE=VGY/RCB
        GO TO 4
    2   ALRATE=0.
    4   CONTINUE
C     COMPUTE INITIAL INNER GIMBAL RATE(BTRATE).
        BTRATE=ELRATE-WGY
C
C   ****************************************
C   * STEP 2: INITIALIZE RANGE TRACKING LOOP *
C   ****************************************
C
C   STEP 2-1: TRANSFORM TARGET C.G. POSITION AND C.G. VELOCITY FROM
C             BODY TO ANTENNA LOS FRAME.
    5   CALL TRNSFM
        CALL PVTRAN
C
C   STEP 2-2: INITIALIZE THE RANGE ESTIMATE REGISTER.
        SRNG=CGRNGE
        IRNG=INTT(SRNG*3.2+0.5)
C
C   STEP 2-3: INITIALIZE THE RANGE RATE ESTIMATE REGISTER.
        IRDOT=INTT(CGVEL*TSAM*3.2+0.5)
C
C   ***********************************************************
C   * STEP 3: SET OPERATING PARAMETERS BASED UPON INITIAL RANGE *
C   *         AND SYSTEM MODE.                                *
C   ***********************************************************
C
C   STEP 3-1: DETERMINE CORRECT RANGE INTERVAL.
        DO 30 I=1,NRI
        MRNG=I
        IF(RI(I) .GT. SRNG) GO TO 40
30      CONTINUE
C
C   STEP 3-2: DETERMINE CORRECT SAMPLE RATE.
40      IF(IMODE.GE.2)  GO TO 44
        IF(MRNG.GT.9) GO TO 42
        MSAM=1
        GO TO 50
42      MSAM=2
        GO TO 50
44      IF(MRNG.GT.4) GO TO 46
```

```
      MSAM=1
      GO TO 50
46    MSAM=2
C
C  STEP 3-3: DETERMINE CORRECT PRF.
50    IF(IMODE.GE.2) GO TO 54
      IF(MRNG.GT.9) GO TO 52
      MPRF=1
      GO TO 60
52    MPRF=3
      GO TO 60
54    IF(MRNG.GT.9) GO TO 56
      MPRF=1
      GO TO 60
56    MPRF=2
60    CONTINUE
C
C  STEP 3-4: SET PRF TRANSITION FLAG.
      MPFOLD=MPRF
C
C  ****************************************
C  * STEP 4: INITIALIZE VELOCITY PROCESSOR *
C  ****************************************
C
C  STEP 4-1: INITIALIZE MOVING WINDOW VELOCITY AVERAGING.
      DO 10 I=1,4
10    VEST(I)=CGVEL*20.
C
C  STEP 4-2: SET INITIAL POSITION OF 5 DOPPLER FILTERS.
      VR=-CGVEL/FLTWID(MPRF)
      IVR=INTT(VR+0.5)+16000.
      XX=AMOD(IVR,32.)
      MDF(3)=INT(XX)
      DO 20 I=1,5
      MD=MDF(3)+I-3+160
20    MDF(I)=MOD(MD,32)
C
C  ******************************
C  * STEP 5: INITIALIZE AGC LOOP *
C  ******************************
      AGCO=1.0
      ITXP=1
C
C  ***************************************************************
C  * STEP 6: SET TRACK INDICATOR TO ALLOW OPERATION OF TRACK LOOP *
C  ***************************************************************
      MTKINT=1
C
      ROLD=0.
      ICLOSE=0
      ICLOLD=0
C
C  NOTE: DEBUGGING PRINT STATEMENTS.
C     WRITE(6,899)
C     WRITE(6,900) AZRATE,ELRATE,ALRATE,BTRATE,AL,BT
C     WRITE(6,901)
C     WRITE(6,902) IRNG,IRDOT,SRNG
C     WRITE(6,903)
C     WRITE(6,904) (VEST(I),I=1,4),(MDF(J),J=1,5)
C     WRITE(6,905)
C     WRITE(6,906) IMODE,MRNG,MSAM,MPRF
  899 FORMAT(//' TRACKER INITIALIZATION:'/' ATRACK: AZRATE',
     2 ',ELRATE,ALRATE,BTRATE,AL,BT')
  900 FORMAT(6F14.6)
```

```
 901  FORMAT(' RTRACK: IRNG,IRDOT,SRNG')
 902  FORMAT(2I8,F14.6)
 903  FORMAT(' VTRACK: VEST,MDF')
 904  FORMAT(4F14.6,5I8)
 905  FORMAT(' CNTL: IMODE,MRNG,MSAM,MPRF')
 906  FORMAT(4I8//)
      RETURN
      END
C
C
C    *********************************************************
C    * THIS SUBROUTINE SIMULATES THE TRACKING MODES OF THE KU-BAND *
C    * RADAR.                                                      *
C    *********************************************************
C
C
      SUBROUTINE TRACK
      COMMON /CNTL/IDUM(3),IASM,ISRCHC,ISRCHG,IAZS,IELS,ISLR,EDRNG,
     2               EDPA,EDRA
      COMMON /OUTPUT/MSWF,MTF,MSF,DUMO(7),IDUMO(4)
      COMMON /ICNTL/IIDUM(13),MTKINT,MRNG,MSAM,MPRF,MBKTRK,IDUM2(9)
      COMMON /SYSDAT/TSAM,DUM2(14)
      COMMON /ATDAT/DUM1(10),PREF,RREF,DUMA(2)
      DIMENSION SLWRTE(2)
      DATA SLWRTE/6.9814E-3,3.4907E-1/
C
C    ************************************************************
C    * STEP 1: INITIALIZE TRACK MODE ---- INITIALIZE ALL TRACK LOOPS *
C    *         AND UPDATE STATUS OF DATA VALID FLAGS.                 *
C    ************************************************************
C
C    STEP 1-1: IF TRACK LOOPS INITIALIZED(MTKINT=1) SKIP STEP 1-2 AND IF
C              ALL DATA VALID FLAGS ARE UP(MTF=1) SKIP STEP 1-2 AND 1-3.
      IF(MTF.EQ.1) GO TO 6
      IF(MTKINT.NE.0) GO TO 5
C
C    STEP 1-1: INITIALIZE RANGE,ANGLE,AND VELOCITY TRACK LOOPS ---- ASSUMES
C               STEADY STATE TRACKING OF TARGET C.G.
      CALL TKINIT
C
C    STEP 2-1: UPDATE DATA VALID FLAG STATUS ---- ONLY WHEN ENTERING
C              TRACK FROM SEARCH.
    5 CALL TGTACQ
C
C    ********************************************
C    * STEP 2: PERFORM TRACKING LOOP UPDATE PROCEDURE *
C    ********************************************
C
C    STEP 2-1: UPDATE TRANSFORMATION MATRICES AND MATRICE RATES.
    6 CALL TRNSFM
C
C    STEP 2-2: TRANSFORM TARGET POSITION AND VELOCITY COMPONENTS FROM
C              ORBITER BODY FRAME-TO-ANTENNA LOS FRAME.
      CALL PVTRAN
C
C    STEP 2-3: GENERATE NOISE-FREE TARGET RETURN SIGNAL AND PROCESS
C              SIGNAL TO PRODUCE NOISE-FREE DISCRIMINANT COMPONENTS.
      CALL SIGNAL
C
C    STEP 2-4: ADD EQUIVALENT NOISE TO DISCRIMINANT COMPONENTS AND FORM
C              ALL REQUIRED DISCRIMINANTS.
      CALL DISCRM
C
C    STEP 2-5: UPDATE STATUS OF BREAK-TRACK FLAG.
```

```
      CALL BRKTRK
C
C  STEP 2-6: CHECK STATUS OF BREAK-TRACK FLAG —— IF BREAK-TRACK FLAG
C            UP (MBKTRK=1) RESET SYSTEM AND RETURN TP SEARCH.
      IF(MBKTRK.NE.1) GO TO 7
      CALL SYSINT
      RETURN
C
C  STEP 2-7: DETERMINE RADAR SIGNAL STRENGTH (FOR DISPLAY METER)
C            AND UPDATE AGC VALUE.
    7 CALL RSS
C
C  STEP 2-8: UPDATE ANTENNA GIMBAL POSITIONS AND RATES AND TARGET
C            ANGLES AND ANGLE RATES FOR DISPLAY (GPC-ACQ AND AUTO
C            MODES ONLY.)
      IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 10
C
C  STEP 2-8A: IF IN GPC-ACQ OR AUTO MODE USE RADAR ESTIMATED TARGET
C             ANGLES AS GIMBAL TRACK SERVO INPUT.
      CALL ATRACK
      GO TO 15
   10 IF(IASM.EQ.4) GO TO 12
C
C  STEP 2-8B: IF IN GPC-DES MODE USE GPC-SUPPLIED ANGLE DESIGNATES AS
C             GIMBAL TRACK SERVO INPUT.
      PREF=EDPA
      RREF=EDRA
      CALL POINT
      GO TO 15
C
C  STEP 2-8C: IF IN MANUAL MODE USE CREW-SUPPLIED SLEW RATES TO DETER
C             MINE GIMBAL TRACK SERVO INPUT.
   12 PREF=PREF+FLOAT(IELS)*SLWRTE(ISLR+1)*TSAM
      RREF=RREF+FLOAT(IAZS)*SLWRTE(ISLR+1)*TSAM
      CALL POINT
C
C  STEP 2-9: UPDATE THE RANGE AND RANGE RATE ESTIMATES.
   15 CALL RTRACK
C
C  STEP 2-10: UPDATE ACCURATE VELOCITY ESTIMATE USING VELOCITY
C            PROCESSOR.
      CALL VELPRO
C
C  STEP 2-11: UPDATE ALL RADAR INTERNAL CONTROLS.
      CALL CNTRLS
   20 RETURN
      END
C
C
C  ****************************************************
C  * THIS SUBROUTINE UPDATES ALL REQUIRED TRANSFORMATION *
C  * MATRICES AND TRANSFORMATION MATRIX RATES.         *
C  ****************************************************
C
C
      SUBROUTINE TRNSFM
      COMMON /INPUT/DUM(9),TBT(3,3),TBTD(3,3)
      COMMON /SYSDAT/DUM2(4),CP,SP,DUM4(9),TRB(3,3)
      COMMON /ATDAT/CA,SA,CB,SB,DUM1(2),ALRATE,BTRATE,AL,BT,DUM3(4)
      COMMON /XFORMS/TLB(3,3),TLBD(3,3),TLT(3,3),TLTD(3,3)
      DIMENSION TLR(3,3)
C
C  ******************************************
C  * STEP 1: UPDATE TRANSFORMATION MATRICES *
```

```
C ****************************************
C
C STEP 1-1: PRELIMINARY COMPUTATIONS.
      CB=COS(BT)
      SB=SIN(BT)
      CA=COS(AL)
      SA=SIN(AL)
C
C STEP 1-2: COMPUTE TRANSFORMATION MATRIX TLB (BODY-TO-LOS FRAME).
      TLR(1,1)=CB
      TLR(1,2)=SB*SA
      TLR(1,3)=-SB*CA
      TLR(2,1)=0.0
      TLR(2,2)=CA
      TLR(2,3)=SA
      TLR(3,1)=SB
      TLR(3,2)=-CB*SA
      TLR(3,3)=CB*CA
      CALL MULT33(TLR,TRB,TLB)
C
C STEP 1-3: COMPUTE TRANSFORMATION MATRIX TLT (TARGET-TO-LOS FRAME).
      CALL MULT33(TLB,TBT,TLT)
C
C ********************************************
C * STEP 2: UPDATE TRANSFORMATION MATRIX RATES *
C ********************************************
C
C STEP 2-1: COMPUTE TLB-DOT.
      TLBD(1,1)=-BTRATE*TLB(3,1)+ALRATE*SB*TLB(2,1)
      TLBD(1,2)=-BTRATE*TLB(3,2)+ALRATE*SB*TLB(2,2)
      TLBD(1,3)=-BTRATE*TLB(3,3)+ALRATE*SB*TLB(2,3)
      TLBD(2,1)=ALRATE*SP*TLB(2,3)
      TLBD(2,2)=-ALRATE*CP*TLB(2,3)
      TLBD(2,3)=ALRATE*CA
      TLBD(3,1)=BTRATE*TLB(1,1)-ALRATE*CB*TLB(2,1)
      TLBD(3,2)=BTRATE*TLB(1,2)-ALRATE*CB*TLB(2,2)
      TLBD(3,3)=BTRATE*TLB(1,3)-ALRATE*CB*TLB(2,3)
C
C STEP 2-2: COMPUTE TLT-DOT.
      DO 20 I=1,3
      DO 20 J=1,3
      TLTD(I,J)=0.0
      DO 20 K=1,3
   20 TLTD(I,J)=TLTD(I,J)+TLBD(I,K)*TBT(K,J)+TLB(I,K)*TBTD(K,J)
      RETURN
      END
C
C
C ****************************************************************
C * THIS SUBROUTINE COMPUTES AN ACCURATE, SMOOTHED VELOCITY USING *
C * THE KU-BAND RADAR VELOCITY PROCESSOR ALGORITHM.                *
C ****************************************************************
C
C
      SUBROUTINE VELPRO
      REAL IRDOT,IRNG,INTT,IVEL,IVDISC,IFVEL,IRVEL,IR1,IR2,IR3,
     2     IF3,IDELTA
      COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)
      COMMON /OUTPUT/IDUM0(3),SRNG,SRDOT,DUM2(5),IDUM(4)
      COMMON /ICNTL/I1DUM(14),MRNG,MSAM,MPRF,IDUM1(10),MPFOLD
      COMMON /SYSDAT/TSAM,DUMS(14)
      COMMON /RTDAT/IRDOT,IRNG,RBIAS,VEST(4),MDF(5)
      COMMON /DSCRM/DUM(2),RDISC,VDSC,RRTE,ODISC,DUM3(3)
      DIMENSION IPROM(128),VT1(3),VT2(3),MW(4,3)
      DATA IPROM/127,127,125,124,122,121,120,118,117,116,114,113,       00027200
     2 111,110,109,107,106,105,103,102,101,99,98,97,95,94,93,92,90,     00027210
     3 89,88,87,85,84,83,82,81,79,78,77,76,75,73,72,71,70,69,68,67,     00027220
     4 66,65,64,63,62,61,60,59,58,57,56,55,54,53,52,51,50,49,49,48,     00027230
     5 47,46,45,44,44,43,42,41,41,40,39,38,38,37,36,36,35,34,34,33,     00027240
     6 32,32,31,31,30,30,29,28,28,27,27,26,26,25,25,24,24,23,23,22,     00027250
     7 22,22,21,21,20,20,19,19,19,18,18,17,17,17,16,16,16,15,15/        00027260
      DATA VT1/1.012592E-2,2.362726E-2,2.633237E-1/,VT2/1.204935,       00027270
     2        0.5163982,0.04633489/                                     00027280
      DATA MW/1,2,3,4,1,1,2,2,1,1,1,1/                                  00027282
C
C
C   ********************************************************
C   SUBROUTINE VELPRO WAS MODIFIED FEB 6 1986 BY M. MEYER
C   MODIFICATIONS CONSISTED OF CHECKING THE VARIABLE MPRF
C   FOR A VALUE OF ONE (IMPLIES 7 KC MODE) AND IF TRUE
C   ASSUMING THE VELOCITY ESTIMATE GIVEN BY THE VELOCITY
C   DISCRIMINANT IS UNAMBIGUOUS.
C   ********************************************************
C                                                                       00027290
C   ********************************************                        00027300
C   * STEP 1: GENERATE AMBIGUOUS VELOCITY ESTIMATE *                    00027310
C   ********************************************                        00027320
C                                                                       00027330
C   STEP 1-1: INTEGERIZE VELOCITY DISCRIMINANT AND CHECK FOR SATURATION. 00027340
      VDISC=5.333333*VDSC                                               00027350
      IVDISC=INTT(VDISC+0.5)                                            00027360
      IF(IVDISC.LT.-128.) IVDISC=-128.                                  00027370
      IF(IVDISC.GT.127.) IVDISC=127.                                    00027380
C                                                                       00027390
C   STEP 1-2: COMPUTE INTEGRAL FILTER NUMBER PORTION OF AMBIGUOUS       00027400
C             VELOCITY ESTIMATE.                                        00027410
      INTEG=MDF(2)                                                      00027420
      IF(IVDISC.LT.0.) INTEG=MOD(INTEG+1,32)                            00027430
C                                                                       00027440
C   STEP 1-3: COMPUTE FRACTIONAL FILTER PORTION OF AMBIGUOUS VELOCITY   00027450
C             ESTIMATE.                                                 00027460
C   ESTIMATE.                                                           00027470
      IV1=INT(ABS(IVDISC))+1                                            00027480
C ********************************
C CHANGED JAN 30 1986 BY H. MAGNUSSON
C ********************************
      IF(IV1.GT.128)IV1=128
      IFRAC=IPROM(IV1)                                                  00027490
      IF(IVDISC.LT.0.) IFRAC=127-IFRAC                                  00027500
C                                                                       00027510
C   STEP 1-4: COMPUTE AMBIGUOUS VELOCITY ESTIMATE --- COMBINE INTEGRAL  00027520
C             AND FRACTIONAL PARTS. NOTE: LSB IS 1/128 OF FILTER WIDTH. 00027530
C   FRACTIONAL PARTS. NOTE: LSB IS 1/128 OF A FILTER WIDTH.             00027540
      IFVEL=FLOAT(IFRAC+128*INTEG)                                      00027550
C   **************************
C   CHANGED FEB 6 1986 BY M. MEYER
C   **************************
C

       IF(MPRF.EQ.1) THEN
          IF(INTEG.GE.0.AND.INTEG.LE.21)THEN
            IRVEL=0.
          ELSE
            IRVEL=4096.
          END IF
          GO TO 8
       END IF
C   ***************************************                             00027570
C   * STEP 2: SCALE ROUGH VELOCITY ESTIMATE *                           00027580
```

```
C   ****************************************
C
C   STEP 2-1: SCALE LSB OF ROUGH RANGE RATE ESTIMATE TO 4 TIMES A DOPPLER
C             WIDTH.
C   DEFINITION:  VT1(MPRF)=(RANGE LSB)/((MAX. UNAMBIGUOUS VELOCITY)/8)
C                  OR VT1(MPRF)=5./(PRF*LAMBDA)
      R1=IRDOT*VT1(MPRF)/TSAM
      IR1=AINT(R1)
C
C   STEP 2-2: PERFORM SOME REQUIRED AUXILIARY CALCULATIONS.
      R2=IR1/8.
      IR2=AINT(R2)
      IRVEL=IR2*4096.
C
C   *****************************
C   * STEP 3: RESOLVE AMBIGUITY *
C   *****************************
C
C   STEP 3-1: COMPUTE 3 MSB'S OF AMBIGUOUS VELOCITY ESTIMATE.
      IF3=AINT(IFVEL/512.)
C
C   STEP 3-2: COMPUTE 3 LSB'S OF SCALED ROUGH RANGE RATE ESTIMATE.
      IR3=ABS(IR1-8.*IR2)
      IF(R1.LE.0.)GO TO 10
      IRVEL=IRVEL+4096.
      IR3=7.-IR3
   10 CONTINUE
C
C   STEP 3-3: COMPARE 3 MSB'S AND 3 LSB'S AND INCREMENT NUMBER OF
C             AMBIGUOUS FILTER BANK WIDTHS APPROPRIATELY.
      IDELTA=IR3-IF3
      IF(IDELTA.GE.4.) IRVEL=IRVEL-4096.
      IF(IDELTA.LE.-4.) IRVEL=IRVEL+4096.
    8 CONTINUE
C
C   *************************************************
C   * STEP 4: COMPUTE UNAMBIGUOUS VELOCITY ESTIMATE. *
C   *************************************************
C
C   STEP 4-1: ADD NUMBER OF AMBIGUOUS FILTER BANK WIDTHS TO ESTIMATE
C             OF FRACTIONAL FILTER BANK WIDTH. NOTE: LSB OF RESULTANT
C             ESTIMATE REPRESENTS 1/4096 OF A FILTER BANK WIDTH.
      IVEL=INTT(IRVEL-IFVEL)
C
C   STEP 4-2: SCALE LSB OF RESULTANT ESTIMATE TO 0.05 FEET/SEC.
C   DEFINITION:  VT2(MPRF)=((FILTER SEPARATION)/128.)/(VELOCITY LSB)
C                  OR VT2(MPRF)=(PRF*LAMBDA)/(0.05*8196).
      IVEL=INTT(IVEL*VT2(MPRF)+0.5)
C
C   *************************************************
C   * STEP 5: COMPUTE SMOOTHED UNAMBIGUOUS VELOCITY *
C   *************************************************
C
C   STEP 5-1: UPDATE REGISTERS OF MOVING WINDOW AVERAGER.
      DO 20 I=1,3
   20 VEST(5-I)=VEST(4-I)
      VEST(1)=IVEL
C
C   STEP 5-2: COMPUTE MOVING WINDOW AVERAGE AND SCALE ANSWER INTO
C             FEET/SEC FROM UNITS OF 0.05 FEET/SEC.
      M=MPRF
      M1=MW(1,M)
      M2=MW(2,M)
      M3=MW(3,M)
      M4=MW(4,M)
      SRDOT=0.0125*(VEST(M1      )+VEST(M2      )+VEST(M3      )+
     2              VEST(M4      ))
C
C    ***********************************
C    * STEP 6: RESET DOPPLER FILTER BANK *
C    ***********************************
C
C    STEP 6-1: USE ON-TARGET DISCRIMINANT AND VELOCITY DISCRIMINANT TO
C              DETERMINE UPDATE OF FILTER BANK POSITION.
C              THE FOLLOWING RULES ARE USED:
C
C              CASE 1: ODISC>0. AND -51.<IVDISC<51. IMPLIES NO CHANGE.
C
C              CASE 2: ODISC>0. AND IVDISC>51. IMPLIES SHIFT -1.
C
C              CASE 3: ODISC>0. AND IVDISC<-51. IMPLIES SHIFT +1.
C
C              CASE 4: ODISC<0. AND IVDISC>0. IMPLIES SHIFT -2.
C
C              CASE 5: ODISC<0. AND IVDISC<0. IMPLIES SHIFT +2.
      IF(ODISC.GE.0.) GO TO 30
      IF(IVDISC.LT.0.) MDF(1)=MOD(MDF(1)+2,32)
      IF(IVDISC.GE.0.) MDF(1)=MOD(MDF(1)+30,32)
      GO TO 40
   30 IF(IVDISC.GT.51.) MDF(1)=MOD(MDF(1)+31,32)
      IF(IVDISC.LT.-51.) MDF(1)=MOD(MDF(1)+1,32)
C
C    STEP 6-2: RESET REMAINING FILTERS IN THE BANK-OF-5.
   40 DO 50 I=1,4
   50 MDF(I+1)=MOD(MDF(1)+I,32)
      RETURN
      END
C
C
C    ***************************************************************
C    * THIS SUBROUTINE DETERMINES WHETHER ANTENNA IS IN ZONE 1 AND/OR *
C    * ZONE 0 (FOR GPC-ACQ AND GPC-DES POINTING MODES ONLY).          *
C    ***************************************************************
C
C
      SUBROUTINE ZONECK
      COMMON /CNTL/IDUMC(9),EDRNG,EDPA,EDRA
      COMMON /OUTPUT/IDUM1(3),DUM1(2),SPANG,SRANG,DUM3(3),IDUM3(4)
      COMMON /ICNTL/IDUM2(10),MZ1,MZ0,IDUM4(15)
      MZ0=0
      MZ1=1
      PII=3.141592653/180.
      RB=-PII*SRANG
      PB=-PII*SPANG
      P=-EDPA
      R=-EDRA
      CPB=COS(PB)
      SPB=SIN(PB)
      CRB=COS(RB)
      SRB=SIN(RB)
      CP=COS(P)
      SP=SIN(P)
      CR=COS(R)
      SR=SIN(R)
      ANGDIF=ACOS(SPB*CRB*SP*CR+SRB*SR+CPB*CRB*CP*CR)/PII
      ANGDIF=ABS(ANGDIF)
      IF(ANGDIF.GT.3.0) RETURN
      MZ0=1
      IF(ANGDIF.GT.0.3) RETURN                                          00012630
      MZ1=1                                                             00012640
      RETURN                                                            00012650
      END                                                               00012660
C
C  SES SMM MODEL AS OF JANUARY 13,1982
C
      SUBROUTINE SMM
C  II.     DIMENSION ARRAYS & DATA STATEMENTS
C     A) DIMENSION STATEMENTS
C       REAL KSEED
      COMMON /SATDAT/RADAR(3),KTAR,R(70,3),SIG(70),ROLD,ICLOSE,ICLOLD
      DIMENSION SIGMA(49),TARG(49,3),PHIMIN(49,3),PHIMAX(49,3)
      DIMENSION OFFSET(49),JHOT(49),JHOT20(49),PHI(49),FG(3)
      DIMENSION VECT(3),COSPHI(49,3),COSPHN(49),ORIENT(49,3)
      DIMENSION ALPH(19,3),V(19,3),DIM(19,3),WRAN(19,3),SDMAX(19,3)
      DIMENSION WSCALE(19,3),DPHI(19),PHIOLD(19),VOLD(19,3),KSEED(19,3)
      DIMENSION TTRAN(3),ABG(19,3),TMAX(49),PL(49),SDMIN(19,3)
C
C     B) DATA STATEMENTS
C
C       1. KSEED- SEEDS FOR RANDOM NUMBER GENERATOR "ZUDU".
      DATA KSEED/45,678,908,607,5678,897,345,7777,67,4,
     1 560,809,444,888,999,555,222,70,80,8000,
     2 5,15,25,35,45,55,65,75,85,95,
     3 7,17,27,37,47,57,67,77,87,97,
     4 9876,984,6666,2398,76,412,7589,409,899,561,
     5 205,3895,9457,9643,937,656,453/
C
C.      2. DIM- THE GENERAL SIZE OF EACH DIFFUSE SCATTERER.
      DATA DIM /57*64.8/
C
C       3. WSCALE- WEIGHTING ASSIGNED TO EACH SIDE OF A DIFFUSE
C                   SCATTERER.
      DATA WSCALE/8*10.84,5.9386,2*5.6804,5.9386,5.6804,4*11.1026,
     1 2*6.7958,
     2 2*6.9068,2*2.7111,2*3.6148,2*2.5174,4.3894,2*5.8095,4.3894,
     3 5.8095,4*17.8803,2*6.7958,19*0./
C
C       4. ORIENT- THE i,j,k COMPONENTS OF THE NORMAL VECTOR OF EACH
C                   TARGET.
C          a) i COMPONENT
      DATA ORIENT/13*0.,.9976,-.9976,.9976,-.9976,1.,-1.,
     1 23*0.,.9976,-.9976,.9976,-.9976,1.,2*-1.,
C          b) j COMPONENT
     2 1.,-1.,2*.6428,2*0.,-.6494,-.6361,1.,.4924,.8704,.6428,-1.,.0637,
     3 2*-.0637,.0637,2*0.,1.,-1.,2*.6428,.9272,.5150,.2924,2*0.,-.6494,
     4 -.6361,2*0.,2*1.,.4924,.8704,.4924,.866,-.8660,-1.,0.,-.6428,
     5 .0637,2*-.0637,.0637,3*0.,
C          c) k COMPONENT
     6 2*0.,-.766,.766,1.,-1.,-.7604,.7716,0.,-.8704,.4924,.766,0.,
     7 .0284,2*-.0284,.0284,4*0.,-.766,.766,.3746,.8572,.9563,1.,-1.,
     8 -.7604,.7716,2*0.,2*0.,-.8704,.4924,.8704,-.5,.5,0.,1.,.766,
     9 .0284,2*-.0284,.0284,3*0./
C
C       5. ABG- ARRAY OF TRANSFORMATION ANGLES(RAD), ALPHA, BETA,
C               GAMMA, FOR DIFFUSE SCATTERERS.
C          a) ALPHA
      DATA ABG/4*3.141593,2*1.570796,2*0.,4*3.141593,0.,1.634563,
     1 -1.50703,1.50703,4.648623,1.570796,4.712389,
C          b) BETA
     2 2*1.570796,2.443392,.6982,0.,3.141593,2.434725,.689444,
     3 1.570796,2.626811,1.055951,.6982,1.570796,1.542392,
     4 2*1.5992,1.542392,2*1.570796,
C          c) GAMMA
     5 4*3.141593,2*1.570796,2*0.,4*3.141593,0.,2*2.723729,.4178642,
     6 2.723729,2*1.570796/
C
C        6. SIGMA- THE CALCULATED RCS FOR EACH TARGET IN M**2.
       DATA SIGMA/2*.1,2*.0154,2*.0274,2*.0133,.0121,2*.0194,.0121,
     2 .0194,4*.7026,2*.0606,2*2419.,373.,7.25,21.84,11.14,18.83,
     3 2*663.,2*321.,2*3.63,.92,.97,470.,82.13,470.,2*83.,470.,83.,
     4 6.34,4*16995.,2*146615.,.3322/
C
C        7. TARG- TARGET POSITION (IN X,Y,Z COORDINATES) RELATIVE TO
C                 THE COORDINATE AXIS OF SMM.
C          a) X COORDINATE
       DATA TARG /9*1.394,4*-.774,.270,.231,.270,.231,2.491,-1.497,
     2 3*1.394,.542,3*1.626,4*1.394,2*0.,-.413,-1.149,8*-.774,.270,
     3 .231,.270,.231,2.491,2*-1.497,
C          b) Y COORDINATE
     4 .862,-.862,2*.555,2*0.,2*.555,.748,.439,1.097,-.3614,-.955,
     5 2*2.233,2*-2.233,2*0.,.826,-.826,2*.555,.658,.568,.439,2*0.,
     6 2*.555,2*0.,2*.748,.439,.865,1.097,.865,-.207,-.955,-.684,
     7 -.3614,2*2.233,2*-2.233,3*0.,
C          c) Z COORDINATE
     8 2*0.,-.929,.929,1.058,-1.058,-.878,.878,0.,-.774,.852,.645,
     9 0.,2*.620,2*-.620,4*0.,-.929,.929,.826,.930,.994,1.058,
     A -1.058,-.878,.878,4*0.,-.774,-.258,.852,.272,.903,0.,.581,
     B .645,2*.620,2*-.620,3*0./
C
C        8. PHIMIN- MINIMUM ANGLE OF DEVIATION FROM SMM COORDINATES
C                   RELATIVE TO TARGET NORMAL.
C          a) MINIMUM ANGLE SUBTENDED IN X-DIRECTION
       DATA PHIMIN /13*0.,2.5,174.5,2.5,174.5,0.,90.,11*88.5,
     2 2*89.2,10*88.5,2.5,174.5,2.5,174.5,0.,2*178.5,
C          b) MINIMUM ANGLE SUBTENDED IN Y-DIRECTION
     3 0.,90.,2*48.5,2*0.,129.,128.,0.,59.,149.,128.5,90.,22.5,
     4 2*154.5,22.5,2*0.,0.,178.5,2*48.5,20.5,57.5,71.5,2*88.5,129.,
     5 128.,0.,90.,2*0.,59.,2*149.,2*148.5,178.5,88.5,128.5,22.5,
     6 2*154.5,22.5,3*88.5,
C          c) MINIMUM ANGLE SUBTENDED IN Z-DIRECTION
     7 2*0.,138.5,38.5,0.,90.,138.,38.,0.,149.,59.,38.5,0.,64.5,
     8 2*112.5,64.5,2*0.,2*88.5,138.5,38.5,66.5,29.5,15.5,0.,178.5,
     9 138.,38.,2*0.,2*88.5,149.,2*58.,118.5,58.5,88.5,0.,
     A 38.5,64.5,2*112.5,64.5,3*88.5/
C
C        9. PHIMAX- MAXIMUM ANGLE OF DEVIATION FROM SMM COORDINATES
C                   RELATIVE TO TARGET NORMAL.
C          a) MAXIMUM ANGLE SUBTENDED IN X-DIRECTION
       DATA PHIMAX /13*180.,5.5,177.5,5.5,177.5,90.,180.,11*91.5,
     2 2*90.8,10*91.5,5.5,177.5,5.5,177.5,1.5,2*180.,
C          b) MAXIMUM ANGLE SUBTENDED IN Y-DIRECTION
     3 90.,180.,2*51.5,2*180.,132.,131.,1.5,62.,152.,131.5,180.,25.5,
     4 2*157.5,25.5,2*180.,1.5,180.,2*51.5,23.5,60.5,74.5,2*91.5,132.,
     5 131.,90.,180.,2*1.5,62.,2*152.,2*151.5,180.,91.5,131.5,25.5,
     6 2*157.5,25.5,3*91.5,
C          c) MAXIMUM ANGLE SUBTENDED IN Z-DIRECTION
     7 2*180.,141.5,41.5,90.,180.,141.,41.,180.,152.,62.,41.5,180.,67.5,
     8 2*115.5,67.5,2*180.,2*91.5,141.5,41.5,69.5,32.5,18.5,1.5,180.,
     9 141.,41.,2*180.,2*91.5,152.,2*62.,121.5,61.5,91.5,1.5,41.5,67.5,
     A 2*115.5,67.5,3*91.5/
C
C       10. OFFSET- POSITION OF TARGET SPECULAR PT. RELATIVE TO TARGET
C                   COORDINATES.
       DATA OFFSET /17*0.,2*0.,11*0.,.7486,.8,14*0.,2*0.,.6518/
C
C       11. MISCELLANIOUS
```

```
      DATA PL/ 30*1.,2*0.,16*1.,0./
      DATA TMAX/19*90.,11*1.5,2*0.,16*1.5,0./
      DATA NTAR/49/,KWIDE/19/,PI/3.141592653/
      DATA TTRAN/3*0.0/,INIT1/1/
      IF(INIT1.NE.1) GO TO 2
C
C        12. SDMIN- MINIMUM ANGLE OF VIEW; TARGET SHADOWING.
C            a) X-COORDINATE
     DATA SDMIN/2*-0.6828,-1.,-0.7467,2*-1.,-0.7467,12*-1.,
C            b) Y-COORDINATE
     1        19*-1.,
C            c) Z-COORDINATE
     2        19*-1./
C
C         13. SDMAX- MAXIMUM ANGLE OF VIEW; TARGET SHADOWING.
C            a) X-COORDINATE
     DATA SDMAX/8*1.,0.4218,3*1.,0.4218,0.5037,0.6046,0.5037,0.6046,
     1        2*1.,
C            b) Y-COORDINATE
     2        19*1.,
C            c) Z-COORDINATE
     3        19*1./
C
C
C III.      RANDOMIZE DIFFUSE SCATTERER RCS VALUES.
C
      ISEED1=100
      ISEED2=83
      DO 107 I=1,1000
107   X=RNDU(ISEED1,ISEED2)
      DO 108 I=1,KWIDE
      X=RNDU(ISEED1,ISEED2)
108   SIGMA(I)=SIGMA(I)*2.*X
C
C  IV.      CONVERT TARGET DATA APPROPRIATELY.
C
      FTM=0.3048
      DO 101 I=1,NTAR
101   SIGMA(I)=SQRT(SIGMA(I))/FTM
      DO 102 J=1,NTAR
      DO 102 I=1,3
102   TARG(J,I)=TARG(J,I)/FTM
      DO 103 J=1,NTAR
      TMAX(J)=COS( TMAX(J)*PI/180.)
      DO 103 I=1,3
      PHIMIN(J,I)=COS(PHIMIN(J,I)*PI/180.)
103   PHIMAX(J,I)=COS(PHIMAX(J,I)*PI/180.)
      DO 105 I=1,NTAR
105   OFFSET(I)=OFFSET(I)/FTM
C
C   V.      INITIALIZATION OF TARGET POSITION & COUNTING PARAMETERS
C     NWIDE & KTAR.
C
      DO 1 K=1,NTAR
      DO 1 I=1,3
    1 TARG(K,I)=TARG(K,I)+TTRAN(I)
       INIT1=0
    2 CONTINUE
      NWIDE=0
      KTAR=0
C
C  VI.      DETERMINE WHICH TARGETS ARE ILLUMINATED.
C
C     WRITE(2,500)
```

```
  500      FORMAT(1X,'TARGET #',2X,'COSPHN')
      DO 15 I=1,NTAR
C
C     A) DETERMINE THE POSITION OF THE RADAR RELATIVE TO
C        TARGET SPECULAR POINT.
C
C       1. "VECT"- POSITION VECTOR
      DO 5 J=1,3
      VECT(J)=RADAR(J)-TARG(I,J)
    5 CONTINUE
C       2. VNORM- MAGNITUDE OF "VECT".
      VNORM=SQRT(VECT(1)**2+VECT(2)**2+VECT(3)**2)
C
C     B) DETERMINE THE COSINE OF THE ANGLE BETWEEN THE
C        RADAR POSITION RELATIVE TO THE TARGET SPECULAR PT. &
C        TARGET NORMAL.
C
C       1. CALCULATE THE ANGLE BY EMPLOYING THE DOT PRODUCT
C          OF THE TWO VECTORS: "COSPHI" & "ORIENT".
      DP=0.
      DO 7 J=1,3
C       2. COSPHI- UNIT VECTOR OF "VECT"; REPRESENTATIVE OF THE
C          COSINE OF THE ANGLE BETWEEN "VECT" & SMM COORDINATE AXIS.
      COSPHI(I,J)=VECT(J)/VNORM
    7 DP=DP+COSPHI(I,J)*ORIENT(I,J)
C       3. COSPHN- COSINE OF THE ANGLE; RESULT OF THE DOT PRODUCT.
      COSPHN(I)=DP
C
C     C) TEST OF ILLUMINATION- TWO METHODS: COMPARE COSPHN W/TMAX
C        OR COMPARE COMPONENTS OF COSPHI W/PHIMIN & PHIMAX.
C
C       1. PL- A FLAG: 0 INDICATES METHOD 1 & 1 INDICATES METHOD 2.
      IF(PL(I).EQ.0.)GO TO 9
C       2. METHOD 1
      IF(COSPHN(I).LT.TMAX(I))GO TO 15
      GO TO 11
C       3. METHOD 2
    9 DO 10 J=1,3
      IF(COSPHI(I,J).LT.PHIMAX(I,J).OR.COSPHI(I,J).GT.PHIMIN(I,J))
     2 GO TO 15
   10 CONTINUE
C
C     D) TARGET SHADOWING
C
C       1. TEST FIRST 19 TARGETS ONLY.
   11      IF(I.GT.19)GO TO 13
C       2. FIND SHADOWING VECTOR BY TRANSFORMATION OF COSPHI
C          FROM SMMS TO TARGET COORDINATES.
     F1=COSPHI(I,1)*COS(ABG(I,1))+COSPHI(I,2)*SIN(ABG(I,1))
     F2=COSPHI(I,2)*COS(ABG(I,1))-COSPHI(I,1)*SIN(ABG(I,1))
     F3=COSPHI(I,3)
     FB2=F2*COS(ABG(I,2))+F3*SIN(ABG(I,2))
     FB3=F3*COS(ABG(I,2))-F2*SIN(ABG(I,2))
     FG(1)=F1*COS(ABG(I,3))+FB2*SIN(ABG(I,3))
     FG(2)=FB2*COS(ABG(I,3))-F1*SIN(ABG(I,3))
     FG(3)=FB3
C       3. TEST FOR TARGET SHADOWING.
     DO 12 J=1,3
     IF(FG(J).GT.SDMAX(I,J).OR.FG(J).LT.SDMIN(I,J))GO TO 15
   12      CONTINUE
C
C     E) COUNT NUMBER OF ILLUMINATED TARGETS.
C
C       1. KTAR- # OF TARGETS ILLUMINATED
```

```
   13 KTAR=KTAR+1
C       2. JHOT- TARGET IDENTIFICATION NUMBER
      JHOT(KTAR)=I
      SIG(KTAR)=SIGMA(I)
C       3. NWIDE- # OF DIFFUSE SCATTERERS
      IF(I.LE.KWIDE) NWIDE=NWIDE+1
C     WRITE(2,100)I,COSPHN(I)
  100     FORMAT(1X,I3,7X,F6.3)
   15 CONTINUE
C
C VII.      UPDATE RANGE OF RADAR RELATIVE TO EACH TARGETS SPECULAR PT.
C
C     A) RANGE UPDATE
C
      DO 20 K=1,KTAR
      I=JHOT(K)
      DO 20 J=1,3
      R(K,J)=TARG(I,J)+OFFSET(I)*COSPHI(I,J)
   20 CONTINUE
     IEE=1
     IF (IEE.EQ.0)GO TO 24
C
C     B) RE-EVALUATE RCS FOR DIFFUSE SCATTERERS
C
      DO 22 K=1,NWIDE
      I=JHOT(K)
      SIG(K)=SQRT(ABS(COSPHN(I)))*SIGMA(I)
   22 CONTINUE
   24 RANGE=SQRT(RADAR(1)**2+RADAR(2)**2+RADAR(3)**2)
C
C     C) TEST FOR CLOSE RANGE
C
      IF((ROLD.LT..01.OR.RANGE-ROLD.LE.0.).AND.RANGE.LE.270.) ICLOSE=1
      IF(RANGE-ROLD.GT.0..AND.RANGE.GT.300.) ICLOSE=0
C      ICLOSE=0
      IF(ICLOSE.EQ.0.OR.NWIDE.EQ.0) GO TO 55
      IF(ICLOLD.EQ.1) GO TO 35
C
C     D) RANGE UPDATE FOR DIFFUSE SCATTERERS
C
C       1. PERFORMS INITIALIZATION OF DIFFERENCE EQUATIONS
C          FOR ALL DIFFUSE SCATTERERS.
C
      DO 30 I=1,KWIDE
      IF(COSPHN(I).GT.1.)COSPHN(I)=1.
      PHIOLD(I)=ACOS(COSPHN(I))
C         a) "V"- WANDERING VECTOR
      DO 25 J=1,3
      V(I,J)=WSCALE(I,J)*(ZUDU(KSEED(I,J))-.5)
      VOLD(I,J)=V(I,J)
   25 CONTINUE
C
C         b) TRANSFORMATION OF "V" FROM TARGET COORDINATES TO
C            SMMS COORDINATES.
      TGAM1=V(I,1)*COS(ABG(I,3))-V(I,2)*SIN(ABG(I,3))
      TGAM2=V(I,1)*SIN(ABG(I,3))+V(I,2)*COS(ABG(I,3))
      TBETA2=COS(ABG(I,2))*TGAM2-SIN(ABG(I,2))*V(I,3)
      TBETA3=SIN(ABG(I,2))*TGAM2+COS(ABG(I,2))*V(I,3)
      V(I,1)=COS(ABG(I,1))*TGAM1-SIN(ABG(I,1))*TBETA2
      V(I,2)=SIN(ABG(I,1))*TGAM1+COS(ABG(I,1))*TBETA2
      V(I,3)=TBETA3
      DO 26 J=1,3
      R(I,J)=R(I,J)+V(I,J)
   26 CONTINUE
   30 CONTINUE
      GO TO 55
C
C        2. UPDATES THE ANGLE BETWEEN THE RADAR VECTOR & THE
C           TARGET NORMAL.
   35 DO 40 I=1,KWIDE
      PHI(I)=ACOS(COSPHN(I))
      DPHI(I)=(PHI(I)-PHIOLD(I))
      PHIOLD(I)=PHI(I)
   40 CONTINUE
C
C        3. UPDATES THE RANGE COMPONENTS DUE TO RADAR BEAM
C           DEFLECTION OVER THE SURFACE OF THE DIFFUSE SCATTERER.
C           THE TRANSFORMATION PERFORMS THE SAME FUNCTION DESCRIBED
C           PREVIOUSLY.
      DO 50 K=1,NWIDE
      I=JHOT(K)
      DO 45 J=1,3
      ALPH(I,J)=EXP(-DIM(I,J)*ABS(DPHI(I)*COSPHN(I)))
      WRAN(I,J)=SQRT(1.-ALPH(I,J)**2)*WSCALE(I,J)*(ZUDU(KSEED(I,J))-.5)
      V(I,J)=ALPH(I,J)*VOLD(I,J)+WRAN(I,J)
      VOLD(I,J)=V(I,J)
   45 CONTINUE
      TGAM1=V(I,1)*COS(ABG(I,3))-V(I,2)*SIN(ABG(I,3))
      TGAM2=V(I,1)*SIN(ABG(I,3))+V(I,2)*COS(ABG(I,3))
      TBETA2=COS(ABG(I,2))*TGAM2-SIN(ABG(I,2))*V(I,3)
      TBETA3=SIN(ABG(I,2))*TGAM2+COS(ABG(I,2))*V(I,3)
      V(I,1)=COS(ABG(I,1))*TGAM1-SIN(ABG(I,1))*TBETA2
      V(I,2)=SIN(ABG(I,1))*TGAM1+COS(ABG(I,1))*TBETA2
      V(I,3)=TBETA3
      DO 46 J=1,3
      R(K,J)=R(K,J)+V(I,J)
   46 CONTINUE
   50 CONTINUE
   55 CONTINUE
      ROLD=RANGE
      ICLOLD=ICLOSE
      RETURN
      END
C
C
C
      FUNCTION ZUDU(KSEED)
C     THIS SUBROUTINE GENERATES RANDOM NUMBERS.
      DATA MU/524287/,XMU/524287./,IETA/997/
      IF(KSEED) 20,10,20
   20 CONTINUE
      KSEED=IETA*KSEED
      IKEEP=KSEED/MU
      KSEED=KSEED-IKEEP*MU
      XRAN=KSEED
      XRAN=XRAN/MU
      ZUDU=XRAN
   10 RETURN
      END
       subroutine readPAT

c      ------------------------------------------------------------
c
c              Read in the sum, phase, and difference patterns
c
c      ------------------------------------------------------------

       real allinear( 41,41 ), ellinear( 41,41 )
      real sa1linear( 41,41 ), se1linear( 41,41 )

      real pa1linear( 41,41 ), pe1linear( 41,41 )

      common / linear / a1linear, e1linear

      common / linear1 / sa1linear, se1linear

      common / linear2 / pa1linear, pe1linear


      open( unit=3, file='[KUBAND.HOWARD.MARK]az1d.dat',
     1       access='sequential', form='unformatted',
     1       status='old', readonly )
      read( 3 ) ( ( a1linear( i,j ), j = 1,41 ), i = 1,41 )
      close( 3 )

      open( unit=3, file='[KUBAND.HOWARD.MARK]el1d.dat',
     1       access='sequential', form='unformatted',
     1       status='old', readonly )
      read( 3 ) ( ( e1linear( i,j ), j = 1,41 ), i = 1,41 )
      close( 3 )

      open( unit=3, file='[KUBAND.HOWARD.MARK]az1s.dat',
     1       access='sequential', form='unformatted',
     1       status='old', readonly )
      read( 3 ) ( ( sa1linear( i,j ), j = 1,41 ), i = 1,41 )
      close( 3 )

      open( unit=3, file='[KUBAND.HOWARD.MARK]el1s.dat',
     1      access='sequential', form='unformatted',
     1      status='old', readonly )
      read( 3 ) ( ( se1linear( i,j ), j = 1,41 ), i = 1,41 )
      close( 3 )

      open( unit=3, file='[KUBAND.HOWARD.MARK]az1p.dat',
     1      access='sequential', form='unformatted',
     1      status='old', readonly )
      read( 3 ) ( ( pa1linear( i,j ), j = 1,41 ), i = 1,41 )
      close( 3 )

      open( unit=3, file='[KUBAND.HOWARD.MARK]el1p.dat',
     1      access='sequential', form='unformatted',
     1      status='old', readonly )
      read( 3 ) ( ( pe1linear( i,j ), j = 1,41 ), i = 1,41 )
      close( 3 )

      return
      end
c     ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
c
c
c       Subroutine: Antenna pattern interpolation.
c       Input: Azimuth and elevation angles in degrees.
c       Output: Interpolated difference, sum, and phase values
c               for all 18 antenna patterns.
c
c
c       ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

        subroutine interp( az, el)

c       ____________________________________________________________
c
c        Linearly interpolate the gain, phase and difference patterns
c
c       ____________________________________________________________

        real a1linear( 41,41 ), e1linear( 41,41 )

        real sa1linear(41,41), se1linear(41,41)

        real pa1linear(41,41), pe1linear(41,41)

        common / linear / a1linear, e1linear

        common / linear1 / sa1linear,se1linear

        common / linear2 / pa1linear,pe1linear

        common / SUDIPH / X,Y,Z,PAZ,PEL

        iax = jint( ( az + 4. ) * 5. )
        iex = jint( ( el + 4. ) * 5. )
        az0 = floatj( iax ) / 5. - 4.
        el0 = floatj( iex ) / 5. - 4.

        iaz = jint ( ( az + 4. ) * 5. ) + 1
        jel = jint ( ( el + 4. ) * 5. ) + 1

c       -------- find azd values ------------------------------

        f0 = 10.**( a1linear( iaz,jel     ) /20. )
        f1 = 10.**( a1linear( iaz+1,jel   ) /20. )
        f2 = 10.**( a1linear( iaz,jel+1   ) /20. )
        f3 = 10.**( a1linear( iaz+1,jel+1 ) /20. )

        fa = f0 + (f1-f0)/.2 * ( az-az0 )
        fb = f2 + (f3-f2)/.2 * ( az-az0 )
        fx = fa + (fb-fa)/.2 * ( el-el0 )

        Y = fx
c       -------- find eld values ------------------------------

        f0 = 10.**( e1linear( iaz,jel     ) /20. )
        f1 = 10.**( e1linear( iaz+1,jel   ) /20. )
        f2 = 10.**( e1linear( iaz,jel+1   ) /20. )
        f3 = 10.**( e1linear( iaz+1,jel+1 ) /20. )

        fa = f0 + (f1-f0)/.2 * ( az-az0 )
        fb = f2 + (f3-f2)/.2 * ( az-az0 )
        fx = fa + (fb-fa)/.2 * ( el-el0 )
```

```
      Z =  fx
c     -------- find azs values ------------------------------
      f0 = 10.**(sa1linear(iaz  ,jel   )/20.)
      f1 = 10.**(sa1linear(iaz+1,jel   )/20.)
      f2 = 10.**(sa1linear(iaz  ,jel+1 )/20.)
      f3 = 10.**(sa1linear(iaz+1,jel+1 )/20.)
      fa = f0 +(f1-f0)/.2*(az-az0)
      fb = f2 +(f3-f2)/.2*(az-az0)
      fx = fa +(fb-fa)/.2*(el-el0)

      X = fx
c     -------- find azp values ------------------------------
      f0 = pa1linear(iaz  ,jel   )
      f1 = pa1linear(iaz+1,jel   )
      f2 = pa1linear(iaz  ,jel+1 )
      f3 = pa1linear(iaz+1,jel+1 )
      fa = f0 +(f1-f0)/.2*(az-az0)
      fb = f2 +(f3-f2)/.2*(az-az0)
      fx = fa +(fb-fa)/.2*(el-el0)

      PAZ=fx                  !  phase in degrees
c     ------- find elp values -------------------------------
      f0 = pe1linear(iaz  ,jel   )
      f1 = pe1linear(iaz+1,jel   )
      f2 = pe1linear(iaz  ,jel+1 )
      f3 = pe1linear(iaz+1,jel+1 )

      fa = f0 +(f1-f0)/.2*(az-az0)
      fb = f2 +(f3-f2)/.2*(az-az0)
      fx = fa +(fb-fa)/.2*(el-el0)

      PEL=fx                   ! phase in degrees

      return

      end
```

APPENDIX C

LINE BY LINE LISTING OF DIFFERENCES BETWEEN

BASELINE PROGRAM AND DELIVERABLE PROGRAM

This appendix lists the lines which have been deleted from the baseline program and those which were added to form the deliverable program.

The deleted and added lines are grouped by program module, and identified by line number and the labels "LINES DELETED FROM BASELINE PROGRAM" or "LINES ADDED TO DELIVERABLE PROGRAM" immediately preceeding the lines deleted or added. The line numbers for the deleted lines refer to lines in the original baseline program. The line numbers identifying the added lines are the line numbers in the final, deliverable program.

```
************
LINES ADDED TO DELIVERABLE PROGRAM
C                 *******************************************
C
C                      MODIFIED 01/27/86 TO COMPUTE AND
C                      PLOT REF. RANGE ACCELERATION.
C
C                 ******************************************
C
C************************************************************************************
C       MDMIN - KUBAND DATA : SSRNG, SSRDOT, SSRANG, SSPANG, SSRRTE, SSPRTE,
C                             SSALP, SSBET
C
C       WHITE SANDS - REF DATA : X, Y, Z, VX, VY, VZ
C
C       REF -> TMR2KU -> ACT : R, ARDOT, SPANG, SRANG, SRRTE, SPRTE,
C                              SALF, SBTA, SAZRTE, SELRTE
C
C       REF -> TMR2KU -> SIM : HRNG, HRDOT, HRANG, HPANG, HRRTE, HPRTE,
C                              HALP, HBET, HELRTE, HALRTE
C
C ***********************************************************************************
      COMMON /TARGET/ITARG,SRCS
      COMMON /ACTDAT/R,ARDOT,SPANG,SRANG,SPRTE,SRRTE,AL,BT,SALF,SBTA
     1 ,ER(3),EV(3),ERTO(3),AZRATE,ELRATE,SAZRTE,SELRTE
     2 ,AX,AY,AZ,AAX,AAY,AAZ,RACCEL
      COMMON /TERM/ITERM,XMO,XDAY,XYR,TBIAS,XJMO,XJDAY,XJYR
      COMMON /OUTPUT/MSWF,MTF,MSF,HRNG,HRDOT,HPANG,HRANG,HPRTE
     2              ,HRRTE,HRSS,MADVF,MRDVF,MARDVF,MRRDVF
     3              ,HALP,HBET
      COMMON /SYSDAT/TS,DUM2(14)
       COMMON /TMR/X,Y,Z,VX,VY,VZ,
     1             DLP(3),DEL(3),DUE(3),
     2             DSU(3),THAZL1,THEL1,THAZU1,A23
      COMMON /INPUT/RO(3),VO(3),EWB(3)
      COMMON /ICNTL/IDUM(16),MPRF
       CHARACTER ANS,REPLY
       CHARACTER*11 FPRO(57)
       CHARACTER*40 IXT,LPRO(57)
       CHARACTER*80 COMMENT
       CHARACTER*11 UNIT7
       INTEGER IREF
       INTEGER*2 IS1,IS2
       DIMENSION TP(2001),D(2001,43)
       DIMENSION ITILT(10)
       DIMENSION RNEW(3),ROLD(3),VNEW(3),VOLD(3)
       BYTE IC(120)
C
C      TEST DATA FROM WS32TDATA1
c
```

```
      DATA LPRO(1)/' SIM DATA PROFILE HL146AB$'/
      DATA LPRO(2)/' SIM DATA PROFILE HL246AB$'/
      DATA LPRO(3)/' SIM DATA PROFILE HJ146AB$'/
      DATA LPRO(4)/' SIM DATA PROFILE HEL30AB$'/
      DATA LPRO(5)/' SIM DATA PROFILE H30SKAB$'/
      DATA LPRO(6)/' SIM DATA PROFILE H30SKAC$'/
      DATA LPRO(7)/' SIM DATA PROFILE HEL30AC$'/
      DATA LPRO(8)/' SIM DATA PROFILE HEL30AD$'/
      DATA LPRO(9)/' SIM DATA PROFILE HL246AC$'/
      DATA LPRO(10)/' SIM DATA PROFILE HL346AB$'/
      DATA LPRO(11)/' SIM DATA PROFILE HL446AB$'/
      DATA LPRO(12)/' SIM DATA PROFILE HL546AB$'/
      DATA LPRO(13)/' SIM DATA PROFILE HL546AC$'/
      DATA LPRO(14)/' SIM DATA PROFILE HL246AD$'/
      DATA LPRO(15)/' SIM DATA PROFILE HL446AC$'/
      DATA LPRO(16)/' SIM DATA PROFILE HL146AC$'/
      DATA LPRO(17)/' SIM DATA PROFILE HL346AD$'/
      DATA LPRO(18)/' SIM DATA PROFILE HJ146AC$'/
      DATA LPRO(19)/' SIM DATA PROFILE HEL30AE$'/
      DATA LPRO(20)/' SIM DATA PROFILE HEL30AF$'/
      DATA LPRO(21)/' SIM DATA PROFILE H30SKAD$'/
      DATA LPRO(22)/' SIM DATA PROFILE H30SKAE$'/
      DATA LPRO(23)/' SIM DATA PROFILE H30SKAF$'/
      DATA LPRO(24)/' SIM DATA PROFILE HEL30AG$'/
      DATA LPRO(25)/' SIM DATA PROFILE HEL30AH$'/
      DATA LPRO(26)/' SIM DATA PROFILE H30SKAG$'/
      DATA LPRO(27)/' SIM DATA PROFILE H30SKAH$'/
      DATA LPRO(28)/' SIM DATA PROFILE H30SKAI$'/
      DATA LPRO(29)/' SIM DATA PROFILE HEL30AI$'/
      DATA LPRO(30)/' SIM DATA PROFILE HEL30AJ$'/
      DATA LPRO(31)/' SIM DATA PROFILE HL546AE$'/
      DATA LPRO(32)/' SIM DATA PROFILE HL246AE$'/
      DATA LPRO(33)/' SIM DATA PROFILE HL446AD$'/
      DATA LPRO(34)/' SIM DATA PROFILE HL146AD$'/
      DATA LPRO(35)/' SIM DATA PROFILE HL346AE$'/
      DATA LPRO(36)/' SIM DATA PROFILE HJ146AD$'/
      DATA LPRO(37)/' SIM DATA PROFILE HL546AF$'/
      DATA LPRO(38)/'TSS SIM DATA PROFILE GEM1$'/
      DATA LPRO(39)/'TSS SIM DATA PROFILE GEM2$'/
      DATA LPRO(40)/'TSS SIM DATA PROFILE GEM3$'/
      DATA LPRO(41)/'TSS SIM DATA PROFILE SAT1$'/
      DATA LPRO(42)/'TSS SIM DATA PROFILE SAT2$'/
      DATA LPRO(43)/'TSS SIM DATA PROFILE SAT3$'/
      DATA LPRO(44)/'TSS SIM DATA PROFILE SAT4$'/
      DATA LPRO(45)/'TSS SIM DATA PROFILE SAT6$'/
      DATA LPRO(46)/'TSS SIM DATA PROFILE SAT8$'/
      DATA LPRO(47)/'TSS SIM DATA PROFILE BAL1$'/
      DATA LPRO(48)/'TSS SIM DATA PROFILE BAL2$'/
      DATA LPRO(49)/'TSS SIM DATA PROFILE BAL5$'/
      DATA LPRO(50)/'TSS SIM DATA PROFILE BAL6$'/
      DATA LPRO(51)/'TSS SIM DATA PROFILE BAL7$'/
      DATA LPRO(52)/' SIM DATA PROFILE HL546AG$'/
      DATA LPRO(53)/' SIM DATA PROFILE HL246AF$'/
      DATA LPRO(54)/' SIM DATA PROFILE HL446AE$'/
      DATA LPRO(55)/' SIM DATA PROFILE HL146AE$'/
      DATA LPRO(56)/' SIM DATA PROFILE HL346AF$'/
      DATA LPRO(57)/' SIM DATA PROFILE HJ146AE$'/
      DATA FPRO(1)/'HL146AB.XXX'/
      DATA FPRO(2)/'HL246AB.XXX'/
      DATA FPRO(3)/'HJ146AB.XXX'/
      DATA FPRO(4)/'HEL30AB.XXX'/
      DATA FPRO(5)/'H30SKAB.XXX'/
      DATA FPRO(6)/'H30SKAC.XXX'/
      DATA FPRO(7)/'HEL30AC.XXX'/
      DATA FPRO(8)/'HEL30AD.XXX'/
      DATA FPRO(9)/'HL246AC.XXX'/
      DATA FPRO(10)/'HL346AB.XXX'/
      DATA FPRO(11)/'HL446AB.XXX'/
      DATA FPRO(12)/'HL546AB.XXX'/
      DATA FPRO(13)/'HL546AC.XXX'/
      DATA FPRO(14)/'HL246AD.XXX'/
      DATA FPRO(15)/'HL446AC.XXX'/
      DATA FPRO(16)/'HL146AC.XXX'/
      DATA FPRO(17)/'HL346AD.XXX'/
      DATA FPRO(18)/'HJ146AC.XXX'/
      DATA FPRO(19)/'HEL30AE.XXX'/
      DATA FPRO(20)/'HEL30AF.XXX'/
      DATA FPRO(21)/'H30SKAD.XXX'/
      DATA FPRO(22)/'H30SKAE.XXX'/
      DATA FPRO(23)/'H30SKAF.XXX'/
      DATA FPRO(24)/'HEL30AG.XXX'/
      DATA FPRO(25)/'HEL30AH.XXX'/
      DATA FPRO(26)/'H30SKAG.XXX'/
      DATA FPRO(27)/'H30SKAH.XXX'/
      DATA FPRO(28)/'H30SKAI.XXX'/
      DATA FPRO(29)/'HEL30AI.XXX'/
      DATA FPRO(30)/'HEL30AJ.XXX'/
      DATA FPRO(31)/'HL546AE.XXX'/
      DATA FPRO(32)/'HL246AE.XXX'/
      DATA FPRO(33)/'HL446AD.XXX'/
      DATA FPRO(34)/'HL146AD.XXX'/
      DATA FPRO(35)/'HL346AE.XXX'/
      DATA FPRO(36)/'HJ146AD.XXX'/
      DATA FPRO(37)/'HL546AF.XXX'/
      DATA FPRO(38)/'GEM1.XXX'/
      DATA FPRO(39)/'GEM2.XXX'/
      DATA FPRO(40)/'GEM3.XXX'/
      DATA FPRO(41)/'SAT1.XXX'/
      DATA FPRO(42)/'SAT2.XXX'/
      DATA FPRO(43)/'SAT3.XXX'/
      DATA FPRO(44)/'SAT4.XXX'/
      DATA FPRO(45)/'SAT6.XXX'/
      DATA FPRO(46)/'SAT8.XXX'/
      DATA FPRO(47)/'BAL1.XXX'/
      DATA FPRO(48)/'BAL2.XXX'/
      DATA FPRO(49)/'BAL5.XXX'/
      DATA FPRO(50)/'BAL6.XXX'/
      DATA FPRO(51)/'BAL7.XXX'/
      DATA FPRO(52)/'HL546AG.XXX'/
      DATA FPRO(53)/'HL246AF.XXX'/
      DATA FPRO(54)/'HL446AE.XXX'/
      DATA FPRO(55)/'HL146AE.XXX'/
      DATA FPRO(56)/'HL346AF.XXX'/
      DATA FPRO(57)/'HJ146AE.XXX'/
C
C ***********************************************************************
C
C     SIMULATION FILE MODIFICATION
C
      A23=24.5
      TS=0.051
      WRITE (6,*) '  INPUT RCS IN SQUARE METERS  '
      READ (5,*)RCSM
******
LINES DELETED FROM BASELINE PROGRAM
      COMMON /TARGET/ITARG,SRCS
     COMMON /ACTDAT/R,ARDOT,SPANG,SRANG,SPRTE,SRRTE,AL,BT,SALF,SBTA,
    1ER(3),EV(3),ERTO(3),AZRATE,ELRATE,AZRTE,ELRTE
```

```
       COMMON /TERM/ITERM
       COMMON /OUTPUT/MSWF,MTF,MSF,SSRNG,SSRDOT,SSPANG,SSRANG,SSPRTE,
     2               SSRRTE,SSRSS,MADVF,MRDVF,MARDVF,MRRDVF
     3 ,SSALP,SSBET
       COMMON /SYSDAT/TS,DUM2(14)
C       TEST DATA FROM WS32TDATA1
        CHARACTER*9 FPRO(18)
        CHARACTER*32 IXT,IYT(22),LPRO(18)
        DATA IXT/'TIME  SECONDS$'/
        DATA IYT(1)/'RANGE  FEET$'/
        DATA IYT(2)/'RANGE RATE  FT/SEC$'/
        DATA IYT(3)/'ROLL ANGLE  DEG$'/
        DATA IYT(4)/'PITCH ANGLE  DEG$'/
        DATA IYT(5)/'ROLL RATE  DEG/SEC$'/
        DATA IYT(6)/'PITCH RATE  DEG/SEC$'/
        DATA IYT(7)/'ALPHA  DEG$'/
        DATA IYT(8)/'BETA  DEG$'/
        DATA IYT(9)/'AZ RATE  DEG/SEC$'/
        DATA IYT(10)/'EL RATE  DEG/SEC$'/
        DATA IYT(11)/' X (NORTH)  FEET$'/
        DATA IYT(12)/' Y (EAST)  FEET$'/
        DATA IYT(13)/'-Z (ALTITUDE)  FEET$'/
        DATA IYT(14)/' ELEVATION ANGLE DEG$'/
        DATA IYT(15)/'DELTA RANGE  FEET$'/
        DATA IYT(16)/'DELTA RANGE RATE  FT/SEC$'/
        DATA IYT(17)/'DELTA ROLL ANGLE  DEG$'/
        DATA IYT(18)/'DELTA PITCH ANGLE  DEG$'/
        DATA IYT(19)/'DELTA ROLL RATE  DEG/SEC$'/
        DATA IYT(20)/'DELTA PITCH RATE  DEG/SEC$'/
        DATA IYT(21)/'DELTA ALPHA  DEG$'/
        DATA IYT(22)/'DELTA BETA  DEG$'/
        DATA LPRO(1)/' SIMULATION PROFILE HJ146$'/
        DATA LPRO(2)/' SIMULATION PROFILE HL146$'/
        DATA LPRO(3)/' SIMULATION PROFILE HL246$'/
        DATA LPRO(4)/' SIMULATION PROFILE HL346$'/
        DATA LPRO(5)/' SIMULATION PROFILE HL446$'/
        DATA LPRO(6)/' SIMULATION PROFILE HL546$'/
        DATA LPRO(7)/' SIMULATION PROFILE BJ146$'/
        DATA LPRO(8)/' SIMULATION PROFILE BL146$'/
        DATA LPRO(9)/' SIMULATION PROFILE BL246$'/
        DATA LPRO(10)/' SIMULATION PROFILE BL346$'/
        DATA LPRO(11)/' SIMULATION PROFILE BL446$'/
        DATA LPRO(12)/' SIMULATION PROFILE BL546$'/
        DATA LPRO(13)/' SIMULATION PROFILE C6P48$'/
        DATA LPRO(14)/' SIMULATION PROFILE C6M48$'/
        DATA LPRO(15)/' SIMULATION PROFILE C6P30$'/
        DATA LPRO(16)/' SIMULATION PROFILE C6M30$'/
        DATA LPRO(17)/' SIMULATION PROFILE CLP16$'/
        DATA LPRO(18)/' SIMULATION PROFILE CLM16$'/
        DIMENSION RID(120)
        DATA FPRO(1)/'HJ146.JSC'/
        DATA FPRO(2)/'HL146.BIN'/
        DATA FPRO(3)/'HL246.BIN'/
        DATA FPRO(4)/'HL346.BIN'/
        DATA FPRO(5)/'HL446.BIN'/
        DATA FPRO(6)/'HL546.BIN'/
        DATA FPRO(7)/'BJ146.BIN'/
        DATA FPRO(8)/'BL146.BIN'/
        DATA FPRO(9)/'BL246.BIN'/
        DATA FPRO(10)/'BL346.BIN'/
        DATA FPRO(11)/'BL446.BIN'/
        DATA FPRO(12)/'BL546.BIN'/
        DATA FPRO(13)/'C6P48.BIN'/
        DATA FPRO(14)/'C6M48.BIN'/
        DATA FPRO(15)/'C6P30.BIN'/
        DATA FPRO(16)/'C6M30.BIN'/
        DATA FPRO(17)/'CLP16.BIN'/
        DATA FPRO(18)/'CLM16.BIN'/
        CHARACTER*9 UNIT7
        BYTE IC(120)
        COMMON /TMR/X,Y,Z,VX,VY,VZ,
     1                DLP(3),DEL(3),DUE(3),
     2                DSU(3),THAZL1,THEL1,THAZU1
       COMMON /INPUT/RO(3),VO(3),EWB(3)
        DIMENSION TP(2001),D(2001,22)
C
        WRITE (6,*)'1 : TEK'
        WRITE (6,*)'2 : VT125'
        WRITE (6,*)'3 : VT240'
        WRITE (6,*)'4 : PC'
        READ (5,*)ITERM
        WRITE(6,*)'PROFILE NUMBER   PROFILE'
        DO L=1,18
        WRITE(6,200)L,LPRO(L)
200     FORMAT(7X,I2,9X,A32)
        ENDDO
        WRITE(6,*)'INPUT PROFILE NUMBER'
        READ(5,*)ITAPE
C       WRITE(6,*) 'ENTER NAME OF BINARY INPUT FILE'
C       READ(5,1001)UNIT7
C1001    FORMAT(A24)
        UNIT7=FPRO(ITAPE)
        OPEN(UNIT=4,FORM='UNFORMATTED',STATUS='OLD',
     2       FILE=UNIT7)
C
C       READ(4)IC
C       WRITE(6,150)(IC(I),I=1,30)
150     FORMAT(60A2)
        IFTRK=0
        WRITE(6,*)' INPUT 1 IF YOU WANT TO FILTER USING TRACK FLAG'
        READ(5,*)IFTRK
        WRITE(6,*)'INPUT RSC IN SQUARE METERS'
        READ (5,*)RCSM
************
************
LINES ADDED TO DELIVERABLE PROGRAM
        ITARG=0
C
        WRITE (6,*)'1 : TEK'
        WRITE (6,*)'2 : VT125'
        WRITE (6,*)'3 : VT240'
        WRITE (6,*)'4 : PC'
        READ (5,*)ITERM
C
        WRITE (6,*)'ENTER :  1 IF YOU ARE PROCESSING TMR DATA'
        WRITE (6,*)'         2 IF YOU ARE PROCESSING CINE DATA'
        WRITE (6,*)'         3 IF YOU ARE PROCESSING BEST DATA'.
        READ (5,*)IREF
C
        WRITE(6,*)'ENTER TIME INTERVAL  ( 0,0 FOR THE WHOLE INTERVAL )'
        READ(5,*)STIME,STTIME
        IF (STTIME.EQ.0)STTIME=999
C
        WRITE (6,*)'DO YOU WANT TO FILTER THE DATA ? (Y/N)'
        READ (5,2322)ANS
 2322   FORMAT(A)
        WRITE(6,*)'PROFILE NUMBER   PROFILE'
        DO L=1,19
```

```
  196              WRITE(6,200)L,LPRO(L)
  197    200       FORMAT(7X,I2,9X,A32)
  198              ENDDO
  199              WRITE (6,*)'ENTER C TO CONTINUE. Q TO QUIT :'
  200              READ (5,101) REPLY
  201    101       FORMAT (A)
  202              IF (REPLY.EQ.'C') THEN
  203                DO L=20,38
  204                  WRITE(6,200)L,LPRO(L)
  205                ENDDO
  206                WRITE (6,*)'ENTER C TO CONTINUE. Q TO QUIT :'
  207                READ (5,101) REPLY
  208                IF (REPLY.EQ.'C') THEN
  209                  DO L=39,57
  210                    WRITE(6,200)L,LPRO(L)
  211                  ENDDO
  212                ENDIF
  213              ENDIF
  214              WRITE(6,*)'INPUT PROFILE NUMBER'
  215              READ(5,*)ITAPE
  216              UNIT7=FPRO(ITAPE)
  217              CALL FIXIT(ITILT,LPRO(ITAPE))
  218              IF (ITAPE.LT.38.AND.ITAPE.GT.51)GO TO 39
  219              IF (ITAPE.GE.38.AND.ITAPE.LE.51)GO TO 49
  220    C
  221     39       IF (IREF.EQ.1) THEN
  222                UNIT7(9:11)='JST'
  223              ELSE IF (IREF.EQ.2) THEN
  224                UNIT7(9:11)='JSC'
  225              ELSE
  226                UNIT7(9:11)='BST'
  227              ENDIF
  228              GO TO 59
  229     49       IF (IREF.EQ.1) THEN
  230                UNIT7(6:8)='JST'
  231              ELSE IF (IREF.EQ.2) THEN
  232                UNIT7(6:8)='JSC'
  233              ELSE
  234                UNIT7(6:8)='BST'
  235              ENDIF
  236     59       OPEN(UNIT=4,FORM='UNFORMATTED',STATUS='OLD',
  237          2        FILE=UNIT7)
  238    C
  239              TOUT=0.
******
LINES DELETED FROM BASELINE PROGRAM
  109              WRITE(6,*)'SRCS=',SRCS
  110              TOUT=0.
************
************
LINES ADDED TO DELIVERABLE PROGRAM
  254              DSU(3)=-5.46
  255    C         WRITE(6,*)' INPUT 1 FOR SCREEN OUTPUT'
  256    C         READ(5,*)TOUT
  257              J=0
  258    C READ START TIME
  259              READ(4)TBIAS,GMTIME,XMO,XDAY,XYR
  260              ILOOP=1
  261    1         CONTINUE
  262              READ(4,END=99)T,SSRNG,SSRDOT,SSRANG,SSPANG,SSRRTE,SSPRTE
  263          1  .X,Y,Z,VX,VY,VZ,AX,AY,AZ,IS1,IS2,RSS,RFPWR,AERR,BERR,ALFX,
  264          1   BETY,SCRR,SCPR
  265              IF (T.LT.STIME) GOTO 1
  266              IJJ=2**13
```

```
 267          ITF=IAND(IS2,IJJ)
 268          IF (ITF.NE.IJJ.AND.ANS.EQ.'Y') GO TO 1
 269          CALL RPAB(SSRANG,SSPANG,SSALP,SSBET)
 270          CALL TMR2KU
 271          DO I=1,3
 272            RNEW(I)=RO(I)
 273            VNEW(I)=VO(I)
 274          END DO
 275          IF(ILOOP.NE.1) GO TO 7
 276     6    CALL EXEC
 277          IF(MPRF.EQ.1) THEN
 278              TS=.051
 279          ELSE
 280              TS=.119
 281          END IF
 282          IF(ILOOP.EQ.1)THEN
 283             T1=T
 284             ILOOP=0
 285             GO TO 196
 286          END IF
 287     7    CONTINUE
 288          T1=T1+TS
 289          IF(T1.GT.T)THEN
 290             T1=T1-TS
 291             GO TO 196
 292          END IF
 293          DO I=1,3
 294             RO(I)=(RNEW(I)-ROLD(I))*(T1-T2)/(T-T2)+ROLD(I)
 295             VO(I)=(VNEW(I)-VOLD(I))*(T1-T2)/(T-T2)+VOLD(I)
 296          END DO
 297          GO TO 6
 298     196  CONTINUE
 299          T2=T
 300          DO I=1,3
 301             ROLD(I)=RNEW(I)
 302             VOLD(I)=VNEW(I)
 303          END DO
 304          HRRTE=HRRTE*180./(3.14159*1000.)
 305          HPRTE=HPRTE*180./(3.14159*1000.)
 306          J=J+1
 307          IF(J.EQ.2001)GO TO 99
 308          IF(T.GE.STTIME)GO TO 99
 309          TP(J)=T
******
LINES DELETED FROM BASELINE PROGRAM
 125          DSU(3)=-5.46
 126          WRITE(6,*)' INPUT 1 FOR SCREEN OUTPUT'
 127          READ(5,*)TOUT
 128          J=0
 129          READ(4,END=99)T,X,Y,Z,VX,VY,VZ
 130          READ(4,END=99)T1,X,Y,Z,VX,VY,VZ
 131          TS=T1-T
 132          WRITE(6,*)' TS= ',TS
 133    1     CONTINUE
 134          READ(4,END=99)T,X,Y,Z,VX,VY,VZ
 135    C DATA IN METERS
 136          CALL TMR2KU
 137          IF(TOUT.EQ.1)THEN
 138          WRITE(6,100)T,SSRNG,SSSRRDOT,SSPANG,SRANG,SSPRTE,SRRTE,SALF,SBTA,
 139         1 AZRATE,ELRATE,AZRTE,ELRTE
 140    100   FORMAT(' ',2F9.1,9F9.3)
 141          ENDIF
 142          CALL EXEC
 143          IF(IFTRK.EQ.1.AND.MTF.EQ.0)GO TO 1
```

```
  144          J=J+1
  145          IF(J.EQ.2001)GO TO 99
  146          TP(J)=T
************
************
LINES ADDED TO DELIVERABLE PROGRAM
  312          D(J,3)=SSRANG
  313          D(J,4)=SSPANG
  314          D(J,5)=SSRRTE
******
LINES DELETED FROM BASELINE PROGRAM
  149          D(J,4)=SSPANG
  150          D(J,3)=SSRANG
  151          D(J,5)=SSRRTE
************
************
LINES ADDED TO DELIVERABLE PROGRAM
  318          D(J,9)=HRNG
  319          D(J,10)=HRDOT
  320          D(J,11)=RO(1)
  321          D(J,12)=RO(2)
  322          D(J,13)=-RO(3)
  323          D(J,14)=ATAND(-RO(3)/SQRT(RO(1)*RO(1)+RO(2)*RO(2)))
  324          D(J,15)=SSRNG-R
******
LINES DELETED FROM BASELINE PROGRAM
  155          D(J,9)=AZRTE
  156          D(J,10)=ELRTE
  157          D(J,11)=X
  158          D(J,12)=Y
  159          D(J,13)=-Z
  160          D(J,14)=ATAND(-Z/(X*X+Y*Y))
  161          D(J,15)=SSRNG-R
************
************
LINES ADDED TO DELIVERABLE PROGRAM
  328          D(J,19)=SSRRTE-SRRTE
  329          D(J,20)=SSPRTE-SPRTE
******
LINES DELETED FROM BASELINE PROGRAM
  165          D(J,19)=SSRRTE-SRTE
  166          D(J,20)=SSPRTE-SPRTE
************
************
LINES ADDED TO DELIVERABLE PROGRAM
  332          D(J,23)=SAZRTE
  333          D(J,24)=SELRTE
  334          D(J,25)=RSS
  335          D(J,26)=RFPWR
  336          D(J,27)=AERR
  337          D(J,28)=BERR
  338          D(J,29)=ALFX
  339          D(J,30)=BETY
  340          D(J,31)=SCRR
  341          D(J,32)=SCPR
  342          IF (HRSS.LE.0) THEN
  343            D(J,33)=0
  344          ELSE
  345            D(J,33)=(32*HRSS)-181.+(40*ALOG10(HRNG))
  346          ENDIF
  347          D(J,34)=RACCEL
  348          D(J,35)=HRNG-R
  349          D(J,36)=HRDOT-ARDOT
  350          D(J,37)=HRANG-SRANG
```

```
      D(J,38)=HPANG-SPANG
      D(J,39)=HRRTE-SRRTE
      D(J,40)=HPRTE-SPRTE
      D(J,41)=HALP-SALF
      D(J,42)=HBET-SBTA
      D(J,43)=HRSS/32.
      IF(J.GT.2000)THEN
      WRITE(6,*)' MORE THAN 2000 POINTS'
      STOP
      ENDIF
      GO TO 1
 99   CONTINUE
      J=J-1
      IXD=0
 94   CONTINUE
      CALL SORT(TP,D,J,ITILT,IXD,IYD,GMTIME,IREF)
      GO TO 94
      END
C ***************************************************************************
      SUBROUTINE SORT(T,D,J,ITILT,IXD,IYD,GMTIME,IREF)
      DIMENSION D(2001,43),X(2001),Y(2001),T(2001)
      CHARACTER*40 IXT,IYT(43),PRONAME
      CHARACTER*4 REFF
      DIMENSION ITILT(10),IXL(10),IYL(10)
      DATA IXT/'TIME  SECONDS$'/
      DATA IYT(1)/'KU MDM RANGE  FEET$'/
      DATA IYT(2)/'KU MDM RANGE RATE  FT/SEC$'/
      DATA IYT(3)/'KU MDM ROLL ANGLE  DEG$'/
      DATA IYT(4)/'KU MDM PITCH ANGLE  DEG$'/
      DATA IYT(5)/'KU MDM ROLL RATE  DEG/SEC$'/
      DATA IYT(6)/'KU MDM PITCH RATE  DEG/SEC$'/
      DATA IYT(7)/'KU MDM ALPHA  DEG$'/
      DATA IYT(8)/'KU MDM BETA  DEG$'/
      DATA IYT(9)/'SIM RANGE FEET$'/
      DATA IYT(10)/'SIM RANGE RATE FT/SEC$'/
      DATA IYT(11)/'WSMR X (NORTH)  FEET$'/
      DATA IYT(12)/'WSMR Y (EAST)  FEET$'/
      DATA IYT(13)/'WSMR  -Z (ALTITUDE)  FEET$'/
      DATA IYT(14)/'WSMR ELEVATION ANGLE DEG$'/
      DATA IYT(15)/'DELTA RANGE  FEET ( KU - WSMR )$'/
      DATA IYT(16)/'DELTA RANGE RATE  FT/SEC ( KU - WSMR )$'/
      DATA IYT(17)/'DELTA ROLL ANGLE  DEG ( KU -  WSMR )$'/
      DATA IYT(18)/'DELTA PITCH ANGLE  DEG ( KU -  WSMR )$'/
      DATA IYT(19)/'DELTA ROLL RATE  DEG/SEC ( KU -  WSMR)$'/
      DATA IYT(20)/'DELTA PITCH RATE  DEG/SEC ( KU -  WSMR )$'/
      DATA IYT(21)/'DELTA ALPHA  DEG ( KU - WSMR )$'/
      DATA IYT(22)/'DELTA BETA  DEG ( KU -  WSMR )$'/
      DATA IYT(23)/'WSMR AZ RATE  DEG/SEC$'/
      DATA IYT(24)/'WSMR EL RATE  DEG/SEC$'/
      DATA IYT(25)/'KU SCANNER RSS ( VOLTS )$'/
      DATA IYT(26)/'KU SCANNER RF POWER ( VOLTS )$'/
      DATA IYT(27)/'KU SCANNER ALPHA ERROR ( VOLTS )$'/
      DATA IYT(28)/'KU SCANNER BETA ERROR ( VOLTS )$'/
      DATA IYT(29)/'KU SCANNER ALPHA X ( VOLTS )$'/
      DATA IYT(30)/'KU SCANNER BETA Y ( VOLTS )$'/
      DATA IYT(31)/'KU SCANNER ROLL RATE ( VOLTS )$'/
      DATA IYT(32)/'KU SCANNER PITCH RATE ( VOLTS )$'/
      DATA IYT(33)/'SIM RADAR CROSS SECTION ( DBSM )$'/
      DATA IYT(34)/'WSMR RANGE ACCELERATION  FT/SEC/SEC$'/
      DATA IYT(35)/'DELTA RANGE FEET (SIM-WSMR)$'/
      DATA IYT(36)/'DELTA RANGE RATE FT/SEC (SIM-WSMR)$'/
      DATA IYT(37)/'DELTA ROLL ANGLE DEG (SIM-WSMR)$'/
      DATA IYT(38)/'DELTA PITCH ANGLE DEG (SIM-WSMR)$'/
      DATA IYT(39)/'DELTA ROLL RATE DEG/SEC (SIM-WSMR)$'/
      DATA IYT(40)/'DELTA PITCH RATE DEG/SEC (SIM-WSMR)$'/
      DATA IYT(41)/'DELTA ALPHA DEG (SIM-WSMR)$'/
      DATA IYT(42)/'DELTA BETA DEG (SIM-WSMR)$'/
      DATA IYT(43)/'SIM RADAR SIGNAL STRENGTH$'/
      IFLAG=1
      IF (IREF.EQ.1)THEN
        REFF=' TMR'
      ELSE IF (IREF.EQ.2) THEN
        REFF='CINE'
      ELSE
        REFF='BEST'
      ENDIF
      DO I=1,43
        L=INDEX(IYT(I),'WSMR')
        IF (L .GT. 0) THEN
          IYT(I)(L:L+3) = REFF
        ENDIF
      ENDDO
 1    CONTINUE
      DO I=1,43
      WRITE(6,68)I,IYT(I)
 68   FORMAT(1X,I4,10X,A40)
      ENDDO
      WRITE(6,*)'INPUT IXD,IYD  IXD=0 FOR TIME'
      IF (IFLAG.EQ.0)THEN
        IFLAG=1
        IXD=0
        IYD=1
        GO TO 731
      ENDIF
      READ(5,*)IXD,IYD
 731  IF(IXD.EQ.0)THEN
      DO I=1,J
******
LINES DELETED FROM BASELINE PROGRAM
      GO TO 1
 99   CONTINUE
      IXD=0
 94   CONTINUE
      WRITE(6,*)'RCS IN METERS=',RCSM
      WRITE(6,*)'PARA         AXES TITLE'
      DO I=1,22
     WRITE(6,68)I,IYT(I)
 68   FORMAT(1X,I4,10X,A32)
      ENDDO
      WRITE(6,*)'INPUT IXD,IYD  IXD=0 FOR TIME'
      READ(5,*)IXD,IYD
      CALL SORT(TP,D,J,ITAPE,IXD,IYD)
      GO TO 94
      END
      SUBROUTINE SORT(T,D,J,ITAPE,IXD,IYD)
      DIMENSION D(2001,22),X(2001),Y(2001),T(2001)
      CHARACTER*32 IXT,IYT(22),LPRO(18)
      DIMENSION ITILT(8),IXL(8),IYL(8)
      DATA IXT/'TIME  SECONDS$'/
      DATA IYT(1)/'RANGE  FEET$'/
      DATA IYT(2)/'RANGE RATE  FT/SEC$'/
      DATA IYT(3)/'ROLL ANGLE  DEG$'/
      DATA IYT(4)/'PITCH ANGLE  DEG$'/
      DATA IYT(5)/'ROLL RATE  DEG/SEC$'/
      DATA IYT(6)/'PITCH RATE  DEG/SEC$'/
      DATA IYT(7)/'ALPHA  DEG$'/
      DATA IYT(8)/'BETA  DEG$'/
      DATA IYT(9)/'AZ RATE  DEG/SEC$'/
```

```
  198            DATA IYT(10)/'EL RATE  DEG/SEC$'/
  199            DATA IYT(11)/' X (NORTH)  FEET$'/
  200            DATA IYT(12)/' Y (EAST)  FEET$'/
  201            DATA IYT(13)/'-Z (ALTITUDE)  FEET$'/
  202            DATA IYT(14)/' ELEVATION ANGLE DEG$'/
  203            DATA IYT(15)/'DELTA RANGE  FEET$'/
  204            DATA IYT(16)/'DELTA RANGE RATE  FT/SEC$'/
  205            DATA IYT(17)/'DELTA ROLL ANGLE  DEG$'/
  206            DATA IYT(18)/'DELTA PITCH ANGLE  DEG$'/
  207            DATA IYT(19)/'DELTA ROLL RATE  DEG/SEC$'/
  208            DATA IYT(20)/'DELTA PITCH RATE  DEG/SEC$'/
  209            DATA IYT(21)/'DELTA ALPHA  DEG$'/
  210            DATA IYT(22)/'DELTA BETA  DEG$'/
  211            DATA LPRO(1)/' SIMULATION PROFILE HJ146$'/
  212            DATA LPRO(2)/' SIMULATION PROFILE HL146$'/
  213            DATA LPRO(3)/' SIMULATION PROFILE HL246$'/
  214            DATA LPRO(4)/' SIMULATION PROFILE HL346$'/
  215            DATA LPRO(5)/' SIMULATION PROFILE HL446$'/
  216            DATA LPRO(6)/' SIMULATION PROFILE HL546$'/
  217            DATA LPRO(7)/' SIMULATION PROFILE BJ146$'/
  218            DATA LPRO(8)/' SIMULATION PROFILE BL146$'/
  219            DATA LPRO(9)/' SIMULATION PROFILE BL246$'/
  220            DATA LPRO(10)/' SIMULATION PROFILE BL346$'/
  221            DATA LPRO(11)/' SIMULATION PROFILE BL446$'/
  222            DATA LPRO(12)/' SIMULATION PROFILE BL546$'/
  223            DATA LPRO(13)/' SIMULATION PROFILE C6P48$'/
  224            DATA LPRO(14)/' SIMULATION PROFILE C6M48$'/
  225            DATA LPRO(15)/' SIMULATION PROFILE C6P30$'/
  226            DATA LPRO(16)/' SIMULATION PROFILE C6M30$'/
  227            DATA LPRO(17)/' SIMULATION PROFILE CLP16$'/
  228            DATA LPRO(18)/' SIMULATION PROFILE CLM16$'/
  229            JPRO=ITAPE
  230            CALL FIXIT(ITILT,LPRO(JPRO))
  231            IF(IXD.EQ.0)THEN
  232            DO I=1,J
*************
************
LINES ADDED TO DELIVERABLE PROGRAM
  461            CALL PLOTIT(ITILT,IXL,IYL,X,Y,J,GMTIME,IYD,IXD)
  462            GO TO 1
  463       2    CONTINUE
  464            RETURN
  465            END
  466     C ****************************************************************************
  467            SUBROUTINE FIXIT(IOUT,IN)
  468            DIMENSION IOUT(10)
  469            CHARACTER*4 ITEMP(10)
  470            CHARACTER*40 IN
  471            ITEMP(1)=(IN(1:4))
******
LINES DELETED FROM BASELINE PROGRAM
  246            CALL PLOTIT(ITILT,IXL,IYL,X,Y,J)
  247            RETURN
  248            END
  249            SUBROUTINE FIXIT(IOUT,IN)
  250            DIMENSION IOUT(8)
  251            CHARACTER*4 ITEMP(8)
  252            CHARACTER*32 IN
  253            ITEMP(1)=(IN(1:4))
************
************
LINES ADDED TO DELIVERABLE PROGRAM
  479            ITEMP(9)=(IN(33:36))
  480            ITEMP(10)=(IN(37:40))
  481             ENCODE(40,999,IOUT)(ITEMP(I),I=1,10)
  482    999      FORMAT(10A4)
  483             RETURN
  484             END
  485    C ************************************************************************
  486             SUBROUTINE PLOTIT(ITILT,IXL,IYL,X,Y,J,GMTIME,IYD,IXD)
  487             COMMON /TERM/ITERM,XMO,XDAY,XYR,TBIAS,XJMO,XJDAY,XJYR
  488              COMMON/TMR/A,B,C,D,E,F,G(3),AH(3),AI(3),AJ(3),THAZL1,THEL1,THAZU1
  489             DOUBLE PRECISION SIG,AVG
  490             BYTE CR(2)
  491             DIMENSION ITILT(8),IXL(8),IYL(8)
  492             DIMENSION X(1),Y(1),TINL(30)
  493             WRITE(6,*)'  1 FOR MEAN AND STANDARD DEVIATION OF Y'
  494             READ(5,*)ISTA
  495             NSC=0
  496             XMAX=X(1)
******
LINES DELETED FROM BASELINE PROGRAM
  261             ENCODE(32,999,IOUT)(ITEMP(I),I=1,8)
  262    999      FORMAT(8A4)
  263             RETURN
  264             END
  265             SUBROUTINE PLOTIT(ITILT,IXL,IYL,X,Y,J)
  266             COMMON /TERM/ITERM
  267             DIMENSION ITILT(8),IXL(8),IYL(8)
  268             DIMENSION X(1),Y(1)
  269             BYTE CR(2)
  270             COMMON/TMR/A,B,C,D,E,F,G(3),AH(3),AI(3),AJ(3),THAZL1,THEL1,THAZU1
  271             CR(1)=27
  272             CR(2)=12
  273             XMAX=X(1)
************
************
LINES ADDED TO DELIVERABLE PROGRAM
  500             GMHOUR1=GMTIME/60./60.
  501             GMHOUR=INT(GMHOUR1)
  502             GMMIN1=(GMHOUR1-GMHOUR)*60.
  503             GMMIN=INT(GMMIN1)
  504             GMSEC=INT((GMMIN1-GMMIN)*60.)
  505             DO I=1,J
******
LINES DELETED FROM BASELINE PROGRAM
  277             DO I=1,J
************
************
LINES ADDED TO DELIVERABLE PROGRAM
  513             IF(YMAX.EQ.YMIN)YMAX=0.1
  514    2        CONTINUE
  515             YMAX1=YMAX
  516             YMIN1=YMIN
  517             IF (ITERM.EQ.1) CALL TEKALL(4114,480,0,1,0)
******
LINES DELETED FROM BASELINE PROGRAM
  285             IF (ITERM.EQ.1) CALL TEKALL(4114,480,0,1,0)
************
************
LINES ADDED TO DELIVERABLE PROGRAM
  520             IF (IYD.EQ.1)CALL RINTL(X,Y,J,TINL,NTINL)
  521             CALL BGNPL(-1)
  522             CALL FLATBD
  523             CALL PAGE(14.,20.)
  524             CALL AREA2D (9.0,14.0)
  525             CALL HEIGHT(.45)
  526    C        CALL TITLE(ITILT,100,IXL,100,IYL,100,9.0,13.5)
```

```
527          CALL MESSAG(ITILT,100,-0.6,16.5)
528          CALL RESET ('HEIGHT')
529          CALL HEIGHT (.3)
530          I100=100
531   C      0.6 WAS SUBTRACTED TO CENTER AND 1 INCHE WERE ADDED IN HEIGHT
532          CALL MESSAG('TEST DATE$',I100,0.7,15.5)
533          IF (XMO.GE.10)THEN
534            CALL REALNO(XMO,0,3.0,15.5)
535          ELSE
536            CALL REALNO(XMO,0,3.3,15.5)
537          ENDIF
538          CALL REALNO(XDAY,0,3.9,15.5)
539          IF (XDAY.GE.10) THEN
540            CALL REALNO(XYR,0,4.8,15.5)
541          ELSE
542            CALL REALNO(XYR,0,4.5,15.5)
543          ENDIF
544          CALL MESSAG(' REVISION 12$',I100,6.0,15.5)
545   C      POSITION CHANGED FROM 13.7 TO 14.2
546   C      X-POSITION MOVED FORWARD BY 1.2
547          CALL MESSAG('T0=          GMT=$',I100,1.2,14.2)
548          CALL REALNO(GMTIME,0,1.8,14.2)
549          CALL REALNO(GMHOUR,0,5.1,14.2)
550          CALL REALNO(GMMIN,0,6.0,14.2)
551          CALL REALNO(GMSEC,0,6.9,14.2)
552          IF(ISTA.EQ.1)THEN
553          AVG=0
554          SIG=0
555          DO I=1,J
556          AVG=AVG+Y(I)
557          SIG=SIG+Y(I)**2
558          END DO
559          AVG=AVG/J
560          SIG=SQRT( SIG/J -AVG*AVG)
561          CALL MESSAG('MEAN= $',I100,-0.9,-2.0)
562          CALL REALNO(AVG,3,'ABUT','ABUT')
563          CALL MESSAG('  STANDARD DEVIATION= $',I100,3.3,-2.0)
564          CALL REALNO(SIG,3,'ABUT','ABUT')
565          ENDIF
566          CALL XNAME(IXL,100)
567          CALL YNAME(IYL,100)
568          CALL INTAXS
569          CALL YAXANG(0.)
570          IF(NSC.EQ.0)THEN
571          CALL GRAF(XMIN,'SCALE',XMAX,YMIN,'SCALE',YMAX)
572          ENDIF
573          IF(NSC.EQ.1)THEN
574          CALL GRAF(XMIN,'SCALE',XMAX,YMIN,'SCALE',YMAX)
575          ENDIF
576           IF (NTINL.NE.0.AND.IXD.EQ.0)THEN
577            DO K=1,NTINL
578                   IVEC=1302
579              CALL RLVEC (TINL(K),YMIN1,TINL(K),YMAX1,IVEC)
580            ENDDO
581           ENDIF
582          CALL CURVE(X,Y,J,0)
583          CALL GRID(1,1)
******
LINES DELETED FROM BASELINE PROGRAM
 288         CALL BGNPL(-1)
 289         CALL FLATBD
 290         CALL PAGE(14.,18.)
 291         CALL HEIGHT(.3)
 292         CALL TITLE(ITILT,100,IXL,100,IYL,100,9.0,13.5)
```

```
  293           I100=100
  294    C      CALL MESSAG('LOWER AZIMUTH=$',I100,1.7,13.)
  295    C      CALL REALNO(THAZL1,2,'ABUT','ABUT')
  296    C      CALL MESSAG('UPPER AZIMUTH=$',I100,1.7,12.5)
  297    C      CALL REALNO(THAZU1,2,'ABUT','ABUT')
  298    C      CALL MESSAG('ELEVATION=$',I100,1.7,12.)
  299    C      CALL REALNO(THEL1,2,'ABUT','ABUT')
  300    C      CALL BLNK1(1.5,7.5,11.9,13.5,4)
  301    C      CALL HEADIN(ITILT,-100,-8,4)
  302    C      CALL HEADIN('LOWER AZIMUTH=$',100,4,4)
  303    C      CALL REALNO(THAZL1,2,'ABUT','ABUT')
  304    C      CALL HEADIN('UPPER AZIMUTH=$',100,4,4)
  305    C      CALL REALNO(THAZU1,2,'ABUT','ABUT')
  306    C      CALL HEADIN('ELEVATION=$',100,4,4)
  307    C      CALL REALNO(THEL1,2,'ABUT','ABUT')
  308           CALL YAXANG(0.)
  309           CALL GRAF(XMIN,'SCALE',XMAX,YMIN,'SCALE',YMAX)
  310           CALL CURVE(X,Y,J,0)
  311    C      KK=J/30
  312    C      K=0
  313    C      DO I=1,KK
  314    C      K=30+K
  315    C      CALL RLINT(K,X(K),Y(K))
  316    C      ENDDO
  317
  318           CALL GRID(1,1)
*************
*************
LINES ADDED TO DELIVERABLE PROGRAM
  586           CALL DONEPL
  587           CR(1)=27
  588           CR(2)=12
  589           WRITE(6,888)CR
  590    888    FORMAT('+',2A1)
  591           WRITE(6,*)'  INPUT 1 TO CHANGE SCALE OF Y AXIS'
  592           READ(5,*)NSC
  593           IF(NSC.EQ.1)THEN
  594           WRITE(6,*)'YMAX=',YMAX,' YMIN=',YMIN
  595           WRITE(6,*)' NEW YMAX'
  596           READ(5,*)YMAX
  597           WRITE(6,*)'NEW YMIN'
  598           READ(5,*)YMIN
  599           GO TO 2
  600           ENDIF
******
LINES DELETED FROM BASELINE PROGRAM
  321    888    FORMAT('+',2A1)
  322           CALL DONEPL
  323    C MICKEY MOUSE FIX
  324           IMM=1
  325           IF(IMM.EQ.0)THEN
  326           REWIND (5)
  327           READ(5,192)IC
  328    192    FORMAT(A1)
  329          WRITE(6,888)CR
  330           ENDIF
*************
*************
LINES ADDED TO DELIVERABLE PROGRAM
  603    C **********************************************************************
  604           SUBROUTINE RPAB(ROLLQ,PITCHQ,ALPHA,BETA)
  605          DEGRAD=57.29576
  606          PSI=67./DEGRAD
  607          PIT=PITCHQ/DEGRAD
```

```
 608          ROL=ROLLQ/DEGRAD
 609          XB=SIN(PIT)
 610          YB=-(SIN(ROL))*SQRT(1.0-XB*XB)
 611          Z=SQRT(1.0-XB*XB-YB*YB)
 612          IF(ROLLQ.LE.90.0.AND.ROLLQ.GE.-90.0)Z=-Z
 613          XR=XB*COS(PSI)+YB*SIN(PSI)
 614          YR=YB*COS(PSI)-XB*SIN(PSI)
 615          YRZR=SQRT(YR*YR+Z*Z)
 616          ALF=ASIN(YR/YRZR)
 617          BTA=ASIN(-XR/SQRT(XR*XR+YR*YR+Z*Z))
 618          ALPHA=ALF*DEGRAD
 619          BETA=BTA*DEGRAD
 620          IF(Z.GE.0.0.AND.YR.LE.0.0)ALPHA=-(180.0+ALPHA)
 621          IF(Z.GE.0.0.AND.YR.GT.0.0)ALPHA=(180.0-ALPHA)
 622           RETURN
 623          END
 624    C ***************************************************************************
 625           SUBROUTINE RINTL(T,R,N,TI,J)
 626           DIMENSION RI(5),R(1),DS(5),TI(30),T(1)
 627           DATA RI/2550.,5750.,11510.,23030.,43510./
 628           RMAX=R(1)
 629           RMIN=R(1)
 630           DO 1 I=1,N
 631           RMAX=AMAX1(RMAX,R(I))
 632           RMIN=AMIN1(RMIN,R(I))
 633    1      CONTINUE
 634           MRMAX=1
 635           MRMIN=1
 636           DO 2 I=1,5
 637           IF(RMAX.GT.RI(I))MRMAX=I
 638           IF(RMIN.GT.RI(I))MRMIN=I
 639    2      CONTINUE
 640           J=0
 641           IF(MRMAX.EQ.MRMIN)RETURN
 642           J=0
 643           DO 3 L=1,5
 644           DS(L)=R(1)-RI(L)
 645    3      CONTINUE
 646           DO 4 I=1,N
 647           DO 5 L=1,5
 648           IF( (R(I)-RI(L)) * DS(L) .LT. 0 )THEN
 649           J=J+1
 650           TI(J)=T(I)
 651           DS(L)=R(I)-RI(L)
 652           ENDIF
 653    5      CONTINUE
 654    4      CONTINUE
 655           RETURN
 656           END
 657    C ***************************************************************************
 658    C **  MODED JWG 2/8/85
 659    C **
 660    C **  INPUT VIA COMMON VIA X,Y,Z,VX,VY,VZ,AX,AY,AZ
 661    C **  OUTPUT VIA COMMON /ACTDAT/
******
LINES DELETED FROM BASELINE PROGRAM
 333           SUBROUTINE TMR2KU
 334    C **  MODED JWG 2/8/85
 335    C **
 336
 337    C **  INPUT VIA COMMON VIA X,Y,Z,VX,VY,VZ
 338    C **  OUTPUT VIA COMMON /ACTDAT/
************
************
```

```
LINES ADDED TO DELIVERABLE PROGRAM
  741             SUBROUTINE TMR2KU
  742             COMMON /TMR/X,Y,Z,VX,VY,VZ,
  743          1               DLP(3),DEL(3),DUE(3),
  744          2               DSU(3),THAZL1,THEL1,THAZU1,A23
  745           COMMON /INPUT/RO(3),VO(3),EWB(3)
  746           COMMON /ACTDAT/R,ARDOT,SPANG,SRANG,SPRTE,SRRTE,AL,BT,SALF,SBTA,
  747        1ER(3),EV(3),ERTO(3),AZRATE,ELRATE,AZRTE,ELRTE
  748        2,AX,AY,AZ,AAX,AAY,AAZ,RACCEL
  749    C      DIMENSION DLP(3),DEL(3),DUE(3),DSU(3)
******
LINES DELETED FROM BASELINE PROGRAM
  418             COMMON /TMR/X,Y,Z,VX,VY,VZ,
  419          1               DLP(3),DEL(3),DUE(3),
  420          2               DSU(3),THAZL1,THEL1,THAZU1
  421           COMMON /INPUT/RO(3),VO(3),EWB(3)
  422           COMMON /ACTDAT/R,ARDOT,SPANG,SRANG,SPRTE,SRRTE,AL,BT,SALF,SBTA,
  423        1ER(3),EV(3),ERTO(3),AZRATE,ELRATE,AZRTE,ELRTE
  424    C      DIMENSION DLP(3),DEL(3),DUE(3),DSU(3)
************
************
LINES ADDED TO DELIVERABLE PROGRAM
  754           DIMENSION APT(3),ALAZ(3),AELV(3),AST(3)
  755           DATA DEGRAD/57.275/,PI/3.14159/
******
LINES DELETED FROM BASELINE PROGRAM
  429           DATA DEGRAD/57.275/,PI/3.14159/
************
************
LINES ADDED TO DELIVERABLE PROGRAM
  780           VPT(3)=VZ
  781           APT(1)=AX
  782           APT(2)=AY
  783           APT(3)=AZ
  784    C
******
LINES DELETED FROM BASELINE PROGRAM
  454           VPT(3)=VZ
  455    C
************
************
LINES ADDED TO DELIVERABLE PROGRAM
  838    C CONVERT TO VELOCITIES REFERENCED TO GIMBALS
  839           CALL MULT31(AZL,VPT,VLAZ)
  840           CALL MULT31(ELV,VLAZ,VELV)
  841           CALL MULT31(AZU,VELV,VST)
  842    C CONVERT TO ACCELATIONS REFERENCED TO GIMBALS
  843           CALL MULT31(AZL,APT,ALAZ)
  844           CALL MULT31(ELV,ALAZ,AELV)
  845           CALL MULT31(AZU,AELV,AST)
  846    C THESE ARE VELOCITIES IN GIMBAL REFERENCE.
******
LINES DELETED FROM BASELINE PROGRAM
  509    C CONVERT TO VELOCITIES REFERENCED TO GIMBALS
  510           CALL MULT31(AZL,VPT,VLAZ)
  511           CALL MULT31(ELV,VLAZ,VELV)
  512           CALL MULT31(AZU,VELV,VST)
  513    C THESE ARE VELOCITIES IN GIMBAL REFERENCE.
************
************
LINES ADDED TO DELIVERABLE PROGRAM
  854             C23=COSD(A23)
  855             S23=SIND(A23)
  856             X1=RO(2)*C23-RO(3)*S23
```

```
******
LINES DELETED FROM BASELINE PROGRAM
  521           C23=COSD(23.)
  522           S23=SIND(23.)
  523           X1=RO(2)*C23-RO(3)*S23
************
************
LINES ADDED TO DELIVERABLE PROGRAM
  868           AAX=AST(2)*C23-AST(3)*S23
  869           AAY=-AST(2)*S23-AST(3)*C23
  870           AAZ=-AST(1)
  871          CALL ACT
******
LINES DELETED FROM BASELINE PROGRAM
  535          CALL ACT
************
************
LINES ADDED TO DELIVERABLE PROGRAM
  878           RETURN
  879          END
  880    C ******************************************************************************
  881          SUBROUTINE AZGEN(AZ,ANGAZ)
******
LINES DELETED FROM BASELINE PROGRAM
  542    C THE EXAMPLE CASE RESULTS ARE:
  543    C     WRITE(6,*)R,ARDOT
  544    C     WRITE(6,*)SRANG,SPANG
  545    C     WRITE(6,*)SRRTE,SPRTE
  546    C     WRITE(6,*)SALF,SBTA
  547    C     WRITE(6,*)AZRTE,ELRTE
  548           RETURN
  549          END
  550          SUBROUTINE AZGEN(AZ,ANGAZ)
************
************
LINES ADDED TO DELIVERABLE PROGRAM
  895    C ******************************************************************************
  896          SUBROUTINE ELGEN(EL,ANGEL)
******
LINES DELETED FROM BASELINE PROGRAM
  564          SUBROUTINE ELGEN(EL,ANGEL)
************
************
LINES ADDED TO DELIVERABLE PROGRAM
  908    C ******************************************************************************
  909    C   SUBROUTINE ACT
******
LINES DELETED FROM BASELINE PROGRAM
  576    C   SUBROUTINE ACT
************
************
LINES ADDED TO DELIVERABLE PROGRAM
  919    C ******************************************************************************
  920          SUBROUTINE ACT
******
LINES DELETED FROM BASELINE PROGRAM
  586          SUBROUTINE ACT
************
************
LINES ADDED TO DELIVERABLE PROGRAM
  923        3,AX,AY,AZ,AAX,AAY,AAZ,RACCEL
  924          COMMON /INPUT/ ERT(3),EVT(3),EWB(3),DUM(18)                  00015210
******
LINES DELETED FROM BASELINE PROGRAM
```

```
  589          COMMON /INPUT/ ERT(3),EVT(3),EWB(3),DUM(18)                     00015210
************
************
LINES ADDED TO DELIVERABLE PROGRAM
  973   C    COMPUTE RANGE ACCELERATION TO TARGET.
  974          VSQ=EV(1)**2+EV(2)**2+EV(2)**2
  975          RACCEL=(VSQ+ER(1)*AAX+ER(2)*AAY+ER(3)*AAZ-ARDOT**2)/R
  976   C    COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH RATE(AZRATE).          00015960
******
LINES DELETED FROM BASELINE PROGRAM
  638   C    COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH RATE(AZRATE).          00015960
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 1059          DR(1)=0.0
 1060          DR(2)=11.130
 1061          DR(3)=-5.79
 1062   C  RANGE BIAS ERROR IS COMPUTED IN SUBROUTINE RTRACK AS
******
LINES DELETED FROM BASELINE PROGRAM
  721          DR(1)=45.738                                                    00029850
  722          DR(2)=11.130                                                       00029860
  723          DR(3)=-5.79                                                    00029870
  724   C  RANGE BIAS ERROR IS COMPUTED IN SUBROUTINE RTRACK AS
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 1072          PSBIAS=PII*0.0
 1073   C                                                                      00029962
 1074   C  ROLL ANGLE ERROR.                                                   00029964
 1075          RLBIAS=PII*0.0
 1076   C  PITCH ANGLE ERROR.                                                  00029968
 1077          PTBIAS=PII*0.0
 1078   C
******
LINES DELETED FROM BASELINE PROGRAM
  734          PSBIAS=PII*0.1                                                  00029960
  735   C                                                                      00029962
  736   C  ROLL ANGLE ERROR.                                                   00029964
  737          RLBIAS=PII*0.25                                                 00029966
  738   C  PITCH ANGLE ERROR.                                                  00029968
  739          PTBIAS=PII*0.25                                                 00029970
  740   C
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 1081           NBIAS=0
 1082           IF(NBIAS.NE.0)GO TO 700
******
LINES DELETED FROM BASELINE PROGRAM
  743           NBIAS=1
  744           IF(NBIAS.NE.0)GO TO 700
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 1230          REAL INTT,K4,K5,K6                                                   0002533
 1231          INTEGER AT1A(10,2),AT1E(10,2),AT2A(10,2),AT2E(10,2)
 1232          COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)                        00025350
******
LINES DELETED FROM BASELINE PROGRAM
  892          REAL INTT,IAZDSC,IELDSC                                         00025335
  893          COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)                        00025350
************
************
```

```
LINES ADDED TO DELIVERABLE PROGRAM
 1242          DIMENSION TX1(3,3),TX2(3,3),TX3(3,3),TBL(3,3)                  00025450
 1243         DIMENSION TDC(3)
 1244     C ********************************************************
 1245     C
 1246     C  ATRACK MODIFIED JAN 28 1986 BY M. MEYER
 1247     C  MODIFICATIONS TO SUBROUTINE ATRACK WERE IMPLEMENTED
 1248     C  TO UPDATE THE LOOP CONSTANTS AND MORE ACCURATELY
 1249     C  SIMULATE THE ACTUAL SIGNAL PROCESSING PERFORMED
 1250     C  BY THE RADAR
 1251     C
 1252     C ********************************************************
 1253     C
 1254     C------- NEW LOOP CONSTANTS JAN 28 1986--------------------
 1255     C
 1256          DATA AT1A/9*5,1,6*13,5,3*1/
 1257          DATA AT1E/9*6,1,6*16,6,2*1,2/
 1258          DATA AT2A/9*407,149,6*662,407,3*149/
 1259          DATA AT2E/9*532,195,6*866,532,3*195/
 1260          DATA K6/3.60E-5/,K4/.0048876/,K5/.236/,DTOR/.0174533/
 1261     C
 1262         DATA TDC/0.05122118,0.1195161,0.2561557/
******
LINES DELETED FROM BASELINE PROGRAM
  903          DIMENSION AT1(10,2),AT2(10,2),TX1(3,3),TX2(3,3),TX3(3,3),TBL(3,3)00025450
  904         DIMENSION TDC(3)
  905          DATA AT1/9*1.5529E-3,2.0106E-4,6*3.9750E-3,1.5529E-3,            00025460
  906       2           3*2.0106E-4/,AT2/9*6.5907E-3,2.3725E-3,                 00025470
  907       3           6*1.0546E-2,6.5907E-3,3*2.3725E-3/                      00025480
  908         DATA TDC/0.05122118,0.1195161,0.2561557/
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 1296     C
 1297     C ------ NEW CODE AS OF JAN 28 1986-------------
 1298     C
 1299     C  UPDATE ESTIMATED TARGET INERTIAL AZIMUTH RATE.                      00025790
 1300           IAZRATE=KSAT(IAZRATE+AT1A(MRNG,IMODE)*IAZDSC)                  00025800
 1301     C  UPDATE ESTIMATED TARGET INERTIAL ELEVATION RATE.                    00025810
 1302           IELRATE=KSAT(IELRATE+AT1E(MRNG,IMODE)*IELDSC)                  00025820
 1303     C
 1304           AZRATE=K6*DTOR*FLOAT(IAZRATE)
 1305           ELRATE=K6*DTOR*FLOAT(IELRATE)
 1306     C
 1307           IALRATE=KSAT(IAZRATE+AT2A(MRNG,IMODE)*IAZDSC)
 1308           IBTRATE=KSAT(IELRATE+AT2E(MRNG,IMODE)*IELDSC)
 1309     C
 1310           IF(IALRATE.GT.0) THEN
 1311               ALRATE=K4*K5*DTOR*FLOAT(IALRATE/32)
 1312           ELSE
 1313               ALRATE=K4*K5*DTOR*FLOAT((IALRATE-31)/32)
 1314           END IF
 1315     C
 1316           IF(IBTRATE.GT.0) THEN
 1317               BTRATE=K4*K5*DTOR*FLOAT(IBTRATE/32)
 1318           ELSE
 1319               BTRATE=K4*K5*DTOR*FLOAT((IBTRATE-31)/32)
 1320           END IF
 1321     C
 1322     C ************************************************              00025840
******
LINES DELETED FROM BASELINE PROGRAM
  942           ADSC=0.0431*IAZDSC                                            00025730
  943           EDSC=0.0431*IELDSC                                            00025740
 944   C  UPDATE ESTIMATED TARGET INERTIAL AZIMUTH RATE.                             00025790
 945           AZRATE=AZRATE+TSAM*AT1(MRNG,IMODE)*ADSC                               00025800
 946   C  UPDATE ESTIMATED TARGET INERTIAL ELEVATION RATE.                           00025810
 947           ELRATE=ELRATE+TSAM*AT1(MRNG,IMODE)*EDSC                               00025820
 948   C                                                                             00025830
 949   C  **********************************************                            00025840
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 1332          ALRATE=(ALRATE+WGZ*SB)/CB-WGX
 1333          GO TO 4                                                               00025950
******
LINES DELETED FROM BASELINE PROGRAM
  959          ALRATE=(AZRATE+AT2(MRNG,IMODE)*ADSC+WGZ*SB)/CB-WGX                    00025940
  960          GO TO 4                                                               00025950
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 1337          BTRATE=BTRATE-WGY
 1338  C
 1339  C --- END OF JAN 28 1986 MODIFICATIONS--------------------
 1340  C                                                                             00026000
******
LINES DELETED FROM BASELINE PROGRAM
  964          BTRATE=(ELRATE+AT2(MRNG,IMODE)*EDSC)-WGY                              00025990
  965  C                                                                             00026000
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 1385  C       WRITE(6,902) AZDISC,ELDISC,IAZDSC,IELDSC                              00026470
 1386     900  FORMAT(' ALR,BTR,AZR,ELR,SRR,SPR=',6F14.9)                            00026480
 1387     901  FORMAT(' TBL 2X2 =',4F10.4)                                           00026490
 1388     902  FORMAT(' AZD,ELD,AD,ED =',2F10.4,2I9)                                    00026500
 1389          RETURN                                                                00026510
 1390          END                                                                   00026520
 1391  C
 1392  C  ****************************************************
 1393  C  * INTEGER FUNCTION KSAT JAN 28 1986                 *
 1394  C  ****************************************************
 1395  C
 1396  C  THIS FUNCTION CHECKS ATRACK LOOP FOR SATURATION
 1397  C
 1398          INTEGER FUNCTION KSAT(K)
 1399  C
 1400          IF(K.GE.0) THEN
 1401               KSAT=JMIN0(K,2**15)
 1402          ELSE
 1403               KSAT=JMAX0(K,-2**15)
 1404          END IF
 1405          RETURN
 1406          END
 1407  C                                                                             00024530
******
LINES DELETED FROM BASELINE PROGRAM
 1010  C       WRITE(6,902) AZDISC,ELDISC,ADSC,EDSC                                  00026470
 1011     900  FORMAT(' ALR,BTR,AZR,ELR,SRR,SPR=',6F10.2)                            00026480
 1012     901  FORMAT(' TBL 2X2 =',4F10.4)                                           00026490
 1013     902  FORMAT(' AZD,ELD,AD,ED =',4F10.4)                                     00026500
 1014          RETURN                                                                00026510
 1015          END                                                                   00026520
 1016  C                                                                             00024530
************
************
LINES ADDED TO DELIVERABLE PROGRAM
```

```
 1689          DATA RI/120.,640.,1520.,2560.,5760.,11520.,23040.,43520.,         00028620
 1690        2          49920.,1.8228E+6/                                         00028630
 1691          DATA FW/7.7215,3.3090,0.2969/,NRI/10/                              00028640
 1692    C
 1693    C
 1694    C ***********************************************************
 1695    C      IMPLEMENTATION OF HYSTERISIS FOR THE SAMPLING RATE
 1696    C      CHANGE AND FOR THE PRF CHANGE ALONG WITH CHANGES IN
 1697    C      RI(RANGE INTERVAL) WAS COMPLETED FEB 6,1986 BY M. MEYER
 1698    C ***********************************************************
 1699    C                                                                        00028650
******
LINES DELETED FROM BASELINE PROGRAM
 1298          DATA RI/120.,240.,780.,2560.,5772.,11544.,23089.,43747.,          00028620
 1299        2          57722.,1.8228E+6/                                         00028630
 1300          DATA FW/7.7215,3.3090,0.2969/,NR1/10/                              00028640
 1301    C                                                                        00028650
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 1719    C***** MODIFIED FEB 6 1986 BY M. MEYER*****************
 1720    74     IF(MSAM.EQ.1)THEN
 1721              IF(XRNG.GT.3200.)THEN
 1722                 MSAM=2
 1723              ELSE
 1724                 MSAM=1
 1725    C***** MODIFIED FEB 17,1986 BY M. MEYER ***************
 1726    C***** GUARANTEES THE CORRECT LOOP BANDWIDTHS**********
 1727    C***** FOR THE HYSTERISIS LOOP*************************
 1728    C
 1729                 IF(XRNG.GT.2560) MRNG=4
 1730    C
 1731    C*****************************************************
 1732              END IF
 1733           ELSE
 1734              IF(XRNG.GT.2560.)THEN
 1735                 MSAM=2
 1736              ELSE
 1737                 MSAM=1
 1738              END IF
 1739           END IF
 1740    C                                                                        00028880
******
LINES DELETED FROM BASELINE PROGRAM
 1321    74     IF(MRNG.GT.4) GO TO 76                                            00028840
 1322           MSAM=1                                                            00028850
 1323           GO TO 80                                                          00028860
 1324    76     MSAM=2                                                            00028870
 1325    C                                                                        00028880
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 1754    C ****** MODIFIED FEB 6 1986 BY M. MEYER ************
 1755    84     IF(MPRF.EQ.1)THEN
 1756              IF(XRNG.GT.49920.)THEN
 1757                 MPRF=2
 1758              ELSE
 1759                 MPRF=1
 1760              END IF
 1761           ELSE
 1762              IF(XRNG.GT.43520.)THEN
 1763                MPRF=2
 1764    C***** MODIFIED FEB 17, 1986 BY M. MEYER*************
 1765    C***** GUARANTEES THE CORRECT CONSTANTS *************
 1766   C***** FOR THE LOW PRF*******************************
 1767   C
 1768                MRNG=10
 1769   C
 1770   C*************************************************
 1771             ELSE
 1772                MPRF=1
 1773             END IF
 1774         END IF
 1775   90    CONTINUE                                                      00029060
******
LINES DELETED FROM BASELINE PROGRAM
 1339   84    IF(MRNG.GT.9) GO TO 86                                        00029020
 1340         MPRF=1                                                        00029030
 1341         GO TO 90                                                      00029040
 1342   86    MPRF=2                                                        00029050
 1343   90    CONTINUE                                                      00029060
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 1895         DIMENSION QNV(2)
 1896   C
 1897   C ----------PS AND QNV CONSTANT CHANGES FEB 17,1986 BY M. MEYER---------
 1898   C
 1899         DATA NFREQ/1,5/,BN/9772.4,616.6/
 1900         DATA PS/9*4.,2.,5*4.,2.,4.,8.,8.,16./
 1901       2       ,PDIA,PDIR,PDIV/1.4142,3.1623,2.0,4.4721,2.8284,6.3246/,    00022940
 1902       3         PT/42658.,3125.,195.3/
 1903         DATA QNV/.00067,.011/
 1904         DATA TDC/0.05122118,0.1195161,0.2561557/
******
LINES DELETED FROM BASELINE PROGRAM
 1463         DATA NFREQ/1,5/,BN/9772.4,616.6/,PS/9*1.,2.,5*1.,2.,4.,8.,8.,16./00022930
 1464       2       ,PDIA,PDIR,PDIV/1.4142,3.1623,2.0,4.4721,2.8284,6.3246/,    00022940
 1465       3         PT/42658.,3125.,195.3/,QNV/.04166666/                     00022950
 1466         DATA TDC/0.05122118,0.1195161,0.2561557/
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 1942   C            WRITE(6,221)YY,SIGBAR
 1943     221      FORMAT('YY,SIGBAR =',2F14.5)
 1944         SNRDTD=10.*ALOG10(SNRDT)                                      00023240
******
LINES DELETED FROM BASELINE PROGRAM
 1504   C       WRITE(6,221)YY,SIGBAR
 1505   C  221      FORMAT('YY,SIGBAR =',F14.5)
 1506         SNRDTD=10.*ALOG10(SNRDT)                                      00023240
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 1954         XX=XX/(XX+QNV(MSAM))
 1955         S1=S1*XX                                                      00023300
******
LINES DELETED FROM BASELINE PROGRAM
 1516         XX=XX/(XX+QNV)                                                00023296
 1517         S1=S1*XX                                                      00023300
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 2603         COMMON /ICNTL/IDUM2(14),MRNG,MSAM,IDUM6(11)
 2604         COMMON /OUTPUT/IDUM7(3),DUM3(6),SRSS,IDUM4(4)                 00029330
 2605         COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD                       00029340
 2606         DIMENSION PS(10,2),QNV(2),A1(2)
 2607         DATA PS/9*4.,2.,5*4.,2.,4.,8.,8.,16./
 2608           DATA QNV/.00067,.011/,A1/.0321,.51/
 2609     C *****************************************************
 2610     C SUBROUITNE RSS HAS BEEN UPDATED TO CORRESPOND TO THE
 2611     C DERIVATION OF AGCERR PRESENTED IN THE FINAL REPORT ON
 2612     C KUBAND COMPUTER SIMULATION. M. MEYER FEB 17, 1986
 2613     C *****************************************************
 2614     C                                                                   00029370
******
LINES DELETED FROM BASELINE PROGRAM
 2165           COMMON /ICNTL/IDUM2(14),MRNG,IDUM6(12)                      00029320
 2166           COMMON /OUTPUT/IDUM7(3),DUM3(6),SRSS,IDUM4(4)               00029330
 2167           COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD                     00029340
 2168           DIMENSION PS(10,2)                                          00029350
 2169           DATA PS/9*1.,2.,5*1.,2.,4.,8.,8.,16./,QNV/0.04166666/       00029360
 2170     C                                                                 00029370
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 2619     C  STEP 1-1: COMPUTE AGC ERROR AND CHECK LIMITS.
 2620     C ————UPDATED FEB 17, 1986——————————————————————————
 2621           AGCERR=A1(MSAM)*4.*PS(MRNG,IMODE)/(AGCO*(SNRDT+1.0)+QNV(MSAM))
 2622           IF(AGCERR.GT.10.) AGCERR=10.0                               00029440
******
LINES DELETED FROM BASELINE PROGRAM
 2175     C  STEP 1-1: COMPUTE AGC ERROR AND CHECK LIMITS.                  00029420
 2176           AGCERR=4.*PS(MRNG,IMODE)/(AGCO*(SNRDT+1.0)+QNV)             00029430
 2177           IF(AGCERR.GT.10.) AGCERR=10.0                               00029440
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 2627     C ————UPDATED FEB 17, 1986——————————————————————————
 2628           IF(AGCO.GT.0.25) AGCO=0.25
 2629           AGCODB=10.*ALOG10(AGCO)                                     00029500
******
LINES DELETED FROM BASELINE PROGRAM
 2182           IF(AGCO.GT.1.0) AGCO=1.0                                    00029490
 2183           AGCODB=10.*ALOG10(AGCO)                                     00029500
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 2635           SRSS=1./AGCO
 2636     C ————UPDATED FEB 17, 1986——————————————————————————
 2637           SRSS=10.*ALOG10(SRSS)-6.0
 2638           RETURN                                                      00029580
******
LINES DELETED FROM BASELINE PROGRAM
 2189           SRSS=1./AGCO                                                00029560
 2190           SRSS=10.*ALOG10(SRSS)                                       00029570
 2191           RETURN                                                      00029580
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 2714           DIMENSION PS(10,2)
 2715     C
 2716     C ——— PS VALUES WERE UPDATED FEB 17,1986 BY M. MEYER—————
 2717     C
 2718           DATA PS/9*4.0,2.,5*4.,2.,4.,8.,8.,16./
 2719           SNF=1.                                                      00035690
 2720           X=AGCO*(SNRDT/(4.*PS(MRNG,IMODE))+1.0)                      00035700
 2721     C *****************************************************
 2722     C X=12.25/X WAS REPLACED BY X=6.25/X TO MORE ACCURATELY
 2723     C REFLECT A/D SATURATION BY M. MEYER FEB 17, 1986
 2724     C *****************************************************
 2725           X=6.25/X
 2726           IF(X.GT.1) RETURN                                              00035720
******
LINES DELETED FROM BASELINE PROGRAM
 2267           DIMENSION PS(10,2)                                             00035670
 2268           DATA PS/9*10.0,2.,5*1.,2.,4.,8.,8.,16./                        00035680
 2269           SNF=1.                                                         00035690
 2270           X=AGCO*(SNRDT/(4.*PS(MRNG,IMODE))+1.0)                         00035700
 2271           X=12.25/X                                                      00035710
 2272           IF(X.GT.1) RETURN                                              00035720
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 3099           COMMON /SUDIPH/ X,Y,Z,PAZ,PEL
 3100           COMPLEX CSUM,CDIFAZ,CDIFEL,CEARLY,CLATE,CDF1,CDF5,CDF2,CDF4,   00020230
******
LINES DELETED FROM BASELINE PROGRAM
 2645           COMPLEX CSUM,CDIFAZ,CDIFEL,CEARLY,CLATE,CDF1,CDF5,CDF2,CDF4,   00020230
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 3108           COMPLEX DAZ,DEL
 3109           DATA ILOOP/1/
 3110     C
 3111     C    *************************************************************
 3112     C
 3113     C    MODIFIED JAN 10 1986 BY M. MEYER
 3114     C    MODIFICATIONS TO SUBROUTINE SIGNAL INCLUDE
 3115     C    CALCULATION OF THE AZIMUTH AND ELEVATION ANGLES
 3116     C    USE OF MEASURED ANTENNA PATTERNS INSTEAD
 3117     C    OF FUNCIONS SPAT AND DPAT AND A
 3118     C    FACTOR IN THE DIFFERENCE CHANNELS SIGNAL
 3119     C    WHICH ACCOUNTS FOR THE FINITE WIDTH PHASE
 3120     C    TRANSITION IN THE REAL PHASE PATTERNS.
 3121     C
 3122     C    *************************************************************
 3123     C
 3124     C    *************************************************************
 3125     C    * STEP 0: READ IN ANTENNA PATTERNTERNS AND SET PHASE BALANCE  *
 3126     C    *************************************************************
 3127     C
 3128           IF (ILOOP.NE.1) GO TO 11
 3129               CALL READPAT
 3130               PBAL=0.
 3131               ILOOP=0
 3132     11    CONTINUE
 3133     C                                                                    00020320
******
LINES DELETED FROM BASELINE PROGRAM
 2653     C                                                                    00020320
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 3176     C    STEP 2-1: COMPUTE AZIMUTH AND ELEVATION ANGLE.                  00020770
 3177           AZ=ATAN2D(RAU(2,K),ABS(RAU(3,K)))
 3178           EL=-ATAN2D(RAU(1,K),ABS(RAU(3,K)))
 3179     C    STEP 2-2: COMPUTE ANTENNA SUM, DIFFERENCE AND PHASE FACTORS     00020780
 3180           CALL INTERP(AZ,EL)                                             00020790
 3181     C                                                                    00020800
******
LINES DELETED FROM BASELINE PROGRAM
 2696     C    STEP 2-1: COMPUTE SUM PATTERN ANGLE.                            00020750
 2697           PSI=ACOS(ABS(RAU(3,K)))                                        00020760
 2698     C                                                                    00020770
 2699     C    STEP 2-2: COMPUTE ANTENNA SUM PATTERN MULTIPLICATION FACTOR.    00020780
 2700          X=SPAT(PSI)                                                  00020790
 2701    C                                                                  00020800
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 3197    C                                                                  00021040
******
LINES DELETED FROM BASELINE PROGRAM
 2717    C  STEP 3-1: COMPUTE AZ AND EL DIFFERENCE PATTERN ANGLES.          00020960
 2718          DELAZ=-ASIN(RAU(2,K))                                        00020970
 2719          DELEL=ASIN(RAU(1,K))                                         00020980
 2720    C                                                                  00020990
 2721    C  STEP 3-2: COMPUTE AZ AND EL DIFFERENCE PATTERN MULTIPLICATION   00021000
 2722    C          FACTORS.                                                00021010
 2723          Y=DPAT(DELAZ)                                                00021020
 2724          Z=DPAT(DELEL)                                                00021030
 2725    C                                                                  00021040
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 3200    C             AND PHASE DIFFERENCE AND BALANCE WEIGHTINGS
 3201          DAZ=XX*Y*CMPLX(COSD(PAZ+PBAL),SIND(PAZ+PBAL))                00021070
 3202          DEL=XX*Z*CMPLX(COSD(PEL+PBAL),SIND(PEL+PBAL))                00021080
 3203    C                                                                  00021090
******
LINES DELETED FROM BASELINE PROGRAM
 2728          DAZ=XX*Y                                                     00021070
 2729          DEL=XX*Z                                                     00021080
 2730    C                                                                  00021090
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 3221       21 RGE=1.0E-4                                                   00021240
 3222          RGL=1.0E-4                                                   00021250
 3223          GO TO 25                                                     00021260
******
LINES DELETED FROM BASELINE PROGRAM
 2748       21 RGE=0.0                                                      00021240
 2749          RGL=0.0                                                      00021250
 2750          GO TO 25                                                     00021260
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 3257    C
 3258    C  STEP 5-3: COMPUTE DOPPLER FILTER WEIGHTING FOR EACH OF FIVE DOPPLER  00021610
******
LINES DELETED FROM BASELINE PROGRAM
 2784    C                                                                  00021600
 2785    C  STEP 5-3: COMPUTE DOPPLER FILTER WEIGHTING FOR EACH OF FIVE DOPPLER  00021610
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 3357    C
 3358    C   NOTE: DEBUGGING PRINT STATEMENTS                               00022610
******
LINES DELETED FROM BASELINE PROGRAM
 2884    C                                                                  00022600
 2885    C   NOTE: DEBUGGING PRINT STATEMENTS                               00022610
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 3362    C       WRITE(6,901) DFWTS(1,K),DFWTS(2,K),DFWTS(3,1),DFWTS(4,1),   00022650
 3363    C     2 DFWTS(5,1)                                                  00022660
 3364      902 FORMAT(' NT,S,DAZ,DEL,RGE,RGL,RGWGT,F3 =',I5,6F10.2,I5)       00022670
```

```
******
LINES DELETED FROM BASELINE PROGRAM
 2889   C      WRITE(6,901) DFWTS(1,K),DFWTS(2,K),DFWTS(3,1),DFWTS(4,1),          00022650
 2890   C    2 DFWTS(5,1)                                                          00022660
 2891     902  FORMAT(' NT,S,DAZ,DEL,RGE,RGL,RGWGT,F3 =',I5,6F10.2,I5)             00022670
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 4035   C ****RI DATA STATEMENT UPDATED FEB 6,1986 BY M. MEYER ************
 4036          DATA RI/120.,640.,1520.,2560.,5760.,11520.,23040.,43520.,          00015350
 4037        2          49920.,1.8228E+6/,NRI/10/,PI/3.141592653/                 00015360
 4038   C                                                                         00015370
******
LINES DELETED FROM BASELINE PROGRAM
 3562          DATA RI/120.,240.,780.,2552.,5772.,11544.,23089.,43747.,           00015350
 3563        2          57722.,1.8228E+6/,NRI/10/,PI/3.141592653/                 00015360
 3564   C                                                                         00015370
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 4101   C  STEP 1-3: COMPUTAE INITIAL INNER AND OUTER GIMBAL RATES.                00016020
 4102   C    COMPUTE INITIAL OUTER GIMBAL RATE(ALRATE).                           00016030
******
LINES DELETED FROM BASELINE PROGRAM
 3627   C  STEP 1-3: COMPUTE INITIAL INNER AND OUTER GIMBAL RATES.                00016020
 3628   C    COMPUTE INITIAL OUTER GIMBAL RATE(ALRATE).                           00016030
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 4413   C
 4414   C
 4415   C  ***********************************************************
 4416   C  SUBROUTINE VELPRO WAS MODIFIED FEB 6 1986 BY M. MEYER
 4417   C  MODIFICATIONS CONSISTED OF CHECKING THE VARIABLE MPRF
 4418   C  FOR A VALUE OF ONE (IMPLIES 7 KC MODE) AND IF TRUE
 4419   C  ASSUMING THE VELOCITY ESTIMATE GIVEN BY THE VELOCITY
 4420   C  DISCRIMINANT IS UNAMBIGUOUS.
 4421   C  ***********************************************************
 4422   C                                                                         00027290
******
LINES DELETED FROM BASELINE PROGRAM
 3939   C                                                                         00027290
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 4442   C ***********************************
 4443   C CHANGED JAN 30 1986 BY H. MAGNUSSON
 4444   C ***********************************
 4445           IF(IV1.GT.128)IV1=128
 4446           IFRAC=IPROM(IV1)                                                  00027490
******
LINES DELETED FROM BASELINE PROGRAM
 3959           IFRAC=IPROM(IV1)                                                  00027490
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 4453   C  ****************************
 4454   C  CHANGED FEB 6 1986 BY M. MEYER
 4455   C  ****************************
 4456   C
 4457             IF(MPRF.EQ.1) THEN
 4458                IF(INTEG.GE.0.AND.INTEG.LE.21)THEN
 4459                  IRVEL=0.
 4460                ELSE
 4461                 IRVEL=4096.
 4462               END IF
 4463               GO TO 8
 4464            END IF
 4465   C ****************************************                                 00027570
******
LINES DELETED FROM BASELINE PROGRAM
 3966   C                                                                          00027560
 3967   C ****************************************                                 00027570
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 4500       8   CONTINUE
 4501   C                                                                          00027920
******
LINES DELETED FROM BASELINE PROGRAM
 4002   C                                                                          00027920
************
************
LINES ADDED TO DELIVERABLE PROGRAM
 4970            subroutine readPAT
 4971
 4972   c
 4973   c
 4974   c                Read in the sum, phase, and difference patterns
 4975   c
 4976   c
 4977
 4978            real a1linear( 41,41 ), e1linear( 41,41 )
 4979
 4980            real sa1linear( 41,41 ), se1linear( 41,41 )
 4981
 4982            real pa1linear( 41,41 ), pe1linear( 41,41 )
 4983
 4984            common / linear / a1linear, e1linear
 4985
 4986            common / linear1 / sa1linear, se1linear
 4987
 4988            common / linear2 / pa1linear, pe1linear
 4989
 4990
 4991            open( unit=3, file='[KUBAND.HOWARD.MARK]az1d.dat',
 4992         1          access='sequential', form='unformatted',
 4993         1          status='old', readonly )
 4994            read( 3 ) ( ( a1linear( i,j ), j = 1,41 ), i = 1,41 )
 4995            close( 3 )
 4996
 4997            open( unit=3, file='[KUBAND.HOWARD.MARK]e11d.dat',
 4998         1          access='sequential', form='unformatted',
 4999         1          status='old', readonly )
```

```
      read( 3 ) ( ( e1linear( i,j ), j = 1,41 ), i = 1,41 )
      close( 3 )

      open( unit=3, file='[KUBAND.HOWARD.MARK]az1s.dat',
     1      access='sequential', form='unformatted',
     1      status='old', readonly )
      read( 3 ) ( ( sa1linear( i,j ), j = 1,41 ), i = 1,41 )
      close( 3 )

      open( unit=3, file='[KUBAND.HOWARD.MARK]el1s.dat',
     1     access='sequential', form='unformatted',
     1     status='old', readonly )
      read( 3 ) ( ( se1linear( i,j ), j = 1,41 ), i = 1,41 )
      close( 3 )

      open( unit=3, file='[KUBAND.HOWARD.MARK]az1p.dat',
     1     access='sequential', form='unformatted',
     1     status='old', readonly )
      read( 3 ) ( ( pa1linear( i,j ), j = 1,41 ), i = 1,41 )
      close( 3 )

      open( unit=3, file='[KUBAND.HOWARD.MARK]el1p.dat',
     1     access='sequential', form='unformatted',
     1     status='old', readonly )
      read( 3 ) ( ( pe1linear( i,j ), j = 1,41 ), i = 1,41 )
      close( 3 )

      return
      end
c     ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
c
c
c     Subroutine: Antenna pattern interpolation.
c     Input: Azimuth and elevation angles in degrees.
c     Output: Interpolated difference, sum, and phase values
c             for all 18 antenna patterns.
c
c
c     ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

      subroutine interp( az, el)

c     ==============================================================
c
c      Linearly interpolate the gain, phase and difference patterns
c
c     ==============================================================

      real a1linear( 41,41 ), e1linear( 41,41 )

      real sa1linear(41,41), se1linear(41,41)

      real pa1linear(41,41), pe1linear(41,41)

      common / linear / a1linear, e1linear

      common / linear1 / sa1linear,se1linear

      common / linear2 / pa1linear,pe1linear

      common / SUDIPH / X,Y,Z,PAZ,PEL

      iax = jint( ( az + 4. ) * 5. )
      iex = jint( ( el + 4. ) * 5. )
      az0 = floatj( iax ) / 5. - 4.
      el0 = floatj( iex ) / 5. - 4.

      iaz = jint ( ( az + 4. ) * 5. ) + 1
      jel = jint ( ( el + 4. ) * 5. ) + 1


c     ------- find azd values -------------------------

      f0 = 10.**( a1linear( iaz,jel      ) /20. )
      f1 = 10.**( a1linear( iaz+1,jel    ) /20. )
      f2 = 10.**( a1linear( iaz,jel+1    ) /20. )
      f3 = 10.**( a1linear( iaz+1,jel+1  ) /20. )

      fa = f0 + (f1-f0)/.2 * ( az-az0 )
      fb = f2 + (f3-f2)/.2 * ( az-az0 )
      fx = fa + (fb-fa)/.2 * ( el-el0 )

      Y = fx
c     ------- find eld values -------------------------

      f0 = 10.**( e1linear( iaz,jel      ) /20. )
      f1 = 10.**( e1linear( iaz+1,jel    ) /20. )
      f2 = 10.**( e1linear( iaz,jel+1    ) /20. )
      f3 = 10.**( e1linear( iaz+1,jel+1  ) /20. )

      fa = f0 + (f1-f0)/.2 * ( az-az0 )
      fb = f2 + (f3-f2)/.2 * ( az-az0 )
      fx = fa + (fb-fa)/.2 * ( el-el0 )


      Z =  fx

c     -------- find azs values ------------------------

      f0 = 10.**(sa1linear(iaz  ,jel    )/20.)
      f1 = 10.**(sa1linear(iaz+1,jel    )/20.)
      f2 = 10.**(sa1linear(iaz  ,jel+1  )/20.)
      f3 = 10.**(sa1linear(iaz+1,jel+1  )/20.)
      fa = f0 +(f1-f0)/.2*(az-az0)
      fb = f2 +(f3-f2)/.2*(az-az0)
      fx = fa +(fb-fa)/.2*(el-el0)


      X = fx

c     ------- find azp values -------------------------

      f0 = pa1linear(iaz  ,jel   )
      f1 = pa1linear(iaz+1,jel   )
      f2 = pa1linear(iaz  ,jel+1 )
      f3 = pa1linear(iaz+1,jel+1 )
      fa = f0 +(f1-f0)/.2*(az-az0)
      fb = f2 +(f3-f2)/.2*(az-az0)
      fx = fa +(fb-fa)/.2*(el-el0)


      PAZ=fx                   !  phase in degrees

c     ------ find elp values --------------------------

      f0 = pe1linear(iaz  ,jel   )
      f1 = pe1linear(iaz+1,jel   )
      f2 = pe1linear(iaz  ,jel+1 )
```

```
 5128            f3 = pel linear(iaz+1,jel+1 )
 5129
 5130            fa = f0 +(f1-f0)/.2*(az-az0)
 5131            fb = f2 +(f3-f2)/.2*(az-az0)
 5132            fx = fa +(fb-fa)/.2*(el-el0)
 5133
 5134            PEL=fx                  ! phase in degrees
 5135
 5136            return
 5137
 5138            end
******
LINES DELETED FROM BASELINE PROGRAM
************

Number of difference sections found: 62
Number of difference records found: 1052

DIFFERENCES /IGNORE=()/MERGED=1/OUTPUT=USER1:[KUBAND.HOWARD.MARK]FINHAC.DIF;1-
    USER1:[KUBAND.HOWARD.MARK]FINSIM1.FOR;8-
    USER1:[KUBAND.HOWARD]HACSIM.FOR;1
```

(THIS PAGE INTENTIONALLY LEFT BLANK)

APPENDIX D

GEOMETRICAL DILUTION OF PRECISION (GDOP) IN TMR RADAR MEASUREMENTS

D-1 INTRODUCTION

This appendix presents the details of the analysis of GDOP. GDOP is the term used to describe the effects of range and range rate measurement errors from sensors at various geometries relative to the target on subsequent calculations of target position and velocity. The problem is best understood by referring to Figure D1, which shows the WSMR range geometry, with the Brass Cap location at the origin. (This is done to simplify the math which follows.)

Each of the three TMR radars measures the range from itself to the target along with line of sight velocity (Range Rate) of the target relative to each radar. These measurements of range, denoted as R1, R2, and R3. are used to compute the X, Y, and Z coordinates of the target, relative to Brass Cap. These will be denoted as X, Y, and Z. Given X, Y, and Z, the range from Brass Cap to the target, R, can be found. Note that the locations of the three radars are denoted by the coordinate sets (X1,Y1,Z1), (X2,Y2,Z2), and (X3,Y3,Z3). Using the above data, range rate (change in R with respect to time) can also be computed.

GDOP occurs when R1, R2, and R3 contain errors. The errors may be bias errors (constant) or randomly varying (stochastic). The overall effect of errors in R1, R2, and R3 is that they cause the computed values of X, Y, and Z , and thus R, to be in error. The detailed analysis of this phenomenon will be developed in the rest of this section. Examples of its effect on the WSMR experimental data will also be presented.

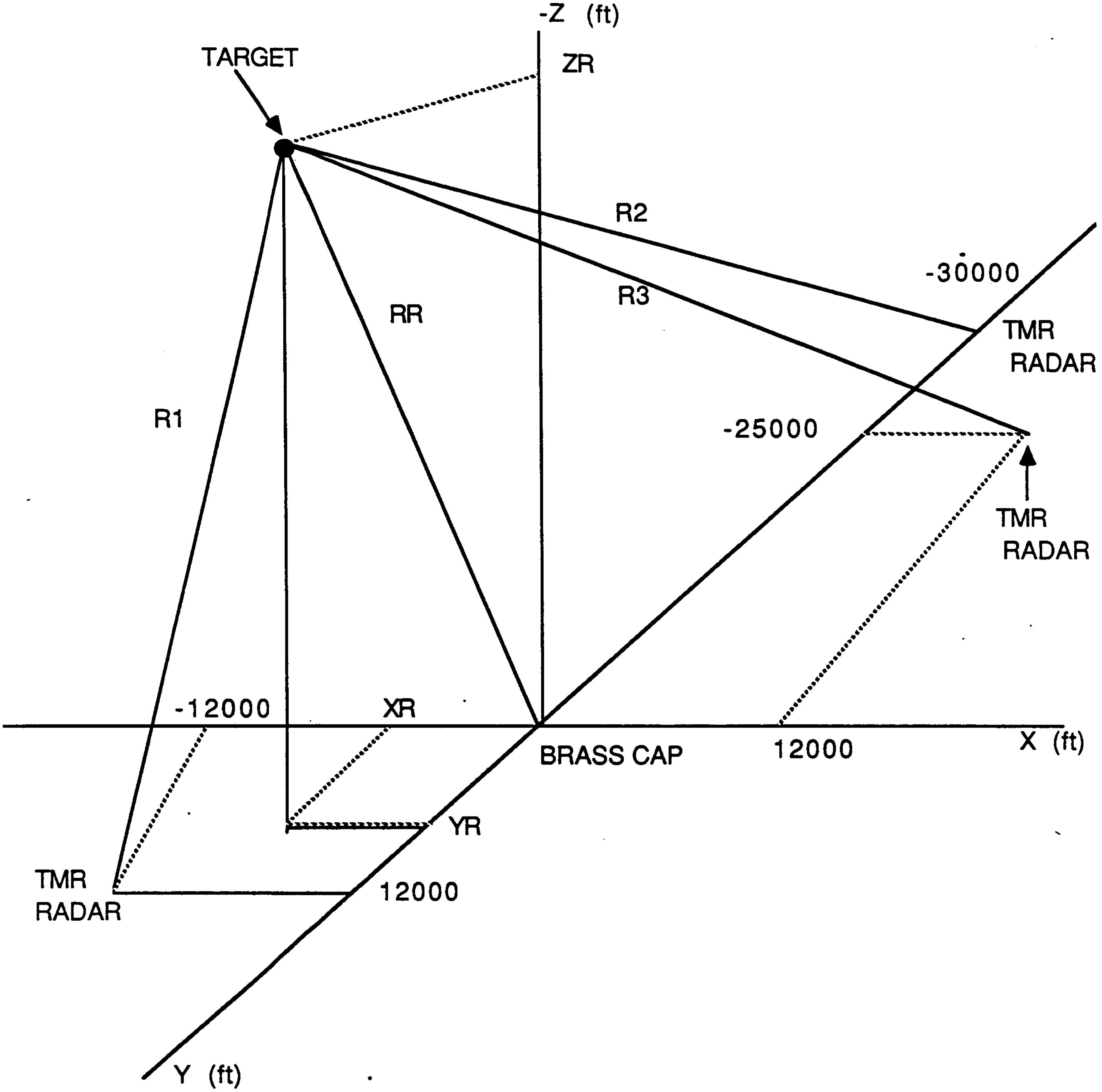


FIGURE D1 GEOMETRY OF TMR RADARS AT WSMR RELATIVE TO BRASS CAP

D-2 RANGE ERRORS

D-2.1 Example

As a intuitive introduction to the range error problem, we will first consider a two dimensional problem shown in Figure D2. In Figure D2, the target is approximately midway between the radars, and is nearly above the "Brass Cap" or reference point. The true range from Brass Cap to target is R.

This is the range we are trying to measure with the radars. The true ranges from the radars to the target are R1 and R2. If the exact values of R1 and R2 were measured by the radars then X and Y could be found by solving the pair of equations below:

$$R1^2 = (X - X1)^2 + (Y - Y1)^2 \tag{1}$$
$$R2^2 = (X - X2)^2 + (Y - Y2)^2$$

And R could be found by substituting X and Y into the equation:

$$R^2 = X^2 + Y^2 \tag{2}$$

This solution is graphically shown in the figure as point P, which is the intersection of the two circles of radii R1 and R2.

If each of the radar range measurements was in error by an amount DR, then the apparent ranges measured would be R1+DR and R2+DR. These ranges are shown as circular arcs in Figure D2. Note that their intersection is at point Q, which has coordinates XQ and YQ. As can be seen in the figure, the range from Brass Cap to point Q is significantly different from the true range. Note that the values XQ and YQ would be obtained by substituting R1+DR and R2+DR in the set of equations above.

The situation portrayed in Figure D2 is "worst case" in the sense that small errors in R1 and R2 produce large errors in R. This is because of the geometry of the situation. Although it will not be described here in detail, the reader should have little trouble convincing himself that other

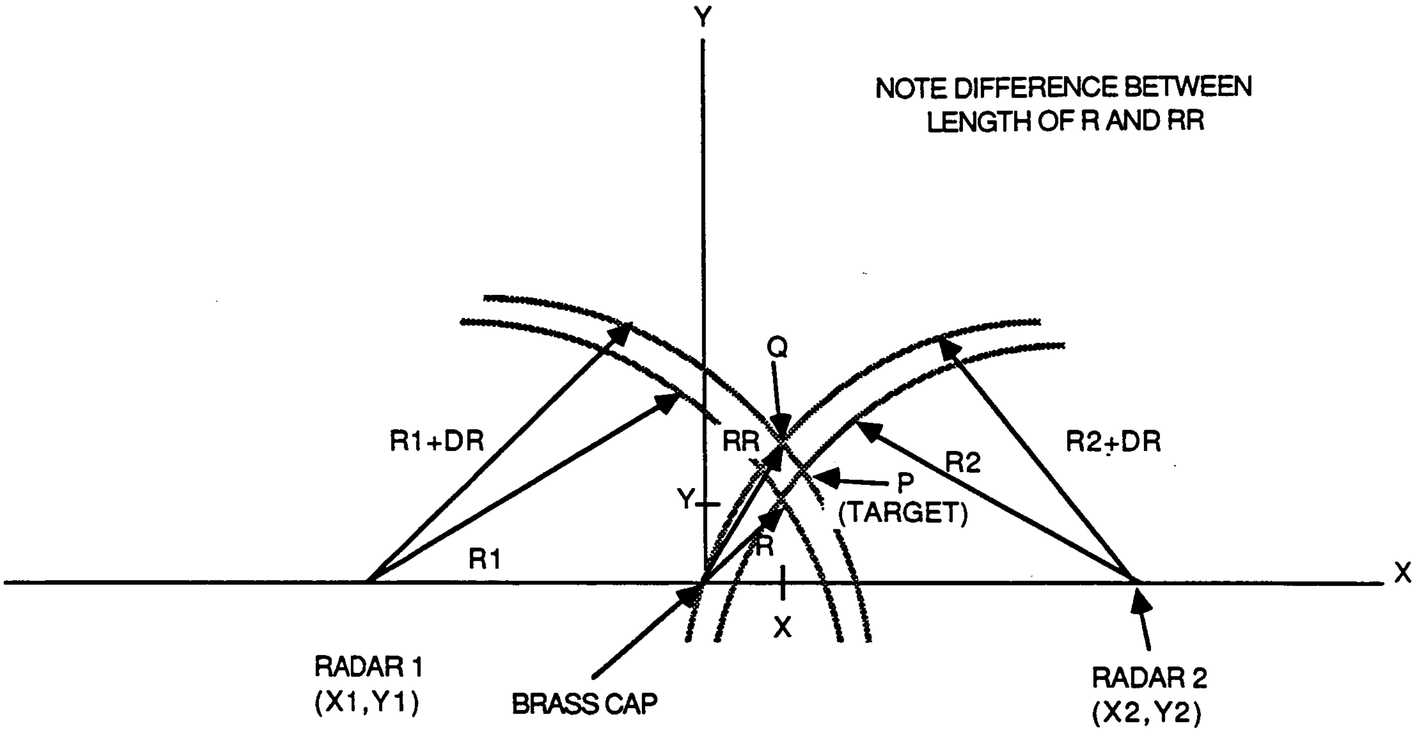


FIGURE D2 TWO DIMENSIONAL DIAGRAM SHOWING EFFECTS OF RADAR RANGE ERRORS ON ESTIMATE OF TARGET RANGE

geometries, for example where the target is far removed from the radars, produce smaller errors.

D-2.2 Mathematical Analysis

We now consider the general three dimensional case. The notation which is used below is consistent with Figure D1.

The true range from Brass Cap to the target is given by:

$$R^2 = X^2 + Y^2 + Z^2 \tag{3}$$

The range as computed using data from the three radars is given by:

$$RR^2 = XR^2 + YR^2 + ZR^2 \tag{4}$$

where XR, YR, and ZR are computed from the radar range data using the set of equations below:

$$\begin{aligned} R1^2 &= (XR - X1^2) + (YR - Y1)^2 + (ZR - Z1)^2 \\ R2^2 &= (XR - X2^2) + (YR - Y2)^2 + (ZR - Z2)^2 \\ R2^3 &= (XR - X3^2) + (YR - Y3)^2 + (ZR - Z3)^2 \end{aligned} \tag{5}$$

In general XR, YR, and ZR will contain errors, because of errors in R1, R2, and R3. To analyze the effects of errors in R1, R2, and R3 on XR, YR, and ZR we first take the total derivative of the expressions for R1, R2, and R3 to get the system of equations shown here. Note that the derivatives have been represented by DX, DY, and DZ.

$$\begin{aligned} R1DR &= (XR - X1)DX + (YR - Y1)DY + (ZR - Z1)DZ \\ R2DR &= (XR - X2)DX + (YR - Y2)DY + (ZR - Z2)DZ \\ R3DR &= (XR - X3)DX + (YR - Y3)DY + (ZR - Z3)DZ \end{aligned} \tag{6}$$

In this system of equations, R1, R2, and R3 are measured by the radars. XR, YR, and ZR are computed as above, X1, Y1, Z1, etc. are known from the range survey data, and DR is assumed to be known from range calibration data. DR may be deterministic, as in a fixed range bias, or may vary statistically. A general model for DR is:

$$(7) \qquad DR = DR + U$$

where U is a random variable with zero mean and some specified variance and probability density function. For the majority of data here, DR will be assumed to be a constant. Existing data from WSMR indicates that DR is approximately 10 feet.

By solving this system of equations, DX, DY, and DZ may be found as a function of the range errors associated with the three radars. The DX, DY, and DZ values may subsequently be used to correct the values XR, YR, and ZR, and thus improve the range estimate RR.

D-2.3 Experimental Data

Figures D3 and D4 are plots of range error computed from two sets of WSMR experimental data. Figure D3 is the range error observed from tracking a target which was close to the Brass Cap location, and at a relatively low altitude. Note that the range errors are large, on the order of 150 feet. Figure D4 is the range error computed for a target which was at a higher altitude and considerably longer range. Note that in this instance, the range errors are approximately 20 feet.

These results are consistent with the example presented at the beginning of this section. They were computed assuming the radar range errors were the same for all radars, and were equal to 10 feet.

D-3 RANGE RATE (VELOCITY) ERRORS

D-3.1 Example

For the example below, refer to Figure D1. Given a target close to the Brass Cap at a very low altitude the range rate measurement of the 3

SIM DATA PROFILE H30SKAF

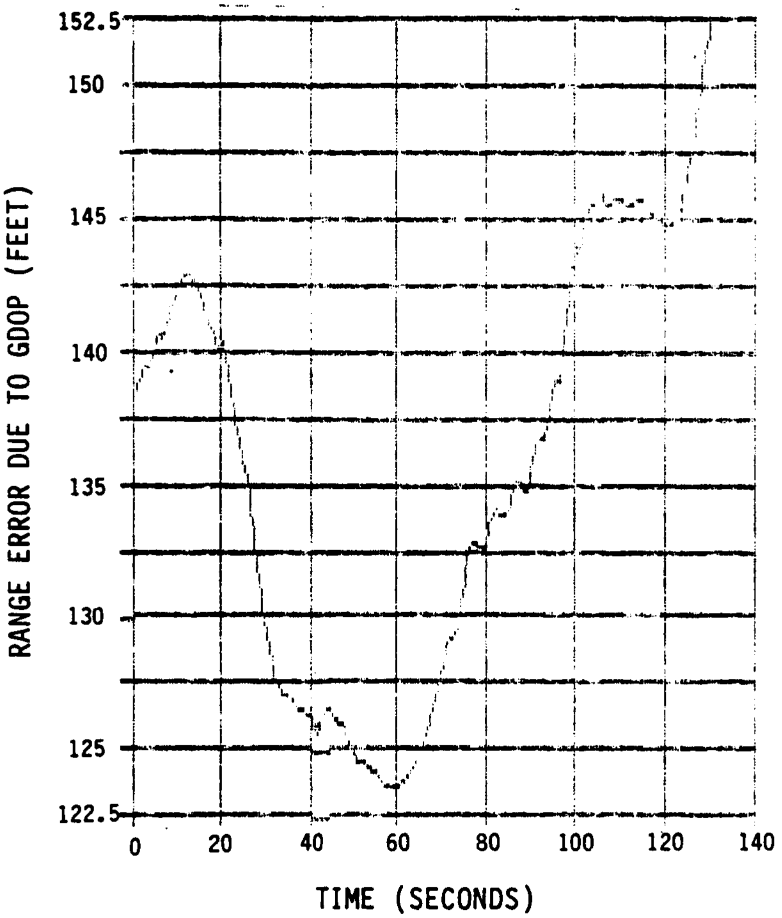


FIGURE D3 RANGE ERRORS DUE TO GDOP
FOR TARGET NEAR BRASS CAP

SIM DATA PROFILE HJ146AD

TEST DATE 10-1-85

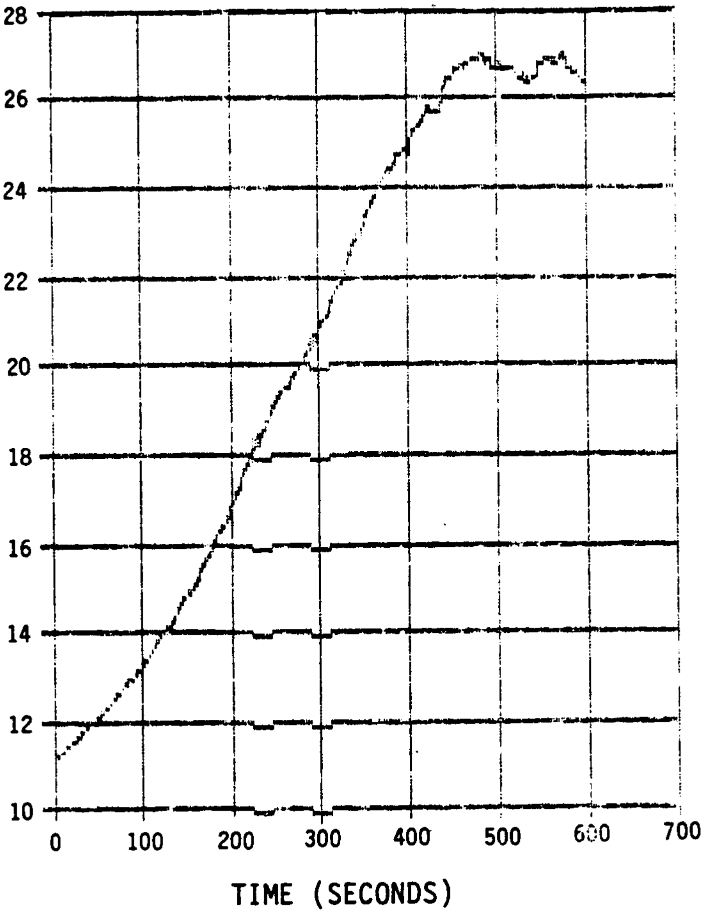


FIGURE D4 RANGE ERRORS DUE TO GDOP
FOR TARGET AWAY FROM BRASS CAP

TMR radars would not be affected significantly by the Vz component of the target. Inversely a small uncertainty in range rate translates into a large uncertainty in the TMR predicted Vz component. With this scenario, the target is practically above the Ku-Band Radar and the actual Vz component affects the Ku-Band range rate measurement significantly. In this case one would expect the GDOP effect to be large. A target whose location was not close to the Brass Cap and had a large altitude one would expect the GDOP effect to be small. Examples of real range data which support this example will be presented in Section D-3.3.

D-3.2 Mathematical Analysis

The range rate of the target relative to a radar can be determined by taking the time derivative of the range equation which is repeated below for reference

(8) $$R1^2 = (XR - X1)^2 + (YR - Y1)^2 + (ZR - Z1)^2$$

When this is done, we obtain equations of the form shown below. The equation shown is for radar R1.

(9) $$R1\frac{DR}{dt} = (XR - X1)\frac{DX}{dt} + (YR - Y1)\frac{dY}{dt} + (ZR - Z1)\frac{dZ}{dt}$$

The sensitivity of X, Y, and Z to small errors in the TMR range rate measurements can be determined by taking the total derivative of the range rate equations and simultaneously solving the set of equations which result. The form of the range rate equation is shown below. The equation shown is for radar R1. The dot (.) superscript denotes derivative.

(10) $$R\dot{1}DR+R1D\dot{R} = XRD\dot{X}+YRD\dot{Y}+ZRD\dot{Z}+(XR-X1)\dot{D}X+(YR-Y1)\dot{D}Y+(ZR-Z1)\dot{D}Z$$

Reqrouping the terms of equation 10, we obtain a more convenient form:

(11) $$\dot{R}1DR+R1\dot{D}R - (XRD\dot{X}+YRD\dot{Y}+ZRD\dot{Z}) = (XR-X1)\dot{D}X+(YR-Y1)\dot{D}Y+(ZR-Z1)\dot{D}Z$$

For compactness, we will adopt matrix notation to write the complete set of equations for the three radars. Rewrite equation 11 as:

$$G \ X \ K = H \ X \ DEL \tag{12}$$

Where G is given by:

$$G = \begin{bmatrix} \dot{R1} & R1 & \dot{XR} & \dot{YR} & \dot{ZR} \\ \dot{R2} & R2 & \dot{XR} & \dot{YR} & \dot{ZR} \\ \dot{R3} & R3 & \dot{XR} & \dot{YR} & \dot{ZR} \end{bmatrix} \tag{13}$$

and K is:

$$K = \begin{bmatrix} DR \\ \dot{DR} \\ -DX \\ -DY \\ -DZ \end{bmatrix} \tag{14}$$

H is the matrix:

$$H = \begin{bmatrix} (XR - X1) & (YR - Y1) & (ZR - Z1) \\ (XR - X2) & (YR - Y2) & (ZR - Z2) \\ (XR - X3) & (YR - Y3) & (ZR - Z3) \end{bmatrix} \tag{15}$$

and DEL is the vector:

$$DEL = \begin{bmatrix} \dot{DX} \\ \dot{DY} \\ \dot{DZ} \end{bmatrix} \tag{16}$$

DX, DY, and DZ are determined from the equations derived in the Range Error Section above. DR and its derivative $\dot{D}R$ are the range error and range rate error associated with the radars. They are assumed to be the same for all three radars, and known from independent measurements. The range error (DR) is assumed to be constant, while the range rate error ($\dot{D}R$) is assumed to be stochastic with zero mean and a standard deviation of 0.2 ft/sec..

The quantities of interest in the above equations are the range rate errors $\dot{D}X$, $\dot{D}Y$, and $\dot{D}Z$. Normal values of XR, YR, ZR, $\dot{X}R$, $\dot{Y}R$, and $\dot{Z}R$ are available as data from the TMR radar solution.

To calculate the variance of $\dot{D}X$, $\dot{D}Y$, and $\dot{D}Z$ we form the covariance matrix of DEL as shown in the equations below. The diagonal elements of the matrix P are the variances of $\dot{D}X$, $\dot{D}Y$, and $\dot{D}Z$ respectively.

(17a) $$P = VAR\ [DEL]$$

(17b) $$P = H^{-1}G\ E[KK^{T}]G^{T}(H^{-1})^{T} - E[DEL]E[DEL^{T}]$$

where

(18) $$E[K] = \begin{bmatrix} DR \\ 0.0 \\ \\ -DX \\ -DY \\ -DZ \end{bmatrix}$$

and

$$(19) \qquad E[KK^T] = \begin{bmatrix} DR^2 & D\dot{R}DR & -DXDR & -DYDR & -DZDR \\ DRD\dot{R} & D\dot{R}^2 & -DXD\dot{R} & -DYD\dot{R} & -DZD\dot{R} \\ -DXDR & -DXDR & DX^2 & DXDY & DXDZ \\ -DYDR & -DYD\dot{R} & DXDY & DY^2 & DYDZ \\ -DZDR & -DZD\dot{R} & DXDZ & DYDZ & DZ^2 \end{bmatrix}$$

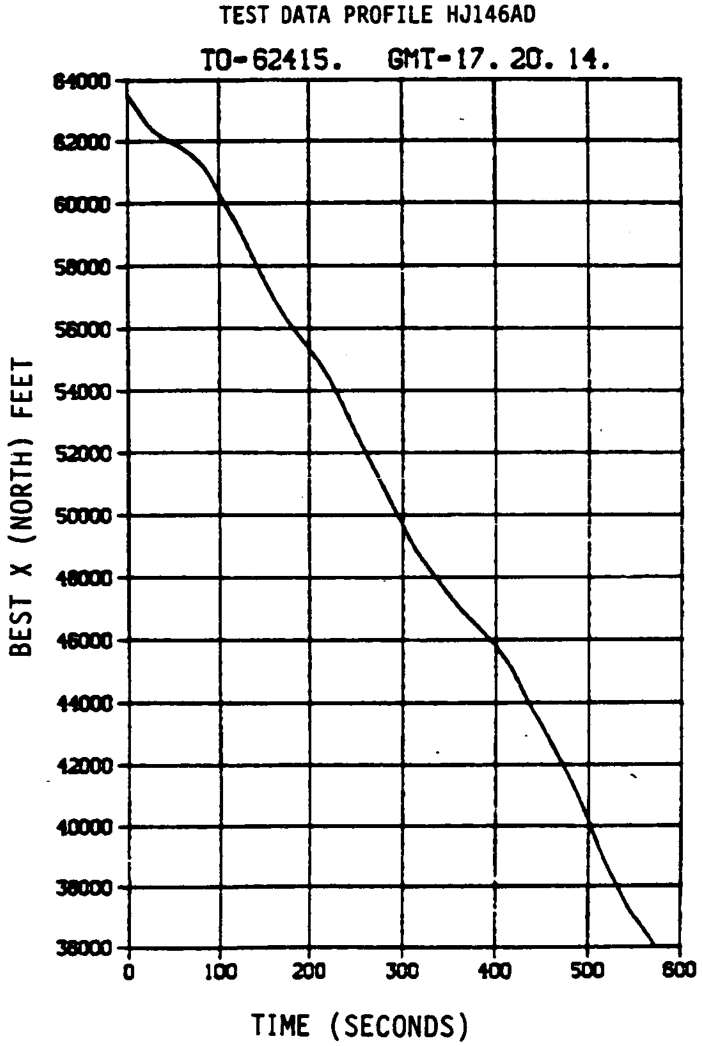
Since $D\dot{R}$ is 0 mean with .2 ft/sec standard deviation, the $E(KK^T)$ reduces to:

$$E(KK^T) = \begin{bmatrix} DR^2 & 0 & -DXDR & -DYDR & -DZDR \\ 0 & .04 & 0 & 0 & 0 \\ -DXDR & 0 & DX^2 & DXDY & DXDZ \\ -DYDR & 0 & DXDY & DY^2 & DYDZ \\ -DZDR & 0 & DXDZ & DYDZ & DZ^2 \end{bmatrix}$$

The effect of the TMR range rate measurement errors on the predicted range rate at the Ku-Band Radar site is approximated by taking the dot product of the Brass Cap range unit vector with the velocity error vector. This approximation is valid because the Brass Cap and the Ku-Band radar were separated by only a few feet and the coordinate transformations involved would not affect the results significantly.

D-3.3 Examples

Figures D5 through D12 show two cases where velocity errors were computed from WSMR experimental data. Figures D5 through D8 demonstrate that for a target at low altitude and close to the Brass Cap the GDOP effect is significant. Figures D5, D6, and D7 show the X, Y, and Z range from the Brass Cap. Figure D8 shows the range rate errors which were computed from the data, using the procedures above. Figures D9 through D12 show that at high altitudes the GDOP effect is minimal. Figures D9 through D11 show the X, Y and Z coordinates which were measured, and Figure D12 the range rate error. Note that the errors in the longer range case, shown in Figure D12 are less than those shown in D8.


TEST DATA PROFILE HJ146AD

TO-62415. GMT-17.20.14.

FIGURE D5 X COORDINATES OF TARGET

TEST DATA PROFILE HJ146AD

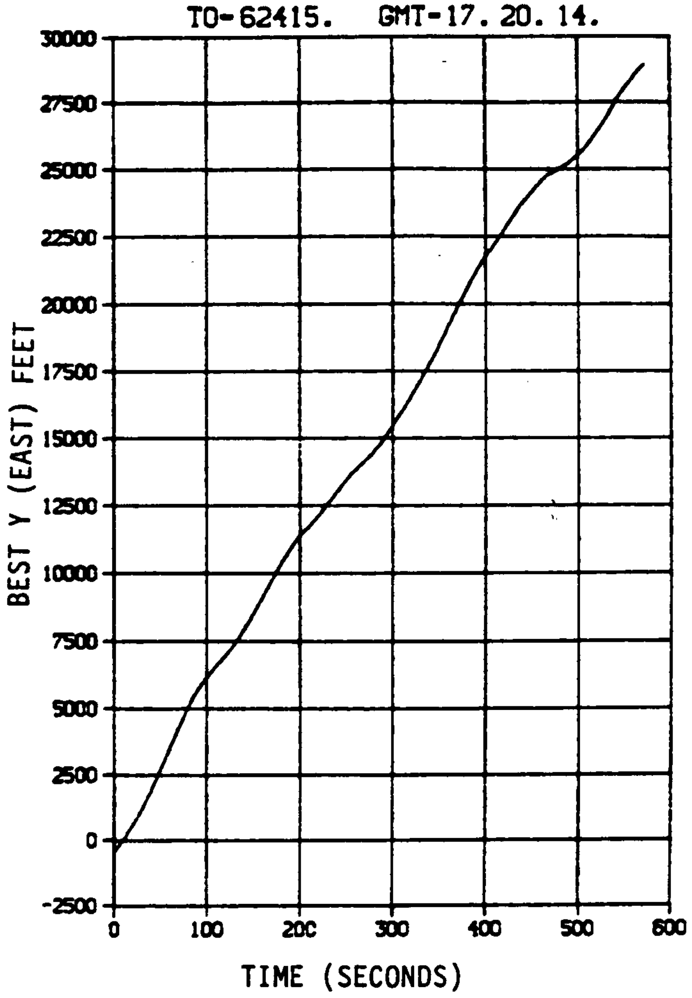


FIGURE D6 Y COORDINATES OF TARGET

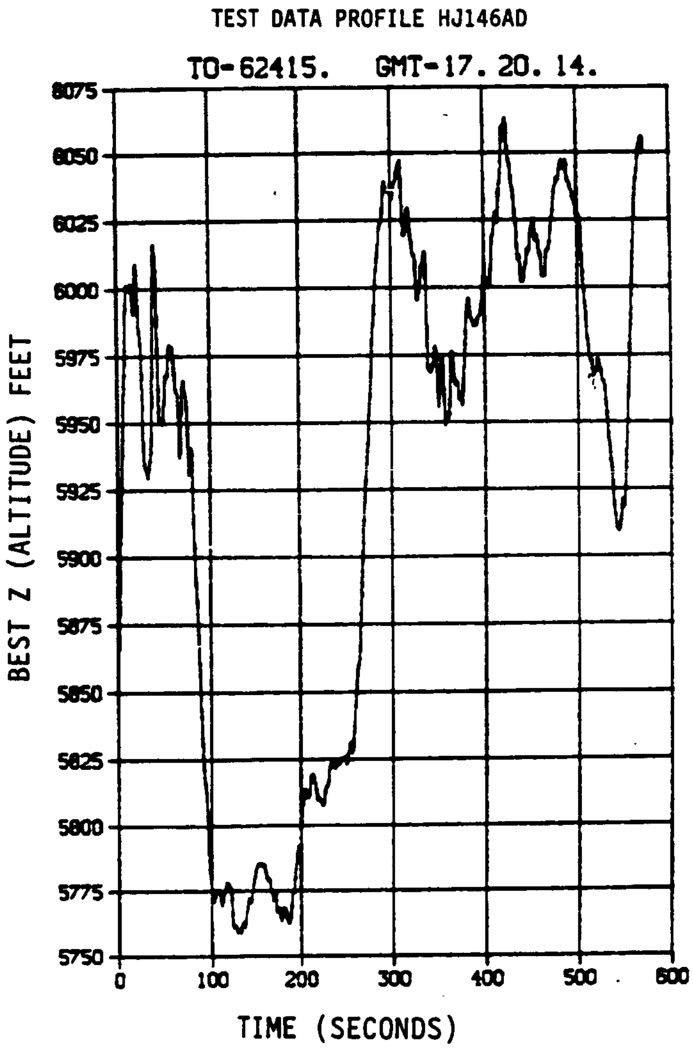
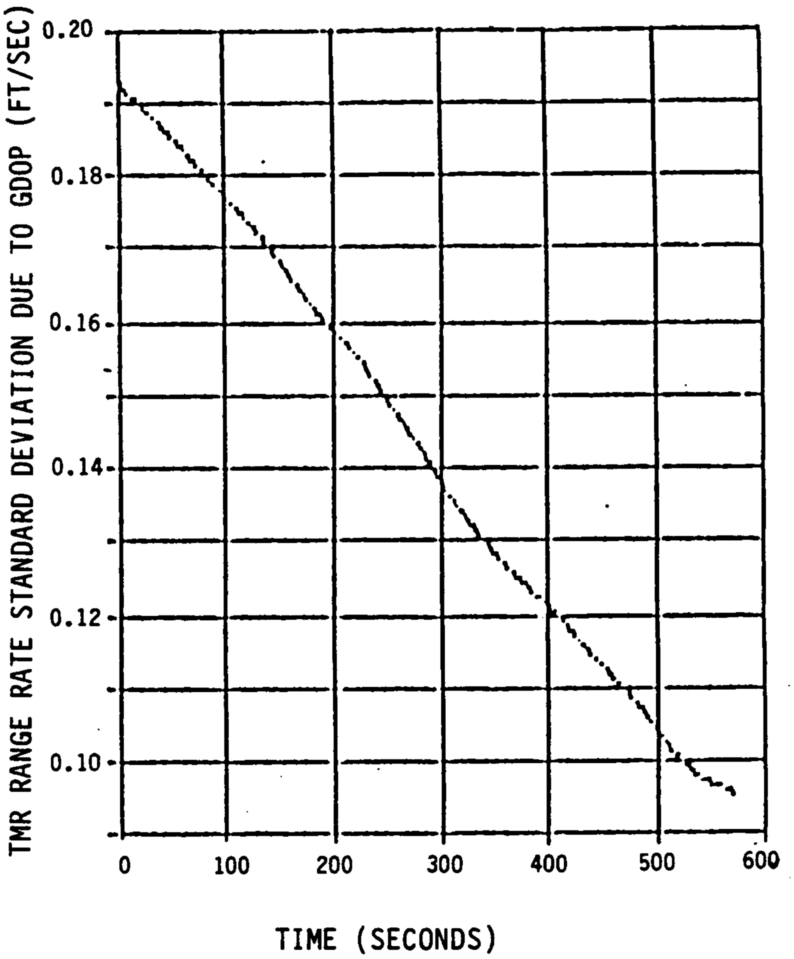


FIGURE D7 Z COORDINATES OF TARGET

SIM DATA PROFILE HJ146AD

TEST DATE 10-5-85



MEAN= 0.142

FIGURE D8 RANGE RATE STANDARD DEVIATION DUE TO GDOP

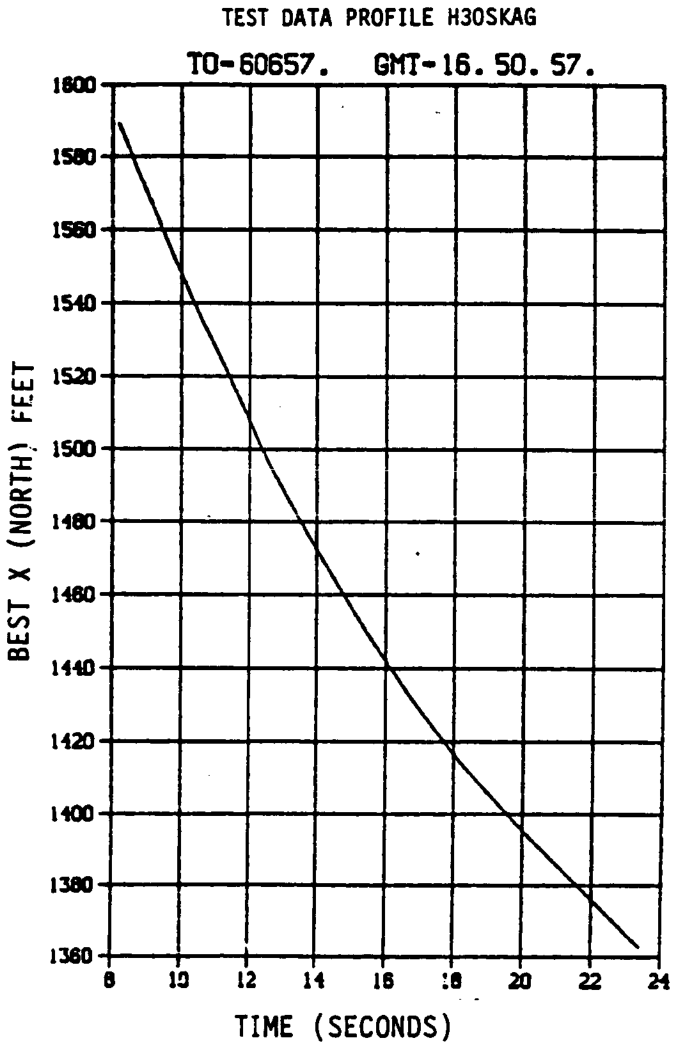


FIGURE D9 X COORDINATES OF TARGET

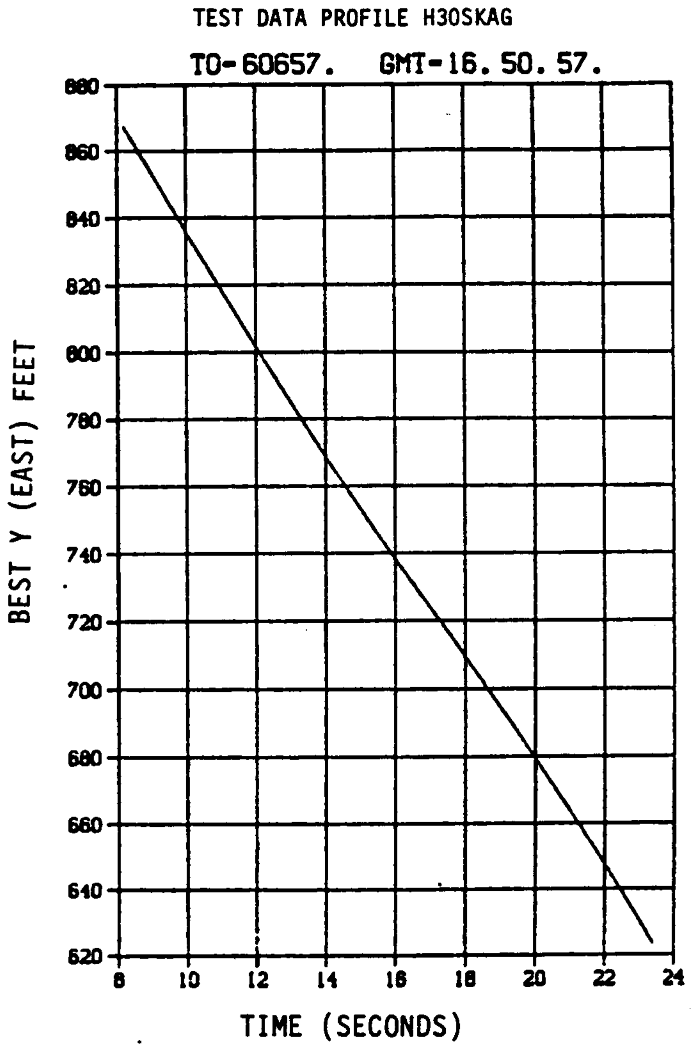


FIGURE D 10 Y COORDINATES OF TARGET

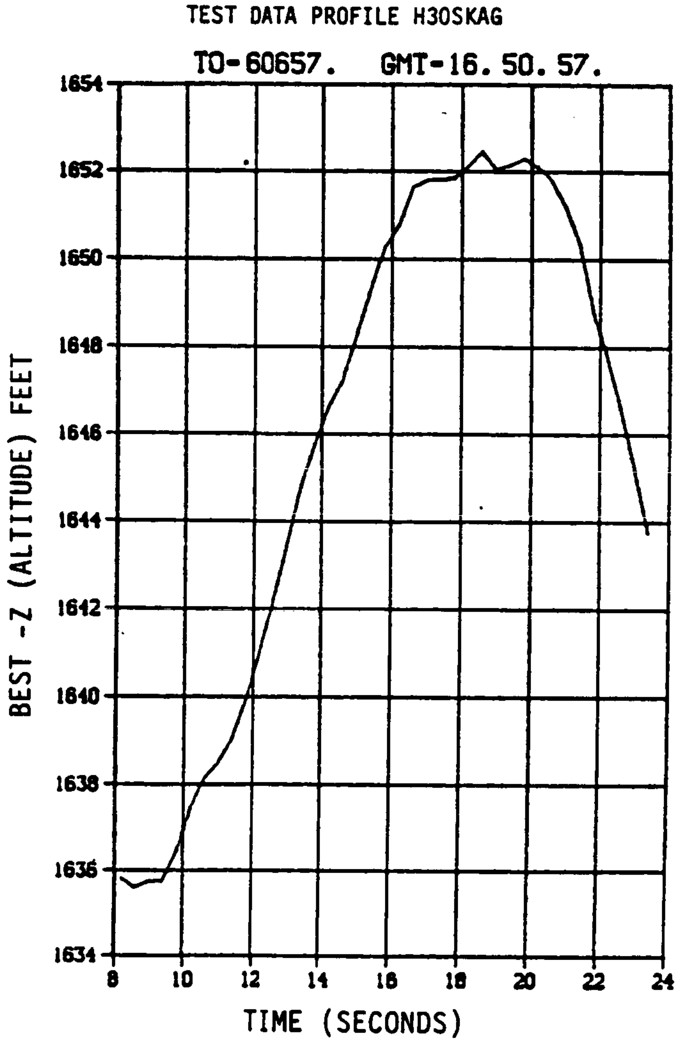


FIGURE D11 Z COORDINATES OF TARGET

-6

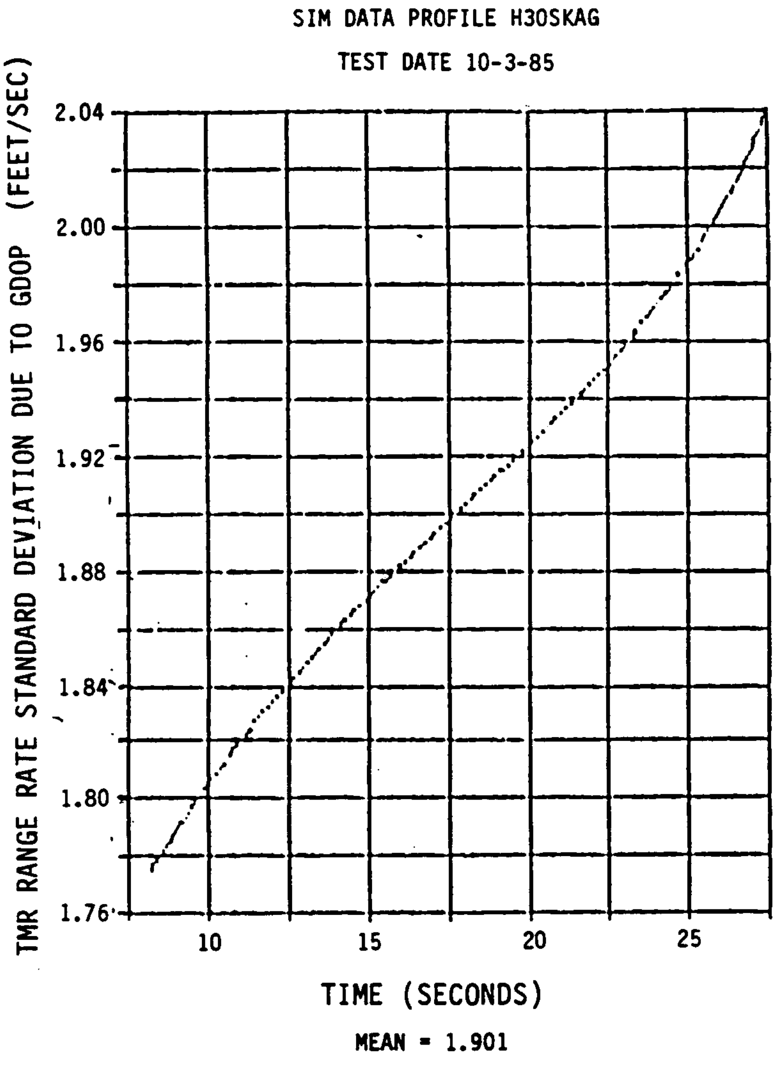


FIGURE D12 RANGE RATE STANDARD DEVIATION DUE TO GDOP

D-4 GDOP ANALYSIS PROGRAM LISTING

Figure D13 is a listing of the program which was used to perform the GDOP analysis. Its inputs are the same WSMR data files used by the other analysis programs, and similar output plots are available.

```
      SUBROUTINE GDOP(RDOTSD)
C
C
C
      COMMON/TMR/X,Y,Z,XDOT,YDOT,ZDOT
      REAL H(3,3),G(3,5),DK(5,1),DKT(1,5),GT(5,3)
      REAL HINV(3,3),HINVT(3,3),M(3,1),MT(1,3),R(3,1)
      REAL XYZMAT(3,3),DELT(3,1),HINVG(3,5),DKDKT(5,5)
      REAL TEMP(3,5),GTHINVT(5,3),MMT(3,3),MDKT(3,5)
      REAL DKMT(5,3),COV1(3,3),COV2(3,3),COV3(3,3),COV(3,3)
      DATA ILOOP/1/
C
C     READ X Y Z POSITIONS OF VECTORS
C
      IF (ILOOP.EQ.1) THEN
      OPEN(UNIT=8,FILE='POS.DAT',STATUS='OLD')
C
      DO I=1,3
        READ(8,*) XYZMAT(I,1),XYZMAT(I,2),XYZMAT(I,3)
      END DO
C
C     READ IN DELTA RANGE,DELTA RANGE RATE MEAN AND
C     DELTA RANGE RATE VARIANCE
C
      READ(8,*) DR,DRDOT,DRDOTSQ
      ILOOP=0
      END IF

      DO I=1,3
         H(I,1)=X-XYZMAT(I,1)
         H(I,2)=Y-XYZMAT(I,2)
         H(I,3)=Z-XYZMAT(I,3)
         G(I,3)=XDOT
         G(I,4)=YDOT
         G(I,5)=ZDOT
         R(I,1)=SQRT(H(I,1)**2+H(I,2)**2+H(I,3)**2)
         G(I,1)=(H(I,1)*XDOT+H(I,2)*YDOT+H(I,3)*ZDOT)/R(I,1)
         G(I,2)=R(I,1)
      END DO
C
C     DO MATRIX TRANSPOSES INVERSES AND MULTIPLICATIONS
C
      CALL MATINV(H,HINV)
      CALL MATMULT(HINV,R,DELT,3,3,1)
C
C     SOLVE FOR DELTA X DELTA Y AND DELTA Z
C
```

FIGURE D13 SOURCE LISTING OF GDOP ANALYSIS PROGRAM

PAGE 1

```
       DO I=1,3
         DELT(I,1)=DELT(I,1)*DR
       END DO
C      SET UP MATRIX DELTA K
C
       DK(1,1)=DR
       DK(2,1)=DRDOT
       DK(3,1)=-DELT(1,1)
       DK(4,1)=-DELT(2,1)
       DK(5,1)=-DELT(3,1)
       CALL MATMULT(HINV,G,HINVG,3,3,5)
       CALL MATMULT(HINVG,DK,M,3,5,1)
       CALL MATTRAN(M,MT,3,1)
       CALL MATTRAN(G,GT,3,5)
       CALL MATTRAN(DK,DKT,5,1)
       CALL MATTRAN(HINV,HINVT,3,3)
       CALL MATMULT(DK,DKT,DKDKT,5,1,5)
C
C      SET DKDKT(2,2) TO VARIANCE OF VELOCITY ERROR
C

       DKDKT(2,2)=DRDOTSQ
       CALL MATMULT(GT,HINVT,GTHINVT,5,3,3)
       CALL MATMULT(M,DKT,MDKT,3,1,5)
       CALL MATMULT(DK,MT,DKMT,5,1,3)
       CALL MATMULT(HINVG,DKDKT,TEMP,3,5,5)
       CALL MATMULT(TEMP,GTHINVT,COV1,3,5,3)
       CALL MATMULT(MDKT,GTHINVT,COV2,3,5,3)
       CALL MATMULT(HINVG,DKMT,COV3,3,5,3)
       CALL MATMULT(M,MT,MMT,3,1,3)
C
C      FORM COVARIANCE MATRIX
C
       DO I=1,3
          DO J=1,3
               COV(I,J)=COV1(I,J)-COV2(I,J)-COV3(I,J)+MMT(I,J)
          END DO
       END DO
       XDSD=SQRT(COV(1,1))
       YDSD=SQRT(COV(2,2))
       ZDSD=SQRT(COV(3,3))
       RANGE=SQRT(X**2+Y**2+Z**2)
       RDOTSD=abs((X*XDSD+Y*YDSD+Z*ZDSD)/RANGE)
       RETURN
       END
       SUBROUTINE MATMULT(A,B,C,M,N,IP)
       DIMENSION A(M,N),B(N,IP),C(M,IP)
C
       DO I=1,M
          DO J=1,IP
             C(I,J)=0.0
             DO K=1,N
               C(I,J)=C(I,J)+A(I,K)*B(K,J)
             END DO
          END DO
       END DO
       RETURN
       END
C
       SUBROUTINE MATTRAN(A,B,IROW,ICOL)
       DIMENSION A(IROW,ICOL),B(ICOL,IROW)
```

FIGURE D13 SOURCE LISTING OF GDOP ANALYSIS PROGRAM

PAGE 2

```
C
      DO I=1,IROW
         DO J=1,ICOL
            B(J,I)=A(I,J)
         END DO
      END DO
      RETURN
      END
C
      SUBROUTINE MATINV(A,B)
      DIMENSION A(3,3),B(3,3)
C
      DET=A(1,1)*A(2,2)*A(3,3)+A(1,2)*A(2,3)*A(3,1)
     #    +A(1,3)*A(2,1)*A(3,2)-A(1,3)*A(2,2)*A(3,1)
     #    -A(1,2)*A(2,1)*A(3,3)-A(1,1)*A(2,3)*A(3,2)

      B(1,1)=A(2,2)*A(3,3)-A(2,3)*A(3,2)
      B(2,1)=A(3,1)*A(2,3)-A(2,1)*A(3,3)
      B(3,1)=A(2,1)*A(3,2)-A(3,1)*A(2,2)
      B(1,2)=A(1,3)*A(3,2)-A(1,2)*A(3,3)
      B(2,2)=A(1,1)*A(3,3)-A(3,1)*A(1,3)
      B(3,2)=A(3,1)*A(1,2)-A(1,1)*A(3,2)
      B(1,3)=A(1,2)*A(2,3)-A(2,2)*A(1,3)
      B(2,3)=A(2,1)*A(1,3)-A(1,1)*A(2,3)
      B(3,3)=A(1,1)*A(2,2)-A(1,2)*A(2,1)
      DO I=1,3
         DO J=1,3
               B(J,I)=B(J,I)/DET
         END DO
      END DO
      RETURN
      END
```

FIGURE D13 SOURCE LISTING OF GDOP ANALYSIS PROGRAM

PAGE 3

APPENDIX E: EFFECTS OF COORDINATE MISALIGNMENT ON DELTA ROLL ANGLES

If we start with two coordinate systems that have the same origins but are not aligned a point in space will have two sets of coordinates; (X,Y,Z) and (X',Y',Z'), as shown in Figure E-1. In our particular case, we let the (X,Y,Z) system represent where the TMR2KU subroutine says the radar is pointing and the (X',Y',Z') system represent where the shuttle radar actually is pointing.

It is possible to go from the (X,Y,Z) system to the (X',Y',Z') system using the rotation matrices:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos A & \sin A \\ 0 & -\sin A & \cos A \end{bmatrix} \begin{bmatrix} \cos B & 0 & \sin B \\ 0 & 1 & 0 \\ -\sin B & 0 & \cos B \end{bmatrix} \begin{bmatrix} \cos C & \sin C & 0 \\ -\sin C & \cos C & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

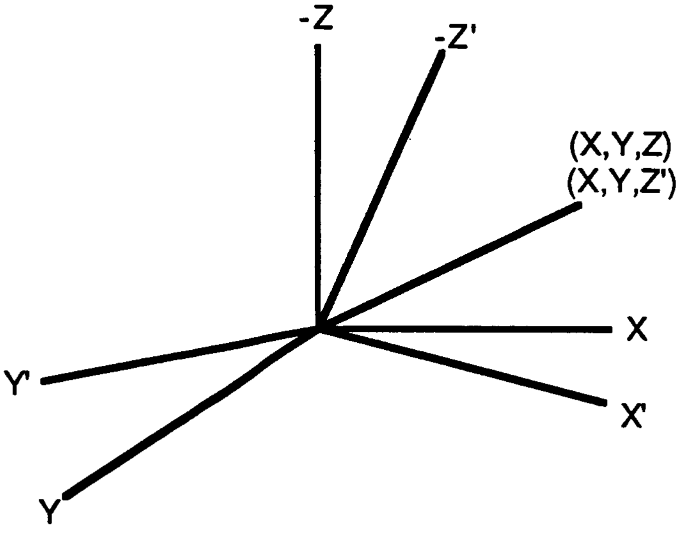


FIGURE E-1 TWO COORDINATE SYSTEMS HAVING SAME ORIGIN, BUT UNALIGNED AXES where A, B, and C are the rotation angles about the coordinate axes. Multiplying through, we obtain

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} (\cos B \cos C) & (\cos B \sin C) & (\sin B) \\ (-\sin C \cos A - \sin B \cos C \sin A) & (\cos A \cos C - \sin A \sin B \sin C) & (\sin A \cos B) \\ (\sin A \sin C - \cos A \sin B \cos C) & (-\sin A \cos C - \cos A \sin B \sin C) & (\cos A \cos B) \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

If we use the small angle approximations of

$$\sin u = u$$
$$\cos u = 1$$

we get

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} 1 & C & B \\ (-C - AB) & (1 - ABC) & A \\ (AC - B) & (-A - BC) & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

or

$$\begin{aligned} X' &= X + YC + ZB \\ Y' &= X(-C - AB) + Y(1 - ABC) + ZA \\ Z' &= X(AC - B) + Y(-A - BC) + Z \end{aligned}$$

if we let

$$\begin{aligned} X' &= X + \Delta X \\ Y' &= Y + \Delta Y \\ Z' &= Z + \Delta Z \end{aligned}$$

we find

$$\begin{aligned} \Delta X &= YC + ZB \\ \Delta Y &= ZA - XC - XAB - YABC \\ \Delta Z &= -XB - YA + XAC - YBC \end{aligned}$$

where these deltas are the errors caused by the rotation angles.

We can now see how the rotation angles would affect delta roll.

By definition,

$$\text{Roll} = \text{ARCTAN}(Y/Z)$$

$$\text{Roll}' = \text{ARCTAN}(Y'/Z')$$

$$\text{Roll} = \text{Roll}' - \text{Roll}$$

If we use small roll angle data, we can approximate

$$\text{ARCTAN}(Y/Z) = Y/Z$$

This makes

$$\text{Roll} = \frac{Y'}{Z'} - \frac{Y}{Z}$$

$$= \frac{Y + \Delta Y}{Z + \Delta Z} - \frac{Y}{Z} = \frac{Z\ \Delta Y - Y\ \Delta Z}{Z^2\left(1 + \frac{\Delta Z}{Z}\right)}$$

using the approximation

$$\frac{1}{1 + X} = 1 - X \quad \text{for small } x$$

$$\Delta\text{Roll} = \frac{Z\ \Delta Y - Y\ \Delta Z - \quad Y\ \Delta Z + \ \Delta Z^2 \underline{Y}}{Z^2}$$

The last term in the numerator, Z^2Y_1 is neglibible for the trajectories used. This leaves

$$\Delta\text{Roll} = \frac{Z\ \Delta Y - Y\ \Delta Z - \quad Y\ \Delta Z}{Z^2}$$

If we now substitute in delta Y and Z from before

$$\begin{aligned}\Delta \mathrm{Roll} = & - XZAB - YZA^2 + XZA^2C - YZABC \\ & + X^2BC + YZAC - Z^2AC^2 + XYBC^2 \\ & + X^2AB^2 + XYA^2B - X^2A^2BC + XYAB^2C\end{aligned}$$

If we keep second order and above terms

$$\begin{aligned}\Delta \mathrm{Roll} = & A(1 + Y^2/Z^2) + B(XY/Z^2) + C(-X/Y) \\ & + A^2(Y/Z) - ACXY/Z^2 + BC((Y^2 - X^2/Z^2)\end{aligned}$$

for the available trajectories, the last two terms become negligible and we have

$$\Delta \mathrm{Roll} = A(1 + Y^2/Z^2) + B(XY/Z^2) + C(-X/Y) + A^2(Y/Z)$$

APPENDIX F - TARGET ACCELERATION EFFECTS

F-1 INTRODUCTION AND ANALYSIS

In order to predict the current velocity of a target, the Ku-Band Radar computes a velocity discriminant. This computation is made assuming that there is no target acceleration. The presence of target acceleration has an adverse effect on the ability of the radar to measure velocity because bias errors are introduced into the velocity discriminant calculation.

In order to form the velocity discriminant, the radar performs the following steps:

1) Transmits 16 pulses per time slot, 4 time slots per frequency, over 5 frequencies with a null time equal to 1/PRF between each time slot.

2) For each time slot the radar performs a DFT from the early range gate and a DFT from the late range gate.

3) Sums up the magnitudes of the "low" filter bin for all range gates, time slots, and frequencies. Similar processing is performed for the "high" filter bin.

4) A velocity discriminant is formed by computing log (low/high).

5. Computes fractional position within a filter by an inverse mapping of the velocity discriminant.

6. Computes velocity estimate from knowledge of the center filter number and the fractional displacement from the center.

If a target is accelerating during a time slot, the effect of the acceleration is to "slide" the target across DFT frequency bins. The contents of a bin are thus the average of the outputs from the various frequencies which the acceleration produced during a time slot. The practical effect of this phenomenon is minimal in many cases because the acceleration values likely to be encountered, and the time slot are both small - the latter is 16 times the reciprocal of the PRF.

Averaging of the individual DFT responses prior to using the velocity discriminant function produces a smoothing effect, which damps acceleration effects. It has been observed that the combination of this averaging, and the inverse mapping which is used to form the velocity estimate are approximately linear for the accelerations which would likely be encountered. The end result is that the velocity error is given by:

VEL ERROR = (Final Velocity - Initial Velocity) / 2

Note that the velocity error is a "bias" error, that is, it is proportional only to the velocity difference and not the values of the individual velocities. Note that if the target's acceleration was oscillating between positive and negative values this bias error could cause the velocity estimate of the radar to have an error standard deviation which exceeded the specification.

F-2 SIMULATION RESULTS

A computer simulation of the velocity estimation signal processing portion of the radar was written to validate the conclusions drawn in the above section. The radar return from a target accelerating at a constant rate was modelled by using a linear ramped FM wave. This simulated signal was processed by the DFT and velocity processor and resulted in velocity errors approximately one half the velocity change over the update period as shown in F1.

The results shown in Figure F1 confirm that the effects of acceleration on velocity computation are linearly predictable as described in Section F1 above.

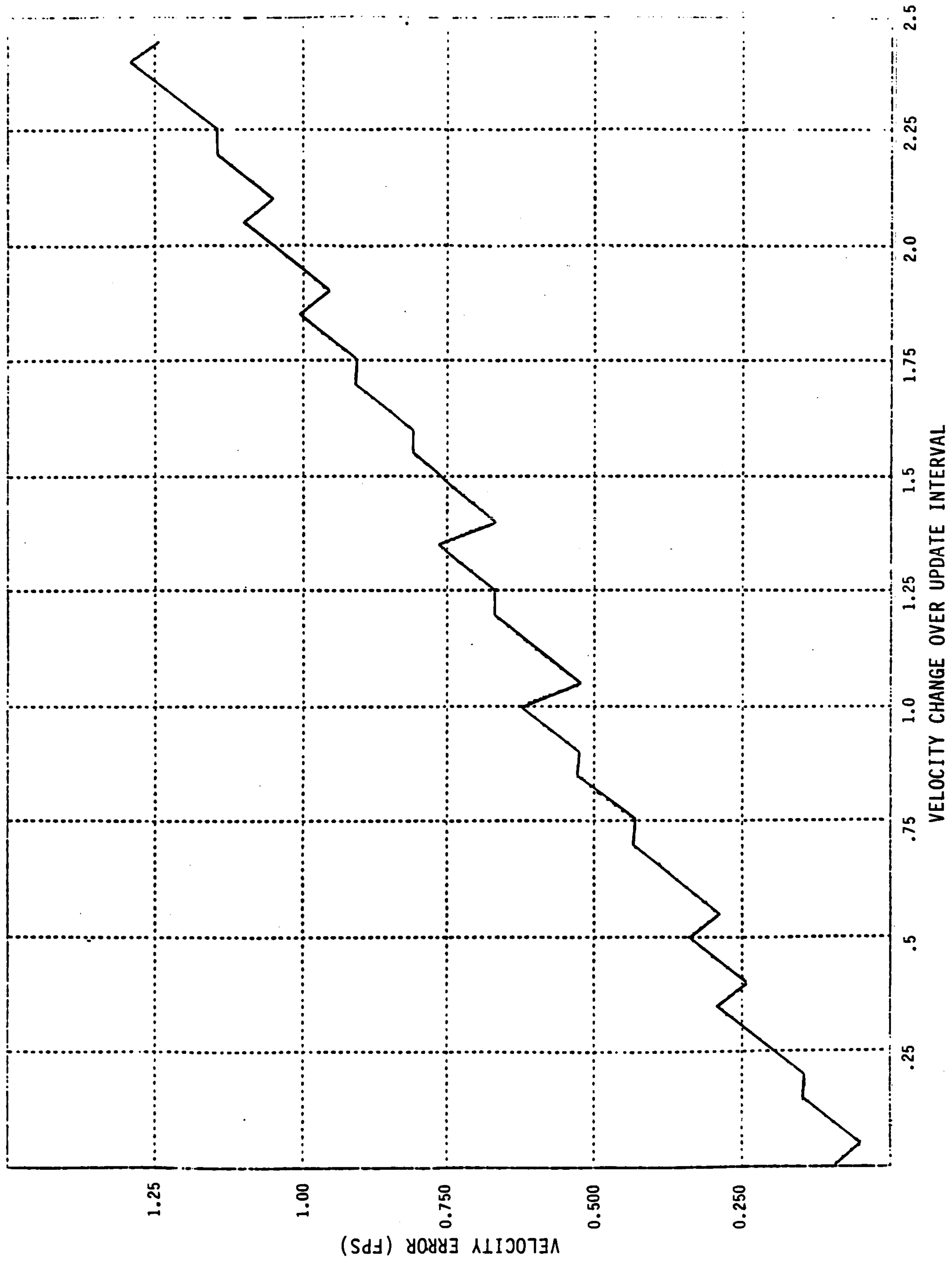


FIGURE F1 VELOCITY CHANGE OVER UPDATE INTERVAL

(THIS PAGE INTENTIONALLY LEFT BLANK)

APPENDIX G

WHITE SANDS MISSILE RANGE FLIGHT TEST DATA SUMMARY

This appendix provides a brief summary of all official flight tests of the space shuttle radar at the White Sands Missile Range (WSMR). The information in the summary was obtained from two sources: (1) the 24 hour reports written by Andy Lindberg of Lockheed Engineering and Management Services Company (LEMSCO), and (2) the reduced flight data provided by NASA JSC and LEMSCO personnel.

This Appendix is structured as follows. Each subsection provides the summaries of all test flights flown on a particular test date. The introduction of each subsection provides the flight conditions for the day, the targets used, and the trajectories flown. The format of each individual flight summary is as follows.

Trajectory: Name of the profile flown.

Range Equipment: WSMR tracking equipment employed.

Flight Profile: The initial and final X, Y and Z coordinates of the target in the Brasscap coordinate system. In this system X is North and -Z is vertical. The arrows (→) indicate the direction of travel.

Duration: The length of the flight in seconds.

Comments: Documentation of large trends, means or standard deviations in the difference data or other anomalies.

In addition, Table G-1 provides a list of all flight tests by trajectory name and the corresponding page numbers within this appendix where a summary of the flight test can be found. Table G-2 contains the statistics for the delta range, range rate, roll, roll rate, pitch, pitch rate and alpha and beta angles for each test.

TABLE G-1 CONTENTS OF THE APPENDIX

TABLE G-1 CONTENTS OF THE APPENDIX

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE G-2 DIFFERENCE MEANS AND STANDARD DEVIATIONS BY TEST RUN AND REFERENCE

PROFILE	REFF		RANGE	RANGE RATE	ROLL	PITCH	PITCH RATE	ROLL RATE	ALPHA	BETA
BAL1	THR	MEAN	14.6296	0.0139	0.4471	-0.5676	0.0384	-0.0259	0.4393	0.6067
		ST.DEV	10.7815	1.3883	0.2778	0.1174	0.1040	0.1018	0.1716	0.2643
BAL2	THR	MEAN	10.5778	-0.0189	0.7363	-0.2540	0.0353	-0.0231	0.5110	1.0106
		ST.DEV	5.8081	3.1086	0.5121	0.2302	0.1281	0.1264	0.1945	0.5457
BAL5	THR	MEAN	24.7658	0.0082	0.4218	-0.6350	-0.0485	0.0688	0.6229	0.5726
		ST.DEV	7.5053	1.2009	0.0794	0.0622	0.0333	0.0406	0.0720	0.0528
BAL6	THR	MEAN	22.8772	0.0109	0.4026	-0.7181	0.0389	-0.0326	0.5875	0.6278
		ST.DEV	9.9824	1.7760	0.2532	0.1367	0.0650	0.0757	0.1313	0.2537
BAL7	THR	MEAN	22.2902	0.0121	0.4132	-0.7480	0.0340	-0.0276	0.5937	0.6559
		ST.DEV	7.6362	2.9110	0.2474	0.1779	0.0442	0.0722	0.1246	0.2770
GEM2	THR	MEAN	5.3789	-0.0667	-0.0867	-0.5254	0.0569	0.0262	0.5573	0.2249
		ST.DEV	35.2816	2.1707	0.3950	0.3022	0.0607	0.0538	0.1089	0.3742
GEM3	THR	MEAN	27.3340	-0.3212	-0.2842	-0.2532	0.0576	0.0066	0.4579	-0.0808
		ST.DEV	43.0562	1.8308	0.4022	0.6565	0.0708	0.0735	0.2943	0.6507
H30SKAE	BEST	MEAN	1.3231	-0.4042	-0.1086	-0.6575	-0.1488	0.2327	0.6538	0.1425
		ST.DEV	10.6636	1.4706	0.2615	0.1284	0.0611	0.1095	0.0531	0.2841
H30SKAE	CINE	MEAN	-0.1066	-0.0377	-0.0700	-0.6599	-0.1651	0.2394	0.6483	0.1748
		ST.DEV	4.5904	0.3192	0.0785	0.0601	0.0793	0.1030	0.0538	0.0735
H30SKAF	BEST	MEAN	10.4932	0.4111	-0.3266	-0.5639	-0.1672	0.0407	0.6460	-0.0816
		ST.DEV	13.0067	1.7252	0.4342	0.2001	0.1604	0.1908	0.0884	0.4711
H30SKAF	CINE	MEAN	3.9267	-0.0061	-0.1210	-0.6367	-0.1564	0.0344	0.6369	0.1365
		ST.DEV	4.5601	0.3309	0.0915	0.0993	0.1691	0.1879	0.1008	0.0891
H30SKAG	BEST	MEAN	-12.1623	-1.1875	0.2374	-0.6505	-0.3421	-0.0009	0.5261	0.4657
		ST.DEV	10.4426	2.4076	0.2659	0.1214	0.0428	0.0788	0.0610	0.2868
H30SKAG	CINE	MEAN	-5.1828	-0.2111	0.0767	-0.5961	-0.3802	0.0533	0.5365	0.2970
		ST.DEV	3.6437	0.7971	0.0842	0.0692	0.0838	0.1045	0.0778	0.0762
H30SKAH	BEST	MEAN	-41.5806	-0.3278	1.4798	-1.0966	-0.1536	0.0641	0.4371	1.7870
		ST.DEV	15.4238	2.2115	0.3372	0.1844	0.1385	0.1437	0.0651	0.3755
H30SKAH	CINE	MEAN	2.8022	0.0741	-0.1118	-0.5628	-0.1630	0.0726	0.5660	0.1160
		ST.DEV	5.7923	1.1462	0.0861	0.0903	0.1282	0.1412	0.0770	0.0903
H30SKAI	BEST	MEAN	-0.3942	-0.2821	-0.0407	-0.7270	0.0184	0.1020	0.6820	0.2522
		ST.DEV	5.9021	1.0652	0.1137	0.0746	0.0554	0.0216	0.0500	0.1266
H30SKAI	CINE	MEAN	5.1442	0.1287	-0.1716	-0.6294	-0.1140	0.0608	0.6455	0.0895
		ST.DEV	5.2854	0.4931	0.0793	0.0740	0.1370	0.0888	0.0849	0.0645
HEL30AF	BEST	MEAN	19.2545	0.0074	-0.1483	-0.5762	-0.0673	0.1044	0.5954	0.0791
		ST.DEV	19.3640	0.7492	0.0731	0.0773	0.0826	0.0913	0.0721	0.0710
HEL30AF	CINE	MEAN	18.9952	0.0438	-0.1298	-0.5791	-0.0677	0.1037	0.5909	0.0969
		ST.DEV	18.8711	0.4893	0.0745	0.0686	0.0847	0.0914	0.0557	0.0807
HEL30AG	BEST	MEAN	9.5942	-0.0190	-0.1686	-0.5188	-0.0687	0.0053	0.5465	0.0491
		ST.DEV	14.9555	0.3741	0.1715	0.0582	0.0936	0.0504	0.0594	0.1716
HEL30AG	CINE	MEAN	9.3744	0.0349	-0.1517	-0.5196	-0.0714	0.0060	0.5417	0.0651
		ST.DEV	20.3794	0.4160	0.0710	0.0469	0.0932	0.0502	0.0523	0.0670
HEL30AI	BEST	MEAN	17.6077	-0.0804	-0.1584	-0.5570	-0.0316	0.0043	0.5746	0.0720
		ST.DEV	17.6219	0.6747	0.2233	0.0847	0.0511	0.0407	0.0728	0.2270
HEL30AI	CINE	MEAN	19.7237	0.0135	-0.2076	-0.5387	-0.0335	0.0051	0.5771	0.0197
		ST.DEV	16.9507	0.3627	0.0660	0.0594	0.0499	0.0409	0.0635	0.0615
HEL30AJ	BEST	MEAN	15.5543	-0.1243	-0.0066	-0.5301	-0.0373	0.0117	0.4913	0.2008
		ST.DEV	24.2236	0.7593	0.7280	0.2008	0.0769	0.0591	0.1127	0.7459
HEL30AJ	CINE	MEAN	19.9391	0.0268	-0.1294	-0.5003	-0.0420	0.0131	0.5124	0.0764
		ST.DEV	14.8156	0.3790	0.0635	0.0503	0.0645	0.0566	0.0447	0.0676
HJ146AC	BEST	MEAN	23.9179	-0.0118	-0.5337	-0.3980	0.0202	0.0600	0.5871	-0.3271
		ST.DEV	40.3428	0.3206	0.1573	0.0835	0.0146	0.0167	0.0860	0.1406
HJ146AC	CINE	MEAN	22.8121	-1.3092	-0.4317	-0.4336	0.0210	0.0634	0.5882	-0.2271
		ST.DEV	41.2481	5.8520	0.0823	0.0654	0.0090	0.0136	0.0776	0.0823
HJ146AD	BEST	MEAN	28.5474	0.0022	-0.0261	-0.6514	0.0137	0.0580	0.6788	0.1057
		ST.DEV	69.6391	0.3608	0.1160	0.0767	0.0131	0.0143	0.0967	0.1129
HJ146AD	CINE	MEAN	28.6869	0.0002	-0.0244	-0.6337	0.0135	0.0589	0.6847	0.1839
		ST.DEV	69.7989	0.6020	0.1007	0.0714	0.0125	0.0141	0.0961	0.0779
HJ146AE	CINE	MEAN	22.4481	-0.0201	-0.0945	-0.6177	-0.0147	-0.0284	0.6608	0.1178
		ST.DEV	56.1602	0.5259	0.2359	0.1029	0.0059	0.0093	0.1490	0.1944
HJ146AE	THR	MEAN	21.8084	-0.0193	-0.0047	-0.6225	-0.0147	-0.0284	0.6626	0.1279
		ST.DEV	56.1907	0.3579	0.2364	0.1031	0.0060	0.0094	0.1501	0.1954

TABLE G-2 DIFFERENCE MEANS AND STANDARD DEVIATIONS BY TEST RUN AND REFERENCE (continued)

PROFILE	REFF		RANGE	RANGE RATE	ROLL	PITCH	PITCH RATE	ROLL RATE	ALPHA	BETA
HL146AE	CINE	MEAN	10.5833	-0.0417	0.0403	-0.6260	-0.0157	-0.0335	0.6116	0.2690
		ST.DEV	25.0777	0.9379	0.1229	0.0732	0.0095	0.0201	0.1021	0.0960
HL146AE	THR	MEAN	8.5280	0.0768	0.0659	-0.6398	-0.0174	-0.0381	0.6151	0.2975
		ST.DEV	25.3803	0.4418	0.1228	0.0726	0.0062	0.0095	0.1014	0.0967
HL246AD	BEST	MEAN	12.4335	0.1168	-0.3988	-0.4227	0.0266	0.0764	0.5664	-0.1897
		ST.DEV	39.6139	0.4775	0.1434	0.0753	0.0176	0.0115	0.0996	0.1351
HL246AD	CINE	MEAN	10.7964	0.0826	-0.3563	-0.4214	0.0235	0.0757	0.5625	-0.1482
		ST.DEV	43.1991	0.6894	0.0667	0.0495	0.0109	0.0086	0.0649	0.0665
HL246AE	BEST	MEAN	0.0276	0.0144	-0.0151	-0.6280	0.0232	0.0630	0.6282	0.2256
		ST.DEV	40.2282	0.5153	0.0934	0.0749	0.0140	0.0135	0.0671	0.1213
HL246AE	CINE	MEAN	-1.0260	0.0252	0.0535	-0.6520	0.0227	0.0632	0.6325	0.2917
		ST.DEV	40.6409	0.7110	0.0725	0.0566	0.0118	0.0135	0.0662	0.0671
HL346AD	BEST	MEAN	16.7585	0.1811	-0.3840	-0.4190	0.0243	0.0620	0.5596	-0.1876
		ST.DEV	27.0040	0.6635	0.0965	0.0589	0.0155	0.0114	0.0794	0.0912
HL346AD	CINE	MEAN	19.3501	0.2210	-0.3856	-0.4251	0.0224	0.0634	0.5782	-0.1820
		ST.DEV	25.1686	0.5975	0.0937	0.0557	0.0142	0.0097	0.0761	0.0913
HL346AE	BEST	MEAN	13.8674	0.0163	0.0367	-0.6586	0.0266	0.0755	0.6453	0.2714
		ST.DEV	32.3583	0.5125	0.1870	0.0964	0.0186	0.0154	0.1202	0.1891
HL346AE	CINE	MEAN	13.7301	0.0456	0.0015	-0.6504	0.0259	0.0760	0.6372	0.2323
		ST.DEV	33.0880	0.6895	0.1622	0.0808	0.0172	0.0152	0.1199	0.1384
HL346AF	CINE	MEAN	13.8561	-0.0021	0.0727	-0.6155	-0.0208	-0.0372	0.5964	0.2881
		ST.DEV	41.8876	0.8238	0.0969	0.0665	0.0046	0.0046	0.0824	0.0794
HL346AF	THR	MEAN	11.8174	0.0135	0.0934	-0.6260	-0.0208	-0.0372	0.5992	0.3105
		ST.DEV	42.1584	0.5622	0.0969	0.0667	0.0046	0.0046	0.0828	0.0805
HL446AC	BEST	MEAN	21.5357	0.1708	-0.4567	-0.4144	0.0243	0.0706	0.6007	-0.2589
		ST.DEV	25.5735	0.4078	0.2439	0.0704	0.0194	0.0105	0.0997	0.2151
HL446AC	CINE	MEAN	23.0432	0.1560	-0.3995	-0.4183	0.0239	0.0690	0.5628	-0.2122
		ST.DEV	25.0711	0.5491	0.0868	0.0530	0.0113	0.0087	0.0727	0.0676
HL446AD	BEST	MEAN	13.3500	0.0060	0.0632	-0.6400	0.0186	0.0608	0.6217	0.2829
		ST.DEV	41.4024	0.5416	0.1030	0.0762	0.0123	0.0180	0.0642	0.1384
HL446AD	CINE	MEAN	12.9660	0.0164	0.0674	-0.6447	0.0195	0.0609	0.6320	0.2852
		ST.DEV	41.4224	0.7305	0.0682	0.0554	0.0104	0.0187	0.0647	0.0619
HL446AE	CINE	MEAN	8.5400	0.0920	0.0640	-0.6191	-0.0180	-0.0374	0.6041	0.2808
		ST.DEV	40.9959	1.2579	0.1960	0.0938	0.0080	0.0101	0.1512	0.1525
HL446AE	THR	MEAN	6.6930	0.0314	0.0920	-0.6335	-0.0183	-0.0381	0.6086	0.3105
		ST.DEV	41.5274	0.5111	0.1972	0.0932	0.0073	0.0076	0.1505	0.1528
HL546AC	THR	MEAN	15.9007	-0.1429	-0.6859	-0.3873	0.0431	0.0752	0.4117	-0.3711
		ST.DEV	45.7245	1.3377	2.8925	0.3770	0.1223	0.0440	0.2369	2.9446
HL546AE	BEST	MEAN	3.9403	0.0649	-0.0037	-0.6350	0.0315	0.0906	0.6338	0.2340
		ST.DEV	34.5596	0.6740	0.0771	0.0653	0.0142	0.0130	0.0652	0.0962
HL546AE	CINE	MEAN	3.9227	0.0716	0.0272	-0.6472	0.0314	0.0907	0.6395	0.2637
		ST.DEV	34.7227	0.7537	0.0724	0.0561	0.0123	0.0126	0.0654	0.0596
HL546AF	THR	MEAN	-0.3150	0.0093	1.4794	-1.1692	0.0228	0.0751	0.6309	1.7302
		ST.DEV	51.6373	0.5354	0.1294	0.0639	0.0116	0.0107	0.1492	0.1728
HL546AG	CINE	MEAN	7.4577	-0.1357	0.0362	-0.6482	-0.0203	-0.0437	0.6330	0.2758
		ST.DEV	27.5850	1.8137	0.1320	0.0740	0.0066	0.0100	0.1055	0.1065
HL546AG	THR	MEAN	4.6124	0.0637	0.0590	-0.6621	-0.0203	-0.0454	0.6310	0.3164
		ST.DEV	25.2104	0.4621	0.1103	0.0685	0.0053	0.0058	0.0950	0.0874
SAT1	THR	MEAN	5.0552	0.0396	0.1044	-0.1525	0.0060	0.0000	-0.0172	0.1605
		ST.DEV	17.6844	2.3341	0.2066	0.2105	0.0764	0.0302	0.2681	0.2437
SAT2	BEST	MEAN	7.0837	-0.1241	0.1043	-0.3943	0.0249	0.0008	0.3889	0.3726
		ST.DEV	33.4679	3.6169	0.5385	0.2181	0.1176	0.0495	0.2382	0.5445
SAT2	CINE	MEAN	2.2107	-0.0106	0.1419	-0.4331	0.0232	0.0014	0.3963	0.3512
		ST.DEV	2.4174	1.4091	0.1407	0.0689	0.1044	0.0402	0.1406	0.1274
SAT3	BEST	MEAN	30.2168	0.2132	-7.0367	-1.9407	-0.0074	0.0851	2.4407	-1.9982
		ST.DEV	51.2225	6.7040	66.4737	2.2230	0.1391	0.4833	2.9931	4.3826
SAT3	CINE	MEAN	2.4247	0.0184	1.3421	-0.4691	-0.0082	0.1000	0.8372	0.2201
		ST.DEV	2.5430	1.8519	1.7889	0.2732	0.0935	0.1412	0.3284	0.2144
SAT4	BEST	MEAN	20.3398	0.0092	-0.1663	-0.7369	0.0313	-0.0193	0.7405	0.1387
		ST.DEV	13.8462	0.7599	0.1736	0.0687	0.0534	0.0382	0.0999	0.1507
SAT4	CINE	MEAN	11.5751	0.0828	0.0813	-0.7683	0.0296	-0.0198	0.7038	0.3832
		ST.DEV	14.4241	0.6408	0.1171	0.0661	0.0515	0.0371	0.0933	0.0924

RANGE MEASURED IN FEET
RANGE RATE MEASURED IN FEET/SECOND
ROLL ANGLE MEASURED IN DEGREES
PITCH ANGLE MEASURED IN DEGREES
ROLL RATE MEASURED IN DEGREES/SECOND
PITCH RATE MEASURED IN DEGREES/SECOND
ALPHA MEASURED IN DEGREES
BETA MEASURED IN DEGREES

G.1 10/1/85 TEST SUMMARIES

Table G-2 summarizes the flight conditions, targets used, and trajectories flown on 10/1/85.

TABLE G-2: 10/1/85 FLIGHT CONDITION SUMMARY

Flight Conditions: Heavy clouds in spots, hindering cinetheodolites. Ceiling was 2500 ft. and cover 2500 ft. thick.

Target: Helicopter

Trajectories flown:

HL546AC
HL246AD
HL446AC
HL346AD
HJ146AC

G.1.1 Individual Flight Test Summaries

Trajectory: HL546AC

Range Equipment: 3 radars (R-350, R-393, R-394), no cines

Flight Profile (Brasscap)	X: 47000 → 27000 ft
	Y: 1500 → 30000 ft
	-Z:oscillates,3500 → 11000 ft

Duration: 450 s

Comments: Delta roll mimics the Z profile, delta pitch is the inverse of the Z profile, large -Z changes, downtrend in delta range, delta roll rate oscillates.

Trajectory: HL246AD

Range Equipment: 3 radars (R-350, R-393, R-394),
5 minutes of cine data.

Flight Profile (Brasscap)	X: 42000 → 30000 ft
	Y: 8000 → 36000 ft
	-Z:oscillates, 5890 →6030 ft

Duration: 500 s

Comments: V-shaped trend in delta roll, down to -.65 deg, oscillating delta pitch, bias of -.4 deg.

Trajectory: HL446AC

Range Equipment: 3 radars (R-350, R-393, R-394),
7 minutes of cine data.

Flight Profile (Brasscap)	X: 48000 → 35000 ft
	Y: 1250 → 31000 ft
	-Z: stable around 6100 ft, drops to 5550 at t=300 s

Duration: 425 s

Comments: Delta roll seems to follow the Z profile, as in HL546AC
delta range mean=22.46, std. dev.=25.36
delta pitch skews up.

Trajectory: HL146AC not available

Trajectory: HL346AD

Range Equipment: 3 radars (R-350, R-393, R-394),
small amounts of cine data.

Flight Profile (Brasscap)	X: 45500 → 32500 ft
	Y: 14000 → 36000 ft
	-Z:oscillates, 5820 →5900 ft

Duration: 400 s

Comments: Delta roll mean= -.38 deg, oscillates, delta pitch mean= -.43 deg, oscillates.

Trajectory: HJ146AC

Range Equipment: 3 radars (R-350, R-393, R-394),
small amounts of cine data.

Flight Profile (Brasscap) X: 64000 → 30000 ft
Y: 0 → 32500 ft
-Z:oscillates, 5500 →6200 ft

Duration: 600 s

Comments: Delta roll appears to mimic the Z profile, delta pitch has a similar pattern, delta range mean= 23.8, std.dev.=40.39

G.2 10/3/85 TEST SUMMARIES

Table G-3 summarizes the flight conditions, targets used, and trajectories flown on 10/3/85

TABLE G-3: 10/3/85 FLIGHT CONDITION SUMMARY

Flight Conditions: Good weather, winds tended to increase the target's velocity and slightly altered the flight path.

Target: Helicopter

Trajectories Flown:

HEL30AE
HEL30AF
H30SKAE
H30SKAF
HEL30AG
HEL30AH
H30SKAG
H30SKAH
H30SKAI
HEL30AI
HEL30AJ

G.2.1 Individual Flight Test Summaries

Trajectory: HEL30AE   not available

Trajectory: HEL30AF

Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap)	X: 9500 → 3250 ft
	Y: 5200 → 5400 → 3400 ft
	-Z: 6000 → 5100 ft

Duration: 160 s

Comments: Large trends in delta roll and delta pitch, on the order of .2 deg, discontinuity of 30 ft in delta range at t=50 s.

Trajectory: H30SKAE

Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap)	X: 2800 → 1000 ft
	Y: oscillates, 1580 → 1340 ft
	-Z: 1600 → 2000

Duration: 100 s

Comments: Delta roll and delta pitch mimic the Y profile, sinusoidal delta range has 30 ft deflections, delta range rate deflections of 5 ft, trends in delta pitch and roll rates.

Trajectory: H30SKAF

Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap)	X: 3200 → 1400
	Y: 2100 → 800
	-Z: 1550 → 1450

Duration: 130 s

Comments: Large trends in delta range (40-50 ft), delta range rate std. dev.= 1.72 large deflections in delta roll and pitch, up to 1.6 deg.

Trajectory: HEL30AG

Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap)	X: 10250 → 4600 ft
	Y: 5000 → 6200 → 1500 ft
	-Z: 6400 → 5200

Duration: 260 s

Comments: Large trend in delta roll (.5 deg), delta pitch discontinuity at t=150 s

---

Trajectory: HEL30AH   not available

Trajectory: H30SKAG
Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap) X: 1590 → 1360 ft
Y: 860 → 620 ft
-Z: 1636 → 1652 → 1644 ft

Duration: 25 s

Comments: Large trend in delta roll and pitch (.7 deg), trends in delta range, 35 ft deflections.

Trajectory: H30SKAH
Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap) X: 3100 → 1400 ft
Y: 1900 → 900 ft
-Z: 1700 → 1800 → 1580 ft

Duration: 110 s

Comments: Large downtrend in delta range (50 ft), delta roll has 1.4 deg deflections, delta pitch has .7 deg deflections, large delta roll and pitch rates (.6 deg/s).

Trajectory: H30SKAI
Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap) X: 2700 → 2300 ft
Y: 1900 → 1890 → 1930 ft
-Z: 1620 → 1660 → 1550 ft

Duration: 40 s

Comments: Trends in delta roll, pitch and range.

Trajectory: HEL30AI
Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap)   X: 10000 → 4000 ft
Y: 6000 → 2100 ft
-Z: 6500 → 3300 ft

Duration: 375 s

Comments: Large delta roll skew (.8 deg), delta pitch skew of .3 deg, discontinuity of 40 ft at t=100 s.

Trajectory: HEL30AJ
Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap)   X: 10500 → 4000 ft
Y: 6000 → 2000 ft
-Z: 6200 → 4000 ft

Duration: 400 s

Comments: Large trend in delta range (90 ft deflections), trends in delta roll (2.5 deg), trends in delta pitch (.8 deg), oscillating delta roll rate.

G.3 10/5/85 TEST SUMMARIES

Table G-4 summarizes the flight conditions, targets used, and trajectories flown on 10/5/85.

TABLE G-4: 10/5/85 FLIGHT CONDITION SUMMARY

Flight Conditions: Good weather, slight winds.

Target: Helicopter

Trajectories Flown:

HL546AE
HL346AE
HL446AD
HL146AD
HJ146AD
HL546AF
HL246AE

G.3.1 Individual Flight Test Summaries

Trajectory: HL546AE

Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap)	X: 47000 → 24000 ft
	Y: 1000 → 32500 ft
	-Z: oscillates, 5900 →6300 ft

Duration: 475 s

Comments: Small trends in delta roll and pitch (.1 deg).

Trajectory: HL346AE

Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap)	X: 47000 → 32000 ft
	Y: 0 → 37000 ft
	-Z: oscillates, 5940 →6170 ft

Duration: 550 s

Comments: Small spikes (.06 deg) in delta roll and pitch.

Trajectory: HL446AD

Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap) X: 49000 → 31000 ft
Y: 0 → 35000 ft
-Z: oscillates, 5850 →6250 ft

Duration: 650 s

Comments: Oscillations in delta roll (.25 deg), and delta pitch (.2 deg).

Trajectory: HL146AD not available

Trajectory: HJ146AD

Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap) X: 64000 → 36000 ft
Y: 0 → 29000 ft
-Z: oscillates, 5775 →6050 ft

Duration: 575 s

Comments: Large delta range std. dev.=69 ft, oscillations in delta roll and pitch (.4 deg).

Trajectory: HL546AF
Range Equipment: 3 radars (R-350, R-393, R-394), no cines

Flight Profile (Brasscap) X: 46000 → 24000 ft
Y: 0 → 32500 ft
-Z: 6850 → 7200 → 6900 ft
Duration: 600 s

Comments: Delta range std. dev.= 51 ft, large bias in delta roll (1.48 deg), and delta pitch (-1.17 deg), V-shaped trends in data.

Trajectory: HL246AE
Range Equipment: 3 radars (R-350, R-393, R-394)

Flight Profile (Brasscap) X: 42000 → 29000 ft
Y: 0 → 36000 ft
-Z: oscillates, 5900 →6175 ft
Duration: 700 s

Comments: Slight trend in delta roll , mean=.02, std. dev.=.09
delta pitch mean=-.65, std. dev.=.06

G.4 10/16/85 TEST SUMMARIES

Table G-5 summarizes the flight conditions, targets used, and trajectories flown on 10/16/85.

TABLE G-5: 10/16/85 FLIGHT CONDITION SUMMARY

Flight Conditions: Drizzling rain, low ceiling (3000 ft).

Target: Gemspheres (free floating).

Trajectories Flown:

GEM1

GEM2

GEM3

G.4.1 Individual Flight Test Summaries

Trajectory: GEM1 not available

Trajectory: GEM2
Range Equipment: 3 radars (R-350, R-393, R-394)

Flight Profile (Brasscap) X: 2000 → 28000 ft
Y: -1500 → 2000 ft
-Z: 2000 → 11000 ft

Duration: 500 s

Comments: Track lost at first, but picked up at range of 4000 ft.
downtrend in delta range, flat delta range rate,
but large std. dev.=2.17,
large trends in delta roll and pitch (1.6 deg)
trends in delta roll and pitch rate.

Trajectory: GEM3
Range Equipment: 3 radars (R-350, R-393, R-394)

Flight Profile (Brasscap) X: 1000 → 24000 ft
Y: -500 → -3500 → 500 ft
-Z: 1500 → 10000 ft

Duration: 500 s

Comments: Initially lost track but required,
large downtrend in delta range (175 ft),
delta range rate std. dev.=1.83,
large trends in delta roll and pitch (2 deg),
also in delta roll and pitch rates.

G.5 10/19/85 TEST SUMMARIES

Table G-6 summarizes the flight conditions, targets used, and trajectories flown on 10/19/85.

TABLE G-6: 10/19/85 FLIGHT CONDITION SUMMARY

Flight Conditions: Good conditions

Target: 2m Gemsphere suspended below 2, 10 ft balloons, tethered flight.

Trajectories Flown:

SAT1
SAT2
SAT3
SAT4
SAT6
SAT8

G.5.1 Individual Flight Test Summaries

Trajectory: SAT1

Range Equipment: 3 radars (R-350, R-393, R-394), 10 cines

Flight Profile (Brasscap)	X: 450 → 300 → 700 ft
	Y:-1340 → -1440 → -1220 ft
	-Z: 2180 → 2050 → 2120 ft

Duration:600 s

Comments: Large trend in delta roll (.6 deg), and delta pitch (.8 deg) trend in delta range (80 ft) delta range rate std. dev.=2.33 trajectory had large roll angles up to -74 deg.

Trajectory: SAT2

Range Equipment: 3 radar (R-350, R-393, R-394), 10 cines

Flight Profile (Brasscap)	X: 725 → 450 → 1075 ft
	Y: -750 → -100 ft
	-Z: oscillates 2290 → 2390 ft

Duration: 600 s

Comments: Large trends in delta roll (1.8 deg), and delta pitch (.6 deg), roll angles up to -62 deg, large trends in delta range (80 ft).

Trajectory: SAT3

Range Equipment: 3 radar (R-350, R-393, R-394), 10 cines

Flight Profile (Brasscap) X: 1000 → -800 ft
Y: 0 → 700 ft
-Z: 2200 → 200 ft

Duration: 600 s

Comments: The balloon tether broke on this flight.
trends in delta range (160 ft),
delta range rate std. dev.=6.78,
anomalies in delta roll and pitch.

Trajectory: SAT4

Range Equipment: 3 radars (R-350, R-393, R-394), 10 cines

Flight Profile (Brasscap) X: 6400 → 8600 ft
Y: 5600 → 7800 → 6300 ft
-Z: 6750 → 5100 ft

Duration: 600 s

Comments: Sporadic delta range (60 ft deflections), trends in delta roll and pitch (.4 deg), deltaroll and pitch rates have damped oscillations.

Trajectory: SAT6 not available

Trajectory: SAT8 not available

G.6 11/4/85 TEST SUMMARIES

Table G-7 summarizes the flight conditions, targets used, and trajectories flown on 11/4/85.

TABLE G-7: 11/4/85 FLIGHT CONDITION SUMMARY

Flight Conditions: Higher altitude winds caused the target balloons to drift back over the Pearl site.

Target: Gemspheres (free floating) and helicopters.

Note: The antenna servo gain had been increased on 11/2.

Trajectories Flown:

BAL1
BAL2
BAL5
BAL6
BAL7
HL546AG
HL246AF
HL446AE
HL146AE
HL346AF
HJ146AE

G.6.1 Individual Flight Test Summaries

Trajectory: BAL1

Range Equipment: 1 radar (R-394), no cines

Flight Profile (Brasscap) X: 500 → 3300 → 900 ft

Y: 300 → 1900 ft

-Z: 500 → 10000 ft

Duration: 600 s

Comments: Large bias and initial skew on delta roll and pitch (2 deg), discontinuity in delta range at t=250 s, delta range rate std. dev.=1.39.

Trajectory: BAL2

Range Equipment: 1 radar (R-394), no cines

Flight Profile (Brasscap) X: 750 → 3500 ft

Y: 200 → 1000 ft

-Z: 300 → 4100 ft

Duration: 300 s

Comments: Large trends in delta roll and pitch (1 deg), oscillations in delta range (7 ft), delta range rate std. dev.=3.08, large delta roll and pitch rate deflections at t=75 s (.6 deg).

Trajectory: BAL5
Range Equipment: 1 radar (R-394), no cines

Flight Profile (Brasscap) X: 3500 → 1600 ft
Y: 1000 → 2000 ft
-Z: 7100 → 10000 ft

Duration: 170 s

Comments: Large bias in delta roll (.42 deg) and delta pitch (-.64 deg), trends in these of .15 deg.
delta range rate std. dev.=1.2 deg/s.

Trajectory: BAL6
Range Equipment: 1 radar (R-394), no cines

Flight Profile (Brasscap) X: 750 → 3750 → 1750 ft
Y: 300 → 2400 ft
-Z: 500 → 10000 ft

Duration: 600 s

Comments: Large initial skew in delta pitch and roll
(1.6 deg), and delta pitch and roll rate
(.55 deg/s)
delta range rate std. dev.=1.78.

Trajectory :BAL7

Range Equipment: 1 radar (R-394), no cines

Flight Profile (Brasscap)   X: 900 → 3800 → 2250 ft
Y: 300 → 2400 ft
-Z: 500 → 10000 ft

Duration: 600 s

Comments: Large initial skew in delta roll and pitch (1.6 deg), delta range rate std. dev.=2.91, also an initial skew in delta roll and pitch rates

---

Trajectory: HL546AG

Range Equipment: 1 radar (R-394), 5 cines

Flight Profile (Brasscap)   X:45000 → 25000 ft
Y: 8000 → 31000 ft
-Z: oscillates 6040 → 6300 ft

Duration: 500 s

Comments: Delta roll within .1 deg std. dev., delta pitch still has mean of -.64 deg.

Trajectory: HL246AF   not available

Trajectory: HL446AE
Range Equipment: 1 radar (R-394), 5 cines

Flight Profile (Brasscap)   X: 48000 → 30000 ft
                               Y: 0 → 35000 ft
                               -Z: oscillates 6000 → 6400 ft
Duration: 550 s

Comments: Large down spike (2 deg) in delta roll and pitch at t= 225-275 s. Due to glitch in KU angles.

Trajectory: HL146AE
Range Equipment: 1 radar (R-394), 5 cines

Flight Profile (Brasscap)   X: 46000 → 25000 ft
                               Y: 0 → 35000 ft
                               -Z: oscillates 6120 → 6340 ft
Duration: 600 s

Comments: Delta roll is fairly flat, mean=.07, std. dev.=.12, delta pitch is still biased mean=-.64 deg, std. dev.=.07.

Trajectory: HL346AF
Range Equipment: 1 radar (R-394), 5 cines

Flight Profile (Brasscap)   X: 47000 → 31000 ft
                               Y: 0 → 37000 ft
                               -Z: oscillates 6125 →6255 ft
Duration: 550 s

Comments: Delta range rate std. dev.=.55, delta roll is fair (mean=.09, std. dev.=.1) delta pitch mean=-.63 std. dev.=.07

Trajectory: HJ146AE

Range Equipment: 1 radar (R-394), 5 cines

Flight Profile (Brasscap) X: 65000 → 35000 ft
Y: 0 → 30000 ft
-Z: oscillates 6080 → 6300 ft

Duration: 600 s

Comments: Delta roll mean=-.08, std. dev.=.24, delta pitch mean=-.62, std. dev.=.1

(THIS PAGE INTENTIONALLY LEFT BLANK)

APPENDIX H

ADDENDUM TO SORTE ANGLE RATE DATA ANALYSIS

The purpose of this appendix is to augment the angle rate data analysis presented in Section 3.6. In particular, in the one case (H30SKAF) that was analyzed in detail, it was found that the principal error source was angle acceleration. Furthermore, the bias-effect on the angle rate was exactly predictable from a knowledge of the acceleration and the natural frequency of the loop, f_n. As noted there, this was the first corroboration that the angle rate loop is properly represented by the model in Figure 3.6-4, and that the f_n value for the widest bandwidth case has been properly implemented in the hardware. Since there are two other bandwidth values for the angle rate tracker, the purpose of this appendix is to verify that the other two f_n values are implemented properly through the use of the angle acceleration data.

Table H-1 summarizes the values of f_n for the different range intervals in the passive tracking mode. As noted earlier, the H30SKAF data was used to analyze the wide bandwidth case. Here, the first 150 seconds of the HEL30AG profile is used to analyze the medium bandwidth case, and HL446AC profile is used to analyze the narrow bandwidth case.

TABLE H-1 VALUES OF THE NATURAL FREQUENCY OF THE ANGLE RATE TRACKER FOR DIFFERENT RANGE INTERVALS

RANGE INTERVAL	f_n, Hz
< 11,510	0.120
11,520 to 23,020	0.070
> 23,030	0.027

Medium Bandwidth Case. Figures H-1 and H-2 compare the angle rate difference data and the corresponding angle acceleration for pitch and roll rate, respectively. As was done in Sectioon 3.6, a time interval was selected in each data set and the angle rate bias formula of equation 3-12 was applied to determine if the relation was satisfied. Table H-2 summarizes the results

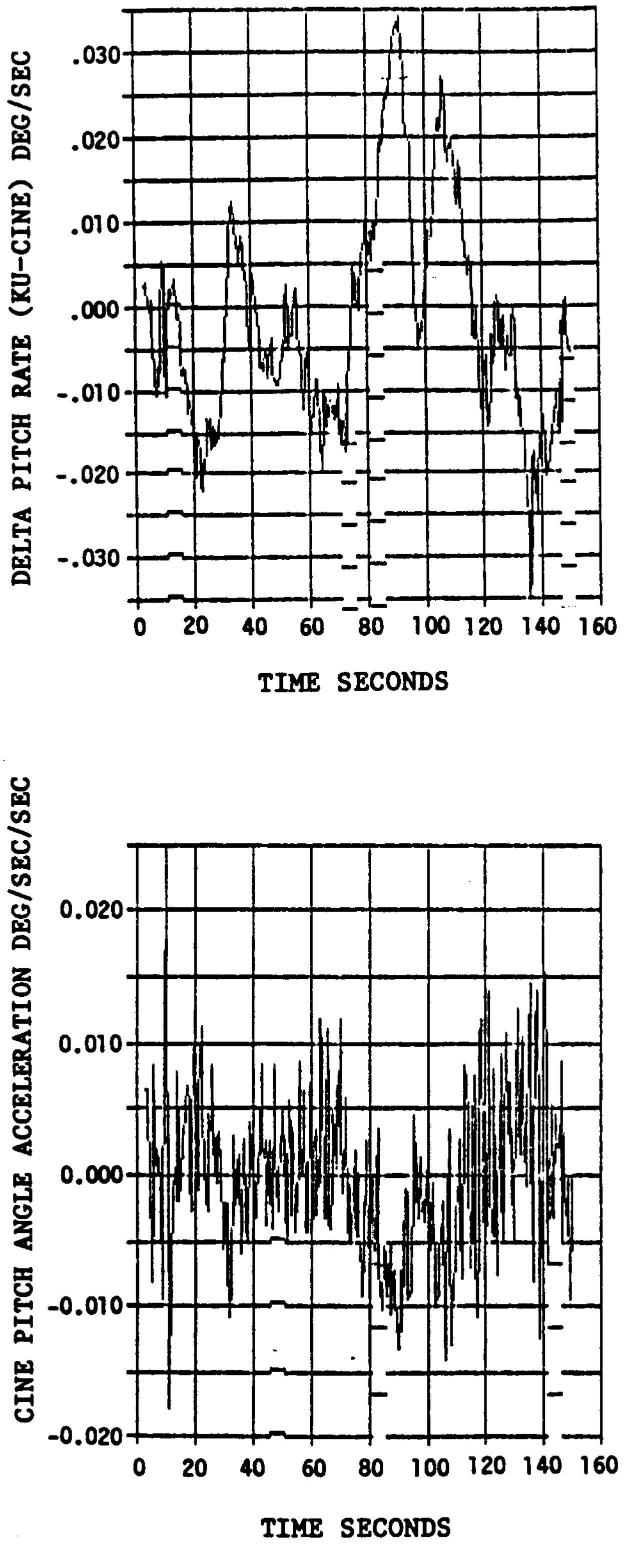
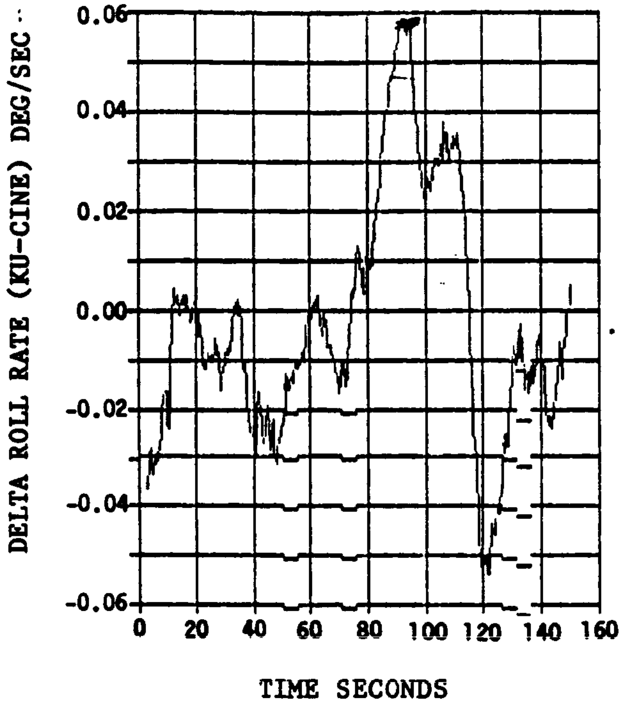


FIGURE H-1 A COMPARISON OF THE CINE PITCH ANGLE ACCELERATION PROFILE AND THE CINE PITCH RATE DIFFERENCE PROFILE FOR THE HEL30AG FLIGHT



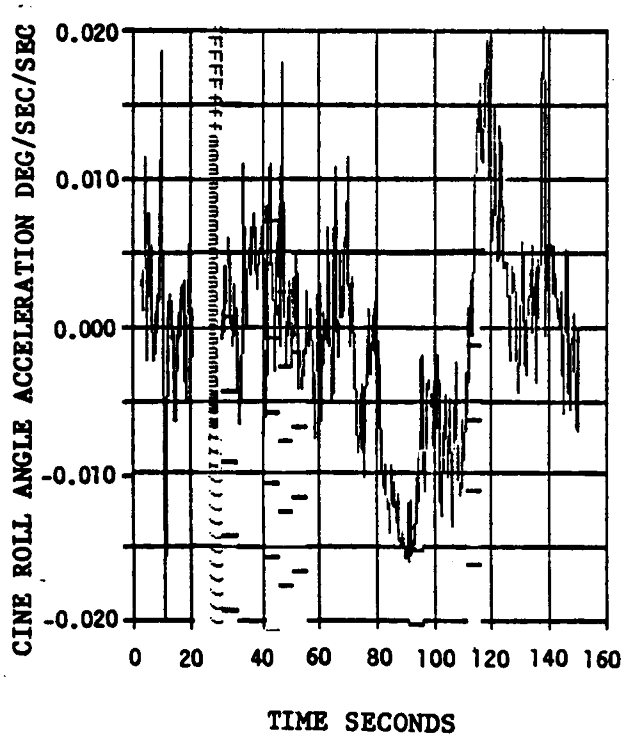


FIGURE H-2 A COMPARISON OF THE CINE ROLL ANGLE ACCELERATION PROFILE AND THE CINE ROLL RATE DIFFERENCE PROFILE FOR THE HEL30AG FLIGHT of these selections and computations. It should be pointed out that the average angle acceleration and the measured angle rate bias are "eyeball" estimates taken from Figure H-1 and H-2. The data of Table H-2 shows a very close match between computed and measured angle rate bias. It can be concluded that the value of f_n (0.07) for this range interval has been correctly implemeuted in the hardware.

TABLE H-2 EVALUATION OF ANGLE ACCELERATION BIAS EFFECTS IN THE MEDIUM BANDWIDTH CASE

	TIME INTERVAL, SEC	AVERAGE ANGLE ACCELERATION	COMPUTED ANGLE RATE BIAS	MEASURED ANGLE RATE BIAS
Roll Rate	40 to 50	0.0060 deg/sec^2	-0.027 deg/sec	-0.023 deg/sec
	85 to 95	-0.0125	0.056	0.050
Pitch Rate	20 to 30	0.0030	-0.0136	-0.015
	82 to 92	-0.0070	0.0318	0.028

Narrow Bandwidth Case. Figures H-3 and H-4 give the angle rate difference and the corresponding angle acceleration for pitch and roll rate, respectively. Table H-3 provides the results of how the angle acceleration bias affects computations. In this case, it is very hard to identify the angle rate bias because it appears to be buried in the thermal noise and other effects. There were some time intervals where the acceleration effects were prominent. In those cases, there was good agreement between the predicted bias and the measured bias.

TABLE H-3 EVALUATION OF ANGLE ACCELERATION BIAS EFFECTS IN THE NARROW BANDWIDTH CASE

	TIME INTERVAL, SEC	AVERAGE ANGLE ACCELERATION	COMPUTED ANGLE RATE BIAS	MEASURED ANGLE RATE BIAS
Roll Rate	150 to 170	0.001 deg/sec^2	-0.0118 deg/sec	-0.009 deg/sec
	25 to 50	-0.0005	0.0059	0.005
Pitch Rate	25 to 35	-0.0008	0.0094	0.008

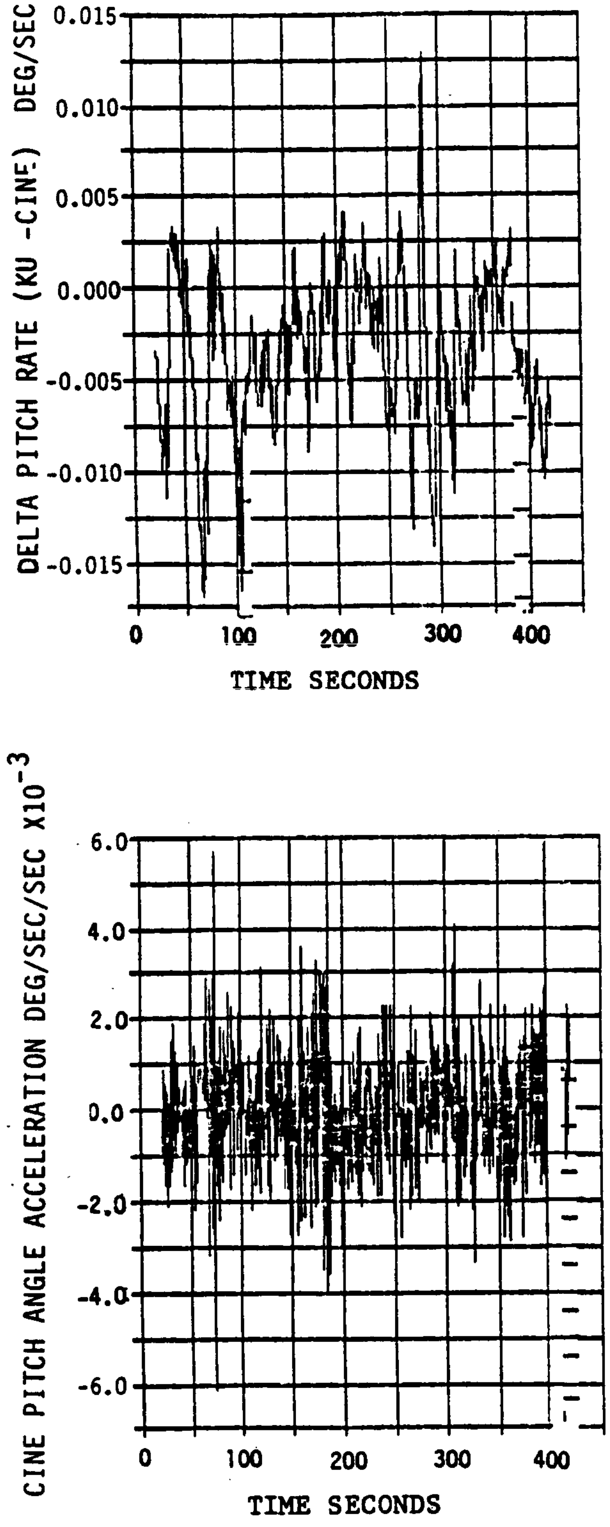
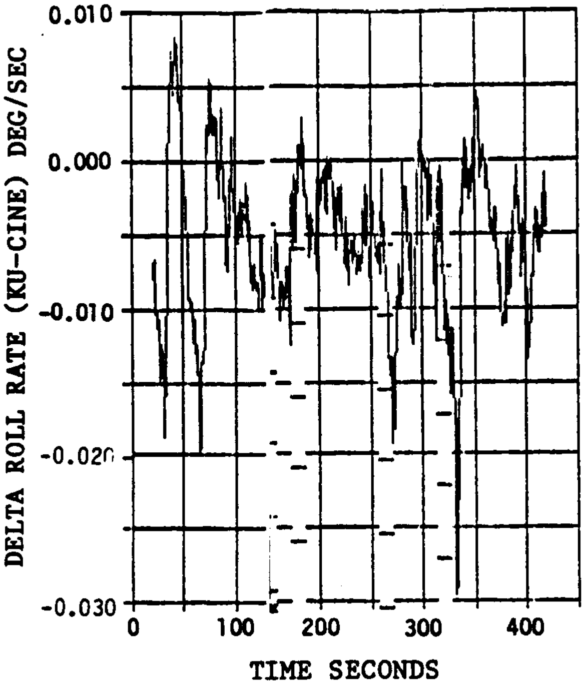


FIGURE H-3 A COMPARISON OF THE CINE PITCH ANGLE ACCELERATION PROFILE AND THE CINE PITCH RATE DIFFERENCE PROFILE FOR THE HL446AC FLIGHT



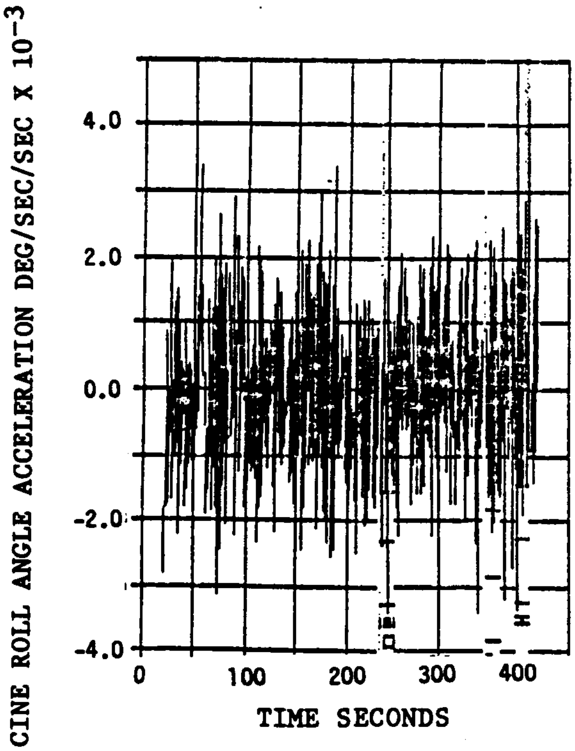


FIGURE H-4 A COMPARSION OF THE CINE ROLL RATE ACCELERATION PROFILE AND THE CINE ROLL RATE DIFFERENCE PROFILE FOR THE HL446AC FLIGHT