REPORT

(NASA-CR-160883) KU-BAND RENDEZVOUS RADAR PERFORMANCE COMPUTER SIMULATION MODEL Final Report (Hughes Aircraft Co.) 326 p HC A15/MF A01 CSCL 17I G3/32

N81-11277

Unclas
29141

KU-BAND RENDEZVOUS RADAR PERFORMANCE COMPUTER SIMULATION MODEL

FINAL REPORT

Radar Systems Group
Hughes Aircraft Company
2000 E. Imperial Highway
El Segundo, California 90245

July 1980

Contract NAS9-15840

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Johnson Space Flight Center
Houston, Texas 77058
J. W. Griffin, Technical Officer

KU-BAND RENDEZVOUS RADAR PERFORMANCE

COMPUTER SIMULATION MODEL

FINAL REPORT

CONTRACT NAS 9-15840

Prepared for: National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
Houston, Texas 77058
J. W. Griffin, Technical Monitor

Prepared by: Radar Subsystem Staff
Engineering Division
Radar Systems Group
Hughes Aircraft Company
El Segundo, CA 90245

July 1980

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (continued)

LIST OF ILLUSTRATIONS (continued)

LIST OF ILLUSTRATIONS (continued)

LIST OF ILLUSTRATIONS (continued)

LIST OF ILLUSTRATIONS (continued)

LIST OF TABLES

LIST OF TABLES (continued)

1. INTRODUCTION AND OVERVIEW

1.1 INTRODUCTION

The objective of this progıam is the preparation of a real time computer simulation model of the Ku-Band Rendezvous Radar to be integrated into the Shuttle Mission Simulator (SMS), the Shuttle Engineering Simulator (SES), and the Shuttle Avionics Integration Laboratory (SAIL) simulator. Primary requirements of the simulation model are to provide crew training and to provide mission planners with representative predictions of the Ku-Band Radar tracking capability against selected candidate targets. The crew training requirement imposes the following design objectives with respect to the track and search modes:

(1) to provide a real time simulation,

(2) to provide accurate timing of discrete events appearing on the radar cockpit display,

(3) to provide accurate operation of cockpit display meters,

(4) to provide accurate responses to all cockpit radar controls.

In addition, the design objectives generated by a desire for accurate prediction of track mode operation against candidate targets are as follows:

(5) to provide representative scattering models for all targets of interest,

(6) to provide accurate processing of the target return signal,

(7) to provide accurate models of all tracking loops.

Based upon our present knowledge of the capabilities of the three simulators, design goal (1) will conflict with design goals (5) through (7). Therefore, some sacrifices were made in target model accuracy and track signal processing accuracy to maintain a real time simulation. The sacrifices in track model accuracy and target scattering model accuracy and the performance limits they impose are discussed in detail in the sequel.

The development of the Ku-Band Rendezvous Radar performance computer model that meets the requirements stated above has been divided into three tasks: (1) development of the radar tracking performance model, (2) development of the radar search and acquisition performance model, and (3) development of a target modeling method. This report documents the results obtained in these three areas. It includes:

- a detailed description of the parent simulation/radar simulation interface requirements,
- a detailed description of the method selected to model target scattering properties, including an application of this method to the SPAS spacecraft,
- a detailed description of the radar search and acquisition mode performance model.
- a detailed description and supporting analysis of the radar track mode signal processor model,
- a detailed description and supporting analysis of the angle, angle rate, range, and range rate tracking loops.

1.2 OVERVIEW OF RADAR PERFORMANCE COMPUTER MODEL

In all of the material that follows the reader's background knowledge of the Ku-Band Rendezvous Radar system is assumed to be on or above the level given in [1] or [2].

1.2.1 Target Scattering Model Summary

Since virtually all target effects work (References 3-9) deals with point scatterer models, our approach is to represent the target as a collection of point scatterers. More specifically, this approach to modeling consists of:

- identifying strong scattering centers ("bright spots") and modeling them as point scatterers with associated cross section functions to express the angular variation,
- modeling intricate or rough-surfaced areas of the target as a random scatterer field, in turn, modeled by point scatterers with random amplitudes and specified angular variation functions.

It is remarked that these angular variation functions account for the shadowing effects due to a point scatterer's position relative to the other scatterers. Also these angular variation functions do not include the phasing terms given in the cross section literature. These factors are reflected in the spatial separation of the model's point scatterers. An example of this modeling method applied to the SPAS spacecraft is described in Section 4.0.

1.2.2 General Computer Model Structure

Figure 1-1 illustrates the general configuration of the computer model. It consists of three major parts: the executive program, the search and acquisition program, and the track program. The functions of the executive program are to initialize the system and target data when the program is first entered, to determine the system operating mode each update period and pass control to the appropriate subprogram, and to initialize the system appropriately when changes in the system controls have occurred. Search and track program details are summarized below.

1.2.3 Radar Search and Acquisition Performance Model Summary

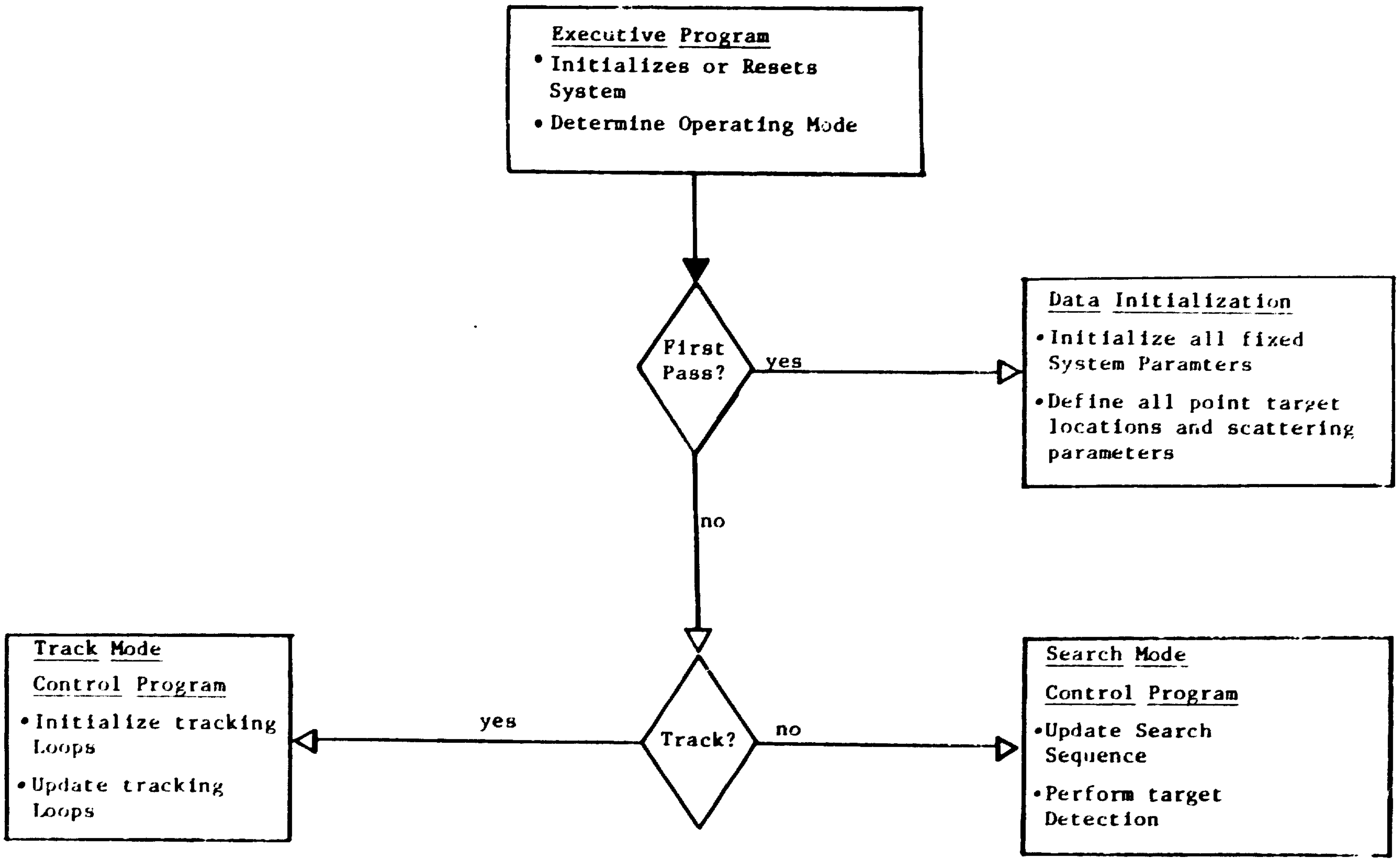
An outline of the search and acquisition performance computer model is given in Figure 1-2. Main elements of this model are:

- antenna gimbal pointing loop model,
- scan model,
- detection model.

Antenna Gimbal Pointing Loop. The antenna α and β gimbal pointing loops were both represented by the second order model shown in Figure 1-3. This model responds to (1) angle designates input from the General Purpose Computer (GPC) and (2) slew rate commands input by the crew from the cockpit. In the present configuration, the loop constants are chosen to give a loop damping factor ρ of 0.7 and a crossover frequency ω_c of 1 hz.

Scan Model. This algorithm models radar system performance when a spiral antenna scan is in progress. The model is invoked by a search initiate command from either the GPC or the crew and operates as follows. It tracks the antenna position,

Figure 1-1 SIMPLIFIED BLOCK DIAGRAM OF THE COMPUTER MODEL


Executive Program
• Initializes or Resets System
• Determine Operating Mode
First Pass?
yes
Data Initialization
• Initialize all fixed System Paramters
• Define all point target locations and scattering parameters
no
Track?
yes
Track Mode Control Program
• Initialize tracking Loops
• Update tracking Loops
no
Search Mode Control Program
• Update Search Sequence
• Perform target Detection

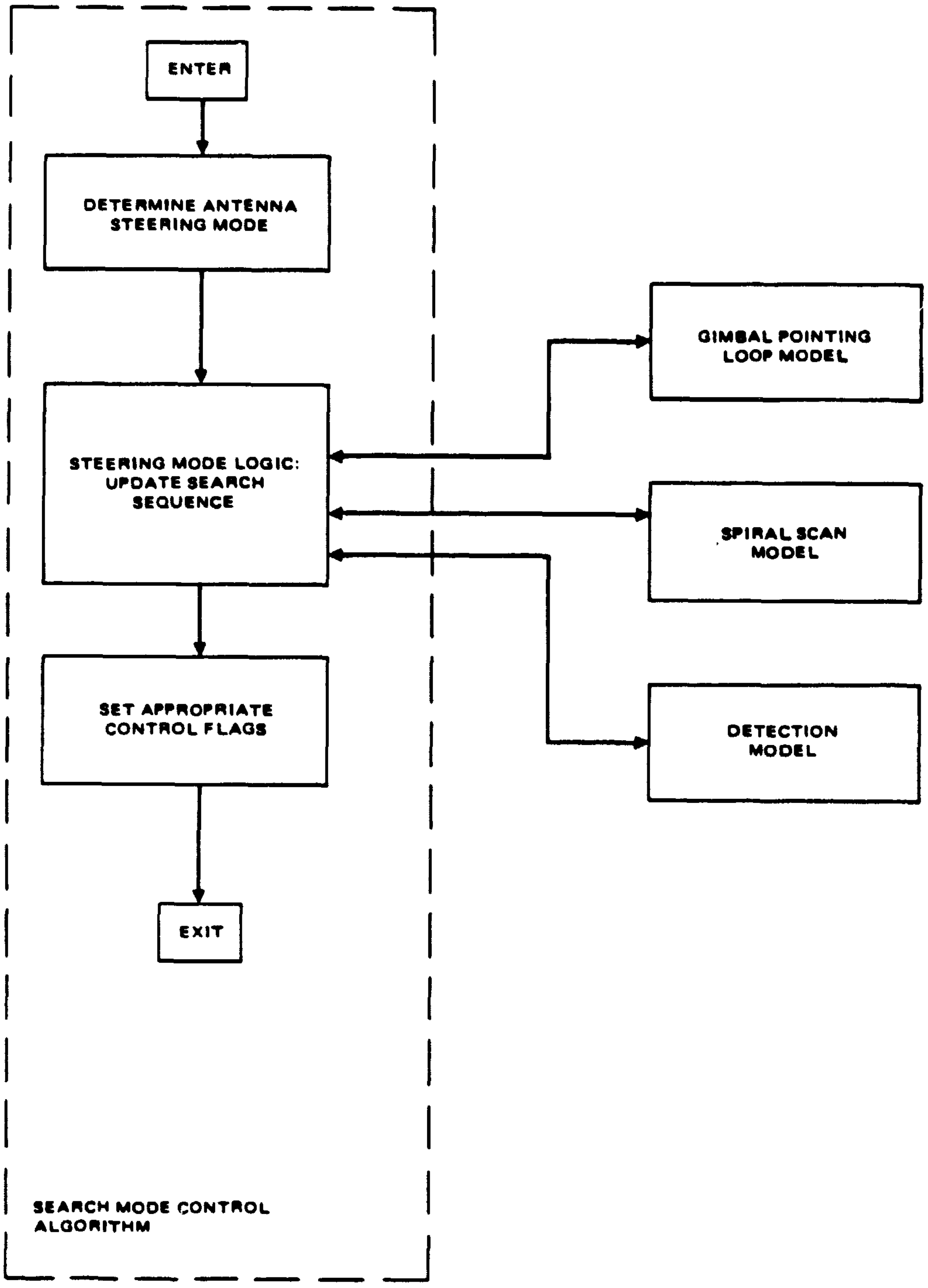
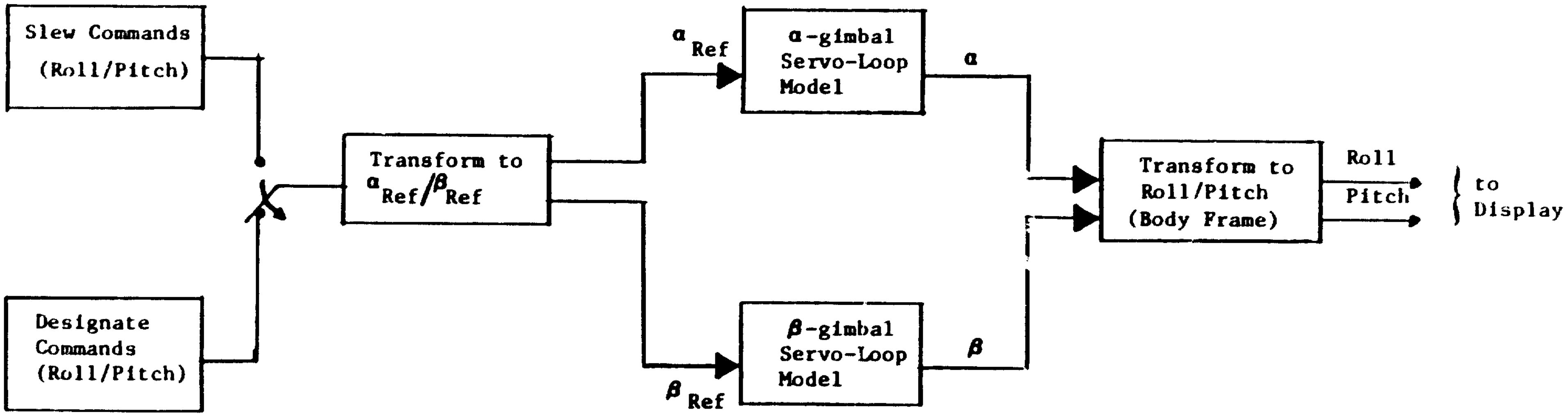

ENTER
DETERMINE ANTENNA STEERING MODE
STEERING MODE LOGIC: UPDATE SEARCH SEQUENCE
SET APPROPRIATE CONTROL FLAGS
EXIT
GIMBAL POINTING LOOP MODEL
SPIRAL SCAN MODEL
DETECTION MODEL
SEARCH MODE CONTROL ALGORITHM

Figure 1-2. Outline of search and acquisition mode computer algorithm.

Figure 1-3 SIMPLIFIED BLOCK DIAGRAM OF THE GIMBAL POINTING LOOP MODEL


Slew Commands (Roll/Pitch)
Designate Commands (Roll/Pitch)
Transform to α_{Ref}/β_{Ref}
α_{Ref}
β_{Ref}
α-gimbal Servo-Loop Model
β-gimbal Servo-Loop Model
α
β
Transform to Roll/Pitch (Body Frame)
Roll
Pitch
to Display

during the scan, to the nearest scan ring (see Figure 5-10) and tracks the target position exactly. It attempts detection if the target and the boresight are in the same scan ring in the present data cycle and were not in the same scan ring in the previous data cycle. The scan model continues in this manner until either the target is detected or an end-of-scan is reached.

Detection Model. This model contains a constant false alarm rate (CFAR) detection algorithm and a single-hit detection algorithm. These two models have the same fundamental construction which is shown in Figure 1-4 with the processing differences between the two detectors being absorbed in the SNR computation and SNR versus P_D curves used in each case. The inaccuracies of these models occur in the beamshape and scan loss computations and in the target radar cross section value. More specifically, an average beamshape/scan loss value is used when the antenna is scanning and the beamshape loss at the beginning of the data cycle is used for the entire data cycle when the antenna is being slewed. The target cross section is inaccurate because it is modeled as a fixed, predetermined value independent of aspect angle.

1.2.4 Radar Tracking Performance Model Summary

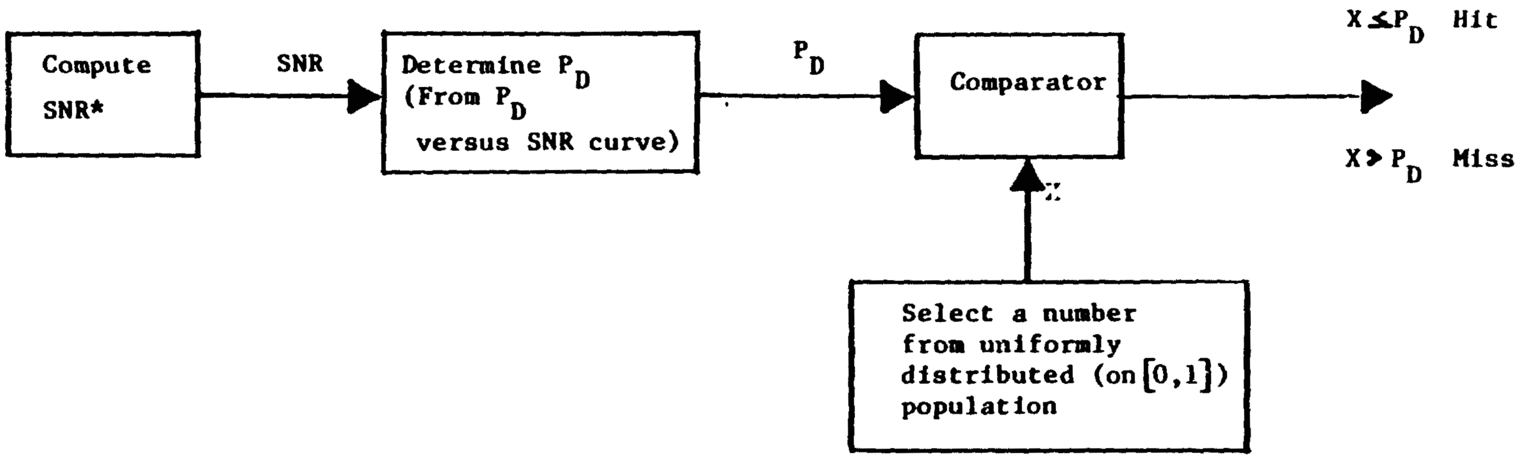
Figure 1-5 gives a simplified illustration of the track mode computer model. This model is comprised of:

- a signal generation and processing model,
- a break-track algorithm,
- an angle and angle rate tracking model,
- a range tracking model,
- a velocity processor algorithm.

The key features of each of these models are summarized below.

Signal Generation and Processing Model. A simplified diagram of the computer model used to generate the target return signal, process this signal, and produce the discriminants for the tracking loops is shown in Figure 1-6. This model is based upon several assumptions about the system and the target mot on. Of these, the ones

Figure 1-4 FUNDAMENTAL DETECTION MODEL CONFIGURATION


Compute SNR*
SNR
Determine P_D (From P_D versus SNR curve)
P_D
Comparator
$X \leq P_D$ Hit
$X > P_D$ Miss
X
Select a number from uniformly distributed (on [0,1]) population

* SNR computed at doppler filter output for CFAR detector and at video filter output for single-hit detector.

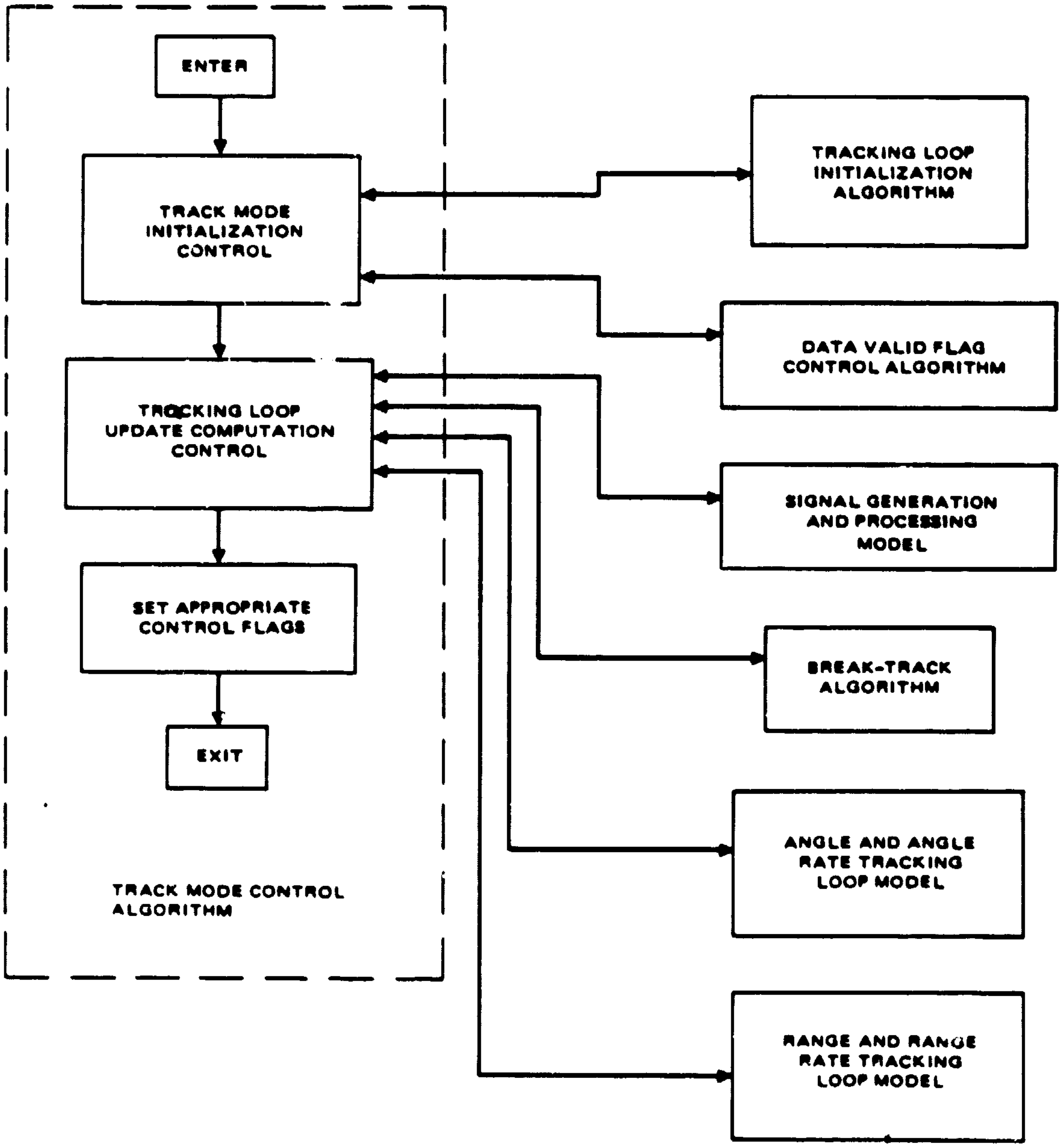
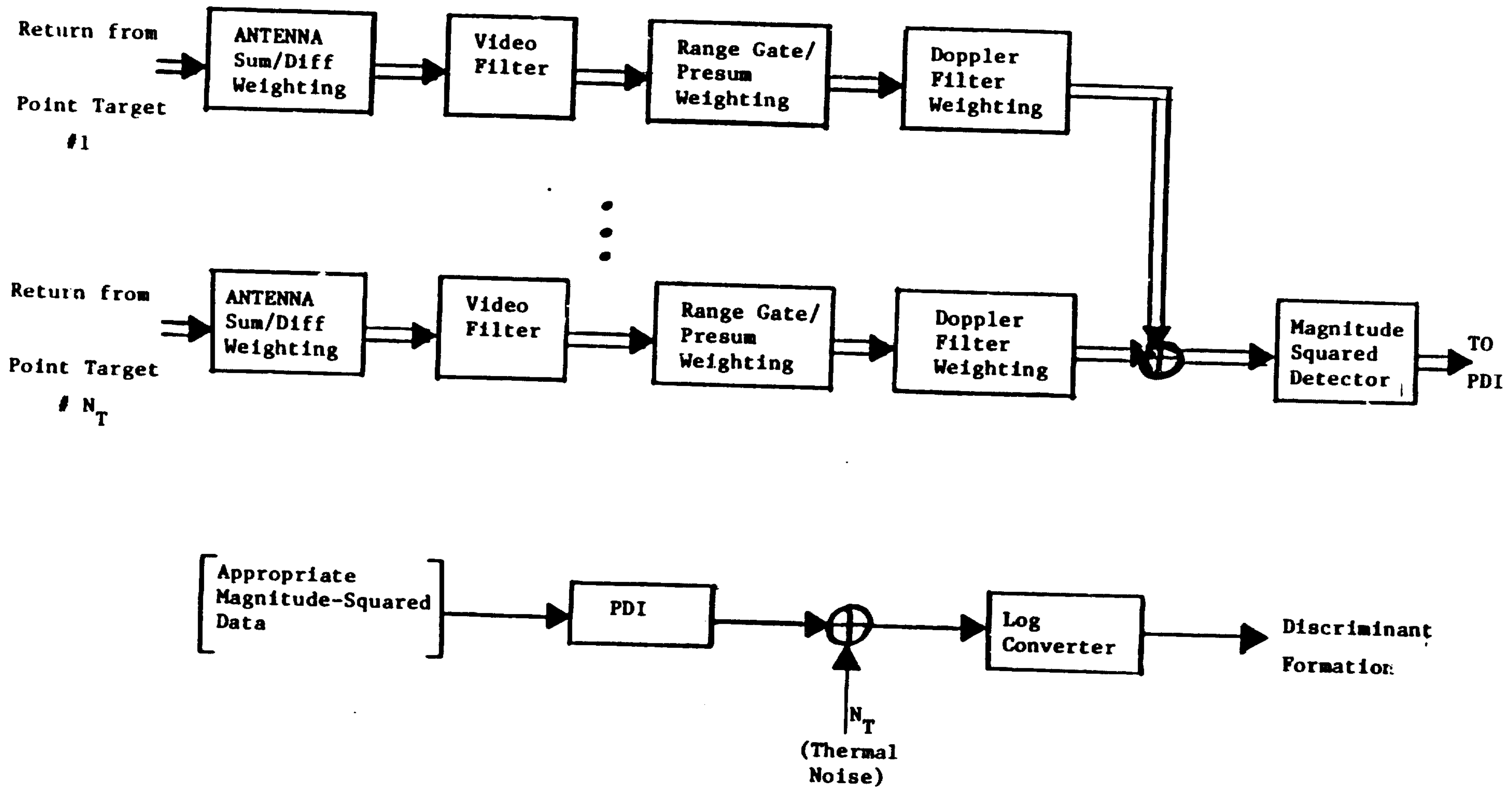

ENTER
TRACK MODE INITIALIZATION CONTROL
TRACKING LOOP UPDATE COMPUTATION CONTROL
SET APPROPRIATE CONTROL FLAGS
EXIT
TRACK MODE CONTROL ALGORITHM
TRACKING LOOP INITIALIZATION ALGORITHM
DATA VALID FLAG CONTROL ALGORITHM
SIGNAL GENERATION AND PROCESSING MODEL
BREAK-TRACK ALGORITHM
ANGLE AND ANGLE RATE TRACKING LOOP MODEL
RANGE AND RANGE RATE TRACKING LOOP MODEL

Figure 1-5. Outline of track mode computer algorithm.

Figure 1-6 TRACK MODE SIGNAL PROCESSOR COMPUTER MODEL



that will have the most impact can be stated as follows:

- any radial acceleration of the point targets over a data cycle is ignored,
- the antenna does not move with respect to the target during the data cycle.
- the receiver's RF and IF electronics work perfectly, (i.e. no coupling loss, the down conversion is error-free, and the filters don't distort the return signal, but the receiver maintains the correct noise figure and noise bandwidth),
- quantization noise contributed by the signal processing chain from the A/D to the log converter is neglected,
- Automatic Gain Control (AGC) is not implemented.

A complete list of model assumptions and approximations is given in Section 6.4.1 and Appendix C. It is noted that this model also generates an estimate of the radar signal strength which is sent to the cockpit display. This value is taken as the SNR referenced to the video filter output and is very accurate for $SNR_V > > 1$, but will not be valid for $SNR_V \overset{<}{\sim} 1$.

Break-Track Algorithm. The computer model of the break-track algorithm is identical to the algorithm used in the Ku-Band Radar system. A simplified block diagram of this algorithm is given in Figure 1-7.

Angle and Angle Rate Tracking Loop Model. This model is used for estimating the target inertial roll and pitch rate and tracking the target roll and pitch angles in the GPC-ACQ and the Auto Track Modes. It consists of two tracking loops: one for each antenna gimbal. The basic loop model adopted for each gimbal is the second order loop shown in Figure 1-8 for the α - loop. These loops are inertially stabilized, as required, and include the following error sources: target error effects (to the extent that the target scattering model is correct), thermal noise, boom deployment error, radar offset error, discriminant error, and gimbal bias error.

Range Tracking Loop Model. A simplified block diagram of the range tracking

Figure 1-7 SIMPLIFIED BLOCK DIAGRAM OF BREAK-TRACK ALGORITHM

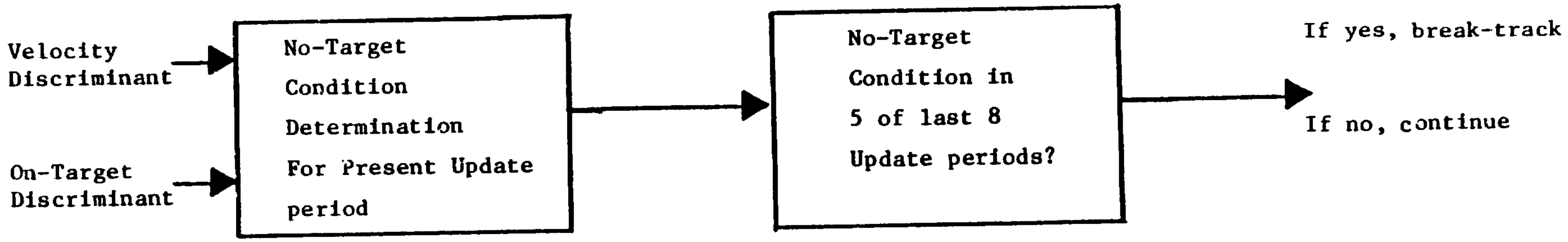
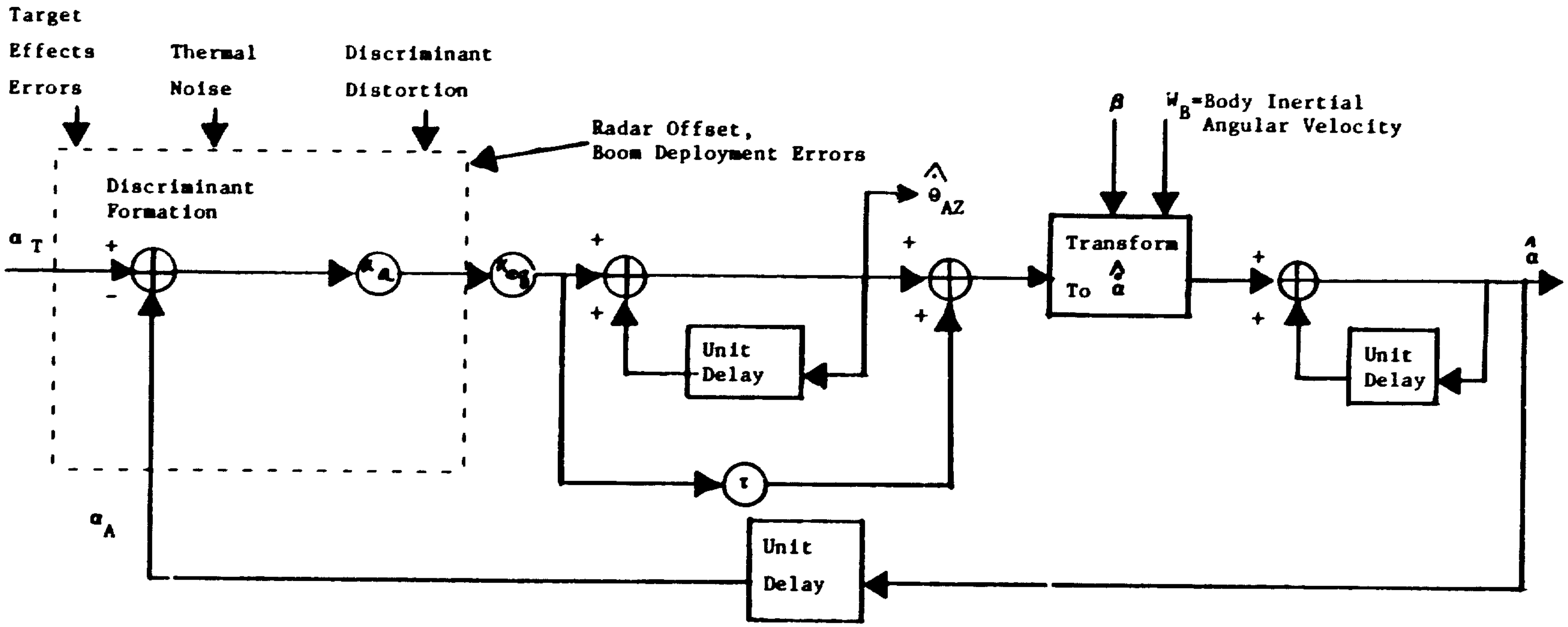

Velocity Discriminant
On-Target Discriminant
No-Target Condition Determination For Present Update period
No-Target Condition in 5 of last 8 Update periods?
If yes, break-track
If no, continue

Figure 1-8 ANGLE AND ANGLE RATE DISCRETE-TIME TRACKING LOOP MODEL



NOTE: The β-gimbal loop is similar in form to the α-gimbal loop shown above.

loop computer model is given in Figure 1-9. The loop filter equations and the loop constants for the model are identical to those used in the Ku-Band Radar system. Error sources incorporated into the model include target-effects, thermal noise, discriminant distortion, and a fixed average range bias error that accounts for unknown and time varying time delays.

Velocity Processor Model. The velocity processor computer model is shown in Figure 1-10. This model of the velocity processor is functionally identical to the algorithm used in the Ku-Band Radar system. That is, the equations, the logic and the number of bits of accuracy at each step are identical. Error sources modeled include target-effects, thermal noise, and discriminant distortion.

1.3 REPORT ORGANIZATION

The remainder of the report is organized in the following manner. In Section 2 all of the coordinate systems and the vector notation required for the description and analysis of the Ku-Band Rendezvous Radar simulation model are defined. In section 3 the parent simulation/rendezvous radar simulation interface requirements are defined. Presented in this discussion are a definition of the data required from the parent simulation by the rendezvous radar simulation, the effects of different computer cycle times on rendezvous radar model tracking accuracies, and the effects of different allowed computing times per cycle on the point target model complexity. Section 4 gives complete details of the target modeling method. In Section 5, a detailed description of the radar search and acquisition performance model is presented and Section 6 gives a complete description plus supporting analysis of the radar tracking performance model.

Figure I-9 RANGE DISCRETE-TIME TRACKING LOOP MODEL

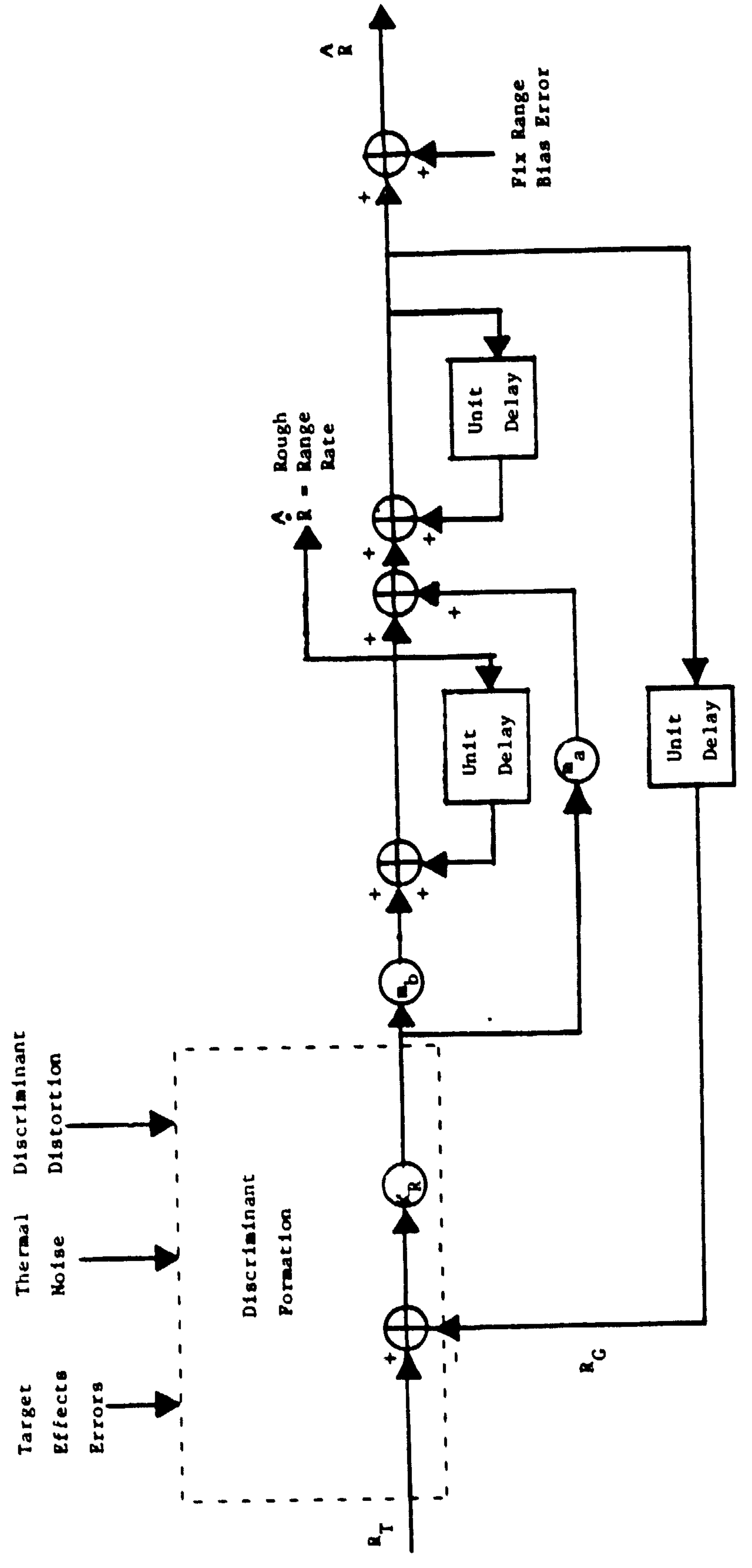
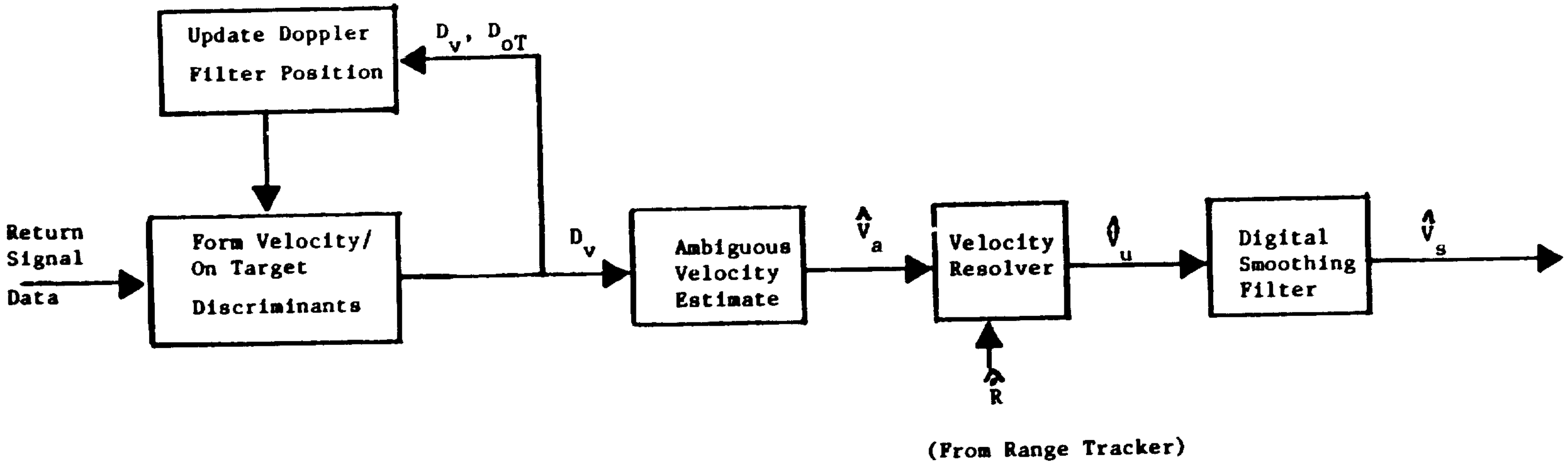


Figure :-10 KU-BAND RADAR VELOCITY PROCESSOR



2. DEFINITION OF COORDINATE SYSTEMS AND VECTOR NOTATIONS

Since vectors, transformation operators, and a variety of coordinate systems pervade the description and analysis of the Ku-Band Radar performance computer model, we begin with definitions of all coordinate systems and vector and operator notation used in this report.

2.1 COORDINATE SYSTEM DEFINITIONS

In all, there are five coordinate systems that are useful in the description of the computer model. All of these coordinate systems have the following properties. Each reference frame is a right-handed coordinate system and positive rotation about a coordinate axis of a given frame is defined by the illustration in Figure 2-1.

Target(T) Frame. This coordinate system is defined to be fixed in the target. It will be most convenient to assume that the frame origin is coincident with the target c.g. and to choose an orientation that most easily accommodates the target description in the computer. Examples of possible target frame orientations for a multiple-point target are given in Figure 2-2.

Orbiter Body(B) Frame. Definition of this reference frame is the same as that given in [10]. The origin of this frame lies at the c.g. of the Shuttle Orbiter. Its x-axis lies along the body with the nose in the positive x-region and its y-axis lies along the wings with the right wing in the positive y-region. This reference frame is shown in Figure 2-3.

Radar(R) Frame. The Radar Frame origin is located at the B-frame coordinates (48, 11, -6), which corresponds to the center of the antenna gimbals. The x-y plane of the Radar frame is parallel to the x-y plane of the Body frame, but the x-y axes of the Radar frame are rotated with respect to the Body frame x-y axes by +67° about the z-axis. This arrangement is illustrated in Figure 2-4.

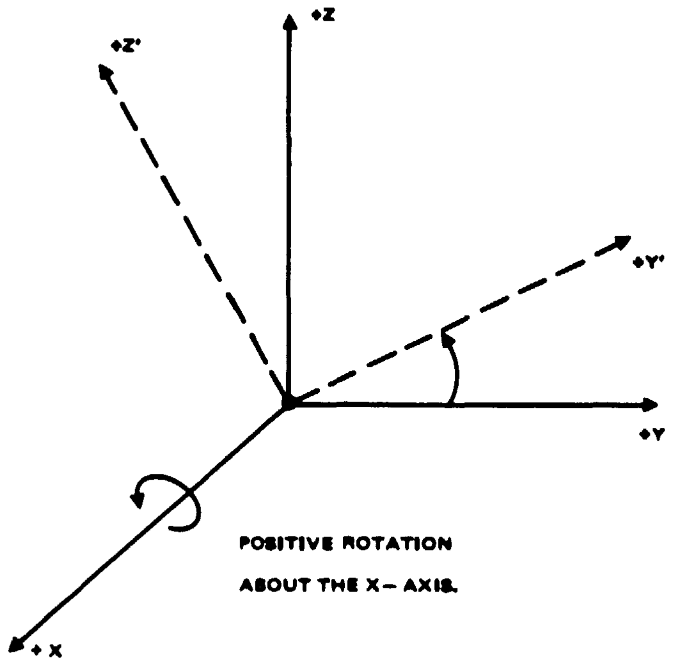


Figure 2-1. Definition of Positive Rotation about a Coordinate Axis.

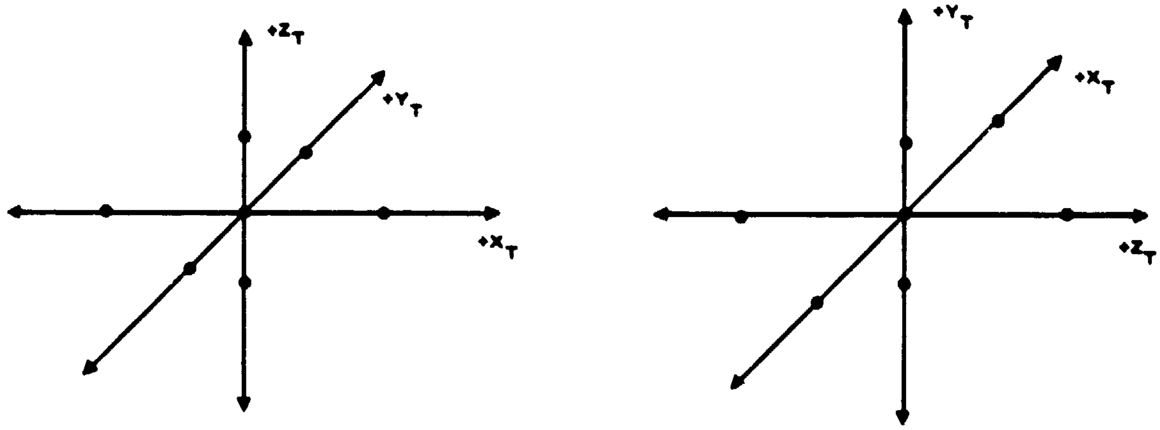


Figure 2-2. Examples of Possible Target (T) Frame Orientations.

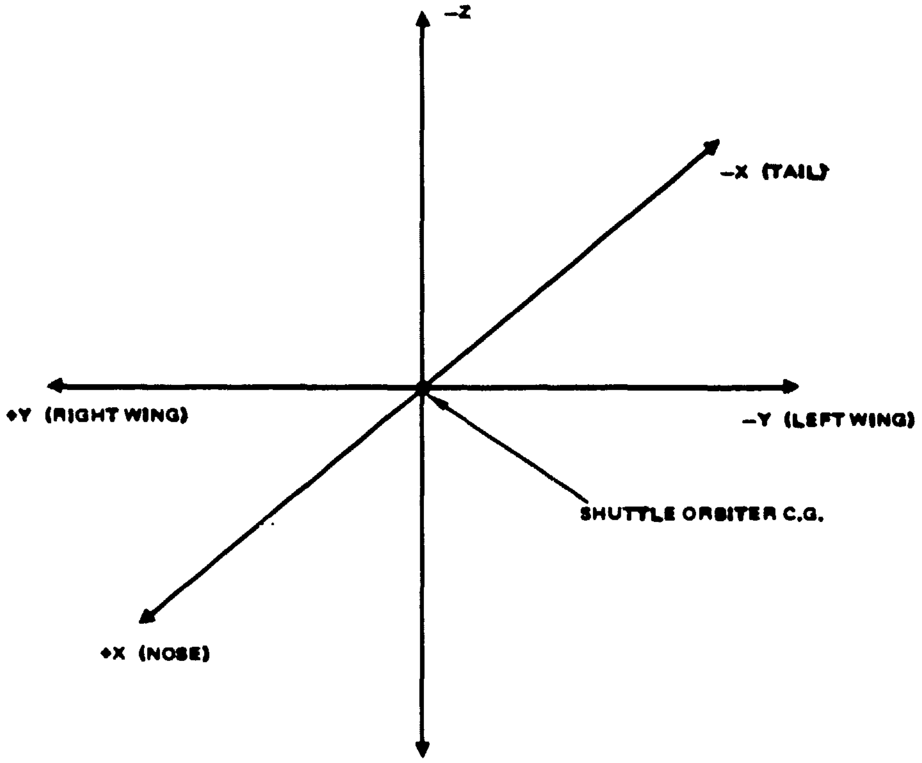


Figure 2-3. Orbiter Body (B) Frame Definition.

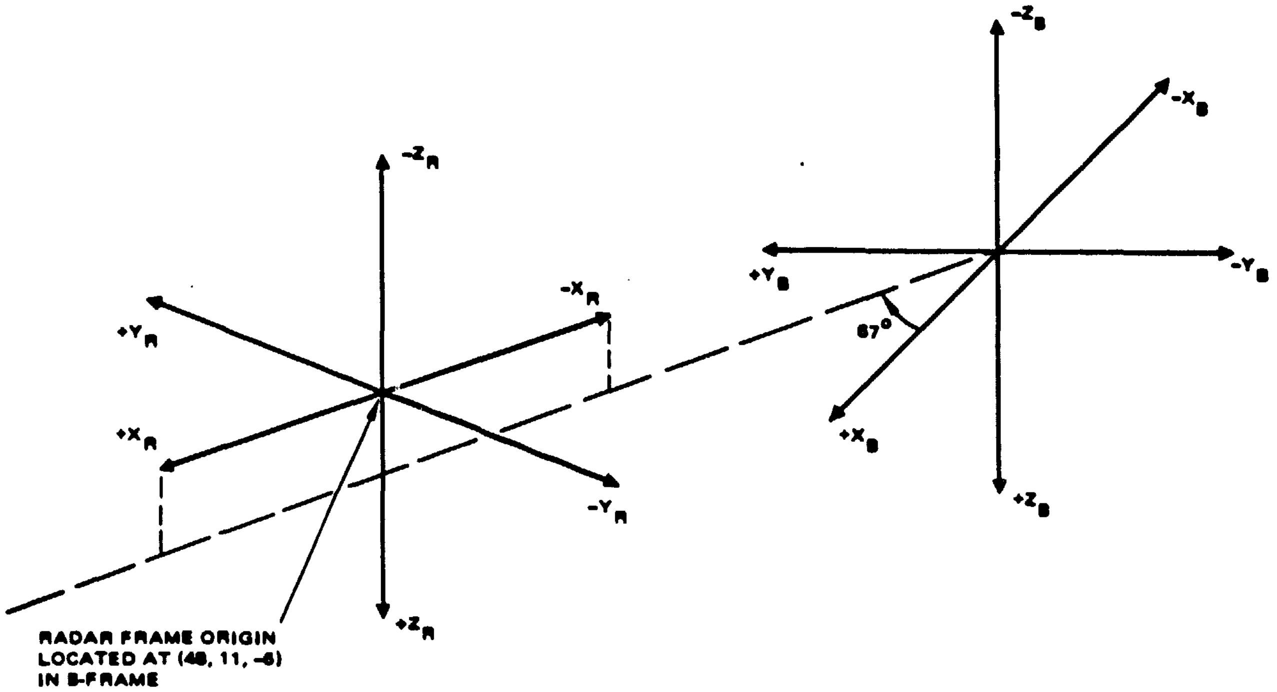


Figure 2-4. Radar (R) Frame Orientation with Respect to the Orbiter Body Frame.

Outer-Gimbal(G) Frame. This frame is fixed in the outer(or α) gimbal. Its origin is coincident with the Radar frame origin and its x-axis is coincident with the Radar frame x-axis. The y-z axes of the outer-gimbal frame are rotated by an amount α (variable) about the x-axis of the Radar frame. The angle α is measured from the minus z-axis of the Radar frame as shown in Figure 2-5.

Antenna LOS(L) Frame. This frame is fixed in the inner(or β) gimbal. . Its origin is coincident with the G-frame origin and its y-axis is coincident with the G-frame y-axis. The x-z axes of this frame are rotated by an amount β (variable) about the y-axis of the G-frame. As shown in Figure 2-6 the angle β is measured from the minus z-axis of the G-frame. It should also be noted the z-axis of the antenna LOS frame is coincident with the antenna boresignt.

Other Useful Frames. The only other useful frames for the present discussion are the Body, Radar, Outer-Gimbal, or Antenna LOS frames translated to the origin of the Target frame. These frames will be denoted by their usual letter and a zero subscript. For example, a frame centered at the target origin with its axes aligned with the antenna LOS frame will be denoted L_o.

2.2 DEFINITION OF VECTOR AND TRANSFORMATION NOTATION

In this subsection, the vector notation used to describe (1) a point scatterer's position and velocity measured in a given frame, (2) the target's inertial angular velocity, and (3) the orbiter's inertial angular velocity are defined. Also the notation for the various operations on these vectors is defined. We start with the vector description of a point scatterer's position and velocity. These are

$\vec{r}_K^{\,P}$ = k th point scatterer position expressed in P-frame coordinates.

and

$\dot{\vec{r}}_K^{\,P}$ or $\vec{v}_K^{\,P}$ = k th point scatterer velocity measured in the P-frame and expressed in P-frame coordinates.

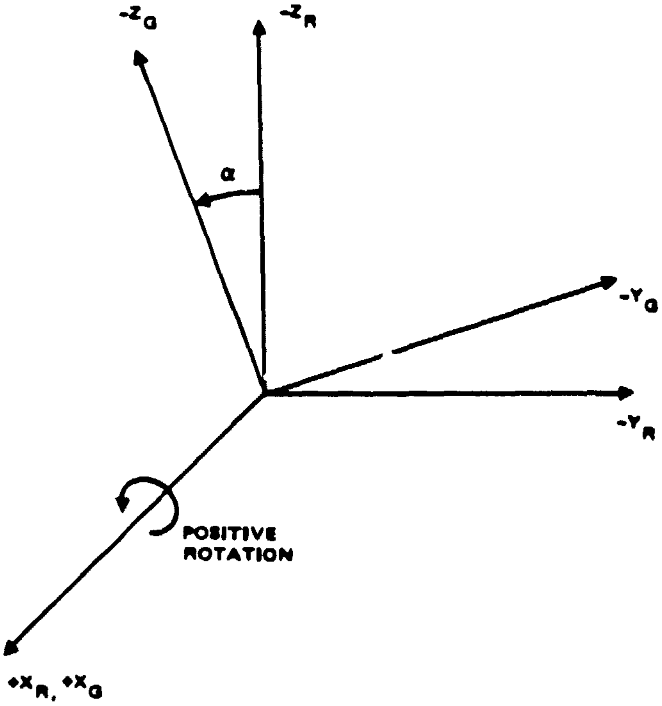


Figure 2-5. Outer Gimbal (G) Frame Orientation with Respect to the Radar Frame.

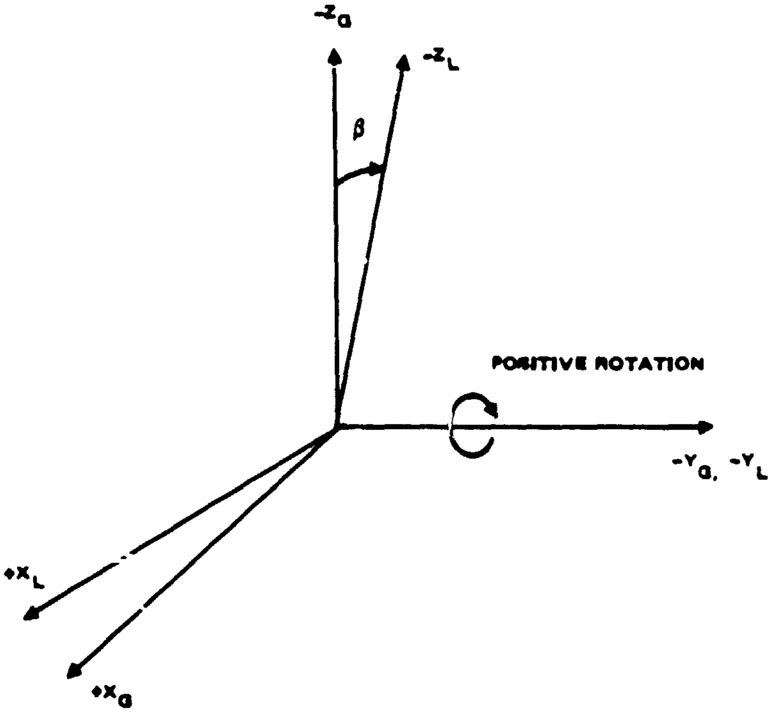


Figure 2-6. Antenna Los (L) Frame Orientation with Respect to the Outer G'mbal Frame.

where k = 1, 2, 3,---,N. Similarly, the vector description of the position and velocity of the target c.g. is given by

$\vec{r}_o^{\,P}$ = target c.g. position expressed in P-frame coordinates,

and

$\dot{\vec{r}}_o^{\,k}$ or $\vec{v}_o^{\,k}$ = target c.g. velocity measured in the P-frame and expressed in P-frame coordinates,

where the subscript o will always be associated with the target c.g. The inertial angular velocity for the target and the orbiter are defined by the notation

$\vec{w}_T^{\,P}$ = inertial angular velocity of the target about a specified point expressed in P-frame coordinates.

and

$\vec{w}_B^{\,P}$ = inertial angular velocity of the Shuttle Orbiter about a specified point expressed in P-frame coordinates.

In component form, any of the above vectors can be expressed as a 3 x 1 column vector. For example,

$$\vec{r}_k^{\,P} = \begin{pmatrix} \vec{r}_{kx}^{\,P} \\ r_{ky}^{P} \\ r_{kz}^{P} \end{pmatrix}$$

where r_{kx}^{P}, r_{ky}^{P}, and r_{kz}^{P} are the components along the x, y, z P-frame axes, respectively. Also, it should be pointed out that if the reference frame under consideration is clear from the text, then the superscript will be dropped from the vector.

The next set of definitions describe the notation used for various vector operations of interest. A primary vector operation used throughout the development is the one that transforms a vector expressed in coordinate system A to a vector expressed in coordinate system B where A and B have the same origin.

This operation will be denoted T_{BA} and has the following features. Combining T_{BA} with the vector notation from the previous paragraph, we obtain

$$\vec{r}_K^{\,B} = T_{BA} \, \vec{r}_K^{\,A}$$

Also, this transformation notation has the useful property that

$$T_{CA} = T_{CB} \, T_{BA}$$

There are two other vector operations that are of use in this report. They are the vector dot-product, denoted by $\vec{a} \cdot \vec{b}$, and the vector cross-product, denoted by $\vec{a} \times \vec{b}$. These two products have the usual meaning.

3. RADAR SIMULATION/PARENT SIMULATION INTERFACE DESCRIPTION AND REQUIRMENTS

Development of the interface between the parent simulation and the Ku-Band Radar performance simulation is based upon the following assumptions:

(1) the amount of information passed across the interface should be kept to a minimum.

(2) the parent simulation (NASA) responsibilities are

- to define and generate all shuttle orbiter and target motion, including translational and rotational motion,
- to provide all cockpit and GPC radar control information to the radar simulation,
- to accept all radar tracking and status data generated by the radar simulation.

(3) the radar simulation (Hughes) responsibilities are

- to define the modeling method that best represents the scattering characteristics of all targets of interest,
- to generate the target return signal and process it during the tracking phase,
- to accept GPC and cockpit control information from the parent simulation,
- to provide target tracking data and radar status data to the parent simulation.

Assumption (1) was motivated by a desire to achieve integration of the radar performance simulation computer model into the three proposed parent simulations, the SMS, the SES, and the SAIL simulator, with relative ease. Assumptions (2) and (3) were partially generated from the following reasoning. All definitions of rendezvous missions, target trajectories, and orbiter trajectories fall under the heading of NASA expertise and, thus, these quantities should be provided by the parent simulation. However, definition of a target scattering model and generation of radar return signals fall in the domain of Hughes expertise and should be provided by the radar simulation. In the following paragraphs, we shall define the radar/parent simulation interface, which is based upon the above assumptions, in detail.

3.1 INPUT DATA REQUIRED FROM THE PARENT SIMULATION

There are two types of input data required from the parent simulation. The first type is radar control data such as the desired operating mode and the target position designates that would normally be passed to the Ku-Band Radar over the modulation-demodulation (MDM) interface in the actual system. The second type of information required from the parent simulation is the data associated with target and orbiter motion, including both rotational and translational motions. This data is required to generate the target return signal and to simulate inertially stabilized tracking.

3.1.1 Required Radar Controls

Table 3-1 and Table 3-2 defines the radar control words required by the radar simulation that must be supplied by the parent simulation. In the actual hardware, each of the controls listed is sent to the radar either in discrete or serial word form through the MDM interface. It should be noted that the list of controls in Table 3-1 and Table 3-2 represents only those controls required in the search, acquisition, and tracking phases.

3.1.2 Required Target/Orbiter Position and Motion Data

All data associated with target and orbiter motion required by the radar simulation from the parent simulation is summarized in Table 3-3. A rationale for each of these data requirements is offered below.

In order to generate the target return signal as described in Section 4, the following information is required: (1) position of each point target and

TABLE 3-1 RADAR CONTROLS REQUIRED FROM PARENT SIMULATION

SYSTEM CONTROL FUNCTION	CONTROL NAME	CONTROL VALUE	CONTROL STATE
System Power Switch	IPWR	1 2 3	Power Off Standby System On
System Mode Switch	IMODE	1 2 3	Radar Active Radar Passive Communications
Transmitter Power Level Switch	ITXP	1 2 3	High Power Medium Power Low Power
Antenna Steering Mode Switch	IASM	1 2 3 4	GPC-ACQ GPC-DES Auto Manual
Search Initiate Switch (From Control Panel)	ISRCHC	0 1	Inhibit Scan Enable Scan
Search Initiate (From GPC)	ISRCHG	0 1	Inhibit Scan Enable Scan
Slew Antenna Left/Right	IAZS	-1 0 1	Slew Left No Slew Slew Right
Slew Antenna Up/Down	IELS	-1 0 1	Slew Down No Slew Slew Up
Antenna Slew Rate	ISLR	0 1	0.4 degrees/sec 20.0 degrees/sec

TABLE 3-2 RADAR CONTROLS REQUIRED FROM PARENT SIMULATION

SYSTEM CONTROL FUNCTION	CONTROL NAME	CONTROL DESCRIPTION	UNITS
Designated Target Range	EDRNG	Estimated Target Range From GPC	Feet
Designated Target Pitch Angle	EDPA	Estimated Target Pitch Angle From GPC	Degrees
Designated Target Roll Angle	EDRA	Estimated Target Roll Angle From GPC	Degrees

Table 3-3 OTHER INPUTS REQUIRED FROM THE PARENT SIMULATION

INPUT	INPUT NAME	INPUT DESCRIPTION	UNITS

$\vec{r}_o^{\,B}$	ERTO(I)
	I=1,2,3
		Components of T-Frame Origin Position in B-frame	Feet

$\vec{v}_o^{\,B}$	EVTO(I)
	I=1,2,3
		Components of T-frame Origin Velocity Measured With Respect to B-frame and Expressed in B-frame Coordinates	Feet Per Second

T_{B_oT}	TBT (I,J) I,J = 1,2,3	Elements of Transformation Matrix that aligns T-frame axes with B-frame axes.	No Units
$\dot{T}_{B_oT}$	TBTD (I,J) I,J=1,2,3	Elements of Matrix which is time derivative of T_{B_oT} Matrix	Seconds^{-1}

W_B^B	EWB(I)
	I=1,2,3
		Orbiter inertial angular velocity expressed in B-frame Coordinates	Radians Per Second

(2) velocity of each point target as measured in the B-frame. (It should be pointed out that, ultimately, we want the point target position and velocity as measured in the L-frame but, since the radar simulation is tracking the antenna gimbal motion with respect to the B-frame, the radar simulation can easily perform the transformation from the B-to-L frame.) For the k th point target these data can be described as follows. Position of the k th scatterer at a fixed time t can be expressed as

$$(3.1) \qquad \vec{r}_k^{\,B} = \vec{r}_o^{\,B} + T_{B_oT}\,\vec{r}_k^{\,T}$$

where Figure 3-1 illustrates the relation between these three vectors. Velocity of the k th scatterer as measured in the B-frame is given by

$$(3.2) \qquad \dot{\vec{r}}_k^{\,B} = \dot{\vec{r}}_o^{\,B} + \dot{T}_{B_oT}\,\vec{r}_k^{\,T}$$

where the dot above a quantity represents time differentiation of that quantity. It is noted that equation (3.2) is obtained by time differentiating equation (3.1) and observing that $\vec{r}_k^{\,T}$ is fixed from the rigid lattice assumption (See Section 4). Since $\vec{r}_o^{\,B}$ and $\dot{\vec{r}}_o^{\,B}$ are associated with target translational motion and since T_{B_oT} and $\dot{T}_{B_oT}$ are associated with target rotational motion, they will be provided by the parent simulation under assumption (2). $\vec{r}_k^{\,T}$ is part of the target model definition and will be provided by the radar simulation under assumption (3).

Orbiter inertial angular velocity $\vec{w}_B^{\,B}$ is required to perform tracking of the target inertial azimuth and elevation rates. The reason for this requirement is shown in Section 6.

3.2 OUTPUT DATA TO THE PARENT SIMULATION

All data output to the parent simulation are defined in Table 3-4. This data includes all cockpit radar display responses and the target tracking

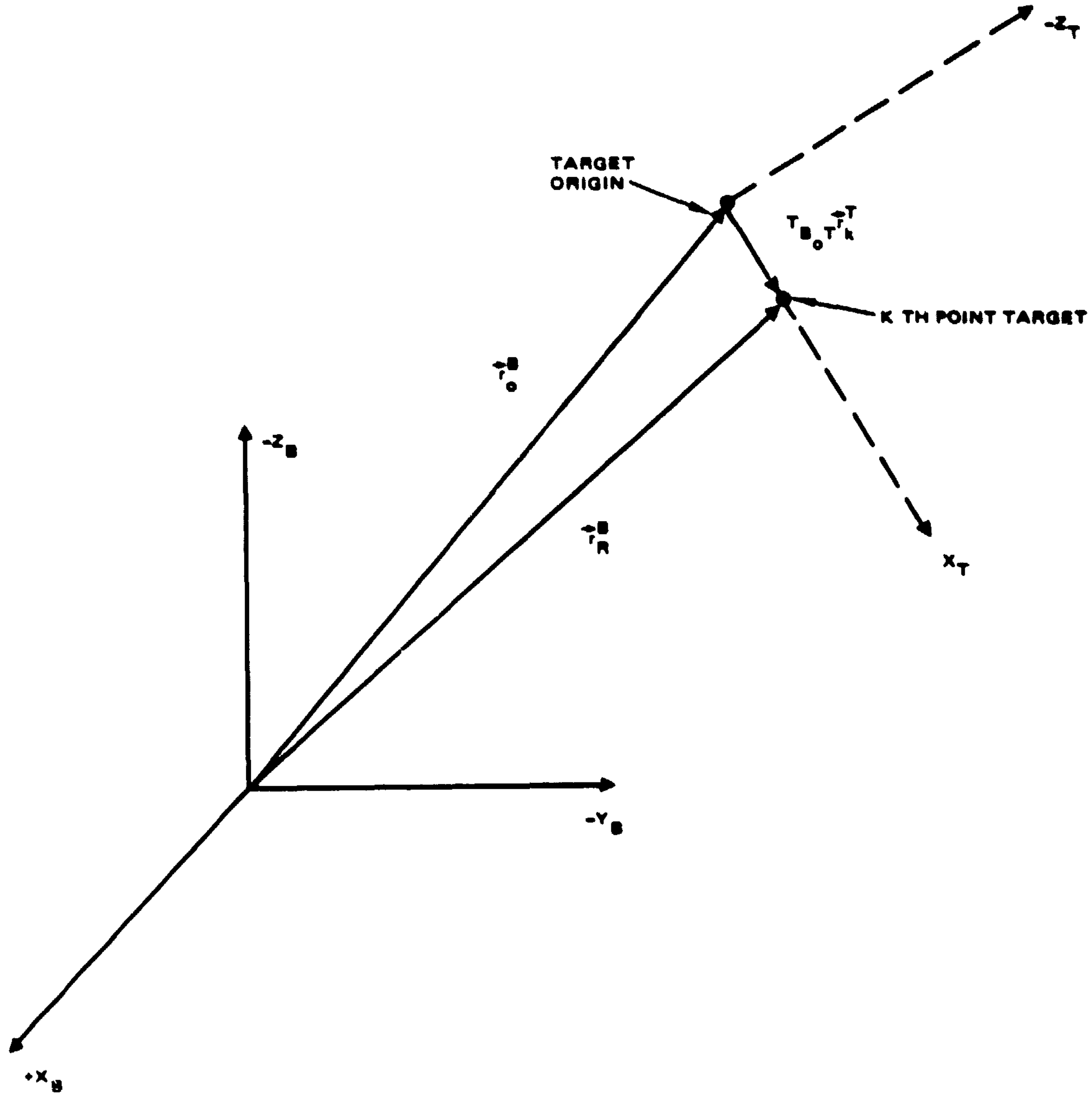


Figure 3-1. Illustration of Orbiter – Point Target Geometry.

Table 3-4 RADAR SIMULATION OUTPUT

OUTPUT DATA DESCRIPTION	OUTPUT NAME	OUTPUT VALUE	OUTPUT STATES	UNITS
Scan Warning Flag	MSWF	0 1	Scan Warning False Scan Warning True	
Track Flag	MTF	0 1	Target Track False Target Track True	
Search Flag	MSF	0 1	Target Search False Target Search True	
Estimated Target Range	SRNG	Variable		Feet
Estimated Target Range Rate	SRDOT	Variable		Feet Per Second
Estimated Target Pitch Angle	SPANG	Variable		Degrees
Estimated Target Roll Angle	SRANG	Variable		Degrees
Estimated Target Pitch Rate	SPRTE	Variable		mrad Per Second
Estimated Target Roll Rate	SRRTE	Variable		mrad Per Second
Estimated Radar Signal Strength	SRSS	Variable		dB
Angle Data Valid Flag	MADVF	0 1	Angle Data invalid Angle Data valid	

Table 3-4 RADAR SIMULATION OUTPUTS (continued)

OUTPUT DATA DESCRIPTION	OUTPUT NAME	OUTPUT VALUE	OUTPUT STATES	UNITS
Angle Rate Data Valid Flag	MARDVF	0 1	Angle Rate Data Invalid Angle Rate Data Valid	
Range Data Valid Flag	MRDVF	0 1	Range Data Invalid Range Data Valid	
Range Rate Data	MRRDVF	0 1	Range Rate Data Invalid Range Rate Data Valid	

data required by the guidance and navigation computer.

3.3 INPUT/OUTPUT DATA FORMAT

The technique used to pass data between the controlling program (parent simulation) and the subprogram (radar simulation) is to establish several labeled common storage areas. Labeled common is useful because it allows one to break a large common block into several smaller, independent common blocks which are distinguished by assigning them different labels. Thus, one can modify a section of common without having to perform bookkeeping on the whole array. Further information about labeled common can be found in [11].

In the development of the radar simulation the common block used for the interface between the two programs is divided into three parts. These are labeled: CNTL, INPUT, and OUTPUT. CNTL contains the radar control data required from the parent simulation and defined in Table 3-1 and Table 3-2. INPUT contains the target/ orbiter motion data required from the parent simulation and defined in Table 3-3. OUTPUT contains the radar data output to the parent simulation and defined in Table 3-4.

3.4 INTERFACE TIMING REQUIREMENTS

Parent/Radar simulation interface timing involves (1) the length of the parent simulation update period called the (cycle time) and (2) the fraction of the period alloted to the radar simulation for computation of required radar outputs. The details of these two topics are summarized below.

3.4.1 Simulati^n Cycle Time Requirements

Table 3-5 summarizes the different update periods for the various Ku-Band Radar tracking modes and the update periods for the three simulators. These data show that the sample interval for each of the tracking modes differs from the update periods of the three parent simulators. This would imply that the radar discrete time tracking loops must operate in an asynchronous-fashion

TABLE 3-5 SUMMARY OF KU-BAND RADAR AND PARENT SIMULATOR CYCLE TIMES

SYSTEM	UPDATE INTERVAL, ms
Ku-Band Radar	
7 khz PRF modes	51.
3 khz PRF modes	120.
268 hz PRF	250.
Parent Simulators	
SMS	TBS
SES (UNIVAC 1108)	200.
SAIL	TBS

w⁴th the parent simulator. However, rather than attempt this type of operation, the radar simulatior is designed to run synchronously with the parent simulation. This means that the sample interval of the discrete time tracking loops will be an integral number of update periods of the simulation computer. Then the primary question is, what is the impact of this design decision on the tracking performance? Observe that the minimum update rate of the three simulators is approximately 4 hz and the maximum loop bandwidth for any of the servos in any of the tracking modes is well under 1 hz. Therefore the minimum sample rate of the computer is at least four to five times the tracking loop bandwidth and the fidelity of the loop response should not be affected. We offer an example to illustrate this point. Consider a target at a range of 0.4 nm (largest bandwidth) which is not moving at time t<o and is being tracked by the radar. At time t=o, the target is given a step of 10 mrad/sec in roll rate with respect to the radar. The angle rate loop step response is then generated using update intervals of 50., 100., 200., and 400. milliseconds and plotted in Figure 3-2. These results show only slight error in the response for sample intervals as large as 200 m sec.

3.4.2 Maximum Computation Time Requirements

Table 3-6 gives the computation time alloted to the radar simulation per cycle for each of the simulation computers. Assuming the present multiple point scatterer target model, these computation times can be converted to the maximum number of points allowed using empirically determined conversion factor. The maximum number of points and the conversion factors for each simulator are listed in the Table 3-7.

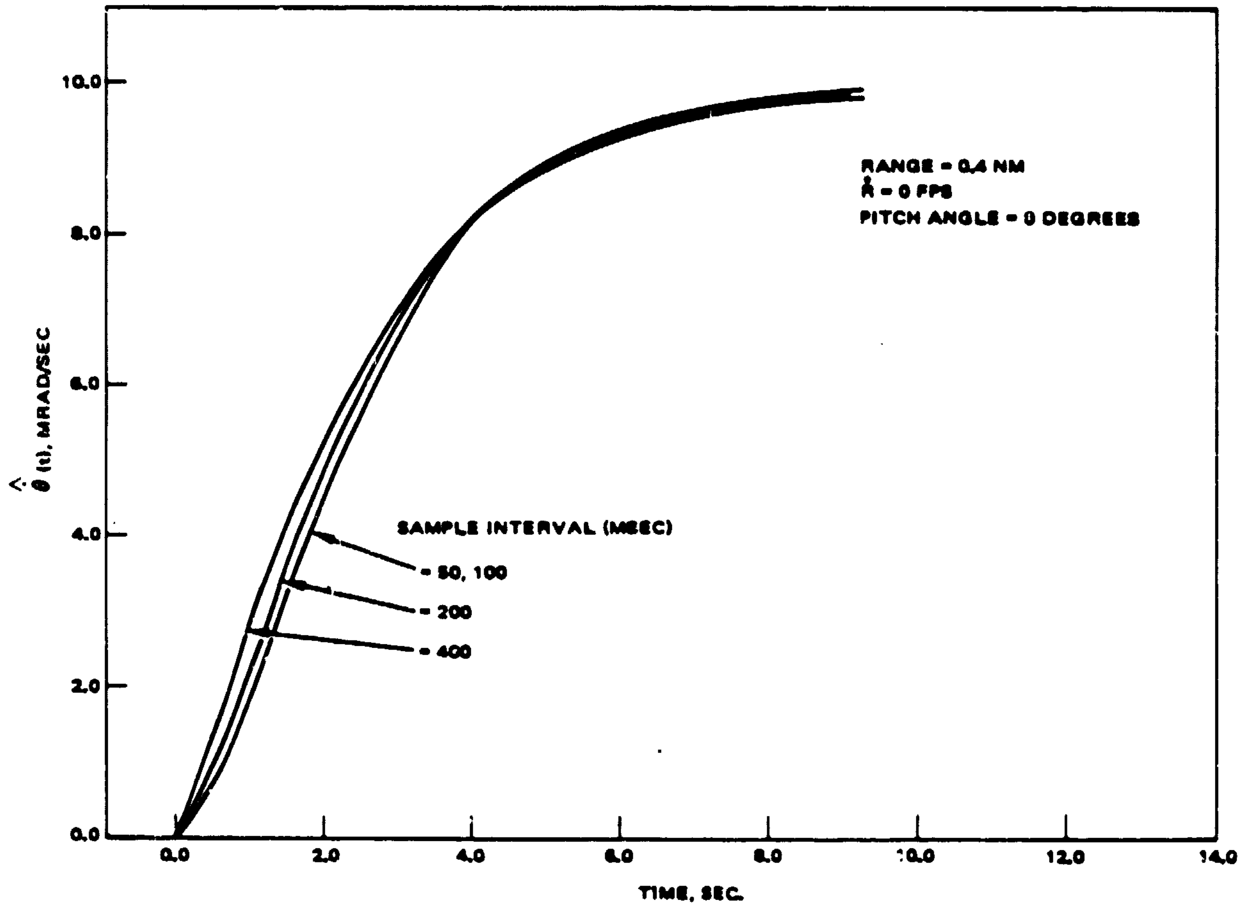


Figure 3-2. Example of Effects of Different Sample Intervals.

Table 3-6 SUMMARY OF ALLOWED COMPUTATION TIME PER CYCLE FOR EACH SIMULATOR

SIMULATOR	COMPUTATION TIME PER CYCLE, ms
SMS	TBS
SES (Univac 1108)	200.
SAIL	TBS

Table 3-7 MAXIMUM NUMBER OF ALLOWED TARGET POINTS FOR EACH SIMULATOR

SIMULATOR	TIME PER TARGET, ms	ALLOWED COMPUTATION TIME,ms	MAXIMUM NUMBER OF POINTS
SMS	TBS	TBS	TBS
SES	$\doteq$ 5.7	200	35
SAIL	TBS	TBS	TBS

4. TARGET MODELING METHOD

The purpose of target modeling is to predict target effects on the radar measurement accuracies. In this section, the general modeling approach is described, an example of the method applied to the SPAS spacecraft is provided, and a mathematical description of the resultant target return signal at the radar is given.

4.1 GENERAL APPROACH

As stated in the proposal [12], virtually all of the target effects analyses in the literature treat the target as a collection of point scatterers. This approach was adopted for the computer simulation described in this report. More specifically, our modeling method divides the spacecraft scatterers into two distinct classes: (1) those associated with simple geometric shapes and (2) those which are not. Simple shapes are modeled as point scatterers with the appropriate locations and their associated cross section functions to express the angular variation. (A review of the quantitative cross section results, taken from the literature, for several useful geometric shapes is provided in the next subsection.) Intricate or rough-surfaced areas of the spacecraft are modeled as random scatterer fields, which in turn are modeled by point scatterers with with random amplitudes and specified angular variation functions. For both types of scatterers, the angular variation of the cross section amplitude includes the approximate effects of shadowing caused by neighboring elements. These cross-section functions do not include phasing. Instead, phase effects are accounted for via the spatial separation of the target's scatterers; this is shown in quantitative terms in section 4.4.

Details of the modeling method, especially the rough-surfaced modeling, are best illustrated by the SPAS modeling example of section 4.3

4.2 SCATTERING CENTERS AND CROSS SECTIONS FOR SIMPLE AND REPRESENTATIVE SHAPES

The cross section literature can be used to extract point-scatterer models for simple geometric shapes, as follows.

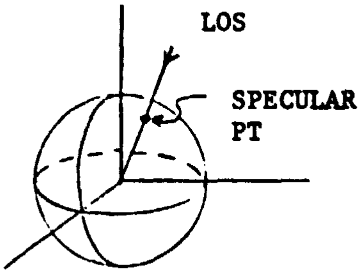
4.2.1 Smoothly Curved Bodies(Reference 13)

A well-known result of the geometrical theory of diffraction is that the main RCS contribution from a curved surface comes from the "specular point" at which the radar line of sight (LOS) is normal to the surface. The cross section is

$$\sigma = \pi R_1 R_2$$

where R_1, R_2 are the surface's principal radii of curvature at the specular point. This principle is illustrated by the following examples.

Sphere. Here $\sigma = \pi a^2$ where a is the sphere's radius. The specular point lies on the sphere's surface at the intersection of the LOS.



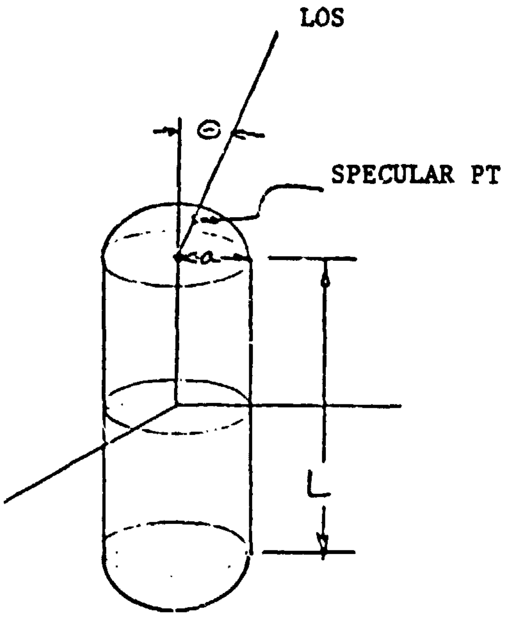
Hemispherical-Ended Cylinder. The specular point is on the upper hemisphere when the LOS is from above. One has

$$\sigma = \pi a^2$$

for all θ except θ = 90°; for the latter, the result for the cylinder (reference 14, p.9) yields

$$\sigma = \frac{2\pi a L^2}{\lambda} = 291\ aL^2$$

with dimensions in meters and λ = 0.0216 meters



The width of the "flare" at 90° can be taken to be $\pm \frac{\lambda}{L} = \pm \frac{1.24 \text{ degrees}}{L}$.

The specular point lies on the intersection of the LOS with the cylinder's surface in the xy-plane.

Torusphericel-Ended Cylinder. In the torusphericel-ended cylinder, the ends consist of a section of large radius joined tangentially to a toroidal section that in turn is joined tangentially to the cylindrical section (See Figure 4-1). Here we have

$$
\begin{aligned}
\sigma &= \pi a_o^{2}, && 0 < |\theta| < \theta_o \\
&= \pi a_1 \left[(a_o - a_1) \frac{\sin \theta_o}{\sin \theta} + a_1 \right], && \theta_o \leq |\theta| < 90^\circ \\
&= \frac{2\pi \, aL^2}{\lambda} && \theta = 90^\circ \pm \frac{1.24}{L} \text{ degrees}
\end{aligned}
$$

where we have used the results of Ref 13, p.114 for the toroid.

When the end is designed for maximum strength (everywhere equally stressed), as appears the case on the SPAS MOMS cannister,

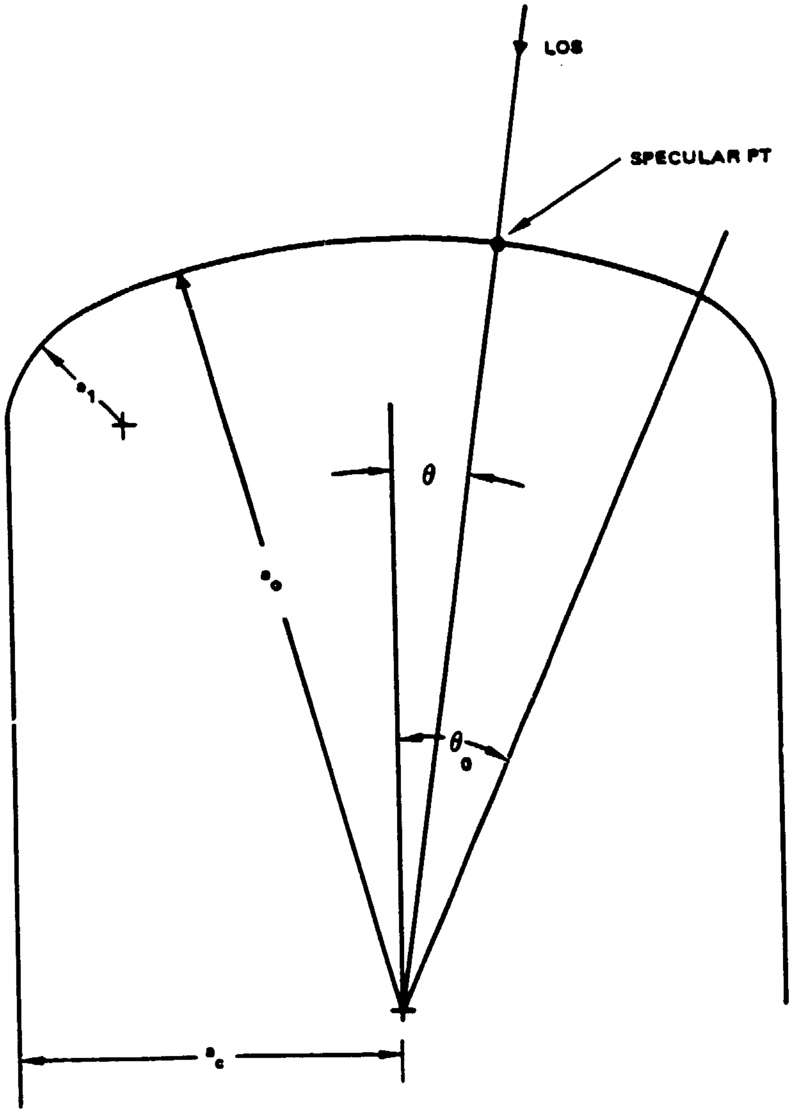
$$a_o = 2a_c$$

$$a_1 = a_c/3$$

and

$$\sin \theta_o = 0.4 \quad (\theta_o = 23.6^\circ)$$

$$
\begin{aligned}
\sigma &= 4\pi a_c^{2}, \; |\theta| < \theta_o \\
&= \frac{\pi a_c^2}{9} \left[\frac{2}{\sin \theta} + 1 \right] && \theta_o \leq |\theta| < 90^\circ \\
&= \frac{2\pi ha}{\lambda} \quad |\theta| = 90^\circ
\end{aligned}
$$



$$\sin \theta_o = \frac{a_c - a_1}{a_o - a_1}$$

Figure 4-1. Toruspherical - Ended Cylinder Geometry.

4.2.2 Other Shapes

Cylinders. Reference 14 provides cross sections for cylinders and discs. The flat-ended cylinder has three specular points at the intersection of the plane containing the LOS and the visible edges of the ends (Figure 4-2). The cross sections associated with these points are

$$\sigma_1 = \frac{.0046\text{m}^2}{\sin\theta} \left[\frac{-1}{1 + 2\cos\frac{2}{3}(\pi + 2\theta)} \pm 1 \right]^2$$

$$\text{Note:} \left\{ \begin{array}{l} + \longrightarrow \textbf{Vertical Polarization} \\ - \longrightarrow \textbf{Horizontal Polarization} \end{array} \right\}$$

$$\sigma_2 = \frac{.0046\text{m}^2}{\sin\theta} \left[\frac{-1}{1 + 2\cos\frac{4\theta}{3}} \pm 1 \right]^2$$

$$\sigma_3 = \frac{.0046\text{m}^2}{\sin\theta} \left[\frac{-1}{1 + 2\cos(\pi - 2\theta)} \pm 1 \right]^2$$

These relations indicate negligible contributions except near normal incidence ($\theta = 0^\circ, 90^\circ$). For $\theta = 0^\circ$, one has

$$\sigma_\theta = 265{,}000 \text{ a}^4 \qquad |\theta| < \frac{1.24^\circ}{2a}$$

And at $\theta = 90^\circ$, one has

$$= 291 \text{ aL}^2 \qquad |\theta| = 90^\circ \pm \frac{1.24^\circ}{L}$$

Wire, Struts. A typical spacecraft has structural elements that are typically modeled as wires, i.e., long thin elements. Reference 13 (p.107) indicates that for a long thin wire (or edge) that

$$\sigma \doteq \frac{\lambda^2 \tan^2\theta \cos^4\phi}{16\pi^3} \qquad \theta < 90^\circ$$

$$= 9.4 \times 10^{-7} \tan^2\theta \cos^4\phi \qquad \text{m}^2$$

$$\doteq \frac{L}{\pi}^2 \cos^4\phi \qquad \theta = 90^\circ$$

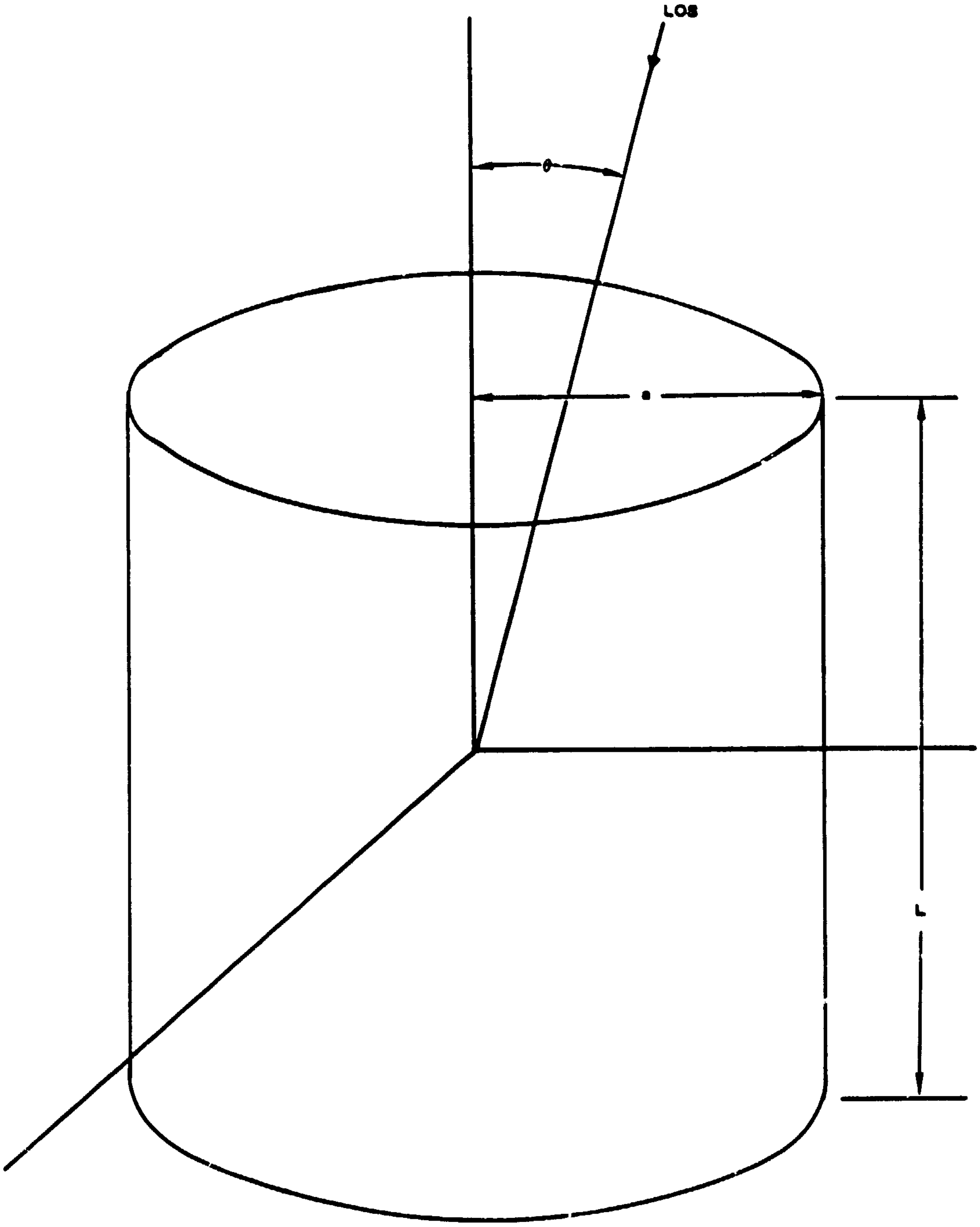


Figure 4-2. Cylinder Geometry.

where ϕ is the angle of polarization incidence and θ is the angle between the LOS and the wire axis. Thus a significant contribution is seen only at broadside, reflecting the conclusions of reference 15 that edges don't provide significant RCS contributions.

Corner Reflectors - Dihedrals. The RCS for a dihedral reflector shown in Figure 4-3 is (Reference 16, p. 589)

$$\sigma = 16\pi\ a^2 b^2\ \mathrm{Sin}^2\left(\frac{\pi}{4} + \phi\right)$$

at incidence perpendicular to the reflector axis and falls off rapidly away from normal.

Corner Reflectors - Trihedrals. Square trihedrals have cross section $\sigma = 4\pi \frac{A^2}{\lambda^2}$ with A the area normal to the LOS for which energy is redirected (Ref. 13, p.239), and for a square reflector,(Ref. 16, p.591)

$$\sigma = \frac{12\ \pi\ L^4}{\lambda^2} = 80{,}802\ L^4\ m^2$$

with L the width of each face. This RCS is maintained over a 23 degree cone about the symmetry axis. A 1-inch corner reflector thus has .033 m^2 cross section.

4.2.3 Reflector Antennas

On boresight, an antenna provides an enormous RCS. Let $G(\theta)$ be the antenna power gain pattern. Then

$$\sigma = \frac{\lambda^2 \rho\ G^2(\theta)}{4\pi}$$

where ρ is the antenna power reflection coefficient, and usually approaches unity out of band. One has

$$G\ (o) = \eta\ \frac{4\ \pi A}{\lambda^2}$$

so

$$\sigma = \eta A \cdot \frac{4\pi\ A\ \rho}{\lambda^2}\quad G_N^2(\theta)$$

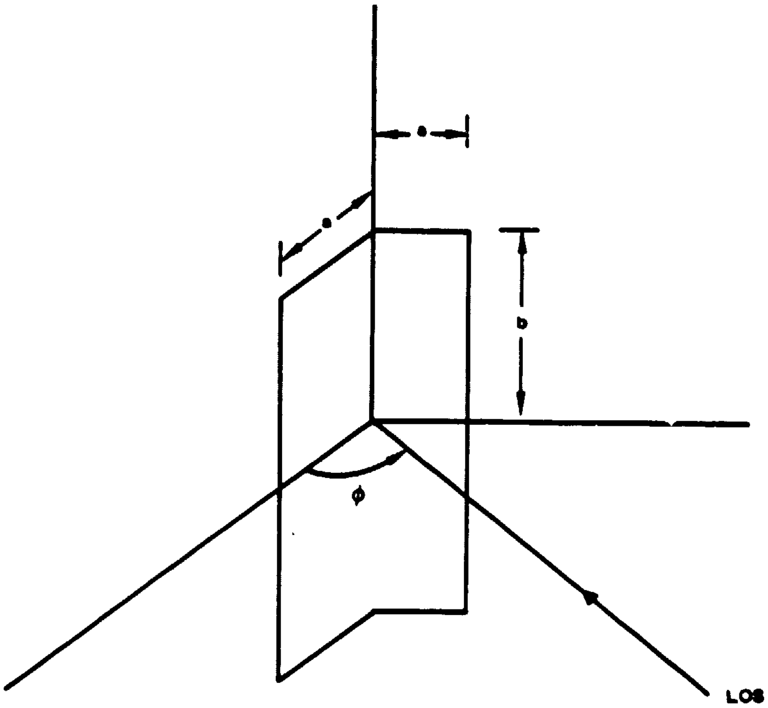


Figure 4-3. Dihedral Corner Reflector Geometry.

where $G_N(\theta)$ is normalized to its maximum value. Taking $\rho = 1$, $\lambda = .0216$ m yields

$$\sigma = 26934\ A^2\eta\ G_N^2(\theta)$$

or for a circular aperture of diameter D

$$\sigma = 16611\ D^4\eta\ G_N^2(\theta)$$

and taking $\eta = 40\%$ yields

$$\begin{aligned}\sigma &= 10774\ A^2G_N^2(\theta)\\ &= 6645\ D^4G_N^2(\theta)\end{aligned}$$

The width (first null) of this flare is about ± λ/D radians or ± 1.24/D degrees.

For a parabolic antenna, the reflector surface provides a significant return over a broad angle. For a body of revolution, Reference 13 gives

$$\begin{aligned}\sigma &= \pi R_1R_2\\ &= \pi\left|\frac{x}{\frac{d^2x}{dz^2}\sin^4\theta}\right|\end{aligned}$$

where the geometry is as shown in Figure 4-4. For the reflector

$$x = 2\sqrt{fz}$$

with f the focal length; then one obtains

$$\sigma = \pi f^2\cos^4\theta$$

and for $f/D = .5$,

$$\begin{aligned}\sigma &= \frac{\pi D^2}{4}\cos^4\theta\\ &= .785\ D^2\cos^4\theta\end{aligned}$$

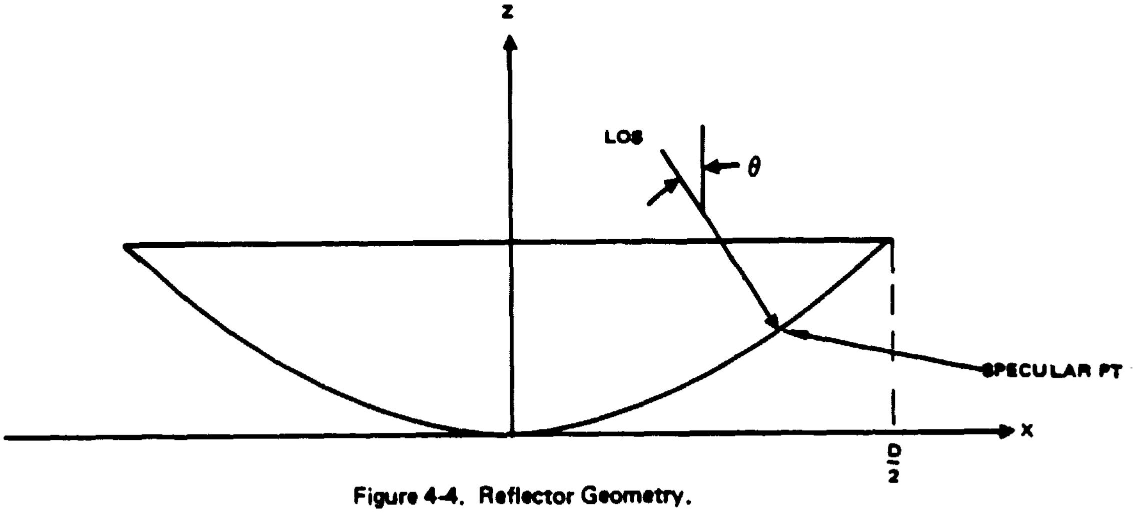


Figure 4-4. Reflector Geometry.

This RCS contribution is seen so long as the LOS intersects the reflector at normal incidence somewhere. This occurs if

$$|\theta| \leq \tan^{-1} \frac{dz}{dx} = \theta_o$$

$$\theta_o = \tan^{-1} \frac{D}{4f}$$

$$= 26.6 \text{ degrees for } \frac{f}{D} = 0.5$$

4.3 SPAS MODEL

4.3.1 Satellite and its Coordinate System

Figure 4-5 shows the SPAS satellite in isometric view and identifies our coordinate system. Figure 4-6 shows a drawing of the satellite. Define the following angles:

$$\phi_x = \text{Angle between LOS and x-axis}$$

$$= \cos^{-1} \left(\hat{u}_L \cdot \hat{u}_x\right)$$

$$\phi_y = \text{Angle between LOS and y-axis}$$

$$= \cos^{-1} \left(\hat{u}_L \cdot \hat{u}_y\right)$$

$$\phi_z = \cos^{-1} \left(\hat{u}_L \cdot \hat{u}_z\right)$$

where $\hat{u}_x$, $\hat{u}_y$, $\hat{u}_z$ are unit vectors aligned with the x, y and z axes, and $\hat{u}_L$ is a unit vector aligned with the LOS.

4.3.2 Scatterer Selection Strategy

Two classes of scatterers may be identified: those that arise due to geometric shapes discussed in Section 4.2, and those that do not. Among the former are tanks, experiment cannisters, mounting pallets, and the S-band antenna. Among the latter are complex areas such as are seen on the SPAS electronics pallets or structural areas, where multiple bounces and corner-reflector-like areas can give rise to significant and relatively orientation-free return. We model the former explicitly, and attempt to model the latter by associating point scatterers with the major complex areas, choosing the scatter cross section using a rough-surface model.

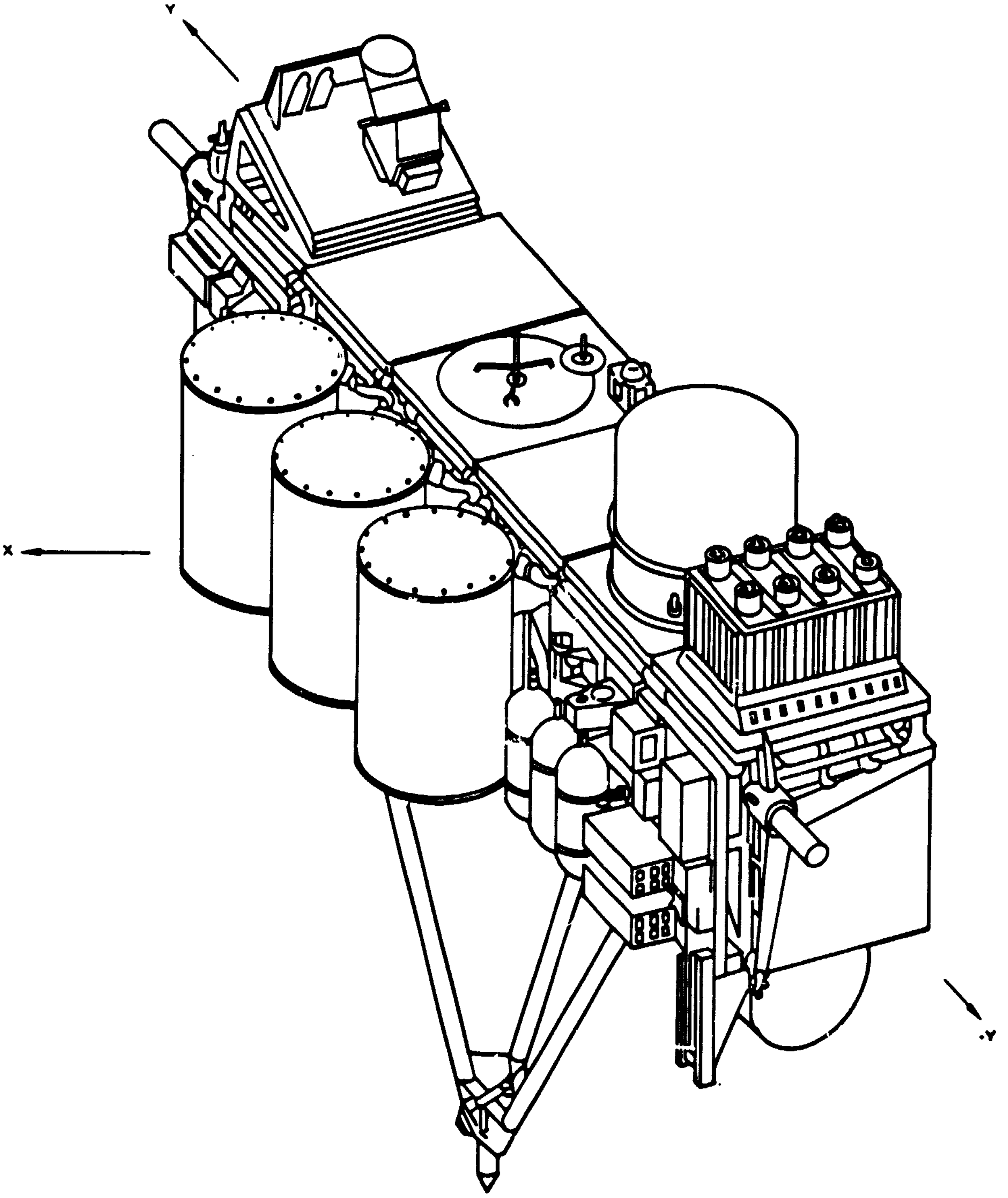
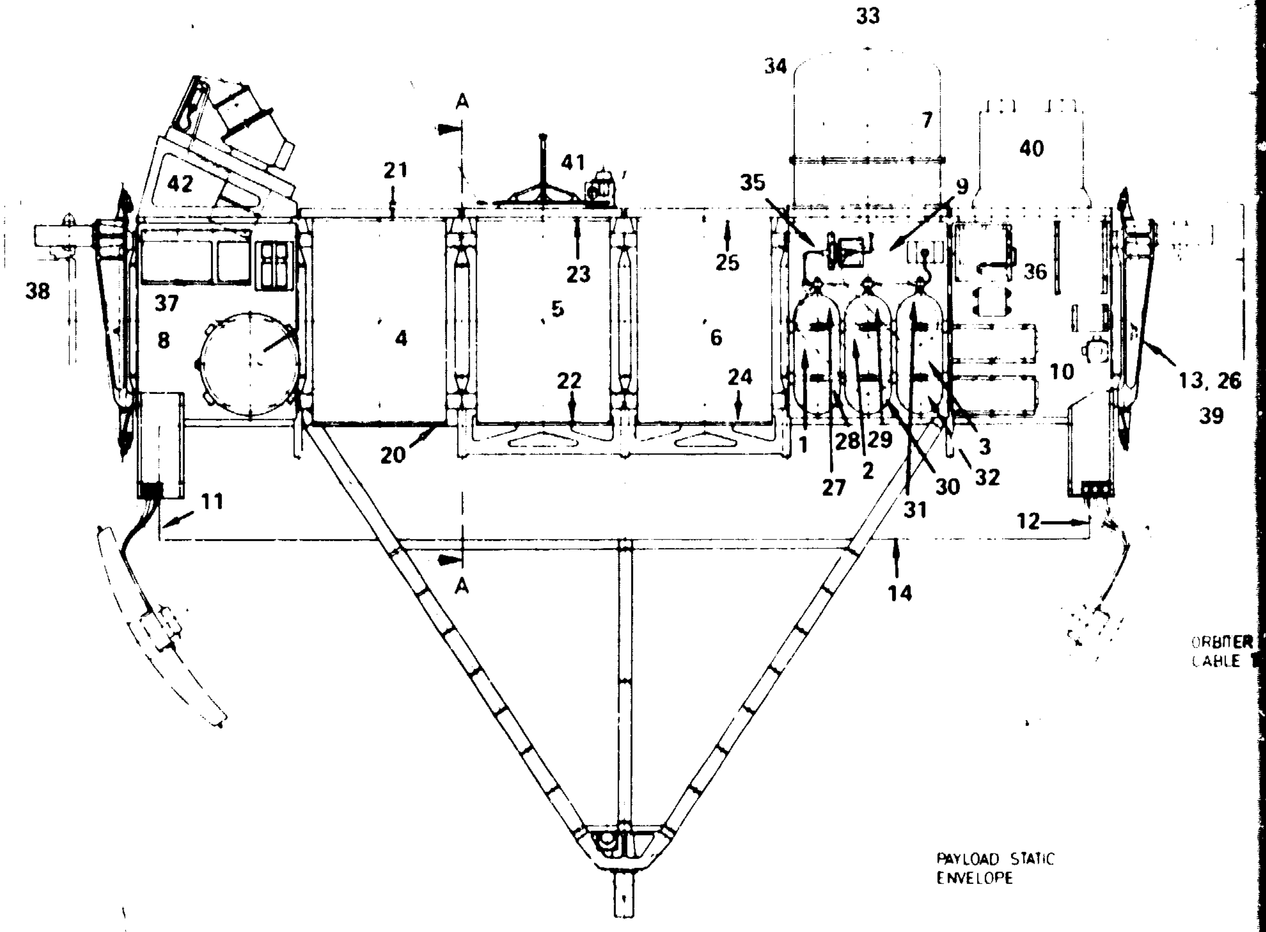
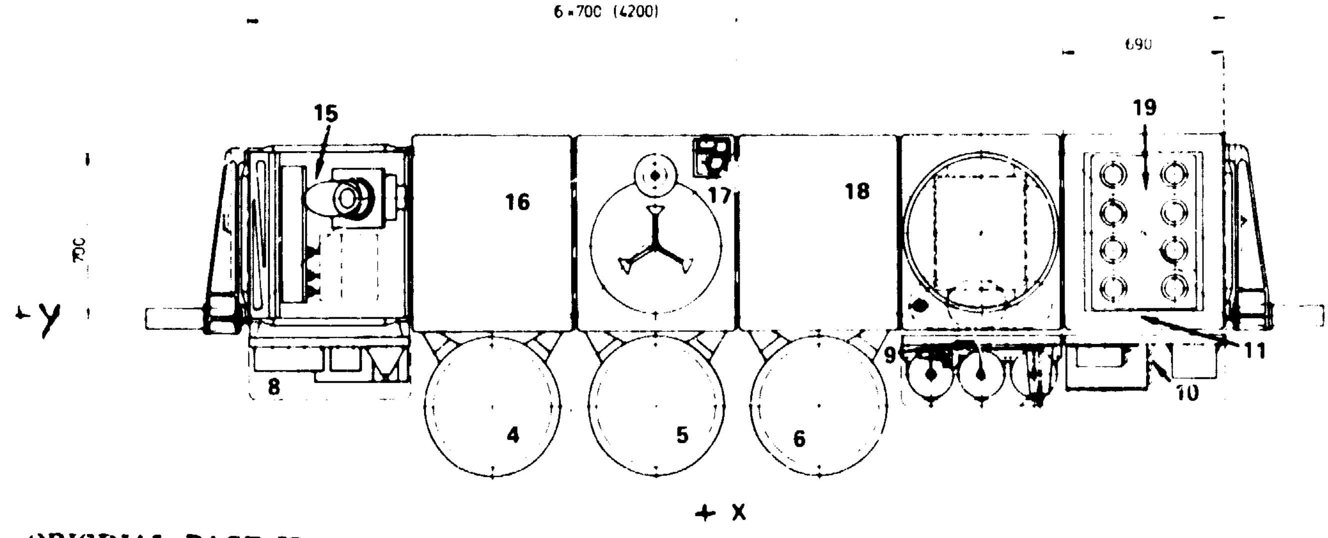
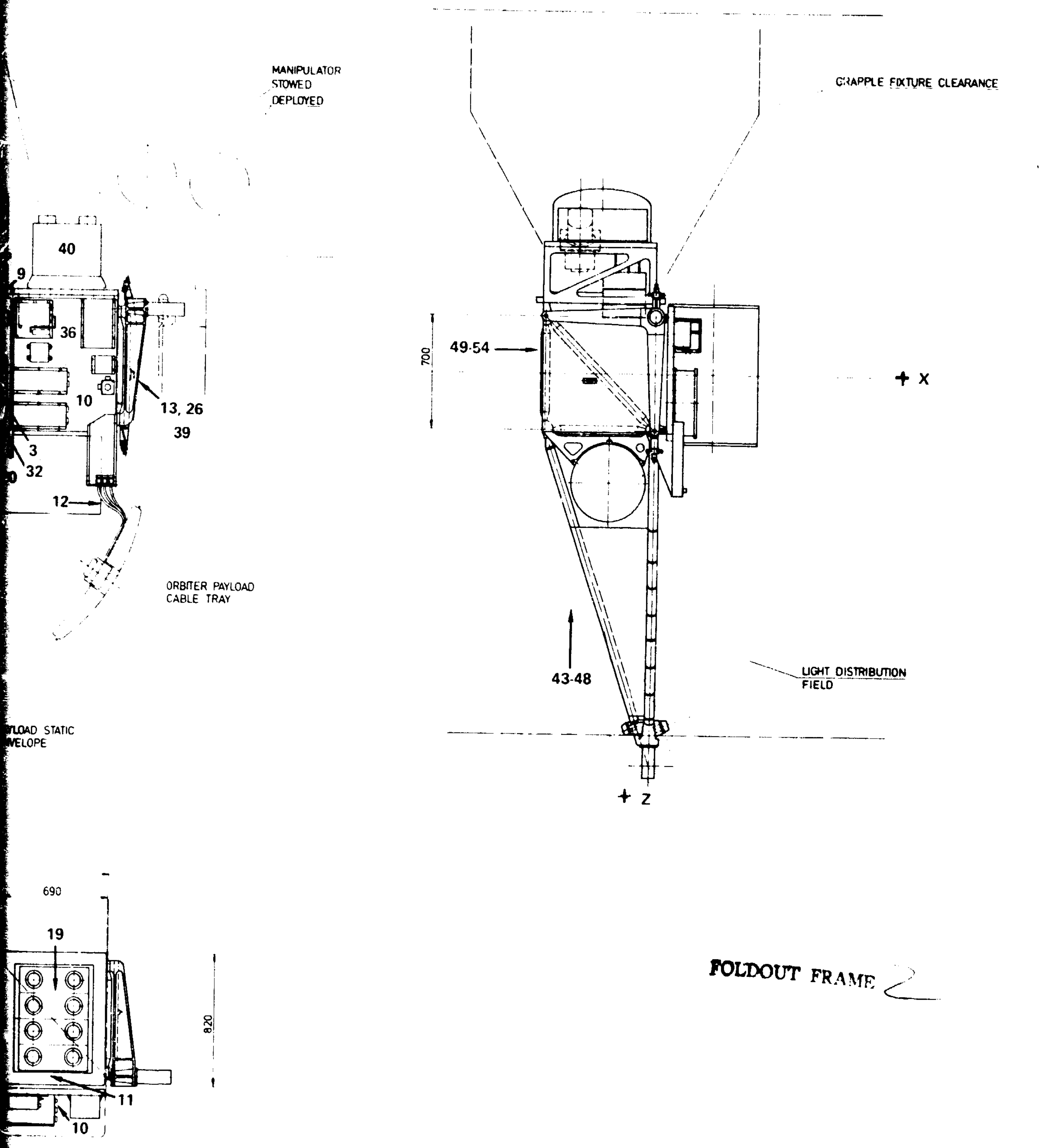


FIGURE 4-5. SPAS ISOMETRIC VIEW



FOLDOUT FRAME





FOLDOUT FRAME 2

Figure 4-6. Drawing of SPAS Spacecraft.

4.3.3 Point-Scatterer Model

Table 4-1 lists the point scatterers that comprise the SPAS model. The angular region of applicability for each scatterer is indicated by the ϕ_x, ϕ_y, and ϕ_z columns and these entries provide an approximate inclusion of shadowing effects. Notes in calculating the cross sections are included as appendix E.

Scatterers 1 through 34 reflect geometries discussed previously. Specular flares due to plates have been limited to $700\text{-}1200\text{m}^2$ to reflect the fact that these surfaces are not usually good enough to provide the several thousand square meters predicted theoretically. Scatterers 35 through 54 are intended to model complex areas. The cross section for each area can only be guessed. The rationale for our guess is as follows. The area of each complex surface is about .5 m^2. Taking a rough surface model (Models 9A4, 9A5 of Ref. 16, p.678) yields

$$\sigma = A\sigma_o$$

$$A = 0.5\ \text{m}^2$$

$$\sigma_o = \eta \cos \phi_I$$

η = Backscatter coefficient

ϕ_I = incidence angle (angle from LOS to surface normal)

The constant η has been determined experimentally for terrain and ranges from -30 to -15 dB for vegetation and ranges up to +10 or +20 dB for cultural areas. We take η = -10 dB to obtain $\sigma = 0.05\text{m}^2$ at normal incidence and allow the RCS to fall off as the cosine of the incidence angle. This value should be randomized to avoid interference effects.

At ranges for which the radar beam encompasses the target, modeling these areas as points still allows the radar's range and angle trackers to wonder over the target since variation in the relative phasing among scattering areas,

TABLE 4-1 SPAS POINT SCATTERER MODEL

Feature		σ, m^2	X_k, m	Y_k, m	Z_k, m	r_k, m	ϕ_x, deg.	ϕ_y, deg.	ϕ_z, deg.
XY Plane									
Viewed from ± X									
Tank	1	2.6	.24	-.83	.15	-.1	<90	50-150	90 ±4.1
	2	2.6	.24	-1.05	.15	-.1	<90	35-150	90 ±4.1
	3	2.6	.24	-1.27	.15	-.1	<90	30-155	90 ±4.1
Cannister	4	61	.37	1.05	0	-.29	<90 90-180	0-145 73-155	90 ±1.5
	5	61	.37	.35	0	-.29	<90 90-180	25-155 47-155	90 ±1.5
	6	61	.37	- .35	0	-.29	<90 90-180	25-180 32-135	90 ±1.5
Dome	7	25.7	-.35	-1.05	-.8	-.315	—	-145-145	90 ±2.3
Plates	8	13 1100	.12	1.9	0	0	< 2.1	-	90 ±1.5
	9	13 900	.12	-1.05	0	0	< 2.1	-	90 ±1.5
	10	13 1000	.12	-1.8	0	0	< 2.1	-	90 ±1.5
Viewed from +Y									
Cyl. End	11	850	-.3	2.0	-.67	0	—	0 ±2.6	90 ±2.6
Viewed from -Y									
Cyl. End	12	1200	-.3	-2.0	-.67	0	—	180±2.6	90 ±2.6
Comm Ant	13	3322	-.35	-2.0	0	0	—	180±2.5	90 ±2.5
XZ Plane									
Cylinder	14	1117	-.3	0	+.67	.24	—	90±.3	0-125
Plates	15	700	-.35	1.7	-.48	0	90±1.5	90±1.5	0±1.5,180±1.5
	16	800	-.35	1.05	-.48	0	90±1.5	90±1.5	0±1.5,180±1.5
	17	1000	-.35	.35	-.48	0	90±1.5	90±1.5	0±1.5,180±1.5
	18	900	-.35	-.35	-.48	0	90±1.5	90±1.5	0±1.5,180±1.5
	19	850	-.35	-1.75	-.48	0	90±1.5	90±1.5	0±1.5,180±1.5
Cannister Face	20	750	.37	1.05	.425	0	90±2	90±2	0±2
	21	850	.37	1.05	-.425	0	90±2	90±2	180 ± 2
	22	850	.37	.35	.425	0	90±2	90±2	0 ± 2
	23	750	.37	.35	-.425	0	90±2	90±2	180 ± 2
	24	920	.37	-.35	.425	0	90±2	90±2	0 ± 2
	25	730	.37	-.35	-.425	0	90±2	90±2	180 ± 2

TABLE 4-1 SPAS POINT SCATTERER MODEL (Continued)

Wide-Ange Scatterers								
Reflector								
26	0.2	-.35	-2.15	0	0	—	180±26.6	90 ±26.6
Tank Hemispheres								
27	.03	.24	-.83	-.02	-0.1	0-90	—	90 - 180
28	.03	.24	-.83	+.3	-0.1	0-90	—	0 - 90
29	.03	.24	-1.05	-.02	-0.1	0-90	—	90 - 180
30	.03	.24	-1.05	+.3	-0.1	0-90	—	0 - 90
31	.03	.24	-1.27	-.02	-0.1	0-90	—	90 - 180
32	.03	.24	-1.27	+.3	-0.1	0-90	—	0 - 90
Dome 33	1.25	-.35	-1.05	-.8	-.35	—	—	156 - 180
34	0.17	-.35	-1.05	-.8	-.35	—	—	90 - 156
Cold Gas Panel 35	$\sigma_{35} \cos \phi_x$	.12	-1.05	0	0	0-90	—	—
Data Handling Panel 36	$\sigma_{36} \cos \phi_x$	.12	-1.75	0	0	0-90	—	—
Power Panel 37	$\sigma_{37} \cos \phi_x$	.12	+1.75	0	0	0-90	—	—
+Y Sill Fitting Assy. 38	$\sigma_{38} \cos\phi_y$	-.35	+2.15	0	0	—	0-90	—
Ant Assy 39	$\sigma_{39}\cos (\phi_y-180)$	-.35	-2.15	0	0	—	90-180	—
MOMS Module 40	$\sigma_{40}\cos(\phi_z-180)$	-.35	-1.75	-.9	0	—	—	90 -- 180
RMS Grapple Fixture 41	$\sigma_{41}\cos(\phi_z-180)$	-.35	.24	-.5	0	—	—	90 - 180
RITA-D 42	$\sigma_{42}\cos(\phi_z-180)$	-.35	+1.75	-.7	0	—	—	90 - 180
Structure								
+Z 43	$\sigma_{43}\cos \phi_z$	-.35	1.75	0	0	—	—	0 - 90
+Z 44	$\sigma_{44}\cos \phi_z$	-.35	1.05	0	0	—	—	0 - 90
+Z 45	$\sigma_{45}\cos \phi_z$	-.35	.35	0	0	—	—	0 - 90
+Z 46	$\sigma_{46}\cos \phi_z$	-.35	-.35	0	0	—	—	0 - 90
+Z 47	$\sigma_{47}\cos \phi_z$	-.35	-1.05	0	0	—	—	0 - 90
+Z 48	$\sigma_{48}\cos \phi_z$	-.35	-1.75	0	0	—	—	0 - 90
-X 49	$\sigma_{49}\cos(\phi_x-180)$	-.35	1.75	0	0	90-180	—	—
-X 50	$\sigma_{50}\cos(\phi_x-180)$	-.35	1.05	0	0	90-180	—	—
-X 51	$\sigma_{51}\cos(\phi_x-180)$	-.35	.35	0	0	90-180	—	—
-X 52	$\sigma_{52}\cos(\phi_x-180)$	-.35	-.35	0	0	90-180	—	—
-X 53	$\sigma_{53}\cos(\phi_x-180)$	-.35	1.05	0	0	90-180	—	—
-X 54	$\sigma_{54}\cos(\phi_x-180)$	-.35	1.75	0	0	90-180	—	—

thaL causes the wander, is modeled by the physical separation of multiple scattering areas.

At short ranges, the radar beam may encompass only one of these areas, and thus the wander effect will not be observed. Appendix F develops a simple model for wander that adds a "wander vector" to the scattering points given for the complex areas.

4.3.4 Effect of Thermal Blanket

Several, if not most, of the spacecraft will be wrapped by multi-layer insulation. The RF properties of this material are not known at present. If it is effectively conductive, it will tend to reduce flares and promote diffuse returns. The effects are almost impossible to predict analytically and measurements would be very desirable.

4.3.5 Recommendation

The validation of an analytical model of as complex an object as a spacecraft requires measurements. It would be very desirable if

a. Data can be taken with the Ku-Band system tracking a spacecraft - like target in the planned White Sands tests.

b. The RCS of a SPAS mockup could be measured with and without thermal blankets.

4.4 MATHEMATICAL DESCRIPTION OF TARGET RETURN SIGNAL

If we assume a single pulse was transmitted, then the expression for the noise-free return signal from a single point scatterer at the antenna sum (difference) channel output is given by

$$S_k(t) = \sigma_k^{\frac{1}{2}} \rho_k A_k \cos\left[2\pi(f_c+f_k)\ (t-t_k)\right] p(\frac{t-t_k}{t_t}) \qquad (4.1)$$

where $A_k = \left(\frac{R_o}{R_k}\right)^4 C_o$

σ_k = RCS of k th scatterer,

$$C_o = \left[\frac{P_T G_o^{\,2}\ \lambda_c^{\,2}}{(4\,\pi^{\,3} R_o^{\,4} L_T}\right]^{1/2}$$

R_k = Range of k th scatterer

R_o = Range of target c.g.,

ρ_k = antenna sum (difference) pattern weighting normalized to the peak gain,

L_T = transmit losses,

P_T = Peak transmit power,

G_o = Peak one-way antenna gain,

λ_c = wavelength of carrier frequency,

f_c = carrier frequency,

$f_k = -\frac{2v_k}{c}$ = doppler shift of k th scatterer,

$t_k = \frac{2\left[R_k - R_0\right]}{c}$ = delay of target return relative to the target c.g. return,

c = speed of light

$$\rho\left(\frac{t}{t_t}\right) = \begin{cases} 1, & 0 \le t \le t_t \\ 0, & \text{otherwise,} \end{cases}$$

t_t = transmit pulsewidth

Then, assuming the antenna is linear, by applying the principle of linear superposition the resultant return signal for the entire collection of point scatterers at the sum (difference) channel output terminal can be written as

(4.1a) $$S\ (t) = \sum_{k=1}^{N} S_k(t)$$

where the target is composed of N point scatterers. A nice feature of the present target model hidden in equations (4.1) and (4.1a) is that this model easily handles the spatial integration of the return signal performed by the antenna.

In the rest of this subsection, additional details of the antenna weighting factor and scatterer phase computation models are given. Computation models for the other terms in equation (4.1) have either been explained earlier or the computation is clear from the definition of the term.

4.4.1 Antenna Weighting Factor Computation

Computation of the antenna sum and difference pattern weighting factors makes the assumption that the return signal from a single point target at the radar is a plane wave propagating from the direction of the scatterer. The sum and difference pattern weights can then easily be determined from the antenna sum and difference pattern models described below.

The sum pattern weighting is computed with the following expression

(4.2) $$\rho_s(\theta) = \frac{\sin x}{x} \text{ (sum pattern weighting)}$$

where $x = 93.80\ \theta,$

θ = target angle off boresight.

Figure 4-7 illustrates the pattern given by equation (4.2). This pattern has a 3 dB two-sided beamwidth of 1.7° and is assumed to be symmetric about the bore: t. For the k th target, the angle off boresight is computed with

(4.3) $$\theta_{ks} = \cos^{-1}\left(r_{kz}^{L} / \left|\vec{r}_k^{\,L}\right|\right) .$$

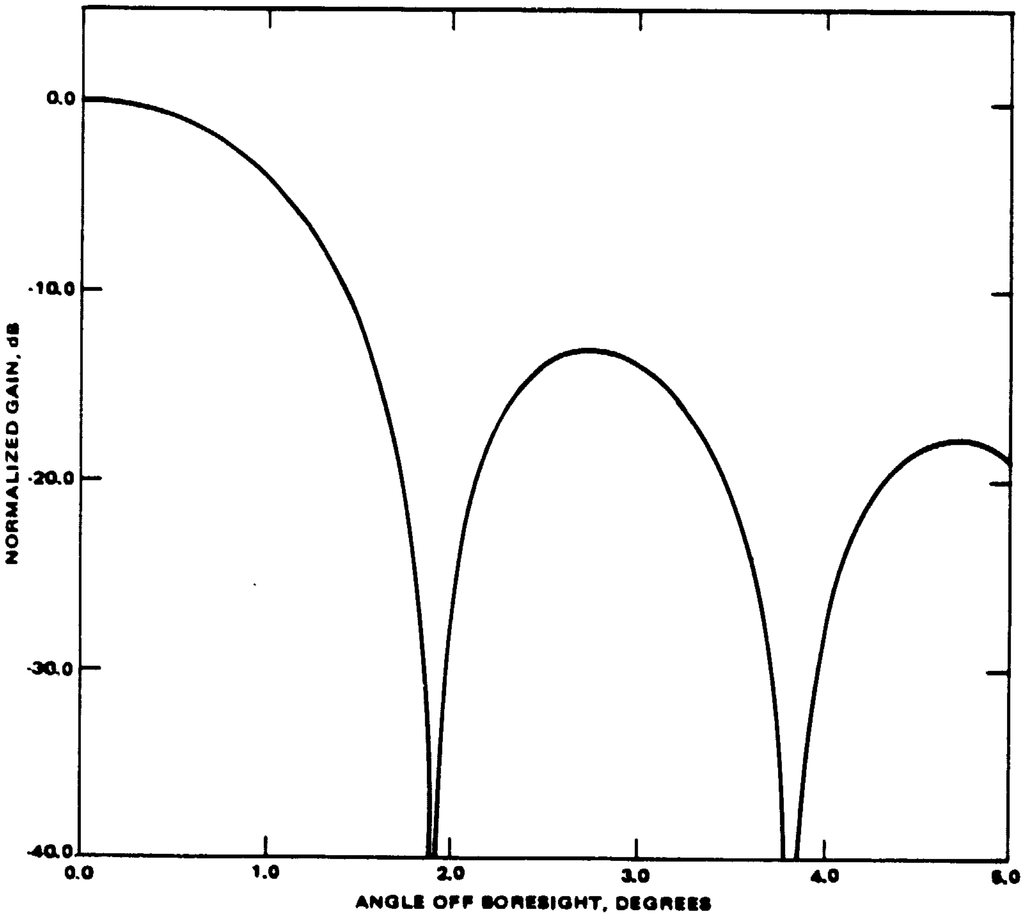


Figure 4-7. Antenna Sum Pattern.

The azimuth difference pattern weighting is computed from

$$\rho_{AZ} = 1.1465 \left[\frac{y \cos y - \sin y}{y2} \right] \quad \text{(difference pattern weighting)} \tag{4.4}$$

where $y = 93.8\Delta$. This pattern is assumed to be symmetric about the y-axis of the LOS frame and is illustrated in Figure 4-8. The angle Δ for the k th target is obtained from

$$\Delta = \theta_{kaz} = -\sin^{-1} (r^L_{ky}/|\vec{r}^L_k|). \tag{4.5}$$

The elevation difference pattern weighting is also computed using equation (4.4) only in this case the angle Δ is given by

$$\Delta = \theta_{kel} = \sin^{-1} (r^L_{kx} /|\vec{r}^L_k|). \tag{4.6}$$

and the pattern is assumed to be symmetric about the x-axis.

4.4.2 Computation of Scatterer Phase

From equation (4.1) the initial (t = o) phase associated with the k th scatterer is given by

$$\phi_k = -2\pi (f_c + f_k) t_k.$$

If we choose the time origin appropriately, then $f_k t_k < < 1$ for all k and as a result

$$\phi_k \doteq 2\pi f_c t_k .$$

For example, the time origin can be located at the center of the range gate or

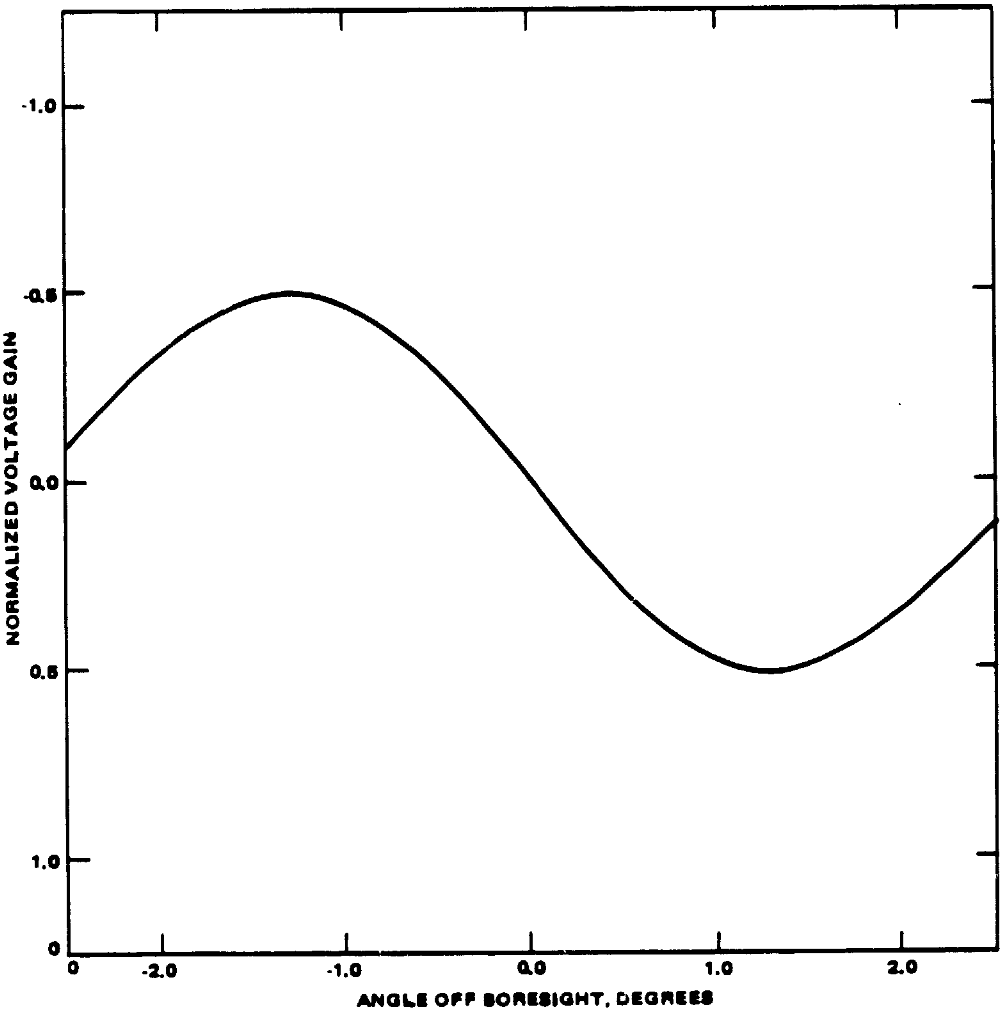


Figure 4-8. Antenna Difference Pattern.

at the leading edge of the return from the target c.g. which would give either

$$\phi_k = \frac{4\pi(R_k - R_g)}{\lambda_c}$$

or

$$\phi_k = \frac{4 \quad (R_k - R_o)}{c}$$

4.5 COMPUTER ALGORITHM DETAILS

Figure 4-9 illustrates the target scattering model computer algorithm. This algorithm computes the value of the RCS in the direction of the radar for each scatterer and the location of the scattering center for each scatterer. Using the modeling description given in sections 4.1 through 4.3 and adhering to the real-time computation constraint, the algorithm was structured as follows:

(1) determine all scatterers with nonzero RCS in the direction of the radar,

(2) determine the specular point location for these scatterers where the geometric optics approximation applies,

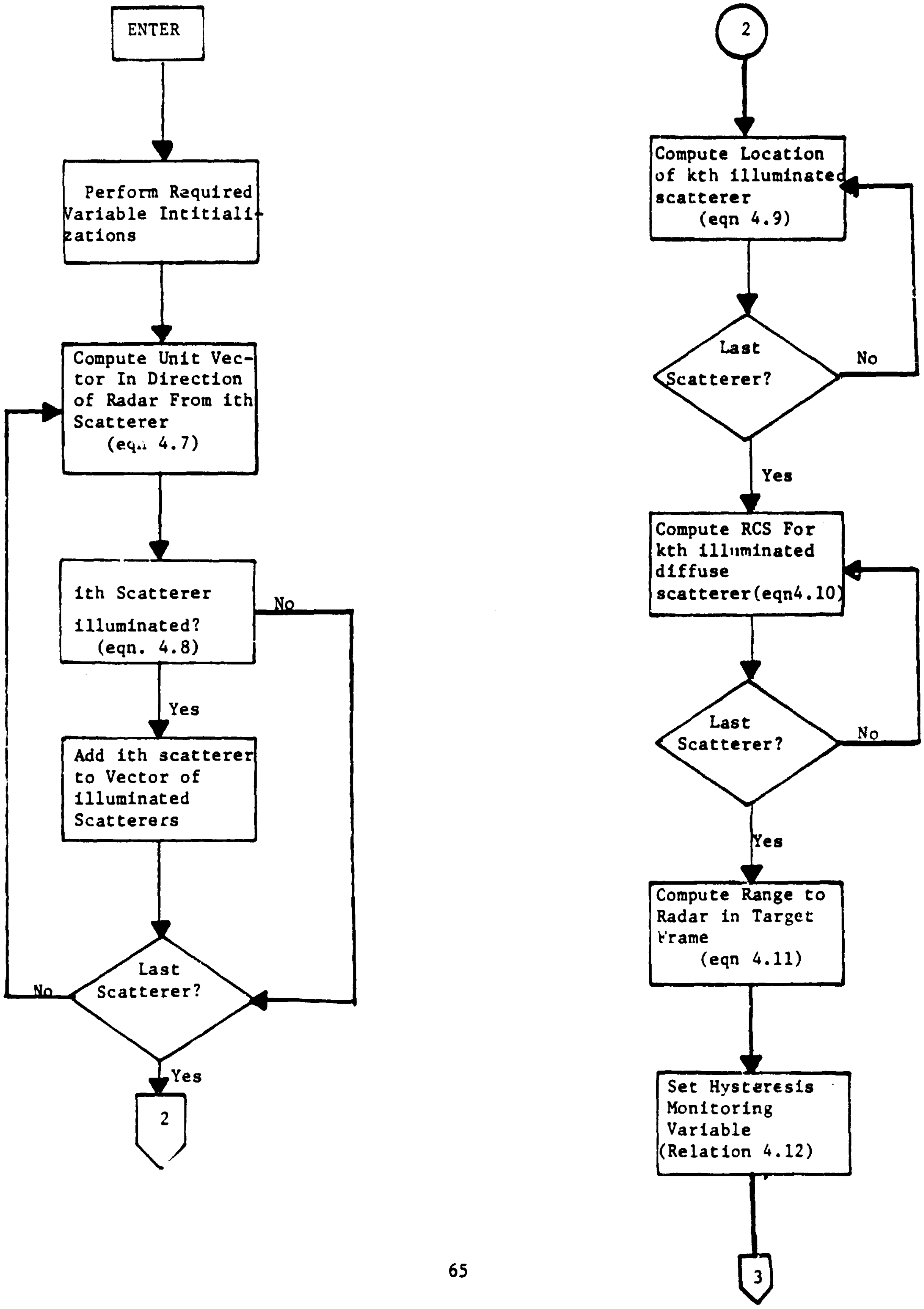
(3) compute the RCS for all rough surface scattering areas that are illuminated,

(4) if at close range, determine the scattering center for the rough surface (or diffuse) scattering area using the method presented in Appendix F.

Details of each of these steps are given in the remainder of this subsection.

The purpose of the first step is to weed out all of those scattering areas which are not illuminated or have, for all practical purposes, no RCS in the direction of the radar. Towards this end, we first compute the direction to the radar from each of the scattering centers using the expression

FIGURE 4-9 TARGET MODEL COMPUTER ALGORITHM (1 of 3)



ENTER
Perform Required Variable Intitializations
Compute Unit Vector In Direction of Radar From ith Scatterer (eqn 4.7)
ith Scatterer illuminated? (eqn. 4.8)
No
Yes
Add ith scatterer to Vector of illuminated Scatterers
Last Scatterer?
No
Yes
2
2
Compute Location of kth illuminated scatterer (eqn 4.9)
Last Scatterer?
No
Yes
Compute RCS For kth illuminated diffuse scatterer(eqn4.10)
Last Scatterer?
No
Yes
Compute Range to Radar in Target Frame (eqn 4.11)
Set Hysteresis Monitoring Variable (Relation 4.12)
3

FIGURE 4-9 TARGET MODEL COMPUTER ALGORITHM (2 of 3)

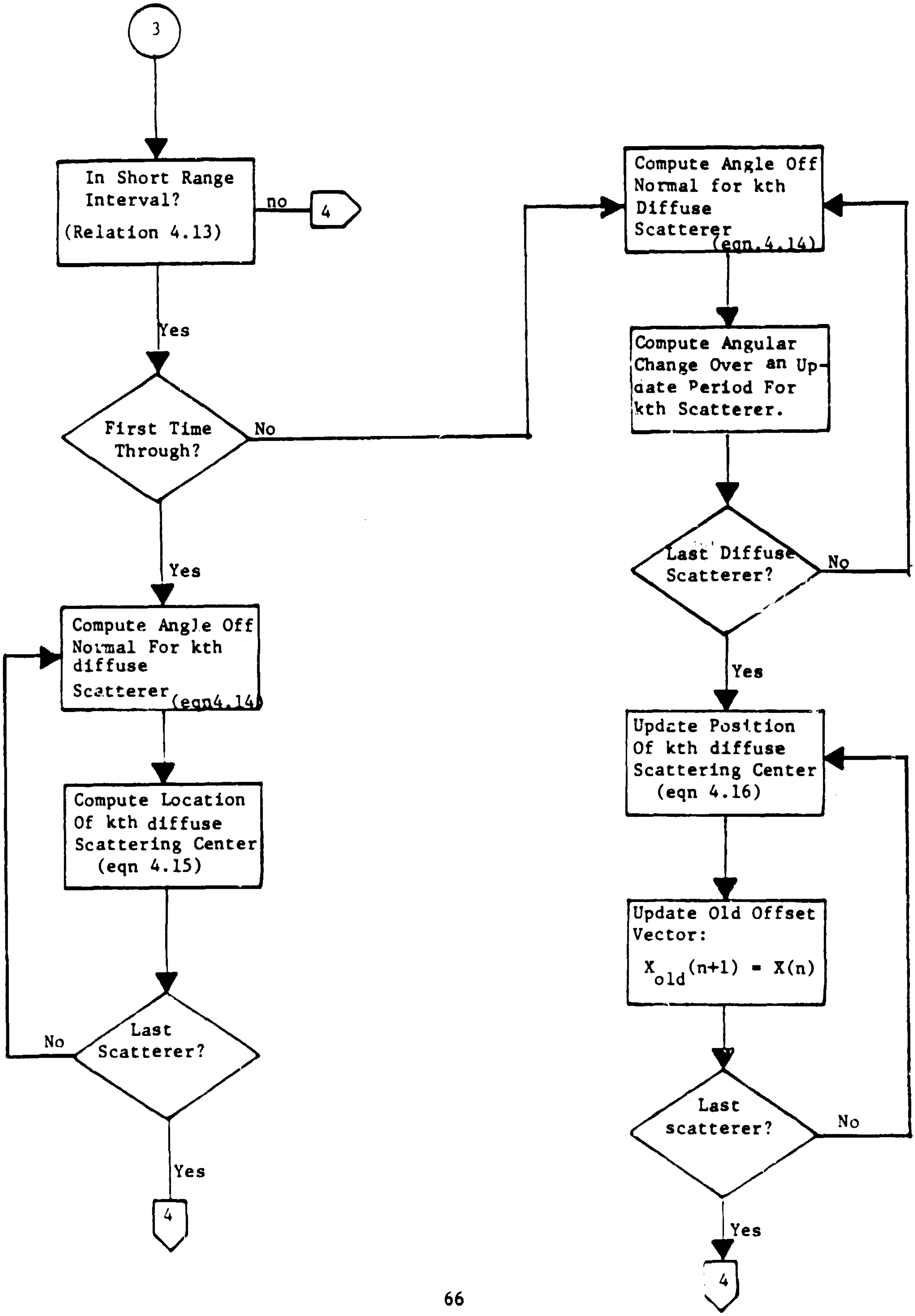
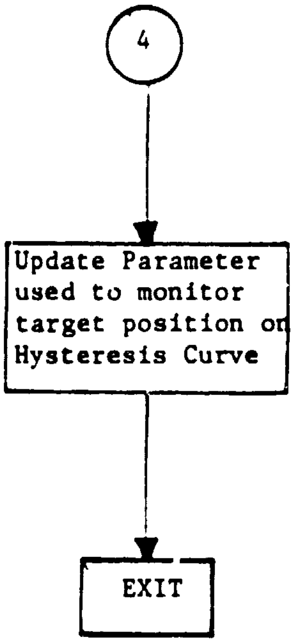

3
In Short Range Interval? (Relation 4.13)
no
4
Yes
First Time Through?
No
Yes
Compute Angle Off Normal For kth diffuse Scatterer (eqn4.14)
Compute Location Of kth diffuse Scattering Center (eqn 4.15)
Last Scatterer?
No
Yes
4
Compute Angle Off Normal for kth Diffuse Scatterer (eqn.4.14)
Compute Angular Change Over an Update Period For kth Scatterer.
Last Diffuse Scatterer?
No
Yes
Update Position Of kth diffuse Scattering Center (eqn 4.16)
Update Old Offset Vector:
$X_{old}(n+1) = X(n)$
Last scatterer?
No
Yes
4

FIGURE 4-9 TARGET MODEL COMPUTER ALGORITHM (3 of 3)


4
Update Parameter used to monitor target position on Hysteresis Curve
EXIT

$$\hat{u}_k^T = (\vec{r}_k^T - \vec{x}_R^T) / |\vec{r}_k^T - \vec{x}_R^T| \tag{4.7}$$

where $\vec{x}_R^T$ = location of the radar in the target coordinate system.

The kth scatterer is declared to have a nonzero RCS in the direction of the radar if the components of the direction vector $\hat{u}_k^T$ satisfy the following inequalities

$$m_{ki} \leq u_{ki}^T \leq M_{ki} \qquad i = x, y, z \tag{4.8}$$

where the m_{ki}'s and the M_{ki}'s are determined using the appropriate method outlined previously.

Step (2) of the algorithm is to determine the location of the specular point (or scattering center) for those scatterers where the geometric optics approximation applies and with nonzero RCS in the direction of the radar. For the SPAS scattering model, all of the specular scatterers have circular or spherical symmetry. In these cases, the specular point can easily be calculated from the simple expression

$$\vec{s}_k^T = \vec{r}_k^T + a_k\hat{u}_k^T \qquad \text{for all } k \tag{4.9}$$

where $\vec{r}_k^T$ = location of the centroid of the simple shape in the target frame,

a_k = represents the appropriate radius for the kth scatterer.

It is remarked that for those scatterers where specular reflection does not apply, the a_k are set equl to zero.

The third step is to compute the RCS for all scatterers which were found in step (1) to have a nonzero RCS in the direction of the radar. Scatterers representing simple geometric shapes require no work since the model for these scatterers are assumed to have a contant RCS over the region where its theoretical RCS is significant and zero where it is not significant. However, the rough-surface scatterers require some calculation to obtain the proper RCS value. In section 4.3, this calculation was given as

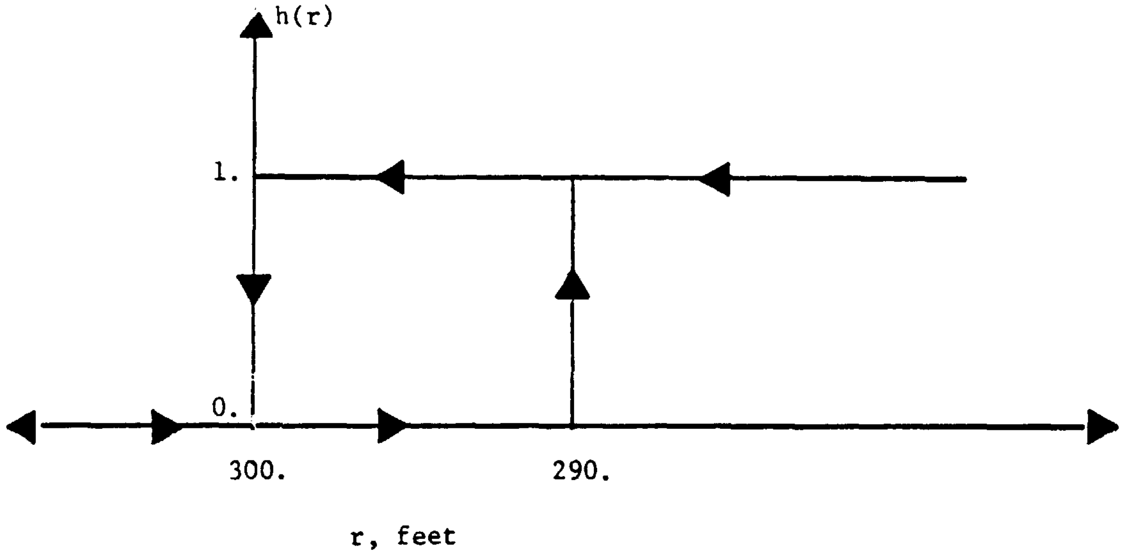
$$\sigma_k = \eta_k \cos \phi_{ki} \tag{4.10}$$

where η_k = backscatter coefficient for the kth scatterer,

$\cos\phi_{ki} = \hat{u}_k^T \cdot \eta_k^T$,

η_k^T = normal to the kth rough surface scatterer.

The fourth and final step is to determine whether the target is at close range (defined below) and, if it is at close range, to compute the position of the rough surface scattering center using the method of Appendix F. The idea is that one wants to avoid using a nonfluctuating scattering model when the target is close enough so that only one (rough surface) scatterer occupies the full 3 dB antenna beamwidth. Since all rough surface scatterers in the SPAS model have the same dimensions of 2.3 feet by 2.3 feet and the 3 dB beamwidth is taken to be 1.68 degrees, the criterion for closeness is easily computed to be a range of 78 feet. As an added measure of safety, the boundary for close range was established as approximately 300 feet. Also a hysteresis loop (shown below) is used so that the close range model is not swapped in



and out rapidly when the target range is jittering about the close range boundary.

To determine whether the short range model applies we first compute the range to the radar in the target frame using

$$r_R^T(n) = |\vec{x}_R^T\ (n)| \tag{4.11}$$

Next, the output of the hysteresis loop for the present update period is obtained from the following relations:

$$\begin{aligned} r_R^T(n) &\leq 290 \rightarrow h(n) = 1 \\ r_R^T(n) &\geq 300 \rightarrow h(n) = 0 \end{aligned} \tag{4.12}$$

$$290 < r_R^T(n) < 300 \text{ and } h(n-1) = 1 \rightarrow h\ (n) = 1$$

$$290 < r_R^T(n) < 300 \text{ and } h(n-1) = 0 \rightarrow h\ (n) = 0 .$$

The short range model is invoked if

$$h(n) = 1. \tag{4.13}$$

If it has been determined from the above procedure that the short range model should be used, the computation of the "wander" in the rough surface scattering center is performed in the following manner. (To facilitate the explanation it is assumed that the normal to the kth rough surface scatterer is parallel to the z-axis of the target frame.) First, the incidence angle is computed with the expression

$$\phi_{ki}(n) = \cos^{-1}\left(u_k^T(n) \cdot \hat{z}^T\right) \tag{4.14}$$

and is then used in the update of the components of the wander vector as follows

$$x_k(n) = \alpha(n)\, x_k(n-1) + \sigma_o \left[1- \alpha^2(n)\right]^{\frac{1}{2}} u\left[-\tfrac{1}{2},\tfrac{1}{2}\right] \tag{4.15}$$

where

$$\alpha(n) = \exp\left[\frac{2D_x\, \delta\phi_{ki}(n) \cos \phi_{ki}(n)}{\lambda}\right]$$

$$\delta\phi_{ki}(n) = \phi_{ki}(n) - \phi_{ki}(n-1),$$

D_x = length of the x-dimension of the rough surface scatterer,

$$\alpha_o^2 = D_x^2 / (12\, N_F)$$

and $u\left[-\frac{1}{2}, \frac{1}{2}\right]$ represents a selection from a population which is uniformly distributed over the interval $\left[-\frac{1}{2}, \frac{1}{2}\right]$. The y-component of the wander vector is obtained by replacing all x's by y's in equation (4.15).

The only detail that remains is the intialization of the difference equation given in (4.15), i.e. determining the value of $x_k(o)$ and $y_k(o)$, when the close range model is first invoked. This is accomplished by choosing the $x_k(o)$ and $y_k(o)$ from a random population with the appropriate statistics. Quantitatively, we have

$$x_k(o) = \sigma_o\, u\left[-\tfrac{1}{2}, \tfrac{1}{2}\right]. \tag{4.16}$$

5. SEARCH AND ACQUISITION MODE COMPUTER MODEL DESCRIPTION

As stated in the introduction, the search and acquisition mode performance computer model is provided for the purpose of crew training only which dictates the following design objectives:

(1) to provide a real-time simulation,

(2) to provide accurate timing of discrete events appearing on the cockpit radar display,

(3) to provide accurate operation of cockpit radar display meters,

(4) to provide accurate responses to all cockpit radar controls.

Since the model is not required for critical engineering evaluation of the Ku-Band Radar search mode performance, the design objectives above can be met while providing only a representative model of the target and detection processor.

Figure 5-1 illustrates the basic structure of the search and acquisition computer model. This model consists of a main control program and three major subprograms dedicated to (1) the gimbal pointing loop model, (2) the spiral scan model, and (3) the target detection model. The functions of the main program are to decide which antenna steering mode has been requested and then update the search sequence for that steering mode. Updating the search sequence requires a check of internal and external controls to determine which of the three models listed above should be invoked. In the remainder of this section the details of the main algorithm and the point, scan, and detection models will be presented.

Before launching a detailed description of the algorithm, we must state a fundamental assumption that was made in the development of the search and acquisition mode computer model: all of the acquisition mode logic was ignored since it is transparent to the crew. Impact of this assumption is to introduce some error into discrete event timing under certain conditions. For example, neglecting the mini-scan will cause a noticeable timing error.

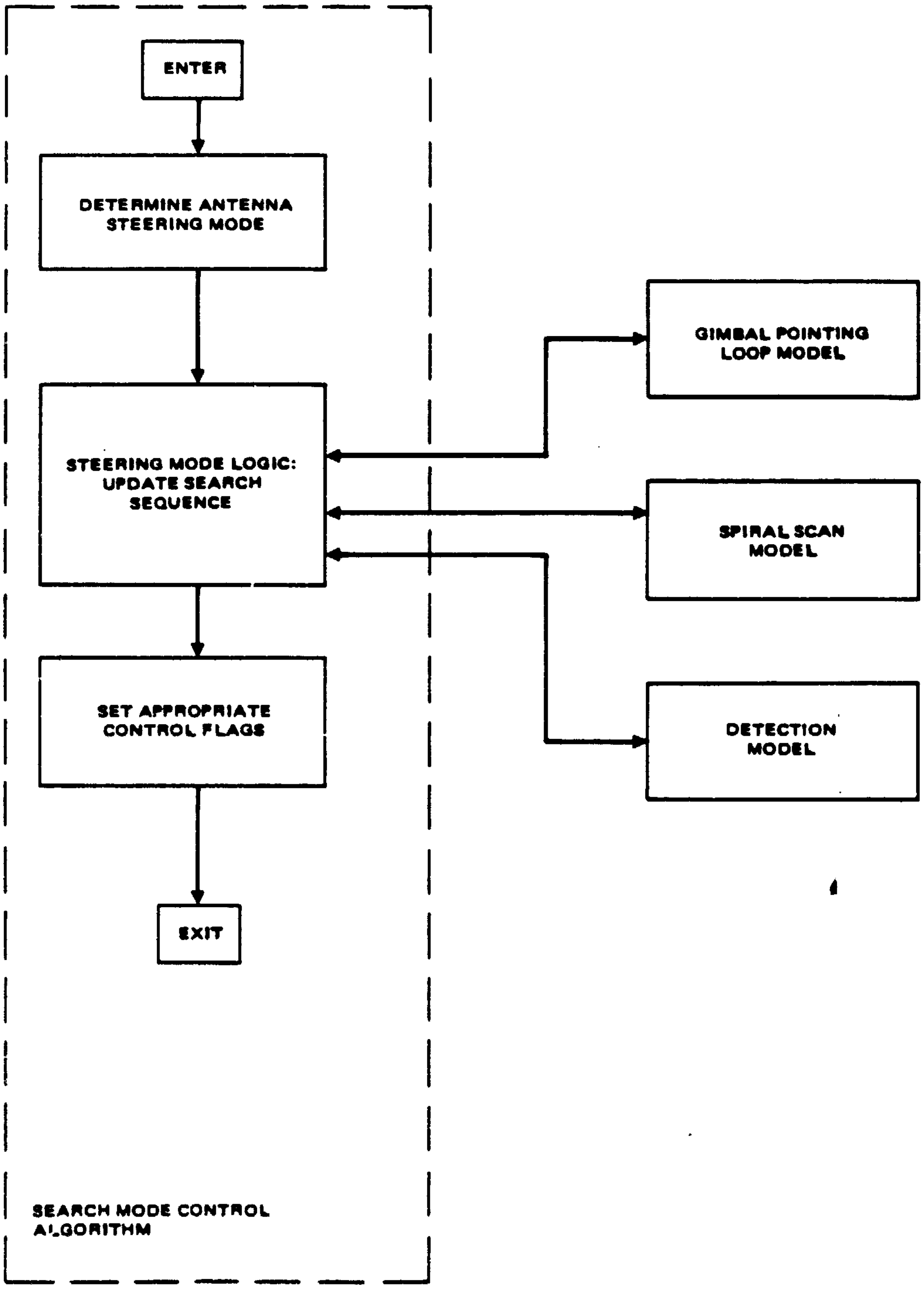

ENTER
DETERMINE ANTENNA STEERING MODE
STEERING MODE LOGIC: UPDATE SEARCH SEQUENCE
SET APPROPRIATE CONTROL FLAGS
EXIT
GIMBAL POINTING LOOP MODEL
SPIRAL SCAN MODEL
DETECTION MODEL
SEARCH MODE CONTROL ALGORITHM

Figure 5-1. Outline of search and acquisition mode computer algorithm.

5.1 SUMMARY OF KU-BAND RADAR SEARCH MODE OPERATION

5.1.1 General Antenna Steering Mode Operation

This subsection provides a brief description of the Ku-Band Radar search mode procedure for each antenna steering mode. For a given antenna steering mode, the general procedure is the same for active and passive targets; the only difference between active and passive are in the waveforms and processing as discussed in the sequel.

GPC-ACQ Search and Acquisition Mode. In this mode, the radar accepts angle designates from the GPC. The antenna then slews towards these designated angles and attempts detection once inside zone 0 (within 3° of the designated angles). If the antenna moves into zone I (within 0.3° of the designated angles) without a detection and the search initiate is low, the antenna stops at the designates, awaits new angle designates or a search initiate from the GPC, and still attempts detection. If the antenna is in zone I and the search initiate command has been given, the antenna begins scanning using a spiral pattern, centered at the inertially held target angle designates. The scan will last for 60 seconds or until a target has been detected, which ever comes first. If a detection does not occur, the antenna returns to the designated angles and awaits new designates or another search initiate command. If a detection occurs the system progresses to the acquisition mode where a mini-scan (if required) and a sidelobe avoidance test are performed. Depending upon the outcome, the system proceeds to the rack mode or returns to the search mode. Details of the acquisition mode are deliberately sketchy because this mode is not modeled as noted earlier.

GPC-DES Search and Acquisition Mode. Search operation in this antenna steering mode is identical to the GPC-ACQ mode minus the spiral scan capability. That is, the antenna only moves if it receives new angle designates from the GPC. Rules for when target detection is allowed are the same as GPC-ACQ and the waveforms and processing for active and passive operation are identical.

Auto Search and Acquisition Mode. In this mode the crew moves the antenna to the desired position using the antenna slew switches on the radar console. Using these slew switches, the antenna can be slewed up or down and left or right at either 20 degrees per second or 0.4 degrees per second. When the antenna is being manually slewed, target detection is only allowed if the slew rate is less than or equal to 0.4 degrees per second. Once the antenna has been slewed to the desired position and no target detection has occurred, the crew can initiate a spiral scan search. After a scan is initiated, the antenna will continue to spiral outwardly for one minute (to 30^{o} off the body-stabilized scan center) or until a target is detected whichever comes first. If a target is not detected then the antenna returns directly to the scan center and awaits either a slew command or another search initiate command. If a target is detected the system proceeds to the acquisition mode.

Manual Search and Acquisition Mode. The manual search mode is identical to the Auto search mode minus the spiral scan capability. That is, the antenna position can only be changed via the slew switches on the radar console and target detection is only allowed if the commanded antenna slew rate is less than 0.4 degrees per second. The transmit waveforms and signal processing for this mode are identical to the Auto mode. Manual control of the antenna is also maintained during the acquisition and tracking phases.

5.1.2 Display Meters

The only meters that are operational during the search and acquisition mode are the roll and pitch angle meters. These meters monitor the antenna position during search and acquisition. All other meters, including the signal strength meter, are zeroed during this phase.

5.1.3 Search Mode Waveforms and Signal Processing.

Two types of detectors are used in the search mode: a single-hit detector shown in Figure 5-2 and a constant false alarm rate (CFAR) detector shown in

Figure 5-2   KU-BAND RADAR SINGLE-HIT DETECTOR

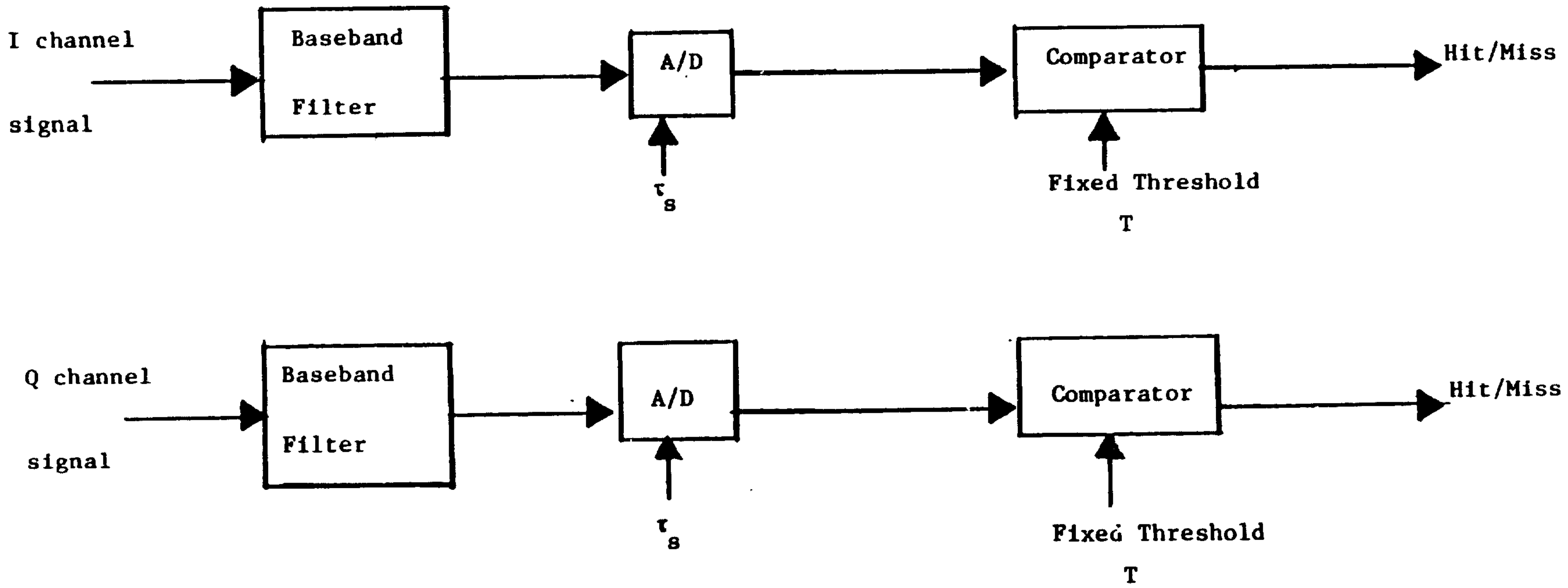


Figure 5-3. The situations where these two detectors are used are summarized below.

Passive GPC Modes. The passive GPC modes use a single hit detector when the designated range is less than 0.42 nm, and the CFAR detector when the designated range is greater than 0.42 nm. In single-hit detection, returns from the first 3000 feet are processed through the hit detector. In CFAR detection two overlapped range gates centered at the target range designate are used to obtain a detection. (We note that the range gates are of width 3/2 t_t and overlapped by t_t where t_t is the transmit pulse width). Figure 5-4 gives the general waveform used for all designated ranges and Table 5-1 summarizes the waveform and processor parameters used at each designated range.

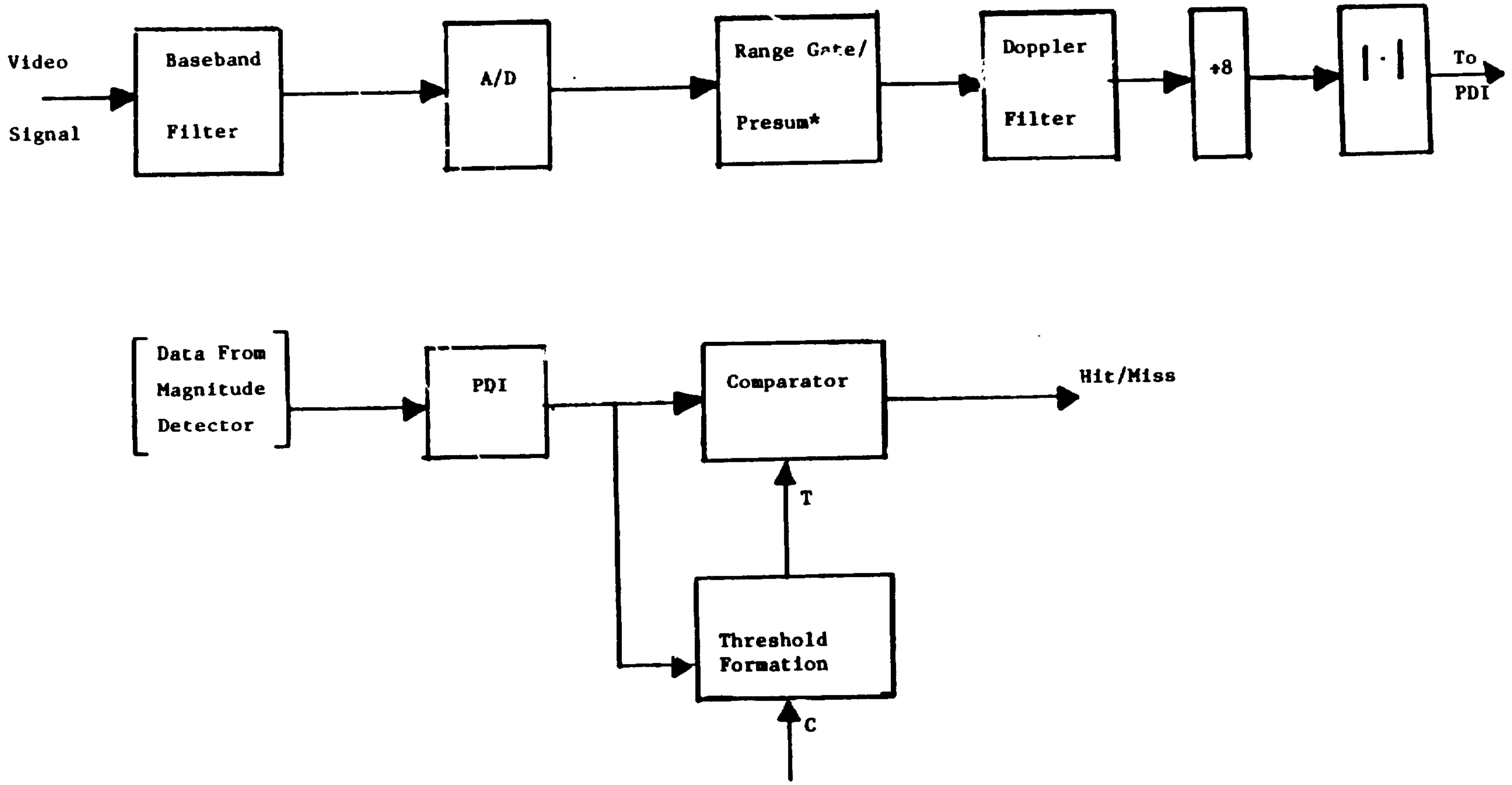
Passive Auto and Manual Modes. These modes use the relatively complex waveform shown in Figure 5-5. As noted in the figure, this waveform requires both types of detectors during an update period. That is, for a given transmit frequency the first three pulses are processed through the hit detector and the last 16 pulses are used in the CFAR detection process. In single-' detection, returns from the first 3000 feet are processed through the hit detector. In CFAR detection, four juxtaposed range gates, of width t_t and covering the interpulse period are used to obtain a target detection. Table 5-2 gives the waveform and processing parameters for these modes.

All Active Modes. Single-hit detection is employed in all active search modes. Only one transmit frequency is used, the PRF is fixed at 268 Hz, the transmit pulsewidth is 4.15 microseconds, and the sample interval is 2.075 microseconds under all conditions in the active mode. Target returns from up to 300 nm are processed through the single-hit detector. Also it is noted that the target range designate is ignored in the GPC active search modes.

5.1.4 Antenna Scan Operation

GPC-ACQ Passive or Active Modes. In this mode, the scan can only be

Figure 5-3 KU-BAND RADAR CFAR DETECTOR



* NOTE: Range gates are overlapped in GPC modes and are juxtaposed in Auto and Manual Modes.

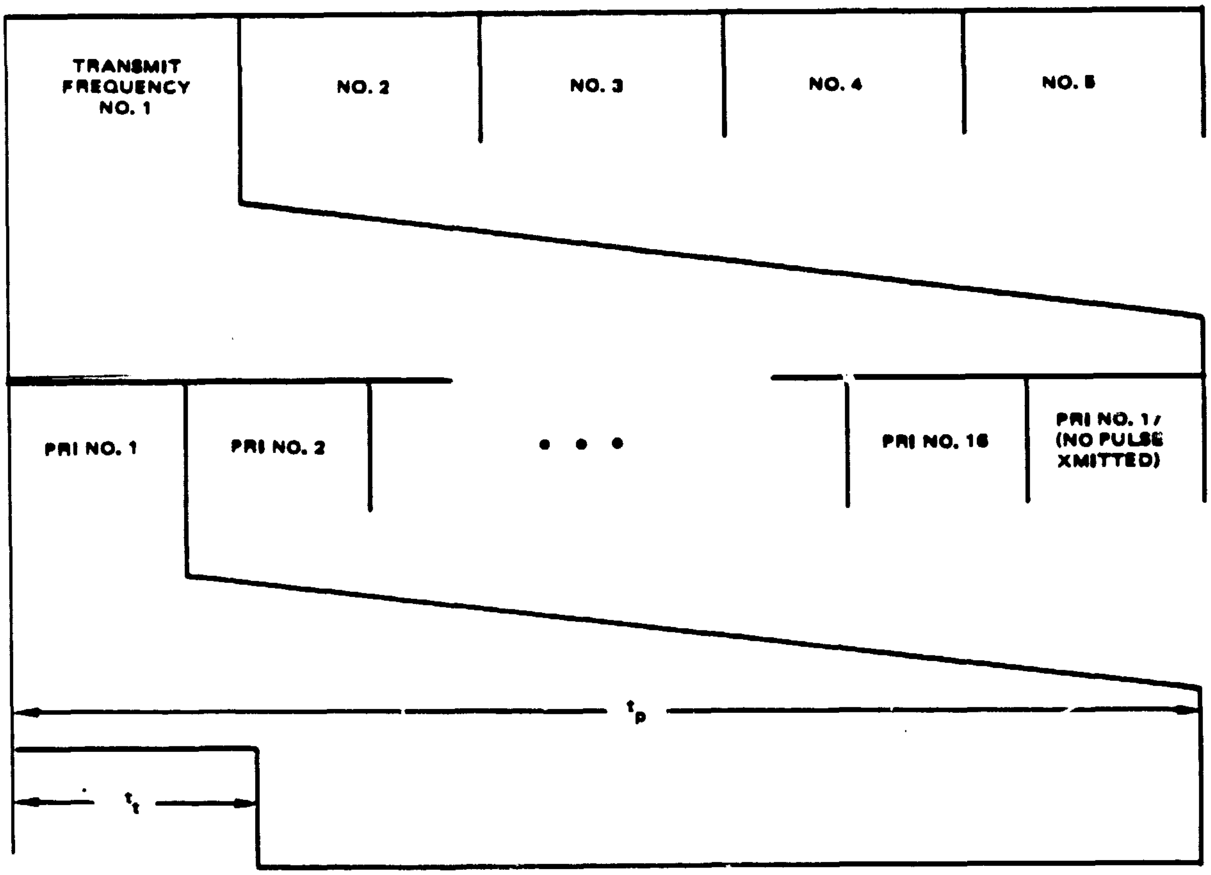


Figure 5-4. Passive GPC Search Mode Waveform.

TABLE 5-1 PARAMETERS FOR GPC PASSIVE SEARCH MODES

Range Interval NM	Frequency,* GHz	PRF,* Hz	Pulsewidth, μsec	Number of Range Bins	Range Bin Type	Sample Interval, μsec	Number of Samples Per Pulse	Detector Type
0 to 0.42	13.883	6969.7	.122	50	Adjacent	.122	1	Single-Hit
0.42 to 0.95	13.883	6969.7	4.15	2	Overlapped	2.075	2	CFAR
0.95 to 1.9	13.883	6969.7	8.3	2	Overlapped	2.075	4	CFAR
1.9 to 3.8	13.883	6969.7	16.6	2	Overlapped	2.075	8	CFAR
3.8 to 7.2	13.883	6969.7	33.2	2	Overlapped	2.075	16	CFAR
7.2	13.883	2987.0	66.4	2	Overlapped	2.075	32	CFAR

* These numbers correspond to frequency number 3 of a 5 frequency sequence. The complete sequence is given below.

Transmit Frequency, GHz	PRF("7000") Hz	PRF ("3000") Hz
13.779	7017.4	3009.0
13.831	6993.5	2998.0
13.883	6969.7	2987.0
13.935	6946.1	2976.2
13.988	6822.6	2965.4

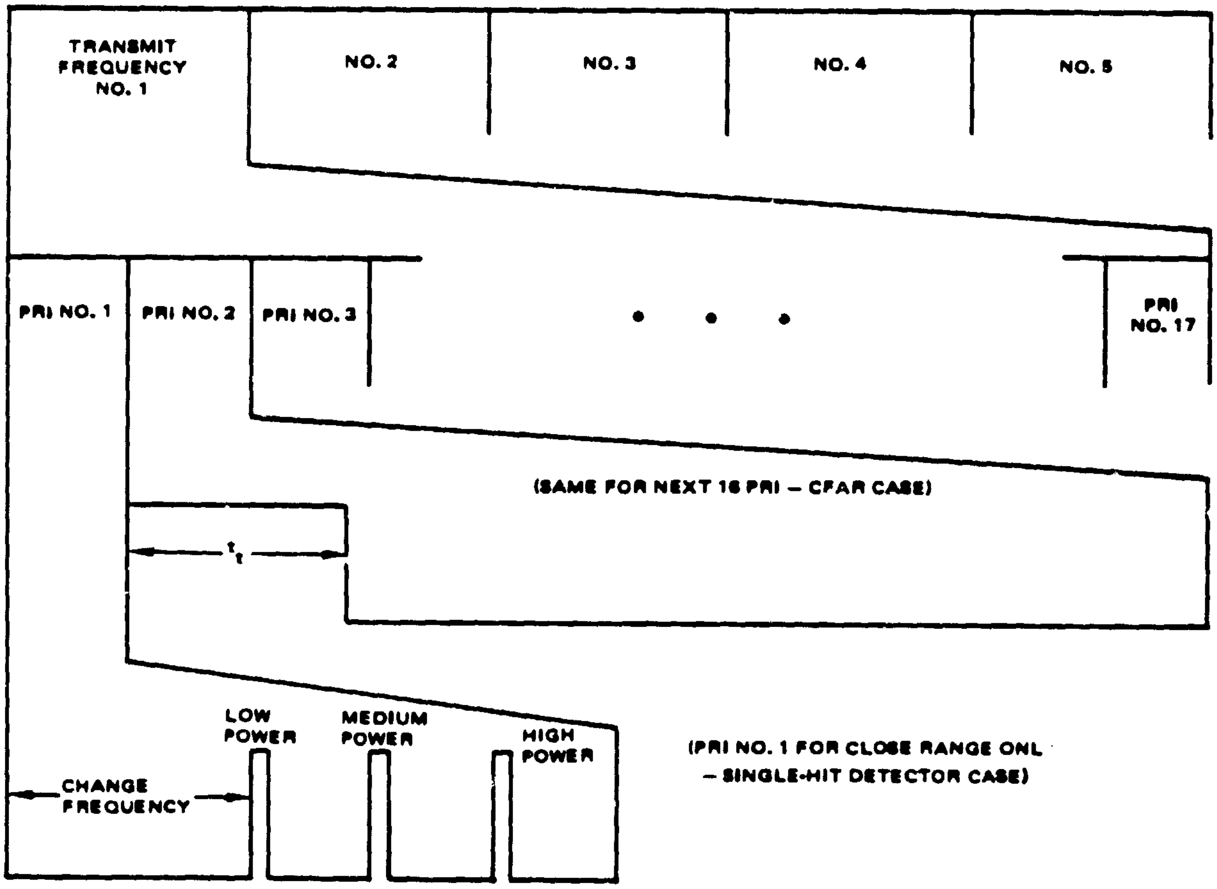


Figure 5-5. Passive Auto and Manual Search Mode Waveform.

TABLE 5-2. PARAMETERS FOR AUTO AND MANUAL PASSIVE SEARCH MODE

RANGE INTERVAL	FREQUENCY,GHz *	PRF, Hz *	PULSEWIDTH, μsec	NUMBER OF RANGE BINS	RANGE BIN TYPE	SAMPLE INTERVAL, μsec	NUMBER OF SAMPLES PER PULSE	DETECTOR TYPE
0 to 3000 ft	13.883	15,060	.122	50	Adjacent	.122	1	Single-Hit
3000 ft to 27 nm	13.883	2987.0	66.4	4	Adjacent	2.075	32	CFAR

* These numbers correspond to frequency number 3 in a 5 frequency sequence.

The complete sequence is given in Table 5-1.

C-2

commanded by a search initiate command from the GPC. The scan is centered at the angle designates received in the frame when the initiate command is given. The antenna begins executing the spiral scan pattern when the antenna has moved to within 0.3 degrees (Zone I) of these angle designates which are inertially held. Once the scan has been initiated the antenna spirals outwardly to a predetermined angle off the scan center, which depends on the target designated range, and begins to spiral inwardly. (These predetermined angles off scan center are called switch points and are summarized in Table 5-3). All scans will last 60 seconds or until a target is detected which ever comes first. It is also noted that the scan will terminate if the system mode or the antenna steering mode is changed.

Auto Passive or Active Modes. In this mode, the crew selects the scan center by slewing the antenna with the switches on the cockpit control panel. Once a scan center is selected, the crew initiates the spiral scan using the search initiate switch on the control panel. The scan pattern is the same in all situations. That is, the antenna spirals outwardly to 30 degrees off scan center and terminates. This procedure lasts for 60 seconds or until a target is detected whichever comes first.

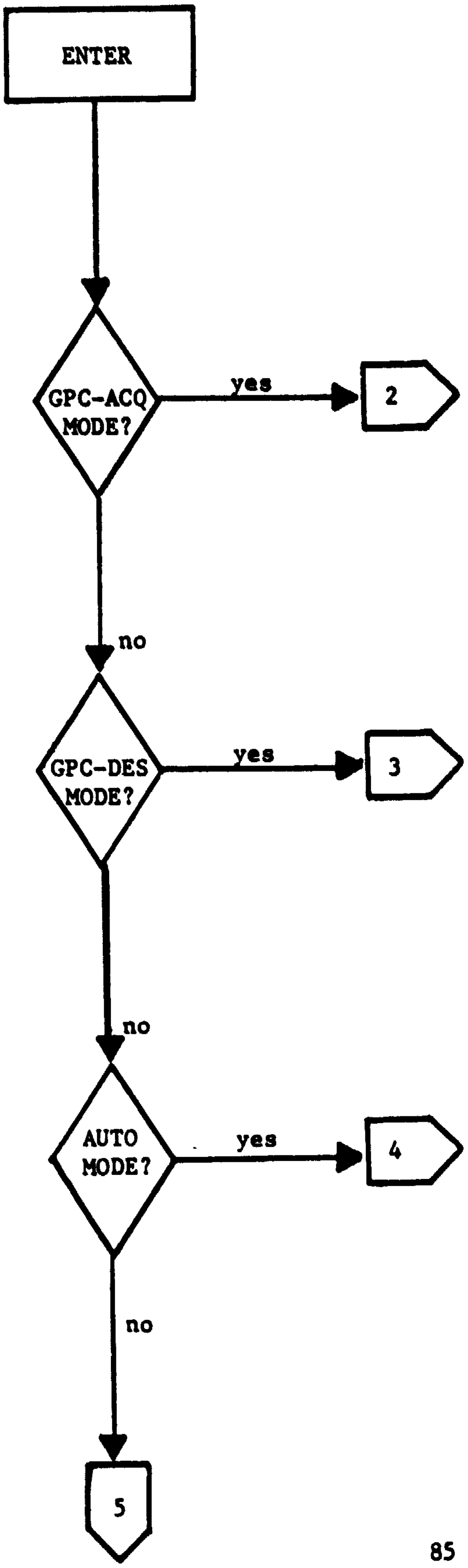
5.2 SEARCH MODE CONTROL ALGORITHM DESCRIPTION

Figure 5-1 provides an outline of the overall structure of the computer implementation of the search mode. The mainstay of this computer model is the search mode control algorithm (enclosed in dashed lines in Figure 5-1). The control sequence is (1) determine the antenna steering mode, (2) update the search operation using the proper antenna steering mode sequence, and (3) set the appropriate flags based upon the outcome of step (2). Figure 5-6 gives the detailed computer algorithm (called SEARCH) used to accomplish this task. Basically, this algorithm is partitioned into four sections where each section of code is dedicated to the complete search procedure for one of the antenna steering modes: GPC-ACQ, GPC-DES, Auto, or Manual. The computer code for each of these

Table 5-3 SCAN SWITCH (FROM OUTWARD TO INWARD SCAN) POINTS IN GPC-ACQ MODE

DESIGNATED RANGE, nm	SWITCH POINT, degree
0 to 8	Outward Scan Only (to 30°)
8 to 9.2	27.7
9.2 to 10.3	24.4
10.3 to 11.8	21.7
11.9 to 15	19.6
15 to 25	16.5
25 to 40	13.4
40 to 65	11.0
65 to 145	8.0
145 to 300	6.2

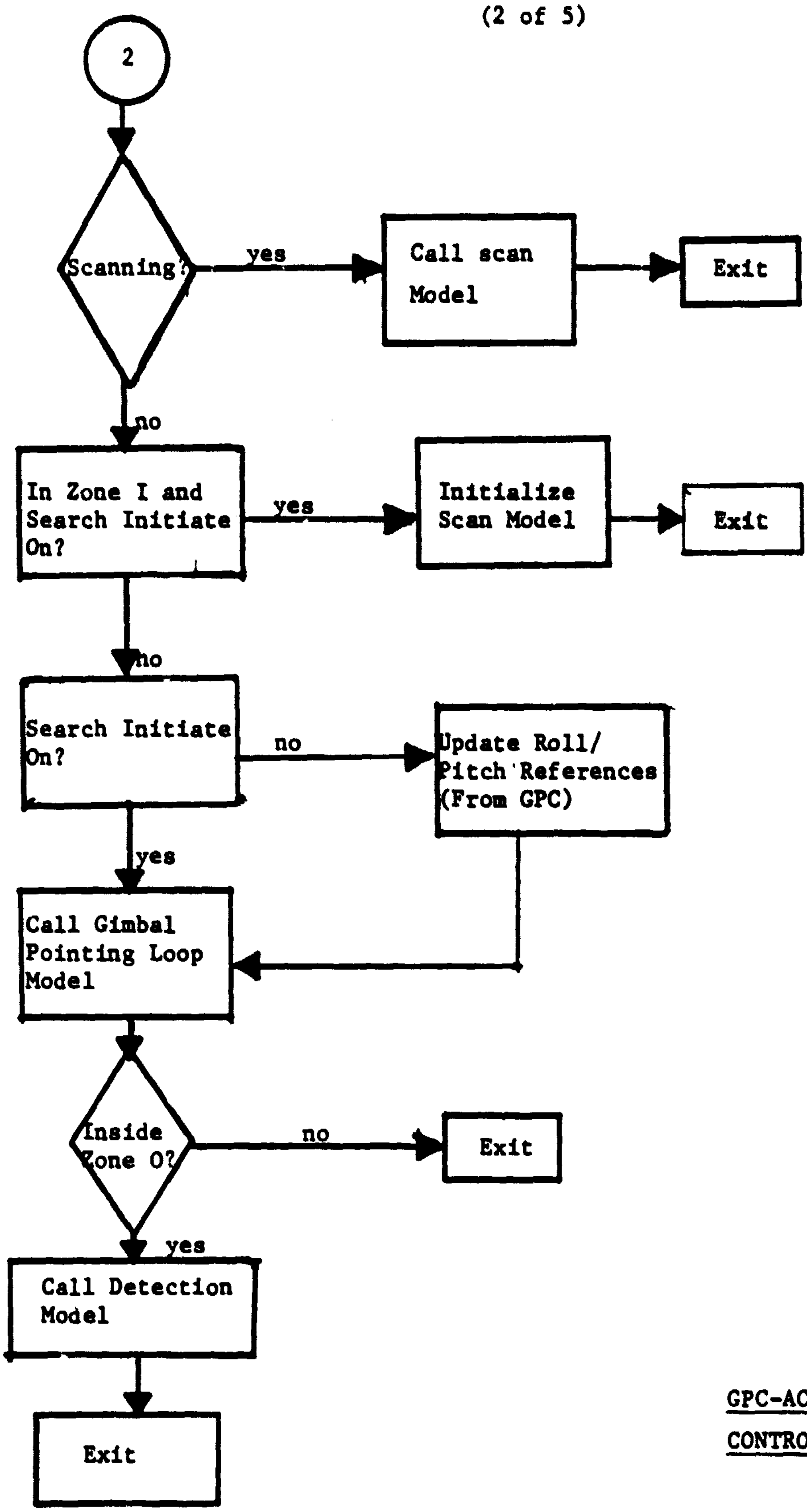
Figure 5-6 SEARCH MODE CONTROL COMPUTER ALGORITHM
(1 of 5)



Antenna Steering

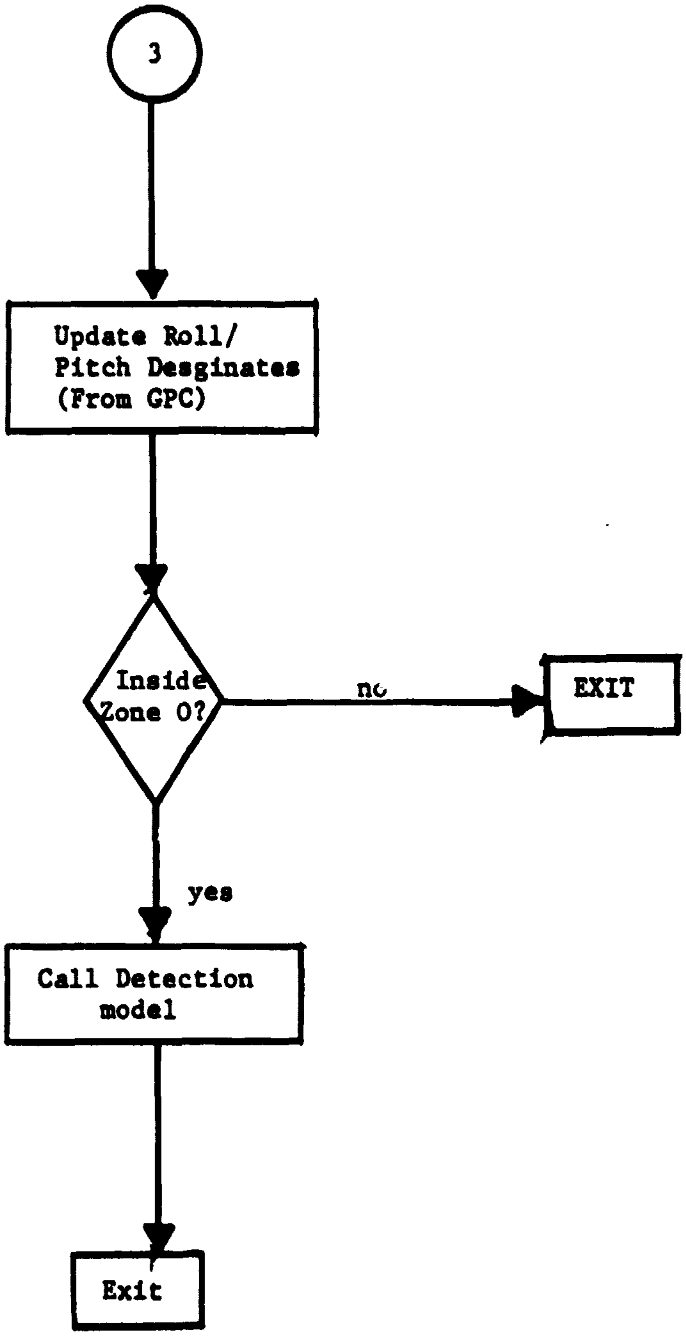
Mode Determination

Figure 5-6 SEARCH MODE CONTROL COMPUTER ALGORITHM (2 of 5)



GPC-ACQ STEERING MODE CONTROL LOGIC

Figure 5-6 SEARCH MODE CONTROL COMPUTER ALGORITHM
(3 of 5)



GPC-DES STEERING

MODE CONTROL LOGIC

Figure 5-6 SEARCH MODE CONTROL COMPUTER ALGORITHM (4 of 5)

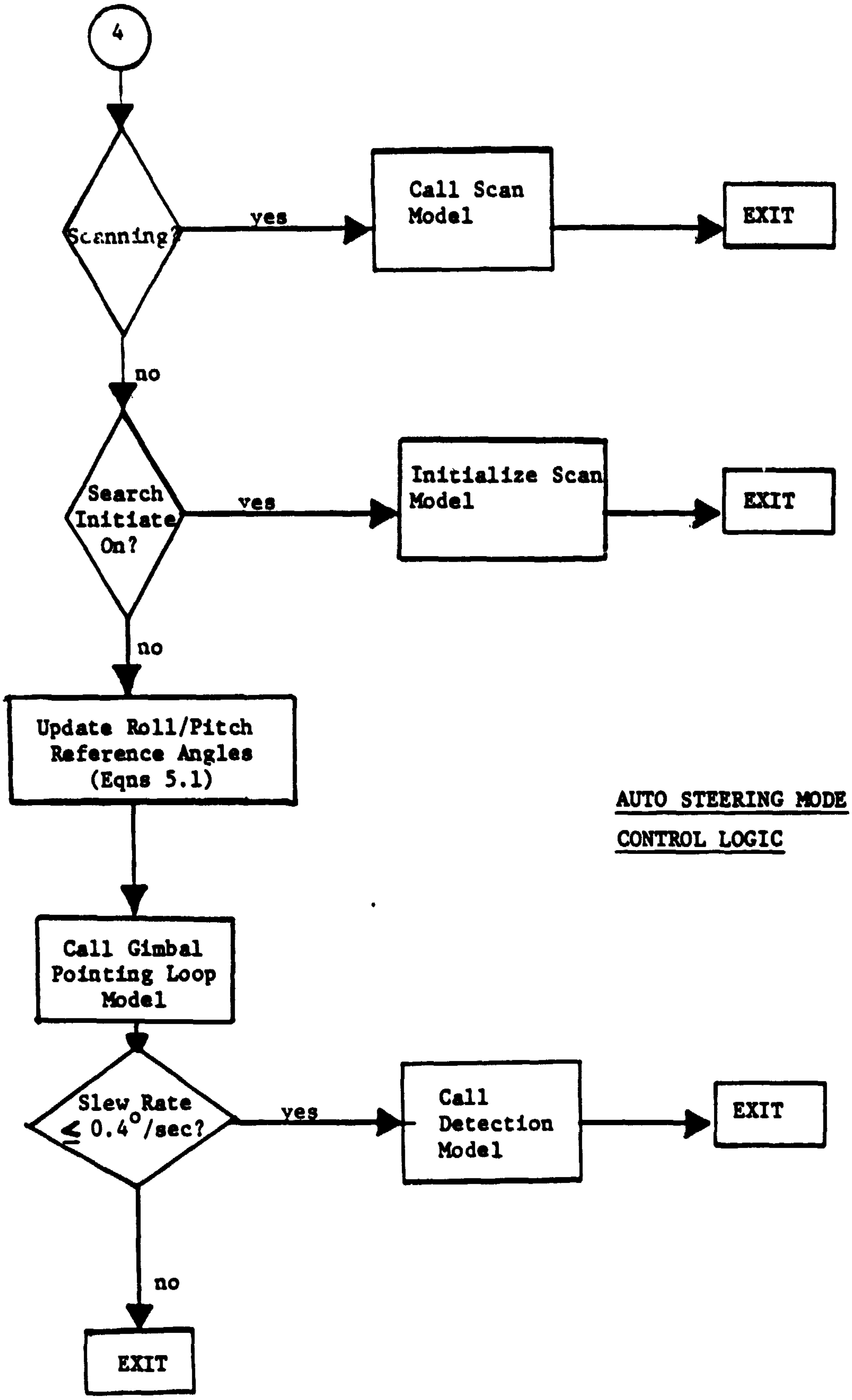
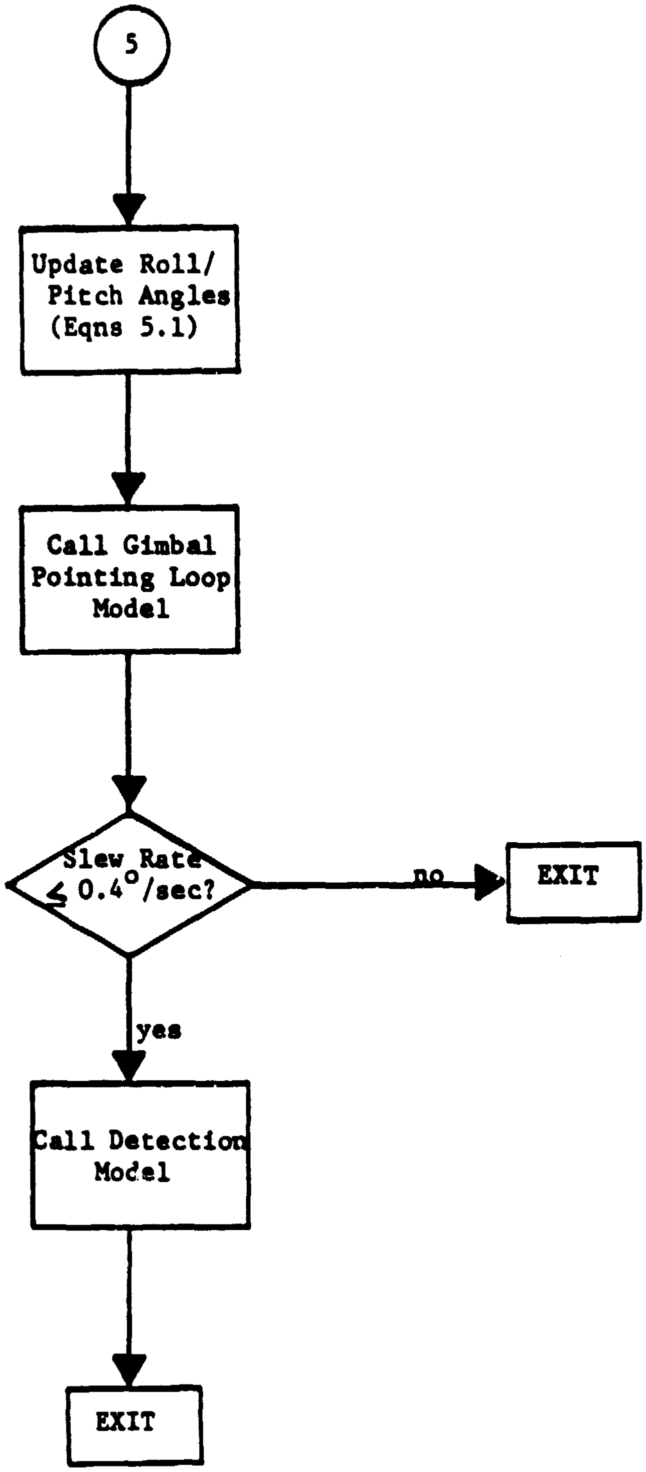


Figure 5-6 SEARCH MODE CONTROL COMPUTER ALGORITHM (5 of 5)


5
Update Roll/ Pitch Angles (Eqns 5.1)
Call Gimbal Pointing Loop Model
Slew Rate ≤ 0.4°/sec?
no
EXIT
yes
Call Detection Model
EXIT

MANUAL STEERING MODE

CONTROL LOGIC

sections closely mimics the operation summary given for the corresponding antenna steering mode in section 5.1.1 and, with the exception of the gimbal pointing loop reference computation, requires no further description.

The gimbal pointing loop reference computation is performed as follows. When the antenna is being slewed manually in either Auto or Manual, the roll and pitch references are updated using the expressions

$$\begin{aligned} \mathrm{Roll}_{\mathrm{Ref}}(n) &= \mathrm{Roll}_{\mathrm{Ref}}(n-1) + T_s\,\dot{\theta}_{AZ}\,(n) \\ \mathrm{Pitch}_{\mathrm{Ref}}\,(n) &= \mathrm{Pitch}_{\mathrm{Ref}}\,(n-1) + T_s\,\dot{\theta}_{EL}\,(n) \end{aligned} \tag{5.1}$$

where T_s = update interval,

$\dot{\theta}_{AZ}\,(n)$ = commanded roll rate at n th time sample,

$\dot{\theta}_{EL}\,(n)$ = commanded pitch rate at n th time sample.

In the GPC modes, the gimbal pointing loop references are set equal to the present angle designates if the search initiate command is low. But if the search initiate is high the pointing loop references are maintained at the angle designate values obtained in the update period when the initiate went high.

5.3 GIMBAL POINTING LOOP MODEL DESCRIPTION

A computer model of the antenna gimbal pointing loop is included in the search model to provide reasonable fidelity in the antenna motion response to

(1) angle designates from the GPC during GPC-ACQ and GPC-DES search modes,

(2) slew commands from the console during Auto and Manual search modes,

(3) slew commands from the console during Manual track mode.

It is noted that the present model does not contain gimbal stops or a cable unwrap capability. A simplified block diagram of the complete antenna gimbal pointing loop computer model is given in Figure 5-7. The description of this model is divided into two parts: (1) a definition of the basic servo loop model and (2) a detailed description of the computer algorithm which implements the process illustrated in Figure 5-7.

5.3.1 Basic Servo Loop Model Definition.

Both antenna gimbal servos were modeled using the second order loop shown in Figure 5-8. This choice for the servo loop model is based on the antenna servo simulation material presented at the March 1978 preliminary design review (PDR) [17], the description of the baseline antenna servo design given in [10] and [18], and discussions with Mr. J. C. Riles the antenna servo system designer for the Ku-Band Radar. Rationale for each of the basic model components is provided below. The first stage of the integration represents smoothing and shaping of the error signal and the second integration stage represents the effect of the gimbal. A limiter was placed between integration stages to represent the fact that the commanded gimbal rate is limited to 58 degrees per second in the hardware. Loop constants k_g and t_g are chosen to best approximate the characteristics of the real antenna gimbal response to slewing and designate commands . At the present time these constants are chosen to give a damping factor of 0.7 and a crossover frequency of 1.0 Hertz.

In order to represent the servo model of Figure 5-8 on the computer, it is approximated by the discrete-time model shown in Figure 5-9. This discrete-time model can be described mathematically as follows. The first step is to compute the error signal and update the output of the first integrator using the equations

(5.2) $$\dot{\alpha}_s(n+1) = \dot{\alpha}_s(n) + T_s k_g \varepsilon(n)$$

where $\dot{\alpha}_s(n)$ = smoothed α-gimbal rate estimate at the n th time sample,

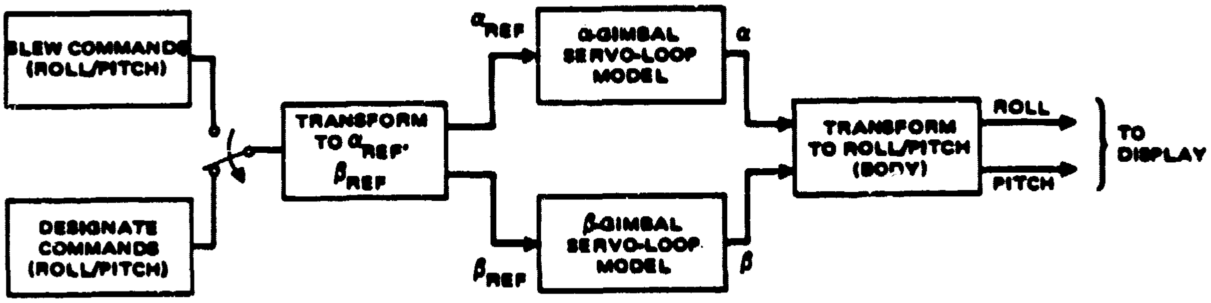

SLEW COMMANDS (ROLL/PITCH)
DESIGNATE COMMANDS (ROLL/PITCH)
TRANSFORM TO α_{REF}, β_{REF}
α_{REF}
β_{REF}
α-GIMBAL SERVO-LOOP MODEL
β-GIMBAL SERVO-LOOP MODEL
α
β
TRANSFORM TO ROLL/PITCH (BODY)
ROLL
PITCH
TO DISPLAY

Figure 5-7. ..nplified Block Diagram of the Gimbal Pointing Loop Model.

Figure 5-8 ANTENNA GIMBAL SERVO LOOP MODEL

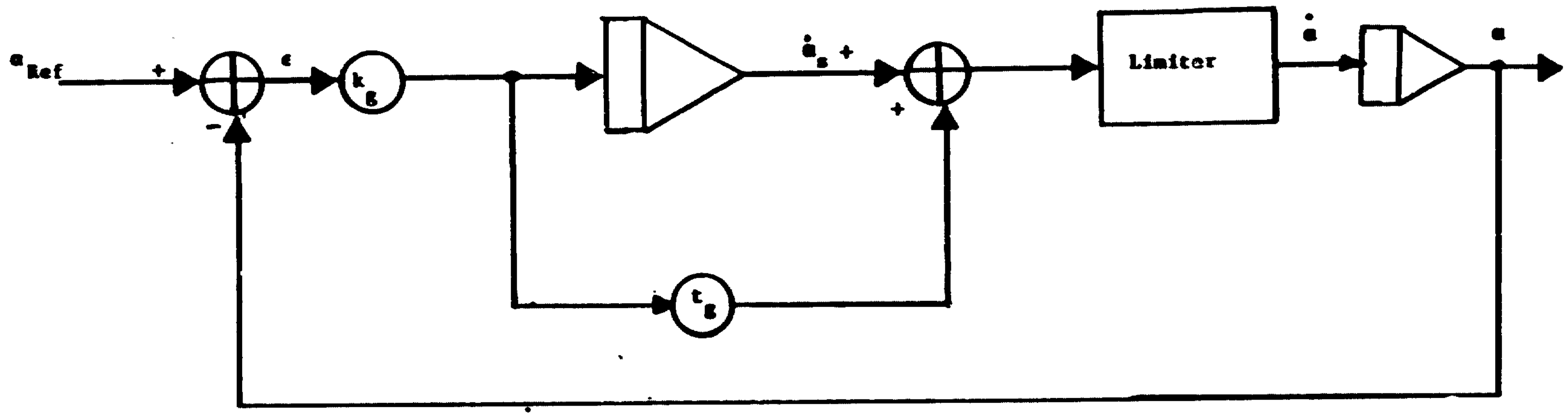
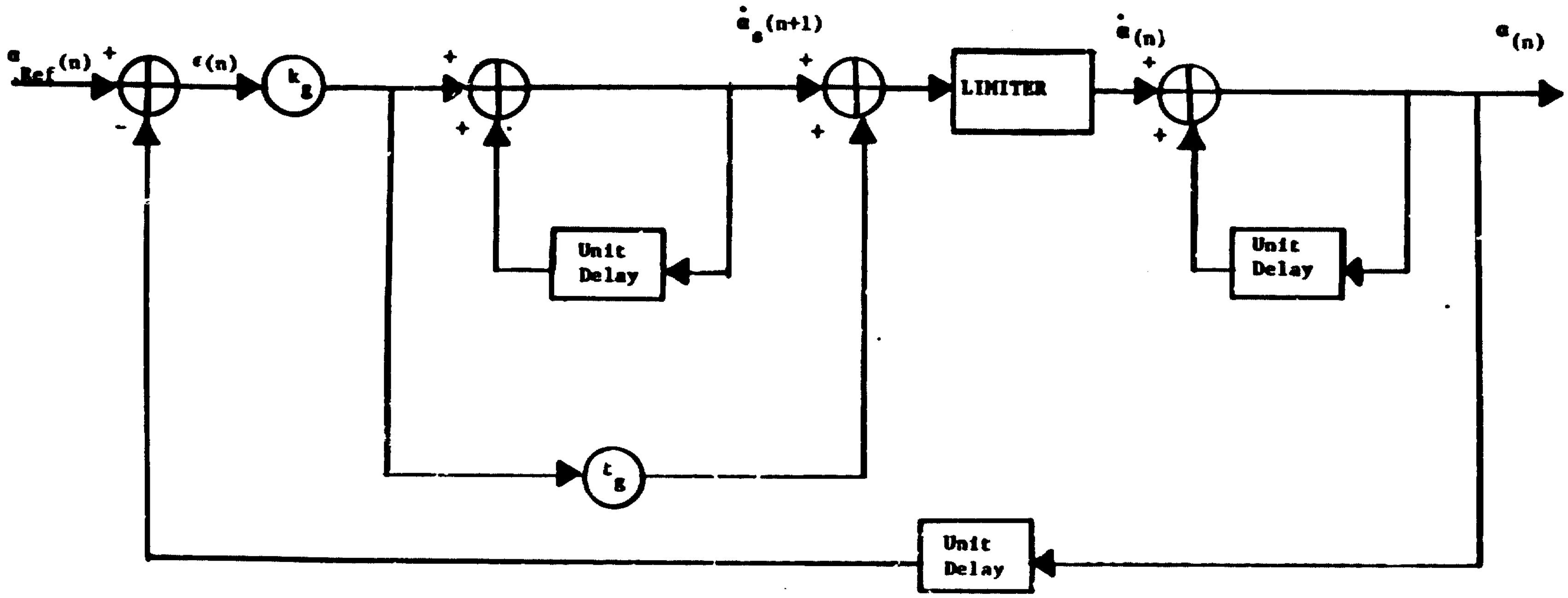


Figure 5-9 DISCRETE-TIME APPROXIMATION OF ANTENNA GIMBAL SERVO LOOP MODEL



$\epsilon(n) = \alpha_{Ref}(n) - \alpha(n)$ = error signal at n th time sample,

$\alpha_{Ref}(n)$ = o-gimbal reference position at time sample n,

$\alpha(n)$ = α-gimbal position at time sample n,

k_g = loop constant.

Next, the gimbal rate is updated by the expression

$$\dot{\alpha}(n+1) = \dot{\alpha}_s(n) + k_g t_g \, \epsilon(n) \tag{5.3}$$

where $\dot{\alpha}(n+1)$ = commanded gimbal rate at the n+1 th time sample,

t_g = loop constant.

The effect of limiting the commanded gimbal rate is given by

$$\dot{\alpha}(n+1) = \begin{cases} -58. & \text{, if } \dot{\alpha}(n+1) \leq -58. \\ \dot{\alpha}(n+1) & \text{, if} -58 \leq \dot{\alpha}(n+1) \leq 58. \\ +58. & \text{, if } \dot{\alpha}(n+1) \geq 58. \end{cases} \tag{5.4}$$

Finally, the α-gimbal position at the (n+1) time sample is obtained from

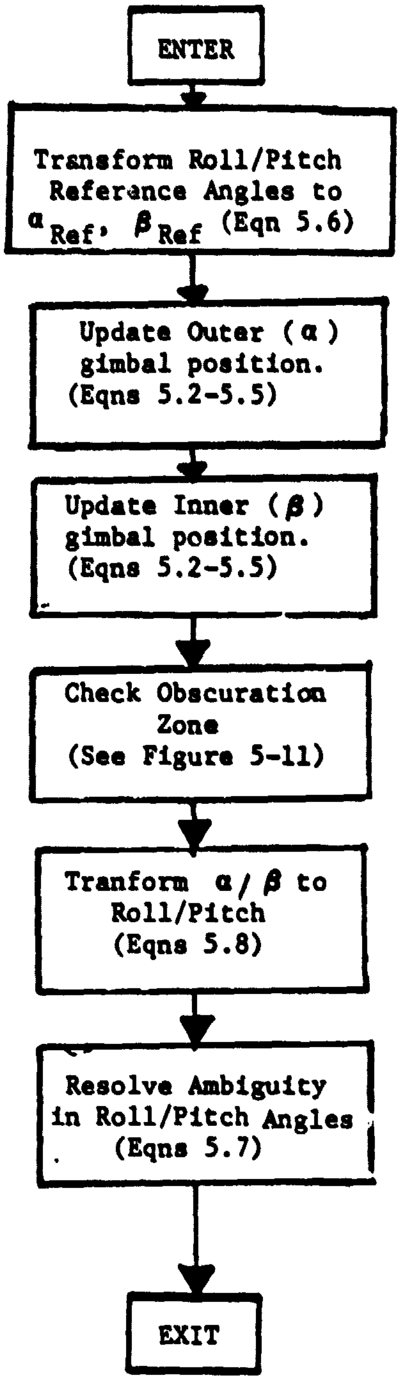
$$\alpha(n+1) = \alpha(n) + T_s \, \dot{\alpha}(n+1). \tag{5.5}$$

5.3.2 Computer Algorithm Details

A flow chart of the antenna gimbal pointing loop computer model is given in Figure 5-10. The required inputs for this model at each update are the desired roll and pitch reference angles. In the GPC modes these references are just the target angle designates and in Auto and Manual the references are obtained from equation (5.1). Using these new roll and pitch angle references, the algorithm updates the α and β gimbal positions using the procedure outlined below.

The first step of the gimbal pointing loop algorithm is to transform the roll and pitch reference angles expressed in body coordinates to α_{Ref}

Figure 5-10 ANTENNA GIMBAL POINTING LOOP COMPUTER ALGORITHM


ENTER
Transform Roll/Pitch Referance Angles to α_{Ref}, β_{Ref} (Eqn 5.6)
Update Outer (α) gimbal position. (Eqns 5.2-5.5)
Update Inner (β) gimbal position. (Eqns 5.2-5.5)
Check Obscuration Zone (See Figure 5-11)
Tranform α/β to Roll/Pitch (Eqns 5.8)
Resolve Ambiguity in Roll/Pitch Angles (Eqns 5.7)
EXIT

and β_{Ref} angles (or, equivalently, roll and pitch angles expressed in the radar frame). This transformation can be expressed as

$$\alpha_{\mathrm{Ref}} = \tan^{-1}\left[\frac{S_g S_p + C_g S_r C_p}{C_r\,C_p}\right]$$

(5.6)

$$\beta_{\mathrm{Ref}} = \sin^{-1}\left[C_g S_p - S_g S_r C_p\right]$$

where g = 6/ degrees,

p = $-\mathrm{pitch}_{\mathrm{Ref}}$,

r = $-\mathrm{roll}_{\mathrm{Ref}}$,

C = cos,

S = sin.

Also, it is noted that this transformation is identical to that used by the radar, i.e. it ignores the radar offset from the orbiter c.g. and the boom deployment error. In the next step, the α and β gimbal positions are updated using the servo-loop model given by equations (5.2) through (5.5).

Then it is determined whether the new α and β values lie in the obscuration zone. This task is accomplished using the algorithm given in Figure 5-11 which can be described as follows. First, the α, β angle ambiguity is resolved using the relations

(5.7)

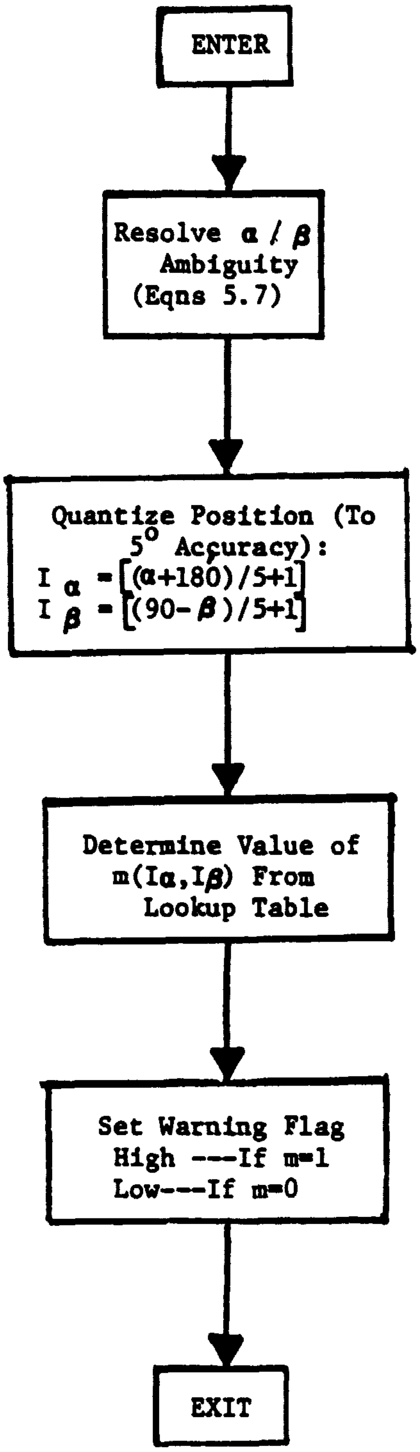
$$-90 < \beta \leq 90$$

$$-180 < \alpha \leq 180 \,.$$

Then a scan warning is determined by comparing the unambiguous α and β position to a map of the scan warning area shown in Figure 5-12 which was digitized to an accuracy of 5 degrees-by-5 degrees and stored in computer memory. If the comparison shows that α, β are in the obscuration zone, the scan warning flag is raised.

The final step in the gimbal pointing loop algorithm is to transform

Figure 5-11 ANTENNA OBSCURATION COMPUTATION ALGORITHM


ENTER
Resolve α / β Ambiguity (Eqns 5.7)
Quantize Position (To 5° Accuracy):
$I_\alpha = [(\alpha+180)/5+1]$
$I_\beta = [(90-\beta)/5+1]$
Determine Value of m(Iα,Iβ) From Lookup Table
Set Warning Flag
High ---If m=1
Low---If m=0
EXIT

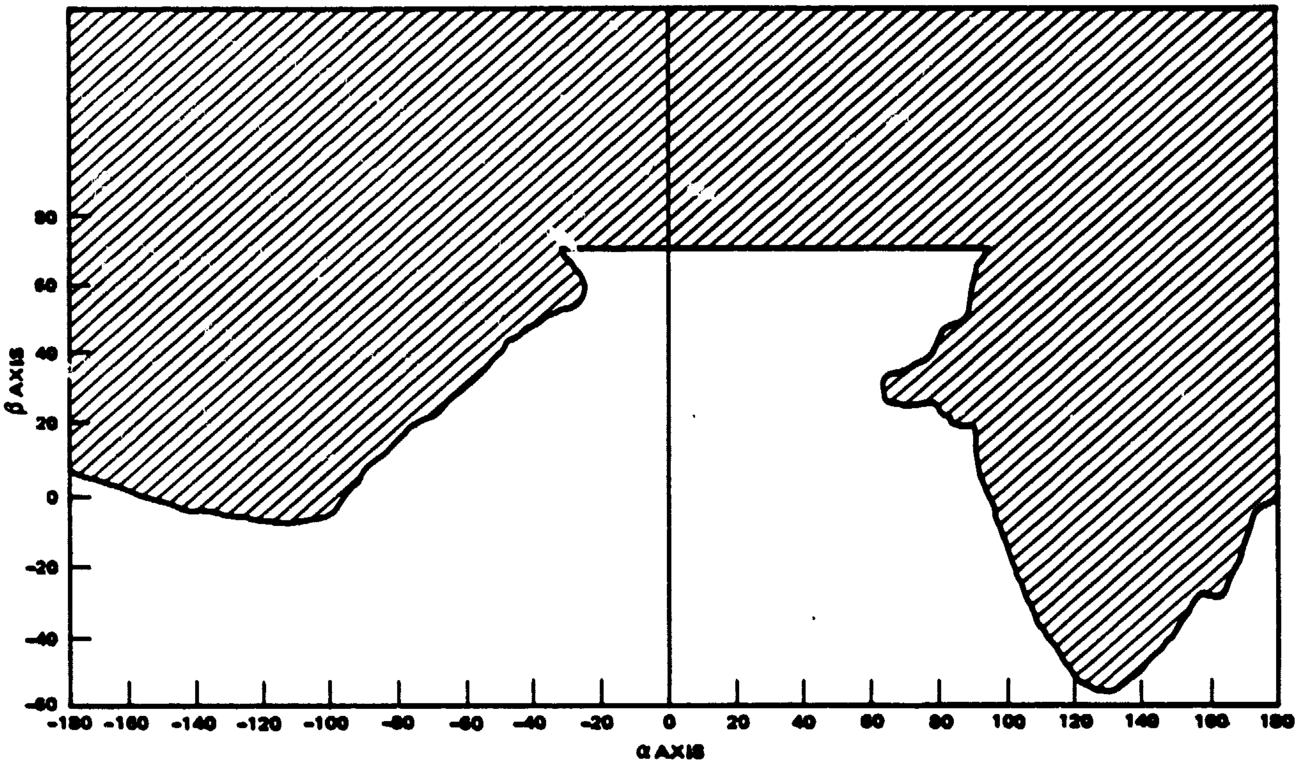


Figure 5-12. Antenna Obscuration Profile.

α and β to roll and pitch using the expressions

$$\begin{aligned} &\text{Roll angle} = -\tan^{-1}\left[\frac{S_g S_\beta + C_g S_\alpha C_\beta}{C_\alpha \; C_\beta}\right] \\ (5.8) \quad &\text{Pitch angle} = -\sin^{-1}\left[C\alpha \; S_\beta + S_g \, S_\alpha \; C_\rho\right]. \end{aligned}$$

These transformations are identical to the Ku-Band Radar transformations. Also any angle ambiguity is resolved using the convention given in (5.7).

5.4 SCAN MODEL DESCRIPTION.

The primary function of the scan model is to provide a simulation of search mode operation whenever the antenna is performing a spiral scan. Description of the scan model makes abundant use of a quantity called a "scan ring". We offer the following definition of this entity. It is noted that in reality the antenna traces out a spiral pattern about the scan center, however, we will approximate the spiral pattern as a set of concentric rings and label these rings as shown in Figure 5-13. With this definition in mind, the scan model can be described as follows.

When a spiral scan has been initiated from the console or the GPC, the model tracks the antenna boresight position to the nearest scan ring (see Figure 5-14) and tracks the target positon exactly. A target detection is attempted only if the boresight and target are in the same ring in the present update period and were not in the same ring in the previous update period. If an attempt at detection is successful, the scan procuedure is terminated and control is handed over to the track routines. If no detection is obtained then the scan procedure continues until another target detection is allowed or the scan is completed.

The main advantage of the model is that it offers reasonably accurate estimates of elapsed time from scan initiate to target detection or an end-of-scan condition for an arbitrary rendezvous situation. Maximum error in elapsed

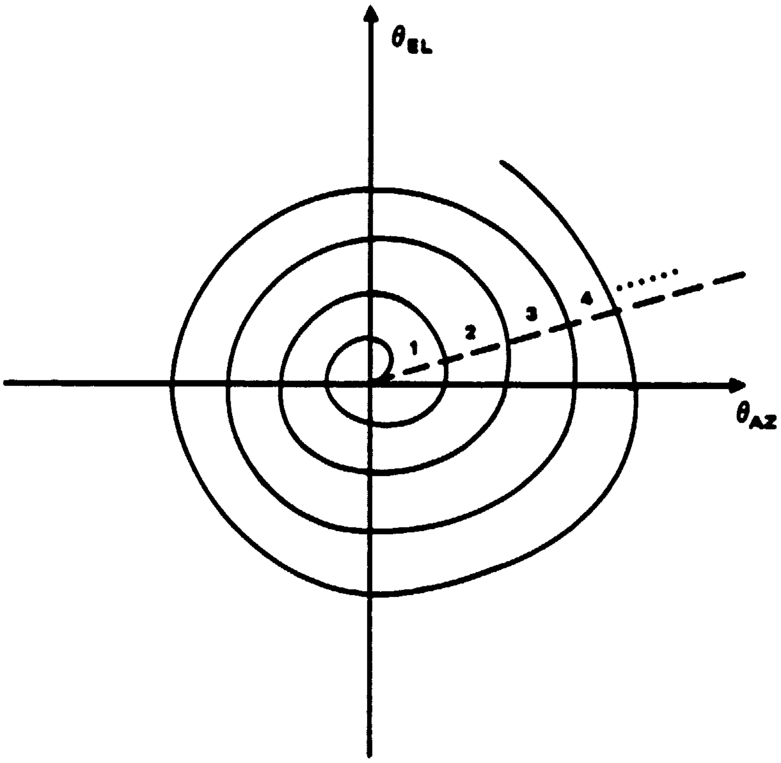
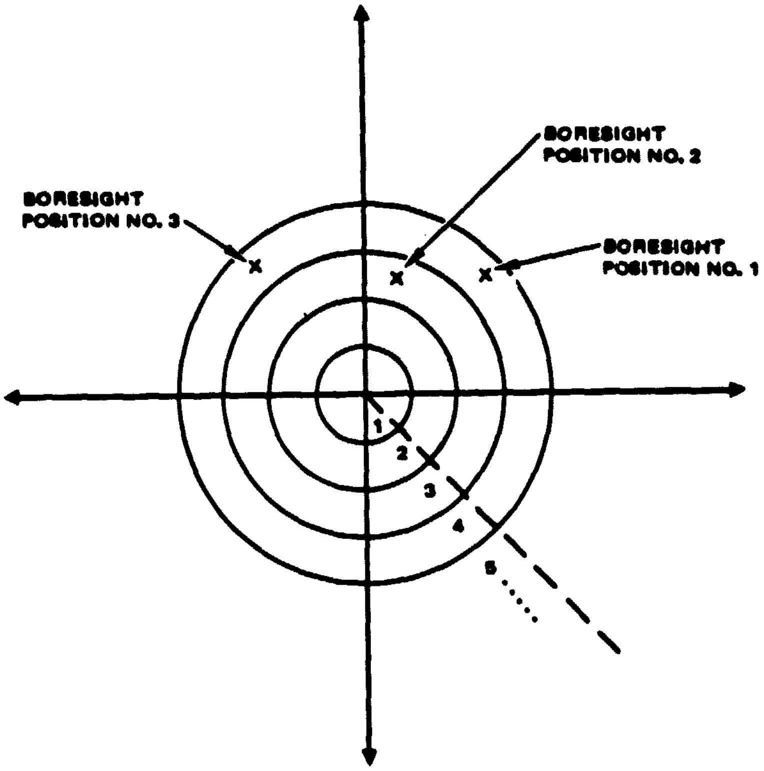


Figure 5-13. Definition of Scan Rings.



ACTUAL BORESIGHT POSITION	RING ASSIGNMENT
NO. 1	RING 4
NO. 2	RING 3
NO. 3	RING 4

Figure 5-14. Illustration of Antenna Boresight Ring Assignment Method.

time for any situation should be no worse than ± 2 seconds. It is noted that there are some deficiencies in the model too. These are that (1) there is no inertial stabilization of the scan center during a scan in the present version and (2) the target detection capability is highly inaccurate under certain target motion conditions. For example, when the target is moving radially with respect to the scan center.

5.4.1 Summary of Scan Operation

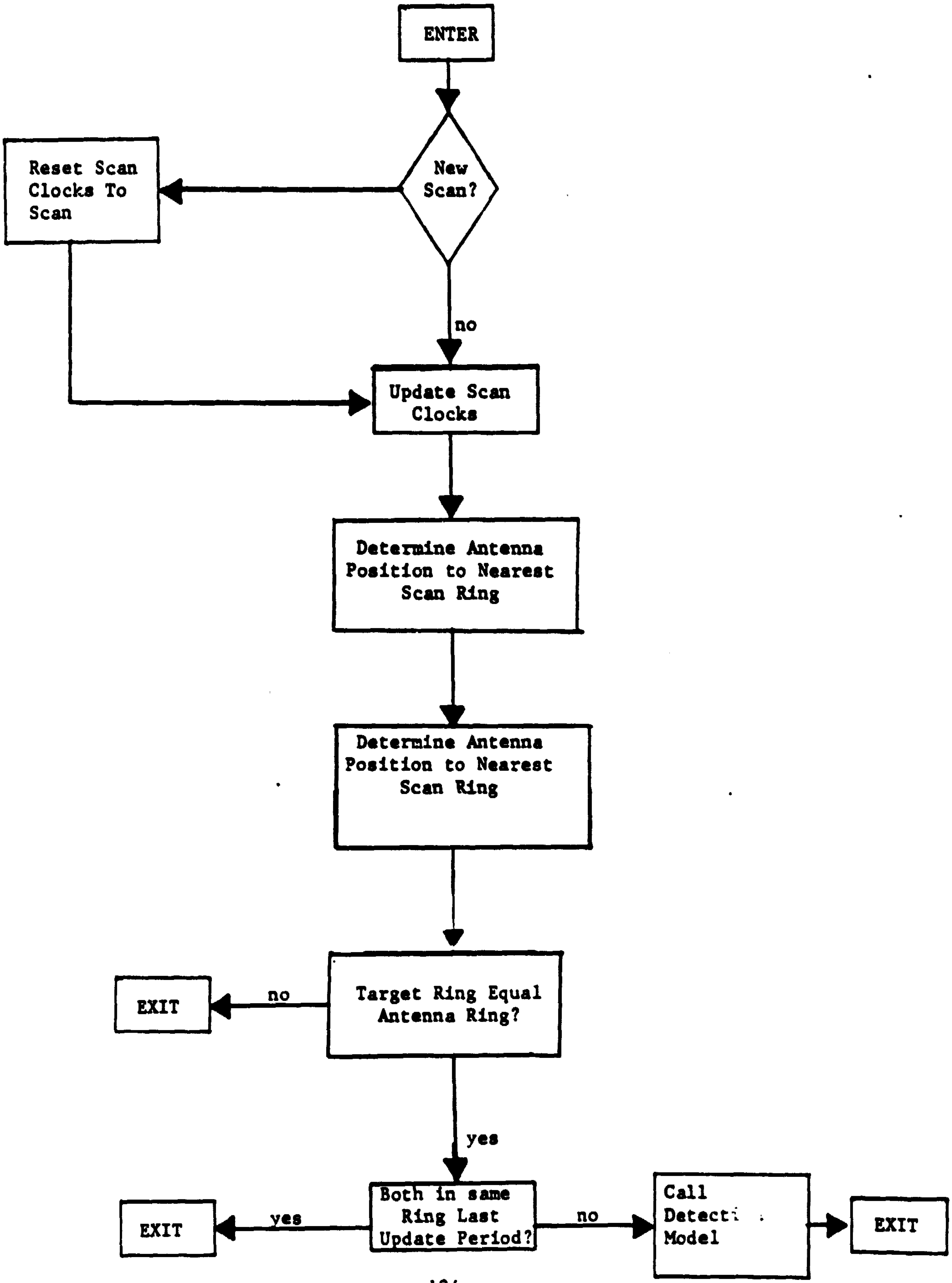
Rules and conditions for scan initiation and termination in the various antenna steering modes are identical to the Ku-Band Radar scan rules summarized in Section 5.1.4.

5.4.2 Computer Algorithm Details

Figure 5-15 gives a flow chart of the scan model computer algorithm. Step one of the procedure is to determine whether or not the scan has just been initiated. If the scan has just been initiated, then the scan model must be initialized. This procedure consists of raising the search flag and resetting the scan clock and other time parameters to zero. In step two the scan clock is updated and checked for an end-of-scan condition. If no end-of-scan is obtained, then we proceed to the next set of steps which involve determination of target and boresight positions in the scan area.

In the third step, the antenna boresight position in the scan area is resolved to the nearest scan ring. When no switch point is involved, i.e. the antenna only spirals outwardly to 30 degrees and stops, determination of the boresight scan ring location is done in the following way. Since the elapsed scan time is known, this value can be used to address a lookup table which contains the boresight scan ring position versus scan time profile shown in Figure 5-16. (Data for this curve was obtained from a detailed simulation of the antenna scan process written by Mr. J.C. Riles of Hughes.) For those modes where a switch

Figure 5-15 SCAN PROCEDURE COMPUTER ALGORITHM


ENTER
New Scan?
Reset Scan Clocks To Scan
no
Update Scan Clocks
Determine Antenna Position to Nearest Scan Ring
Determine Antenna Position to Nearest Scan Ring
Target Ring Equal Antenna Ring?
no
EXIT
yes
Both in same Ring Last Update Period?
yes
EXIT
no
Call Detection Model
EXIT

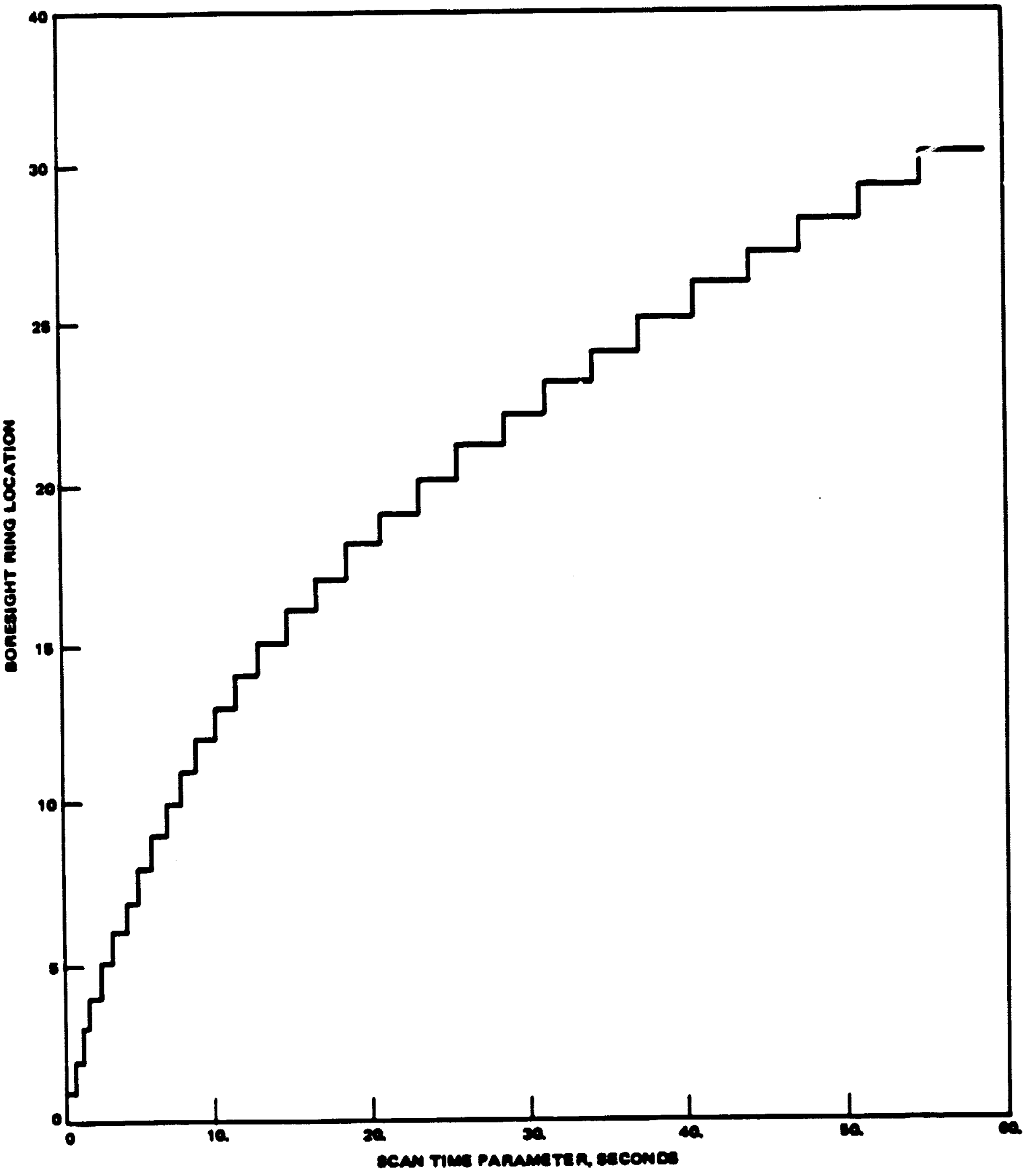


Figure 5-16. Boresight Ring Position as a Function of the Scan Time Parameter (T_Δ , eqn 5.9).

point is involved, the following assumption is used: at the switch point, the boresight begins to retrace the profile given in Figure 5-16. With this assumption, the boresight position can be determined in all possible scan cases using the profile of Figure 5-16 and defining the time parameter

$$(5.9) \qquad t_{\Delta}(n) = \begin{bmatrix} t_{SN}(n-1) + T_s & , & t_{SN}(n-1) \leq t_{switch} \\ t_{SN}(n-1) - T_s & , & t_{SN}(n-1) > t_{switch}. \end{bmatrix}$$

where $t_{sn}(n)$ = elapsed scan time at n th sample time,

T_s = update interval,

t_{switch} = time at which switch occurs (measured from scan initiation).

The fourth step in the scan model procedure is to determine in which scan ring the target is located. To do this,we first compute the target's angle off scan center, call it Θ_{SN}, using the expression

$$(5.10) \qquad \Theta_{SN}(n) = \cos^{-1} \left[\hat{r}_o^{\,L} \cdot \hat{r}_s^{\,L}\right]$$

where $\hat{r}_o^{\,L}$ = unit vector in the direction of the target c.g. expressed in L-coordinates,

$\hat{r}_s^{\,L}$ = unit vector in the direction of the scan center expressed in L-coordinates.

This value of $\Theta_{SN}(n)$ is used to obtain the target's scan ring position from the scan ring number versus Θ_{SN} curve shown in Figure 5-17 (Data for this curve was also provided by Mr. J. C. Riles) and stored in the computer. It is noted that in practice only the ring transition points, denoted by Θ_i, are stored in memory. Then, one determines the target scan ring location by choosing Θ_i such that $\Theta_i \leq \Theta_{SN} \leq \Theta_{i+1}$.

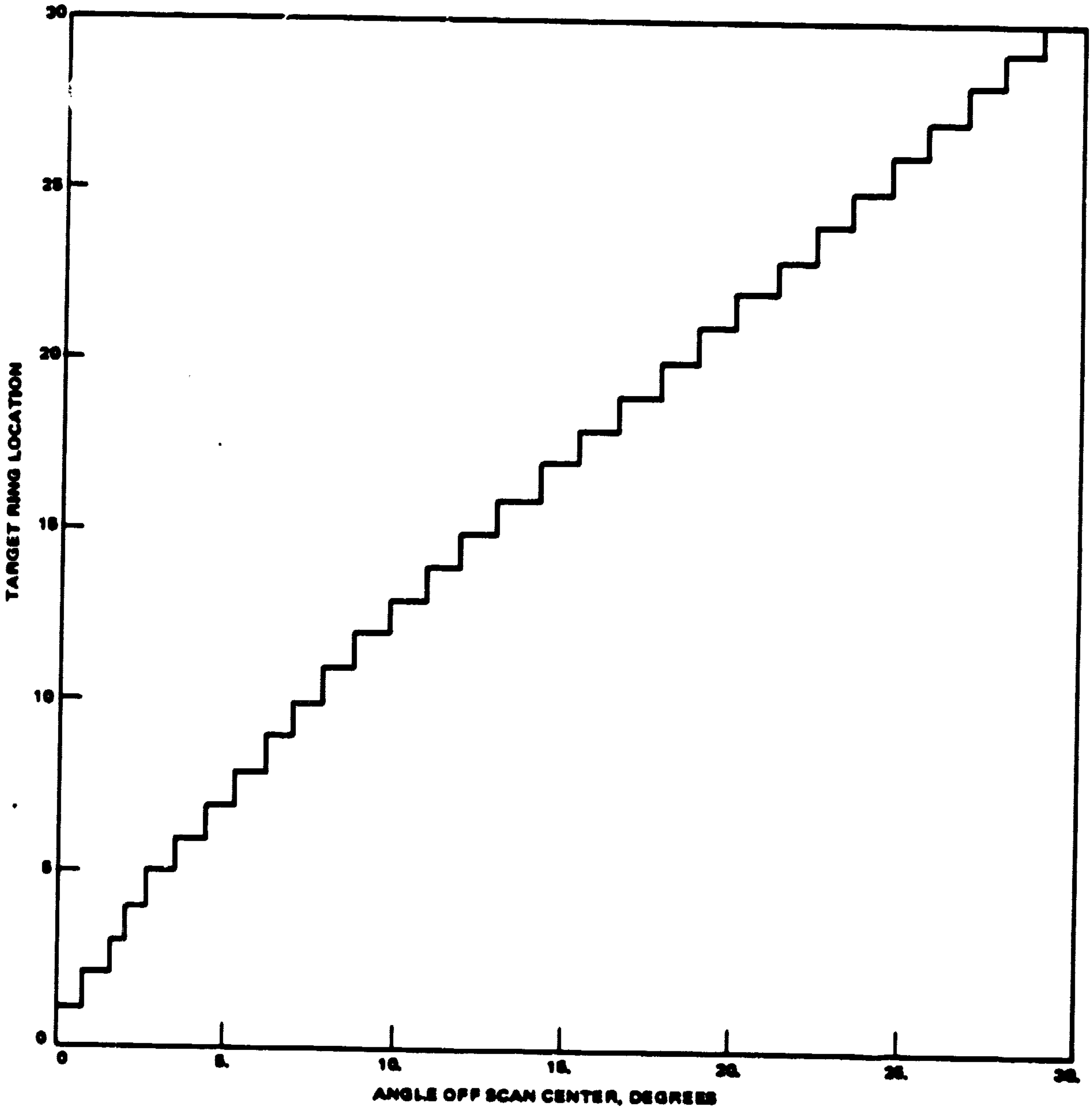


Figure 5-17. Target Ring Location as a Function of Angle (Θ_{SN}) off Scan Center.

In the final step, it is determined whether a detection attempt (using the detection model described in Section 5.5) should be made. A detection will be attempted if the target ring number and the boresight ring number satisfy the following two conditions:

(1) they are equal in the present update period,

(2) they were not equal in the previous update period.

If a target detection is attempted the following logic governs the outcome of the process. If the target is not detected the scan is continued. If the target is detected the scan is halted, the target present flag is raised (MTP=1) the search flag is lowered (MSF=0), and control is handed to the track subroutine.

It is emphasized that no acquisition mode operations are modeled, including the mini-scan sequence. It is also assumed that any detection is a mainlobe detection and the track mode is initialized accordingly.

5.5 DETECTION MODEL DESCRIPTION

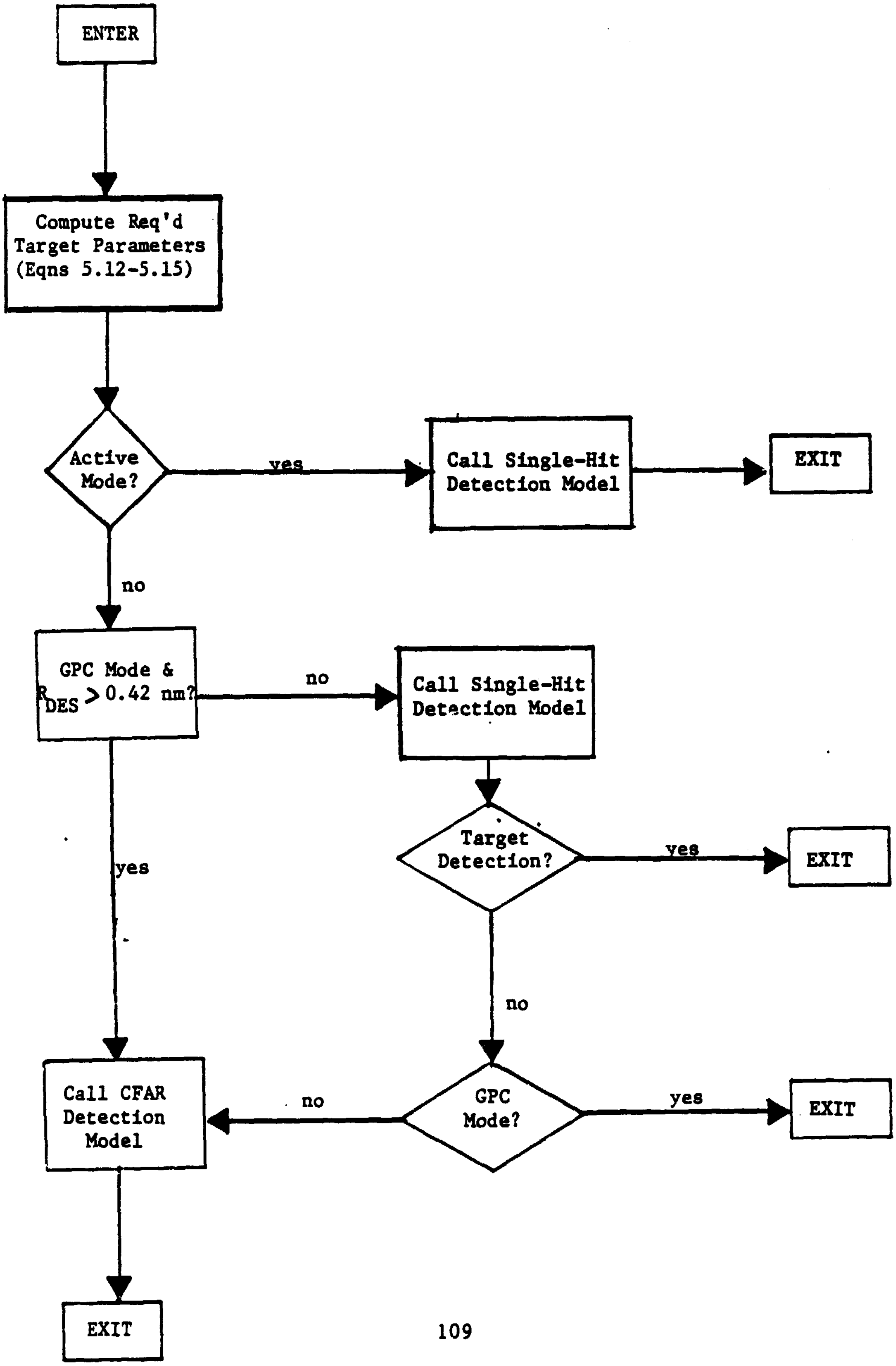
The detection model provides a simulation of the detection process for each of the search modes. Figure 5-18 gives a simplified flow chart of the detection model computer algorithm. The basic model consists of two types of detectors, a CFAR model and a single-hit model, and some control logic that decides which operating parameters and detector types should be used. In the remainder of this subsection, the modeling assumptions are listed, the two detector models are described, and the computer algorithm details are provided.

5.5.1 Model Assumptions

The following assumptions were used in the development of both detector models:

(1) the target is a point scatterer with a slowly fluctuating (Swerling II) RCS which has a fixed, predetermined average value for all aspect angles,

FIgure 5-18 DETECTION MODEL COMPUTER ALGORITHM


ENTER
Compute Req'd Target Parameters (Eqns 5.12-5.15)
Active Mode?
yes
Call Single-Hit Detection Model
EXIT
no
GPC Mode & $R_{DES} > 0.42$ nm?
no
Call Single-Hit Detection Model
Target Detection?
yes
EXIT
yes
no
Call CFAR Detection Model
no
GPC Mode?
yes
EXIT
EXIT

(2) the target radial velocity does not change over the data cycle,

(3) for all nonscanning modes, the beamshape loss obtained at the beginning of the data cycle is used for the entire data cycl .

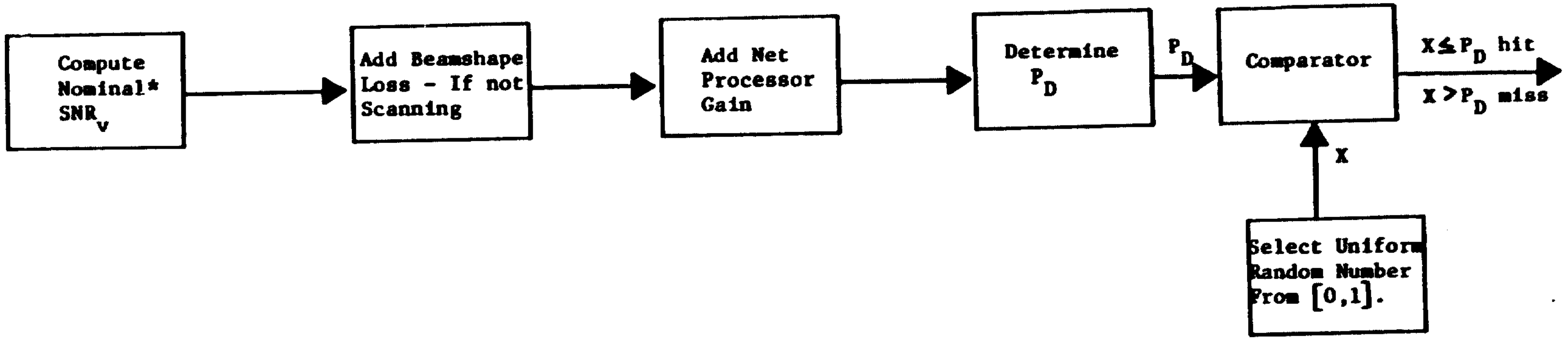
(4) for all scanning modes, an average beamshape/scan loss, based on the target position in the scan pattern and computed using a simulation documented in [19] , is used instead of computing the loss on a pulse-by-pulse basis.

In some cases, assumption (1) can have a significant impact upon fidelity of the detection process. However, if the fixed average RCS is chosen carefully, then this model should provide the crew with a reasonable feel for the target detection capability. Assumptions (2) and (3) are forced on us by the constraints of real-time operation. That is, the motion is only updated at the sample rate. We are stuck with assumption (4) because of the method selected to model the scan process and the real-time constraint.

5.5.2 CFAR Detection Model

Figure 5-19 illustrates the CFAR detection process. For a given target range and velocity, target position in the scan area, and average target RCS value, the basic idea of the procedure is as follows. First, the SNR at the doppler filter output is computed. (This value excludes beamshape/scan loss when scanning but includes beamshape loss when not scanning.) This value of SNR_D is then used to determine the probability of detection P_D from a precomputed P_D versus SNR_D curve which is stored in computer memory. The statistical character of the detection process is injected by selecting a number x from a population which is uniformly distributed on $[0,1]$ and deciding the outcome of the detection process using the algorithm

Figure 5-19 CFAR DETECTION MODEL



*Nominal Means Beamshape and Scan Loss is not included.

$$(5.11) \quad \begin{cases} X \leq P_D & \text{target detected} \\ X > P_D & \text{target not detected.} \end{cases}$$

It is noted that the P_D versus SNR_D data implicitly contains the beamshape/scan loss in those cases when the antenna is scanning. Further details of the P_D curves are provided in Section 5.5.4.

5.5.3 Single-Hit Detection Model

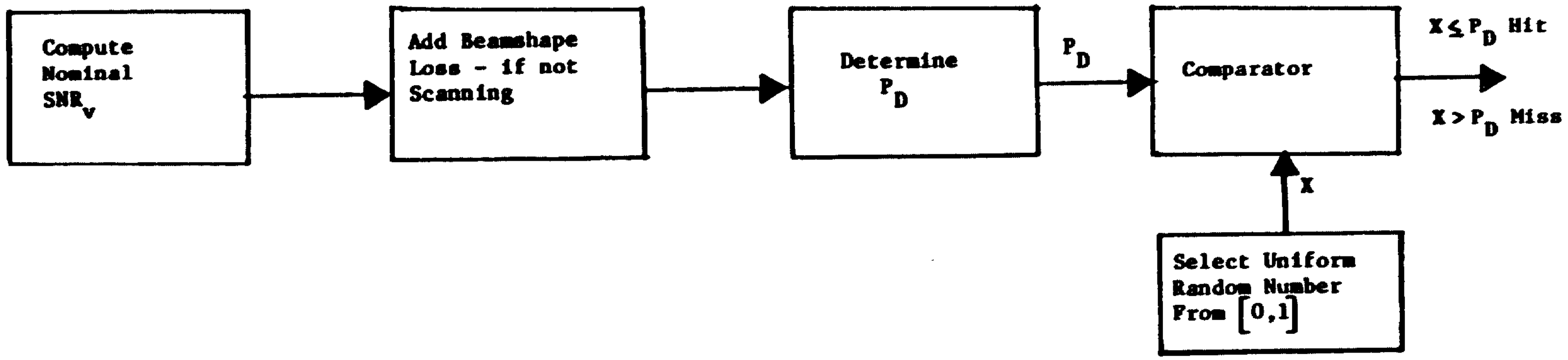
A block diagram of the single-hit detection model is given in Figure 5-20. Fundamentally, this procedure is identical to the CFAR detection model. However, in this case the SNR at the video filter output is used to obtain the required probability of detection P_D from the proper P_D versus SNR_v data stored in a lookup table. Once the P_D is selected the rest of the procedure is identical to the CFAR procedure.

As in the CFAR case, the SNR_v computation excludes the beamshape/scan loss when scanning but includes the beamshape loss when not scanning. The P_D curves will implicitly contain the average beamshape/scan loss in the scanning cases.

5.5.4 Determination of P_D(SNR) Data

All of the P_D curves required by the detection algorithm were generated using a very accurate model of the Ku-Band Radar search mode processor, modified appropriately for each case. This simulation model, documented in Reference [19] can be described as follows. It contains a very accurate model of the signal processor and a slowly fluctuating (Swerling II-type) target model. The model also contains the capability for including an average scan loss in the P_D computation. This capability is too involved to describe here. Therefore the reader is referred to [19] for complete details of the average scan loss model. It is noted that the scan model uses the scan parameters, i.e. target dwell time and beam overlap, associated with the outer edge of the scan pattern for all cases, regardless of target position in the scan pattern. If time permits, the scan loss will be modeled more accurately.

Figure 5-20 SINGLE-HIT DETECTION MODEL



For a given set of conditions, the simulation model of [19] computes the SNR versus P_D profile in the following manner. For a given nominal SNR_V (nominal means ignore the beamshape/scan loss), the Monte Carlo technique is employed to determine the associated P_D. Then, performing this computation for a range of SNR_V values gives the desired P_D versus SNR data. We note that the range of SNR is always chosen so that the data points adequately described the curve from 5% to 95% P_D. Figure 5-21 gives an example of the resulting data for the GPC-ACQ CFAR scanning case.

As a side remark we note that the data generated by this model for the CFAR case implicitly contains an average beamshape/scan loss (if scanning), the losses associated with the magnitude detector, and the losses due to the threholding technique. In the single-hit detection cases this data will contain the average beamshape/scan loss implicitly (if scanning).

The P_D (SNR) data will be modeled on the computer in the following way. For all CFAR cases, P_D data will be generated by the simulation described above for SNR_D values ranging from 0 dB to 20 dB spaced at 1/2 dB increments. For any SNR_D values above 20 dB, the P_D is set equal to 1 and for values of SNR_D below 0 dB, the P_D is set equal to 0. For all single-hit cases, P_D data will be generated for SNR_V values ranging from -25 dB to -5 dB spaced at 1/2 dB increments. SNR_V values outside this range are treated in the same manner as the CFAR case.

5.5.5 Computer Algorithm Details

The detection computer model consists of a set of three algorithms: (1) the control algorithm shown in Figure 5-18, (2) the CFAR detection algorithm shown in Figure 5-19, and (3) the single-hit detection algorithm shown in Figure 5-20. The control algorithm first computes the target parameters required by the detection algorithms and then it decides which detector algorithm should be called based on the operating mode. In the first step of the control algorithm, the point target range, the point target radial velocity with respect to the

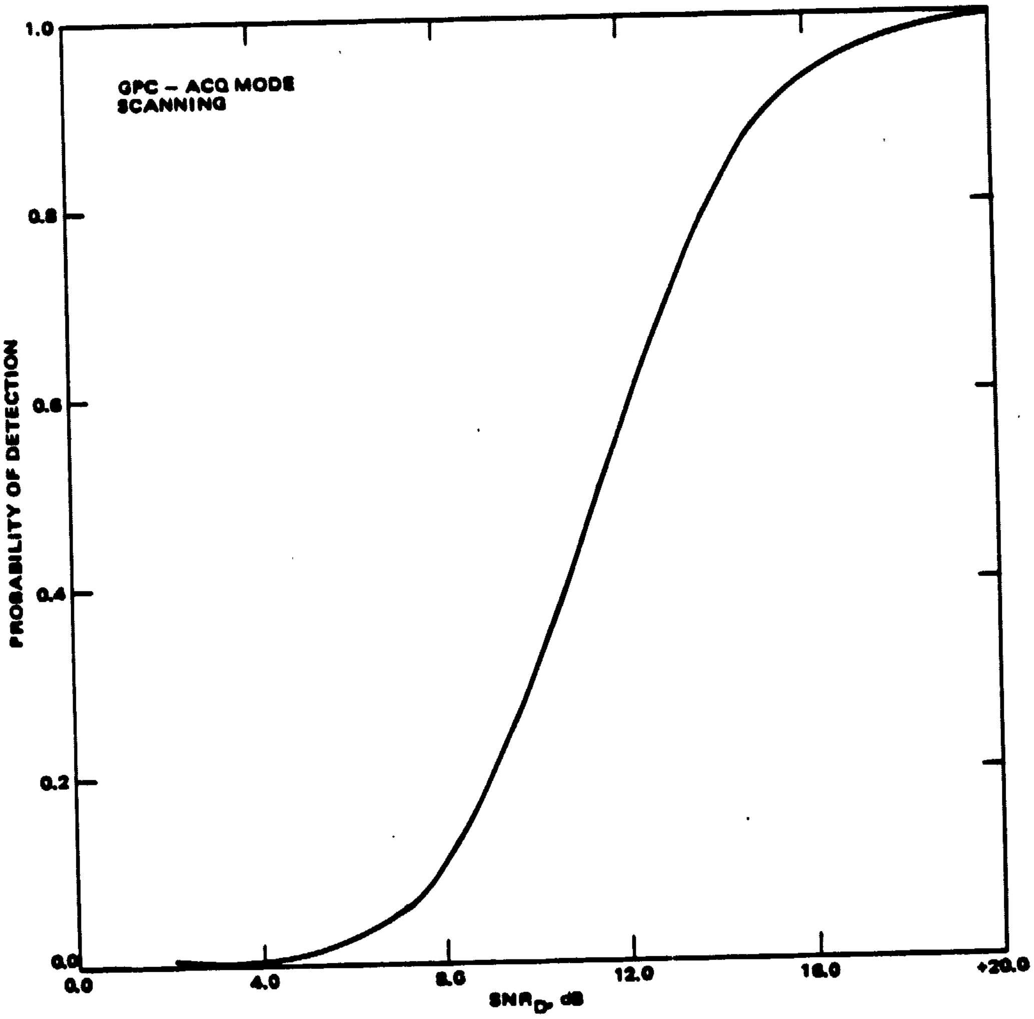


Figure 5-21. Example of P_D Versus SNR_D Data

radar, and the angle off the boresight are computed as follows. The range is given by the expression

$$(5.12) \qquad r_o^{\,L} = \left| T_{LB} \left(\vec{r}_o^{\,B} - \vec{X}^B \right) \right| \qquad \text{(Range)}$$

where $\vec{r}_o^{\,B}$ is provide.' by the parent simulation, $\vec{X}^B$ is the fixed radar offset from the orbiter C.G. and T_{LB} is given by

$$T_{LB} = \begin{pmatrix} C\beta & 0 & -S\beta \\ 0 & 1 & 0 \\ S\beta & 0 & C\beta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & C\alpha & S\alpha \\ 0 & -S\alpha & C\alpha \end{pmatrix} \begin{pmatrix} C\gamma & S\gamma & 0 \\ -S\gamma & C\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

where α, β = most recent positions of the antenna gimbals,

γ = yaw angle of R-frame with respect to B-frame,

C = cos,

S = sin.

The radial velocity is computed using the equation

$$(5.14) \qquad \dot{r}_o^{\,L} = \dot{\vec{r}}_o^{\,L} \cdot \hat{r}_o^{\,L} \qquad \text{(Radial velocity)}$$

where

$$\dot{\vec{r}}_o^{\,L} = T_{LB}\, \dot{\vec{r}}_o^{\,B} + \dot{T}_{LB}\, \vec{r}_o^{\,B}, \qquad \text{(velocity)}$$

$$\hat{r}_o^{\,L} = \vec{r}_o^{\,L} / \left| \vec{r}_o^{\,L} \right| \qquad \text{(direction)}$$

and $\dot{\vec{r}}_o^{\,B}$ is provided by the parent simulation. Lastly, the angle-off boresight is computed using

$$(5.15) \qquad \theta_S = \cos^{-1} \left(r_{oz}^{\,L} / \left| \vec{r}_o^{\,L} \right| \right).$$

The second step of the detection control logic is to decide which detection model should be invoked. The rules for this decision are identical to those for Ku-Band Radar summarized in Section 5.1.3. Once this decision is made control is passed to the proper detection model to attempt a target detection.

CFAR Detection Algorithm. In the CFAR algorithm of Figure 5-22, the first step is to compute the SNR at the video filter output, neglecting the beam-shape and/or scan loss. In the sequel, this will be referred to as the nominal SNR_v. It is computed using the expression

$$(5.16) \qquad SNR_v = \frac{P_T G^2 \lambda_c^2}{(4\pi)^3 R^4 k L_T B_n T} \qquad \text{(All Passive Modes)}$$

where

P_T = peak transmit power,

G = one-way antenna gain,

λ_c = carrier wavelength,

σ = average radar cross section,

R = target range

k = Boltzmann's constant

L_T = transmit losses,

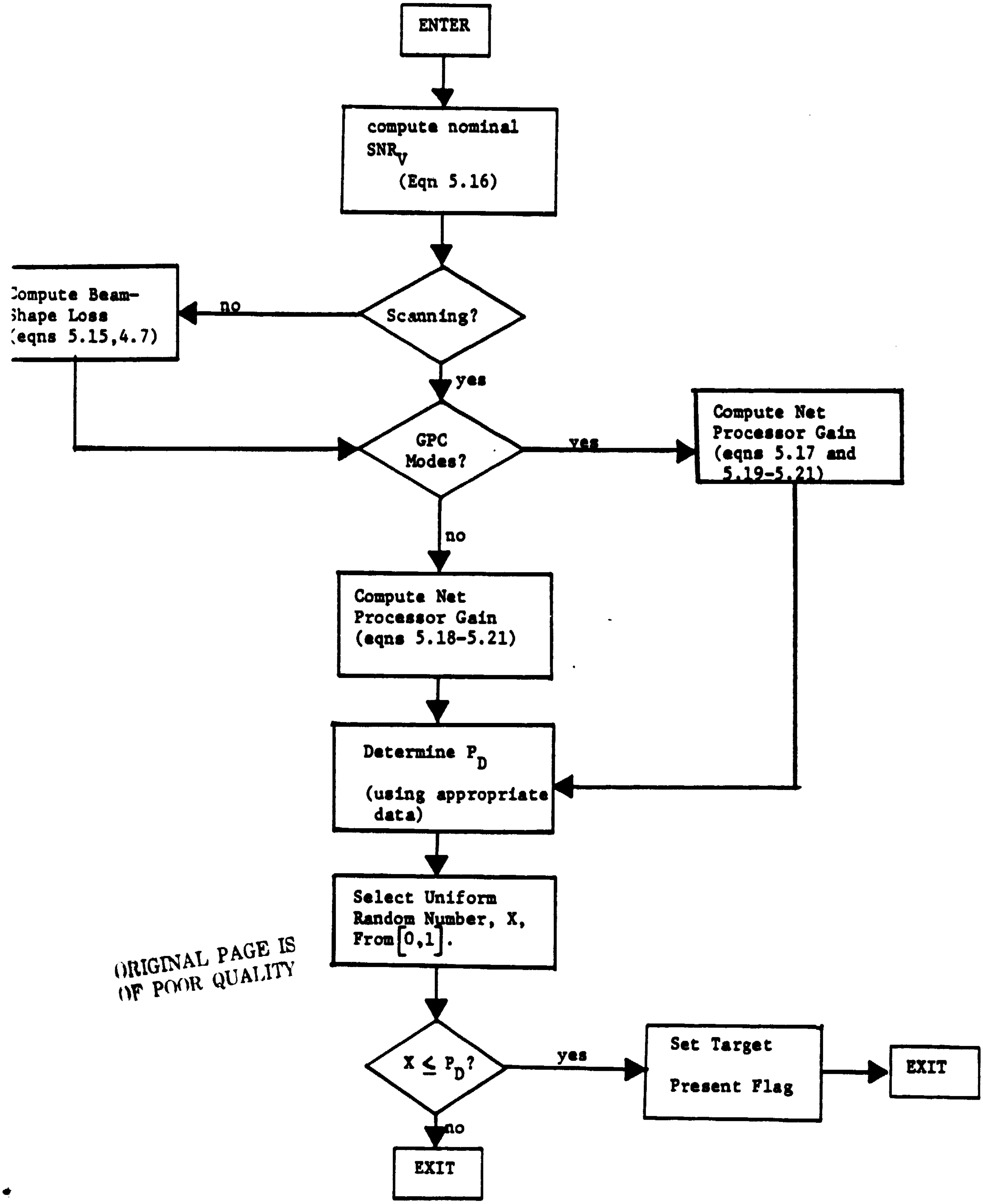
B_n = receiver noise bandwidth,

T = system noise temperature.

The next step is to compute the net gain of the processor from the baseband filter output to the doppler filter output and to combine it with the nominal SNR_v to form the nominal SNR_D. The net gain is comprised of (1) range gate loss, (2) net presum gain, and (3) net doppler filter gain. Each of these budget entries is expressed quantitatively below.

Of the three budget entries only the range gate loss differs for the GPC modes and the Auto and Manual Modes. Range gate loss for the GPC modes where overlapped range gates are used, includes the effects of misdesignation loss and widened range gates and is given by

Figure 5-22 CFAR DETECTION MODEL COMPUTER ALGORITHM


ENTER
compute nominal SNR$_V$ (Eqn 5.16)
Scanning?
no
Compute Beam-Shape Loss (eqns 5.15,4.7)
yes
GPC Modes?
yes
Compute Net Processor Gain (eqns 5.17 and 5.19-5.21)
no
Compute Net Processor Gain (eqns 5.18-5.21)
Determine P$_D$ (using appropriate data)
Select Uniform Random Number, X, From [0,1].
X ≤ P$_D$?
yes
Set Target Present Flag
EXIT
no
EXIT

ORIGINAL PAGE IS OF POOR QUALITY

$$(5.17) \qquad RGL = \begin{cases} 2/3\,, & \text{if } x \leq \tfrac{1}{2} \\ 2/3\ (3/2-x)^2, & \text{if } \tfrac{1}{2} \leq x < 3/2 \\ 0, & \text{if } x > 3/2 \end{cases} \qquad \text{(Range Gate Loss --- GPC Modes)}$$

where $x = \dfrac{2\left|R_G - R_o^L\right|}{ct_t}$,

R_G = target designated range (center of gates),

c = speed of light,

t_t = transmitted pulse width.

For the Auto and Manual modes, where the range gates are juxtaposed, this loss is given by

$$(5.18) \qquad RGL = \begin{cases} X^2 & , \text{ if } X < 1 \\ (1-\ (X-[X])\ )^2, & \text{if } 1 < X < 9/2 \text{ and } X-[X] < \tfrac{1}{2}, \\ (X-[X])^2 & , \text{ if } 1 < X < 9/2 \text{ and } X-[X] > \tfrac{1}{2} \end{cases} \qquad \text{(Range Gate Loss --- Auto and Manual Modes)}$$

where $X = \dfrac{2\ R_o^L}{ct_t}$

$[\bullet]$ = greatest integer in $\bullet$.

The net presum gain computation is identical for all antenna steering modes and includes the coherent gain of the presumming process and the loss due to doppler mismatch. This value is computed using the expression

$$(5.19) \qquad PSG = \frac{\sin^2(N_p y)}{\sin^2(y)\,N_p} \qquad \text{(Net presum gain)}$$

where $y = \pi f_d\ \tau_s$,

f_d = target doppler shift = $-\ \dfrac{2\ \dot{r}_o^L}{\lambda c}$,

τ_s = A/D sample interval,

N_p = number of A/D samples per pulse width.

The net doppler filter gain computation is identical for all antenna steering modes and includes the coherent gain of the doppler filter and the loss caused by filter straddling. It is computed from the expression

$$(5.20) \qquad DFG = \frac{\sin^2 (16z)}{16 \sin^2 (z)} \qquad \text{(Doppler Filter Gain)}$$

where $z = \pi (m/32 - f_d t_p)$,

t_p = PRI,

m = filter nearest f_d.

The net processor gain is obtained from

$$(5.21) \qquad \text{Net Processor Gain} = (RGL)(PSG)(DFG)$$

in all cases where a CFAR detector is required.

The third step is to decide whether the antenna is scanning or not. If the antenna is scanning we proceed to step four. If the antenna is not scanning the beamshape loss is computed, using equations (5.15) and (4.2), and combined with the nominal SNR_D.

In the fourth step, the value of SNR_D is used to address the look-up table, containing the proper P_D profile for the given conditions, to determine the P_D value in the present case. It is noted that if the computed nominal SNR_D falls between stored data points then linear (in dB) interpolation is used to obtain the P_D. This value of the P_D will implicitly contain an average beamshape and scan loss (if scanning), the losses associated with the magnitude detector, and the losses due to the thresholding technique.

The fifth and final step is to select a number x from a population which is uniformly distributed on the interval $\left[0,1\right]$, compare this value of x with the P_D from step (4) and make a target detection/no detection decision based on the algorithm of (5.11).

Single-Hit Detection Algorithm. The single-hit detection algorithm of Figure 5-23 is similar in form to the CFAR algorithm. That is, first the nominal SNR_v is computed using equation (5.16) for the passive modes or the expression

$$SNR_v = \frac{P_B G_B G \lambda_c^2}{(4\pi)^2 R^2 L_B k B_n T} \qquad \text{(All Active Modes)} \tag{5.22}$$

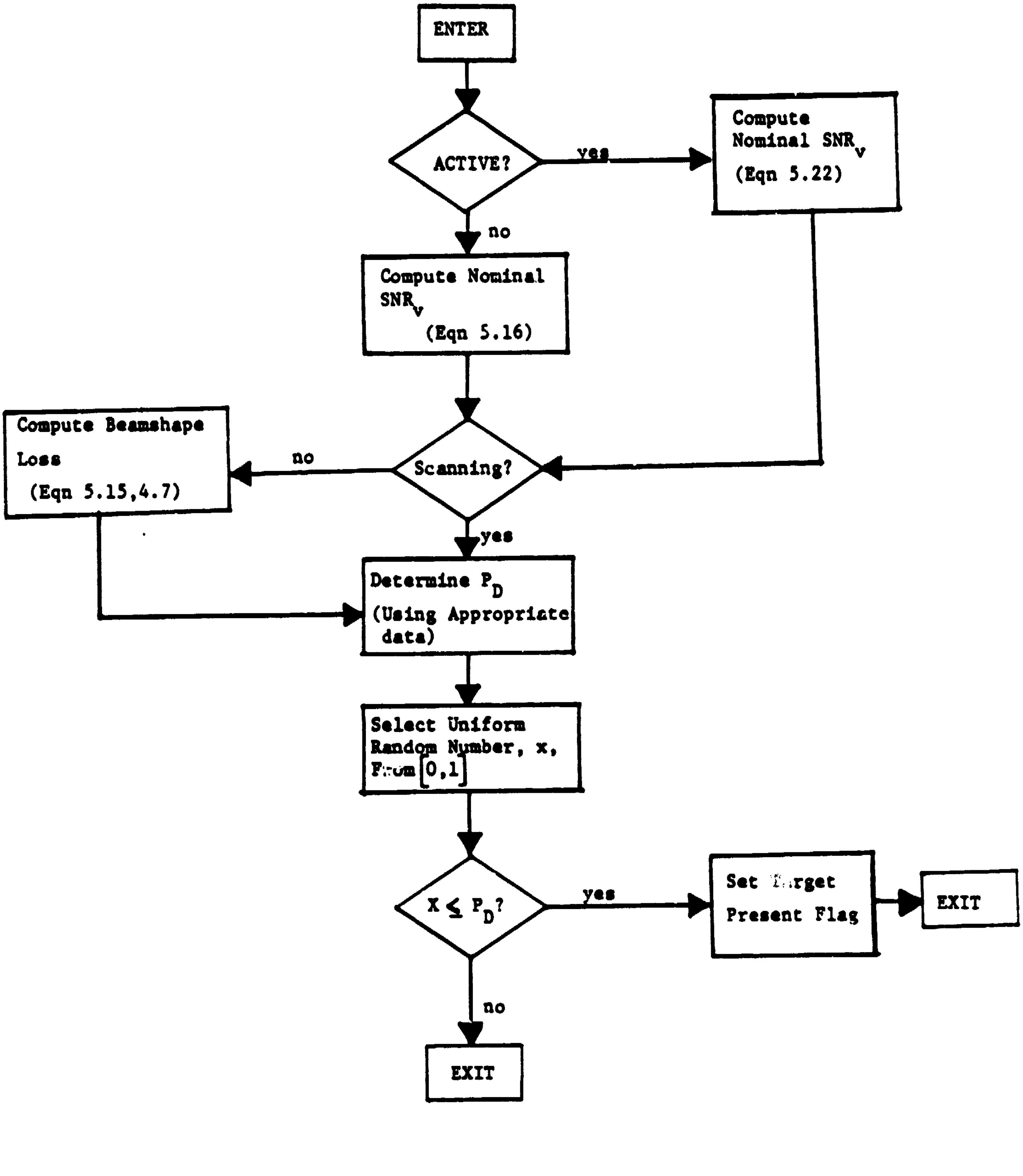
where P_B = Peak beacon transmit power,

G_B = one-way gain of the beacon antenna,

L_B = beacon transmit losses.

for the active modes. In the second step it is determined whether the antenna is scanning or not. If the antenna is scanning we proceed to step (3), but if it is not scanning then the beamshape loss is computed in the same manner as the CFAR case and combined with the nominal SNR_v. The fourth step is to determine the P_D associated with present value of SNR_v. This determination is identical to the CFAR case. The last step is to select a uniform random number x, compare it to P_D, and decide hit or miss with the algorithm of (5.11) as in the CFAR case.

Figure 5-23 SINGLE-HIT DETECTION MODEL COMPUTER ALGORITHM


ENTER
ACTIVE?
yes
Compute Nominal SNR$_v$ (Eqn 5.22)
no
Compute Nominal SNR$_v$ (Eqn 5.16)
Compute Beamshape Loss (Eqn 5.15,4.7)
no
Scanning?
yes
Determine P$_D$ (Using Appropriate data)
Select Uniform Random Number, x, From [0,1]
X ≤ P$_D$?
yes
Set Target Present Flag
EXIT
no
EXIT

6. TRACK MODE COMPUTER MODEL DESCRIPTION

An illustration showing the basic structure of the track mode computer algorithm is given in Figure 6-1. The structure of this algorithm is similar in form to the search mode computer algorithm of Figure 5-1. It consists of a main control program and several subprograms dedicated to various tracking functions. The main program, called the track mode control algorithm, has two purposes: (1) it controls initialization of all tracking loops and updating of the status of all data valid flags when control is first passed from search to track and (2) it controls the computation sequence required to update all tracking loops during the tracking phase. The initialization procedure requires two major subprograms. One is dedicated to initialization of the tracking loops when tracking first starts and the other subprogram decides when the various data valid flags should be raised, indicating the track estimates are accurate. The track loop update procedure requires several major subprograms which perform the following tasks: (1) target return signal generation and processing, (2) break-track determination, (3) angle and angle rate estimate updates, and (4) range and range rate estimate updates. In the following subsections, models for each of these functions are described, analysis is supplied whenever appropriate, and details of the computer algorithm for each function are given.

6.1 SUMMARY OF KU-BAND RADAR TRACK MODE OPERATION

6.1.1 General Antenna Steering Mode Operation

In this subsection, a short description of the Ku-Band Radar tracking procedure for each antenna steering mode is provided. For each steering mode, the general procedure is the same for active or passive targets. The only difference between active and passive modes are the transmit waveforms as discussed below.

GPC-ACQ Track Mode. In GPC-ACQ the radar performs target angle, inertial angle rate, range and range rate tracking. Angle and inertial angle rate tracking are accomplished using the amplitude comparison monopulse technique.

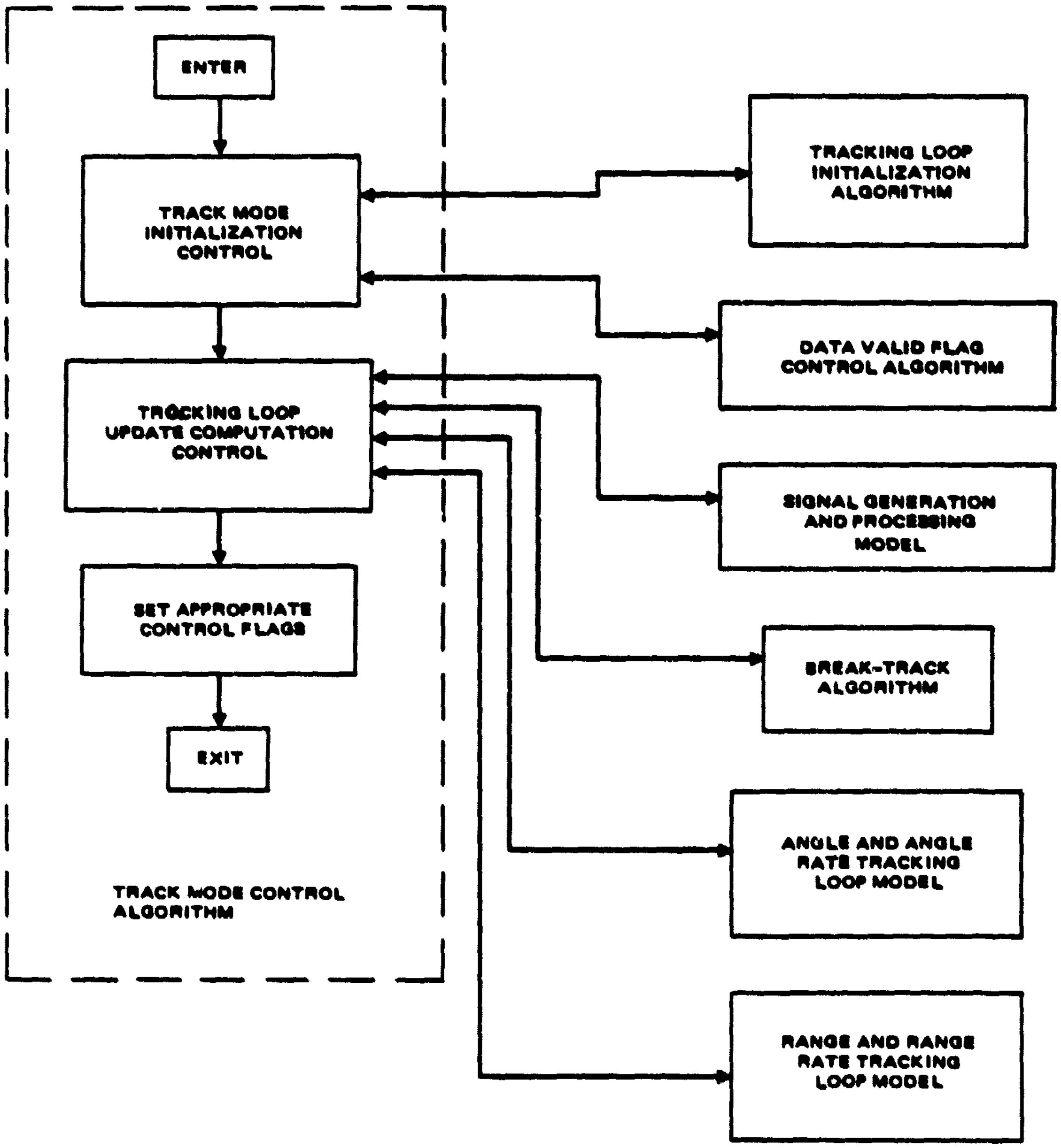

ENTER
TRACK MODE INITIALIZATION CONTROL
TRøCKING LOOP UPDATE COMPUTATION CONTROL
SET APPROPRIATE CONTROL FLAGS
EXIT
TRACK MODE CONTROL ALGORITHM
TRACKING LOOP INITIALIZATION ALGORITHM
DATA VALID FLAG CONTROL ALGORITHM
SIGNAL GENERATION AND PROCESSING MODEL
BREAK-TRACK ALGORITHM
ANGLE AND ANGLE RATE TRACKING LOOP MODEL
RANGE AND RANGE RATE TRACKING LOOP MODEL

Figure 6-1. Outline of track mode computer algorithm.

Range tracking is achieved by maintaining the return pulse centered in two juxtaposed range gates. Velocity tracking is performed using the algorithm described in section 6.7.

GPC-DES Track Mode. In this mode the radar tracks the target range and range rate only. Target angle tracking is performed by the GPC which supplies target angle designates to the gimbal pointing loop during tracking. There is no tracking of the target's inertial angle rate. Range and range rate tracking are identical to the GPC-ACQ mode.

Auto Track Mode. This mode is identical to the GPC-ACQ mode.

Manual Track Mode. In this mode, the radar performs range and range rate tracking using the same method as the other three steering modes. Angle tracking is performed by the crew using the antenna slew switches on the cockpit radar console. There is no inertial angle rate tracking in this mode.

6.1.2 Data Valid Flags

The data valid flag representing a given quantity will be raised when all transients in the loop tracking that quantity have settled out. The time allotted from tracker initialization to raising the data valid flag is precomputed based on maximum allowable errors in the quantities tracked and linearized loop models. Precomputed times for the angle, angle rate, range, and range rate data valid flags as a function of the loop bandwidth are summarized in Table 6-1 for active and passive operation.

The only data valid flag that is allowed to drop during tracking, without a break-track condition is the range rate data valid flag. Conditions under which this flag is lowered are (1) when the PRF is switched from 7 kHz to 268 Hz in the active track mode or (2) when the predicted target velocity moves to a new filter in two out of the last five update periods, including the present update period. In either of these cases it is raised again if the predicted velocity remains in the same doppler filter for 15 consecutive update periods.

6.1.3 Display Meters

Display meters are provided on the cockpit radar console for

Table 6-1 **DATA VALID FLAG TIMEOUTS (AFTER CLOSING TRACKING LOOPS) FOR ACTIVE AND PASSIVE MODES**

RANGE INTERVAL,nm	DATA VALID FLAG TIMEOUT, SECONDS		
	RANGE & RANGE RATE	ANGLE	ANGLE RATE
Passive Modes			
R < 3.8	6.97	1.02	8.2
3.8<R<7.2	6.97	1.02	26.23
7.2 <R	29.76	2.33	29.76
Active Modes			
R < 9.5	6.15	1.02	8.20
9.5 <R	28.69	5.12	28.69

- target roll and pitch angles,
- target range and range rate,
- target inertial roll and pitch rates,
- target signal strength,

The target signal strength meter which is zeroed during search and acquisition immediately becomes operational when the track mode is entered regardless of the antenna steering mode. Inertial roll and pitch rate meters operate only in GPC-ACQ and Auto steering modes; in these modes they become operational when track is first initialized. The target inertial rate meters are zeroed in GPC-DES and Manual steering modes. Range and range rate display meters become operational when the track mode is first entered. They are operational in all antenna steering modes. Roll angle and pitch angle display meters operate in all antenna steering modes and become operational when the track mode is first entered.

6.1.4 Break-Track Algorithm

The basic idea of the break-track algorithm is simple. If a no-target condition is obtained in five of the last eight update periods, including the present update period, then a break-track condition is declared and the system is returned to the search mode. The determination of a no-target condition is slightly more involved and is discussed in detail in section 6.5.

6.1.5 Track Waveforms

Passive Modes. The general track waveform for passive modes is illustrated in Figure 6-2. This waveform consists of five consecutive transmit frequency intervals with four time slots per frequency interval and 17 pulse repetition intervals (PRI) per time slot. For a given transmit frequency the receiver dedicates each of the four time slots to the following information:

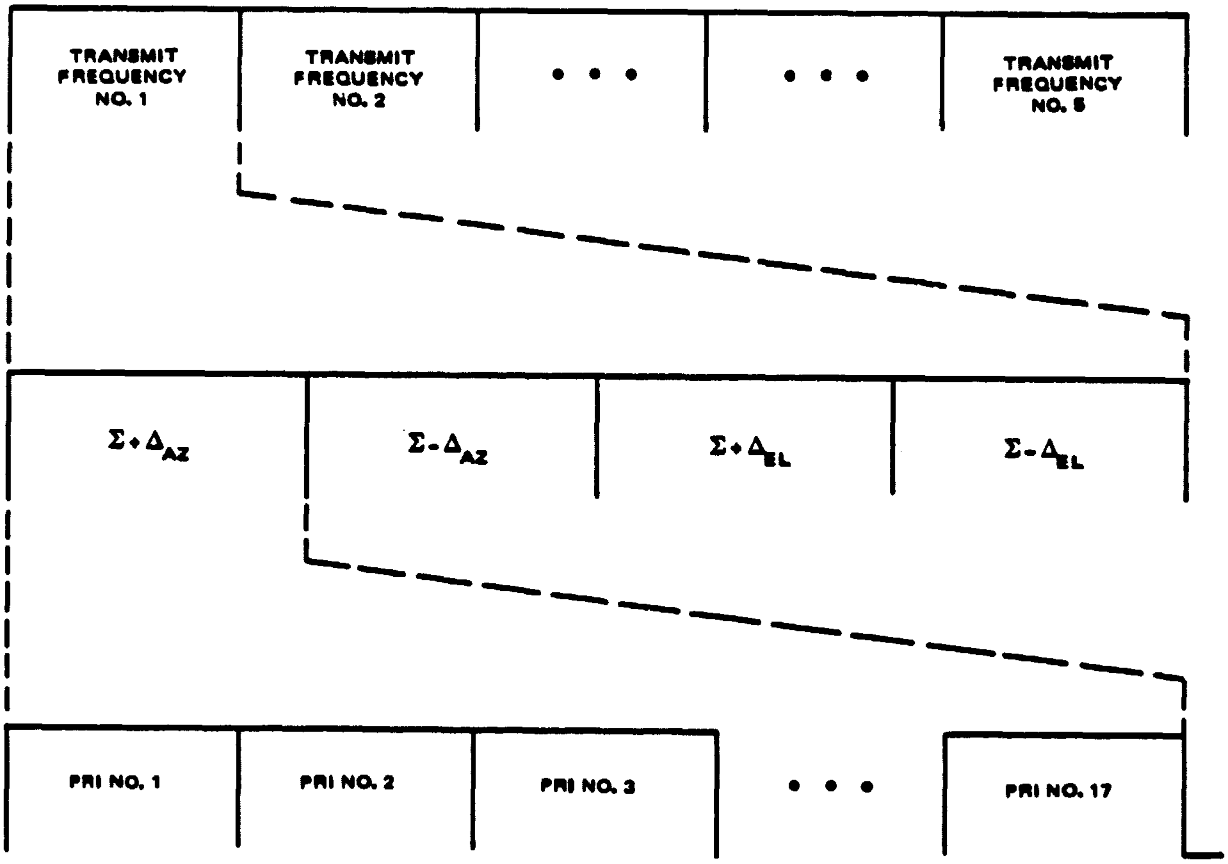


Figure 6-2. Waveform for Passive Track Modes.

Slot 1 = (Sum channel output) + (Azimuth Difference Channel Output)

Slot 2 = (Sum Channel Output) - (Azimuth Difference Channel Output

Slot 3 = (Sum Channel Output) + (Elevation Difference Channel Output)

Slot 4 = (Sum Channel Output) - (Elevation Difference Channel Output)

The receiver processes 16 pulses for each of these time slots. The waveform parameters are a function of range and are summarized in Table 6-2.

Active Modes. The general track waveform for all active modes is illustrated in Figure 6-3. For these modes only one transmit frequency interval is used. This interval is divided into four time slots with 17 PRI per time slot as in the passive modes. The waveform parameters are listed in Table 6-3.

6.1.6 Tracking Loops and Signal Processor Operation

The signal processor configuration is described in Section 6.4, the angle and angle rate tracking loops are described in Section 6.6, and the range and range rate tracking loops are described in Section 6.7.

6.2 TRACK MODE CONTROL ALGORITHM DESCRIPTION

The track mode algorithm is entered immediately upon detection of a target in the search mode. A detailed block diagram of the track mode control algorithm is given in Figure 6-4. As noted in the introduction to this section this subroutine has two functions: (1) to control tracking loop initializations and (2) to control the computation of tracking loop estimates. These two functions are described below.

6.2.1 Track Mode Initialization Control

The first task is to initialize each of the tracking loops. This means initial values for the target parameters being tracked must be computed to allow

Table 6-2 WAVEFORM AND SIGNAL PROCESSING PARAMETERS FOR PASSIVE TRACK MODES

RANGE INTERVAL,nm	PULSE WIDTH, μsec	PRF, hz	SAMPLE INTERVAL, μsec	SAMPLES PER RANGE BIN
$R < 0.42$	0.122	6970	0.122	1
$0.42 \leq R < 0.95$	2.07	6970	2.075	1
$0.95 \leq R < 1.9$	4.15	6970	2.075	2
$1.9 \leq R < 3.8$	8.3	6970	2.075	4
$3.8 \leq R < 7.2$	16.6	6970	2.075	8
$7.2 \leq R < 9.5$	16.6	6970	2.075	8
$9.5 \leq R < 18.9$	33.2	2987	2.075	16

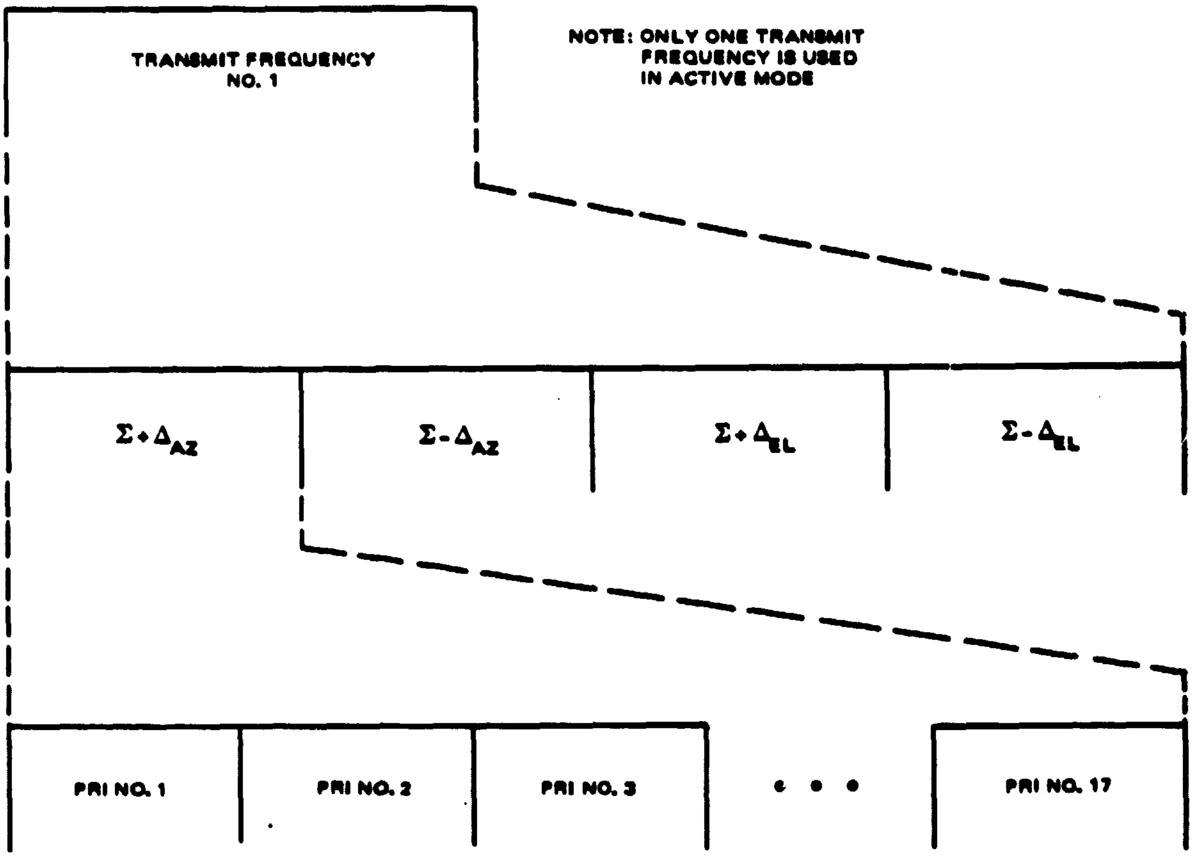
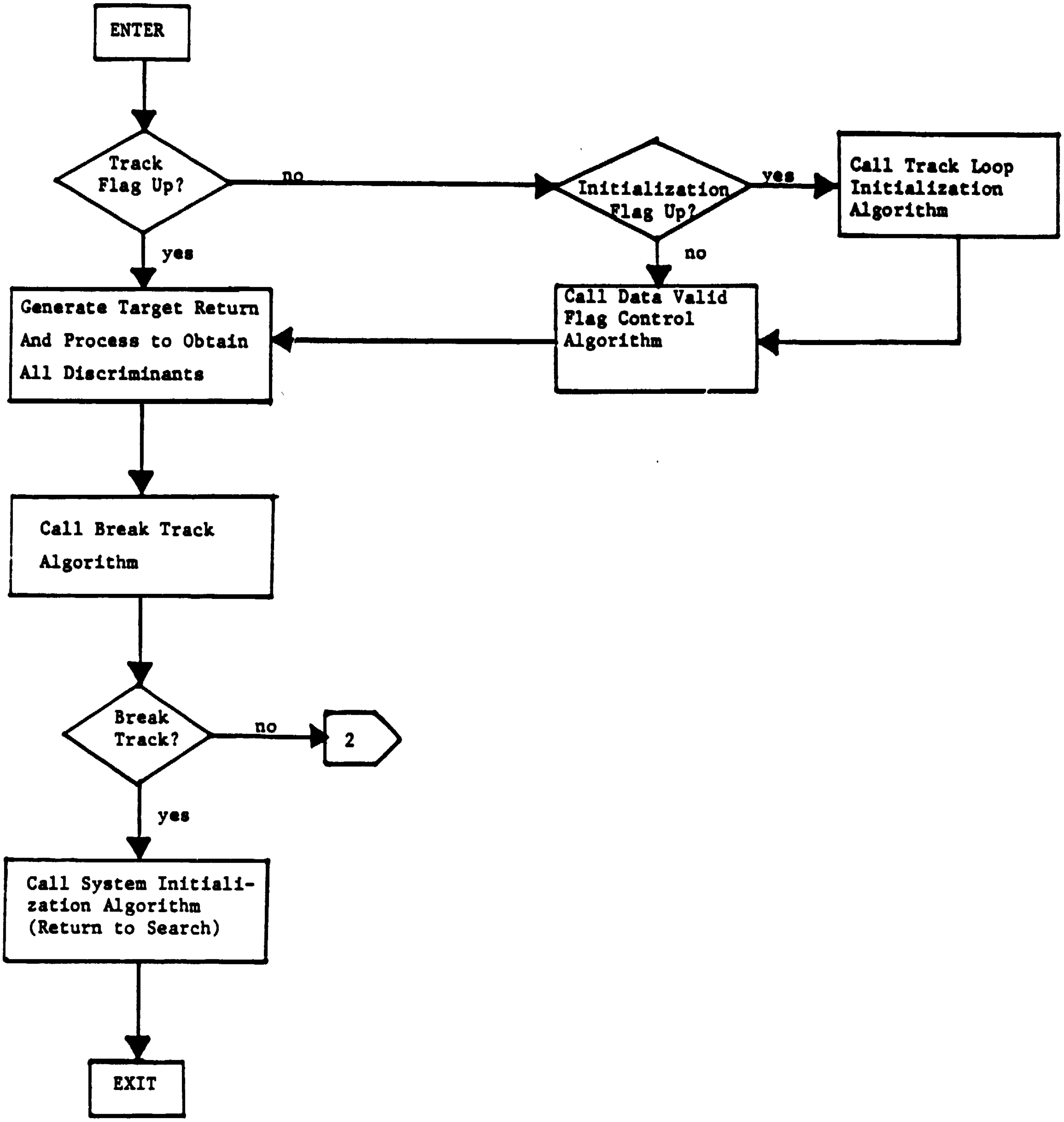


Figure 6-3. Waveform for Active Track Modes.

Table 6-3 WAVEFORM AND SIGNAL PROCESSING PARAMETERS FOR ACTIVE TRACK MODES

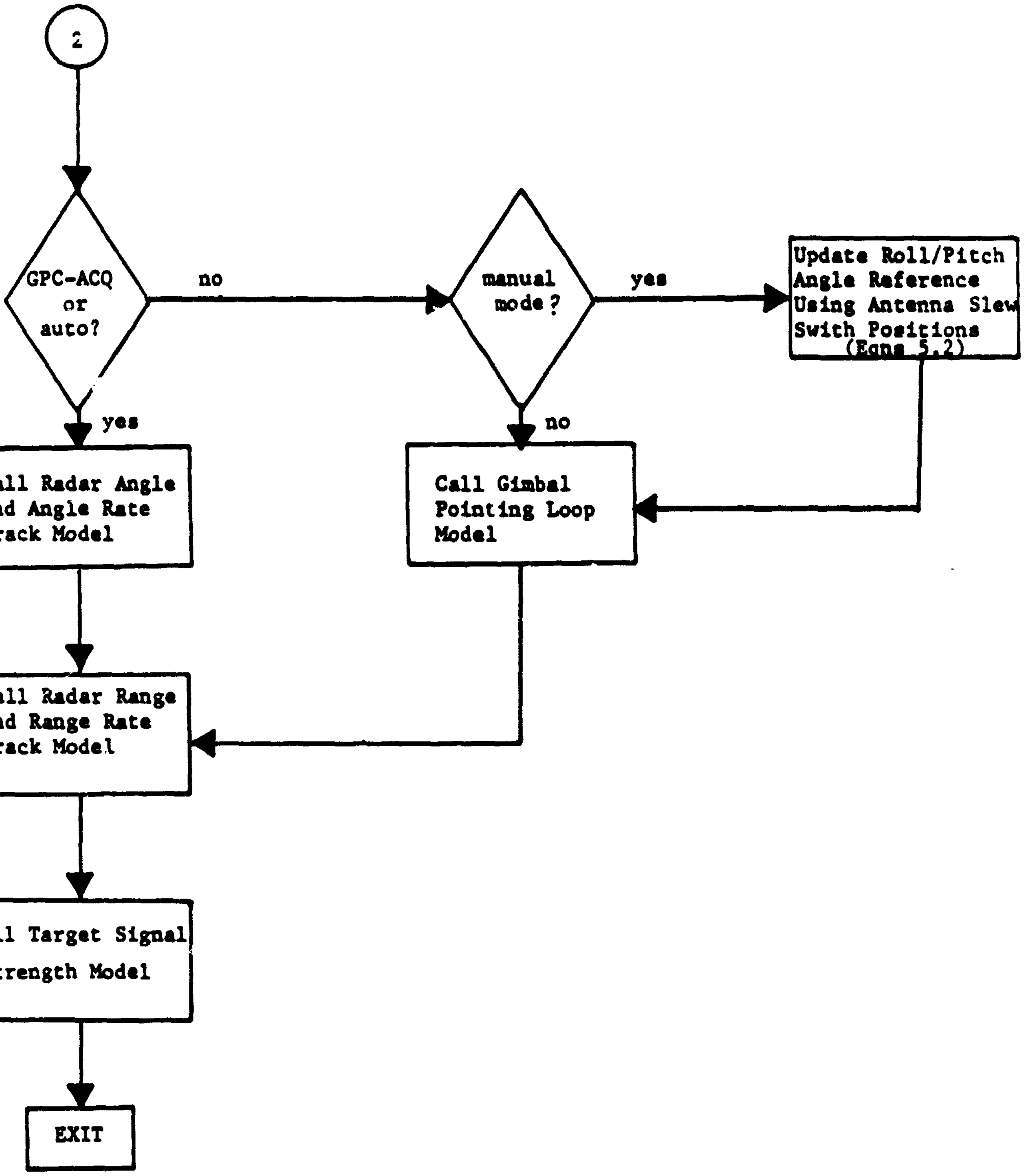
RANGE INERVAL, nm	PULSE WIDTH, μsec	PRF, hz	SAMPLE INTERVAL, μsec	SAMPLE PER RANGE BIN
$R < 9.5$	0.122	6970.	0.122	1
$R \geq 9.5$	4.15	268.	2.075	2

Figure 6-4 TRACK MODE CONTROL COMPUTER ALGORITHM
(1 of 2)



ENTER
Track Flag Up?
no
Initialization Flag Up?
yes
Call Track Loop Initialization Algorithm
yes
no
Generate Target Return And Process to Obtain All Discriminants
Call Data Valid Flag Control Algorithm
Call Break Track Algorithm
Break Track?
no
2
yes
Call System Initialization Algorithm (Return to Search)
EXIT

Figure 6-4 TRACK MODE CONTROL COMPUTER ALGORITHM (2 of 2)


2
GPC-ACQ or auto?
no
manual mode ?
yes
Update Roll/Pitch Angle Reference Using Antenna Slew Swith Positions (Eqns 5.2)
yes
Call Radar Angle And Angle Rate Track Model
no
Call Gimbal Pointing Loop Model
Call Radar Range and Range Rate Track Model
Call Target Signal Strength Model
EXIT

the difference equation representations of the loops to begin tracking the parameter changes. This initialization is done during the first update period after control has been passed from the search algorithm to the track algorithm. The choice of parameter initializations and the equations that compute these values are discussed in detail in Section 6.3 and Appendix A.

The other task is to initialize the clock used to time the data valid flags and continue to update this clock until the appropriate data valid flags for a given antenna steering mode are all raised. Clock initialization is performed in the first update period after control has been passed to the track algorithms. The subprogram used by the track mode control algorithm to perform these tasks is shown in Figure 6-5 and the data valid flag timeout periods are given in Table 6-1.

6.2.2 Tracking Loop Update Control

The other responsibility of the track mode control algorithm is to control the computations leading to updated estimates by the various tracking loops. This is a complex procedure involving several steps and is outlined below. The first step is to generate a target return signal, based on the latest target - radar configuration and process this signal to produce error signals in the form of discriminants to be used by the appropriate tracking loop models. A set of four subprograms are required to perform this computation. Complete details of this package of algorithms will be given in Section 6.4.

The second step in the update procedure is to check for a break-track condition. This is done using the algorithm described in section 6.5. If a break-track condition is obtained, then the system is reset to the search mode using the algorithm shown in Figure 6-6. If a break-track condition is not obtained then the computation sequence proceeds to the third step which is to update the antenna position and the target inertial angle rates(if appropriate). If the antenna steering mode is GPC-ACQ or Auto, then the target roll and pitch angles, i.e.

Figure 6-5 DATA VALID FLAG CONTROL ALGORITHM

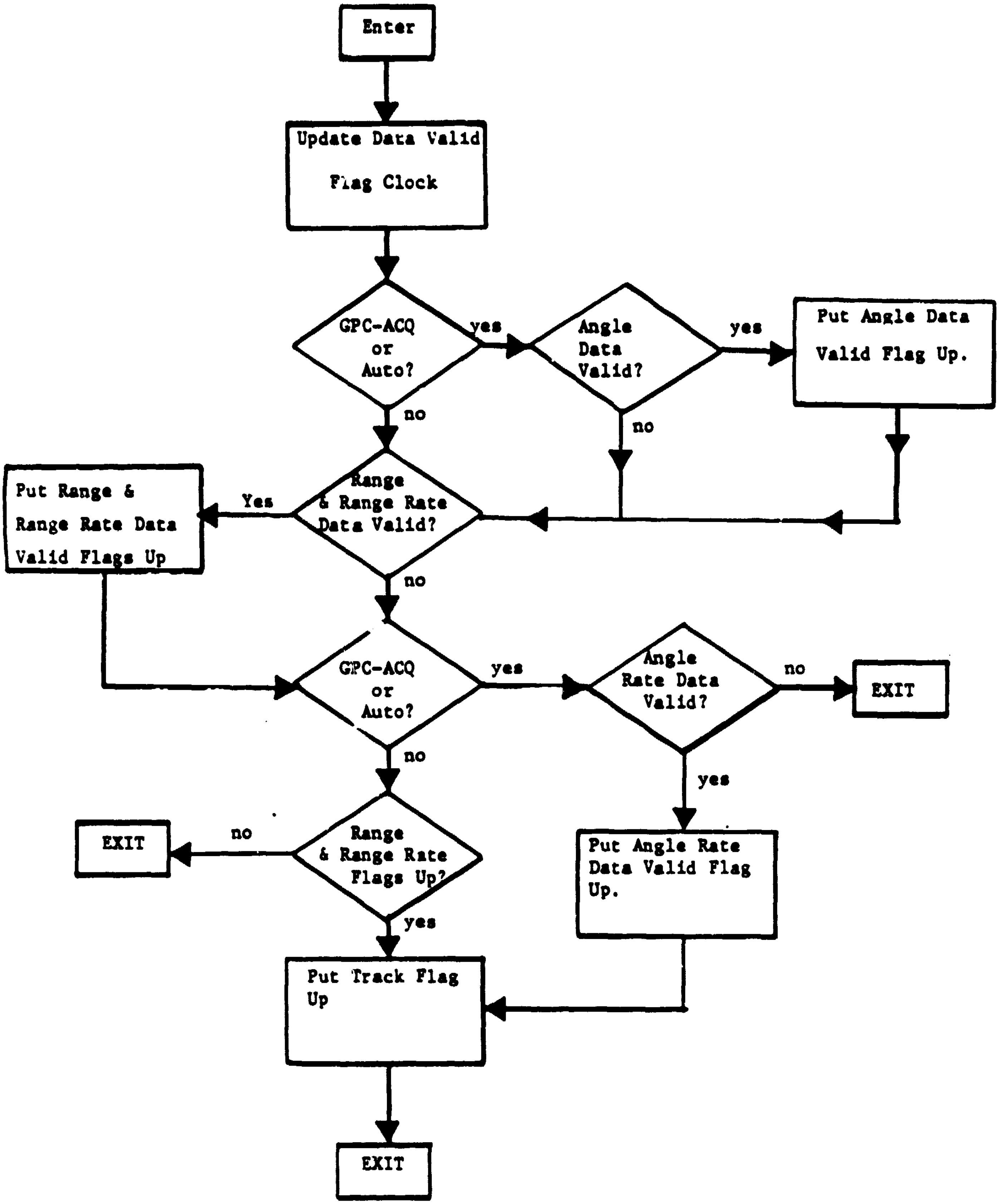
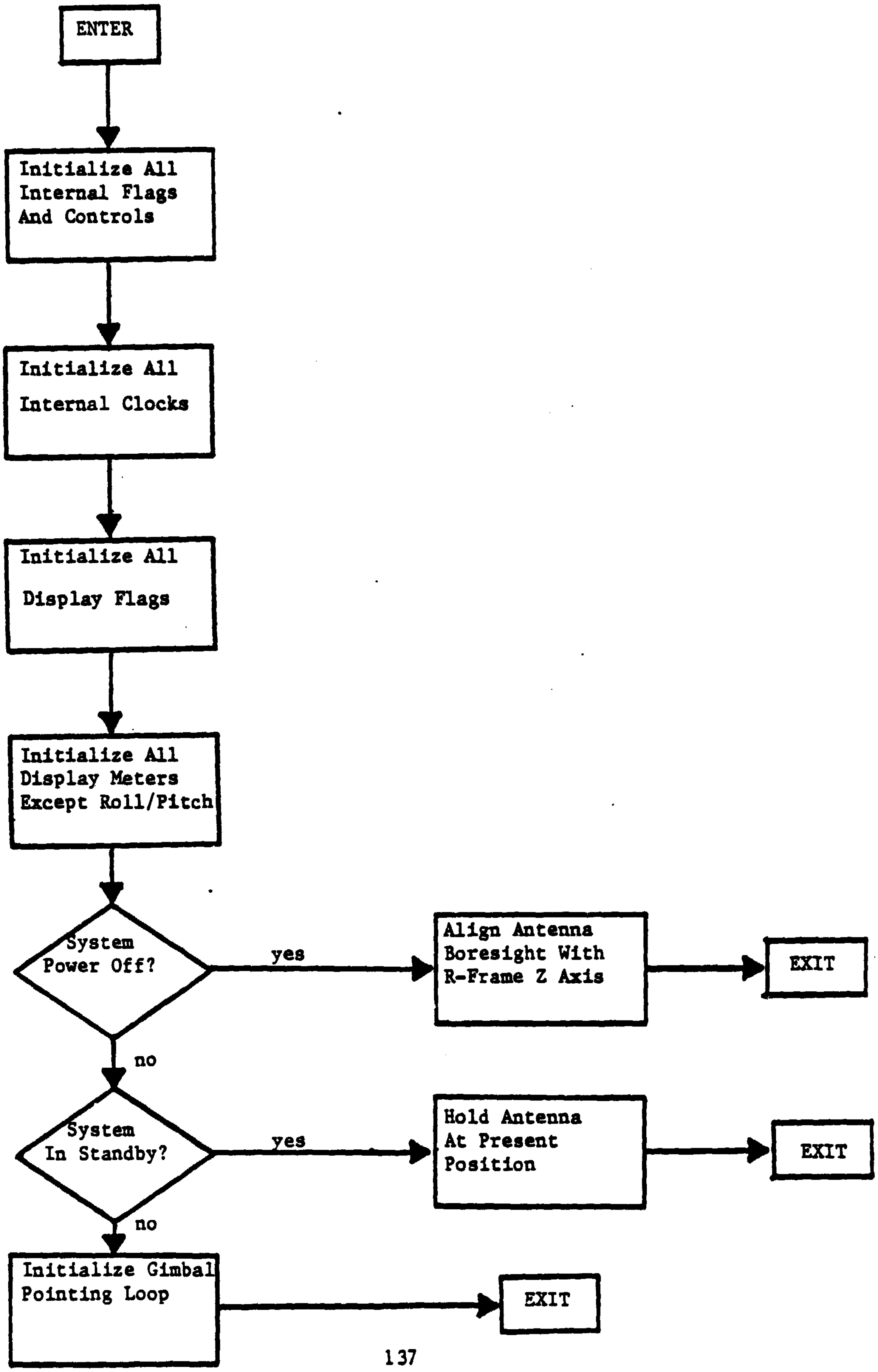

Enter
Update Data Valid Flag Clock
GPC-ACQ or Auto?
yes
Angle Data Valid?
yes
Put Angle Data Valid Flag Up.
no
no
Put Range & Range Rate Data Valid Flags Up
Yes
Range & Range Rate Data Valid?
no
GPC-ACQ or Auto?
yes
Angle Rate Data Valid?
no
EXIT
no
yes
EXIT
no
Range & Range Rate Flags Up?
Put Angle Rate Data Valid Flag Up.
yes
Put Track Flag Up
EXIT

Figure 6-6 SYSTEM INITIALIZATION ALGORITHM


ENTER
Initialize All Internal Flags And Controls
Initialize All Internal Clocks
Initialize All Display Flags
Initialize All Display Meters Except Roll/Pitch
System Power Off?
yes
Align Antenna Boresight With R-Frame Z Axis
EXIT
no
System In Standby?
yes
Hold Antenna At Present Position
EXIT
no
Initialize Gimbal Pointing Loop
EXIT

the antenna position, and the inertial roll and pitch rate estimates are updated using the model described in section 6.6. If the system is in the GPC-DES mode, the antenna gimbal pointing loop (section 5.2) is updated using the latest roll and pitch designates from the GPC. If the system is in the Manual mode, the antenna gimbal pointing loop is updated using the latest positions of the antenna slew switches on the cockpit radar console and equations (5.2). Target inertial roll and pitch rates are not tracked in the GPC-DES and Manual modes.

In the fourth step, the target range and velocity estimates are updated using the model described in section 6.7. This step is performed in all antenna steering modes. The fifth and final step is to compute an estimate of the target signal strength using the algorithm described in section 6.4.

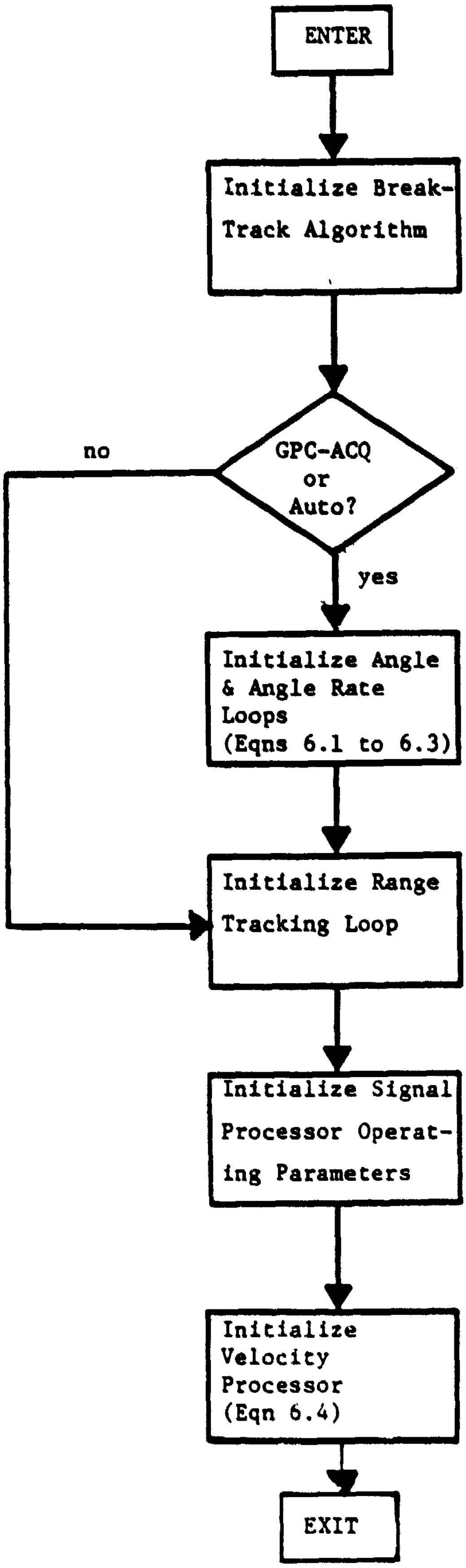
6.3 TRACKING LOOP INITIALIZATION ALGORITHM DESCRIPTION

This subsection gives a detailed description of the algorithm, illustrated in Figure 6-7, used to compute the intial state of each tracking loop for a given antenna steering mode. The basic philosophy is to set the initial states of the angle, angle rate, range and range rate tracking loops equal to the respective values of the target c.g. parameters in the update period in which initialization takes place. The general procedure is to initialize each of the following items:

- Break-track algorithm,
- Angle and angle rate tracking model (if required),
- Range tracking model,
- Parameters for signal processor,
- Velocity processor model,
- Signal strength algorithm,

in the order shown. Initialization of each item is described in detail below.

Figure 6-7 TRACKING LOOPS INITIALIZATION ALGORITHM


ENTER
Initialize Break-Track Algorithm
GPC-ACQ or Auto?
no
yes
Initialize Angle & Angle Rate Loops (Eqns 6.1 to 6.3)
Initialize Range Tracking Loop
Initialize Signal Processor Operating Parameters
Initialize Velocity Processor (Eqn 6.4)
EXIT

6.3.1 Break-Track Algorithm Initialization

This initialization requires setting the break-track flag low and zeroing the registers used to track the number of no-target conditions obtained in the previous 7 update periods.

6.3.2 Angle and Angle Rate Tracking Model Initialization

This model is used only in the GPC-ACQ and Auto antenna steering modes. The initial α and β gimbals positions, the initial α and β gimbal rates, and the initial target inertial LOS azimuth and elevation rate must be computed in order to start the tracking loops. These values are initialized using the following procedure. First, the positions of the α and β gimbals are determined so that the antenna boresight points directly at the target c.g. using the equations

$$\beta = -\tan^{-1} (r_{ox}^{R} /s)$$

(6.1)

$$\alpha = -\tan^{-1} (r_{oy}^{R} / r_{oz}^{R})$$

where $\vec{r}_o^{\,R} = T_{RB} (\vec{r}_o^{\,B} - \vec{x}^B),$

$\vec{x}^B$ = Radar offset from orbiter body C.G. expressed in body coordinates,

$s^2 = (r_{oy}^{R})^2 + (r_{oz}^{R})^2.$

In the next step, the target inertial LOS azimuth and elevation rate tracking loops are initialized using the expressions

$$w_{Tx}^{L} = v_{oy}^{L} / |\vec{r}_o^{\,L}| + w_{Bx}^{L}$$

(6.2)

$$w_{Ty}^{L} = -V_{ox} / |\vec{r}_o^{\,L}| + w_{By}^{L}$$

where $\vec{v}_o^{\,L}$ = target velocity measured in the B-frame and expressed in L-frame coordinates,

$\vec{w}_T^{\,L}$ = target inertial angular velocity expressed in L-frame coordinates,

$\vec{w}_B^{\,L}$ = body inertial angular velocity expressed in L-frama coordinates.

These relations are derived in Appendix A.

In the final step, the initial rates of the α and β gimbals are computed using the expressions

$$\dot{\alpha} = v_{oy}^{L} \,/\, (\,|\vec{r}_o^{\,L}|\cos\beta)$$

$$\dot{\beta} = w_{Ty}^{L} - w_{By}^{L} \tag{6.3}$$

which are also derived in Appendix A.

6.3.3 Range Tracking Model Initialization

The range tracker is an α - β tracker (see section 6.7 for details) that generates an estimate of the target range and range rate at each update period. It is initialized by setting the first range and range rate estimates equal to the target c.g. range and velocity, respectively. For the range this is accomplished by digitizing the CG range so that the least significant bit (LSB) represents 5/16 feet and loading it into the digital integrator that produces the range estimate at its output (see figure 6-32). For the range rate it is accomplished by digitizing the CG velocity so that the LSB represents $5/(16t_s)$ feet per second, where t_s is the update interval, and loading it into the digital integrator that produces the smoothed range rate estimate at its output (as shown in Figure 6-32).

6.3.4 Signal Processor Parameter Initialization

Several signal processor and tracking loop constants change with a different target range interval, processor A/D sample rate, and PRF. Therefore, the internal controls MRNG, MSAM, and MPRF were defined to apprise the system of changes in the target range interval, the A/D sample rate, and the PRF, respectively. These controls are defined in Table 6-4 and they are intialized as follows. MRNG is determined using the CG range and MSAM and MPRF are determined using MRNG and the system mode switch (IMODE) position.

6.3.5 Velocity Processor Model Initialization

The velocity processor tracks the target velocity by using five adjacent doppler filters, always maintaining the target in the center filter. The predicted velocity estimate in the present update period is averaged with the velocity estimate from the three previous update periods to obtain the final velocity estimate, i.e. the velocity is smoothed using the moving window average technique. Thus, initialization of the velocity processor involves (1) setting each entry of the array used for averaging equal to the C.G. velocity and determining the location of the center filter (of the five filter bank) using the equation

$$m_c = \mathrm{mod}\left(\left[\vec{V}_o \;/\Delta + 0.5 \quad, 32\right]\right) \tag{6.4}$$

where mod (. , 32) = modulo 32,

$[\bullet]$ = greatest integer in . ,

$\Delta = \lambda_c$ PRF/64.

m_c = number of the center doppler filter.

See section 6.7 for complete details of the velocity processor model.

6.3.6 Signal Strength Algorithm Initialization

The model which is used to compute the radar signal strength in the present version of the computer simulation is quite simple and does not require initialization.

Table 6-4 DEFINITION OF INTERNAL CONTROL PARAMETERS

RANGE INTERVAL,nm	MRNG	MSAM	MPRF
Passive Modes:			
0-120 ft	1	1	1
120-240 ft	2	1	1
240-720 ft	3	1	1
720ft-0.42	4	1	1
0.42-0.95	5	2	1
0.95-1.9	6	2	1
1.9-3.8	7	2	1
3.8-7.2	8	2	1
7.2-9.5	9	2	1
9.5-18.9	10	2	2
Active Modes:			
0-120 ft	1	1	1
120-240 ft	2	1	1
240-720 ft	3	1	1
720ft-0.42	4	1	1
0.42-0.95	5	1	1
0.95-1.9	6	1	1
1.9-3.8	7	1	1
3.8-7.2	8	1	1
7.2-9.5	9	1	1
9.5-18.9	10	2	3

6.4 SIGNAL GENERATION AND PROCESSING MODEL DESCRIPTION

This subsection gives a detailed description of the model used to generate the target return signal and process this signal to obtain all of the discriminants. The objective of this model is to generate the most accurate discriminant estimates possible given the present target scattering model selection and the constraint of real-time computer operation. The method selected to achieve this goal is heavily dependent upon the fact that the target is modeled as a collection of point scatterers and can be roughly outlined as follows. Instead of forming the target return signal at the antenna output and processing the resultant signal on a sample-by-sample basis using the exact Ku-band radar processing configuration shown in Figure 6-8, the processor model uses assumed linearity of the processor from the antenna to the doppler filter output and the assumptions listed in section 6.4.1 to compute the resultant signal at the doppler filter output in closed-form. Then, except for replacing the magnitude detector by a magnitude-squared detector, the remainder of the signal processing model is identical to the corresponding Ku-band radar processor functions. This computation model is illustrated in Figure 6-9. By using this model, sample-by-sample processing can be abandoned, thereby reducing the computation time per update cycle significantly without sacrificing signal processing model accuracy.

In the remainder of this subsection, we will present: (1) the model assumptions, (2) the technique for updating the position and motion of the point targets, (3) complete details of all discriminant generations, including the thermal noise model, (4) the radar signal strength computation algorithm, and (5) the computer model details.

Figure 6-8 SIMPLIFIED DIAGRAM OF KU BAND RADAR TRACK MODE SIGNAL PROCESSING

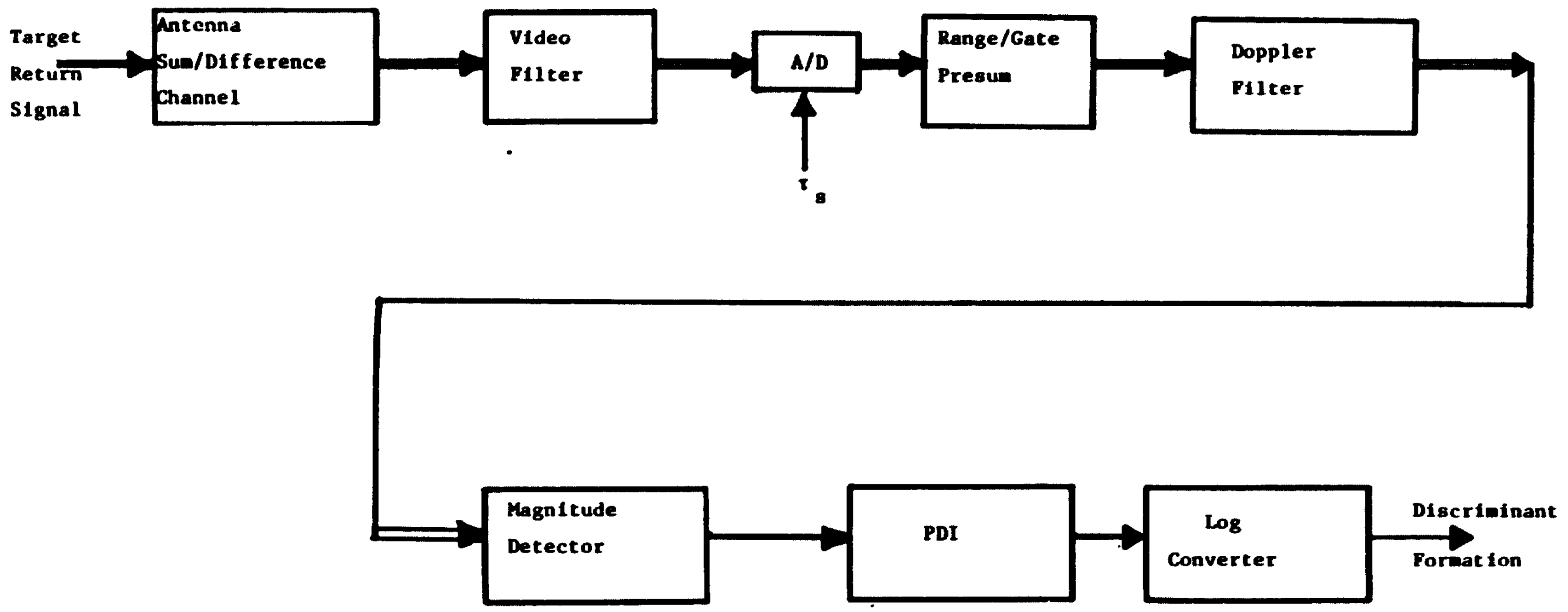
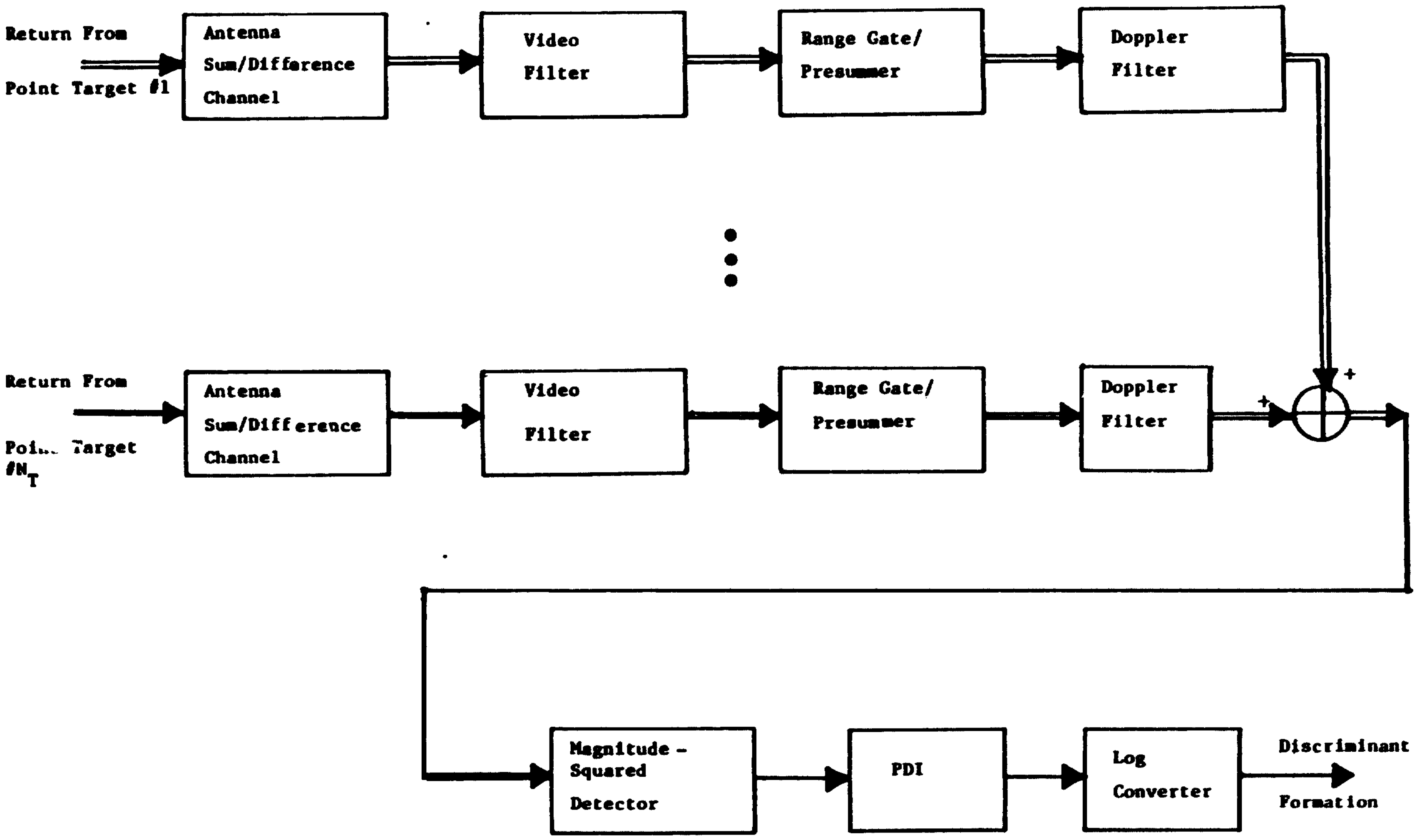


Figure 6-9 TRACK MODE SIGNAL PROCESSOR COMPUTER MODEL



6.4.1 Model Assumptions

The signal generation and processing model is based upon the assumptions listed below. Target assumptions are that

(1) the target is composed of a collection of point scatterers with the properties described in section 4,

(2) radial acceleration of the point targets during a data cycle is ignored.

Radar assumptions are that

(3) the waveform described in Figures 6-2 and 6-3 are transmitted without any distortion,

(4) the antenna does not move with respect to the target during the data cycle,

(5) the receiver's RF and IF electronics work perfectly, (i.e. the down conversion is error-free and the filters do not distort the return signal, but the receiver maintains the same noise figure and noise bandwidth).

(6) the baseband (video) filter has a perfect rectangular impulse response of width equal to the A/D sample interval,

(7) the A/D is treated as an ideal zero-order sample and hold,

(8) quantization noise contributed by the signal processing chain from the A/D to the log converter (see Figure 6-8) is neglected,

(9) the magnitude detector was replaced by a magnitude-squared detector,

(10) Automatic Gain Control (AGC) is not implemented.

Motivation for assumption (1) was discussed in section 4. Assumptions (2) and (4) are forced on us by the real-time processing constraint which rules out sample-by-sample or even pulse-by-pulse processing of the target return signal. Assumption (3) will have little effect on processor accuracy. On the other hand, assumption (5) can have a significant impact upon the fidelity of the angle tracking estimates because the difference channel coupling losses are ignored. The baseband filter assumption will have little effect upon processing accuracy. Impact of assumptions (7) and (8) is not known at the present; if time permits, an equivalent quantization noise will be added to the discriminant computation model. The magnitude-squared detector assumption will have only a slight impact upon the model accuracy. The system AGC will be implemented if time permits.

6.4.2 Target Position and Motion Computation Model

Generation of the discriminants in a given data cycle requires a knowledge of each point target's position and radial velocity with respect to the LOS frame. To obtain these quantities, we utilize the following information. Firstly, the parent simulation provides (1) $\vec{r}_o^{\,B}$ and $\vec{v}_o^{\,B}$, the present position and velocity of the target C.G. with respect to the orbiter body frame, and (2) T_{B_oT} and $\dot{T}_{B_oT}$, which describe the rotation of the T-frame with respect to the B -frame as discussed in sections 2 and 3. Secondly, the radar simulation tracks the angular position and rates of the antenna relative to the orbiter body frame. From these two facts, the position of the kth point target, located at an arbitrary but known position in the T-frame, can be computed in the L-frame using the expression,

$$\vec{r}_k^{\,L} = T_{LB}(\vec{r}_o^{\,B} + T_{B_oT}\,\vec{r}_k^{\,T} - \vec{X}^B)$$

or, regrouping the terms,

$$(6.5)\quad \vec{r}_k^{\,L} = T_{LB}(\vec{r}_o^{\,B} - \vec{X}^B) + T_{L_oT}\vec{r}_k^{\,T}$$

where $\vec{X}^B$ = vector describing the offset of the radar from the orbiter C.G.,

$T_{LB}(\vec{r}_o^{\,B} - \vec{X}^B)$ = position of target C.G. in the LOS frame.

The velocity of the arbitrary point target as measured in the LOS frame, can be obtained by time differentiating equation (6.5) and noting that $\vec{X}^B$ and $\vec{r}_k^{\,B}$ are constant vectors. This gives

$$(6.6)\quad \vec{v}_k^{\,L} = \left[\dot{T}_{LB}(\vec{r}_o^{\,B} - \vec{X}) + T_{LB}\dot{\vec{r}}_o^{\,B}\right] + \dot{T}_{L_oT}\,\vec{r}_k^{\,T}$$

where the expression in the square brackets is the velocity of the target C.G. as measured in the L-frame and expressed in L-frame coordinates. Finally, the target radial velocity as measured in the L-frame is obtained by computing the component of velocity in the direction of the radar. Quantitatively, this can be expressed as

$$(6.7)\quad v_k^L\ (\text{radial}) = \vec{v}_k^{\,L} \cdot \hat{r}_k^{\,L}$$

where $\hat{r}_k^{\,L}$ = unit vector in the direction of the target.

6.4.3 Angle Discriminant Computation Model

The angle discriminant is essentially formed by comparing the sum channel plus the difference channel signal to the sum channel minus the difference channel signal where both signals are appropriately integrated over the five transmit frequencies and two range bins. In the sequel, :hese two quantities will be referred to as the components of the angle discriminant. With this in mind, we proceed to a description of the angle discriminant computation model which is divided into three parts:

(1) computation of the noise-free discriminant components,

(2) computation of the equivalent thermal noise,

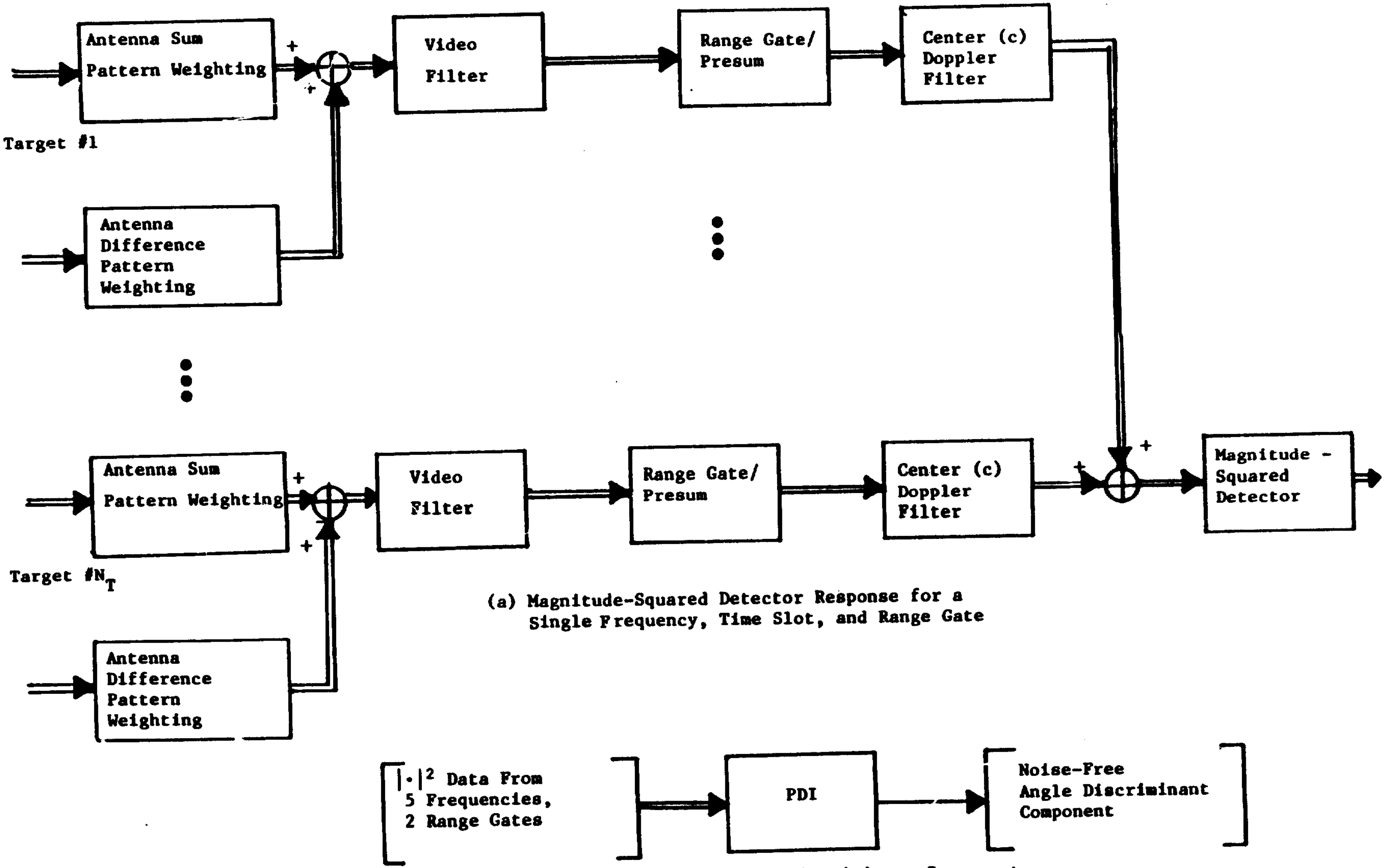
(3) computation of the angle discriminant.

Noise-Free Discriminant Component Computation. Figure 6-10 gives a block diagram of the model used to compute the noise-free discriminant components; this model is derived in Appendix C. Figure 6-10a shows the computation of the target response at the magnitude-squared detector output for a given transmit frequency and range bin. Figure 6-10b illustrates the post-detection integration (PDI) of the detector output over frequency and range bin to form the noise-free discriminant component. A detailed description of these steps is given below.

The total response of the target return at the doppler filter output is computed using the assumptions listed in section 6.4.1 and the assumption that the receiver/signal processor configuration is linear from the antenna to the doppler filter output. First, the response of each point target at the doppler filter output is computed in closed-form. Then, using the linearity assumption and the superposition principle, the resultant response for the complete target is computed by vectorially summing the individual responses.

Computation of the doppler filter response for a single point target requires a more detailed explanation. This computation is performed as follows. Using assumptions (1) through (5) and equation (4.1), the return

Figure 6-10 NOISE-FREE ANGLE DISCRIMINANT COMPONENT COMPUTATION MODEL


Antenna Sum Pattern Weighting
Target #1
Antenna Difference Pattern Weighting
+
+
Video Filter
Range Gate/ Presum
Center (c) Doppler Filter
Antenna Sum Pattern Weighting
Target #N_T
Antenna Difference Pattern Weighting
+
+
Video Filter
Range Gate/ Presum
Center (c) Doppler Filter
+
+
Magnitude - Squared Detector
|·|² Data From 5 Frequencies, 2 Range Gates
PDI
Noise-Free Angle Discriminant Component

(a) Magnitude-Squared Detector Response for a Single Frequency, Time Slot, and Range Gate

(b) Formation of Noise-F-ee Discriminant Cmponent from the kth point target over a single time slot referenced at the input to the baseband filter is given by the expression,

$$(6.8) \qquad S_k(t) = \sigma_k^{\frac{1}{2}} A_k(\rho_{Sk} + \rho_{Dkj}) \sum_{n=o}^{15} \exp\left\{j\left[2\pi f_k t - \phi_{ki}\right]\right\} \quad P\left(\frac{t-nt_p \; -t_k}{t_t}\right)$$

$$\rho_{Sk}, \rho_{Dkj} = \text{kth target sum and difference pattern weightings,}$$

$$\emptyset_{ki} = 2\pi \dot{f}_{ci} t_k$$

and the other terms are defined after equation (4.1).

After filtering, sampling, range gating and presumming the signal in equation (6.8), we obtain

$$(6.9) \qquad S_k = \sigma_k^{\frac{1}{2}} A_k R_k(\rho_{Sk} + \rho_{Dkj}) \exp \{j \left[2\pi n \, f_k t_p - \emptyset_{ki}\right]\}$$

where n=0, 1, 2, ----, 15. The factor R_k in the above expression represents the range gate and presum weighting. It is noted that this factor ignores the mismatch in the presummer due to the target doppler shift. This assumption will have no impact in the short pulse modes and slight effect in the long pulse modes. Quantitatively, the expression for the range gate/presum weighting is

$$(6.10) \qquad R_k = F(t_k)$$

where

$$F(t_k)=N_p \begin{cases} 0 \quad , & \text{if} \quad \Delta \geq \frac{3}{2} \text{ or } \Delta \leq -\frac{3}{2} \\ \frac{3}{4}+\frac{\Delta}{2}, & \text{if} \quad -\frac{3}{2} \leq \Delta \leq -\frac{1}{2} \\ \frac{1}{2} \quad , & \text{if} \quad -\frac{1}{2} \leq \Delta \leq \frac{1}{2} \\ \frac{3}{4}-\frac{\Delta}{2}, & \text{if} \quad \frac{1}{2} \leq \Delta \leq \frac{3}{2} , \end{cases}$$

and $\Delta = \frac{t_k}{t_t}$.

There is a very important assumption made at this step. The total energy of the pulse within the range gates is assumed to be exactly split between the early and late gate contribution. However, the phase associated with the point target's true position in the range gate is maintained. This is illustrated in quantitative terms in Appendix C. The assumption was made to enhance the computation speed but it may have a significant impact in those cases where the range tracker does not follow the target with fidelity. If that is the case, the assumption may have to be abandoned.

The final step in this sequence is to compute the response of the doppler filter to the signal given in equation (6.9). (As an aside, it is noted that the target velocity is tracked using 5 adjacent doppler filters where the velocity always seeks to maintain the target in the center (C) filter of the 5. Only information from filter C is used in the formation of the angle discriminant). Since the signal in (6.9) represents a pure doppler tone by assumption (2) of section 6.4.1, we can easily write the response of the kth target for a single frequency and range gate. It is given by the expression

$$(6.11) \qquad S_k = \sigma_k^{1/2} A_k R_k (\rho_{SK} + \rho_{Dkj}) \frac{\sin (16Z_k)}{\sin(Z_k)} \exp \left[-j(15Z_k + \Phi_{ki})\right]$$

where $Z_k = \pi(m_c/32 - f_k t_p)$

m_c = number of center filter.

As noted earlier, the complete target return signal at the doppler filter output is obtained by assuming the processing channel from the antenna to the doppler filter output is linear and applying the linear superposition principle. For the ith transmit frequency and ℓth range gate, this gives

$$(6.12) \qquad S(i,\ell) = \sum_{k=1}^{N_T} S_k(i,\ell)$$

where N_T is the total number of point targets. Magnitude-squared detecting $S(i,\ell)$ and PDIing over the appropriate number of transmit frequencies and range gates, we obtain the noise-free angle discriminant component

$$(6.13) \qquad A = \sum_{\ell=1}^{2} \cdot \sum_{i=1}^{N_F} \left| S(i,\ell) \right|^2 = 2 \sum_{i=1}^{N_F} \left| S(i) \right|^2$$

where N_F is the number of transmit frequencies and the summation over the range gate is replaced by the factor 2 using the assumption stated earlier.

Equivalent Thermal Noise. If we assume that the target signal plus white gaussian noise is introduced at the front end of the receiver, the noise appearing at the PDI output can be shown (see appendix D) to be additive and approximately gaussian with mean and variance given by

(6.14) $$\text{mean} = 2N_A\sigma_o^2$$

(6.15) $$\text{variance} = 4\ N_A\ \sigma_o^4\ \left[2\ SNR_D + 1\right]$$

where N_A = PDI ratio for the angle discriminant,

SNR_D = Signal-to-Noise Ratio referenced to the doppler filter output,

σ_o^2 = variance of noise at doppler filter output.

Angle Discriminant Computation. The angle discriminant is then computed by the expression

(6.16) $$D_A = 10 \log \left(\frac{A_\sigma + \eta_\sigma}{A_\delta + \eta_\delta}\right)$$

where A_σ represents the sum plus difference noise-free discriminant component, A_δ represents the sum minus difference noise-free discriminant component, and η_δ and η_σ are samples from statistically independent random sequences where each member has the statistics described above. It is noted that A_σ and A_δ are computed using equation (6.13) with the appropriate antenna weighting factor.

6.4.4 Range Discriminant Computation Model

The range discriminant is formed by comparing the energy from the late range gate to the energy from the early range gate. Description of this computation model will follow the same format as the angle discriminant computation model description.

Noise-Free Discriminant Component Computation. A model for the computation of the noise-free range discriminant component is derived in Appendix C and is shown in Figure 6-11. The basic configuration of the model is identical to the corresponding angle discriminant model. However, there are some differences in the weighting factors between the two and for this reason we outline the processing of the range discriminant component below.

We start with a description of the computation of the single point target response at the doppler filter output. As in the angle discriminant case, the signal at the input to the baseband filter is given by equation (6.8) with ρ_{Dkj} set equal to zero even if the boresight is not pointing directly at the target. Proceeding to the filtering, sampling, range gating, and presumming process, we obtain the same expression as equation (6.9), but now separate range gate/presum weighting factors, R_k, must be computed for the early and late range gates. For the early gate the weighting factor is given by

$$(6.17) \qquad R_{Ek} = F_E(t_k) \qquad \text{(Early)}$$

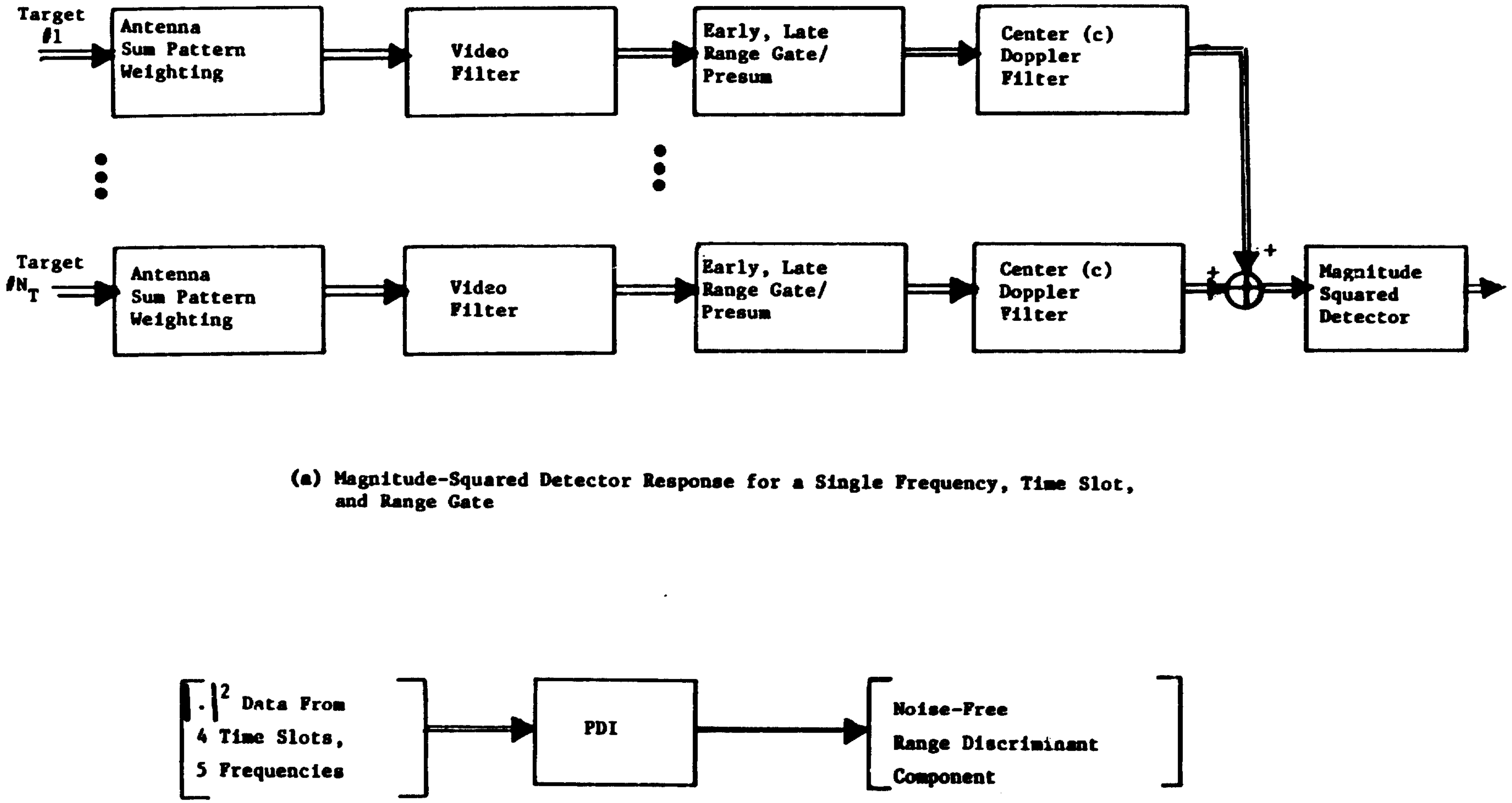
where

$$F_E(t_k) = N_P \begin{bmatrix} 0 \;, & \text{if} \quad \Delta \leq -3 \text{ or } \Delta \geq 1 \\ \frac{3+\Delta}{2} \;, & \text{if} \quad -3 \leq \Delta \leq -1 \\ \frac{1-\Delta}{2} \;, & \text{if} \quad -1 \leq \Delta \leq 1 \end{bmatrix}$$

and for the late gate this factor is given by

$$(6.18) \qquad R_{Lk} = F_L(t_k) \qquad \text{(Late)}$$

Figure 6-11 NOISE-FREE RANGE DISCRIMINANT COMPONENT COMPUTATION MODEL


Target #1
Antenna Sum Pattern Weighting
Video Filter
Early, Late Range Gate/ Presum
Center (c) Doppler Filter
Target $\#N_T$
Antenna Sum Pattern Weighting
Video Filter
Early, Late Range Gate/ Presum
Center (c) Doppler Filter
+
+
Magnitude Squared Detector
$|\cdot|^2$ Data From 4 Time Slots, 5 Frequencies
PDI
Noise-Free Range Discriminant Component

(a) Magnitude-Squared Detector Response for a Single Frequency, Time Slot, and Range Gate

(b) Formation of Noise-Free Discriminant Component where

$$F_L(t_k) = N_p \begin{bmatrix} \frac{1+\Delta}{2} \ , & \text{if} \quad -1 \leq \Delta \leq 1 \\ \frac{3-\Delta}{2} \ , & \text{if} \quad 1 \leq \Delta \leq 3 \\ 0 \ , & \text{if} \quad \Delta \geq 3 \text{ or } \Delta \leq -1 \ . \end{bmatrix}$$

The range discriminant components only use information from the center doppler filter and therefore they have the same doppler filter weighting as the angle discriminant components. Thus, the kth target response for the early range gate at the doppler filter output is given by

$$(6.19) \quad S_{Ek} = \sigma_k^{\frac{1}{2}} A_k R_{Ek} \rho_{Sk} \frac{\sin(16Z_k)}{\sin Z_k} \exp \left[-j(15Z_k + \emptyset_{ki})\right]$$

and the complete target response for the ith transmit frequency, the jth time slot, and the early range gate at the doppler filter output is given by the expression,

$$(6.20) \qquad S_E(i,j) = \sum_{k=1}^{N_T} S_{Ek}(i,j) \ .$$

Expressions for the late gate single target doppler filter response and the total target doppler filter response are identical to equations (6.19) and (6.20), respectively, with E replaced by L.

The noise-free range discriminant component is obtained by magnitude-squared detecting the doppler filter response (6.20) and PDIing over transmit frequencies and time slots. This can be expressed as

$$(6.21) \qquad R_E = \sum_{i=1}^{N_F} \sum_{j=1}^{4} \left| S_E(i,j) \right|^2 = 4 \sum_{i=1}^{N_F} \left| S_E(i) \right|^2$$

where the last equality is obtained by assuming that all time slot components for a given frequency are equal.

Equivalent Thermal Noise. The thermal noise added to the noise-free range discriminant component has properties which are identical to the angle discriminant noise, except that the PDI ratio becomes N_R, representing the range PDI ratio, instead of N_a.

Range Discriminant Computation. The range discriminant is computed using the expression

$$(6.22) \qquad D_R = 10 \log \left(\frac{R_L + \eta_L}{R_E + \eta_E} \right)$$

where R_E and R_L are computed from equation (6.21) and η_E and η_L are samples from statistically independent random sequences where each member has the statistics described above.

6.4.5 Velocity Discriminant Computation Model

Definition of the velocity and the on-target discriminants rely heavily upon the configuration of the doppler filters used to track the target velocity. The configuration is comprised of five adjacent filters as shown in Figure 6-12 where the tracker seeks to maintain target velocity in the center filter. In the sequel these filters will be labeled (from lowest frequency to highest frequency) Low Outrigger (LO), Low (L), Center (C), High (H), and High Outrigger (HO), respectively. The velocity discriminant is then formed by comparing all of the energy from the low (L) filter to all the energy from the high (H) filter. The form of this model is identical to the range and angle discriminant model and its description will follow the same format.

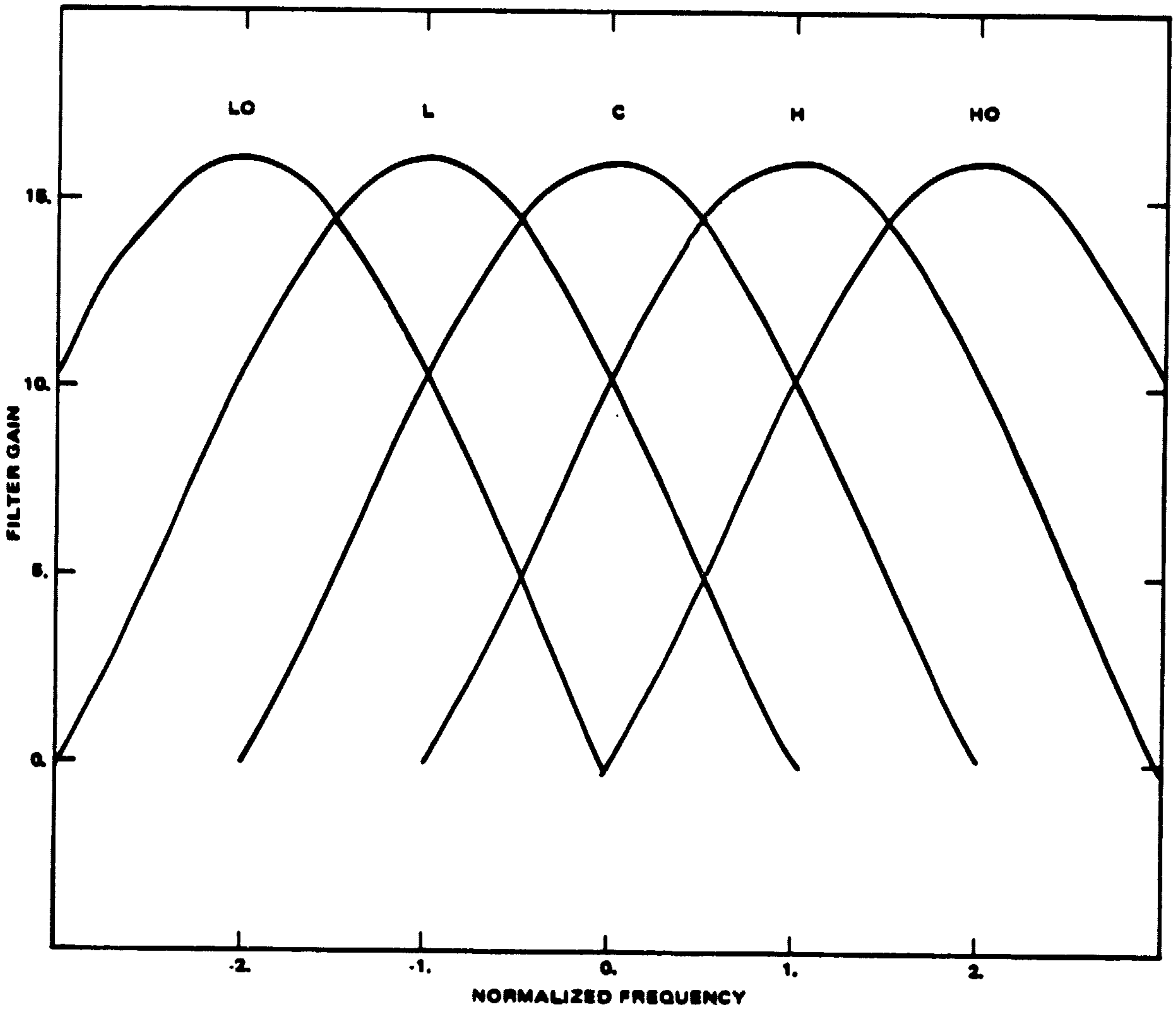


Figure 6-12. Track Mode Doppler Filter Configuration (Only Mainlobe Response Shown).

Noise-Free Discriminant Component Computation. Figure 6-13 gives a block diagram of the computation model and Appendix C gives a derivation of this model. Since the processing has the same form as the range and angle discriminant models, we will only provide the various weighting factors used in this case and point out any differences.

The antenna weighting factor is given by ρ_{Sk} where ρ_{Sk} is computed from equations (4.7) and (4.8). As in the range discriminant case, the difference pattern weighting is set to zero even though the antenna may not be pointing directly at the target. The range gate weighting R_k is computed as described in equation (6.10). The range gate weighting assumption used in the angle discriminant computation, applies in this case, as well. The doppler filter weighting factor is the same as the range and angle case with m_L (or m_H, depending on the component) replacing m_C.

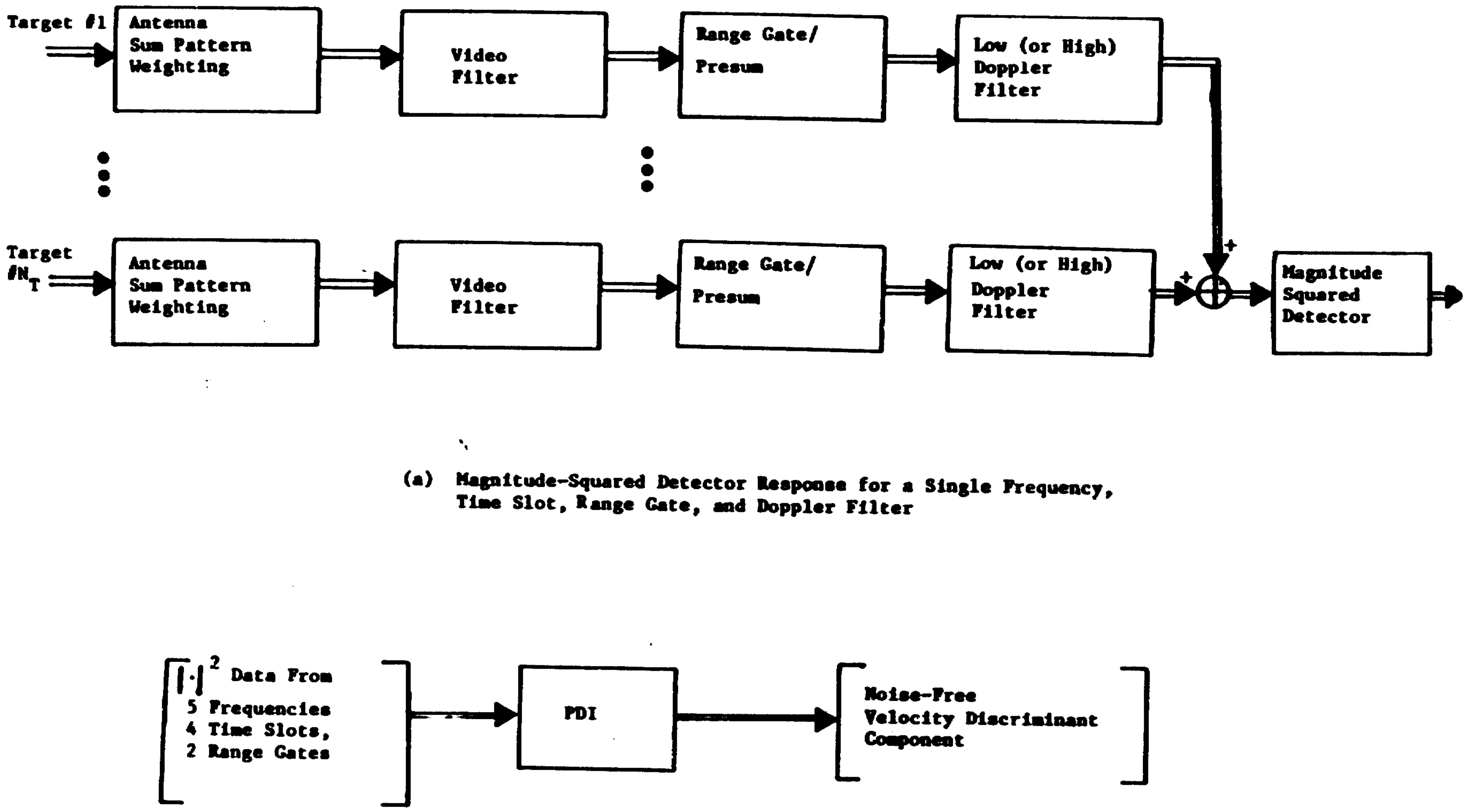
If we let the total response for the ith frequency, the jth time slot, and the ℓth range gate at the L-doppler filter output be given by $V_L(i,j,\ell)$, then the noise-free velocity discriminant component is obtained by magnitude-squared detecting and PDIing over frequencies, time slots, and range gates. This is expressed as .

$$(6.23) \qquad F_L = \sum_{i=1}^{N_F} \sum_{j=1}^{4} \sum_{\ell=1}^{2} \left| V_L(i,j,\ell) \right|^2 = 8 \sum_{i=1}^{N_F} \left| V_L(i) \right|^2$$

where the last equality is obtained by assuming that all time slot and range gate components are equal for a given transmit frequency.

Equivalent Thermal Noise. The velocity discriminant noise has the same form as the angle and range discriminant case with N_V, the velocity PDI ratio, replacing N_A in equations (6.14) and (6.15).

Figure 6-13 NOISE-FREE VELOCITY DISCRIMINANT COMPONENT COMPUTATION MODEL


Target #1
Antenna Sum Pattern Weighting
Video Filter
Range Gate/ Presum
Low (or High) Doppler Filter
Target $\#N_T$
Antenna Sum Pattern Weighting
Video Filter
Range Gate/ Presum
Low (or High) Doppler Filter
+
+
Magnitude Squared Detector
$|\cdot|^2$ Data From 5 Frequencies 4 Time Slots, 2 Range Gates
PDI
Noise-Free Velocity Discriminant Component

(a) Magnitude-Squared Detector Response for a Single Frequency, Time Slot, Range Gate, and Doppler Filter

(b) Formation of Noise-Free Discriminant Components

Velocity Discriminant Computation. The velocity discriminant is computed from the expression

$$(6.24) \qquad D_V = 10 \log \left(\frac{F_L + \eta_L}{F_H + \eta_H} \right)$$

where F_L and F_H are computed using equation (6.23) and η_L and η_H are samples from statistically independent random sequences where each member has the statistics described above.

6.4.6 On-Target Discriminant Computation Model

The On-Target discriminant is formed by comparing the total energy from the center (C) doppler filter to the combined total energy from the LO and HO doppler filters over a data cycle. Computation of the noise-free discriminant components is identical to the velocity discriminant case with m_C, m_{LO}, or m_{HO} replacing m_L and m_H in the doppler filter weighting factor. The On-Target discriminant noise characteristics are identical to those given for the velocity discriminant case above. Therefore, the expression for the On-Target discriminant is

$$(6.25) \qquad D_{OT} = 10 \log \left(\frac{F_C + \eta_C}{F_{LO} + F_{HO} + \eta_{LO}} \right)$$

where F_C, F_{LO}, and F_{HO} are computed from equations (6.23) and η_C and η_{LO} are samples from statistically independent random sequences where each member has the statistics described above.

6.4.7 Radar Signal Strength Computation Model

In the Ku-Band radar, the radar signal strength meter is designed to work in the tracking mode only. During the track mode the radar signal strength meter is determined from the following algorithm

$$(6.26) \qquad \mathrm{RSS} = \begin{bmatrix} \mathrm{AGC} - \mathrm{AGC}\big|_{\mathrm{initial}}, & \text{if } \mathrm{AGC} \geq \mathrm{AGC}\big|_{\mathrm{initial}} \\ 0 & , \text{ if } \mathrm{AGC} < \mathrm{AGC}\big|_{\mathrm{initial}} \end{bmatrix}$$

where $\mathrm{AGC}\big|_{\mathrm{initial}}$ is the setting of the system AGC when the track mode is first entered. The AGC is designed to maintain the average signal plus noise voltage at 1.4 quantization levels at the video filter output.

Since the system AGC is not modeled in the present version of the track mode simulation, the algorithm of (6.26) is replaced by the following computation:

$$(6.27) \qquad \mathrm{RSS} = \mathrm{SNR}_V$$

where SNR_V is the signal-to-noise ratio at the video filter output and is computed from equation (5.16) for passive modes and equation (5.22) for active modes. This approximation will be highly accurate for $\mathrm{SNR}_V >> 1$, but will break down for $\mathrm{SNR}_V \leq 1$. If time permits, the Ku-Band Radar AGC algorithm and signal strength algorithm will be simulated more accurately.

6.4.8 Computer Model Details

The computer model shown in Figure 6-14 consists of five subroutines. A separate subroutine is dedicated to each of the following functions:

(1) updating of all transformation matrices,

(2) updating of the LOS position, velocity and RCS value for each scatterer and updating of the LOS position and velocity for the target C.G.

(3) computation of all noise-free discriminant components,

(4) computation of all discriminants (including thermal noise),

(5) computation of target signal strength.

Each of these subroutines is described in detail below.

Figure 6-14 SIGNAL GENERATION AND PROCESSING MODEL COMPUTER ALGORITHM (1 of 3)

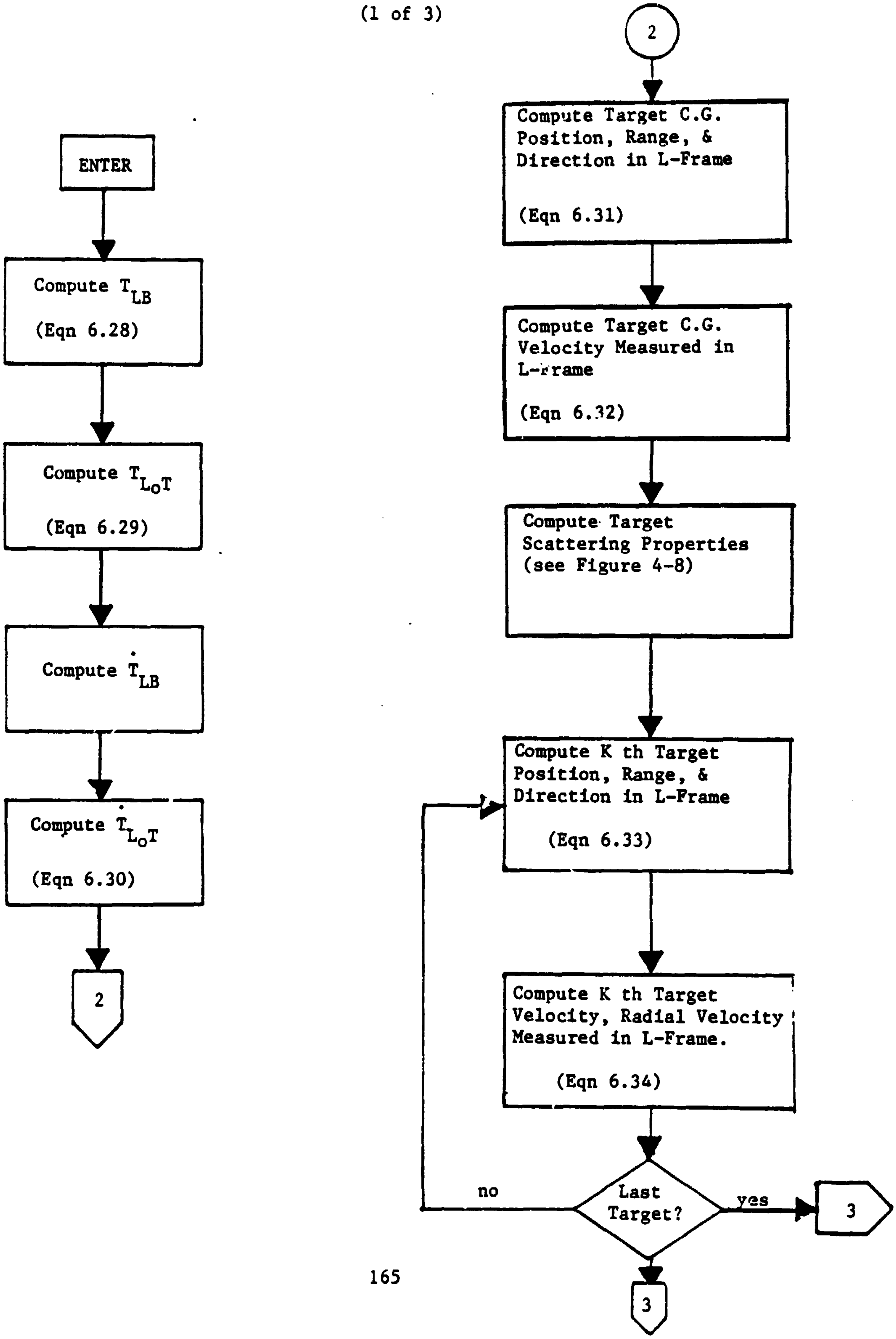

ENTER
Compute T_{LB}
(Eqn 6.28)
Compute T_{L_oT}
(Eqn 6.29)
Compute $\dot{T}_{LB}$
Compute $\dot{T}_{L_oT}$
(Eqn 6.30)
2
2
Compute Target C.G. Position, Range, & Direction in L-Frame
(Eqn 6.31)
Compute Target C.G. Velocity Measured in L-Frame
(Eqn 6.32)
Compute Target Scattering Properties (see Figure 4-8)
Compute K th Target Position, Range, & Direction in L-Frame
(Eqn 6.33)
Compute K th Target Velocity, Radial Velocity Measured in L-Frame.
(Eqn 6.34)
Last Target?
no
yes
3
3

Figure 6-14 SIGNAL GENERATION AND PROCESSING MODEL COMPUTER ALGORITHM (2 of 3)

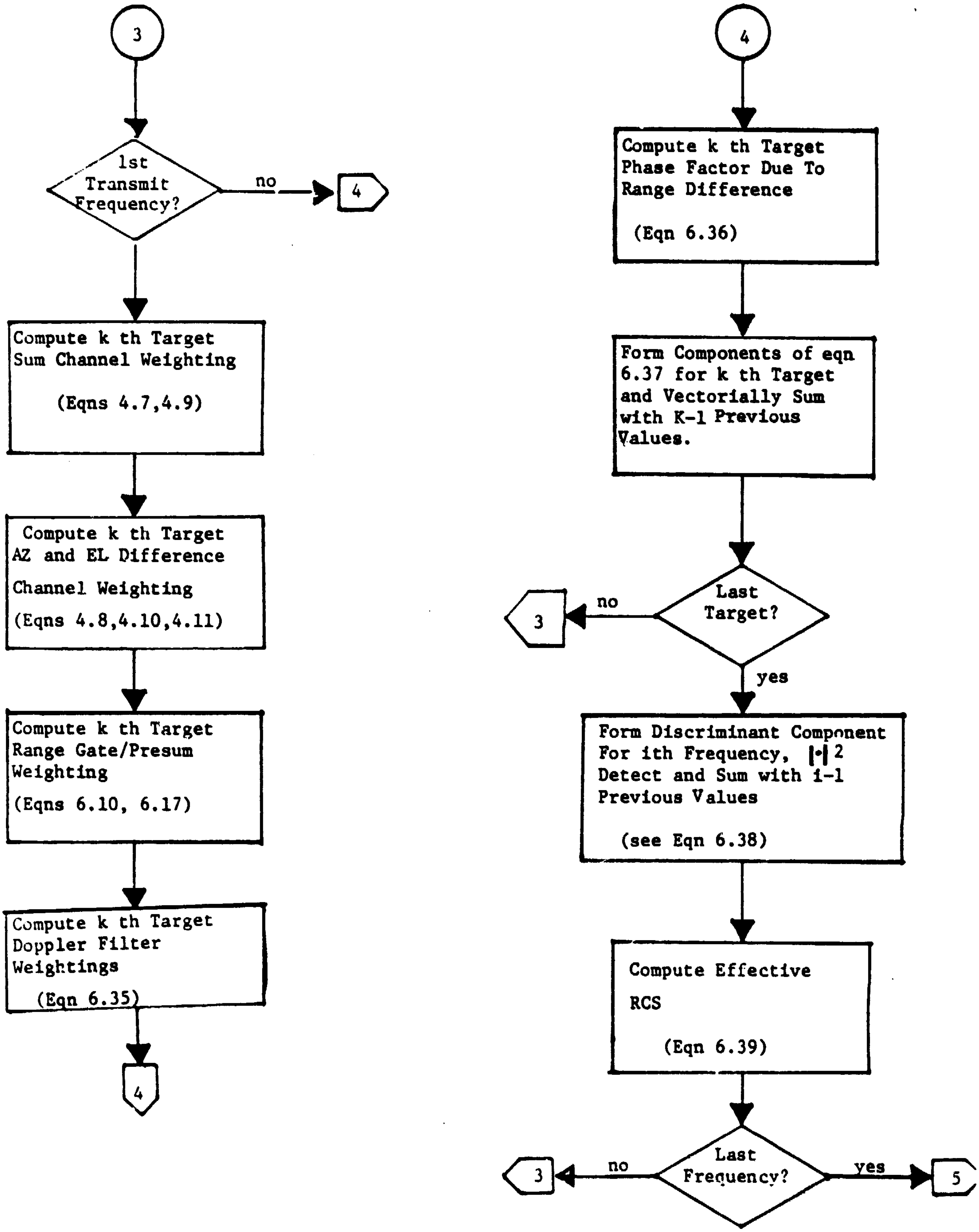
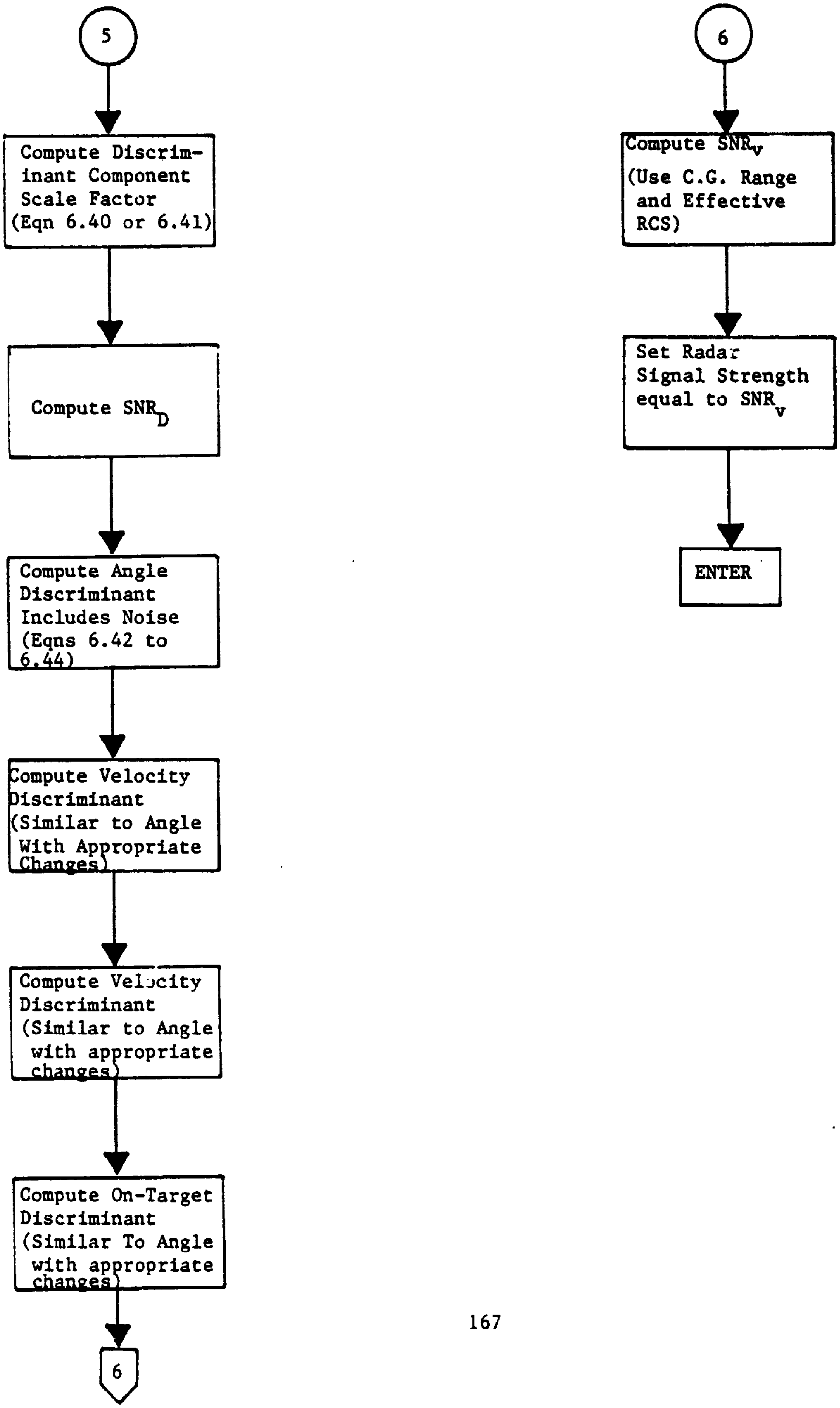

3
1st Transmit Frequency?
no
4
Compute k th Target Sum Channel Weighting (Eqns 4.7,4.9)
Compute k th Target AZ and EL Difference Channel Weighting (Eqns 4.8,4.10,4.11)
Compute k th Target Range Gate/Presum Weighting (Eqns 6.10, 6.17)
Compute k th Target Doppler Filter Weightings (Eqn 6.35)
4
4
Compute k th Target Phase Factor Due To Range Difference (Eqn 6.36)
Form Components of eqn 6.37 for k th Target and Vectorially Sum with K-1 Previous Values.
Last Target?
no
3
yes
Form Discriminant Component For ith Frequency, |•|2 Detect and Sum with i-1 Previous Values (see Eqn 6.38)
Compute Effective RCS (Eqn 6.39)
Last Frequency?
no
3
yes
5

Figure 6-14 SIGNAL GENERATION AND PROCESSING MODEL COMPUTER ALGORITHM (3 of 3)


5
Compute Discriminant Component Scale Factor (Eqn 6.40 or 6.41)
Compute SNR_D
Compute Angle Discriminant Includes Noise (Eqns 6.42 to 6.44)
Compute Velocity Discriminant (Similar to Angle With Appropriate Changes)
Compute Velocity Discriminant (Similar to Angle with appropriate changes)
Compute On-Target Discriminant (Similar To Angle with appropriate changes)
6
6
Compute SNR_v (Use C.G. Range and Effective RCS)
Set Radar Signal Strength equal to SNR_v
ENTER

Update of Transformation Matrices (TRNSFM). This subroutine updates T_{LB}, T_{L_oT}, $\dot{T}_{LB}$, and $\dot{T}_{L_oT}$. The transformation matrix T_{LB} is computed with the expression

$$T_{LB} = \begin{pmatrix} C\beta & 0 & -S\beta \\ 0 & 1 & 0 \\ S\beta & 0 & C\beta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & C\alpha & S\alpha \\ 0 & -S\alpha & C\alpha \end{pmatrix} \begin{pmatrix} C\gamma & S\gamma & 0 \\ -S\gamma & C\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6.28}$$

where α, β = latest measurement of antenna gimbal position,

γ = yaw angle of R-frame with respect to B-frame (nominally 67°)

C = cos

S = sin .

The t ansformation T_{LT} is obtained from

$$T_{L_oT} = T_{L_oB_o} T_{B_oT} \tag{6.29}$$

where $T'_{L_oB_o} = T_{LB}$ is computed in equation (6.28) and T_{B_oT} is pro ided by the parent simulation.

The matrix $\dot{T}_{LB}$ is computed by time differentiating T_{LB} as given in equation (6.28) and noting that α, β vary with time but γ is fixed. Finally, the matrix $\dot{T}_{L_oT}$ is computed from the expression

$$\dot{T}_{L_oT} = T_{L_oB_o} \dot{T}_{B_oT} + \dot{T}_{L_oB_o} T_{B_oT} \tag{6.30}$$

where $T_{L_oB_o}$ and $\dot{T}_{L_oB_o}$ are defined above and T_{B_oT} and $\dot{T}_{B_oT}$ are provided by the parent simulation.

Target Position and Velocity Update (PVTRAN). This subroutine computes (1) the position, range and direction vector in LOS coordinates for each scatterer and the target C.G., (2) the velocity and radial velocity component as measured in the LOS frame for each scatterer and the target c.g., and (3) the RCS value for each scatterer. The subroutine is organized as follows. Since each scatterer's LOS position and velocity computation requires the C.G. position and velocity in the LOS frame, the c.g. parameters are computed first then the parameters for each point target are computed.

The target C.G. position, range, and direction vector in LOS coordinates are computed from

$$(6.31)\qquad \begin{aligned} \vec{r}_o^{\,L} &= T_{LB}(\vec{r}_o^{\,B} - \vec{X}^B) && \text{(position)} \\ r_o^{L} &= |\vec{r}_o^{\,L}| && \text{(range)} \\ \hat{r}_o^{L} &= \vec{r}_o^{\,L} / |\vec{r}_o^{\,L}| && \text{(direction)} \end{aligned}$$

where $\vec{r}_o^{\,B}$ is provided by the parent simulation, $\vec{X}^B$ is the fixed radar offset from the orbiter C.G., and T_{LB} is computed above. The C.G. velocity and radial velocity component as measured in the LOS frame and expressed in LOS coordinates are given by

$$(6.32)\qquad \begin{aligned} \dot{\vec{r}}_o^{\,L} &= T_{LB}\dot{\vec{r}}_o^{\,B} + \dot{T}_{LB}\vec{r}_o^{\,B} && \text{(velocity)} \\ \dot{r}_o^{L} &= \dot{\vec{r}}_o^{\,L} \cdot \hat{r}_o^{L} && \text{(radial component)} \end{aligned}$$

where $\dot{\vec{r}}_o^{\,E}$ and $\vec{r}_o^{\,B}$ are provided by the parent simulation and T_{LB} and $\dot{T}_{LB}$ are computed above.

The next step is to update the scatterer positions in the target frame and the scatterer RCS values using the subprogram described in Section 4.5 (see Figure 4-9). Then, each of the scatterer's position and velocity parameters are computed. LOS position, range, and direction for the kth scatterer are given by

$$\vec{r}_k{}^L = \vec{r}_o{}^L + T_{L_oT}\vec{r}_k{}^T \qquad \text{(velocity)}$$

$$(6.33) \qquad r_k{}^L = |\vec{r}_k{}^L| \qquad \text{(range)}$$

$$\hat{r}_k{}^L = \vec{r}_k{}^L \,/\, |\vec{r}_k{}^L| \qquad \text{(direction)}$$

where $\vec{r}_k{}^T$ is fixed and $\vec{r}_o{}^L$ and T_{L_oT} are computed above. LOS frame velocity of the kth scatterer is computed from the expression

$$\dot{\vec{r}}_k{}^L = \dot{\vec{r}}_o{}^L + \dot{T}_{L_oT}\vec{r}_k{}^T \qquad \text{(velocity)}$$

$$(6.34) \qquad \dot{r}_k{}^L = \dot{\vec{r}}_k{}^L \cdot \hat{r}_k{}^L \qquad \text{(radial component)}$$

where $\dot{\vec{r}}_o{}^L$ and $\dot{T}_{L_oT}$ are computed above.

Signal Generation and Processing (SIGNAL). The main purpose of this subroutine is to compute all of the noise-free discriminant components. It performs the calculations which are described in sections 6.4.3 through 6.4.6 and derived in Appendix C. This task is performed as follows.

First, for the kth target and ith transmit frequency, all of the weighting factors required to form the various responses at the doppler filter output are computed. These include the sum pattern weighting, difference pattern weighting, range gate/presum weighting, doppler filter weighting and initial phase computation. Antenna sum and difference pattern weightings are computed using equations (4.2) through (4.7) and the antenna models described in section 4. The range gate/presum weighting factor is given by equation (6.10) for non-range discriminant components and by equation (6.17) for range discriminant components. The doppler filter weighting is computed from the expression

(6.35) $$F_k(m) = \frac{\sin(16Z_k)}{\sin(Z_k)} \exp\left[-j15Z_k\right]$$

where $Z_k = \pi(\frac{m}{32} - f_k t_p)$,

$m = m_{LO}, m_L, m_C, m_H, m_{HO}$.

The doppler weights need only be computed for the first transmit frequency, since the PRF is adjusted at each new frequency to maintain a constant filter position for a nonaccelerating target. Finally, the phase of the kth target return referenced to the leading edge of the C.G. return is computed from

(6.36) $$\phi_{ki} = \frac{4\pi}{\lambda_{ci}}(r_k^L - r_o^L)$$

where λ_{ci} is the wavelength of the ith transmit frequency. It is remarked that one can just as well compute the phase using the expression

$$\phi_{ki} = \frac{4\pi}{\lambda_{ci}}(r_k^L - r_G)$$

where r_G is the range to the center of the range gates.

The next step is to form the following responses for the kth target and ith frequency at the doppler filter output:

(6.37)
$$\begin{aligned}
A_{1k} &= \text{Sum Component} = \sqrt{\sigma_k}\rho_{Sk}R_kF_k(m_c)\exp(j\phi_{ki}) \\
A_{2k} &= \text{Azimuth Difference Component} = \sqrt{\sigma_k}\rho_{AZk}R_kF_k(m_c)\exp(j\phi_{ki}) \\
A_{3k} &= \text{Elevation Difference Component} = \sqrt{\sigma_k}\rho_{ELk}R_kF_k(m_c)\exp(j\phi_{ki}) \\
R_{1k} &= \text{Early Component} = \sqrt{\sigma_k}\rho_{Sk}R_{Ek}F_k(m_c)\exp(j\phi_{ki}) \\
R_{2k} &= \text{Late Component} = \sqrt{\sigma_k}\rho_{Sk}R_{Lk}F_k(m_c)\exp(j\phi_{ki})
\end{aligned}$$

$$V_{1,2,3,4k} = \text{L,H,LO,HO Components} = \sqrt{\sigma_k}\rho_{Sk}R_kF_k(\cdot)\exp(j\phi_{ki})$$

where $\cdot = m_L, m_H, m_{LO}, m_{HO}$, respectively. Once formed these components are vectorially summed over the number of targets to form the complete target response for the component at the doppler filter output for the ith frequency.

Then, these components are combined appropriately to form the noise-free discriminant components at the doppler filter output for the i th frequency. After this step, the newly formed components are magnitude-squared detected and summed over N_F transmit frequencies. Result of all of this processing gives the following noise-free discriminant components at the PDI output:

$$AZ_\sigma = \sum_{i=1}^{N_F} \left| \sum_{k=1}^{N_T} (A_{1k} + A_{2k}) \right|^2$$

$$AZ_\delta = \sum_{i=1}^{N_F} \left| \sum_{k=1}^{N_T} (A_{1k} - A_{2k}) \right|^2$$

(for EL_σ, EL_δ replace the subscript 2 by 3)

(6.38) $$R_E = \sum_{i=1}^{N_F} \left| \sum_{k=1}^{N_T} R_{1k} \right|^2$$

$$R_L = \sum_{i=1}^{N_F} \left| \sum_{k=1}^{N_T} R_{2k} \right|^2$$

$$V_L = \sum_{i=1}^{N_F} \left| \sum_{k=1}^{N_T} V_{1k} \right|^2$$

$$V_H = \sum_{i=1}^{N_F} \left| \sum_{k=1}^{N_T} V_{2k} \right|^2$$

(for V_{LO}, V_{HO} replace subscripts 1,2 by 3,4)

where each of the components above is within a constant scale factor of the actual noise-free discriminant component seen at the PDI output.

One final step is performed in this subroutine and that is to compute a quantity called the average effective cross-section. This quantity is a measure of the average target cross-section weighted by the normalized antenna sum pattern value. This value is averaged over five frequencies and is given by the expression

$$(6.39) \qquad \text{Effective Cross Section} \triangleq \left[\frac{1}{NF} \sum_{i=1}^{N_F} \left| \sum_{k=1}^{N_T} \sigma_k^{\frac{1}{2}} \rho_{Sk} e^{j\phi_{ki}} \right|^2 \right].$$

Discriminant Component Computation (DISCRM). This subroutine adds equivalent thermal noise to each of the discriminant components and computes the discriminants with the resulting component values. We first compute the scale factor alluded to earlier. This factor contains many of the range equation terms. For the passive mode it is given by

$$(6.40) \qquad S_1 = \frac{4\, G^2 \lambda_c{}^2 P_T N_P}{(4\pi)^3 (R_o{}^L)^4 L_T k T_S B_n N_F} \qquad \text{(Passive Mode)}$$

where N_P is the number of samples per pulse and the other terms are defined in Section 5. For the active mode this factor becomes

$$(6.41) \qquad S_1 = \frac{4\, G \lambda^2 P_{BT} N_P}{(4\pi)^2 (R_o{}^L)^2 L_T K T_S B_n N_F} \qquad \text{(Active Mode)}$$

where $$P_{BT} = \frac{P_B G_B}{L_B},$$

P_B = peak transmit power of beacon,

G_B = one-way beacon antenna gain,

L_B = beacon transmit losses.

The next step is to tackle the angle discriminant computation. This includes computing the statistics of the noise for this discriminant. Mean and variance of the noise are computed as follows:

$$\text{Mean}_{AZ} = N_A \text{ (same for } \sigma \text{ and } \delta \text{ components)}$$

$$\text{var}_{AZ\sigma} = \left[2\ N_A\ S_1 AZ_\sigma + N_A\right] \qquad (6.42)$$

$$\text{var}_{AZ\delta} = \left[2\ N_A\ S_1 AZ_\delta + N_A\right]$$

where S_1 is computed in equation (6.40 or 6.41), AZ and AZ are computed in equations (6.38) and N_A is the angle PDI ratio. It is important to note that the variance of the I,Q noise components at the doppler filter output are assumed to be equal to unity for convenience in the computation. In the next step, the equivalent noise is added to the angle discriminant components and we obtain

$$D_{AZ\sigma} = \left| N_A\ S_1 AZ_\sigma + \text{Mean}_{AZ} + \sqrt{\text{Var}}_{AZ\sigma}\ N(0,1) \right|$$

(6.43)

$$D_{AZ\delta} = \left| N_A\ S_1 AZ_\delta + \text{Mean}_{AZ} + \sqrt{\text{Var}}_{AZ\delta}\ N(0,1) \right|$$

where N(0,1) is defined as a random selection from a gaussian population with zero mean and unit variance. The last step is to compute the angle discriminant

$$D_{AZ} = 10 \log \left(D_{AZ\sigma}/D_{AZ\delta}\right). \qquad (6.44)$$

where the logorithm computation is assumed to be base 10.

The range, velocity and on-target discriminants are computed in an identical manner, making the appropriate changes in scale factors and components.

Radar Signal Strength Computation (RSS). As discussed in Section 6.4.7, the radar signal strength is computed very simply in the present version of the tracking simulation. The radar signal strength is set equal to the SNR at the video filter output where it is assumed that the transmitter is at full power. This computation is done using equation (5.16) for the passive mode where in this case P_T is always the maximum peak transmitter power and σ (the RCS) is computed using equation (6.39). For the active mode equation (5.22) is used to obtain the SNR_V.

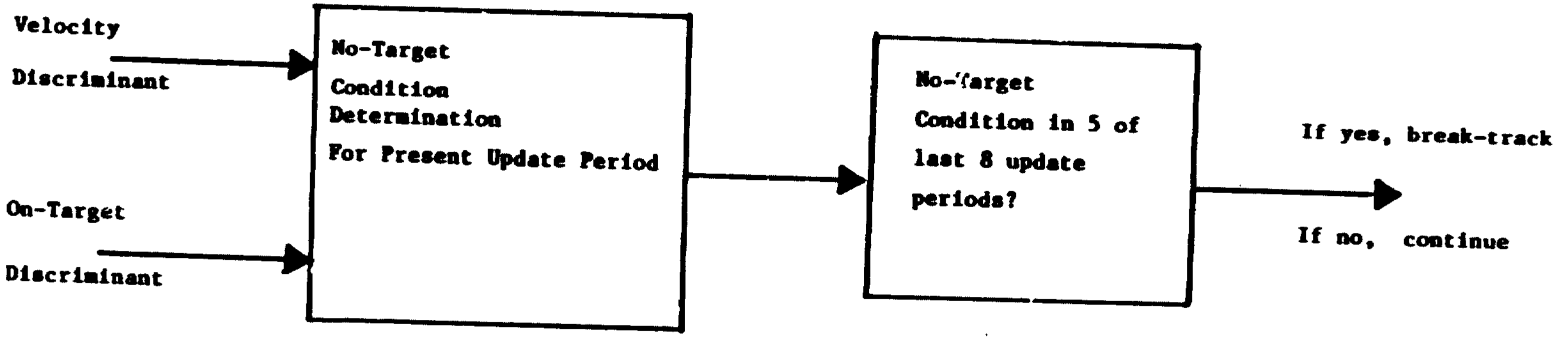
6.5 BREAK-TRACK ALGORITHM DESCRIPTION

The break-track algorithm used in the track mode simulation is functionally identical to the logic used in the Ku-Band radar signal processor. Figure 6-15 gives a simplified block diagram of the break-track algorithm. Key components are the two discriminants and the no-target condition determination. In this subsection, we will describe the no-target condition and its determination, describe the break-track condition, and describe the implementation of this algorithm on the computer.

6.5.1 Noise-Free Discriminant Response Functions

Both discriminants take advantage of the shape of the doppler filter's mainlobe and its relative position with respect to the other filter mainlobes in order to determine target location in the filter bank and absence or presence of the target. This is most clearly seen from plots of the noise-free velocity and on-target discriminants as a function of target doppler velocity given in Figures 6-16 and 6-17, respectively.

Figure 6-15 SIMPLIFIED BLOCK DIAGRAM OF BREAK-TRACK ALGORITHM



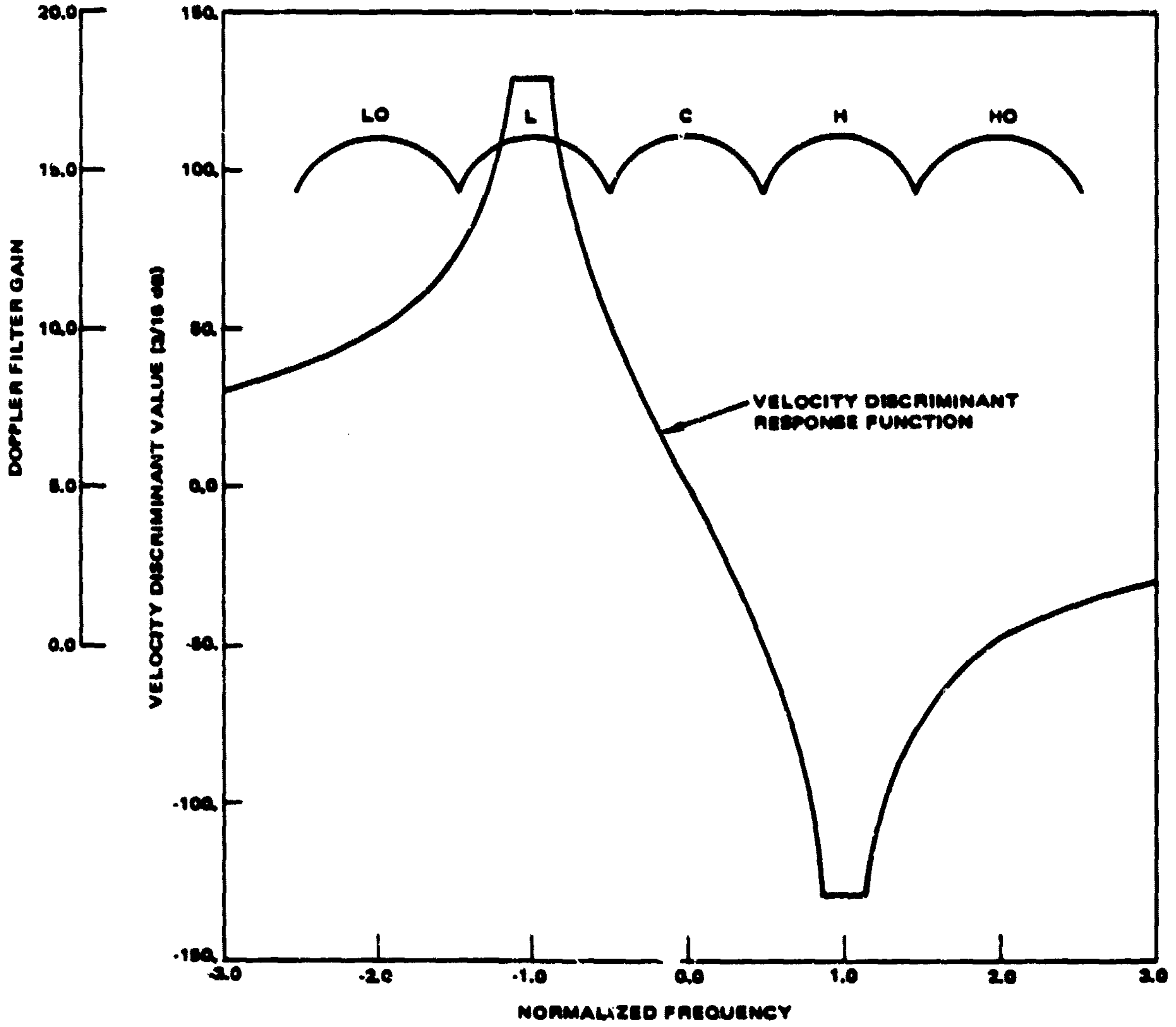


Figure 6-16. Noise-Free Velocity Discriminant Frequency Response.

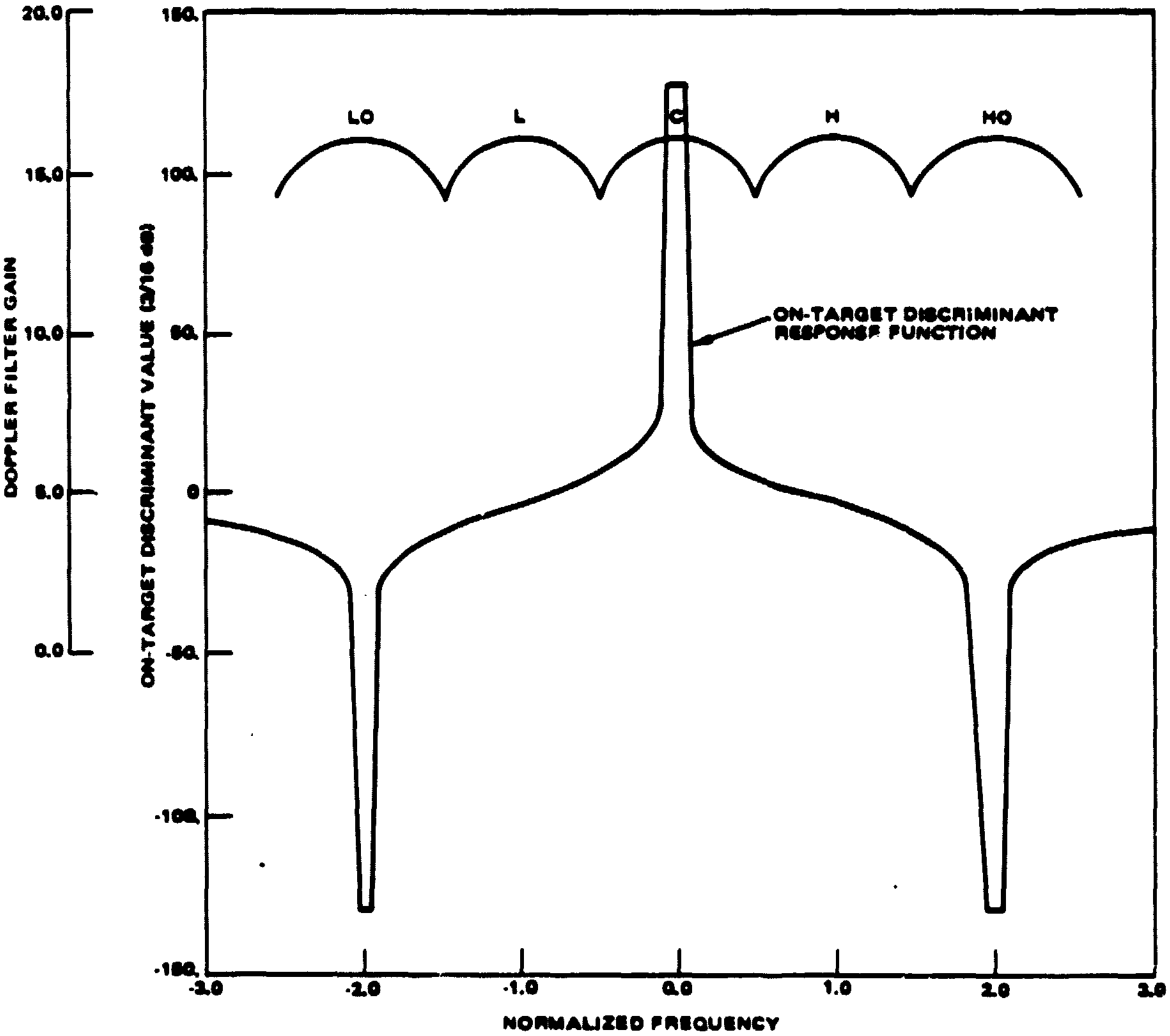


Figure 6-17. Noise-Free On-Target Discriminant Frequency Response.

6.5.2 Determination of a No-Target Condition

The basic idea behind the no-target determination is as follows. If a noise-only condition exists then the velocity discriminant value will be in the neighborhood of 0 dB and the on-target discriminant value will be in the vicinity of -3 dB. However, if a target with sufficient strength exists within the five filters, at least one of the discriminant values will not be near its noise only bias value as shown in Figures 6-16 and 6-17. Thus, the method for determination of a no-target condition is to establish thresholds about the bias values in each case (shown as dashed lines in Figures 6-16 and 6-17) and compare the discriminant values to their respective thresholds. A no-target condition is declared if both discriminants lie between their thresholds, i.e. in the region of their no-target bias values.

Qunatitatively, the no-target condition can be described as follows. First, the following quantities (called discretes) are defined:

$$(6.45)\qquad \begin{aligned} FTH &= \left[\begin{array}{ll} 1, & \text{if } |D_V| \le T_V \\ 0, & \text{if } |D_V| > T_V \end{array}\right. , \\ OT &= \left[\begin{array}{ll} 1, & \text{if } D_{OT} \le T_H \\ 0, & \text{if } D_{OT} > T_H \end{array}\right. , \\ AOT &= \left[\begin{array}{ll} 1, & \text{if } D_{OT} > T_L \\ 0, & \text{if } D_{OT} \le T_L \end{array}\right. . \end{aligned}$$

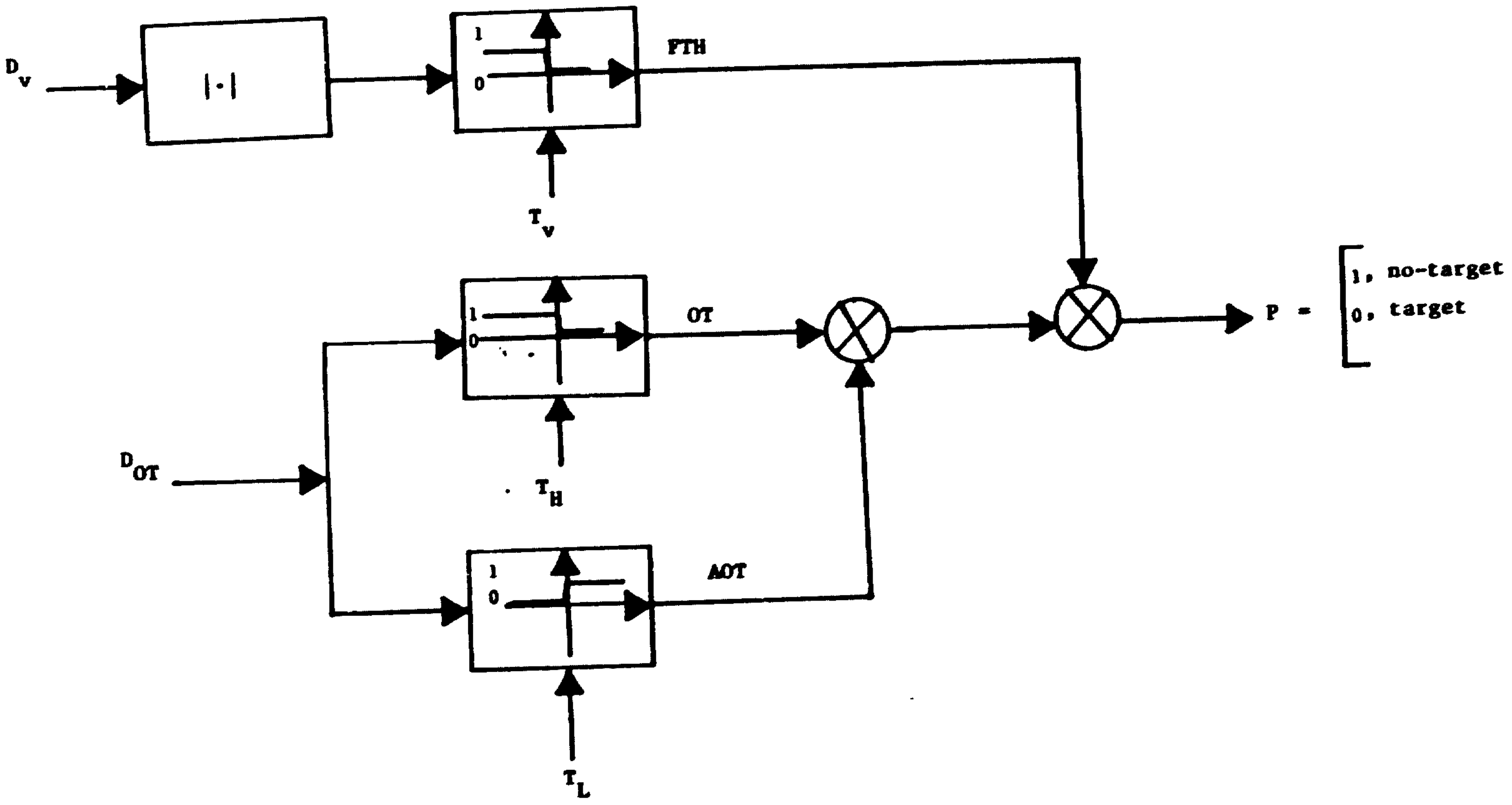
Then, target/no-target decision is based upon the product of the discretes as follows:

$$(6.46)\qquad (FTH)\ (OT)\ (AOT) = \left[\begin{array}{ll} 1 & \text{, no target} \\ 0 & \text{, target} \end{array}\right. .$$

This decision logic is illustrated in Figure 6-18.

C - 3

Figure 6-18 NO-TARGET DETERMINATION ALGORITHM



6.5.3 Break-Track Determination

A break-track condition is declared if a no-target condition is obtained in the present update period and four of the last seven update periods.

⸱ 5.4 Computer Algorithm Details

Figure 6-19 illustrates the computer algorithm used for the break-track determination. It implements the equations described above and requires no further description.

6.6 ANGLE AND ANGLE RATE TRACKING LOOP MODEL DESCRIPTION

In the GPC-ACQ and Auto modes, the radar provides estimates of the target inertial roll and pitch rates and tracks the target roll and pitch angles in the Orbiter Body coordinate system. A simplified block diagram of its mechanization in the Ku-Band radar is illustrated in Figure 6-20. In this subsection, we shall describe (1) the mathematical model used to represent this tracking system, (2) the major assumptions and approximations underlying this model, (3) the system and target error effects incorporated into the model, and (4) the computer implementation of the model.

6.6.1 The Model

As noted in Figure 6-20, the tracking system is composed of an α and β gimbal tracking loop. It was suggested in reference [20] that these loops be approximated by the second order continuous-time models shown in Figures 6-21 and 6-22. The loop constants w_n and τ were designed (see reference [21]) so that the angle rate estimator is critically damped and so that the loop transient response is damped out as quickly as possible while still meeting the loop noise specification. The design values of w_n and τ are given in Table 6-5 for reference.

Figure 5-19 BREAK-TRACK COMPUTER ALGORITHM

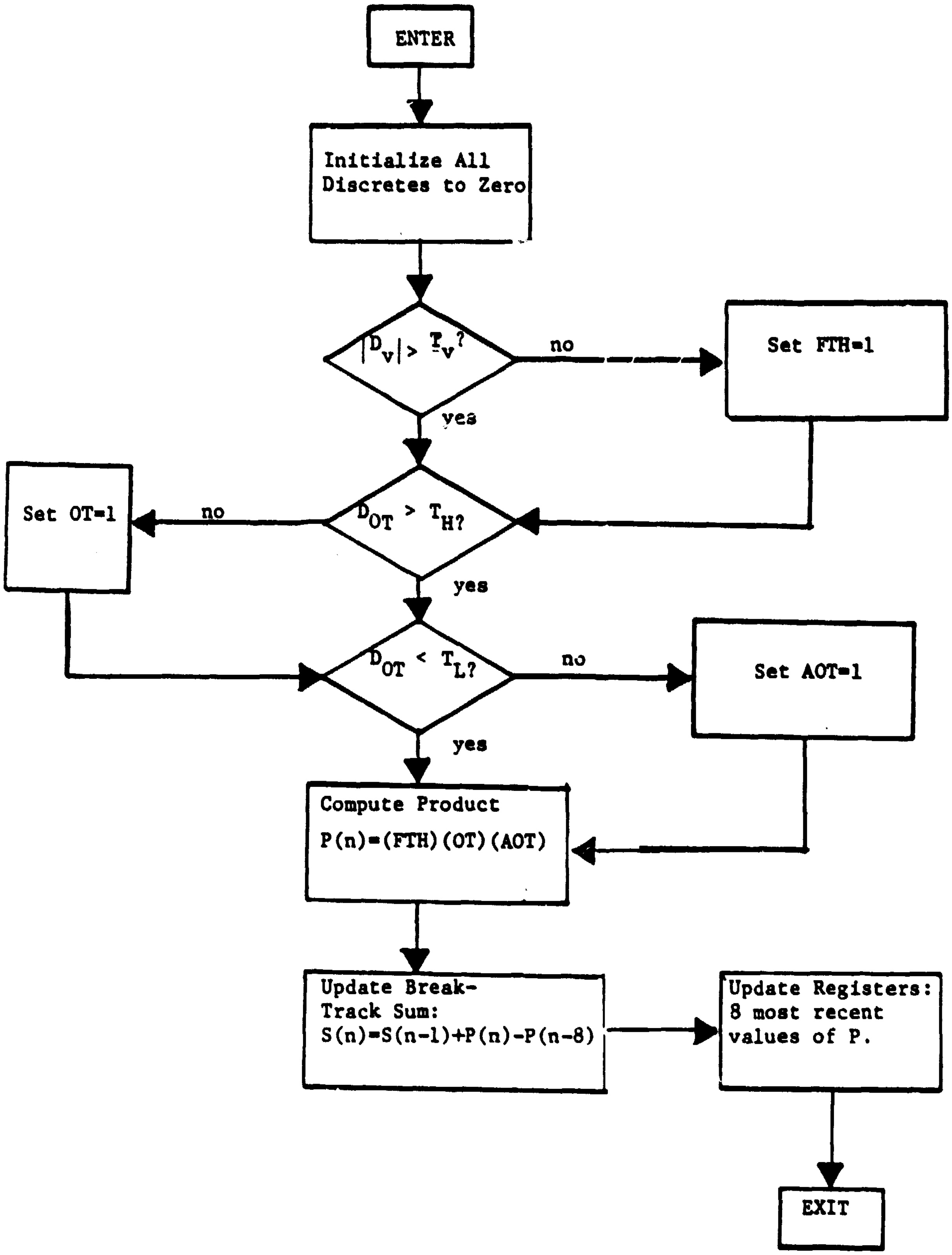


Figure 6-20 SIMPLIFIED DIAGRAM OF KU-BAND ANGLE RATE AND ANGLE TRACKER

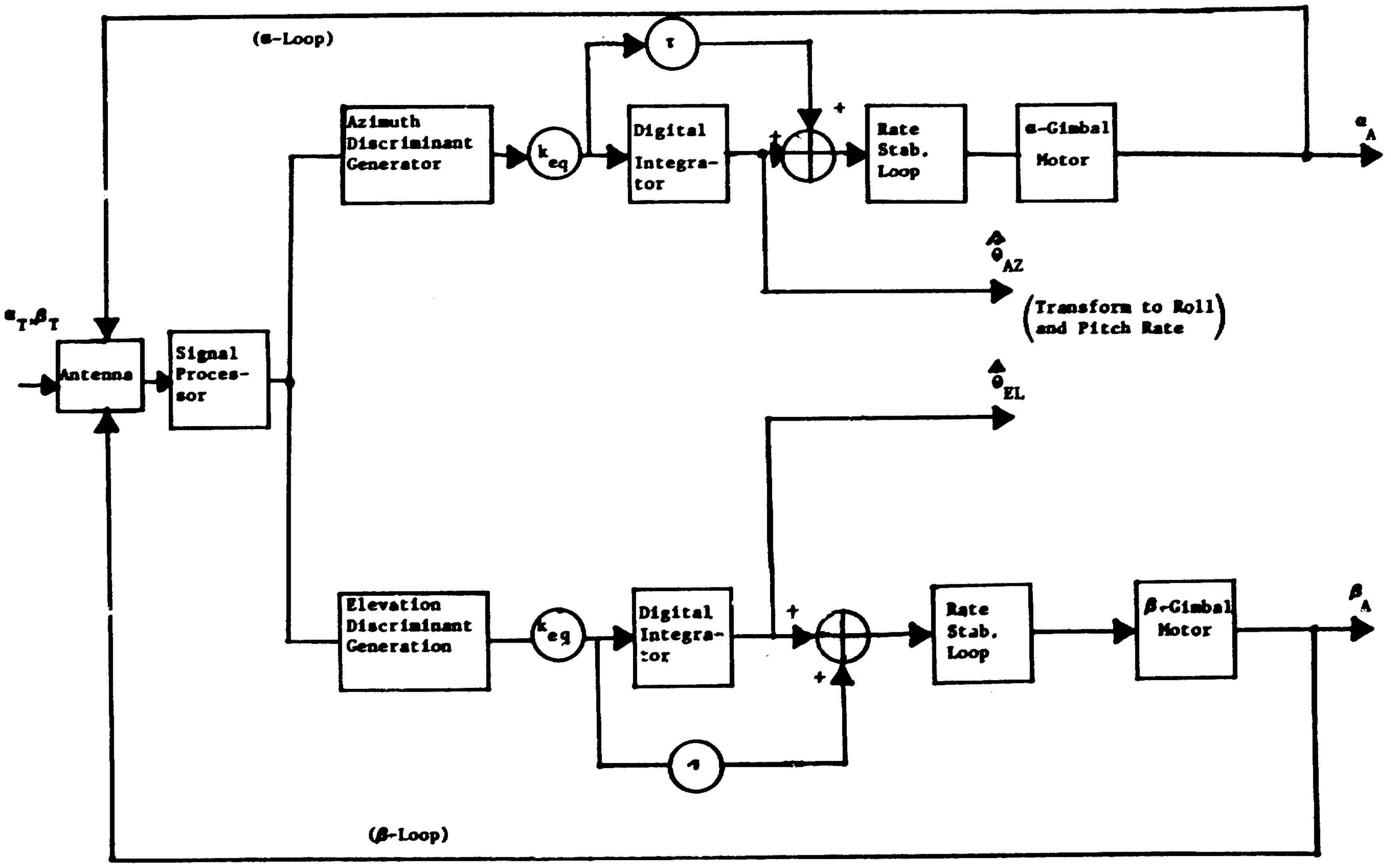


Figure 6-21 α-LOOP MODEL

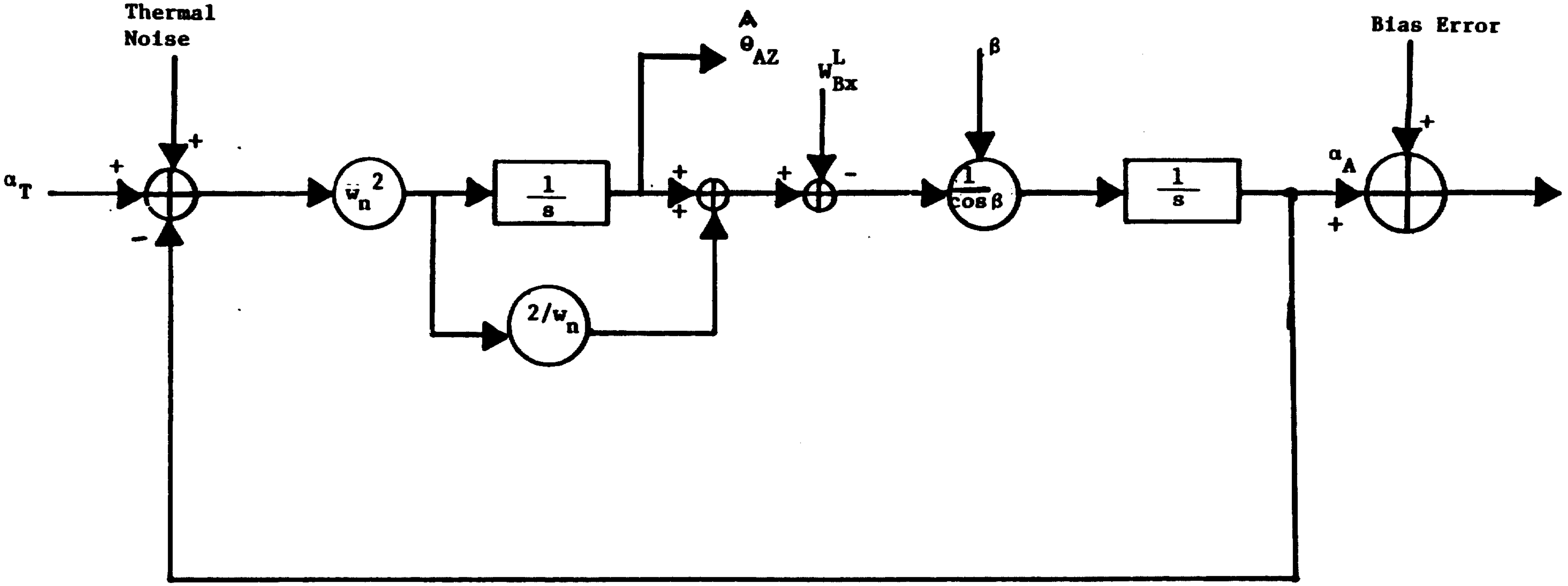


Figure 6-22 β - LOOP MODEL

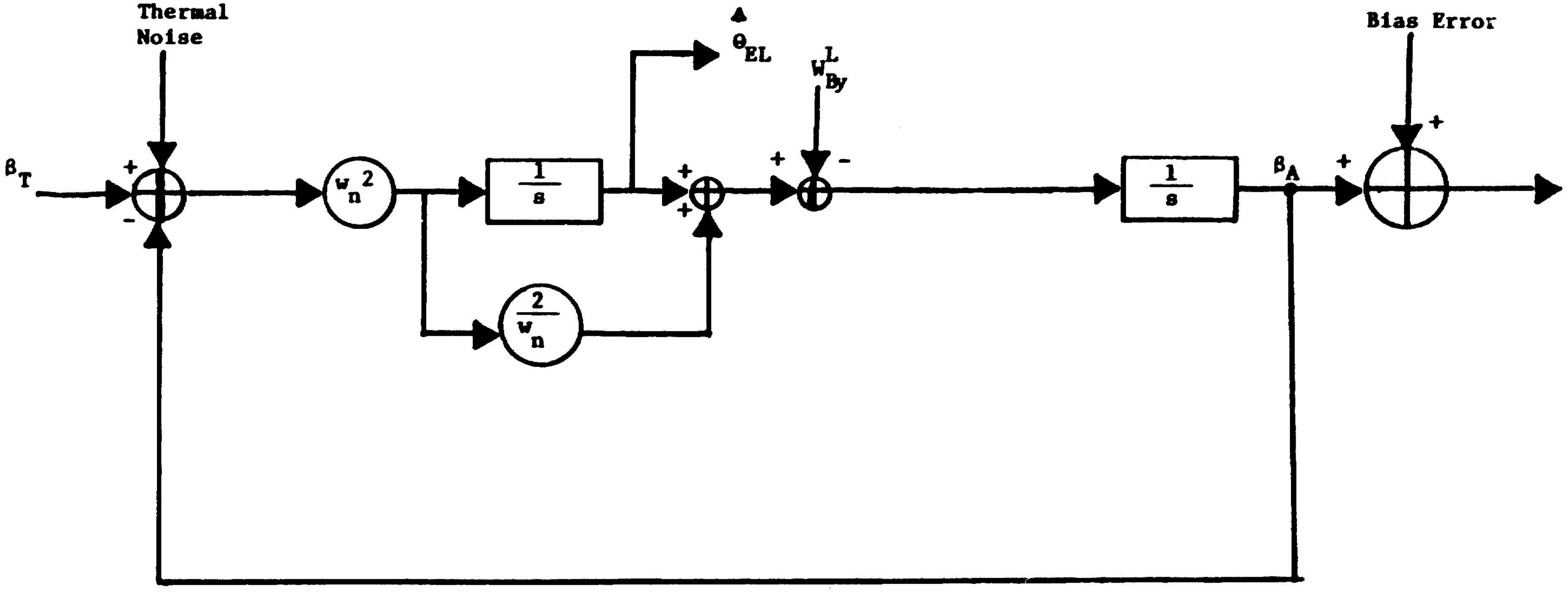


Table 6-5 ANGLE TRACKING LOOP CONSTANTS fn AND τ

RANGE INTERVAL,nm	fn, hz	τ,* SEC
Passive Mode		
$R > 9.5$	0.027	11.8
$3.8 \leq R \leq 9.5$	0.027	11.8
$1.9 \leq R \leq 3.8$	0.075	4.2
$R < 1.9$	0.12	2.7
Active Mode		
$R > 9.5$	0.027	11.8
$R \leq 9.5$	0.075	4.2

* $\rho = 1$ in the angle rate loop design. Therefore $\tau = 2/\omega_n$.

The α and β loop models shown in Figures 6-21 and 6-22 were adopted, with one modification, as the basis for the angle tracking performance computer model. The one modification is that the return signal is actually generated and processed to produce the angle discriminants (see section 6.4). This provides more flexibility and accuracy in the modeling of angle tracker error sources as discussed below. Introduction of the discriminant generation into the model requires calculation of the equivalent loop constant k_{eq}. This is done using the expression

$$k_{eq} = \frac{w_n^2}{4\, k_a \rho} \tag{6.46}$$

where w_n = loop natural frequency given in Table 6-5,

k_a = slope of normalized antenna difference pattern given in Figure 4-8,

$$\rho = \frac{1}{1+\mathrm{SNR}^{-1}}$$

and it is assumed that SNR >>1 so that ρ=1. Table 6-6 summarizes the results of the k_{eq} calculation for each value of w_n listed in Table 6-5.

6.6.2 Model Assumptions and Approximations

The analog models of the α and β tracking loops are based upon the following assumptions. In the area of the antenna electronics, the antenna gimbal motors are treated as perfect analog integrators and any filtering used for signal shaping, predistortion, or smoothing is assumed to work ideally. The rate stabilization loop is assumed to act instantaneously to remove the body inertial angular velocity from the estimates of the target inertial LOS azimuth and elevation rates. This is a reasonable assumption, since the rate stabilization loop bandwidth is much wider than the angle rate loop bandwidth. Also, any errors such as gyro drift or thermal noise introduced by the antenna electronics, are ignored.

Table 6-6 EQUIVALENT ANGLE TRACKING LOOP CONSTANTS k_{eq} and k'*

RANGE INTERVAL, nm	k_{eq}, deg/sec^2	k', deg/sec
<u>Passive Mode</u>		
$R > 0.5$	$2.0106 \times 1\bar{0}^{4}$	$2.3725 \times 1\bar{0}^{3}$
$3.8 \leq R \leq 9.5$	$2.0106 \times 1\bar{0}^{4}$	$2.3725 \times 1\bar{0}^{3}$
$1.9 \leq R \leq 3.8$	$1.5529 \times 1\bar{0}^{3}$	$6.5907 \times 1\bar{0}^{3}$
$R < 1.9$	$3.9750 \times 1\bar{0}^{3}$	$1.0546 \times 1\bar{0}^{2}$
<u>Active Mode</u>		
$R > 9.5$	$2.0106 \times 1\bar{0}^{4}$	$2.3725 \times 1\bar{0}^{3}$
$R \leq 9.5$	$1.5529 \times 1\bar{0}^{3}$	$6.5907 \times 1\bar{0}^{3}$

$k_{eq} = w_n^2/\left[4 \text{ (normalized difference pattern slope)}\right]$

$k' = k_{eq}\tau$

The signal generation and processing assumptions that affect the present version of the angle tracking model are (1) the choice of antenna sum and difference patterns, (2) ignoring the quantization noise introduced by the digital signal processor, and (3) ignoring the difference in microwave loss between the sum and difference channel. Neglecting the microwave loss difference will have a noticeable impact upon angle tracker performance at low SNR. This error source will be included if time permits. Quantization effects are ignored in all processor steps except the last one: computation of the angle discriminants. These discriminants are quantized to 3/16 dB accuracy. Thus, one of the major sources of quantization is included in the model and the impact of assumption (2) is not too significant. However, quantization of the discriminant does increase the importance of the antenna difference pattern selection, as this choice will affect the resolution capability of the angle rate estimator and the angle tracker.

The last major assumption involves the implementation of the continuous-time loops of Figures 6-21 and 6-22 on the computer. These models are approximated by the discrete-time loops shown in Figures 6-23 and 6-24. The fundamental assumption used in the discretization process was to replace the analog integrators by the digital integration model of Figure 6-25. Effect of this approximation is to constrain the antenna to move in a stair-case fashion. That is, the antenna position remains constant during an update period and is moved to the new predicted α and β at the beginning of the next update period. Therefore, the digital integrator approximation will be practical provided the commanded α and β rates are not too large.

Figure 6-23 DISCRETIZED α-LOOP

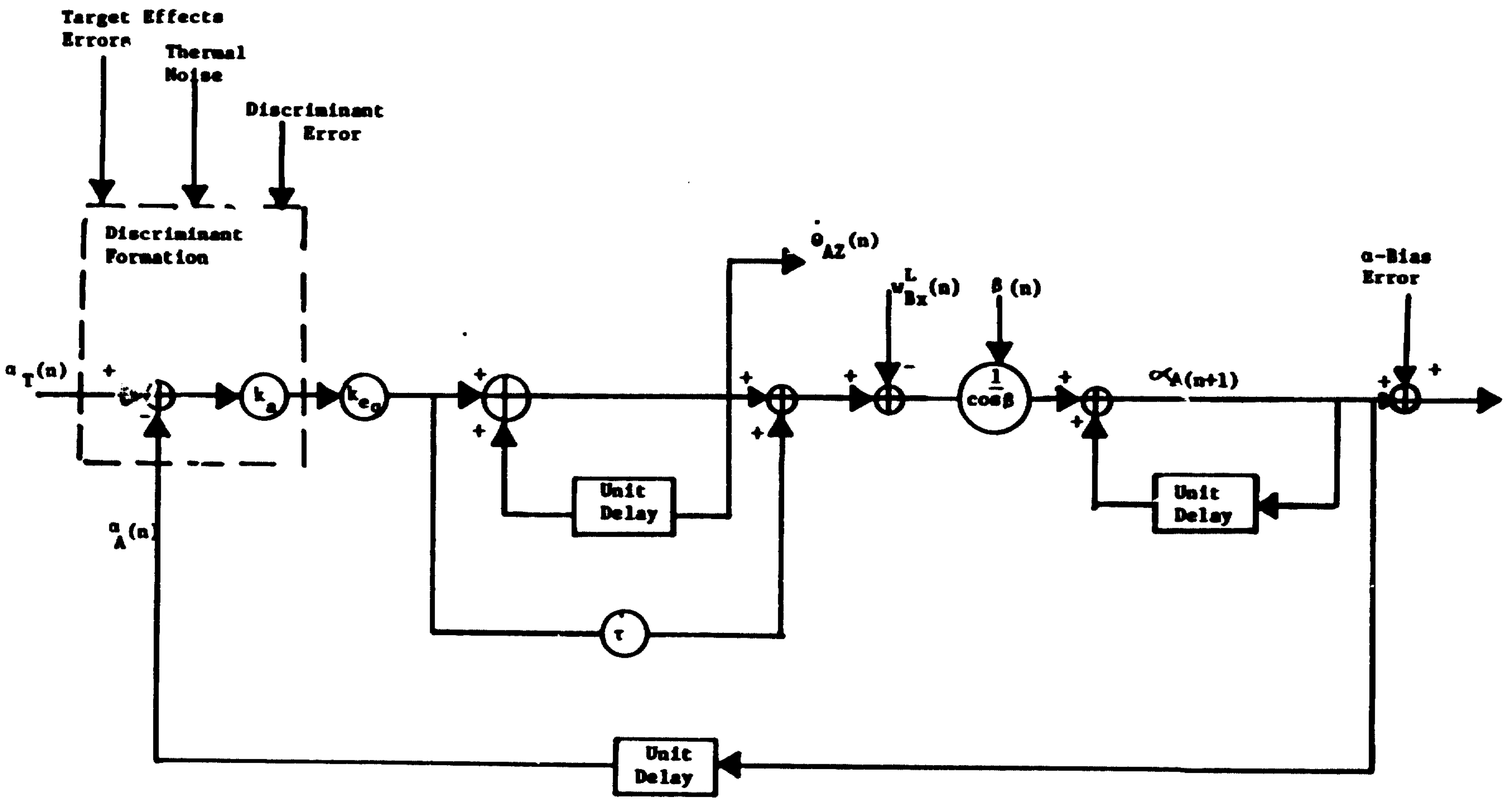


Figure 6-24 DISCRETIZED β-LOOP

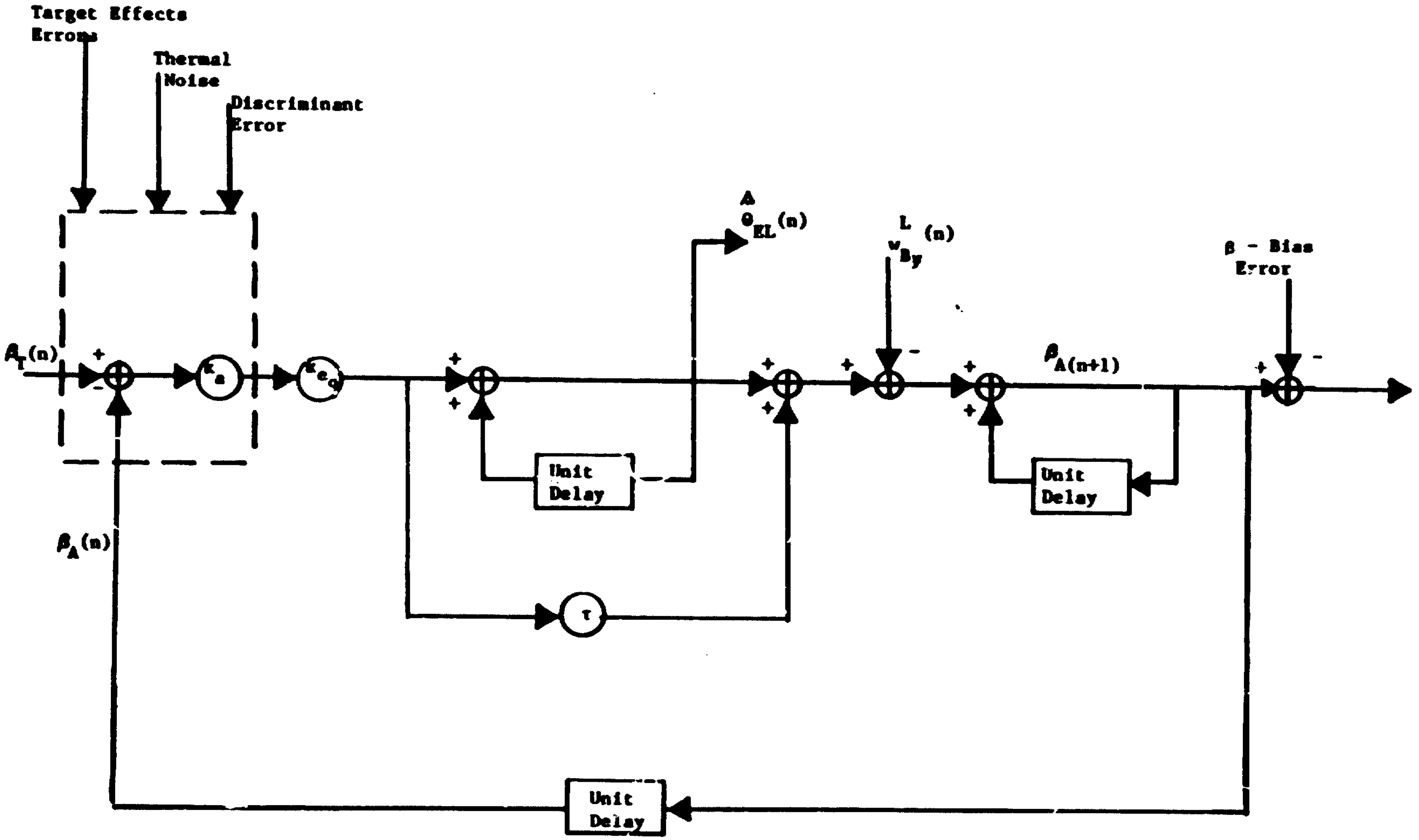
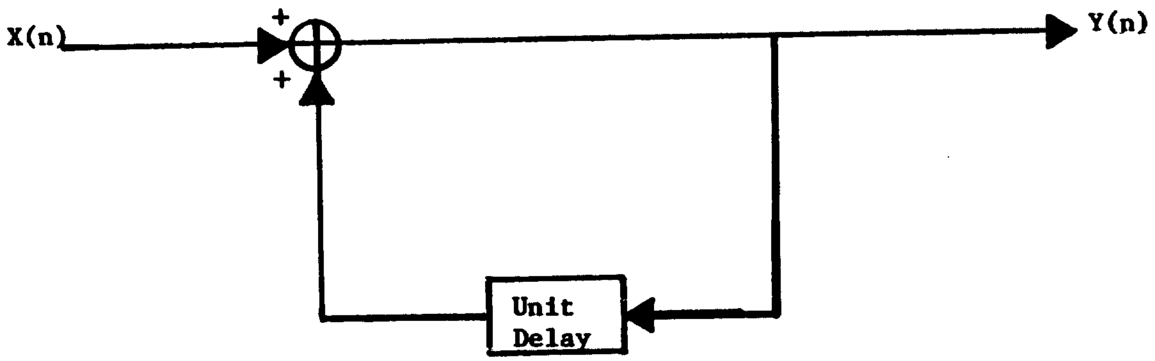


Figure 6-25 DISCRETE REPRESENTATION OF INTEGRATOR



6.6.3 Error Sources Modeled

Error sources in the Ku-Band radar angle and angle rate tracking loop that are modeled in the simulation include

- Target error effects (to the extent that the target scattering model is correct),
- Thermal noise,
- Boom deployment error,
- Radar offset error,
- Discriminant error,
- Gimbal bias error.

All of the errors listed above, except the gimbal bias error, are included in the generation of the angle discriminant. Target-induced angle and angle rate measurement errors are included by virtue of the fact that the return signal is generated and processed. Then, provided the target scattering model is accurate, the target-induced errors will be accurately represented. Thermal noise is based upon the receiver and signal proces or configuration; the exact method of computation is derived in Appendix D. Boom deployment and radar offset errors occur because the radar transforms the radar estimates of angle and angle rate to body coordinates assuming the radar is located at the oribiter body C.G. and the radar frame is yawed +67° with respect to the body frame. These two errors are included by computing the target return signal based upon a radar offset from the orbiter C.G. and then transforming the resulting radar estimates with the same equations that are used by the radar microprocessor. Discriminant error is the distortion introduced by the method of discriminant computation. This distortion is induced by (1) large target angle errors or (2) low SNR as illustrated in Figure 6-26. The last error source included in the model is gimbal bias error, i.e., the error in the gimbal position

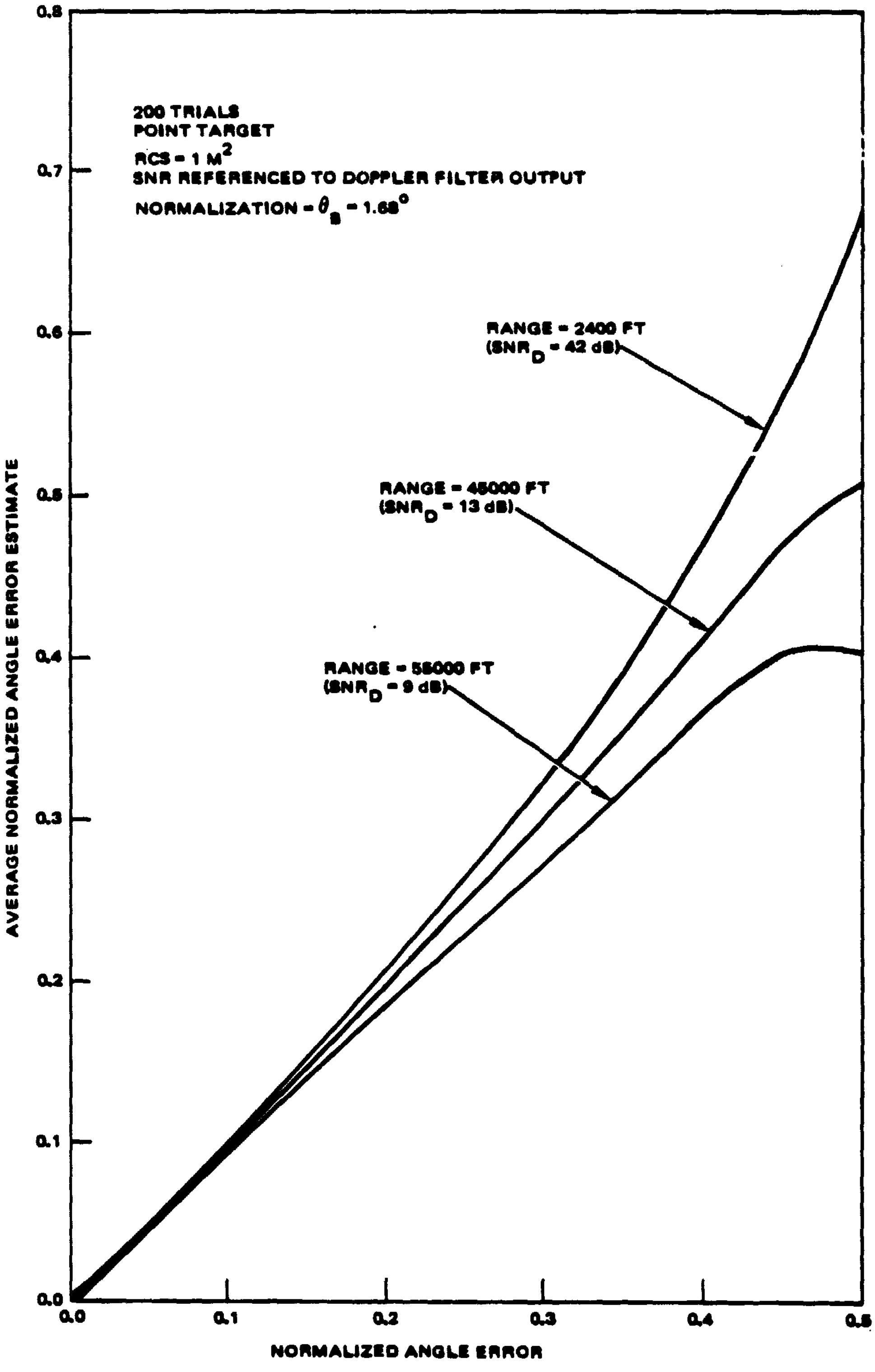


Figure 6-28. Angle Discriminant Test Results.

reading. This source is easily incorporated into the model as illustrated in Figure 6-21.

6.6.4 Model Performance

This subsection presents the results of tests of the angle and angle rate tracking model. Two tests of the angle tracking model were performed. In the first test, the response of the angle discriminant generator was computed as a function of angle error and SNR. This test was performed with the quantization removed. Results of this test are given in Figure 6-26 and they agree reasonably well with the theoretically predicted discriminant function given in [2] and [22]. The second test checked the step response of the angle rate estimator to determine whether it is behaving as a second order critically damped loop. This test made the following assumptions: (1) the SNR was much greater than unity, (2) the loop was run in an "analog" fashion, i.e. all quantization and discretization of the quantities involved was ignored, and (3) the α-loop was used for the test with β set equal to zero. Results of this test are shown in Figures 6-27 through 6-29 for each loop bandwidth. The 2% convergence times obtained from the computer model agree quite well with the theoretical values as obtained from the solution of the following transcendental equation

$$(6.47) \qquad (\alpha+1) = 0.02\, e^{\alpha}$$

where $\alpha = w_n t_2$,

w_n = loop natural radiar frequency,

t_2 = 2% convergence time .

We make the following disclaimer about the tests and tests results discussed above. These tests show only that the simulation model provides accurate representation of the theoretical design of the angle tracking loops. It does not prove that the model response will closely approximate the actual hardware response under all conditions.

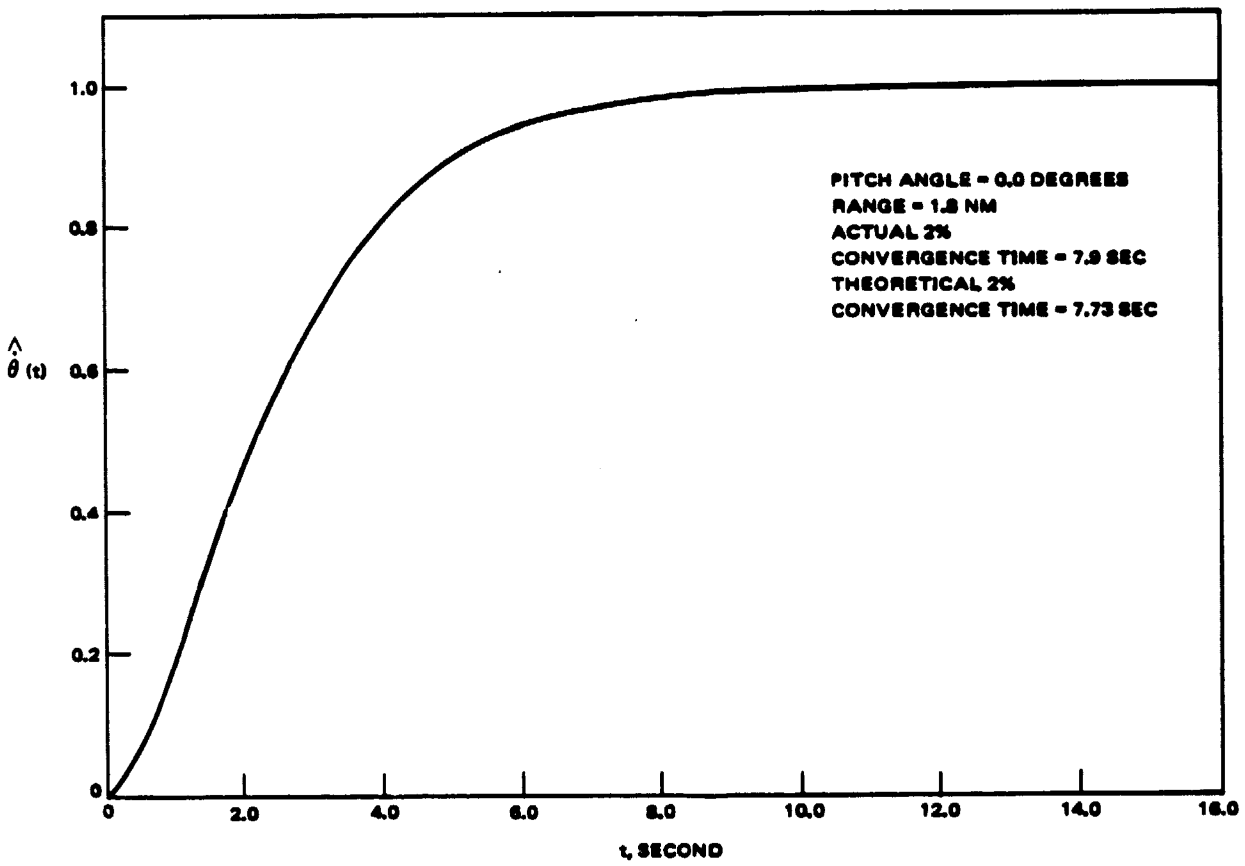


Figure 6-27. Angle Rate Loop Step Response (R < 1.9 n.mi.).

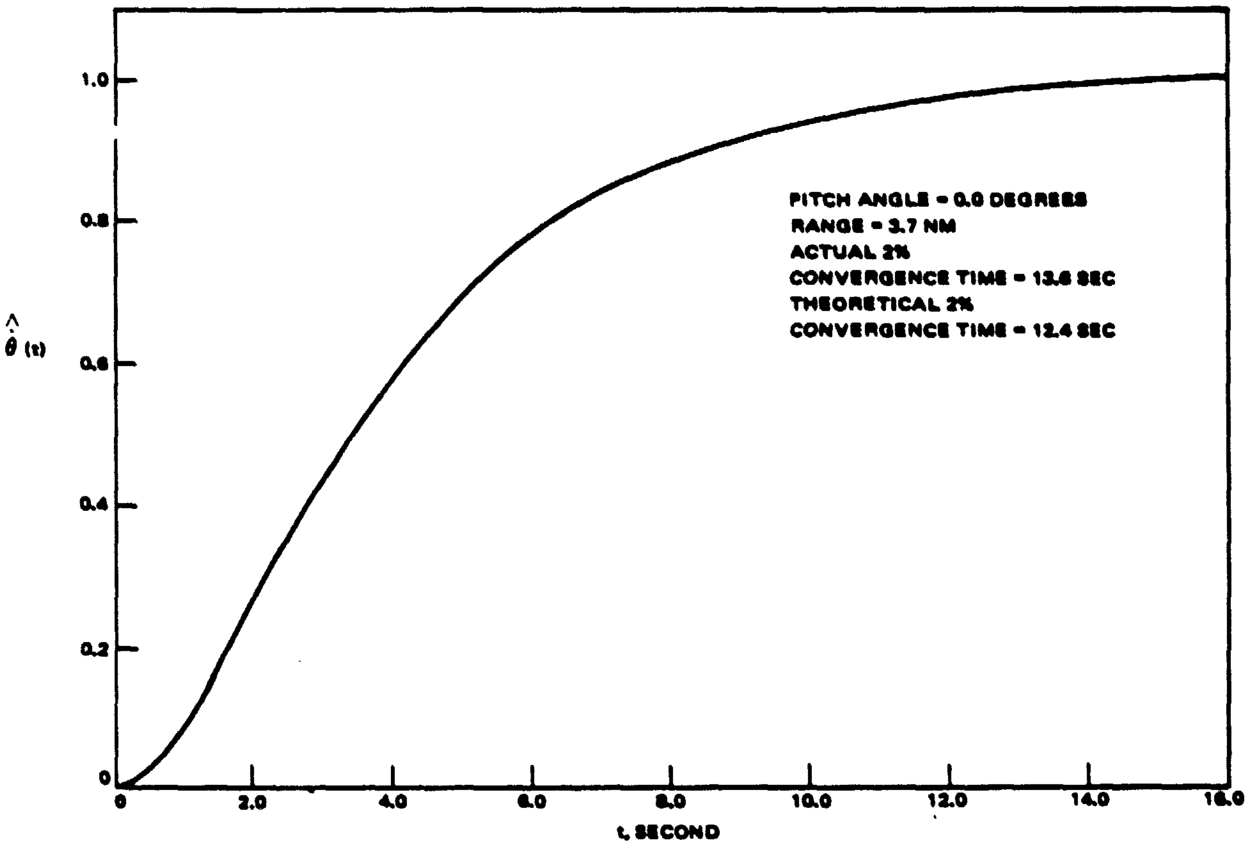


Figure 6-28. Angle Rate Loop Step Response (1.9 n.mi. < R < 3.8 n.mi.).

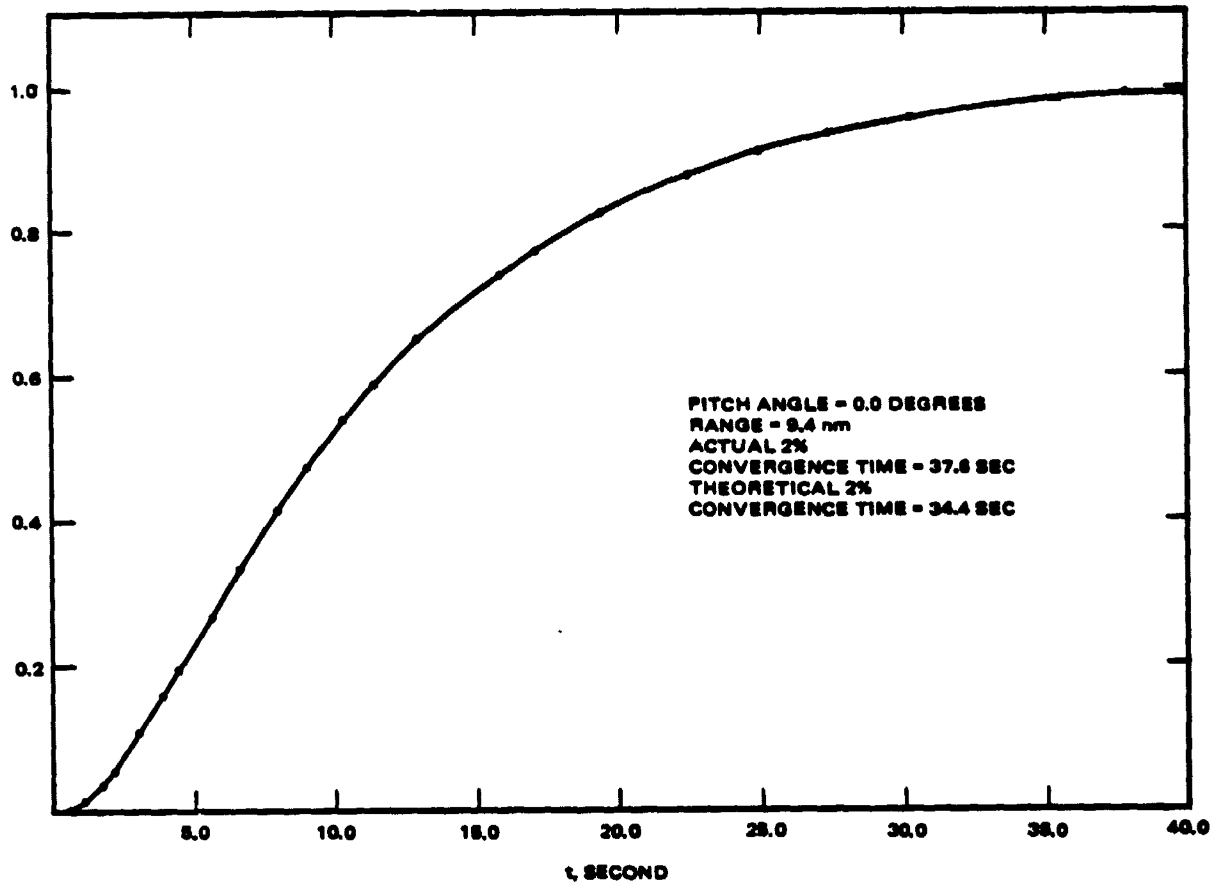


Figure 6-29. Angle Rate Loop Step Response. (3.8 nmi<R<9.5 nmi).

6.6.5 Computer Model Details

The computer model of the α and β tracking loops is broken into two distinct parts: (1) the generation of the discriminants and (2) using these new discriminants to generate new target inertial roll and pitch rates and new target roll and pitch angles. The first step is included in the signal generation and processing algorithm and was described in section 6.4. In this subsection we present a detailed description of step (2).

The algorithm used to update the angle rates and angles is shown in Figure 6-30. A stepwise description of the algorithm follows below. A preliminary step that is required prior to updating the angle and angle rate filter equation is to quantize the angle discriminants to 3/16 dB accuracy:

$$D_{AZ}(n) = \left[\frac{16}{3} D_{AZ}(n)\right]$$

$$D_{EL}(n) = \left[\frac{16}{3} D_{EL}(n)\right]$$

where $[\cdot]$ means the greatest integer in $\cdot$. Using this preliminary computation, we have

$$\hat{\dot{\theta}}_{AZ}(n) = \hat{\dot{\theta}}_{AZ}(n-1) + T_s K_{eq} D_{AZ}(n) \quad (6.49)$$

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

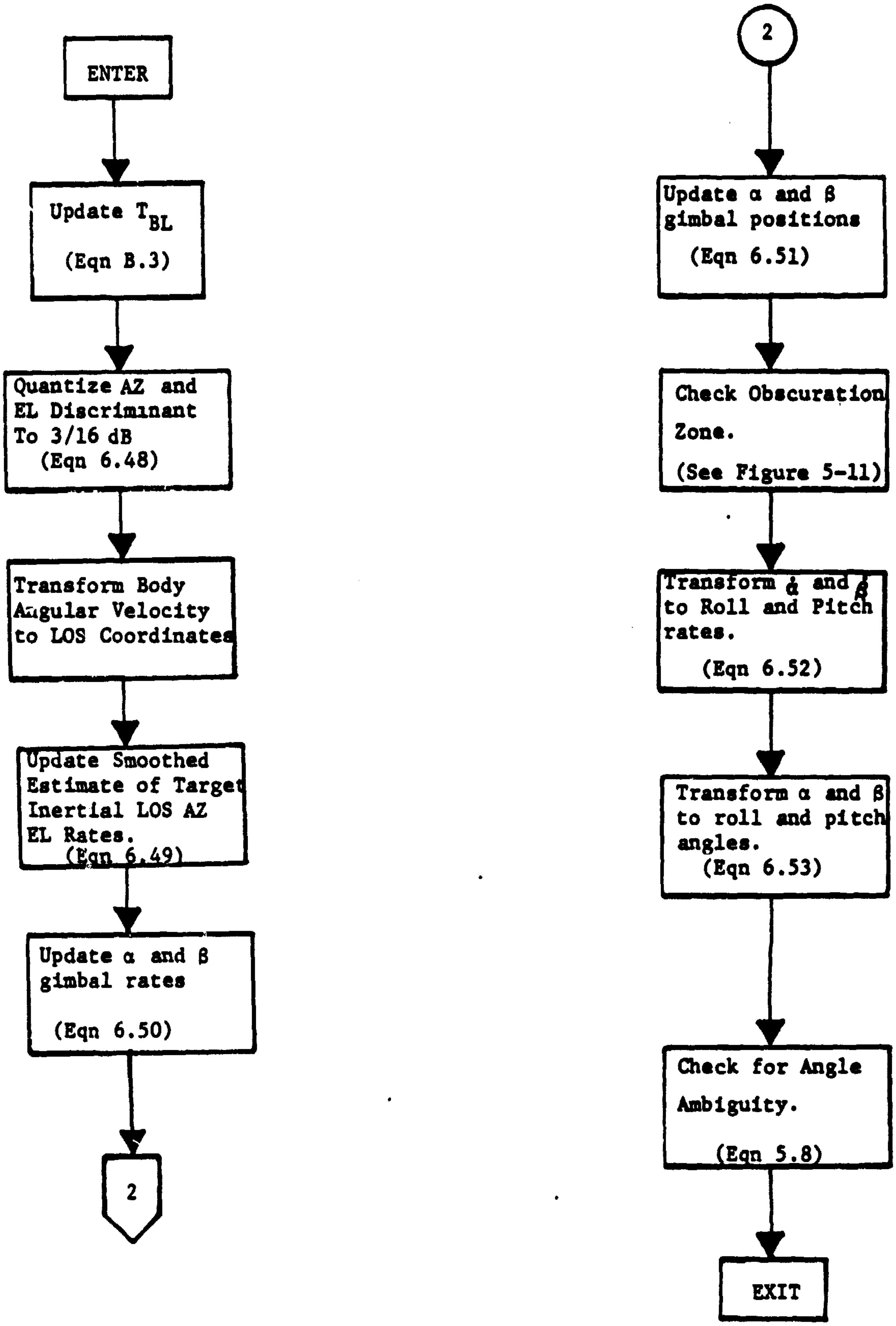
where $\hat{\dot{\theta}}_{EL} \triangleq$ smoothed target inertial LOS elevation rate,

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

$T_s =$ update interval,

$K_{eq} =$ loop constant computed from equation 6-46.

Figure 6-30 ANGLE AND ANGLE RATE TRACK LOOP FILTER COMPUTER ALGORITHM


ENTER
Update T_{BL} (Eqn B.3)
Quantize AZ and EL Discriminant To 3/16 dB (Eqn 6.48)
Transform Body Angular Velocity to LOS Coordinates
Update Smoothed Estimate of Target Inertial LOS AZ EL Rates. (Eqn 6.49)
Update α and β gimbal rates (Eqn 6.50)
2
2
Update α and β gimbal positions (Eqn 6.51)
Check Obscuration Zone. (See Figure 5-11)
Transform $\dot{\alpha}$ and $\dot{\beta}$ to Roll and Pitch rates. (Eqn 6.52)
Transform α and β to roll and pitch angles. (Eqn 6.53)
Check for Angle Ambiguity. (Eqn 5.8)
EXIT

The second step is to update the α and β gimbal rates, $\dot{\alpha}$ and $\dot{\beta}$. This is accomplished by subtracting thu body inertial angular velocity from the new estimate of the target inertial rate and transforming appropriately. Quantitatively, the new $\dot{\alpha}$ and $\dot{\beta}$ estimates are obtained from the expression

$$\text{(6.50)} \qquad \begin{aligned} \dot{\alpha}(n) &= \left[\omega^{L}_{TX}(n) - \omega^{L}_{BX}(n)\right] / \cos\beta \\ \dot{\beta}(n) &= \omega^{L}_{TY}(n) - \omega^{L}_{BY}(n) \end{aligned}$$

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.

Equations (6.50) are derived in Appendix A, section A.2 (see equations (A.7) and (A.9), respectively).

In the fourth step, we update the α and β gimbal positions to be used for the next update period. This is easily accomplished by using the digital approximation of the analog integrator illustrated in Figure 6-25 to obtain the expression

$$\text{(6.51)} \qquad \begin{aligned} \alpha(n) &= \alpha(n-1) + T_s\,\dot{\alpha}(n) \\ \beta(n) &= \beta(n-1) + T_s\,\dot{\beta}(n)\ . \end{aligned}$$

The fifth step is to transform the smoothed estimates of the target inertial LOS azimuth and elevation rates to target inertial roll and pitch rates. This is done using the present values of α and β and the expression

$$\text{Target Inertial Roll Rate} = -1000.\left(T_{BL}(1,1)\hat{\dot{\theta}}_{AZ}(n) + T_{BL}(1,2)\hat{\dot{\theta}}_{EL}(n)\right)$$

(6.52)

$$\text{Target Inertial Pitch Rate} = -1000.\left(T_{BL}(2,1)\hat{\dot{\theta}}_{EL}(n) + T_{BL}(2,2)\,\hat{\dot{\theta}}_{EL}(n)\right)$$

where T_{BL} is computed using $\alpha(n-1)$ and $\beta(n-1)$. The above equations are derived in Appendix B section B.3. The final step is to transform the present values of α and β gimbal position, $\alpha(n-1)$ and $\beta(n-1)$, to target roll and pitch angle in the orbiter body (B) frame. This is accomplished by the following expressions

$$\text{Target Roll Angle} = -\tan^{-1}\left[-\frac{T_{BL}(2,3)}{T_{BL}(3,3)}\right]57.29576$$

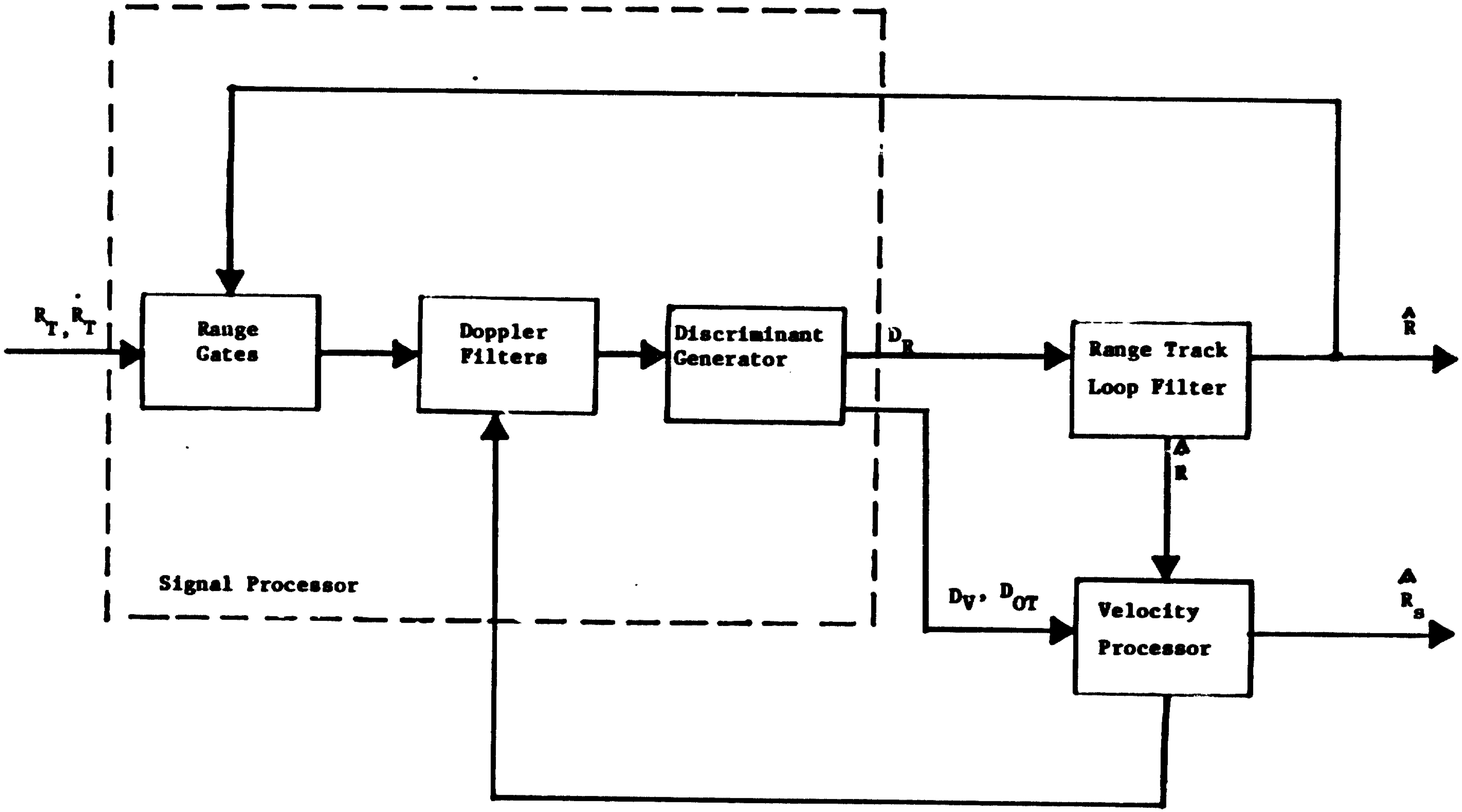
$$\text{Target Pitch Angle} = -\sin^{-1}\left[T_{BL}(1,3)\right]57.29576$$

which are derived in Appendix B section B.2. Once the new target roll and pitch angles have been computed, any ambiguity in these angles is removed using the relations (5.7).

6.7 RANGE AND RANGE RATE TRACKING MODEL DESCRIPTION

The range and range rate tracking simulation model is functionally identical to the Ku-Band radar range and range rate tracker. Figure 6-31 provides a simplified block diagram of the range and range rate tracking loop model. It is composed of three major algorithms: (1) the signal processor which generates the range and velocity discriminants, (2) a tracking loop filter which uses the range discriminant to produce estimates of the range and range rate, and (3) a velocity processor which uses the velocity discriminant and the rough range rate estimate to produce a very accurate estimate of the target

Figure 6-31 SIMPLIFIED DIAGRAM OF RANGE AND RANGE RATE TRACKING LOOP



velocity. The signal processing algorithm which generates the range and velocity discriminants has already been described in section 6.4. Therefore, this subsection will focus on the details of the track filter model and the velocity processor model.

6.7.1 Range Tracker Model Description

The range tracker algorithm is composed of a signal processing and a discriminant generator algorithm and a discrete-time range tracking filter algorithm. The signal processing and range discriminant generation algorithm closely approximate the corresponding function in the Ku-Band radar as discussed in section 6.4. The discrete-time tracking loop filter shown in Figure 6-32 is modeled exactly. This includes quantizing the range discriminant to 3/16 dB, quantizing the output range estimate to 5/16 feet, quantizing the output range rate to $\frac{5}{(16\,T_s)}$ feet per second where T_s is the update interval, and using the same values for m_a and m_b, the loop constants. These loop constants were calculated in [23] and are summarized in Table 6-7 for the various operating conditions.

Assumptions. One of the major simplifications in the range tracker involves the filtering at IF and baseband. It is assumed that the IF filters pass the perfect rectangular target return pulses without distortion. Also the baseband (or video) filter impulse response is assumed to be perfectly rectangular and of width equal to the A/D sample interval. Impact of these simplifications should be minimal. The only other assumption that might have some impact on model fidelity is neglecting the quantization noise contributed by the signal processing chain from the A/D to the discriminant generator. This assumption will have varying impact upon the model fidelity, depending upon target return signal strength.

Figure 6-32 RANGE TRACKING LOOP DISCRETE-TIME FILTER

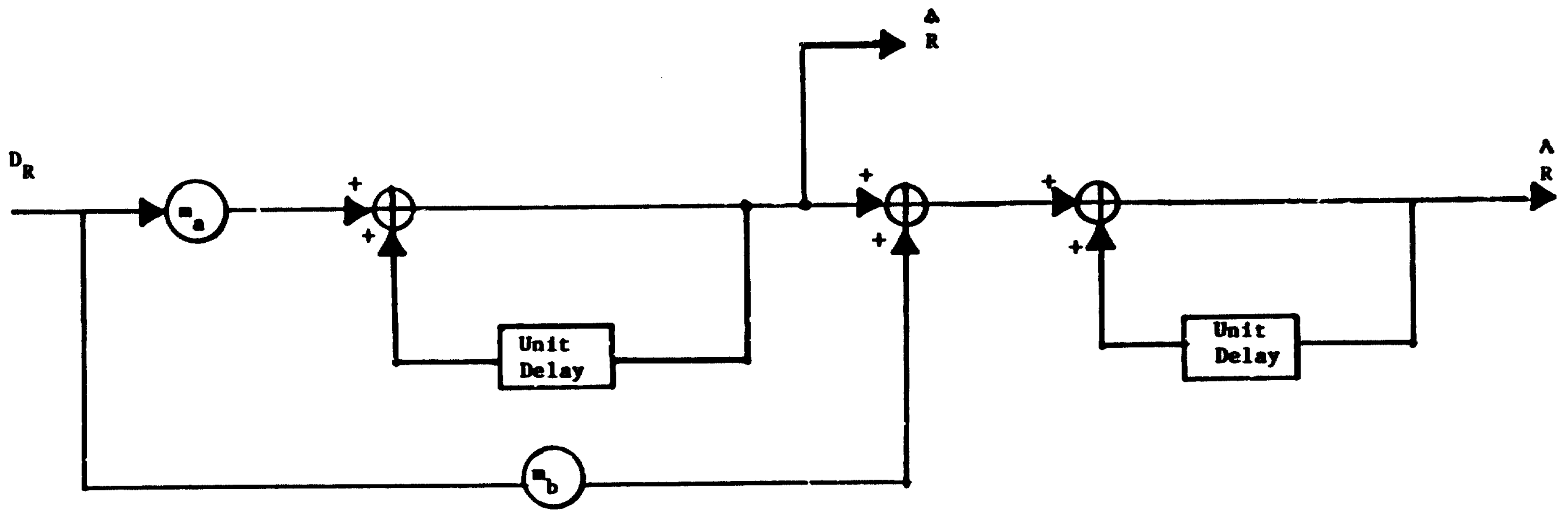


Table 6-7 EQUIVALENT RANGE TRACKING LOOP CONSTANTS m_a and m_b*

RANGE INTERVAL,nm	m_a	m_b
Passive Mode		
$R > 9.5$	16.0	0.25
$3.8 \leq R \leq 9.5$	16.0	0.5
$1.9 \leq R \leq 3.8$	16.0	2.0
$0.95 \leq R \leq 1.9$	8.0	1.0
$0.42 \leq R \leq 0.95$	8.0	2.0
$R < 0.42$	0.5	0.125
Active Mode		
$R > 9.5$	4.0	0.25
$R \leq 9.5$	0.5	0.125

Error Sources. Error sources incorporated into the range tracking model include

- Target-induced errors,
- Thermal noise,
- Discrimina : Distortion
- Range bias.

Target induced errors, thermal noise, and discriminant error are included in the range discriminant computation. As noted in the angle tracker model discussion, the targec induced errors will be accurate to the extent that the scattering model for the target is accurate. Thermal noise is computed using the model discussed in section 6.4 and derived in Appendix D. Range bias is treated as a fixed number representing errors in the system time delay calibration and time-varying system delays.

Model Performance. Testing of the range tracker model is analogous to the angle tracker model testing. That is, the range discriminant was checked for accurate performance and the tracking loop was tested for proper design and loop constant values. The rules for the range discriminant test were the same as the angle discriminant test: quantization was ignored and the discriminant was computed for several values of range error and SNR. Results of the discriminant computations are shown in Figure 6-33 and agree with the theoretical calculations shown in reference [2].

The second test verifies that the loop is operating as designed and that the constants are correct. It is performed by applying a constant range acceleration to the target and computing the range response. The range tracker should respond with a steady state range estimate bias error that is related to the value of β (or m_B) and the target range acceleration by the expression (taken from reference [24]),

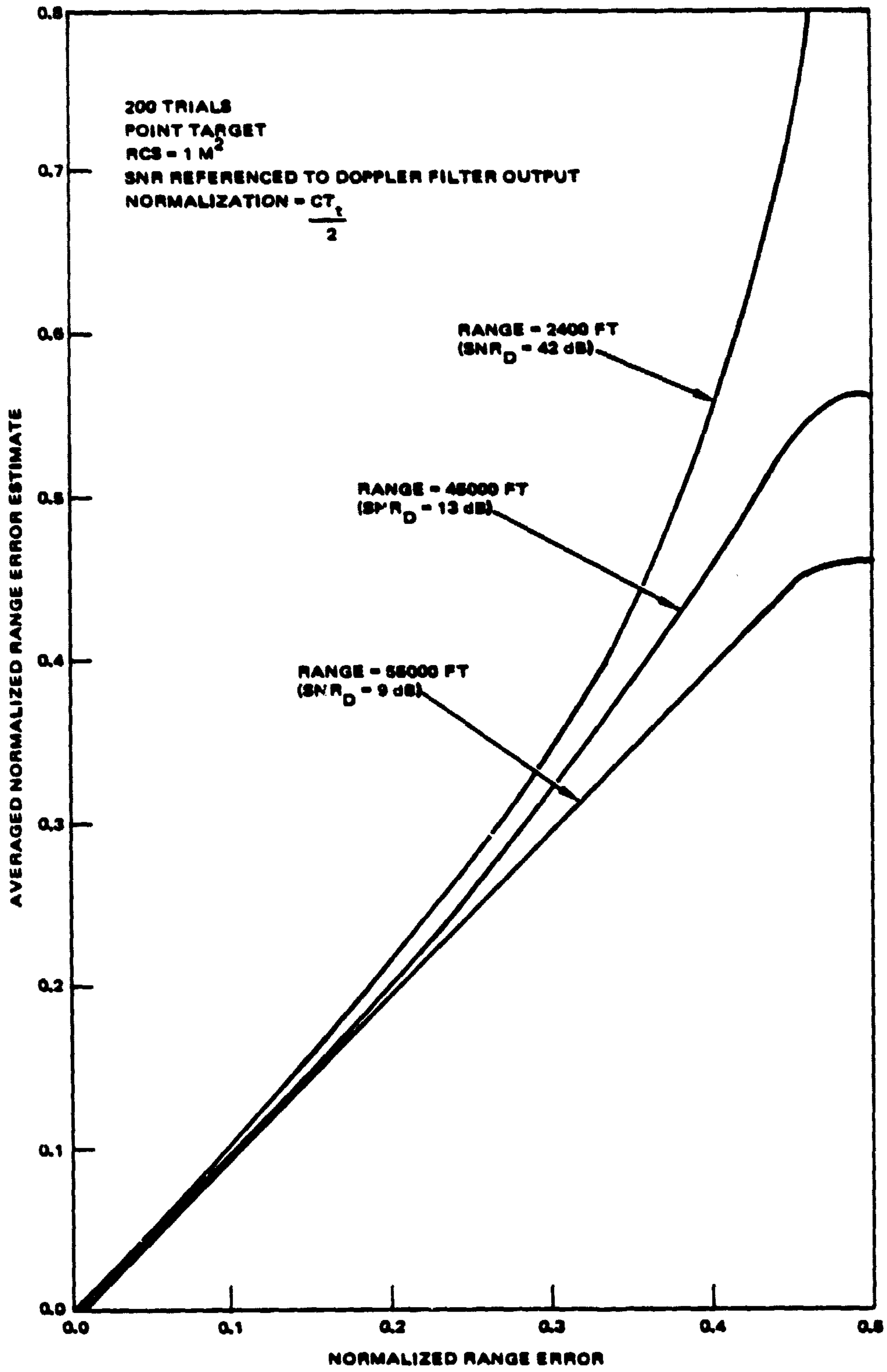


Figure 6-33. Range Discriminant Test Results.

(6.54) $$\text{Range Bias Error} = -\frac{a\,T_s^{2}}{\beta}$$

where a = value of target range acceleration,

$$\beta = \frac{m_b\ 80}{ct_t\ \ln 2}$$

$$\rho = \frac{1}{1+\mathrm{SNR}^{-1}}$$

T_S = update interval.

The test described above was performed under the following assumption: the range tracker was operated in "analog fashion". That is, the discriminant was not quantized and all multiplications and additions were done in floating point arithmetic. The results of accelerating the target at 10 fps^2 are shown in Figures 6-34 through 6-37 for range intervals possessing different β values. These data are in excellent agreement with the theoretically predicted results, indicating proper operation of the range tracking loop.

6.7.2 Velocity Processor Model Description

Model. The simulation velocity processor algorithm is functionally identical to the Ku-Band radar velocity processor algorithm. A simplified block diagram of the algorithm is shown in Figure 6-38. It is composed of two major tasks: (1) determination of the unambiguous velocity estimate and (2) updating the position of the doppler filter bank.

As shown in Figure 6-38, the first task is accomplished by computing the ambiguous velocity estimate and then using this estimate and the rough range rate estimate $\hat{\dot{r}}$ from the range tracker to determine the unambiguous velocity estimate. Figure 6-39 gives a block diagram of the algorithm used to compute the ambiguous velocity estimate. The basic idea of this algorithm

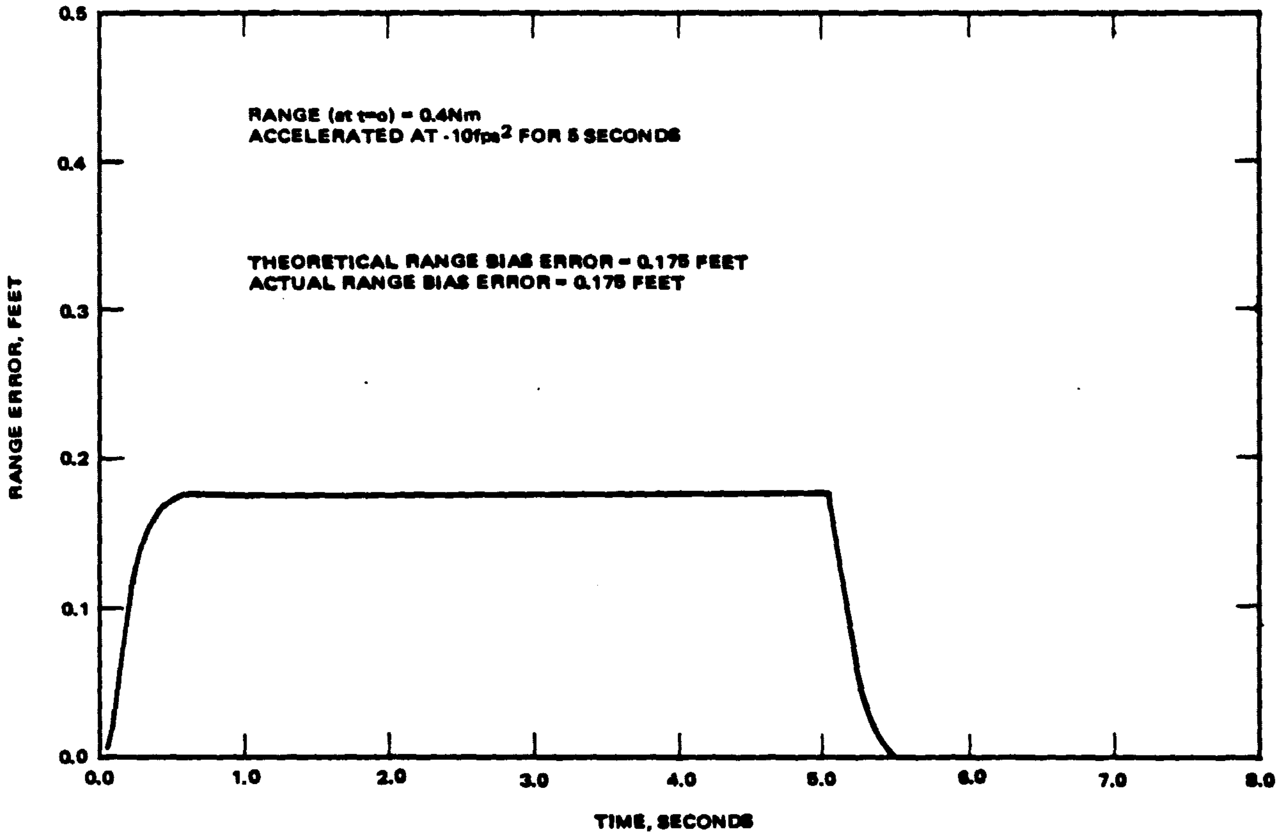


Figure 6-34. Range Tracking Loop Transient Response for Ranges Less Than 0.42 NM.

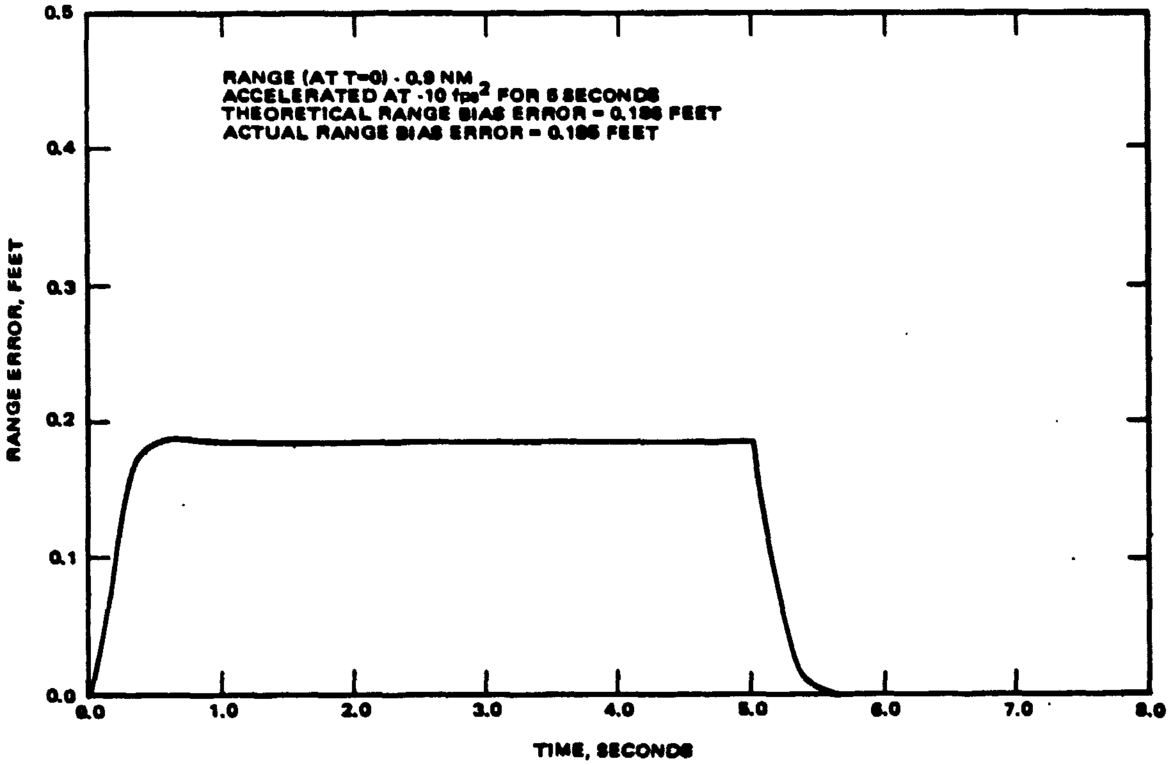


Figure 6-35. Range Tracking Loop Transient Response for Ranges 0.42NM<R<9.95NM.

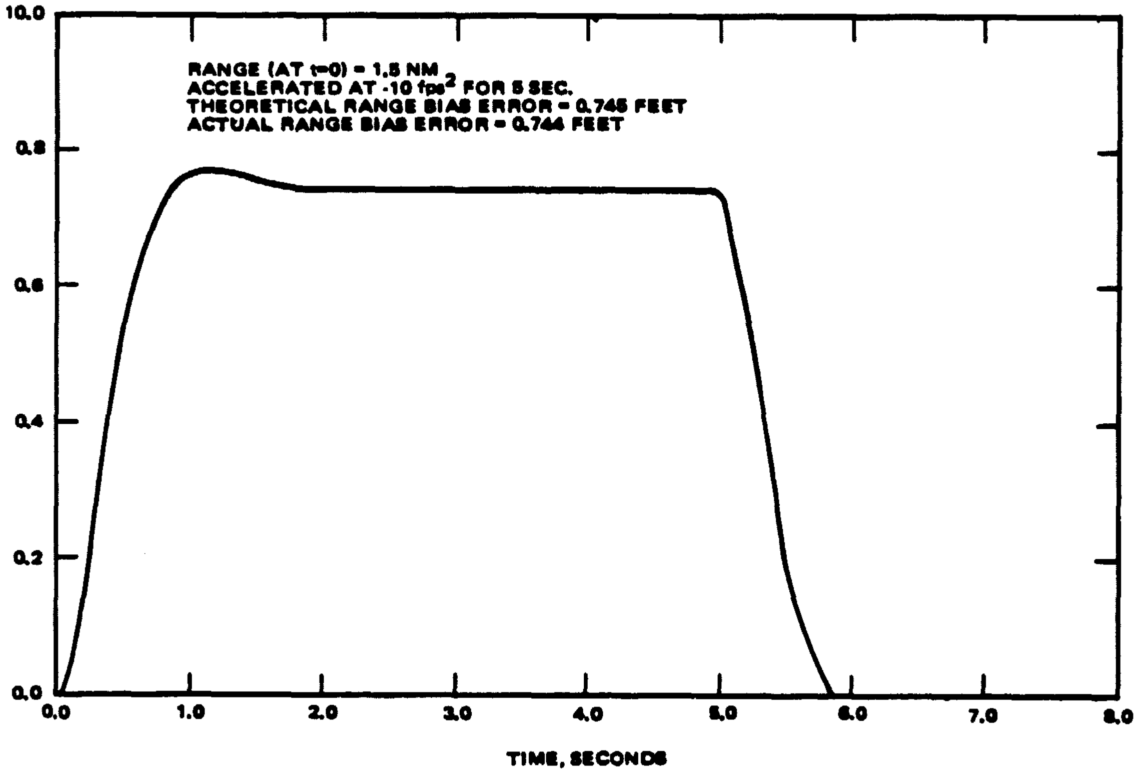


Figure 6-36. Range Tracking Loop Transient Response for Ranges 0.95≤R<3.8 NM.

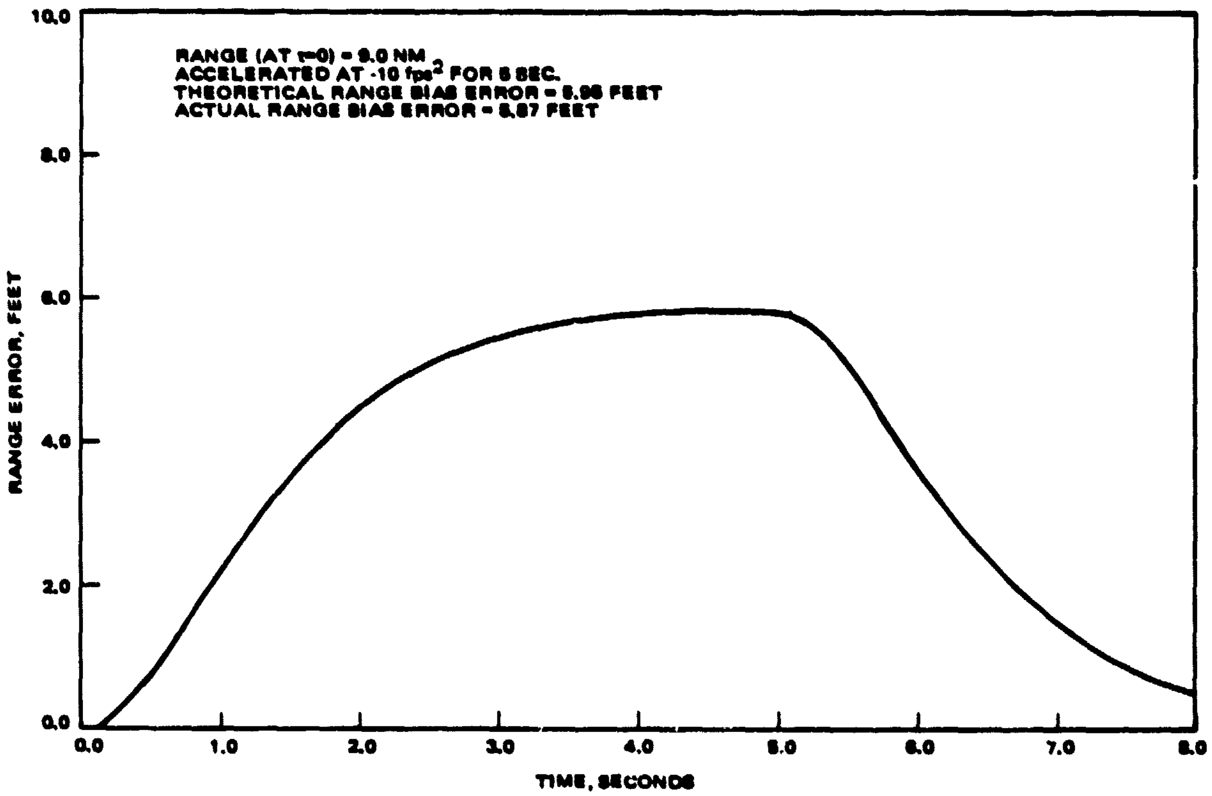


Figure 6-37. Range Tracking Loop Transient Response for Ranges 3.8 NM<R<9.5 NM.

Figure 6-38 KU-BAND RADAR VELOCITY PROCESSOR

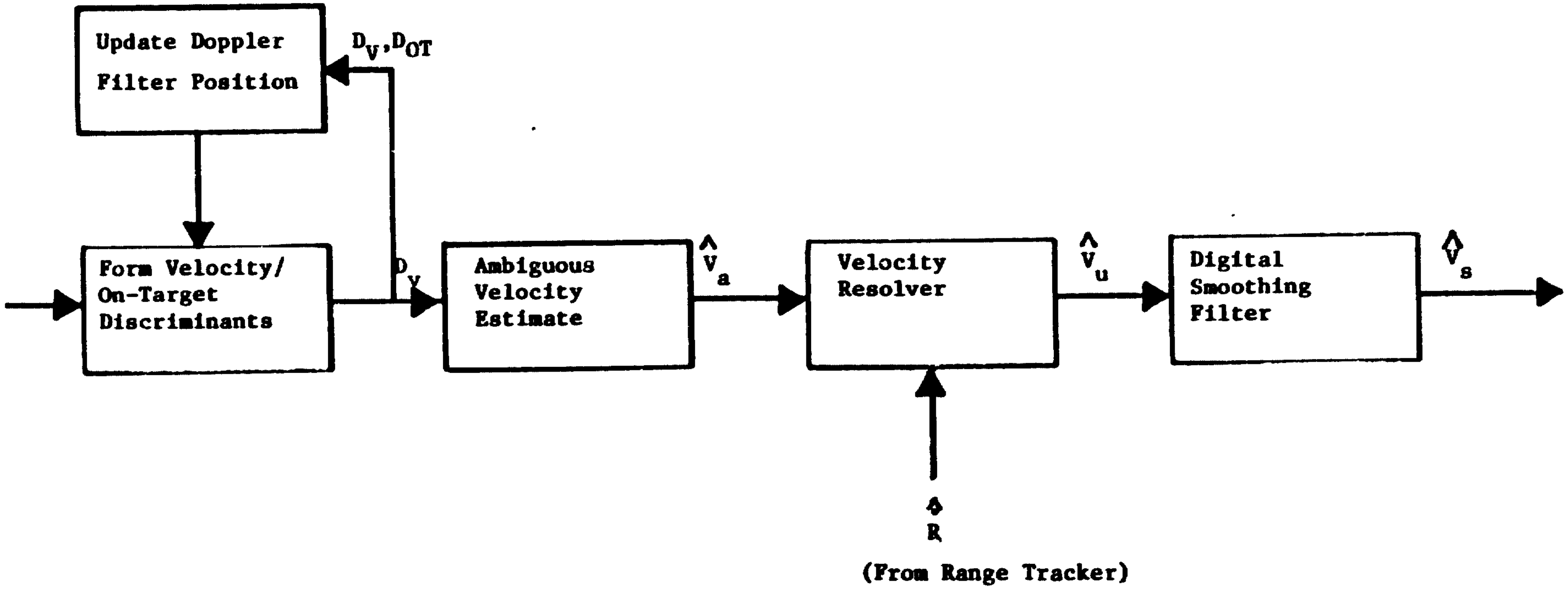
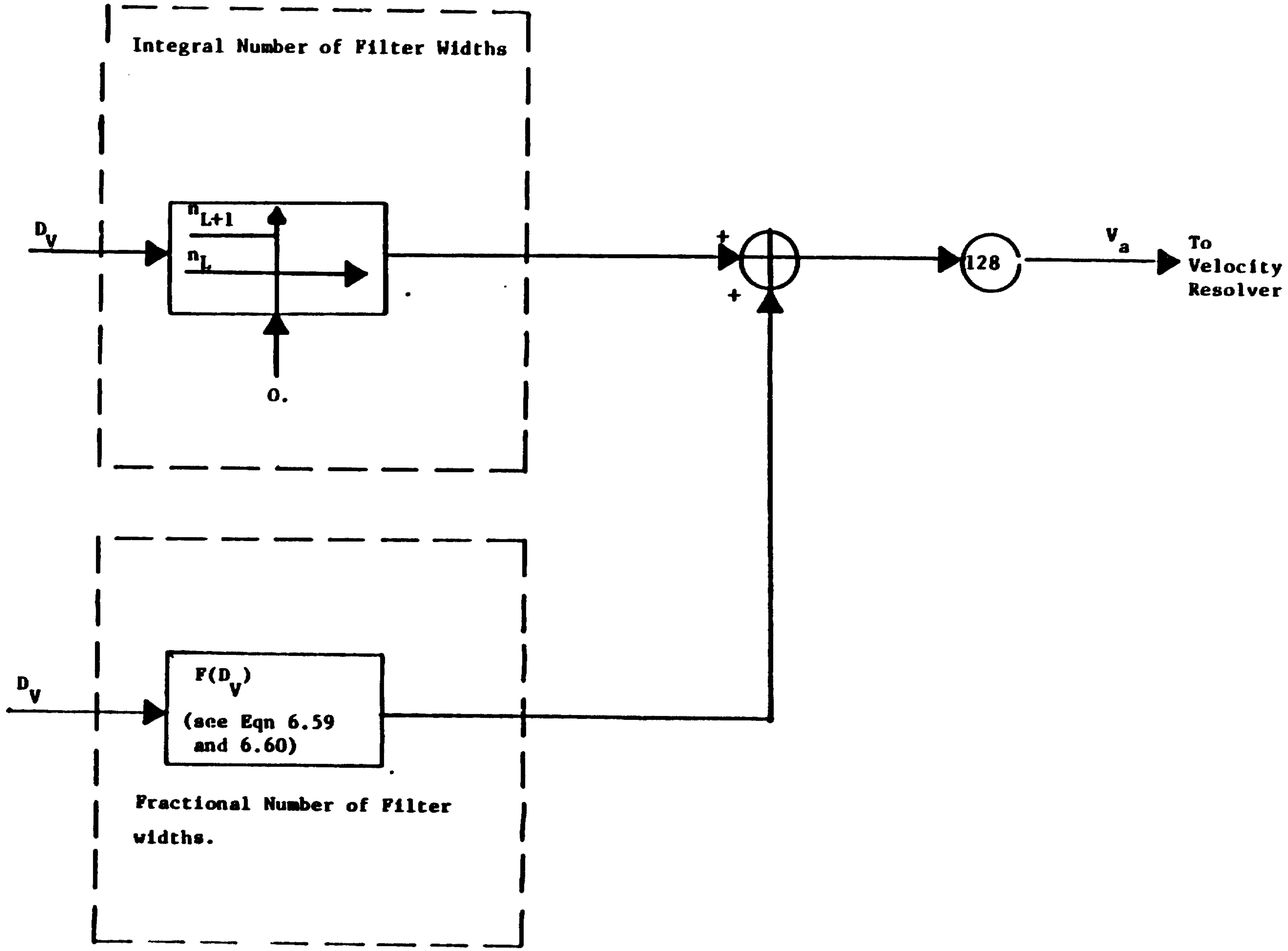


Figure 6-39 SIMPLIFIED DIAGRAM OF AMBIGUOUS VELOCITY ESTIMATION PROCESS



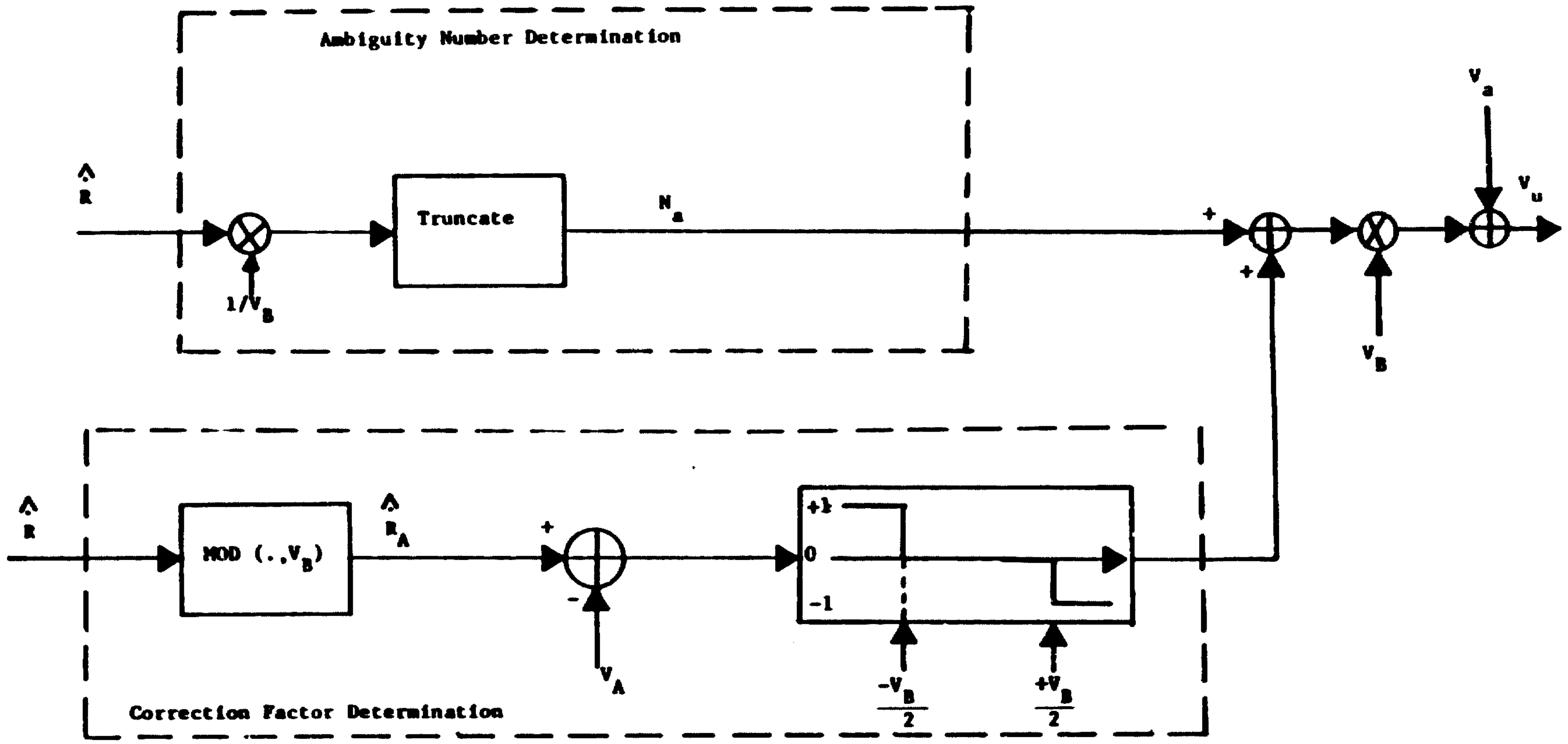
is as follows: compute the integral number of filter widths between zero frequency and the target location, combine it with the fractional filter width that remains, and scale appropriately to obtain the ambiguous velocity estimate. It is noted that the fractional part is determined to an accuracy of 1/128 of a filter width in all cases.

The ambiguous velocity resolver algorithm is given in Figure 6-40. This algorithm operates using the following principle. The ambiguous velocity is used to give a very accurate location of the target in the doppler filter bank and the rough range rate estimate is used to estimate the integral number N_a of filter banks that the target velocity is removed (either up or down) from zero frequency. The unambiguous velocity is then obtained by combining the fractional filter bank width with the integral number of filter bank widths and scaling appropriately.

It is worth mentioning here that the resolver has some additional protection against inaccurate determinations of N_a (the number of filter bank ambiguities) caused by noisy $\hat{\dot{r}}$ values, especially when the target velocity falls near either edge of the ambiguous filter bank. The portion of the resolver algorithm that provides this protection is enclosed in dashed lines in Figure 6-40 and works in the following way. If the computed position of the rough range estimate in the ambiguous filter bank, call it $\hat{\dot{r}}_a$, is more than half a filter bank (16 filters) from the ambiguous velocity estimate, then the ambiguity number N_a is increased or decreased by one, depending upon the sign of the difference between v_a and $\hat{\dot{r}}_a$.

The other major task of the velocity processor algorithm is to update the position of the five adjacent doppler filters, always maintaining the target in the center filter (provided target acceleration is not too great). The initial position of the filter set is determined by the filter in which

Figure 6-40 SIMPLIFIED DIAGRAM OF VELOCITY RESOLUTION PROCESS



NOTE: $V_B = \left| \frac{\lambda_c \cdot PRF}{2} \right|$

target detection occurred. This position is then updated during track using the algorithm shown in Figure 6-41. Depending on the values of the velocity and on-target discriminants, the position can be moved by 0, ± 1, or ± 2 filter widths. The exact decision algorithm is given in the figure.

Assumptions. Modeling assumptions that affect the velocity processor are that accele-ation of the target is not allowed during a data cycle and quantization error contributed by the signal processing chain from the A/D to the discriminant generator is ignored. Target acceleration during a data cycle causes broadening of the signal energy spectrum (a spreading over the doppler filter outputs), causing some degradation in velocity processor performance. Thus, the zero-acceleration constraint will give an optimistic estimate of performance in those cases where the target is accelerating. The effects of neglecting the quantization error has not been analyzed yet.

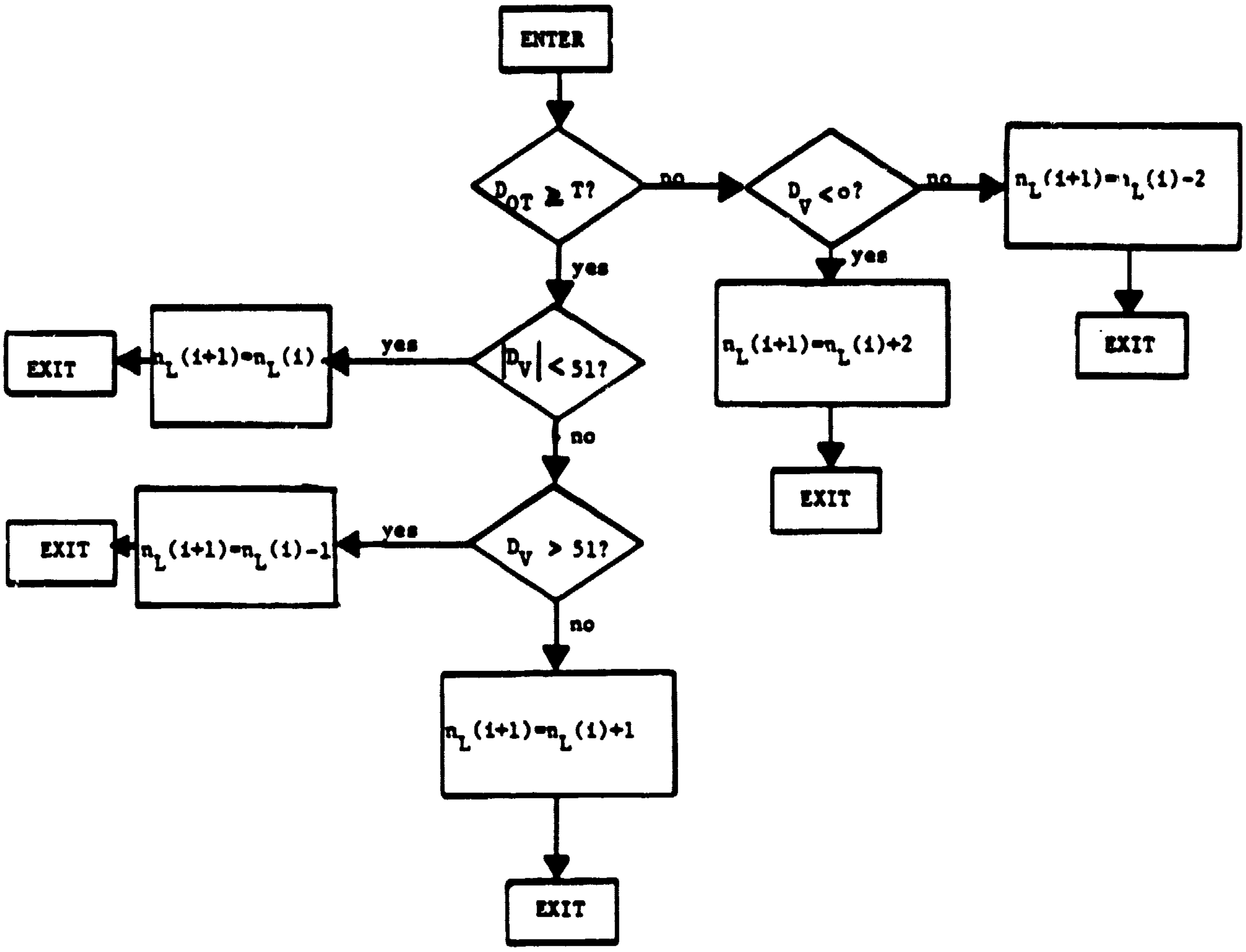
Error Sources Modeled. Velocity processor error sources include

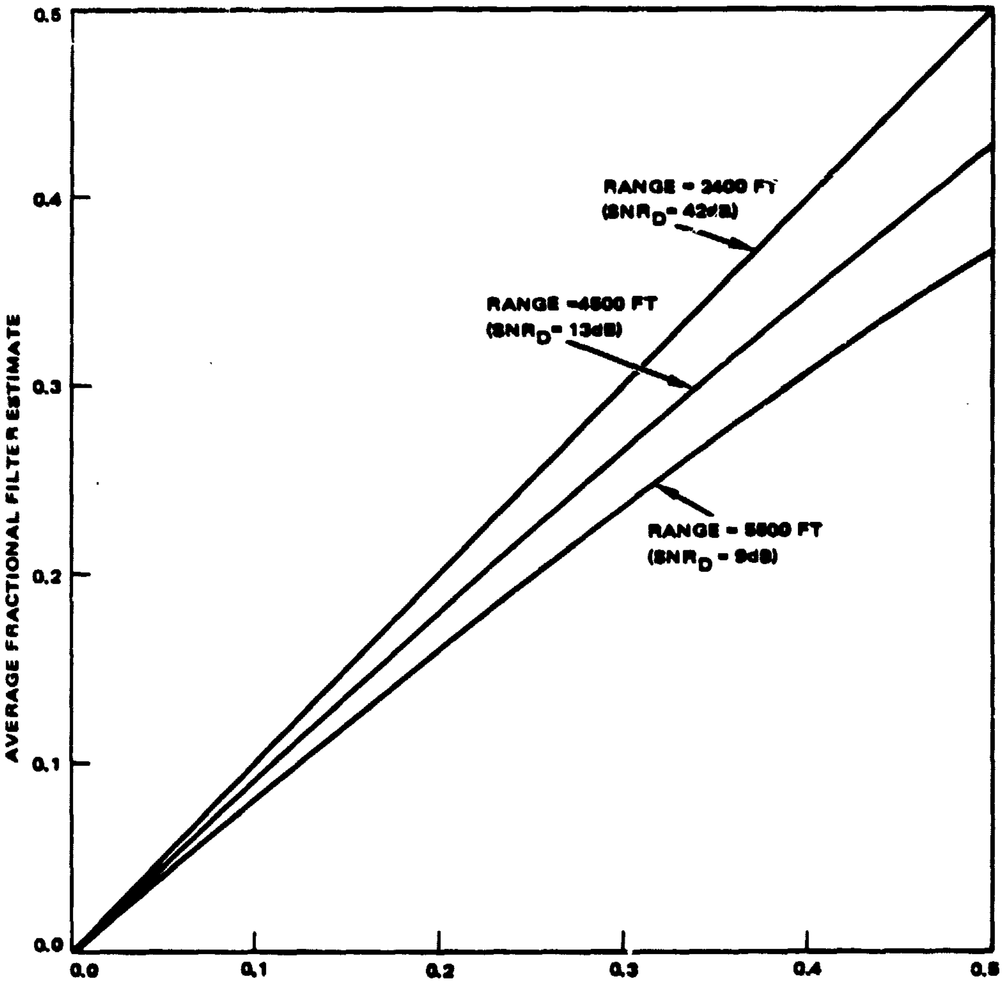
- Target-induced errors,
- Thermal noise,
- Discriminant distortion.

All three of these errors are included in the computation of the velocity and the on-target discriminants. Target-induced error modeling is achieved in the same manner as in the angle and range tracker models. Thermal noise is injected using the method described in section 6.4 and Appendix D. Discriminant error is generated by using an accurate discriminant computation model.

Model Performance. Performance of the ambiguous velocity estimator. was tested in the following way. The target estimated ambiguous velocity was computed as a function of target position over a filter width for high and low SNR values. Results of this test are shown in Figure 6-42 and agree with the theoretically predicted performance given in [25].

Figure 6-41 FILTER POSITION UPDATE ALGORITHMp





200 TRIALS
POINT TARGET
RCS = $1m^2$
SNR REFERENCED TO DOPPLER FILTER OUTPUT

Figure 6-42. Velocity Discriminant Test Results.

The ambiguity resolver has not been tested. However, it is noted that for ranges less than 12 nm (7 kHz PRF) the error in the rough range rate estimate would have to exceed ± 123 feet per second before the ambiguity number is in error.

6.7.3 Computer Algorithm Details

Construction of the range and range rate tracking loop computer model is identical to the angle and angle rate tracking loop computer model. That is, the computer model is broken into two distinct parts. One set of algorithms is dedicated to generation and processing of the target return signal to produce the required discriminants. These algorithms were described in section 6.4. The other part of the model is dedicated to updating of the range and velocity estimates and updating of internal control parameters. This part is described in this subsection.

Figure 6-43 gives the range tracker and velocity processor computer model. This algorithm is divided into four tasks: (1) updating the tracking loop filter difference equations which give the latest estimate of the range and rough range rate, (2) ambiguous velocity determination, (3) unambiguous velocity determination and (4) updating of the system internal control parameters. Each of these tasks are described in detail below.

Range and Rough Range Rate Estimate Update. The first step is to quantize the range discriminant to 3/16 dB using

$$D_R(n) = \left[(16/3)\ D_R(n) + 1/2 \right] \qquad (6.55)$$

where $[\cdot]$ means take the greatest integer in $\cdot$ .Then, the range and range rate estimates are updated using the difference equations

Figure 6-43 RANGE AND RANGE RATE TRACKING LOOP COMPUTER ALGORITHM (1 of 2)

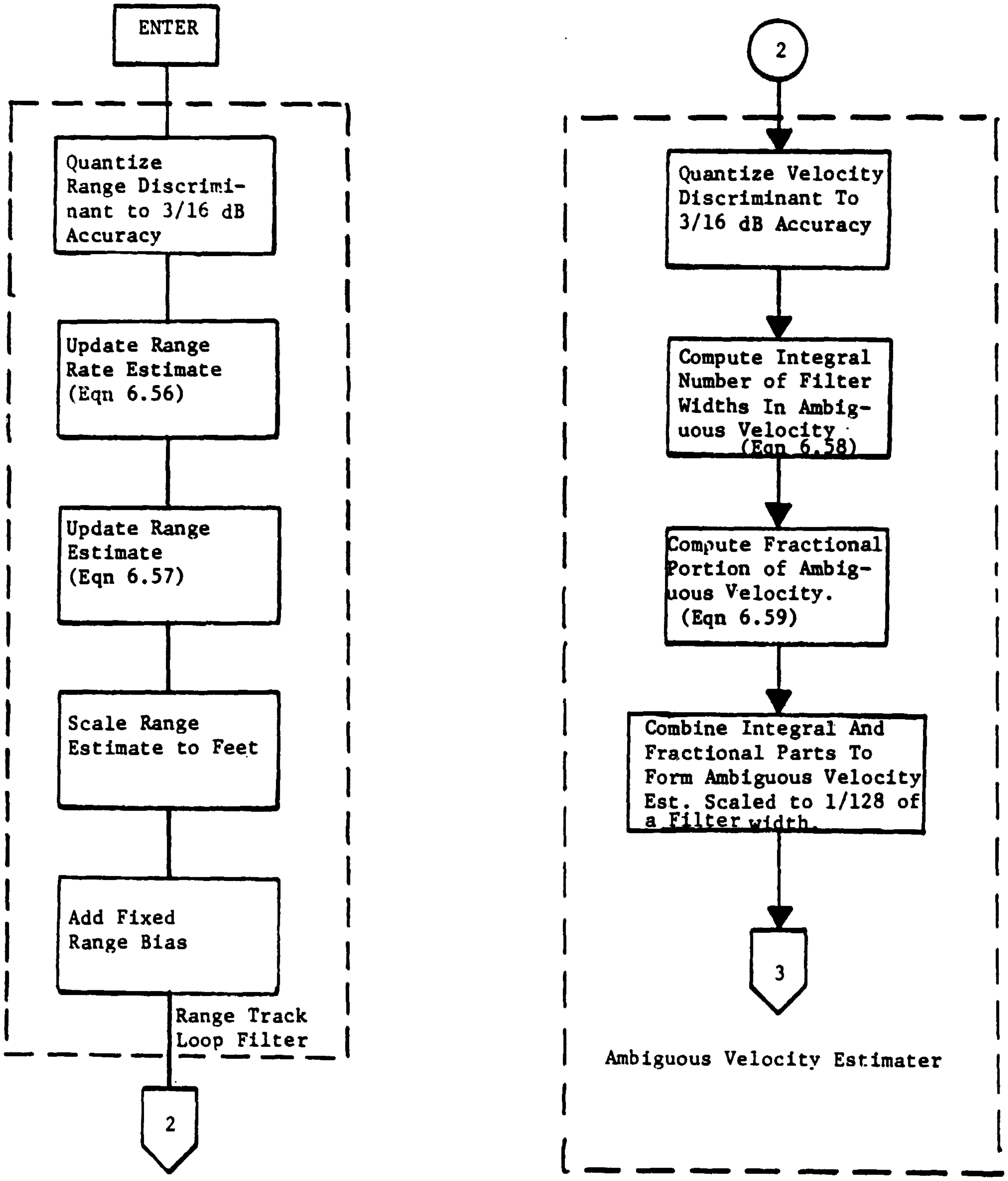
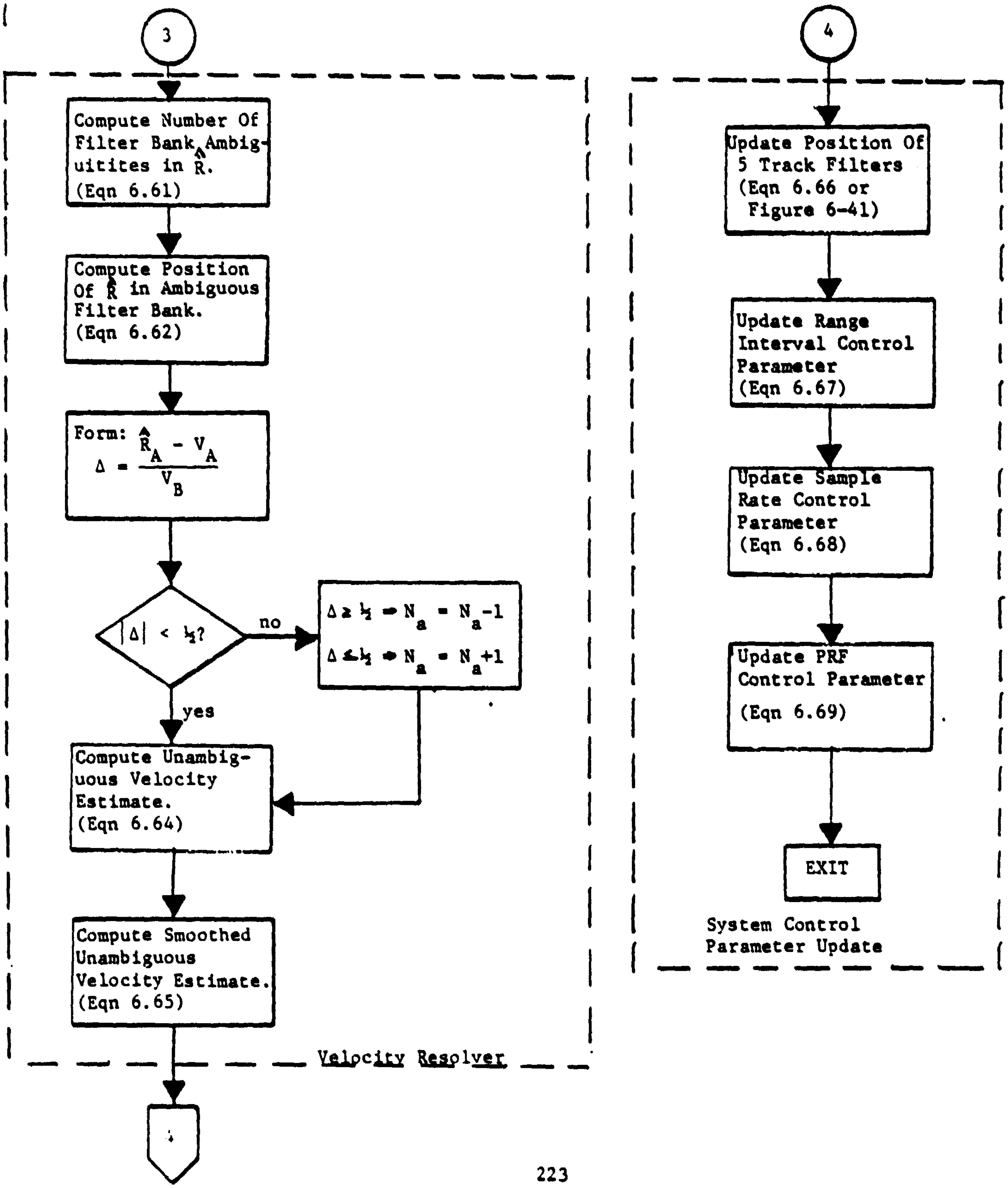

ENTER
Quantize Range Discriminant to 3/16 dB Accuracy
Update Range Rate Estimate (Eqn 6.56)
Update Range Estimate (Eqn 6.57)
Scale Range Estimate to Feet
Add Fixed Range Bias
Range Track Loop Filter
2
2
Quantize Velocity Discriminant To 3/16 dB Accuracy
Compute Integral Number of Filter Widths In Ambiguous Velocity (Eqn 6.58)
Compute Fractional Portion of Ambiguous Velocity. (Eqn 6.59)
Combine Integral And Fractional Parts To Form Ambiguous Velocity Est. Scaled to 1/128 of a Filter width.
3
Ambiguous Velocity Estimater

Figure 6-43 RANGE AND RANGE RATE TRACKING LOOP COMPUTER ALGORITHM (2 of 2)


3
Compute Number Of Filter Bank Ambig-uitites in $\hat{R}$. (Eqn 6.61)
Compute Position Of $\hat{R}$ in Ambiguous Filter Bank. (Eqn 6.62)
Form: $\Delta = \frac{\hat{R}_A - V_A}{V_B}$
$|\Delta| < ½?$
no
$\Delta \geq ½ \Rightarrow N_a = N_a - 1$
$\Delta \leq ½ \Rightarrow N_a = N_a + 1$
yes
Compute Unambig-uous Velocity Estimate. (Eqn 6.64)
Compute Smoothed Unambiguous Velocity Estimate. (Eqn 6.65)
Velocity Resolver
4
Update Position Of 5 Track Filters (Eqn 6.66 or Figure 6-41)
Update Range Interval Control Parameter (Eqn 6.67)
Update Sample Rate Control Parameter (Eqn 6.68)
Update PRF Control Parameter (Eqn 6.69)
EXIT
System Control Parameter Update

(6.56) $$\hat{\dot{R}}(n) = \hat{\dot{R}}(n-1) + m_B D_R(n)$$

(6.57) $$\hat{\dot{R}}(n) = \hat{R}(n-1) + \hat{\dot{R}}(n) + m_a D_R(n)$$

where it should be noted that $\hat{R}$ is scaled to 5/16 feet and $\hat{\dot{R}}$ is scaled to 5/(16 T_S) feet per second. The last step is to scale the range estimate to feet and add a fixed range bias to form the radar predicted range estimate.

Ambiguous Velocity Estimator. In the first step the velocity discriminant is quantized to 3/16 dB by replacing D_R by D_V in equation (6.55). Next the integral number of filter widths between zero frequency and the target location in the doppler filter bank is updated using the equation

(6.58) $$\begin{matrix}\text{Integral Number of}\\ \text{Filter Widths}\end{matrix} = \left[\begin{matrix} m_L, & \text{if } D_V \geq 0 \\ m_c, & \text{if } D_V < 0 \end{matrix}\right.$$

where m_L = filter number of low filter (see Figure 6-12),

m_c = filter number of center filter.

Then, the fractional filter width remainder is determined to 1/128 of a filter width accuracy in the following way:

(6.59) $$\begin{matrix}\text{Fractional Filter}\\ \text{Width Remainder}\end{matrix} = \left[\begin{matrix} F(D_V), & \text{if } D_V \geq 0 \\ 1-F(D_V), & \text{if } D_V < 0 \end{matrix}\right.$$

where the function F is shown in Figure 6-44. This function is predetermined using the expression

(6.60) $$D_V = \left[\frac{160}{3} \log \left(\frac{3\text{in } 16X_L \text{ Sin } X_H}{\sin X_L \text{ Sin } 16X_H}\right)\right]$$

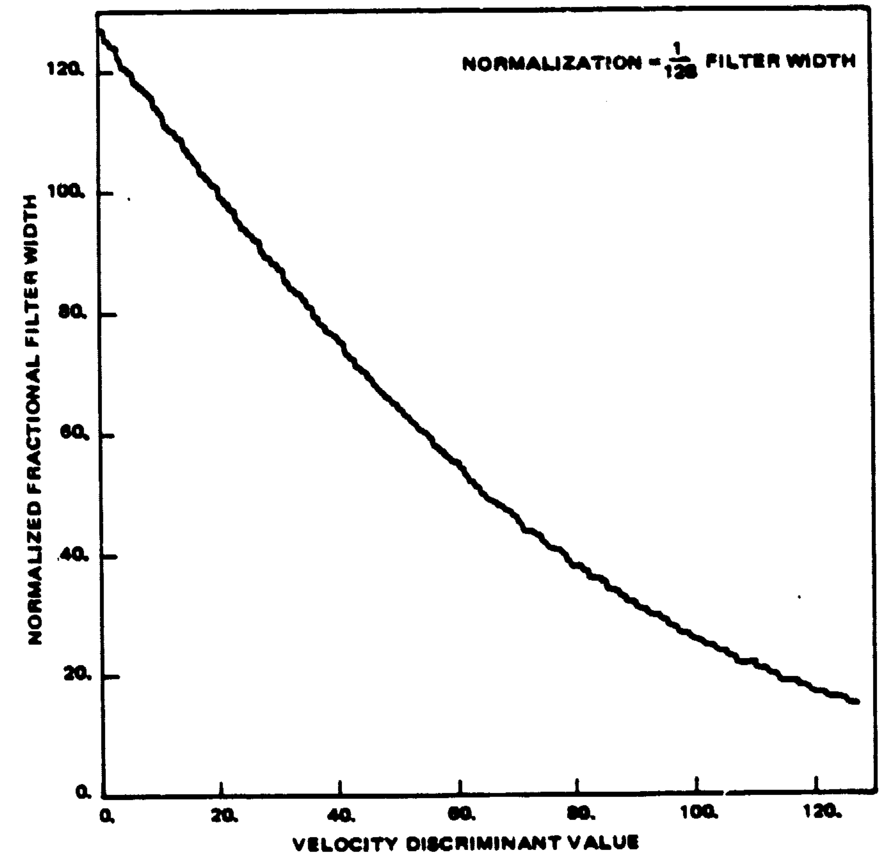


Figure 6-44. Fractional Filter Width as a Function Velocity Discriminant Value.

where $$X_L = \pi(-\frac{1}{32} - ft_p)$$

$$X_H = \pi(\frac{1}{32} - ft_p)$$

$$t_p = \text{PRI}$$

$$[\cdot] = \text{greatest integer in } \cdot,$$

to associate a D_V value with each of the following values of f:

$$(\frac{0}{256t_p}, \frac{1}{256t_p}, ---, \frac{127}{256t_p}).$$

These values are stored in the computer in look-up table fashion. The last step is to compute the ambiguous velocity estimate by adding the results of equations (6.58) and (6.59).

Velocity Resolver. The first task is to compute the number N_a of ambiguous doppler filter bank widths in $\hat{\dot{R}}(n)$. This is achieved using

(6.61) $$N_a = \left[\hat{\dot{R}}(n) / V_B\right]$$

where V_B is the maximum unambiguous velocity. Then, N_a is checked for accuracy using the following procedure: the position of $\hat{\dot{R}}(n)$ in the ambiguous filter bank, call it $\hat{\dot{R}}_a(n)$, is computed using the equation

(6.62) $$\hat{\dot{R}}_a(n) = \text{mod } (\hat{\dot{R}}(n), V_B)$$

and is compared to the ambiguous velocity V_a obtained from the first step. The ambiguity number is corrected, depending upon the result of this comparison, as follows

(6.63) $$N_a = \begin{cases} N_a + 1, & \text{if } \hat{\dot{R}}_a - V_a \leq -\frac{V_B}{2} \\ N_a, & \text{if } -\frac{V_B}{2} < (\hat{R}_a - V_a) < \frac{V_B}{2} \\ N_a - 1, & \text{if } \hat{R}_a - V_a \geq \frac{V_B}{2} \end{cases}.$$

Once the ambiguity number has been correctly determined, it is combined with the result from step one to obtain the unsmoothed, unambiguous velocity estimate, $V_u(n)$, i.e.

$$(6.64) \qquad V_u(n) = V_a(n) + N_a V_B \ .$$

The final step is to pass this value of $V_u(n)$ through a digital smoothing filter. This filter is a moving window average which averages the previous three V_u values with the present value. Quantitatively, we have

$$(6.65) \qquad V_s(n) = \sum_{i=n-3}^{n} V_u(i) \ .$$

<u>Internal Control Parameter Update.</u> Based on the new estimates of the range, the velocity discriminant and on-target discriminant the following internal controls are updated: (1) filter bank position, (2) the range interval parameter, MRNG, (3) the PRF parameter, MPRF and (4) the sample rate parameter, MSAM. The filter position update requires the on-target and velocity discriminant values and the following algorithm:

$$(6.66) \qquad m_c = \begin{cases} m_c-2 & \text{if } D_V > 0 \text{ and } D_{OT} < T, \\ m_c-1 & \text{if } D_V > 51 \text{ and } D_{OT} \geq T, \\ m_c+0 & \text{if } |D_V| \leq 51 \text{ and } D_{OT} \geq T, \\ m_c+1 & \text{if } D_V < -51 \text{ and } D_{OT} \geq T, \\ m_c+2 & \text{if } D_V < 0 \text{ and } D_{OT} < T \ . \end{cases}$$

The range interval parameter MRNG is determined by finding the integer i such that

$$R_{i-1} \leq \hat{R}(n) < R_i \tag{6.67}$$

where the R_i are listed in Table 6-4. MSAM is computed using the following algorithm

$$\text{MSAM} = \begin{cases} 1\ , & \text{if MRNG} \leq 9 \text{ and IMODE} = 1 \\ & \text{or MRNG} \leq 4 \text{ and IMODE} = 2 \\ 2\ , & \text{if MRNG} > 9 \text{ and IMODE} = 1 \\ & \text{or MRNG} > 4 \text{ and IMODE} = 2\ . \end{cases} \tag{6.68}$$

Finally, the PRF parameter, MPRF, is updated by

$$\text{MPRF} = \begin{cases} 1\ , & \text{if MRNG} \leq 9 \text{ and IMODE} = 1, \\ & \text{or MRNG} \leq 9 \text{ and IMODE} = 2, \\ 2\ , & \text{if MRNG} > 9 \text{ and IMODE} = 2, \\ 3, & \text{if MRNG} > 9 \text{ and IMODE} = 1\ . \end{cases} \tag{6.69}$$

The values for MRNG, MSAM and MPRF as a function of range interval and system mode are summarized in Table 6-4.

7. RECOMMENDATIONS FOR FURTHER STUDY AND DEVELOPMENT

7.1 SYSTEM ANALYSIS

The present computer simulation model is a very useful tool for evaluation of the Ku-Band Radar track mode design. As an example of a useful system analysis where the model can immediately be applied, consider the following problem. At the present, it is not clear that one should PDI over all five frequencies in the track mode. Instead, it has been conjectured that performance would be improved by selecting the largest return of the five frequencies, especially when the return signal is weak and the target scattering properties are sensitive to small changes in transmit frequency. In this case, the computer simulation model can easily be adapted to perform an analysis of this problem.

7.2 RADAR MODEL FIDELITY IMPROVEMENT

Some of the areas where the radar simulation model may be improved are

- reducing computation time,
- discriminant model accuracy,
- AGC model accuracy,
- search model fidelity.

Reducing computation time is always desirable, since it will provide room for improvement in the model accuracy. For example, a reduction in computation time would allow us to use a more accurate discriminant generation model (see Appendix C). An accurate AGC model will not consume an appreciable amount of computation time, but it will require a significant amount of time to develop, install, and test an accurate algorithm. Accurate AGC estimates would be useful in predicting radar performance when a target fades rapidly and providing accurate signal strength estimates under weak target (low SNR) conditions. Although the search model has enough fidelity to provide adequate crew training, significant improvements can be made in this area if desired.

7.3 TARGET MODEL FIDELITY IMPROVEMENT

If the target scattering measurements recommended in section 4.3.5 are performed, then it would be very useful to correlate these data with the predictions of the present target model and, if feasible, make the necessary adjustments in the present model.

REFERENCES AND BIBLIOGRAPHY

1. "Proposal for Ku-Band Integrated Radar and Communication Equipment for the Space Shuttle Orbiter Vehicle, " Vol. II, Hughes Aircraft Company, Ref. No. D7768/SCG 60176P, May 1976.

2. C. L. Weber, W. K. Alem, and M. K. Simon, "Study to Investigate and Evaluate Means of Optimizing the Radar Function for the Space Shuttle," Axiomatix Report No. R7710-5, October 28, 1977.

3. J. W. Wright, On the Statistical Modeling of Radar Targets, PhD dissertation, University of Illinois at Urbana, Champaign, Illinois (University Microfilms No. 73-17483).

4. J. H. Landreth, "Simulation of a Radar Tracking a Glinting Aircraft Target in a Multipath Environment," IEEE Int. Radar Conf. Recora, 1977.

5. D. D. Howard, "Radar Target Angular Scintillation in Tracking and Guidance Systems Based on Echo Signal Phase - Front Distortion," Proc of NEC, 15, pp. 840-849, 1959.

6. J. H. Dunn, and Howard, D. D., "Radar Target Amplitude, Angle and Doppler Scintillation from Analysis of the Echo Signal Propagating in Space," IEEE Trans on MTT, 16, pp. 715-728, September, 1968.

7. D. C. Cross, and Evans, J. E. "Target-Generated Range Errors," IEEE Int. Radar Conf. Record, April 21-23 1975.

8. R. H. Delano "A Theory of Target Glint or Angular Scintillation in Radar Tracking," Proc of IRE, 41, pp. 1778-1784, December, 1953.

9. D. D. Howard, "Predicting Target-Caused Errors in Radar Measurements," Microwave Systems News, pp. 88-95, February 1978.

10. J. C. Riles; "Baseline Definition of the Ku-Band Servo System," IDC HS-237-318, Hughes Aircraft Company, El Segundo, California, February 1977.

11. D. D. McCracken, A Guide to Fortran IV Programming, New York, John Wiley and Sons, Inc.,1965.

12. "Proposal for the Preparation of a Ku-Band Radar Performance Computer Model," Report No. FR-79-27-506, HAC Ref. No. E4567, January, 1979.

13. J. W. Crispin and K. M. Siegc , Methods of Radar Cross-Section Analysis, New York, Academic Press, 1968.

14. M. E. Bechtel and R. A. Ross, "Handbook of Scattering Coefficients," Cornell Aeronautical Laboratory Rept. No. UF 2457-E-1, November, 1967.

15. R. W. Moss, "SMM Radar Cross Section Analysis," Westinghouse Electric Corporation Memorandum No.78-082, 19 May 1978.

16. G. T. Ruck, et al., Radar Cross Section Handbook, New York, Plenum Press, 1970.

17. "Ku-Band Integrated Radar and Communication Equipment for the Space Shuttle Orbiter Vehicle Preliminary Design Review," Vol. I, Hughes Aircraft Company, Ref. No. HS237-1531-5, March 1978.

18. J. C. RIles, "Correction to IDC dated 23 February 1977," IDC HS-237-318A, Hughes Aircraft Company, El Segundo, California, March,1977.

19. H. G. Magnusson and D. K. Shingledecker, "Results of Computer Analysis of Ku-Band Radar Search Mode Dectection Probability," IDC 2705.00/109 Hughes Aircraft Company, El Segundo, California, January, 1979.

20. J. E. Hutchins, "Ku-Band Radar Angle Model," Memorandum No. EJ2-78-087, NASA JSC,Houston, Texas, May, 1978.

21. S. C. Iglehart, "Ku-Band Radar Angle and Angle Rate Measurement Accuracy," IDC 2705.00/024, Hughes Aircraft Company, El Segundo, California, January 1978.

22. S. Thaler, "Angle Measurement," IDC 2241.20/232, Hughes Aircraft Company, El Segundo, California, June, 1975.

23. S. Thaler, "Range Tracker Random and Transient Errors," IDC 2310/83, Hughes Aircraft Company, El Segundo, California, September, 1977.

24. E. Brookner, "Tracking, Prediction, and Smoothing - Part 1, "Boston IEEE,"Modern Radar Techniques, Components, and Systems," Lecture Series Notes, Fall, 1976.

25. S. Thaler, "Doppler Derived Range Rate," IDC 2310/82, Hughes Aircraft Company, El Segundo, California, September, 1977.

26. H. Goldstein, Classical Mcchanics, Addison-Wesley Publishing Company, Inc., Cambridge, Massachusetts, 1956.

27. A. Papoulis, Prnbability,Random Variables, and Stochastic Processes, McGraw-Hill Book Company, New York, 1965.

28. A. E. Bryson and Y. C. Ho, Applied Optimal Control, Blaisdell Publishing Company, Waltham, Massachusetts, 1969.

29. J. R. Coleman, H. C. Heath, and F. K. Oshino, "Radar Cross-Section Calculations," Northrop Corporation Report No. NOR-76-214, December, 1976 (ADB017220L).

APPENDIX A

DERIVATION OF ANGLE AND ANGLE RATE TRACKING LOOP MODEL INITIALIZATION

The purpose of this appendix is to derive the equations used to initialize (1) the target inertial LOS azimuth and elevation rates and (2) the α and β gimbal rates. Fundamental to both derivations is the following fact taken from [26]. Consider two reference frames A and B with a common origin. Suppose B is rotating uniformly with angular velocity $\vec{w}$ with respect to A. Then the velocity of a free point target as measured by an observer fixed in frame A is related to the velocity of an observer fixed in frame B by the equation

$$\vec{v}|_A = \vec{v}|_B + \vec{w} \times \vec{r} \tag{A.1}$$

where $\vec{v}|_A$ = velocity measured in the A frame,

$\vec{r}$ = position vector of the point target,

and all of the vectors in equation (A.1) are expressed in the same, but arbitrary, coordinate system centered at origin of the A (or B) frame.

An important assumption that is used in both derivations is that the target c.g. is assumed to be on the antenna boresight axis, or, equivalently, the negative z axis of the L- frame at the time of initialization. Thus, the position vector $\vec{r}_o^{\,L}$ has the form

$$\vec{r}_o^{\,L} = \begin{pmatrix} 0 \\ 0 \\ -|\vec{r}_o^{\,L}| \end{pmatrix}. \tag{A.2}$$

A.1 DERIVATION OF TARGET INERTIAL LOS AZIMUTH AND ELEVATION RATE′ INITIALIZATIONS

Using the assumption stated in the previous paragraph, we can define the target inertial LOS azimuth and elevation rates by the expressions

$$\text{(A.3)} \qquad \begin{array}{l} \text{Inertial LOS} \\ \text{Azimuth Rate} \end{array} \triangleq w_{Tx}^{L} = \frac{v_{oy}^{L}\big|_{I}}{|\vec{r}_{o}^{\,L}|}$$

$$\begin{array}{l} \text{Inertial LOS} \\ \text{Elevation Rate} \end{array} \triangleq w_{Ty}^{L} = -\frac{v_{ox}^{L}\big|_{I}}{|\vec{r}_{o}^{\,L}|}$$

where $\vec{v}_{o}^{\,L}\big|_{I}$ = velocity of target c.g. as measured in the inertial frame and expressed in LOS coordinates.

We can now begin the derivation. Given that the orbiter body has the inertial angular velocity $\vec{w}_{B}$, equation (A.1) can be written

$$\vec{v}_{o}^{\,L}\big|_{I} = \vec{v}_{o}^{\,L}\big|_{B} + \vec{w}_{B}^{\,L} \times \vec{r}_{o}^{\,L} .$$

Using the assumption given in equation (A.2), the x and y components of $\vec{v}_{o}^{\,L}\big|_{I}$ can be written

$$v_{ox}^{L}\big|_{I} = v_{ox}^{L}\big|_{B} - w_{By}^{L}\,|\vec{r}_{o}^{\,L}|$$

$$\text{(A.4)} \qquad v_{oy}^{L}\big|_{I} = v_{oy}^{L}\big|_{B} + w_{Bx}^{L}\,|\vec{r}_{o}^{\,L}| .$$

Dividing equations (A.4) by $|\vec{r}_{o}^{\,L}|$ and using the definitions of target inertial LOS rates given in equations (A.3), we obtain

$$\begin{array}{l} \text{Inertial LOS} \\ \text{Elevation Rate} \end{array} = w_{Ty}^{L} = -\frac{v_{ox}^{L}\big|_{B}}{|r_{o}^{L}|} + w_{By}^{L}$$

$$\text{(A.5)} \qquad \begin{array}{l} \text{Inertial LOS} \\ \text{Azimuth Rate} \end{array} = w_{Tx}^{L} = \frac{v_{oy}^{L}\big|_{B}}{|r_{o}^{L}|} + w_{Bx}^{L} .$$

A.2 DERIVATION OF α AND β GIMBAL RATE INITIALIZATIONS

The α gimbal rate is defined as the rate of rotation of the outer gimbal (or G) frame about the x-axis of the R frame. If we assume the rotation is uniform, then from equation (A.1) we have

$$\vec{v}_o^{\,G}\Big|_B = \vec{v}_o^{\,G}\Big|_G + \begin{pmatrix}\dot{\alpha}\\0\\0\end{pmatrix} X\,\vec{r}_o^{\,G}\,. \tag{A.6}$$

Noting that

$$\vec{r}_o^{\,G} = T_{GL}\,\vec{r}_o^{\,L} = \begin{pmatrix}\cos\beta & 0 & \sin\beta\\ 0 & 1 & 0\\ -\sin\beta & 0 & \cos\beta\end{pmatrix}\begin{pmatrix}0\\0\\-|\vec{r}_o|\end{pmatrix}$$

or

$$\vec{r}_o^{\,G} = \begin{pmatrix}-|\vec{r}_o^{\,L}|\sin\beta\\ 0\\ -|\vec{r}_o^{\,L}|\cos\beta\end{pmatrix}$$

and writing out the y-component of equation (A.6), we then have

$$v_{oy}^{G}\Big|_B = v_{oy}^{G}\Big|_G + \dot{\alpha}\,|\vec{r}_o^{\,L}|\cos\beta.$$

But the y-component of the target velocity as measured in the G-frame and expressed in the G-frame coordinates, i.e. $v_{oy}^{G}\Big|_G$, is zero. Therefore

$$\dot{\alpha} = \frac{v_{oy}^{G}\Big|_B}{|\vec{r}_o^{\,L}|\cos\beta} = \frac{v_{oy}^{L}\Big|_B}{|\vec{r}_o^{\,L}|\cos\beta} = \frac{w_{Tx}^{L} - w_{Bx}^{L}}{\cos\beta} \tag{A.7}$$

The β gimbal rate, $\dot{\beta}$, is defined as the rate of rotation of the inner gimbal (or L) frame about the outer gimbal (or G) frame y-axis. Using this fact and equation (A.1), we obtain

$$\vec{v}_o^{\,G}\Big|_G = \vec{v}_o^{\,G}\Big|_L + \begin{pmatrix}0\\ \dot{\beta}\\ 0\end{pmatrix} X\,\vec{r}_o^{\,G}$$

Noting that $\vec{v}_o^{\,G}\Big|_L = 0$ by assumption and substituting the resultant expression for $\vec{v}_o^{\,G}\big|_G$ into equation (A.6), gives

$$v_o^{\,G}\Big|_B = \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ 0 \end{pmatrix} \mathrm{X}\ \vec{r}_o^{\,G}$$

Transforming to L-frame coordinates,

$$\vec{v}_o^{\,L}\Big|_B = T_{LG} \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ 0 \end{pmatrix} \mathrm{X}\ T_{LG}\ \vec{r}_o^{\,G} \tag{A.8}$$

$$= \begin{pmatrix} -\dot{\beta}\,|\vec{r}_o| \\ \dot{\alpha}\,|\vec{r}_o| \cos\beta \\ 0 \end{pmatrix}$$

The expression for β can be obtained from the x-component of equation (A.8). It is

$$\dot{\beta} = \frac{-v_{ox}^{\,L}\big|_B}{r_o} = w_{Ty}^{\,L} - w_{By}^{\,L} \tag{A.9}$$

where equation (A.5) was used to obtain the last equality.

APPENDIX B

DERIVATION OF TARGET PITCH ANGLE, ROLL ANGLE, INERTIAL ROLL RATE, AND INERTIAL PITCH RATE TRANSFORMATIONS

This appendix presents the derivations of (1) the transformation of α and β, which are tracked by the radar, to roll and pitch angles in the Orbiter Body (B) frame and (2) the transformation of the target inertial LOS azimuth and elevation rates, which are estimated by the radar, to target inertial roll and pitch rates in the B frame.

B.1 DEFINITIONS AND ASSUMPTIONS

We first provide definitions of all quantities which are pertinent to the derivations given below. The α and β gimbal angles were defined in Section 2.1, while the roll and pitch angles are defined as follows:

- Target Roll Angle is the angle between the $-Z_B$ axis and the projection of the target direction vector on the Z_B-Y_B plane as shown in Figure B-1.
- Target Pitch Angle is the angle between the target direction vector and the projection of the target direction vector on the Z_B-Y_B plane.

Quantitatively, these definitions can be expressed as

$$\text{Roll angle} \triangleq -\tan^{-1}\left[\frac{\hat{r}_o \cdot \hat{Y}_B}{-\hat{r}_o \cdot \hat{Z}_B}\right] \tag{B.1}$$

$$\text{Pitch angle} \triangleq -\sin^{-1}\left[\hat{r}_o \cdot \hat{X}_B\right]$$

where $\hat{r}_o$ = unit vector in direction of the target,

$\hat{X}_B, \hat{Y}_B, \hat{Z}_B$ = unit vectors along the X_B, Y_B, Z_B axis of the B-frame, respectively.

The target inertial LOS azimuth and elevation rates were defined in

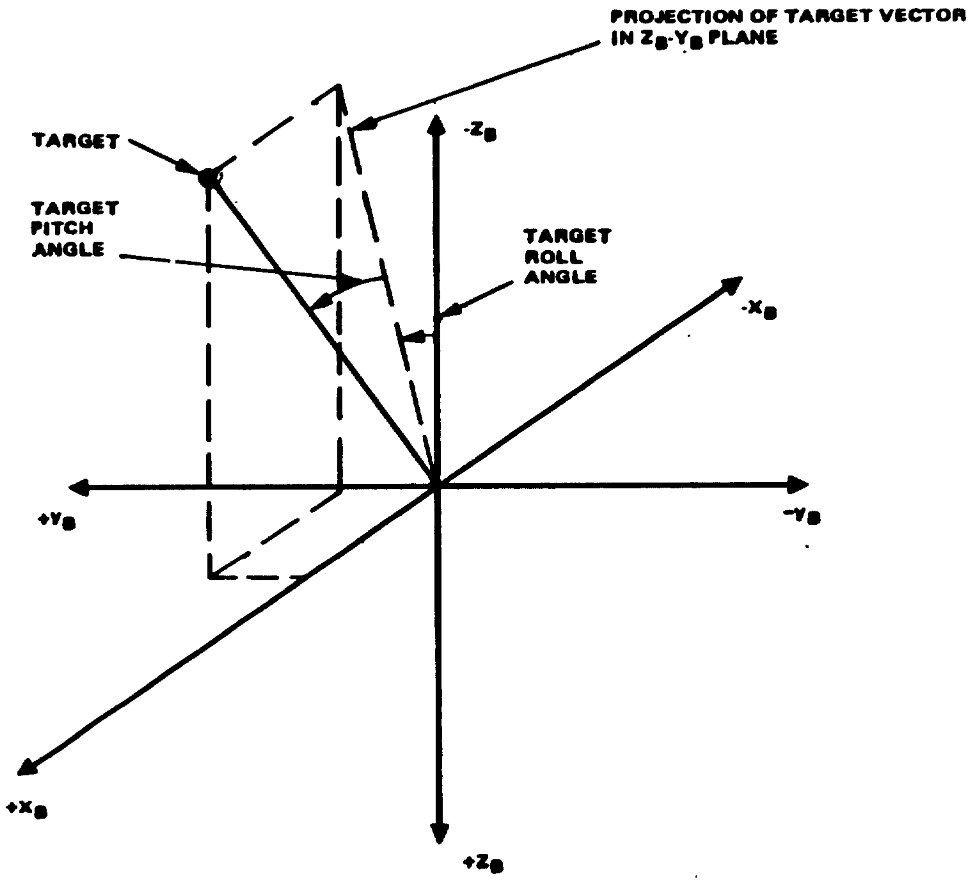


Figure B-1. Definition of Target Roll and Pitch Angles.

Appendix A equation (A.3). Inertial roll and pitch rate are defined as

- Inertial Roll Rate is the projection of the target inertial angular velocity (estimated by the radar) along the X_B-axis.
- Inertial Pitch Rate is the projection of the target inertial angular velocity along the Y_B-axis.

Again, mathematically we have

$$\text{(B.2)} \qquad \begin{aligned} &\text{Inertial Roll Rate} \triangleq -\vec{w}_T \cdot \hat{X}_B, \\ &\text{Inertial Pitch Rate} \triangleq -\vec{w}_T \cdot \hat{Y}_B \,. \end{aligned}$$

There are two basic assumptions that were made in the development of the required transformations. These are that

(1) the radar is located at the origin (or C.G.) of the B-frame, i.e. no offset,

(2) the 67° yaw angle between the B and R frames is assumed to be exact, i.e. no boom deployment error.

With these assumptions under our belt we can define one last, but very useful, quantity. The transformation matrix T_{BL}, which transforms a vector expressed in L coordinates to a vector expressed in B coordinates (see section 2), is defined by

$$\text{(B.3)} \qquad T_{BL} = \begin{pmatrix} c\gamma & -s\gamma & 0 \\ s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & c\alpha & -s\alpha \\ 0 & s\alpha & c\alpha \end{pmatrix} \begin{pmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{pmatrix}$$

where $c = \cos$,
$s = \sin$,
$\gamma = +67^{\circ}$.

B.2 DERIVATION OF TARGET ROLL AND PITCH ANGLE TRANSFORMATIONS

As mentioned in the introduction we are given the α , β angles and desire to convert these angles to roll and pitch. Consider the following argument. $\hat{r}_o$, the unit vector in the direction of the target, lies along the Z_L-axis in the LOS frames. This can be written

$$\hat{r}_o^{\,L} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

Tranforming this vector to the Body Frame coordinates, we have

$$T_{BL}\,\hat{r}_o^{\,L} = \begin{pmatrix} -T_{BL}\ (1,3) \\ -T_{BL}\ (2,3) \\ -T_{BL}\ (3,3) \end{pmatrix} \tag{B.4}$$

Using equation (B.4) and the definitions of roll and pitch angles given in equations (B.1), we obtain

$$\text{Roll angle} = -\tan^{-1}\left[-\frac{T_{BL}(2,3)}{T_{BL(3,3)}}\right] \tag{B.5}$$

$$\text{Pitch angle} = -\sin^{-1}\left[T_{BL}(1,3)\right].$$

B.3 <u>DERIVATION OF TARGET INERTIAL ROLL AND PITCH RATE TRANSFORMATIONS</u>

In this case, the radar estimates the components of the target inertial angular velocity vector in the LOS frame, and we would like this vector transformed to the B-frame coordinates. The argument begins by noting that in the LOS frame the Z_L- component is always zero. That is,

$$\vec{w}_T^{\,L} = \begin{pmatrix} \hat{\theta}_{AZ} \\ \hat{\theta}_{EL} \\ 0 \end{pmatrix} = \begin{pmatrix} w_{Lx}^{T} \\ w_{Ly}^{T} \\ 0 \end{pmatrix}.$$

Transforming this vector to body coordinates, we have

$$\vec{w}_T^{\,B} = \begin{pmatrix} T_{BL}\,(1,1)\,\hat{\theta}_{AZ} + T_{BL}\,(1,2)\,\hat{\dot{\theta}}_{EL} \\ T_{BL}\,(2,1)\,\hat{\theta}_{AZ} + T_{BL}\,(2,2)\,\hat{\dot{\theta}}_{EL} \\ T_{BL}\,(3,1)\,\hat{\theta}_{AZ} + T_{BL}\,(3,2)\,\hat{\dot{\theta}}_{EL} \end{pmatrix}$$

Using equation (B.6) and the definition of roll and pitch rate given in equations (B.2), target roll and pitch rates can be written as

$$\text{Target Roll Rate} = -\left[T_{BL}\,(1,1)\,\hat{\dot{\theta}}_{AZ} + T_{BL}\,(1,2)\,\hat{\dot{\theta}}_{EL}\right]$$

(B.7)

$$\text{Target Pitch Rate} = -\left[T_{BL}(2,1)\,\hat{\dot{\theta}}_{AZ} + T_{BL}(2,2)\,\hat{\dot{\theta}}_{EL}\right] .$$

APPENDIX C

DERIVATION OF THE NOISE-FREE DISCRIMINANT COMPONENTS COMPUTATION MODEL

This appendix provides a derivation of the noise-free discriminant component (see section 6.4.3 for a definition of discriminant component) computation model. A simplified diagram of the model, illustrated in Figure C-1, shows that each of the noise-free discriminant components is computed at the PDI output. Derivation of this model is structured as follows. First, the complete target response representing the i th frequency, the j th time slot, the l th range gate, and the n th doppler filter is computed at the magnitude-detector output and then the individual noise-free discriminant components are formed by summing (PDIing) the magnitude-squared detected response over the appropriate i, j, l, and n indices.

C.1 MODEL ASSUMPTIONS

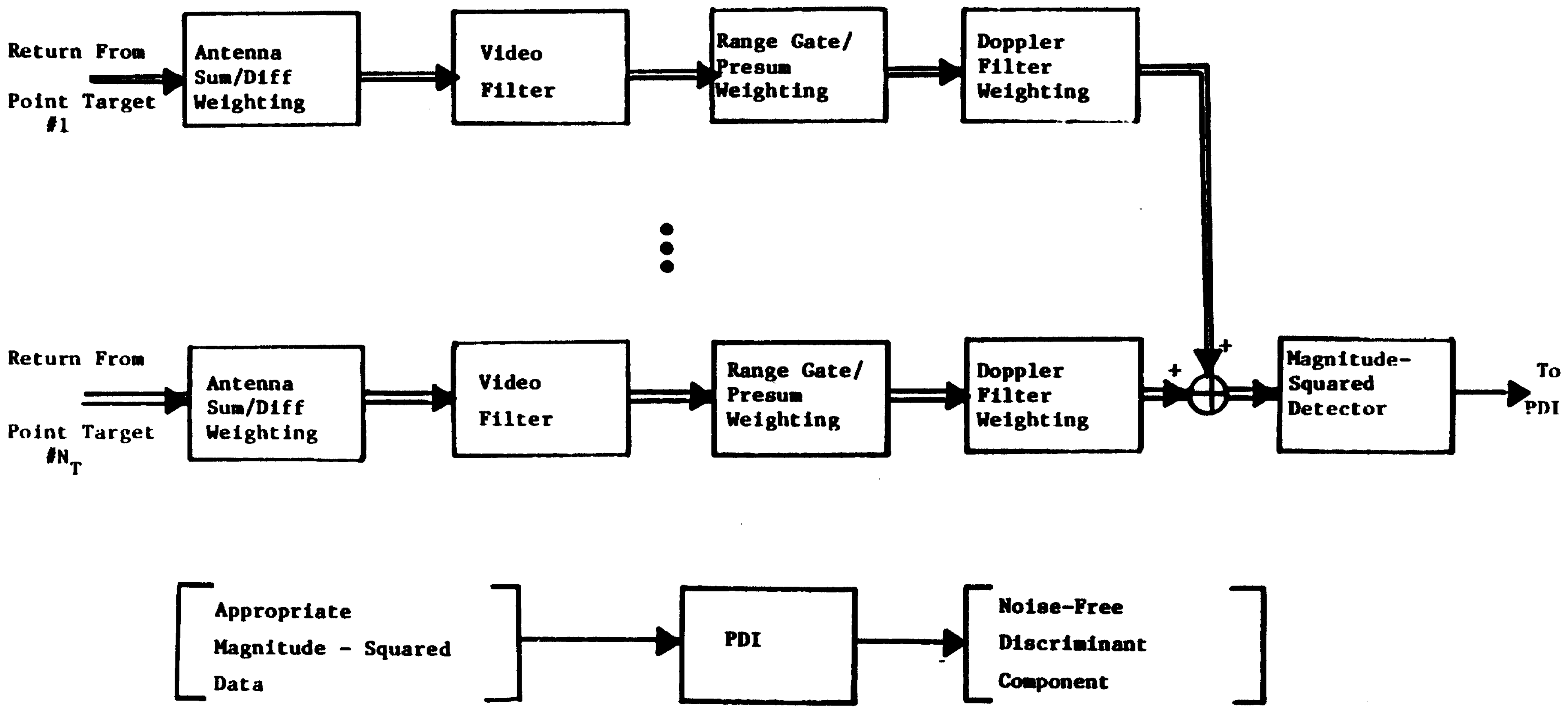
Assumptions used in the development of the computational model are listed in section 6.4.1. Rather than repeating the list here, use of each of the assumptions will be noted at the appropriate point in the derivation.

C.2 NOISE-FREE MAGNITUDE-SQUARED DETECTOR RESPONSE DERIVATION

The development of the magnitude-squared detector response is broken into several steps. These are (1) compute the doppler filter response for a single point scatterer, (2) using the assumed linearity of the processor from the antenna to the doppler filter, compute the complete target response by vectorial summation of the individual responses, and (3) compute the magnitude-square of the result. These steps are illustrated in Figure C-1 and described in detail below.

Doppler Filter Response For a Single Scatterer. We first write the expression for the k th target response at the baseband filter input. This response represents that portion of the received waveform associated with the entire j th time slot at the i th frequency (see Figures 6-2 and 6-3). The expression

Figure C-1 SIMPLIFIED DIAGRAM OF THE NOISE-FREE DISCRIMINANT COMPONENT COMPUTATION MODEL



for this response is obtained by starting with the point scatterer's single pulse return at the antenna output terminals given in equation (4.1) and applying assumptions (1) through (5) of section 6.4.1. This gives

$$\text{(C.1)} \qquad S_k(t) = A_k \sigma_k^{\frac{1}{2}} (\rho_{sk} + \rho_{dkj}) \sum_{m=o}^{15} \exp\left[j\,(2\pi f_k t - \phi_{ki})\right] \bullet P\left[\frac{t - mt_p - t_k}{t_t}\right]$$

where

A_k is defined in equation(4.1),

ρ_{sk} = antenna sum pattern weighting for k th scatterer,

ρ_{dkj} = antenna difference pattern weighting for k th scatterer and j th time slot,

σ_k = RCS for k th scatterer,

f_k = k th target doppler shift,

t_t = transmit pulse width,

t_p = PRI,

t_k = $2\,(R_k^L - R_G)\,/c$,

ϕ_{ki} = $4\pi(R_k^L - R_o^L)\,/\lambda_i$,

λ_i = wavelength associated with i th transmit frequency,

$$P(t) = \begin{cases} 1, 0 \leq t \leq 1 \\ 0, \text{ otherwise.} \end{cases}$$

The next step is to compute the response of the presummer to the m th pulse in the above expression. This computation includes several intermediate steps: baseband filtering, sampling, range gating, and, finally, presumming. Assumptions (6) through (8) are used in the filtering, sampling, and ranging gating process. The result of this process is best described by the illustration provided in Figure C-2, showing the sampled pulse response with respect to the early and

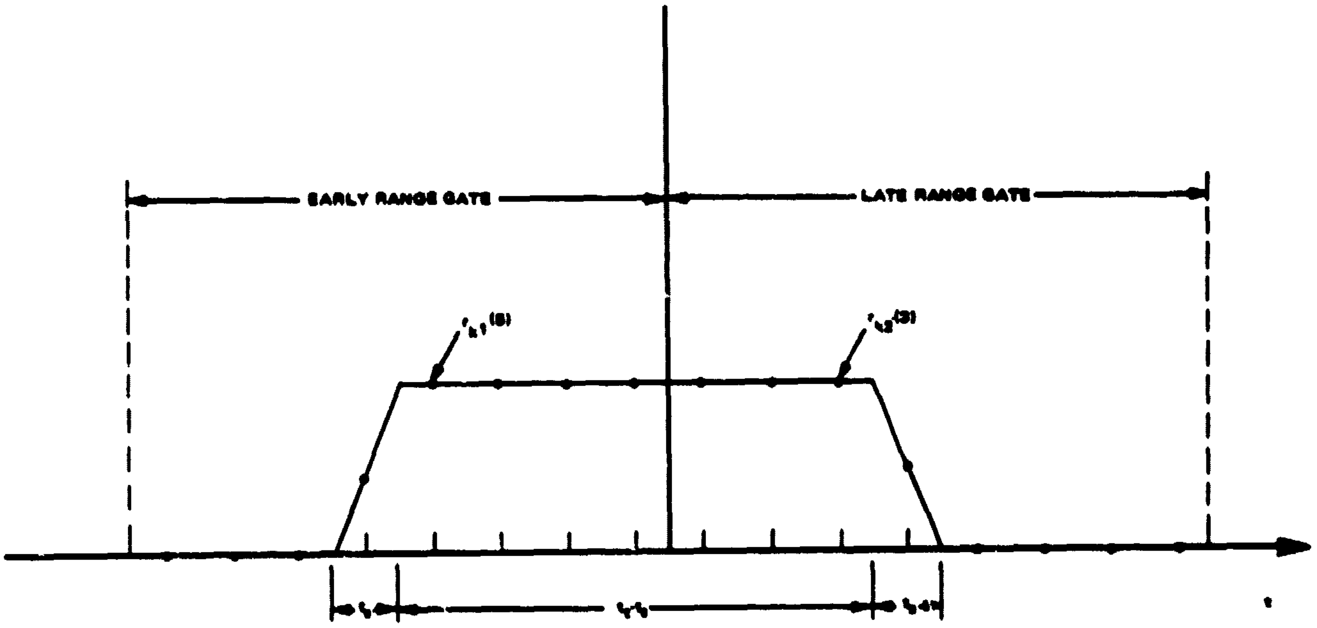


Figure C-2. Illustration of the Result of the Filtering, Sampling, and Range Gating Process late range gates. With these assumptions, the general response of the presummer for the i th frequency, the j th time slot, the l th range gate, and the m th pulse is

$$\text{(C.2)} \qquad S_k(i,j,l,m) = A_k \sigma_k^{\frac{1}{2}} (\rho_{sk} + \rho_{dkj}) \exp j(2\pi f_k\, m t_p + \phi_{ki})$$

$$\bullet \sum_{n=1}^{N_p} r_{k\ell}(n) \exp \left[j2\pi f_k \quad (n + (\ell-3/2)\, N_p - 1/2)\,\tau_s - t_k \right]$$

where $r_{kl}(n)$ is the magnitude of the n th sample in the l th range gate and depends upon the position of the filtered pulse in the range gate as illustrated in Figure C-2. Quantitative expression of each $r_{kl}(n)$ is delayed until the following approximation is stated:

> Approximation: It is assumed that the phase progression over a pulsewidth can be ignored.

Using this approximation, the summation in (C.2) simplifies to

$$\begin{bmatrix} \text{Summation} \\ \text{in C.2} \end{bmatrix} = \sum_{n=1}^{N_p} r_{kl}\,(n)$$

$$= N_p\, R_\ell(t_k)$$

and $R_\ell(t_k)$ is defined by

$$\text{(C.3)} \qquad R_1(t_k) = \begin{cases} 0, \text{ if } \Delta \leq -3 \text{ or } \Delta \geq 1 \\ \frac{3+\Delta}{2}, \text{ if } -3 \leq \Delta \leq -1 \\ \frac{1-\Delta}{2}, \text{ if } -1 \leq \Delta \leq 1 \end{cases} \qquad \text{(Early Gate)}$$

or

$$(C.4)\qquad R_2(t_k) = \begin{cases} \frac{1+\Delta}{2}, & \text{if } -1 \leq \Delta \leq 1 \\ \frac{3-\Delta}{2}, & \text{if } 1 \leq \Delta \leq 3 \\ 0, & \text{if } \Delta \geq 3 \text{ or } \Delta \leq -1 \end{cases} \qquad \text{(Late Gate)}$$

where $\Delta = t_k/t_t$. It is noted that the approximation given above is excellent for all short pulse modes. However, it may introduce some degradation in the long range case where large pulsewidths are used.

Calculation of the n th doppler filter response to the k th scatterer is easily accomplished by using equation (C.2) and (C.3)(or (C.4)) and forming the summation

$$(C.5)\qquad S_k(i,j,l,n) = \sum_{m=o}^{15} S_k(i,j,l,m) \exp(-j \frac{2\pi mn}{32}).$$

Performing the summation, we obtain

$$S_k(i,j,l,n) = C_k(i,j,l) \frac{\sin(16z_k)}{\sin(z_k)} \exp(-j15z_k)$$

where $$C_k(i,j,l) = A_k\sigma_k^{1/2}(\rho_{sk} + \rho_{dkj}) N_p R_\ell(t_k) \exp(-j2\pi\phi_{ki}),$$

$$z_k = \pi(\frac{n}{32} - f_k t_p).$$

Magnitude-Squared Detector Response. The magnitude-squared detector response is obtained by vectorially summing the doppler filter responses of all N_T scatterers using the assumed linearity of the processor, and then squaring the magnitude of the resultant sum. The result of these steps is the expression

(C.7) $$S(i,j,l,n) = \left| \sum_{k=1}^{N_T} S_k(i,j,l,n) \right|^2 .$$

C.3 DISCRIMINANT COMPONENT COMPUTATION

This subsection derives the closed-form expression used to model each of the three discriminant types: angle, range, and velocity.

C.3.1 Angle Discriminant Component Computation

The angle discriminant component corresponding to the j th time slot is obtained by performing a post-detection summation of the energy from the center doppler filter $(n = m_c)$ over N_F frequencies and both range gates. This gives the expression

(C.8) $$D_{Aj} = \sum_{i=1}^{N_F} \sum_{\ell=1}^{2} S(i,j,\ell,m_c) .$$

Practical Aspects of Computer Implementation. In order to reduce the amount of computation by a factor of two, the following approximation was used.

> Approximation: For a given pulse from a single target, the early and late gate presum weights are equal and are given by $\frac{1}{2}\left[R_1(t_k) + R_2(t_k)\right]$. However, the phase associated with the true position in the range gate is retained.

This approximation has the effect of altering the form of $C_k(i,j,\ell)$ used in equation (C.8). These coefficients now have the form

(C.9) $$C_k(i,j,\ell) \doteq A_k \sigma_k^{\frac{1}{2}} (\rho_{sk} + \rho_{dkj}) N_p \left[R_1(t_k) + R_2(t_k)\right]/2 \left[\cdot \exp(-j2\pi\phi_{ki})\right] .$$

We note that the approximation becomes exact when the target is composed of a single point scatterer. However, for a multiple point target, this approximation may be invalid, especially if the range tracker does not keep the return pulses close to the center of the range gates.

As mentioned above the original motivation for this approximation was to insure adequate computation speed. If it turns out that there is room for additional computation after the target has been represented adequately, then this approximation will be abandoned.

C.3.2 Range Discriminant Component Computation

The range discriminant component corresponding to the l th range gate is obtained by performing a post-detection summation of the energy from the center doppler filter over N_F frequencies and four time slots. The expression for the range gate discriminant component is

$$D_{R_\ell} = \sum_{i=1}^{N_F} \sum_{j=1}^{4} S\,(i,j,\ell,m_c). \tag{C.10}$$

Practical Aspects of Computer Implementation As in the angle discriminant case, we desire to speed the computation by making approximations in D_{R_ℓ}. In this case, we make the following approximation

Approximation: ρ_{dkj} are identically zero for all k and all j.

In effect, this approximation makes the assumption that the angle tracker is working perfectly. The result of the approximation is to alter the C_k's as follows

$$C_k(i,j,\ell) = A_k \sigma_k^{\frac{1}{2}} \rho_{sk} N_p R_\ell(t_k) \exp\left[-j2\pi\phi_{ki}\right]. \tag{C.11}$$

C.3.3 Velocity Discriminant Component Computation

We note that the velocity discriminant components and the on-target discriminant components are computed in an identical manner. Therefore, only the velocity discriminant component computation is described. The velocity discriminant component corresponding to the m_L (or m_H) filter is obtained by performing a post-detection summation of all energy from the m_L (or m_H) filter. This can be exoressed as

$$\text{(C.12)} \qquad D_{VL} = \sum_{i=1}^{N_F} \sum_{j=1}^{4} \sum_{\ell=1}^{2} S\,(i,\, j, \ell,\, m_L)\ .$$

Practical Aspects of Computer Implementation. To enhance the computer speed in this case we use both approximations stated above for the angle discriminant and range discriminant. Therefore, the $C_k\ (i,\ j, \ell)$ for equation (C.12) are given by

$$\text{(C.13)} \qquad C_k(i,\ j, \ell) = A_k \sigma_k^{\frac{1}{2}} \rho_{sk} N_p \left[R_1(t_k) + R_2(t_k) \right] / 2 \left[\exp(-j 2\pi\, \phi_{ki}) \right] .$$

APPENDIX D

DERIVATION OF THE THERMAL NOISE MODEL

As described in section 6.4, the computational model for the noisy discriminant values generates the noise-free target response at the PDI output and adds the equivalent thermal noise sample, obtained from the appropriate statistics, to the noise-free value. This model is illustrated in Figure D-1. Motivation for injecting the noise at this point, rather than at the signal processor input or some intermediate point was to enhance the real-time processing capability of the track mode. That is, it was desired to maximize the number of point scatterers allowed in the target model. The purpose of this appendix is to demonstrate that the equivalent noise can be represented as an additive noise process and to derive the statistical characteristics, i.e. the mean, the variance, and the probability density function (pdf) for each member of this random sequence.

D.1 MODEL ASSUMPTIONS

Derivation of the noise model is based upon the following set of assumptions. The primary assumption is that the form of the signal, including thermal noise, at the doppler filter output is given by the expression

$$\text{(D.1)} \qquad v(n) = v_I(n) + v_q(n) = (S_I(n) + n_I(n)) + j\ (S_q(n) + n_q(n)).$$

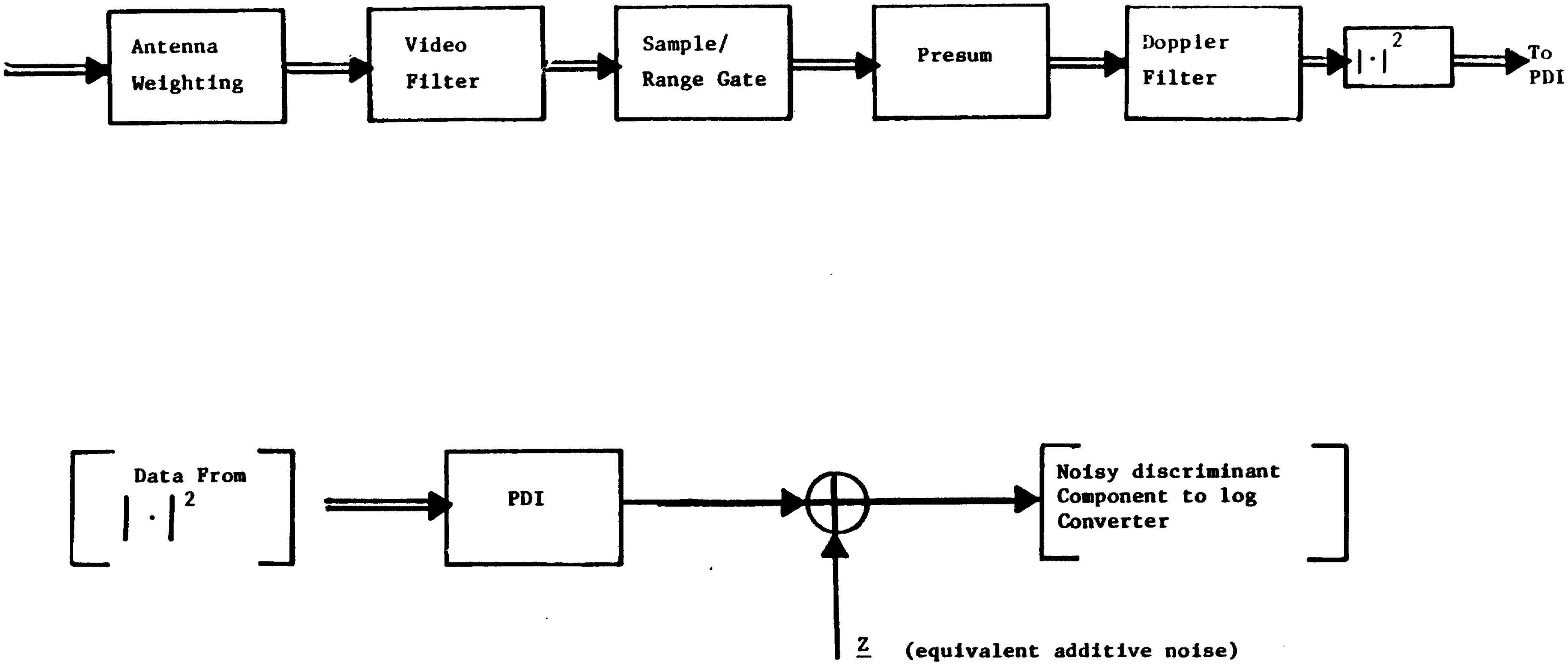
$S_I(n)$ and $S_q(n)$ are the in-phase and quadrature components of the noise-free target response at the doppler filter output for the n th time sample. The quantities $n_I(n)$ and $n_q(n)$ are the in-phase and quadrature components of the thermal noise process for the n th time sample. These components are assumed to have the following statistical characteristics:

(1) both are Gaussian random sequences,

(2) n_I, n_q are statistically independent for all values of n,

(3) $n_I(i)$, $n_I(j)$ (and $n_q(i)$, $n_q(j)$ are statistically independent

Figure D-1 ILLUSTRATION OF MODEL WHICH GENERATES NOISY DISCRIMINANT COMPONENT



for all values of i, j such that $i \neq j$.

(4) the mean and variance of n_I, n_q are

$$m_I = m_q = o$$

$$\sigma_I^2 = \sigma_q^2 = \sigma_o^2.$$

The last assumption is that all signal processor quantization effects are ignored.

D.2 NOISE MODEL DERIVATION

D.2.1 Derivation of Mean and Variance at PDI Output

We begin the derivation by calculating the output of the magnitude-squared detector when the sequence of equation (D.1) is applied at the input. The resulting output is given by the expression,

$$X(n) = |v(n)|^2 = v_I^2(n) + v_q^2(n)$$

$$= (S_I^2 + 2S_I n_I + n_I^2) + (S_q^2 + S_q n_q + n_q^2)$$

$$= (S_I^2 + S_q^2) + (2S_I n_I + 2S_q n_q + n_I^2 + n_q^2)$$

Computing the mean of X(n), we have

$$\overline{X}(n) = \overline{|v(n)|^2} = \overline{S_I^2} + \overline{S_q^2} + \overline{2S_I n_I} + \overline{2S_q n_q} + \overline{n_I^2} + \overline{n_q^2}$$

where the bar over a quantity means to compute the expected value of the quantity. Using the assumptions given in section D.1, this expression reduces to

$$\overline{X}(n) = S_I^2 + S_q^2 + 2\sigma_o^2 = |S|^2 + 2\sigma_o^2 \qquad \text{(D.2)}$$

Calculation of the variance of X(n) is straight-forward, but quite tedious, to perform. Therefore we will only provide the result of that computation:

(D.3) $$\text{var } X(n) = 4\sigma_o^2 \; |S(n)|^2 + 4\sigma_o^4 \quad .$$

The next step is to calculate the PDI output signal and its associated mean and variance. Assuming the PDI ratio is N, the output signal has the form

(D.4) $$y(n) = \sum_{n=1}^{N} X(n) \; .$$

The mean of y(n) is computed from

$$\overline{y(n)} = \sum_{n=1}^{N} \overline{X}(n) = \sum_{n=1}^{N} |S(n)|^2 + 2N\sigma_o^2$$

or, defining $\overline{|S(n)|^2} = \frac{1}{N} \sum_{n=1}^{N} |S(n)|^2$, we have

(D.5) $$\overline{y(n)} = N\overline{|S(n)|}^2 + 2N\sigma_o^2 \; .$$

Calculation of the variance of y(n) is based upon the following fact which is stated without proof. (The proof is straight-forward, but quite messy.) Since it was assumed that $n_I(i)$, $n_I(j)$ (and $n_q(i)$, $n_q(j)$) are statistically independent for all i, j such that i ≠ j, it can be shown that (x(i), x(j) are uncorrelated (and satistically independent) . Using this fact, and the well-known relation,

$$\text{var } (x + y) = \text{var } x + \text{var } y$$

where x and y are uncorrelated, one can easily write the expression for the variance of y(n) as

$$\text{(D.6)} \qquad \operatorname{var} y = \sum_{n=1}^{N} \operatorname{var} X(n)$$

$$= 4N\sigma_o^2 \overline{|S|^2} + 4N\sigma_o^4$$

We can define the new random variable

$$\text{(D.7)} \qquad Z = y - N\overline{|S|^2}$$

which has the mean and variance

$$\text{(D.8)} \qquad \overline{Z} = \overline{y} - N \overline{|S|^2}$$

$$\operatorname{var}Z = \operatorname{var}y \quad .$$

Thus, from equation (D.7), it is seen that the output of the PDI can be expressed as the sum of the noise-free target response $(N\overline{|S|^2})$ and a sample from the random variable z which has the mean and variance given in equation (D.8) and the pdf, P_z, which is derived in the next subsection.

Ľ.2.2 Derivation of the PDF for Z

The pdf for the random variable Z can be derived as follows. Define the random variable

$$\text{(D.9)} \qquad w = \frac{1}{\sqrt{N}}\left[Z - 2N\sigma_o^2\right]$$

$$= \frac{1}{\sqrt{N}} \sum_{n=1}^{N} \left[x(n) - |S(n)|^2 - 2\sigma_o^2\right]$$

or

$$w = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} w_n$$

where the w_n are continuous, mean zero, and statistically independent. When N is reasonably large (we note that $N \geq 10$ for passive tracking modes), the pdf for w approaches a normal distribution of the form

$$(D.10) \qquad P_{\underline{w}}(w) = \frac{1}{\sqrt{2\pi}\,\sigma_w} e^{-(\frac{w^2}{2\sigma_w^2})}$$

where $\sigma_w^2 = \frac{1}{N}\sum_{n=1}^{N} \sigma_{w_n}^2$ from the central limit theorem [2] . Now, from equation (D.9), we have that

$$Z = \sqrt{N}\, w + 2N\sigma_o^2$$

and thus the pdf for Z is normal with mean and variance

$$(D.11) \qquad \overline{Z} = \sqrt{N}\,\overline{w} + \overline{2N\sigma_o^2} = 2N\sigma_o^2$$

$$\text{var } Z = N \text{ var } w = \text{var } y$$

as shown in section D.2.1.

D.3 PRACTICAL ASPECTS OF MODEL IMPLEMENTATION

Given the value of $\overline{|S^2|}$, the PDI output can be generated using the model of the previous section as follows:

$$(D.12) \qquad y = N\overline{|S^2|} + 2N\sigma_o^2 + 2\sqrt{N}\sigma_o^2\left[\overline{|S|^2}/\sigma_o^2 + 1\right] N(0,1)$$

where N(0,1) denotes a sample from a normally distributed population with zero mean and unit variance. It is important to point out that, although the probability is very small, in some instances the resulting value of y obtained from equation (D.12) can be negative. This result is totally unaccpetable since it does not in practice and it will cause the log conversion process to become undefined.

Therefore, to prevent this situation, we make our final approximation: we simply set y equal to the absolute value of the quantity on the right hand side of equation (D.12).

It is also noted that since the discriminant formation process computes the ratio of the noisy discriminant components, any scale factors that are common to both components can be ignored. We chose to ignore the factor, $2\sqrt{N}\,\sigma_o^2$. Combining this fact with the absolute value approximation explained above. equation (D.12) becomes

$$\text{(D.13)} \qquad y = \left| \sqrt{N} \; \frac{\overline{|S|^2}}{2\sigma_o^2} + \sqrt{N} + \left[\frac{2\,\overline{|S|^2}}{2\,\sigma_o^2} + 1 \right]^{\frac{1}{2}} N(0,1) \right| .$$

APPENDIX E

CROSS SECTION CALCULATION NOTES

Features 1-3

$$\sigma_{\perp} = 291\ aL^2 = 2.6m^2$$

Features 4-6

$$\sigma_{\perp} = 291\ aL^2 = 61m^2$$

Feature 7

$$\sigma_{\perp} = 291\ aL^2 = 25.7m^2$$

Features 8-10

$$\sigma_{\perp} = \frac{4\pi A}{\lambda^2} = 26934\ A$$

$$A_e = 0.2m^2$$

$$\sigma_{\perp} = 5000m^2 \text{ - Limit to } 1000m^2$$

Features 11-12

$$A = (.24m)^2 x \pi =$$

$$\sigma_{\perp} = 5000m^2 \text{ - Limit to } 1000m^2$$

Features 13-26

Take $D = .5m$

$f/D = .5$

Mainlobe (13)

$$\sigma_{\perp} = 6645D^2 = 3322\ m^2$$

Reflector (20)

$$\sigma = .785D^2 \cos^4 \theta = .2 \cos^4 \theta$$

Take $\cos \theta = 1$.

Feature 14

$a = .24$

$L = 4\text{m}$

$\sigma_{\perp} = 291\ aL^2 = 1117\text{m}^2$

Features 15-25

$\sigma = 4\pi\Lambda/\lambda^2$ (See text)

Features 27-32

$a = .1\text{m}$

$\sigma = \pi a^2 = .03\ \text{m}^2$

Features 33,34

$a_c = .315\text{m}$

$\theta_o = 23.6^\circ$

$\sigma(\theta < \theta_o) = 4\pi a_c^{\ 2} = 1.25\text{m}^2$

$$\sigma(\theta > \theta_o) \approx \frac{5\pi\ a_c^{\ 2}}{9} = 0.17\text{m}^2$$

APPENDIX F

A MODEL FOR CENTROID WANDER IN

ROUGH SURFACE MODELS

The areas modeled as rough surfaces can be expected to experience wander of the apparent center. We take the mean position to be x_k, y_k, z_k in Table 4-1 and add a vector $\underline{v}$ that reflects the wander. Consider features 35, perpendicular to the x-axis, .7 x .7 m in extent. Then we take $\bar{v}$ in the yz plane with y and z components to be random variables with

$$\bar{y} = \bar{z} = o$$

$$E\ yz = o$$

$$\sigma_y^2 = \sigma_z^2 = \frac{D^2}{12\ N_f}$$

$$D = \text{Area dimension} = .7m$$

$$N_f = \text{No. of frequencies averaged}$$

$$= 5$$

The vector changes as the target aspect changes. We model this behavior as follows. Let $\underline{v}_m$ be the wander vector at the m th simulation update. Take $\bar{v}_m$ to be a first order Markov process (Ref. 28 p. 324) with uncorrelateo components and with, for example, the z component given by

$$z_{m+1} = z_m + w_m$$

where w_m is zero mean, uniform, of variance

$$\sigma_w^2 = (1-\alpha^2)\ \sigma_z^2 \ .$$

We now choose α to match the correlation time of the model to that of the target. One has

$$\rho_k = \frac{E z_{m+k} z_m}{E z_m^2} \alpha^k$$

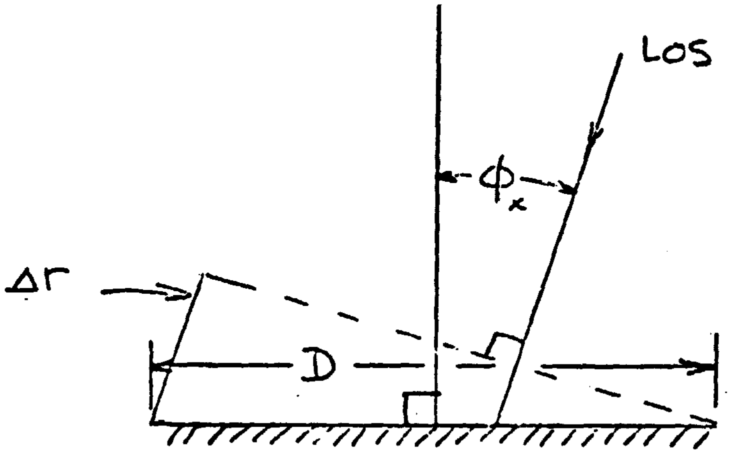
and taking the "correlation interval" as

$$N_c = \frac{-1}{\operatorname{Ln} \alpha}$$

$$\alpha = \exp(-1/N_c)$$

This is the number of iterations for which the correlation ρ_N falls to 1/e.

Now consider the target



The target return becomes decorrelated every time its aspect changes enough to cause another 2π propagation phase change from one edge of the area to the other. This corresponds to half wave.ength differential range:

$$\text{Change in } \Delta r = \text{Change in D} \sin \phi_x \;\; = -\frac{\lambda}{2}$$

Let $\emptyset_x$ change by $\Delta\emptyset_x$. Then

$$\Delta(\Delta r) \doteq \Delta\emptyset \frac{d}{d\emptyset} (\Delta R)$$

$$= \Delta\emptyset \;\; D \cos \phi_x = \frac{\lambda}{2}$$

so that the angle change corresponding to decorrelation is

$$\Delta\emptyset_x = \frac{\lambda}{2D \cos \emptyset_x}$$

Let the change in one simulation update period be $\delta\phi_x$; then the number of update cycles required for decorrelation is

$$N_u = \frac{\Delta\phi_x}{\delta\phi_x} = \frac{\lambda}{2D\delta\phi_x \cos\phi_x}$$

Matching N_u to N_c then yields

$$\alpha = \exp\left[\frac{-2D\ \delta\phi_x\ \cos\ \phi\ _x}{\lambda}\right] .$$

APPENDIX G

LISTING OF SIMULATION MODEL COMPUTER CODE

```
C  *************************************************
C  *  EXECUTIVE PROGRAM: INTERFACE WITH PARENT SIMULATION *
C  *************************************************
C
C
      SUBROUTINE EXEC
      COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)
      COMMON /OUTPUT/MSWF,MTF,MSF,DUM(7),IDUM2(4)
      COMMON /ICNTL/IOLDPW,IOLDMD,IOLDSM,ISHOLD,KMSCLK,KWMUP,IDUM1(3),
     2              MTP,IDUM5(17)
      DATA DATINT/1.0/
      KWMUP=1
C
C  *******************************************
C  * STEP 0: INITIALIZE ALL TARGET AND SYSTEM DATA *
C  *******************************************
      IF(DATINT.NE.1.0) GO TO 1
      CALL DATA
      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  **************************************************************
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
        RETURN
        FND
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=0
        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=C
        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
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 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(MZO.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 I-1: UPDATE ROLL/PITCH REFERENCE ANGLES.
   20  PREF=EDPA
       RREF=EDRA
C
C  STEP I-2: UPDATE POSITION OF GIMBALS.
       CALL POINT
C
C  STEP I-3: DETERMINE WHETHER BORESIGHT IN ZONE I AND/OR ZONE O AND
C            TAKE APPROPRIATE ACTIN.
       CALL ZONECK
C  IF BORESIGHT NOT IN ZONE O, THEN TARGET DETECTION NOT ALLOWED.
       IF(MZO.EQ.0) P TURN
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
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
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
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).
      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 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
C
C  STEP 5-2: IF DETECTION SUCCESSFUL --- SET TARGET PRESENT FLAG
C            HIGH AND INITIALIZE ACQUISITION CLOCK.
   35 MTP=1
      KACCLK=0
      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
C     10   MSAM=2
C
C     DETERMINE PRF
      MPRF=1
      IF(MRNG.GE.RI(6)) MPRF=2
      GO TO 20
C
C     CONTROL SETTINGS FOR AUTO/MANUAL MODES.
C
C     SET RANGE INTERVAL.
C     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(IASN.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 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 FUNCTION COMPUTES THE "NOMINAL" SNR AT THE VIDEO OUTPUT *
C     * --- IT ASSUMES NO BEAMSHAPE OR SCAN LOSS.                      *
C     ******************************************************************
C
C
      FUNCTION SNRV(SIGMA,RANGE)
      COMMON /CNTL/IPWR,IMODE,ITXP,IDUMC(6),DUMC(3)
      COMMON /ICNTL/IDUM(12),MSS,MTKINT,MRNG,MSAM,MPRF,IDUM2(10)
      COMMON /SYSDAT/DUM(12),TGTSIG,GPS,GAS
      DIMENSION PT(3),BN(2)
      DATA PT/47.,23.,7./, BN/69.5,57.2/
C
C     *********************************************
C     * DETERMINE WHETHER ACTIVE OR PASSIVE MODE *
C     *********************************************
      IF(IMODE.EQ.1) GO TO 10
C
C     **************************************
C     * PASSIVE MODE VIDEO SNR CALCULATION *
C     **************************************
      SNRV=GPS+PT(ITXP)+10.*ALOG10(SIGMA)-BN(MSAM)-40.*ALOG10(RANGE)
      SNRV=10.**(0.1*SNRV)
      RETURN
C
C     *************************************
C     * ACTIVE MODE VIDEO SNR CALCULATION *
C     *************************************
   10 SNRV=GAS-20.*ALOG10(RANGE)
      SNRV=10.**(0.1*SNRV)
      RETURN
      END
C
C
C     ****************************************************************
C     * THIS SUBROUTINE UPDATES THE POSITION OF THE ANTENNA GIMBALS *
C     ****************************************************************
C
C
      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 UBSCURATION 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 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
      BETA=B
      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 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
      MZ1=1
      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)
      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).
        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 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.EORNG.
     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: DETERMINE IF A BREAK TRACK CONDITION HAS OCCURRED.
      CALL BRKTRK
C
C    CHECK STATUS OF BREAK-TRACK FLAG (MBKTRK=1 --- BREAK-TRACK).
      IF(MBKTRK.NE.1) GO TO 7
C
C    IF BREAK-TRACK HAS OCCURRED --- RESET THE SYSTEM AND RETURN TO
C    SEARCH.
      CALL SYSINT
      RETURN
C
C  STEP 2-6: UPDATE ANTENNA GIMBAL POSITIONS AND RATES AND TARGET
C            ANGLES AND ANGLE RATES FOR DISPLAY (GPC-ACQ AND AUTO
C            MODES ONLY.)
    7 IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 10
C
C    FOR GPC-ACQ OR AUTO USE RADAR ESTIMATED TARGET ANGLES FOR
C    TRACK SERVO INPUT.
      CALL ATRACK
      GO TO 15
   10 IF(IASM.EQ.4) GO TO 12
C
C    FOR GPC-DES MODE USE GPC-SUPPLIED ANGLE DESIGNATES FOR TRACK SERVO
C    INPUT.
      PREF=EDPA
      RREF=EDRA
      CALL POINT
      GO TO 15
C
C    FOR MANUAL MODE USE CREW-SUPPLIED SLEW RATES TO DETERMINE TRACK
C    SERVO INPUT.
   12 PREF=PREF+FLOAT(IELS)*SLWRTE(ISLR+1)*TSAM
      RREF=RREF+FLOAT(IAZS)*SLWRTE(ISLR+1)*TSAM
      CALL POINT
C
C  STEP 2-7: UPDATE THE RANGE AND RANGE RATE ESTIMATES.
   15 CALL RTRACK
C
C  STEP 2-8: DETERMINE RADAR SIGNAL STRENGTH (FOR DISPLAY METER)
      CALL RSS
   20 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
      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)
      COMMON /SYSDAT/TSAM.DR(3).CP.SP.PSI.PSBIAS.DUM2(7)
      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/AGC.AGCOLD
      DIMENSION TRB(3.3).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     COMPUTE TRANSFORMATION MATRIX (ROTATES FROM BODY TO RADAR.)

        CALL PHI(TRB.PSI+PSBIAS)
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 BUDY 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)
C
C   STEP 2-3: INITIALIZE THE RANGE RATE ESTIMATE REGISTER.
      IRDOT= INTT(CGVEL*TSAM*3.2)
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   *******************************************
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)+32000
      MDF(3)=MOD(IVR.32)
      DO 20 I=1.5
      MD=MDF(3)+I-3+32000
20    MDF(I)=MOD(MD.32)
C
C  ***************************************************************
C  * STEP 5: INITIALIZE SIGNAL STRENGTH ALGORITHM PARAMETERS *
C  ***************************************************************
      AGCOLD=0.0
      ITXP=1
C
C  **********************************************************************
C  * STEP 6: SET TRACK INDICATOR TO ALLOW OPERA''ION OF TRACK LOOP *
C  **********************************************************************
      MTKINT=1
C
      ROLD=0.
      ICLOSE=0
      ICLOLD=0
C
C  NOTE: DEBUGGING PRINT STATEMENTS.
      WRITE(6.899)
      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     WRIT=(6.905)
C     WRITE(6-90 ) 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 UPDATES THE DATA VALID FLAG.STATUS *
C  ******************************************************
C
C
      SUBROUTINE TGTACQ
      COMMON /CNTL/IPWR.IMODE.ITXP.IASM.IDUMC(5).DUMC(3)
      COMMON /OUTPU /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  *********************************************************
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.
   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 UPDATES ALL REQUIRED TRANSFORMATION *
C  * MATRICES AND TRANSFORMATION MATRIX RATES.            *
C  ********************************************************
C
C
C
       SUBROUTINE TRNSFM
       COMMON /INPUT/DUM(9),TBT(3,3),TBTD(3,3)
       COMMON /SYSDAT/DUM2(4),CP,SP,DUM4(9)
       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)
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).
       TLB(1,1)=CB*CP-SB*SA*SP
       TLB(1,2)=CB*SP+SB*SA*CP
       TLB(1,3)=-SB*CA
       TLB(2,1)=-CA*SP
       TLB(2,2)=CA*CP
       TLB(2,3)=SA
       TLB(3,1)=SB*CP+CB*SA*SP
       TLB(3,2)=SB*SP-CB*SA*CP
       TLB(3,3)=CB*CA
```

```
C
C   STEP 1-3: COMPUTE TRANSFORMATION MATRIX TLT (TARGET-TO-LOS FRAME).
        DO 10 I=1.3
        DO 10 J=1.3
        TLT(I.J)=0.0
        DO 10 K=1.3
    10  TLT(I.J)=TLT(I.J)+TLB(I.K)*TBT(K.J)
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 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 /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)
        DIMENSION ROR(3).ROD(3).V1(3).RA(3).RL(3).RAD(3).RLD(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
      :5    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)
            II=0
            IF(II.EQ.1) GO TO 24
C
C     ****************************************************************
C     * STEP 3:  COMPUTE TARGET SCATTERING CHARACTERISTICS --- # OF *
C     *          ILLUMINATED POINTS. THE POINT LOCATIONS. AND THE   *
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     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)=1.0
            DO 25  I=1.3
      25    RT(1.I)=0.0
            GO TO 40
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)
C
C     STEP 3-3:  COMPUTE TARGET SCATTERING CHARACTERISTICS.
            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.
      CALL MULT31(TLTD,RT,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 SUBROUTINE GENERATES THE NOISE-FREE ANGLE, RANGE, VELOCITY *
C   * AND ON-TARGET DISCRIMINANT COMPONENTS.                           *
C   ********************************************************************
C
C
      SUBROUTINE SIGNAL
      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/IDUM6(2),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),
     2          RHOL(3)
      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/177.439,176.05,178.71,176.71,178.04/,
     2     ALAMD/1.2724E-2,2.9691E-2,3.3092E-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
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).
C
      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.
   20 DELX=CTP(MRNG,IMODE)*(RANGE(K)-SRNG)
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 FACTURS.
      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: COMFUTE 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: CUMPUTE 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 o-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
      WRITE(6,900) (I,SIG(I), I=1,NT)
  900 FORMAT(' I,SIG =',I8,F14.4)
      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 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
      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)
      COMMUN /TGTDAT/NT,DUM5(506),CGRNGE,CGVEL
      COMMON /DSCRM/AZDISC,ELDISC,RDISC,VDISC,RRTE,ODISC,SIGBR1,SNRD,
     2              SIGDB
      COMMUN /SIGDAT/SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE,DF1,DF5,
     2              DF2,DF4,SIGRAR
      COMMON /NOISE/NS1,NS2,NN(10),GAUSS(200)
      DIMENSION NFREQ(2),PDIA(2),PDIR(2),PDIV(2),PS(10,2),BN(2),PT(3)
      DATA NFREQ/1,5/,BN/9100.,526./,PS/9*1.,2.,5*1.,2.,4.,8.,8.,16./,
     2       PDIA,PDIR,PDIV/1.4142,3.1623,2.0,4.4721,2.8284,6.3246/,
     3       PT/50000.,3125.,195.3/
      REAL LATE,MEAN
```

```
C
C     NOTE: DEBUGGING PRINT STATEMENTS.
           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     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.
    5      YY=GP*PS(MRNG,IMODE)*PT(ITXP)/(CGRNGE**4*BN(MSAM))
           S1=YY/FLOAT(NFREQ(IMODE))
C
C     STEP 1-2: COMPUTE THE SNR AT THE OUTPUT OF THE DOPPLER FILTER
C                 (NOTE: THIS IS USED FOR DEBUGGING PURPOSES ONLY).
   10      SNRD=YY*SIGBAR
           SNRD=10.*ALOG10(SNRD)
           SIGDB=10.*ALOG10(SIGBAR)
           SIGBR1=SIGBAR
C
C     STEP 1-3: UPDATE NOISE SEQUENCE.
           NN(1)=MOD(NN(1)+1,200)+1
           DO 15 I=2,6
   15      NN(I)=MOD(NN(I-1)+29,200)+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.
           IF(IASM.GE.2.OR.IASM.GE.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.
           ID4=NN(4)
           SPAZ=ABS(ASCALE*SPAZ+MEAN+VARPAZ*GAUSS(ID1))
           SMAZ=ABS(ASCALE*SMAZ+MEAN+VARMAZ*GAUSS(ID4))
           ID2=NN(2)
           ID5=NN(5)
           SPEL=ABS(ASCALE*SPEL+MEAN+VARPEL*GAUSS(ID2))
           SMEL=ABS(ASCALE*SMEL+MEAN+VARMEL*GAUSS(ID5))
C
C     STEP 2-5: COMPUTE AZ AND EL DISCRIMINANT COMPONENTS.
           AZDISC=10.*ALOG10(SPAZ/SMAZ)
           ELDISC=10.*ALOG10(SPEL/SMEL)
```

```
C
C   ****************************************************
C   * SfEP 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.)
         VARLTE=SQRT(2.*S1*LATE+1.)
C
C   STEP 3-3: ADD EQUIVALENT NOISE TO RANGE DISCRIMINANT COMPONENT
C              SIGNALS.
         ID3=NN(3)
         ID6=NN(6)
         EARLY=ABS(RSCALE*EARLY+MEAN+VARELY*GAUSS(ID3))
         LATE=ABS(RSCALE*LATE+MEAN+VARLTE*GAUSS(ID6))
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.
         DF2=ABS(VSCALE*DF2+MEAN+VARDF2*GAUSS(ID1))
         DF4=ABS(VSCALE*DF4+MEAN+VARDF4*GAUSS(ID5))
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.
         DF1=ABS(VSCALE*DF1+MEAN+VARDF1*GAUSS(ID2))
         DF5=ABS(VSCALE*DF5+MEAN+VARDF5*GAUSS(ID6))
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.
```

```
      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  * THIS SUBROUTINE IMPLEMENTS THE BREAK-TRACK ALGORITHM *
C  ********************************************************
C
C
      SUBROUTINE BRKTRK
      COMMON /ICNTL/IDUM2(17),MBKTRK,MBTSUM,MBT(8)
      COMMON /DSCRM/DUM(3),VDISC,DUM1,ODISC,DUM2(3)
      INTEGER THRSHO,THRSHC
      DATA IVMAX,THRSHC,THRSHO/51,5,-11/
C
C  NOTE: VALUES FOR THRSHC AND THRSCHC ARE NOT THE VALUES
C        USED IN THE RADAR. THESE VALUES MUST BE CHANGED TO THE
C        RADAR VALUES.
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=INT(VDISC*5.333333+0.5)
C
C  STEP 1-2: DETERMINE STATUS OF L-H DISCRETE.
      IFTH=0
      IF(IABS(IVDISC).GT.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=INT(ODISC*5.333333+0.5)
C
C  STEP 2-2: DETERMINE STATUS OF ON-TARGET DISCRIMINANT.
      IOT=0
      IF(IODISC.GT.THRSHC) IOT=1
C
C  *****************************************************************
C  * STEP 3: DETERMINE STATUS OF ADJACENT ON-TARGET DISCRETE (AOT) *
C  *****************************************************************
      IAOT=0
      IF(IODISC.LT.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
      RETURN
      END
C
C
C   ************************************************************
C   * THIS SUBROUTINE UPDATES AZ AND EL INERTIAL LOS RATES, THE *
C   * ALPHA AND BETA GIMBAL RATES, THE ALPHA AND BETA GIMBAL    *
C   * POSITIONS, AND THE TARGET PITCH AND ROLL ANGLES FOR THE   *
C   * DISPLAY.                                                  *
C   ************************************************************
C
C
      SUBROUTINE ATRACK
      COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)
      COMMON /INPUT/DUM(6),EWB(3),DUM2(18)
      COMMON /OUTPUT/I1DUM(3),D1DUM(2),SPANG,SRANG,SPRTE,SRRTE,SRSS,
     2               IDUM1(4)
      COMMON /ICNTL/I2DUM(14),MRNG,IDUM2(12)
      COMMON /SYSDAT/TSAM,DR(3),CP,SP,PSI,PSBIAS,ALBIAS,BTBIAS,
     2               DUM4(5)
      COMMON /ATDAT/CA,SA,CB,SB,AZRATE,ELRATE,ALRATE,BTRATE,AL,BT,
     2               DUM3(4)
      COMMON /DSCRM/AZDISC,ELDISC,DUM1(7)
      DIMENSION AT1(10,2),AT2(10,2),TX1(3,3),TX2(3,3),TX3(3,3),TBL(3,3)
      DATA AT1/9*1.5529E-3,2.0106E-4,6*3.9750E-3,1.5529E-3,
     2         3*2.0106E-4/,AT2/9*6.5907E-3,2.3725E-3,
     3         6*1.0546E-2,6.5907E-3,3*2.3725E-3/
C   DEFINITION:  AT1=KEQ=(WN**2)/(4.*DIFFERENCE PATTERN SLOPE) WHERE
C                    WN IS NATURAL FREQUENCY OF THE LOOP.
C   DEFINITION:  AT2=KEQ*TAU WHERE TAU IS PROPORTIONAL TO STEP RESPONSE
C                    CONVERGENCE TIME.
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 2: UPDATE ESTIMATED TARGET INERTIAL AZIMUTH AND ELEVATION *
C   *         RATES IN ANTENNA LOS FRAME.                            *
C   ****************************************************************
C
C   QUANTIZE THE ANGLE DISCRIMINANTS TO 3/16 DB.
        IAZDSC=INTT(5.333333*AZDISC)
        IELDSC=INTT(5.333333*ELDISC)
        ADSC=0.0431*FLOAT(IAZDSC)
        EDSC=0.0431*FLOAT(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 REWUIRED COMPONENTS OF URBITER 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   ***************************************
C   * STEP 5: ANTENNA IN OBSCURATION REGION? *
C   ***************************************
C       CALL SCNWRN
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
        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  ***********************************************************
C  * THIS SUBROUTINE UPDATES RANGE AND RANGE RATE ESTIMATES, *
C  * UPDATES DOPPLER FILTER BANK POSITION, AND SYSTEM PARA-  *
C  * METERS BASED UPON RANGE INTERVAL.                       *
C  ***********************************************************
C
C
      SUBROUTINE RTRACK
      COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)
      COMMON /OUTPUT/IDUMO(3),SRNG,SRDOT,DUM2(5),IDUM(4)
      COMMON /ICNTL/I1DUM(14),MRNG,MSAM,MPRF,IDUM1(10)
      COMMON /SYSDAT/TSAM,DUMS(14)
      COMMON /RTDAT/IRDOT,IRNG,RBIAS,VEST(4),MDF(5)
      COMMON /DSCRM/DUM(2),RDISC,VDISC,RRTE,ODISC,DUM3(3)
      DIMENSION IPROM(128),RI(10),RT1(10,2),RT2(10,2),VT1(3),VT2(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 RI/120.,240.,780.,2552.,5772.,11544.,23089.,43747.,
     2        57722.,1.8228E+6/,RT1/9*0.125,0.25,4*0.125,
     3        2.,1.,2.,2*0.5,0.25/,RT2/9*0.5,4.0,4*0.5,8.,8.,
     4        4*16./,VT1/1.0125E-2,2.3627E-2,2.6334E-1/,VT2/1.20495,
     5        0.51638,0.046331/,NRI/10/
C
C
C  ****************************************************************
C  ***************** RANGE TRACKER MODEL **************************
C  ****************************************************************
C
C
C  *******************************************
C  * STEP1: UPDATE ROUGH RANGE RATE ESTIMATE *
C  *******************************************
C  INTEGERIZE RANGE DISCRIMINANT AND CHECK FOR SATURATION.
      RDISC=5.333333*RDISC
      IRDISC=INTT(RDISC)
      IF(IRDISC.GT.255) IRDISC=255
      IF(IRDISC.LT.-256) IRDISC=-256
C  ROUGH RANGE RATE PREDICTION FROM ALPHA-BETA TRACKING EQUATIONS.
C  DEFINITION:  RT1(MRNG,IMODE) CORRESPONDS TO BETA IN ALPHA-BETA TRACK.
      RR1=FLOAT(IRDISC)*RT1(MRNG,IMODE)
      IRDOT=IRDOT+INTT(RR1+0.5)
C
C  *********************************
C  * STEP 2: UPDATE RANGE ESTIMATE *
C  *********************************
C  DEFINITION: RT2 CORRESPONDS TO ALPHA IN ALPHA-BETA TRACKER.
      R1=FLOAT(IRDISC)*RT2(MRNG,IMODE)
      IRNG=IRNG+IRDOT+INTT(R1)
C  CONVERT RANGE ESTIMATE (IRNG) TO FEET USING THE FACT THAT THE LSB
C  OF IRNG REPRESENTS 5/16 FEET.
      RNG=0.3125*FLOAT(IRNG)
C  ADD FIXED BIAS TO FINAL RANGE ESTIMATE.
      SRNG=RNG+RBIAS
C
C
```

```
C     ***********************************************************************
C     ****************** VELOCITY PROCESSOR MODEL ***************************
C     ***********************************************************************
C
C
C     ********************************************
C     * STEP 1: GENERATE AMBIGUOUS VELOCITY ESTIMATE *
C     ********************************************
C     INTEGERIZE VELOCITY DISCRIMINANT AND CHECK FOR SATURATION.
            VDISC=5.333333*VDISC
            IVDISC=INTT(VDISC+0.5)
            IF(IVDISC.LT.-128) IVDISC=-128
            IF(IVDISC.GT.127) IVDISC=127
C     COMPUTE INTEGRAL FILTER NUMBER PORTION OF AMBIGUOUS VELOCITY
C     ESTIMATE.
            INTEG=MDF(2)
            IF(IVDISC.LT.0) INTEG=MOD(INTEG+1,32)
C     COMPUTE FRACTIONAL FILTER PORTION OF AMBIGUOUS VELOCITY
C     ESTIMATE.
            IV1=IABS(IVDISC)+1
            IFRAC=IPROM(IV1)
            IF(IVDISC.LT.0) IFRAC=127-IFRAC
C     COMPUTE AMBIGUOUS VELOCITY ESTIMATE --- COMBINE INTEGRAL AND
C     FRACTIONAL PARTS. NOTE: LSB IS 1/128 OF A FILTER WIDTH.
            IFVEL=IFRAC+128*INTEG
C
C     **********************************
C     * STEP 2: SCALE ROUGH VELOCITY ESTIMATE *
C     **********************************
C     SCALE LSB OF ROUGH ESTIMATE TO 4 TIMES A DOPPLER FILTER WIDTH
C     DEFINITION:  VT1(MPRF)=(RANGE LSB)/((MAX. UNAMBIGUOUS VELOCITY)/8)
C                  OR VT1(MPRF)=5./(PRF*LAMBDA)
C
            R1=FLOAT(IRDOT)*VT1(MPRF)/TSAM
            IR1=INT(R1)
C     SOME REQUIRED AUXILIARY CALCULATIONS.
            R2=FLOAT(IR1)/8.
            IR2=INT(R2)
            IRVEL=IR2*4096
C
C     ************************
C     * STEP 3: RESOLVE AMBIGUITY *
C     ************************
C     COMPUTE 3MSB'S OF AMBIGUOUS VELOCITY ESTIMATE.
            IF3=INT(FLOAT(IFVEL)/512.)
C     COMPUTE 3 LSB'S OF SCALED ROUGH VELOCITY ESTIMATE.
            IR3=IABS(IR1-8*IR2)
            IF(IR1.LE.0) GO TO 10
            IRVEL=IRVEL+4096
            IR3=7-IR3
   10     CONTINUE
C     COMPARE IF3 AND IR3.
            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     ******************************************
            IVEL=INTT(FLOAT(IRVEL-IFVEL))
C     SCALE LSB TO 0.05 FEET/SEC.
C     DEFINITION:   VT2(MPRF)=((FILTER SEPARATION)/128.)/(VELOCITY LSB)
C                   OR VT2(MPRF)=(PRF*LAMBDA)/(0.05*8196).
C
            IVEL=INTT(FLOAT(IVEL)*VT2(MPRF)+0.5)
C
```

```
C
C  ************************************************
C  * STEP 5: COMPUTE SMOOTHED UNAMBIGUOUS VELOCITY *
C  ************************************************
       DO 20 I=1,3
   20  VEST(5-I)=VEST(4-I)
       VEST(1)=FLOAT(IVEL)
C  AVERAGE AND SCALE ANSWER INTO FEET/SEC FROM 0.05 FEET/SEC.
       SRDOT=0.0125*(VEST(1)+VEST(2)+VEST(3)+VEST(4))
C
C  *************************************
C  * STEP 6: RESET DOPPLER FILTER BANK *
C  *************************************
C  CHECK ON-TARGET DISCRIMINANT FOR LARGE ACCELERATION DURING A DATA
C  CYCLE OR LOSS OF SIGNAL.
       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  RESET  REMAINING FILTERS IN BANK.
   40  DO 50 I=1,4
   50  MDF(1+1)=MOD(MDF(1)+I,32)
C
C
C  ******************************************************************
C  ************* UPDATE SYSTEM INTERNAL CONTROLS ********************
C  ******************************************************************
C
C
C  ****************************************
C  * STEP 1: SET RANGE INTERVAL PARAMETER *
C  ****************************************
       DO 60 I=1,NRI
       IF(RNG.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 60
76     MSAM=2
C
C  *****************************
C  * STEP 3: SET PRF PARAMETER *
C  *****************************
80     IF(IMODE.GE.2) GO TO 64
       IF(MRNG.GT.9) GO TO 62
       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
       RETURN
       END
```

```
C
C
C    ************************************************************
C    * THIS SUBROUTINE COMPUTES THE RADAR SIGNAL STRENGTH --- IT *
C    * IS SET EQUAL TO THE SNRV IN THE PRESENT VERSION.          *
C    ************************************************************
C
C
      SUBROUTINE RSS
      COMMON /CNTL/IDUM(2),ITXP,IDUMC(6),DUMC(3)
      CUMMUN /OUTPUT/IDUM2(3),DUM3(6),SRSS,IDUM4(4)
      COMMON /TGTDAT/NT,DUM1(500),RO(3),ROU(3),CGRNGE,CGVEL
      COMMON /DSCRM/DUM(6),SIGBAR,DUM2(2)
      COMMON /AGCDAT/AGC,AGCOLD
C
C    ************************************
C    * STEP 1: DEFINE TARGET PARAMETERS *
C    ************************************
C    SET THE RANGE TO TARGET C.G.
      RANGE=CGRNGE
C    SET TARGET RADAR CROSS-SECTION (INCLUDES BEAMSHAPE LOSS).
      SIGMA=SIGBAR
C
C    ***************************************
C    * STEP 2: COMPUTE THE SNR AT VIDEO OUTPUT *
C    ***************************************
      SNR=SNRV(SIGMA,RANGE)
C
C    ****************************
C    * STEP 3: COMPUTE RSS IN DB *
C    ****************************
      SRSS=10.*ALOG10(SNR)
      RETURN
      END
C
C
C    **************************************************************
C    * THIS SUBRUUTINE INITIALIZES ALL DATA REQUIRED BY THE SEARCH, *
C    * ACGUISITION, AND TRACK SUBPROGRAMS.                          *
C    **************************************************************
C
C
      SUBROUTINE DATA
      COMMON /RTDAT/IDUM1(2),RBIAS,DUM1(9)
      CUMMON /SYSDAT/TSAM,DR(3),CP,SP,PSI,PSBIAS,ALBIAS,BTBIAS,GP,GA,
     2               TGTSIG,GPS,GAS
      CUMMUN /NUISE/NS1,NS2,NN(10),GAUSS(200)
      REAL LT,KTS
C    *********************
C    * SYSTEM PARAMETERS *
C    *********************
      PI=3.1415926
      PI1=PI/180.
C    RADAR FRAME YAW ANGLE IN BODY COORDINATES (DEGREES).
      PSI=PI1*67.0
C     PSI=PI1*0.0
      CP=COS(PSI)
      SP=SIN(PSI)
C    RADAR LOCATION OFFSET FROM ORBITER C.G. IN BODY COORD. (FEET)
      DR(1)=48.0
      DR(2)=11.0
      DR(3)=-6.0
C     DO 4 I=1,3
C   4 UR(I)=0.0
C    RANGE BIAS ERROR.
      RBIAS=0.0
C    ALPHA GIMBAL BIAS.
      ALBIAS=0.0
C    BETA GIMBAL BIAS.
      BTBIAS=0.0
C    RADAR YAW ANGLE ERROR WRT BUDY FRAME.
      PSBIAS=0.0
```

```
C
C     **************************
C     * SYSTEM SAMPLE INTERVAL *
C     **************************
      TSAM=0.2
C
C     ************************
C     * COMPUTE SNR CONSTANT *
C     ************************
C     EQUIVALENT ONE-SIDED NOISE POWER SPECTRAL DENSITY (MW/KHZ)
      KTS=-136.6
      KTS=10.**(0.1*KTS)
C     SYSTEM LOSSES ON TRANSMIT (DB).
      LT=3.7
      LT=10.**(0.1*LT)
C     ONE-WAY ANTENNA GAIN (DB).
      G=38.5
      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=1,200
    2 GAUSS(I)=ANORM(NS1,NS2)
C
C     ****************************
C     * DEFINE TARGET PARAMETERS *
C     ****************************
C     TARGET SEARCH CROSS-SECTION ( FIXED TEMPORARILY).
      TGTSIG=10.0
      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 GIVES THE ANTENNA DIFFERENCE PATTERN WEIGTHING OF *
C  * THE RADAR SIGNAL FOR THE GIVEN ANGLE(IN RADIANS) OFF BORESIGHT. *
C  * NOTE: THIS PATTERN IS THE DERIVATIVE OF THE SUM PATTERN         *
C  *******************************************************************
C
C
      FUNCTION DPAT(X)
      IF(ABS(X).GT.1.E-4) GO TO 10
      DPAT=-0.6228*X
      RETURN
10    Y=93.80*X
      DPAT=1.1465*(Y*COS(Y)-SIN(Y))/(Y*Y)
      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=6.2831652
      ANORM=SQRT(-2.*ALOG(Y1))*COS(TPI*Y2)
      RETURN
      END
C
C
C  ****************************************************************
C  * THIS FUNCTION GENERATES A RANDOM NUMBER FROM A UNIFORM 90,1I *
C  * DISTRIBUTION.                                                *
C  ****************************************************************
      FUNCTION RNDU(IRAN)
      DATA MU/524287/,XMU/524287./,IETA/997/
      IF(IRAN) 20,10,20
   20 CONTINUE
      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 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))
      DUPFIL=NUMER/DENOM
      RETURN
      END
```

```
C
C     ****************************************************************
C     * THIS FUNCTION CHECKS FOR NEGATIVE ARGUMENT FOR INT FUNCTION *
C     * AND CORRECTS THE QUANTIZATION PROCEDURE.                     *
C     ****************************************************************
C
C
      INTEGER FUNCTION INTT(Y)
      X=Y
      IF(X.LT.0.0) X=X-1.0
      INTT=INT(X)
      RETURN
      END
C
C
C     ******************************************************************
C     * THIS SUBROUTINE GENERATES A (3X3) MATRIX TPHD THAT REPRESENTS *
C     * THE DERIVATIVE OF A MATRIX THAT REPRESENTS UNIFORM ROTATION    *
C     * ABOUT THE Z-AXIS. THE ROTATION SPEED IS W AND THE ANGLE AT     *
C     * WHICH THE DERIV. IS TAKEN IS PH.                               *
C     ******************************************************************
C
C
      SUBROUTINE PHID(TPHD,PH,W)
      DIMENSION TPHD(3,3)
      DO 10 I=1,3
      TPHD(3,I)=0.0
   10 TPHD(I,3)=0.0
      TPHD(1,1)=-W*SIN(PH)
      TPHD(2,2)=TPHD(1,1)
      TPHD(1,2)=W*COS(PH)
      TPHD(2,1)=-TPHD(1,2)
      RETURN
      END
C
C     ****************************************************************
C     * THIS SUBROUTINE MULTIPLIES THE (3X3) MATRIX A AND THE (3X3) *
C     * MATRIX B TO OBTAIN THE (3X3) MATRIX C.                       *
C     ****************************************************************
C
C
      SUBROUTINE MULT33(A,B,C)
      DIMENSION A(3,3), B(3,3), C(3,3)
      DO 10 I=1,3
      DO 10 J=1,3
      C(I,J)=0.0
      DO 10 K=1,3
   10 C(I,J) = C(I,J)+A(I,K)*B(K,J)
      RETURN
      END
C
C     ****************************************************************
C     * THIS SUBROUTINE MULTIPLIES THE (3X3) MATRIX A AND THE (3X1) *
C     * VECTOR B TO OBTAIN THE (3X1) VECTOR C.                       *
C     ****************************************************************
C
C
      SUBROUTINE MULT31(A,B,C)
      DIMENSION A(3,3),B(3),C(3)
      DO 10 I=1,3
      C(I)=0.0
      DO 10 J=1,3
   10 C(I) = C(I)+A(I,J)*B(J)
      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 GENERATES A (3X3) MATRIX TPH THAT PRODUCES *
C  * A ROTATION OF PH RADIANS ABOUT THE Z-AXIS.                 *
C  **************************************************************
C
C
      SUBROUTINE PHI(TPH,PH)
      DIMENSION TPH(3,3)
      DO 10 I=1,3
      DO 10 J=1,3
10    TPH(I,J)=0.0
      TPH(3,3)=1.
      TPH(1,1)=COS(PH)
      TPH(2,2)=TPH(1,1)
      TPH(1,2)=SIN(PH)
      TPH(2,1)=-TPH(1,2)
      RETURN
      END
C
C
C  **************************************************************
C  * THIS SUBROUTINE GENERATES A (3X3) MATRIX TGA THAT PRODUCES *
C  * A ROTATION OF GA RADIANS ABOUT THE Y-AXIS.                 *
C  **************************************************************
C
C
   10 TGA(I,J)=0.0
```

```
C
C
C     ******************************************************
C     * THIS SUbROUTINE MODELS THE SPAS SPACECRAFT SCATTERING *
C     * PRuPERTIES.                                         *
C     ******************************************************
C
C
      SUBROUTINE SPAS
      COMMON /SATDAT/RADAR(3),KTAR,R(70,3),SIG(70),ROLD,ICLOSE,ICLOLD
      DIMENSION SIGMA(63),TARG(63,3),PHIMIN(63,3),PHIMAX(63,3)
      DIMENSION OFFSET(63),JHOT(63),JHOT20(63),PHI(63,3)
      DIMENSION VECT(3),COSPHI(63,3)
      DIMENSION ALPH(20,3),V(20,3),NORMAL(20),DIM(20,3),WRAN(20,3)
      DIMENSION WSCALE(20,3),DPHI(20),PHIOLD(20),VOLD(20,3),KSEED(20,3)
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/45,678,908,607,5678,697,345,77777,67,4,
     1 560,809,444,888,999,555,222,70,60,8000,
     2 5,15,25,35,45,55,65,75,85,95,
     3 7,17,27,37,47,57,67,77,87,97,
     4 9876,984,6666,2398,76,412,7639,409,899,561,
     5 205,3695,9457,9643,93467,967656,453,980,567,2154/
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 /60*64.8/
      DATA WSCALE /60*0.2965/
C
C     FOR EACH DIFFUSE SCATTERER, SPECIFY NORMAL COMPONENT
      DATA NORMAL /3*1,2,2,9*3,6*1/
C
C
C     SQUARE ROOT OF RCS VALUES ( FEET).
      DATA SIGMA/20*.734,3*5.29,6*25.6,16.6,109.,98.4,104.,95.7,114.,
     2 189.,1.467.,110.,2*87.,2*92.8,2*104.,2*98.4,2*95.6,89.9,2*95.6,
     3 89.9,99.5,88.6,1.47,6*.568,3.67,1.35/
C
C     X-COORDINATES OF SCATTERERS IN SPAS FRAME (FEET)
      DATA TARG /3*.39,17*-1.15,3*.79,6*1.21,-1.15,3*.39,2*-.98,
     1 2*-1.15,-.98,10*-1.15,6*1.21,-1.15,6*.79,2*-1.15,
C
C     Y-COORDINATES OF SCATTERERS IN SPAS FRAME (FEET)
     2 -3.44,-5.74,5.74,7.05,-7.05,-5.74,.79,2*5.74,3.44,1.15,-1.15,
     3 -3.44,-5.74,5.74,3.44,1.15,-1.15,3.44,5.74,-2.72,-3.44,-4.17,
     4 2*3.44,2*1.15,2*-1.15,-3.44,6.23,-3.44,-5.91,6.56,3*-6.56,0.,
     5 2*5.58,2*3.44,2*1.15,2*-1.15,2*-5.74,3.44,3.44,2*1.15,2*-1.15,
     6 -7.05,2*-2.72,2*-3.44,2*-4.17,2*-3.44,
C
C     Z-COORDINATES OF SCATTERERS IN SPAS FRAME (FEET)
     7 5*0.,-2.95,-1.64,-2.30,12*0.0,3*.49,6*0.0,-2.62,3*0.0,2*-2.20,
     8 2*0.0,2.2,10*-1.57,1.39,-1.39,1.39,-1.39,1.39,-1.39,0.0,-0.07,
     9 .98,-0.07,.98,-0.07,.98,2*-2.62/
```

```
C
C   MINIMUM SUBTENDED ANGLE IN X-DIRECTION
        DATA PHIMIN /14*1.,6*0.,4*1.,0.,1.,0.,1.,0.,9*1.,10*.026177,
     1 6*.034899,9*1.,
C
C   MINIMUM SUBTENDED ANGLE IN Y-DIRECTION
     2 4*1.,0.,15*1.,.64279,.81915,.86603,1.,.29237,.90631,.682,.90631,
     3 .64805,5*1.,-.99897,-.99905,-.89415,.00524,10*.02618,6*.034899,
     4 -.89415,8*1.0,
C
C   MINIMUM SUBTENDED ANGLE IN Z-DIRECTION
     5 5*1.,3*0.,12*1.,3*.071497,6*.02618,.04013,3*.02618,2*.04536,
     6 .04362,.43837,2*1.,-.99966,1.,-.99966,1.,-.99966,1.,-.99966,
     7 1.,-.99966,1.,-.99939,1.,-.99939,1.,-.99939,.44776,0.,1.,0.,1.,
     8 0.,1.,-.91355,0./
C
C   MAXIMUM SUBTENDED ANGLE IN X-DIRECTION
        DATA PHIMAX /3*0.,17*-1.,4*0.,-1.,0.,-1.,0.,2*-1.,3*.99933,5*-1.,
     2 10*-.02618,6*-.034899,-1.,6*0.,2*-1.,
C
C   MAXIMUM SUBTENDED ANGLE IN Y-DIRECTION
     3 3*-1.,0.,16*-1.,2*-.86603,-.90631,-.81915,3*-.90631,-1.,
     4 -.70711,-.81915,3*-1.,.99897,3*-1.,-.00524,10*-.02618,6*-.034899,
     5 9*-1.,
C
C   MAXIMUM SUBTENDED ANGLE IN Z-DIRECTION
     6 8*-1.,6*0.,6*-1.,3*-.071497,6*-.02618,-.040132,3*-.02618,
     7 2*-.04536,-.04362,-.43837,-.57358,.99966,-1.,.99966,-1.,
     8 .99966,-1.,.99966,-1.,.99966,-1.,.99939,-1.,.99939,-1.,.99939,
     9 -1.,-.44776,-1.,0.,-1.,0.,-1.,0.,-1.,-.91355/
C
C   RADII OF THE SCATTERERS (FEET)
        DATA OFFSET /20*0.0,3*-.33,6*-.95,-1.03,7*0.,-.79,17*0.,
     1 6*-.33,2*-1.15/
C
C   MISCELLANEOUS DATA.
        DATA NTAR/63/,KWIDE/20/,PI/3.141592653/
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.
        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
        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)
    22  SIG(K)=SQRT(ABS(COSPHI(I,NORMAL(I))))*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.300.) 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
        PHIOLD(I)=ACOS(COSPHI(I,NORMAL(I)))
        DO 25 J=1,3
        IF(J.EQ.NORMAL(I)) 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
        PHI(I,NORMAL(I))=ACOS(COSPHI(I,NORMAL(I)))
        DPHI(I)=(PHI(I,NORMAL(I))-PHIOLD(I))
        PHIOLD(I)=PHI(I,NORMAL(I))
    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
      IF(J.EQ.NORMAL(I)) GO TO 45
      ALPH(I,J)=EXP(-DIM(I,J)*ABS(DPHI(I)*COSPHI(I,NORMAL(I))))
      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
      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
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)
      WRITE(6,904) (I,JHOT(I),SIG(I),(R(I,J),J=1,3),
     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     WRITE(6,906) I,PHI(I,NORMAL(I)),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
```