Technical Report 32-1305

Mariner IV and V Disturbance Torques and Limit Cycles

Daniel A. Prelewicz

October 1, 1968
Mariner IV and V Disturbance Torques and Limit Cycles

Daniel A. Prelewicz

Approved by:

G. E. Sweetnam, Manager
Spacecraft Control Section

October 1, 1968
Foreword

This report explains the procedure used to compute the disturbance torques affecting the Mariner IV and Mariner V spacecraft during selected periods of flight and presents representative computer printouts showing these torques and the resulting spacecraft limit cycles.

More extensive examples of the computer printouts, which provide more complete quantitative information, are given in the Addendum to this report (TR 32-1305, Addendum: Mariner IV and V Disturbance Torques and Limit Cycles).
Acknowledgments

The author wishes to express his thanks to the numerous JPL personnel who contributed to this endeavor, especially to Boris Dobrotin, whose keen interest and encouragement made this work possible, and to E. H. Kopf, Jr., whose advice proved to be most valuable and who was responsible for the computer program from which the Lister program in this study was adapted.

Also, special thanks are due to Mrs. Mary Fran Buehler and the other members of the Publications Section who put this report into final form.
# Contents

## I. Introduction .......................... 1
   A. Mariner Attitude Control System .... 1
   B. Telemetry Data ..................... 2
   C. Spacecraft Dynamics ............... 2

## II. Determination of the Disturbance Torques ........... 3
   A. Position Sensor Calibration ....... 3
   B. Single-Value and Interval Analyses . 3
   C. Spacecraft Rotation in the XYZ Coordinate System . 5
   D. The Limit Cycles .................. 5

## III. Character of the Disturbance Torques ............ 11
   A. Explanation of the Data ........... 11
   B. Mariner IV Disturbance Torques .... 11
   C. Mariner V Disturbance Torques ..... 12

## IV. Conclusions ........................ 12

References ................................ 12

Appendix A. Fitting a Parabola by Interval Analysis .... 53

Appendix B. Computer Programs ................ 54

## Figures
1. Mariner coordinate reference system .................. 2
2. Edplot data for pitch, roll, and yaw, Mariner IV, day 193, 1965 ..... 4
3. Mariner V pitch sun sensor calibration curve ........... 5
4. Reduced position data vs time ..................... 6
5. Pitch, roll, and yaw least-squares fit residuals, Mariner V, day 177, 1967 .... 7
6. Typical plot of Canopus sensor output vs angular displacement .... 10
7. Ambiguous data points in the limit cycles, Mariner IV, day 362, 1964 .... 13
8. Improper detection of attitude control thruster firing, Mariner IV, day 366, 1964 . 18
9. Angular velocity for a single thruster firing, Mariner IV, day 366, 1964 .... 22
10. Evidence of restoring torque, Mariner IV, days 361–362, 1964 ........ 25
11. Torque change accompanying valve firing, Mariner IV, day 362, 1964 .... 34
12. Limit cycles at the low bit rate of 8½ bits/s, Mariner IV, day 105, 1965 .... 40
13. Torque characteristics, Mariner V, day 183, 1967 ............ 45
Abstract

The disturbance torques acting upon the Mariner IV and Mariner V spacecraft during cruise mode operation are described. The history of the spacecraft rotational motion was obtained by processing sun sensor and Canopus sensor telemetry data, which were used in conjunction with the spacecraft dynamics to obtain qualitative and quantitative disturbance torque characteristics. Interval analysis was used to account for quantization error introduced by the telemetry system. This procedure, together with the assumption of parabolic limit cycles, established definite upper and lower bounds on the disturbance torques at any time.

Data for both Mariner IV (Mars 1964) and Mariner V (Venus 1967) indicate that low disturbance torques are present for both spacecraft. Mariner IV has a comparatively large (between 10 and 30 dyn-cm), slowly varying bias torque (apparently a solar torque) as well as a smaller component which changes by as much as 3–5 dyn-cm when a control valve fires. A small restoring torque (~1/4 dyn-cm/mrad) in pitch and yaw indicates that the spacecraft is stable about the sun line.

Mariner V is symmetrical about the sun line and hence does not have a large bias solar torque. Since there are no solar vanes, the solar restoring torque is also considerably smaller. However, the disturbance torque, which varies randomly with valve firing (by as much as 2–3 dyn-cm), is present.
1. Introduction

The limit cycle operation of a spacecraft bang-bang three-axis attitude control system under conditions of constant bias torque and large ratios of control torque to disturbance torque is well understood. However, a spacecraft with a similar attitude control system in interplanetary travel (cruise) is subject to different conditions. This report presents the disturbance torques acting on such a spacecraft as well as the spacecraft response. Since this is the first time that such data have been available, they should provide material for advances in the state of the art for three-axis stabilized interplanetary cruise attitude control systems.

A. Mariner Attitude Control System

Mariner spacecraft are attitude-stabilized with respect to the sun and the star Canopus. The attitude control system is described in Ref. 1. Briefly, the system is a bang-bang three-axis-stabilized attitude control system using the sun and Canopus as references.

Shortly after launch and again after the midcourse maneuvers, the attitude control system operates in the acquisition mode to attain the following spacecraft orientation, which is shown in Fig. 1.

(1) For a Venus (Mars) mission, Z (−Z) of the standard XYZ spacecraft fixed coordinate system is coincident with the sun vector S.

(2) The X axis forms a constant angle (X axis clock angle) with the S–Canopus vector C plane.

During the remainder of the flight (barring large disturbances or commands from earth) the system operates in the cruise mode to maintain this attitude. The solar panels are then properly exposed to the sun, and the low-gain antenna is pointed toward the earth.

Deviations from this nominal attitude are measured by position sensors mounted on the spacecraft. Rotations about the X axis (pitch) and the Y axis (yaw) are measured by the sun sensors which put out error signals related to the rotations. The Canopus sensor does not measure rotations about Z but rather about an axis V (roll) which never differs from Z by more than about 15 deg. This angle changes during a flight as the direction of Canopus changes relative to the spacecraft. For attitude control purposes, this measurement is used along with those of the sun sensors to establish angular position deadbands. When the spacecraft rotates to the edge of a deadband, cold gas thrusters are fired for a fixed minimum time period (nominally 20 ms), applying a restoring torque about the X, Y, or Z axis, depending upon which deadband limit was reached. Note that the system applies a restoring torque about the Z axis to establish a deadband about the V axis. If there are no null offsets in the sensors, the deadbands are centered on the nominal attitude and are approximately 1 deg wide for pitch and yaw and ½ deg
wide for roll. Cruise mode attitude control thus establishes limit cycle rotational motion within the angular position deadbands.

Although the gas systems are essentially the same on Mariner IV and Mariner V, the torque environment in which they operate is not. Unlike Mariner V, Mariner IV is unsymmetrical about the sun vector, and, as a result, unbalanced solar torques in pitch and yaw are comparatively large (between 10 and 30 dyn-cm). Solar vanes were attached at the ends of the solar panels in an attempt to reduce these unbalanced torques. Each time a thruster fired, the appropriate vanes were stepped 0.01 deg in the direction which decreased the unbalanced torque. However, the solar vanes were only partially successful and hence the torque environment is not the same for both spacecraft.

B. Telemetry Data

At the beginning of a Mariner mission, when the spacecraft is near the earth, data are sent at the high rate (33½ bits/s). Approximately 40 days into the mission, the rate is cut to 8½ bits/s. During cruise mode operation, the output of the position sensors is sampled every 12.6 (50.4) s when transmission is at 33½ (8½) bits/s. The output is converted to a 7-bit data word (a binary number between 0 and 127) by a data encoder on the spacecraft and sent via the telemetry channel to tracking stations of the Deep Space Network. Processing of these data to obtain a history of the limit cycle motion in the X, Y, Z system is discussed in Section II.

C. Spacecraft Dynamics

Consider a coordinate system X₁, X₂, X₃ fixed to the spacecraft with its origin at the center of mass. In such a coordinate system, the rotational motion of the spacecraft is governed by the following equations adapted from Ref. 2.¹

\[
\begin{align*}
M_x &= \dot{\omega}_x J_{xx} + \omega_y \omega_z (J_{yx} - J_{yz}) - J_{xy} (\omega_y \omega_z - \dot{\omega}_y) \\
&\quad + J_{xz} (\dot{\omega}_z + \omega_y \omega_z) + J_{yz} (\omega_z^2 - \omega_y^2) \\
M_y &= \dot{\omega}_y J_{xy} + \omega_z \omega_x (J_{zx} - J_{xz}) - J_{yz} (\omega_z \omega_x - \dot{\omega}_z) \\
&\quad + J_{yz} (\dot{\omega}_z + \omega_x \omega_z) + J_{xz} (\omega_z^2 - \omega_x^2) \\
M_z &= \dot{\omega}_z J_{zz} + \omega_x \omega_y (J_{zy} - J_{zx}) - J_{xy} (\omega_x \omega_y - \dot{\omega}_x) \\
&\quad + J_{xz} (\dot{\omega}_x + \omega_x \omega_z) - J_{yz} (\omega_x^2 - \omega_z^2)
\end{align*}
\]  

(1)

or, in vector form,

\[
T = J \ddot{\omega} + \omega \times J \omega
\]

(2)

where T is the vector torque, \(\omega\) is the angular velocity about the center of mass as seen from an inertial reference, and J is the inertia matrix. That is,

\[
\begin{bmatrix}
T_1 \\
T_2 \\
T_3
\end{bmatrix} = \begin{bmatrix}
\frac{d\theta_1}{dt} \\
\frac{d\theta_2}{dt} \\
\frac{d\theta_3}{dt}
\end{bmatrix} \quad \begin{bmatrix}
\ddot{\theta}_1 \\
\ddot{\theta}_2 \\
\ddot{\theta}_3
\end{bmatrix} = \begin{bmatrix}
\frac{d^2\theta_1}{dt^2} \\
\frac{d^2\theta_2}{dt^2} \\
\frac{d^2\theta_3}{dt^2}
\end{bmatrix}
\]

(3)

\[
J_{ij} = \int \rho \left( \delta_{ij} r^2 - x_i x_j \right) \, dm
\]

\[
r^2 = x_1^2 + x_2^2 + x_3^2
\]

If X₁, X₂, X₃ are taken parallel to XYZ respectively,² then the rotational motion in the X₁, X₂, X₃ system is the same as that in the XYZ system.

¹The inertia matrix J is used in Eq. (1), rather than the inertia tensor I of Ref. 2, in order that Eq. (1) can be written in the vector form (Eq. 2).

²Since the origin of X, Y, Z is not at the center of mass, the two systems differ by a translation.
Since the nominal attitude changes slowly with respect to an inertial reference as the spacecraft orbits the sun, as explained above, the rotational motion measured in XYZ (and hence relative to the nominal attitude) is not the same as the rotational motion viewed from an inertial reference. However, for purposes of determining disturbance torque levels, the difference between the two motions is negligible and the rotational motion in XYZ (or equivalently X, Y, Z) can be used in Eq. (2). Also, for cruise-mode limit cycle motion, the magnitude of the angular velocity is small enough to make the second term on the right in Eq. (2) very small (usually on the order of 0.01 dyn-cm). Hence the governing equation can be simplified to

$$T = J\ddot{\omega} = J\ddot{\theta}$$

where $\ddot{\omega}$ is the angular acceleration measured relative to the nominal attitude.

In Section II, $\ddot{\theta}$ is determined as accurately as possible from the position sensor telemetry data and substituted into Eq. (4) to obtain the disturbance torque levels.

II. Determination of the Disturbance Torques

Telemetry data from the position sensors provide a rather crude record of the spacecraft rotational motion. The central problem of this section is to reconstruct the limit cycles as accurately as possible from these data. Equation (4) can then be used to determine the disturbance torques.

As previously mentioned, the telemetry data provide a list of data numbers (DN) at the sample times for each of the position sensors. Here DN is the 7-bit data word used in the telemetry. A plot of these raw data (called an Edplot) is shown in Fig. 2. (The plot shows celestial sensor data as they were received during the flight.) The best data from all Deep Space Network tracking stations are stored on the Master Data Library (MDL) tapes. Input to a computer program that processes the raw data is obtained from these tapes.

A. Position Sensor Calibration

Before the spacecraft is flown, the position sensors are calibrated to determine the angular position–DN relationship. For calibration purposes, DN can be considered to be a continuous variable (which is later rounded off to an integer value by the data encoder). The value of DN corresponding to a number of angular displacements (13 for Mariner V) is determined, and a polynomial is fitted to these points. A typical calibration curve is shown in Fig. 3.

In addition to being sampled, the raw data are also quantized, since only integer values of DN are sent via the telemetry. Hence, a given DN indicates that the angular position is within some interval. For example, a DN of 64 in the pitch channel for Mariner V indicates that the angular position is between $-0.0944$ and $+0.1488$ mrad, a range corresponding to DN = 64.5 and DN = 63.5, respectively, on the calibration curve.

B. Single-Value and Interval Analyses

At this point, a single value (rather than an interval) for the angular displacement could be obtained from the calibration curve by assuming that the DN of the telemetry data is exact. The angular displacement would then assume values from a discrete set; the values would, however, be contaminated with quantization errors. These data could be processed to obtain displacements in the XYZ system. When limit cycle curves are fitted to these data, quantization errors would be expected to average out. This is true to some extent, and this procedure is one of those used in the data reduction.

Alternatively, the same computation is done using interval analysis (Ref. 3). The displacements are then characterized by an ordered pair corresponding to the interval limits discussed earlier. Henceforth, either a barred variable (e.g., $\bar{T}$, $\bar{\theta}$) or a bracketed ordered pair, i.e., [Upper, Lower] will be used to denote an interval variable.

Several benefits of using interval analysis are:

1. The intervals are overlapped to account for inaccuracies in data acquisition. For example, since the data encoder cannot round off exactly, there is a range of variables which may round off either way. Also, there is electrical noise in the sensor output (especially the Canopus sensor) which can cause transmission of an erroneous DN. In the example of the Mariner V pitch channel, interval overlapping results in the association of the displacement interval $[+0.1659 \text{ mrad}, -0.1114 \text{ mrad}]$ (corresponding to DN = 63.43 and DN = 64.57 respectively on the calibration curve) with a DN of 64. Other pitch and yaw displacement intervals are also overlapped by the amount cited in this example. Overlapping for the roll displacements is discussed below in Section C.
Fig. 2. Edplot data for pitch, roll, and yaw, *Mariner IV*, day 193, 1965
D. The Limit Cycles

Figure 4 is a typical plot of angular position for the X, Y, Z, and V axes at the sample times. For interval analysis, a displacement interval is associated with each point. The rotational motion of the spacecraft will now be determined by fitting continuous curves to these data.

Consider limit cycle segments terminated by an attitude control thruster firing on any one of the three axes. As a starting point for the analysis it is assumed that $T$ is constant during each such segment. Equation (4) then implies that the limit cycle segments are parabolas. Subsequently, it will be established that there is, in fact, a restoring torque, and therefore this assumption is not strictly valid. However, this restoring torque is small enough to be treated as a perturbation on the general parabolic nature of the limit cycle segments. When parabolas are fitted to these segments by least squares and by interval analysis, the following was observed:

1. The residuals of the least-squares fit appear to have the character of quantization error only (see Fig. 5).

2. The limit cycles were never so "nonparabolic" that no parabola could be fitted through all the intervals used (see Appendix A).

It can also be inferred that the disturbance torque component about any one axis does not change significantly during a complete limit cycle on that axis. However, a thruster firing on any axis applies a significant torque to the other two axes via inertial cross coupling and thruster misalignment. Hence, the complete limit cycles are not parabolic. If the misalignments were known, the torque input due to firings on each axis could be determined and the data adjusted to make the limit cycles appear parabolic. However, since there is no information at this time regarding thruster misalignment, the rotational motion was reconstructed by fitting parabolas to the limit cycle segments. For purposes of ordinary analysis, the standard least-squares method of curve fitting was used. The method of fitting parabolas to the data using interval analysis is discussed in Appendix A.

Since many of the segments do not contain enough points for a meaningful curve fit, the record of rotational motion contains time gaps. That is, when a limit cycle segment contained less than 16 points for data at 8½ bits/s or less than 64 points for 33½ bits/s, the torque interval was so large (around 5 dyn-cm) that it was not worth the effort of processing the data.

C. Spacecraft Rotation in the XYZ Coordinate System

Recall that the position sensors measure rotations in the XYV coordinate system, while the dynamical equations hold in the $X_1 X_2 X_3$ system. Angular displacements in the $X_1 X_2 X_3$ (or equivalently the XYZ) system are determined from XYV rotations by the following set of equations.

\[
\begin{align*}
\theta_x &= \theta_x \\
\theta_y &= \theta_y \\
\theta_z &= \begin{cases} 
\theta_y - \cos \alpha (\theta_y \sin \beta + \theta_z \cos \beta) & \text{(Mars mission)} \\
\theta_y - \cos \alpha (\theta_y \sin \beta - \theta_z \cos \beta) & \text{(Venus mission)}
\end{cases}
\end{align*}
\]

where $\alpha$ is the Canopus cone angle and $\beta$ is the X axis clock angle. These equations are accurate to first order in the small angles $\theta_x$, $\theta_y$, and $\theta_z$ and are easily derived by geometrical arguments.
Fig. 4. Reduced position data vs time. Time is in hours, minutes, and seconds (GMT). Symbols: * = pitch; $X$ = yaw; $O$ = V-axis roll; $H$ = Z-axis or true roll; $\$ = superimposed data points
Fig. 5. Pitch, roll, and yaw least-squares fit residuals, Mariner V, day 177, 1967. Time is in hours, minutes, and seconds (GMT). Symbols: * = pitch; X = yaw; O = V-axis roll
Fig. 5 (contd)
Also, considerable overlapping of the roll displacement intervals was necessary to allow for the large electrical noise component in the output of the Canopus sensor. A typical plot of Canopus sensor output vs angular displacement (Fig. 6) reveals the character of this noise component, while the sampled data plots (Fig. 4) show its obvious effects (compare the noisy roll data with the pitch and yaw data).

The displacement intervals for the roll axis were overlapped as follows: the angular displacement interval associated with each DN of the telemetry data corresponds to the range of values between DN +0.75 and DN −0.75 on the calibration curve. Thus, for Mariner V, the interval [2.236 mrad, 1.535 mrad], corresponding to DN = 63.25 and DN = 64.75 respectively on the calibration curve, is associated with DN = 64.

As a consequence of this rather large overlapping, the roll torque intervals tend to be large. However, it should be noted that the scarcity of information regarding the roll torque does not significantly affect the determination of disturbance torque characteristics about the pitch and yaw axis.

Fig. 6. Typical plot of Canopus sensor output vs angular displacement
III. Character of the Disturbance Torques

The following amount of data was processed using a computer program which reduces the MDL tape data:

(1) 6 days (days 361–366 of 1964) of the Mariner IV mission at 33½ bits/s.
(2) 28 days (days 102–129 of 1965) of the Mariner IV mission at 8½ bits/s.
(3) 12 days (days 172–183 of 1967) of the Mariner V mission at 33½ bits/s.
(4) The changeover from 33½ to 8½ bits/s for the Mariner V mission (days 204–214 of 1967).

Selected sequences of these data are presented in Figs. 7–13. These particular sequences are typical of the data acquired and are presented to illustrate the types of behavior encountered. The two computer programs used, the Lister program and the Data Reduction program are reproduced in Appendix B.

A. Explanation of the Data

Rotations about the V and Z axes (denoted by O, roll, and H, true roll, respectively) and the X and Y axes are shown on the limit cycle plots. Disturbance torques are computed by both of the previously mentioned methods (least-squares and interval analysis parabola fits).

The heavy vertical lines denote the end of one limit cycle segment and the beginning of the next. The location of these lines was determined by a computer program which sought particular patterns in the data points (only points near the edge of the deadband were considered) typical of an attitude control thruster firing. Since failure to detect a firing resulted in a complete loss of torque data for the limit cycle segment in question, the firing detection routine is purposely oversensitive. In many cases (especially for data at 33½ bits/s) more than one vertical line may correspond to a single attitude control thruster firing. This is necessary since it may be impossible to assign a particular data point to a given limit cycle segment with complete certainty. Since limit cycle segments with less than 64 (16) points for data at 33½ (8½) bits/s are not processed, as explained above, the information conveyed by these “ambiguous data points” is lost. (It might be possible to recover this information by more sophisticated analysis of each firing.) As an example of “ambiguous points,” consider the firing on the yaw axis at approximately 7 h, 59 min of day 362 (see Fig. 7). It is impossible to assign all the data points to either the preceding or following limit cycle segments, and thus some points (those included between the two heavy vertical lines) are not included in either segment.

In addition to the oversensitivity of the firing detection routine, the curve-fitting routine also allows for this “uncertainty of firing time” by not using the end points for curve-fitting purposes. As an example of the difficulties that arise when a firing is not properly detected, consider the limit cycle segment separated by the pitch firing at approximately 12 h, 19 min of day 366 (see Fig. 8). At first glance, it appears that any ambiguous points have been excluded. The first hint of trouble is the fact that the torque (for the pitch axis) as determined by the least-squares fit does not fall within the torque interval determined by interval analysis. (Unless something is amiss, the least-squares torque falls within the torque interval.) Looking further, it can be deduced that a double firing has occurred. The change in angular velocity caused by a single firing is constant. In this case the change is approximately twice this minimum constant. This is easily seen by comparison with a single firing; e.g., the firing on the pitch axis at approximately 10 h, 50 min of the same day (see Fig. 9). The broken vertical line indicates where the limit cycle segment should have ended. The data points between this line and the heavy vertical line on the right should not have been included in the limit cycle segment. Inclusion of these points resulted in the computation of torque levels which were obviously erroneous.

Selection of the firing detection routine was then a trade-off between (1) including as many points as possible in each limit cycle segment so as to get as much information as possible, (2) avoiding cases of erroneous torque levels resulting from including too many points, and (3) simplicity of the routine to keep computation time to a minimum.

Since the character of the disturbance torques is somewhat different for each spacecraft, we consider the two separately.

B. Mariner IV Disturbance Torques

Figures 7–12 pertain to the Mariner IV flight. The first conclusion that can be drawn is that a restoring torque of approximately ¾ dyn-cm/mrad is present on the pitch and yaw axes. That is, a torque proportional to the angular displacement reaches a magnitude of 1 dyn-cm at the edge of the deadband. The presence of this restoring torque is established by considering a long limit cycle...
which is divided into a number of limit cycle segments by firings on the other axes. Consider the pitch limit cycle beginning at around 23 h, 35 min of day 361 (see Fig. 10). Torque levels are computed for four limit cycle segments before the pitch attitude control thrusters fire again at approximately 1 h, 23 min of day 362. The two middle segments have rms torque levels that are about 1 dyn-cm less than those of the end segments. This pattern is consistent on all such long limit cycles, many of which can be observed in the Addendum to this report.

In addition to this restoring torque, there appears to be a bias torque (some part of which is probably a solar bias torque) which changes in what appears to be a random manner from one limit cycle to the next. This change in torque level can be observed in Fig. 7, accompanying the pitch firing at about 8 h, 51 min of day 362, where a change of about 1 dyn-cm is noted. Another example occurs at the pitch firing at 14 h, 58 min on day 362 (see Fig. 11).

Relatively large changes in bias torque over long periods can be noted by comparing the torque levels of Figs. 7–11 with those of Fig. 12, which gives torque levels some 3 months later. The change is most apparent for the pitch axes, where a change of about 15 dyn-cm occurred.

C. Mariner V Disturbance Torques

Figure 13 pertains to Mariner V. In this case, the restoring torque appears to be much smaller than the Mariner IV restoring torque. It is tempting to assert that one can deduce the presence of such a restoring torque from the data. However, if such a restoring torque does exist, it is smaller than the resolution of this data-reduction procedure. Changes in torque level on the order of 1 dyn-cm were noted to accompany some of the valve firings. (See, for example, the pitch valve firings of Fig. 13.) Not enough data were processed for Mariner V to detect long-term changes in torque level such as occurred on Mariner IV.

IV. Conclusions

Mariner IV has a comparatively large (∼25 dyn-cm), slowly varying bias torque (apparently a solar torque) as well as a smaller component which changes when a control valve fires. This change may be as high as 3–5 dyn-cm.

Mariner V is symmetrical about the sun line and hence does not have a large bias solar torque. Since there are no solar vanes, the solar restoring torque is also considerably smaller. However, the disturbance torque, which varies randomly with valve firing (by as much as 2–3 dyn-cm), is present.

References


Fig. 7. Ambiguous data points in the limit cycles, Mariner IV, day 362, 1964. Time is in hours, minutes, and seconds (GMT). Symbols: * = pitch; X = yaw; O = V-axis roll; H = Z-axis or true roll; $ = superimposed data points.
Fig. 7 (contd)
| 8.31 6 | + | + | + | + | + | H | O | X | + | + | + |
| 8.32 6 | | | | | | | | | | | |
| 8.33 13 | + | + | + | + | + | H | O | X | + | + | + |
| 8.34 18 | + | + | + | + | + | H | O | X |
| 8.35 24 | + | + | + | + | + | H | O | X |
| 8.36 30 | + | + | + | + | + | M | O | X |
| 8.37 36 | + | + | + | + | + | H | O | X |
| 8.38 42 | + | + | + | + | + | H | O | X |

**Fig. 7 (contd)**
Fig. 7 (contd)
Fig. 8. Improper detection of attitude control thruster firing, Mariner IV, day 366, 1964. Time is in hours, minutes, and seconds (GMT). Symbols: * = pitch; X = yaw; O = V-axis roll; H = Z-axis or true roll; $ = superimposed data points.
Fig. 8 (contd)
<table>
<thead>
<tr>
<th>PITCH</th>
<th>YAW</th>
<th>ROLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPPER</td>
<td>$-0.88982958E+01$</td>
<td>$-0.39986955E+01$</td>
</tr>
<tr>
<td>LOWER</td>
<td>$-0.93378593E+01$</td>
<td>$-0.45052115E+01$</td>
</tr>
<tr>
<td>RMS ERROR OF PARAB FIT</td>
<td>$0.70858079E-01$</td>
<td>$0.57509277E-01$</td>
</tr>
</tbody>
</table>

Fig. 8 (contd)
Fig. 8 (contd)
Fig. 9. Angular velocity change for a single thruster firing, Mariner IV, day 366, 1964. Time is in hours, minutes, and seconds (GMT). Symbols: * = pitch; X = yaw; O = V-axis roll; H = Z-axis roll; $ = superimposed data points.

TORQUE LIMITS ARE--UPPER -0.94303311E 01 -0.4408254E 01 0.21638473E 02
LOWER -0.10735570E 02 -0.6917290E 01 0.12615590E 02
TORQUE BY LEAST SQUARES -0.10226257E 02 -0.51087741E 01 0.13305245E 02
RMS ERROR OF PARAB FIT 0.26912418E-01 0.39651066E-01 0.45509729E 00
Fig. 9 (contd)
<table>
<thead>
<tr>
<th></th>
<th>PITCH</th>
<th>YAW</th>
<th>ROLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>TORQUE LIMITS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UPPER</td>
<td>-0.80093015E 01</td>
<td>-0.46997963E 01</td>
<td>0.19787596E 02</td>
</tr>
<tr>
<td>LOWER</td>
<td>-0.10292994E 02</td>
<td>-0.60644476E 01</td>
<td>0.12466967E 02</td>
</tr>
<tr>
<td>TORQUE BY LEAST SQUARES</td>
<td>-0.87859795E 01</td>
<td>-0.51049834E 01</td>
<td>0.12010508E 02</td>
</tr>
<tr>
<td>RMS ERROR OF PARAB FIT</td>
<td>0.57123935E-01</td>
<td>0.57293269E-01</td>
<td>0.14765694E 00</td>
</tr>
</tbody>
</table>

Fig. 9 (contd)

24

JPL TECHNICAL REPORT 32-1305
Fig. 10. Evidence of restoring torque, Mariner IV, days 361–362, 1964. Time is in hours, minutes, and seconds (GMT). Symbols: * = pitch; X = yaw; O = V-axis roll; H = Z-axis or true roll; $ = superimposed data points
Fig. 10 (contd)
<table>
<thead>
<tr>
<th></th>
<th>PITCH</th>
<th>YAW</th>
<th>ROLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>TORQUE LIMITS ARE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UPPER</td>
<td>-0.29540440E 01</td>
<td>-0.1902840E 01</td>
<td>0.10822467E 02</td>
</tr>
<tr>
<td>LOWER</td>
<td>-0.34621786E 01</td>
<td>-0.24284536E 01</td>
<td>0.66953079E 01</td>
</tr>
<tr>
<td>RMS ERROR OF PARAB FIT</td>
<td>0.58664983E-01</td>
<td>0.56122658E-01</td>
<td>0.17087475E 00</td>
</tr>
</tbody>
</table>

Fig. 10 (contd)
<table>
<thead>
<tr>
<th>PITCH</th>
<th>YAW</th>
<th>ROLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>TURQUE LIMITS ARE--UPPER</td>
<td>-0.68567010E-01</td>
<td>-0.95117522E 00</td>
</tr>
<tr>
<td>LOWER</td>
<td>-0.31172685E 01</td>
<td>-0.20344956E 01</td>
</tr>
<tr>
<td>TURQUE BY LEAST SQUARES</td>
<td>-0.28755690E 01</td>
<td>-0.19639487E 01</td>
</tr>
<tr>
<td>RMS ERRUR OF PARAB FIT</td>
<td>0.49003376E-01</td>
<td>0.53135793E-01</td>
</tr>
</tbody>
</table>

Fig. 10 (contd)
Fig. 10 (contd)
Fig. 10 (contd)
Fig. 10 (contd)
Fig. 11. Torque change accompanying valve firing, Mariner IV, day 362, 1964. Time is in hours, minutes, and seconds (GMT). Symbols: * = pitch; X = yaw; O = V-axis roll; H = Z-axis or true roll; $ = superimposed data points.
<table>
<thead>
<tr>
<th>PITCH</th>
<th>YAW</th>
<th>ROLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>TORQUE LIMITS ARE--UPPER</td>
<td>-0.1933017E 01</td>
<td>-0.91654321E 00</td>
</tr>
<tr>
<td>LOWER</td>
<td>-0.23650971E 01</td>
<td>-0.11534658E 01</td>
</tr>
<tr>
<td>TORQUE BY LEAST SQUARES</td>
<td>-0.21804808E 01</td>
<td>-0.10590380E 01</td>
</tr>
<tr>
<td>RMS ERROR OF PARAB FIT</td>
<td>0.16469255E-01</td>
<td>0.64986542E-01</td>
</tr>
</tbody>
</table>

Fig. 11 (contd)
Fig. 11 (contd)
Fig. 12. Limit cycles at the low bit rate of 8½ bits/s, Mariner IV, day 105, 1965. Time is in hours, minutes, and seconds (GMT). Symbols: *= pitch; X = yaw; O = V-axis roll; H = Z-axis or true roll; $ = superimposed data points
<table>
<thead>
<tr>
<th>PITCH</th>
<th>YAH</th>
<th>PCLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>TURRENT LIMITS ARE--LIPPER</td>
<td>-0.2390973E 02</td>
<td>0.2745252E 02</td>
</tr>
<tr>
<td>LIPPER</td>
<td>-0.223595E 02</td>
<td>0.2764545E 02</td>
</tr>
<tr>
<td>TURRENT BY LEAST SQUARES</td>
<td>-0.2536252E 02</td>
<td>0.2714259E 02</td>
</tr>
<tr>
<td>RPS ERRCHR CF PARAE FIT</td>
<td>C.5118661E-01</td>
<td>C.2012158E-01</td>
</tr>
</tbody>
</table>

Fig. 12 (contd)
Fig. 12 (contd)
Fig. 13. Torque characteristics, Mariner V, day 183, 1967. Time is in hours, minutes, and seconds (GMT). Symbols: * = pitch; X = yaw; O = V-axis roll; H = Z-axis or true roll; $ = superimposed data points.
Fig. 13 (contd)
Fig. 13 (contd)
Appendix A
Fitting a Parabola by Interval Analysis

The problem can be stated as follows: given a set of \( N \) discrete values of a variable (in this case, time) and an interval (angular displacement) associated with each, determine the interval parabola

\[
\tilde{\theta} = \tilde{a} t^2 + \tilde{b} t + \tilde{c}
\]  

(A-1)

which passes through these intervals. Note that the interval parabola is a third-order infinity of parabolas, every one of which passes through the \( N \) intervals.

The interval parabola

\[
\tilde{\theta} = \tilde{x} t^2 + \tilde{y} t + \tilde{z}
\]  

(A-2)

which passes through the intervals \((\theta_i, \theta_j, \theta_k)\) associated with any three of the \( N \) values of the time \((t_i, t_j, t_k)\) is determined by the following equations:

\[
\begin{align*}
\tilde{x} t_i^2 + \tilde{y} t_i + \tilde{z} &= \tilde{\theta}_i \\
\tilde{x} t_j^2 + \tilde{y} t_j + \tilde{z} &= \tilde{\theta}_j \\
\tilde{x} t_k^2 + \tilde{y} t_k + \tilde{z} &= \tilde{\theta}_k
\end{align*}
\]  

(A-3)

where

\[ t_i, t_j, t_k = 1, N \]

\[ i \neq j \neq k \]

That is,

\[
\begin{bmatrix}
\tilde{x} \\
\tilde{y} \\
\tilde{z}
\end{bmatrix} = 
\begin{bmatrix}
t_i^2 & t_i & 1 \\
t_j^2 & t_j & 1 \\
t_k^2 & t_k & 1
\end{bmatrix}^{-1}
\begin{bmatrix}
\tilde{\theta}_i \\
\tilde{\theta}_j \\
\tilde{\theta}_k
\end{bmatrix}
\]  

(A-4)

This equation is solved for every combination of \( N \) sample points taken three at a time. The intersection of all solutions yields \( \tilde{a}, \tilde{b}, \tilde{c} \). If there is no parabola which passes through all of these intervals, then the intersection is empty.

As \( N \) increases, the number of equations (of the type A-3) which must be solved increases rapidly:

\[
\text{Number of equations} = \binom{N}{3} = \frac{N!}{(N - 3)!3!}
\]  

(A-5)

Therefore, it was necessary to judiciously select combinations of sample points to be used. Since sample points with the largest separation tend to give the most information about the curvature (the torque level is determined from the curvature), points used in Eq. (A-4) were selected, one each from three groups of points: one centered near the middle of the limit cycle segment and one near each end. For data at 8 % (33 %) bits/s, a total of 11 (26) points are included in the three groups; 5 (12) points in the middle group and 3 (7) points in the others. Hence, the interval parabola passes through 26 (or 11 if the bit rate is 8 % bits/s) judiciously selected intervals rather than all \( N \) intervals. The improvement obtained by using all \( N \) points did not justify the increase in computation time.
Appendix B
Computer Programs

I. Discussion
The Lister program and the Data Reduction program are discussed and reproduced in this Appendix. Both require a plotting routine. The routine used with the programs in this study was JPL T3.

A. Lister Program
The Lister program consists of a main program and ten subroutines that (1) read MDL tapes to select attitude control data, (2) output these data in the form of printer plots of angular position vs time, and (3) store the appropriate data on tape for use in the data-reduction program.

A description of the main program and the subroutines follows.

1. Main program—$IBFTC ACL. This is essentially an "indexing" program which calls the appropriate subroutines to read one data record of MDL tape at a time and then output the data.

2. Subroutines. The subroutines are as follows:
   (1) $IBMAP SPLT—Splits data word into appropriate data bits.
   (2) $IBMAP CLOK2—DSL/90 simulator clock.
   (3) $IBMAP TIMR—Time converter.
   (4) $IBMAP RDR—MDL tape reader.
   (5) $IBFTC DECOM—Decommutation of MDL data.
   (6) $IBFTC MRVSDN—Canopus measurement correction.
   (7) $IBFTC ANCLE—Block data used by MRVSDN.
   (8) $IBMAP PRPLT—Printer plot file.
   (9) $IBFTC JPLT—Plot routine.
   (10) $IBMAP URPLT4—Plotter.

Subroutines 5, 6, and 7 are specialized for Mariner IV and Mariner V. That is, a slight variation of the same program is needed to account for variations in the parameters of the two missions.

This program has the following capability:
   (1) It provides printer plots which approximate the telemetry quantization level.
   (2) It reads only the angular displacement channels from the MDL tape: two sun sensors and the Canopus sensor.
   (3) By mathematically deducing true roll from the three angular position measurements, it corrects for the fact that the Canopus sensor does not measure true roll.
   (4) It writes the appropriate angular displacement data on tape. This tape serves as an input to the Mariner data reduction program.

B. Data Reduction Program
The Data Reduction program consists of a main program and 18 subroutines that (1) detect attitude control thruster firings and, in so doing, divide the angular motion data into limit cycle segments; (2) fit parabolas to these segments by the least-squares and interval analysis methods; (3) compute the torque levels using these curve fits, and (4) compute the rate increments induced by the firings when this is possible.

A description of the main program and the subroutines follows.

1. Main program—$IBFTC MARDAT. This program reads the tape input produced by the Lister program, calls the appropriate subroutines to analyze the data, and outputs the torque level and minimum rate increment data.

2. Subroutines. The subroutines are as follows:
   (1) $IBFTC PARINT—Interval analysis parabola fit.
   (2) $IBFTC MAXVT—Multiplies a $3 \times 3$ matrix by a $3 \times 1$ interval vector.
   (3) $IBFTC IDMT—Multiplies an interval by a scalar constant.
   (4) $IBFTC ISUBTR—Subtracts two intervals.
   (5) $IBFTC IMULTP—Multiplies two intervals.
(6) $IBFTC$ INTX—Finds the intersection of two intervals.

(7) $IBFTC$ IDA—Adds two intervals.

(8) $IBFTC$ PARF—Least-squares fits a parabola.

(9) $IBFTC$ DECT—Detects thruster firings.

(10) $IBFTC$ PAT—Used by $IBFTC$ DECT; looks for data point patterns typical of thruster firings.

(11) $IBFTC$ MATT—Inverts a $3 \times 3$ matrix.

(12) $IBFTC$ CHECK—Checks for outages and bit errors in the data.

(13) $IBFTC$ TRANSF—Multiplies a $3 \times 3$ matrix by a $3 \times 1$ matrix (linear transformation of a vector).

(14) $IBFTC$ RELT—Adds time in days, hours, minutes, seconds format.

(15) $IBFTC$ MNRT—Computes the minimum rate increment.

(16) $IBFTC$ MRVSDN—Calibrates the position sensor.

(17) $IBFTC$ IANG—Computes angular position intervals from the data number.

(18) $IBFTC$ CANCOR—Corrects the Canopus measurement.

(19) $IBFTC$ ANGLE—Block data for CANCOR and IANG.

(20) $IBMAP$ PRPLT—Printer plot file.

(21) $IBFTC$ JPLT—Plot routine.

(22) $IBMAP$ URPLT4—Plotter.

Subroutines 16–19 are specialized for Mariner IV and Mariner V.
II. Lister Program

*IBFC ACL
C
C MARINER ATTITUDE CONTROL LISTER
C
001 DIMENSION IA(500), K(5)
C
002 DIMENSION TH(3)
003 DATA IFHMH, IEOF, IFND/6H100003, 6H100004, 6H100007, 6H100008, 6H400000/
004 DATA (K(1), 1=1, 5) / 6H200005, 6H200006, 6H200007, 6H200008, 6H400000/
005 INTEGER GYRO, VDN(3)
006 REAL V(4)
007 NDX=0
008 KC=0
C
C INPUT DATA TIME=RUNNING TIME
C IDAY=OUTPUT OF DATA STARTS AT BEGINNING OF THIS DAY
C
012 NAMELIST / CONTRL/ TIME, IDAY
013 CALL CLOCK(TL)
014 READ (5, CONTRL)
020 IQ=0
021 CALL START9
022 CALL OR
23 GYRO=2
024 J=0
025 J=J+1
026 IF(INDEX, EQ, 0) WRITE(6, 126) J
027 DO 117 II=1, 4
C
C READ ONE RECORD OF MDL TAPE
C
028 CALL READER(IA, N)
029 IF(N) 40, 40, 30
C
C LOOK FOR DATA RECORD—321 WORDS
C
030 IF(IA(2) EQ IFH OR IA(2) EQ MH) GO TO 28
031 IF(IA(2) NE IEOF) GO TO 35
032 GO TO 28
035 IF(IA(2) EQ IFND) GO TO 28
036 DO 38 I=1, 5
037 J0=I
038 IF(IA(2) EQ K(I)) GO TO 42
039 CONTINUE
040 CALL CR
041 STOP
042 IF(J0 EQ 4) GO TO 28
042 IF(NDFX, NE, 0) GO TO 54
043 CALL TIMER(IA(3), ID, IHR, IM, IS)
044 IF(ID, LT, IDAY) GO TO 28
05101 WRITE (6, 128) ID, IHR, IM, IS
052 WRITE (6, 129)
C
C FIND SYNC
C
054 CALL DECOM(IA, VDN, IS0, V30, OK, I2Z, V22, GYRO, ID, IHR, IM, IS)
062 DO 117 JJ=1, 10
C
C FIND AND STORE DATA FROM DESIRED CHANNELS
C
063 CALL GETIM(IA, VDN, IS0, V30, OK, I2Z, V22, GYRO, ID, IHR, IM, IS)
06300 IF(VDN(1) .LT. 1 .AND. VDN(2) .LT. 1 .AND. VDN(3) .LT. 1) GO TO 117
06301 DO 6302 JJ=1,3
06302 TH(JJ)=VDN(JJ)
06305 CALL ANGPOS(TH, V9ID)
06308 IF( NOTOK0) GO TO 80
078 IF(I30.EQ.0) GO TO 80
079 CAN=V30
080 IF(NDEX .NE. 0) GO TO 102
084 DATA OUTPUT
101 WRITE(6,130) ID, IHR, IM, IS, V(1), V(2), V(3), V(4), CAN
102 WRITE(17) ID, IHR, IM, IS, (VDN(KK), KK=1,3), CAN
103 IF(IQ.EQ.0) WRITE(9,137) ID
104 CALL JPLTS(12,5,-12,5,0,0,4, IQ, V, IHR, IM, IS)
105 IQ=1
117 CONTINUE
118 KC=KC+1
119 IF(KC .NE. 30) GO TO 118
120 NDFX=2
121 CALL CLOCK(T)
122 CALL CR
123 WRITE(6,138) ID, IHR, IM, IS, V(1), V(2), V(3), CAN
124 END FILE 17
125 STOP
126 FORMAT(1H1, 25X, 46HATTITUDE CONTROL MARINER ROL LISTER -- PAGE, S I4)
127 FORMAT(4I5)
128 FORMAT(4/5X, 3HDAY, I4, 5H HOUR, I3, 4H MIN, I3, 4H SEC, I3)
129 FORMAT(10X,6HHR, 5H MIN, SEC, PITCH YAW ROLL TRU
130 0 ROLL CAN CONF)
131 FORMAT(10X, 13, 2X, I3, 4X, 13, 3X, I3, 2X, 3F7.2, 4X, F7.2)
132 FORMAT(1H1/)
133 FORMAT(80I5)
134 FORMAT(1H1, 15X, 51H PLOT OF MARINER PITCH YAW AND ROLL CHANNELS IN M
135 $RAD, 5X, 12H STARTING DAY, 14//
136 $25X, 9H* = PITCH //25X, 7H* = YAW //25X, 8H* = ROLL //
137 $25X, 18H* = CORRECTED ROLL //)
138 FORMAT(10X, 2OH LAST DATA ON TAPE 15, 3X, 5X, 4F7.2)
139 END
140 $IRMAP SPLIT
141 * CALL SPLIT(WORD, DNI, DNI2)
142 * SPLIT
143 ENTRY SPLIT
144 SXA END, 4
145 CLA 4, 4
146 STA SXA1
147 CLA 5, 4
148 STA SXA2
149 CAL* 3, 4
150 LRS 18
151 ANA MASK
152 SLW* 4, 4
153 LLS 18
154 ANA MASK
155 SLW* 5, 4
156 CAL* 3, 4
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LXA</td>
<td>PZE*4</td>
</tr>
<tr>
<td>COM</td>
<td></td>
</tr>
<tr>
<td>PBT</td>
<td></td>
</tr>
<tr>
<td>SXA1</td>
<td>SXA **,4</td>
</tr>
<tr>
<td>LRS</td>
<td>17</td>
</tr>
<tr>
<td>SXA2</td>
<td>SXA **,4</td>
</tr>
<tr>
<td>LBT</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td>AXT **,4</td>
</tr>
<tr>
<td>TRA</td>
<td>1,4</td>
</tr>
<tr>
<td>PZE</td>
<td>PZE 333</td>
</tr>
<tr>
<td>MASK</td>
<td>OCT 080000000177</td>
</tr>
<tr>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>

`$LIBMAP CLOK2
`*

** DSL/90 SIMULATOR CLOCK**

** ENTRY CLOCK
** ENTRY CLOK3
** CLOCK ZET FLAG
** TRA ZERO
** CAL 5
** ORA EXP
** FAD ZIP
** XCA FHP SCALE
** STO* 3,4
** TRA 1,4
** ZERO STZ* 3,4
** STZ 5
** STZ FLAG
** TRA 1,4

** CLOK3 ZAC NBT FLAG
** TRA OK
** STO FLAG
** STO 5
** LDO 5
** MPY FX100
** DVH FX6
** XCA SUB* 3,4
** CHS
** TRA 1,4
** FX100 OCT 000000000144
** FX6 OCT 000000000006
** FLAG DEC 1,0
** ZIP DEC 0,0
** EXP OCT 233000000000
** SCALE DEC 16.666667

** END

`$LIBMAP TIMR
`*

** ENTRY TIMER
** TIMER ZAC CALL TIMER(TIME,IDAY,IMIN,ISEC)
** LDO* 3,4
** DVH DAY
** XCA
** ADD =1
** STO* 4,4
<table>
<thead>
<tr>
<th>ZAC</th>
<th>DVH</th>
<th>HOUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>STO*</td>
<td>5,4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LRS</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>DVH</td>
<td>MIN</td>
</tr>
<tr>
<td>STO*</td>
<td>6,4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LRS</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>DVH</td>
<td>SFC</td>
</tr>
<tr>
<td>STO*</td>
<td>7,4</td>
<td></td>
</tr>
<tr>
<td>DAY</td>
<td>TPA</td>
<td>1,4</td>
</tr>
<tr>
<td>DEC</td>
<td>27668800</td>
<td></td>
</tr>
<tr>
<td>HOUK</td>
<td>DFC</td>
<td>115200</td>
</tr>
<tr>
<td>MIN</td>
<td>DFC</td>
<td>1920</td>
</tr>
<tr>
<td>SEC</td>
<td>DFC</td>
<td>32</td>
</tr>
<tr>
<td>END</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*MAP RDR- **MLD TAPE READER (SETUP ON CK1)
*INFILE FILE MLD, INPUT, CK1, INPUT, BLOCK=330, MBCD
*
<table>
<thead>
<tr>
<th>ENTRY</th>
<th>READER</th>
<th>CALL READER(AREA, NOWDS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENTRY</td>
<td>OR</td>
<td>OPN REWD</td>
</tr>
<tr>
<td>ENTRY</td>
<td>QNR</td>
<td>OPN NO REWD</td>
</tr>
<tr>
<td>ENTRY</td>
<td>CR</td>
<td>CLSE REWD</td>
</tr>
<tr>
<td>ENTRY</td>
<td>CNR</td>
<td>CLSE NO REWD</td>
</tr>
</tbody>
</table>

** OR SXA EOR,4
** TSX *OPEN,4
** PZE INFILE
FOR AXT **,4
TRA 1,4
ONR SXA EONR,4
TSX *OPEN,4
EONR AXT **,4
TRA 1,4
CR SXA ECR,4
TSX *CLOSE,4
PTW INFILE
ECR AXT **,4
TRA 1,4
CNR SXA ECNR,4
TSX *CLOSE,4
ECR AXT **,4
TRA 1,4
READ SXA END,4
CALL READER(AREA, NOWDS)
CLA 3,4
STA READ,3
READ TSX *READ,4
PZE INFILE, FOR
PZE EOF, ERR
IORT **, **
LXD **1,4
PXA 0,4
FND AXT **,4
STO* 4,4
TRA 1,4
EOB HR 0
EOF LXA END,4
CLA = -1
13 CAI
14 I  
15 PZJ  
16 P2F  
17 LXA  
18 CLA  
19 STO*  
20 TRA  
21 ERMG  
22 BCI  
23 END  
24
25 THIS IS FOR MM-64  
26 DECOMMIT MAR arsen MDL DATA--CHANNELS 105,106, AND 114  
27
28 SURROGUE DFCOM(RUFI,V,K30,V30,OK,K22,V22,GYRO,DAY,HR,MN,SE)  
29 LOGICAL OK  
30 INTEGER DN1,DN2,GYRO,V(3)  
31 DIMENSION RUI(330),LP(40),LOPOS(3,40)  
32 HIGH RATE  
33 OK=TRUE  
34 CALL TIMER(BUFI,1,DAY,HR,MN,SE)  
35 FIND DK200 SYNC  
36
37 KS=7  
38 DO 15 I=1,10  
39 KS=KS+32  
40 N=1  
41 CALL SPLIT(BUI(KS),DN1,DN2)  
42 IF(DN1 .EQ. 127 .AND. DN2 .EQ. 127) GO TO 18  
43 CONTINUE  
44 OK=FALSE  
45 CONTINUE  
46 KC=7  
47 RETURN  
48
49 SYNC MEDIUM,LOW, AND LOW-LOW  
50
51 IF(KS .EQ. 313) GO TO 81  
5101 CALL SPLIT(BUI(KS+32),DN1,DN2)  
52 LOW POSITION INDICATOR  
53
54 DATA (LP(J),J=1,40)/0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,  
55 24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,63/  
56 DO 23 I=1,40  
57 ILP=I  
58 IF(DN2 .EQ. LP(I)) GO TO 28  
59 CONTINUE  
60 ILP=2  
61 GO TO 28  
62 DATA (LOPOS(I),I=1,120)/  
64 429,440,300,430,401,301,421,402,302,422,403,303,423,404,304,424,  
66 400,300,420,411,301,431,412,302,432,413,303,433,414,304,434,

28 KC=7
29 ILP=ILP-1
2901 IF(ILP .EQ. 0) ILP=40
30 I20=210-N-1
31 I21=220-N-1
3101 I22=230-N-1
32 ICNTR=0
3201 RETURN
3202 ENTRY GETTM(BUF,V,K30,V30,OK,K22,V22,GYRO,DAY,HR,MIN,SEC)
3203 ICNTR=ICNTR+1
3204 IF(ICNTR .GT. 10) RETURN
3205 K20=0
3206 K30=0
3210 K22=0
33 I20=I20+1
3401 I22=I22+1
35 KC=KC+32
36 IF(KC .NE. KS) GO TO 40
37 I20=200
3801 I22=220
39 ILP=ILP+1
3901 IF(ILP .EQ. 41) ILP=1

C

CHANNELS 105,106 AND 114

40 CALL SPLIT(RUF(KC+2),DN1,DN2)
41 V1=DN2
42 CALL SPLIT(RUF(KC+3),DN1,DN2)
43 V2=DN1
44 CALL SPLIT(BUF(KC+7),DN1,DN2)
45 V3=DN1
46 IF(I22 .NE. 221) GO TO 63
4901 CALL SPLIT(BUF(KC+5),DN1,DN2)
4902 K22=I22
4903 V22=DN1
493 IF(I20 .NE. 202) GO TO 79
494 I30=LOPOS(2,ILP)
495 IF(I30 .NE. 300) GO TO 79
49601 CALL SPLIT(RUF(KC),DN1,DN2)

XX=DN2
II=XX/12.-0.5
V3N=I1
48 K3N=130
79 CALL TIMER(RUF(KC=22),DAY,HR,MIN,SEC)
80 RETURN

C

LAST FRAME HAS DK 200 SYNC

C

81 CALL SPLIT(RUF(25),DN1,DN2)
82 DO 85 I=1,40
83 ILP=I+1
84 IF(DN2 .EQ. LP[I]) GO TO 28
85 CONTINUE
86 GO TO 24
END

$TRFTC DECOM-
THIS IS FOR MV-67

DECOMMUTATE MDL DATA—CHANNELS 103, 104, AND 114

SURROUNTE DECOM(BUS, V, K30, V30, OK, K22, V22, GYRO, DAY, HR, MIN, SEC)

LOGICAL OK

INTEGER DN1, DN2, GYRO, V(3)

DIMENSION BUF(330), LP(40), LPOSO, (3, 40)

HIGH RATE

OK = TRUE

CALL TIMER(BUF(3), DAY, HR, MIN, SEC)

FIND DK200 SYNC

KS = 7

DO 15 I = 1, 10

N = I - 1

CALL SPLIT(BUF(KS), DN1, DN2)

IF(DN1 .EQ. 127 .AND. DN2 .EQ. 127) GO TO 18

CONTINUE

OK = FALSE

ICNTR = 0

KC = 7

RETURN

SYNC MEDIUM, LOW, AND LOW-LOW

IF(KS .EQ. 313) GO TO 81

CALL SPLIT(BUF(KS + 32), DN1, DN2)

LOW POSITION INDICATOR

DATA (LP(J), J = 1, 401)/0, 1, 2, 3, 4, 5, 6, 7, 8, 15, 16, 17, 18, 19, 20, 21, 22, 23,

$ 24, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 63/

DO 23 I = 1, 40

ILP = I

IF(DN2 .EQ. LP(I)) GO TO 28

CONTINUE

ILP = 2

GO TO 28

DATA (LPOSO, I, 1, I = 1, 120)/

$ 411, 301, 343, 412, 302, 432, 413, 303, 433, 414,
$ 304, 436, 415, 375, 416, 376, 436, 417, 307, 437, 418, 308, 438, 419, 309,
$ 439, 410, 300, 430, 401, 301, 421, 402, 302, 422, 403, 303, 423, 404, 304, 424,
$ 400, 300, 420, 411, 301, 431, 412, 302, 432, 413, 303, 433, 414, 304, 434,
$ 410, 300, 430, 401, 301, 421, 402, 302, 422, 403, 303, 423, 404, 304, 424,
$ 400, 300, 420/

KC = 7

ILP = ILP - 1

IF(ILP .EQ. 0) ILP = 40

ICNTR = 0

RETURN

ENTRY GETIM(BUF, V, K30, V30, OK, K22, V22, GYRO, DAY, HR, MIN, SEC)
3203  ICNTR=ICNTR+1
3204  IF(ICNTR .GT. 10) RETURN
3205  K2=0
3206  K3=0
3207  K22=0
3210  I2o=I20+1
3401  I22=I22+1
35  KC=KC+32
36  IF(KC .NE. KS) GO TO 40
37  I20=200
3801  I22=220
39  ILP=ILP+1
3901  IF(ILP .EQ.41) ILP=1
C
C CHANNELS 103,104,AND 114
C
40  CALL SPLIT(BUF(KC+1),DN1,DN2)
41  V(1)=DN2
42  CALL SPLIT(BUF(KC+2),DN1,DN2)
43  V(2)=DN1
44  CALL SPLIT(BUF(KC+7),DN1,DN2)
45  V(3)=DN1
46  IF(I22 .NE.22) GO TO 63
4901  CALL SPLIT(BUF(KC+5),DN1,DN2)
4902  K22=122
4903  K27=DN1
49  IF(I20 .NE.200) GO TO 79
64  I3n=LOPOS(2,ILP)
65  IF(I30 .NE.300) GO TO 79
6601  CALL SPLIT(BUF(KC),DN1,DN2)
66  XX=DN2
67  II=XX/12.-0.5
68  V3n=II
69  KS=130
70  CALL TIMER(BUF(KC-2),DAY,HR,MIN,SEC)
80  RETURN
C
C LAST FRAME HAS DK 200 SYNC
C
81  CALL SPLIT(BUF(25),DN1,DN2)
82  DO 85 I=1,40
83  ILP=I+1
84  IF(DN2 .EQ. LP(I)) GO TO 28
85  CONTINUE
86  GO TO 24
END
SIRFC CANBUS
C
C CANBUS MEASUREMENT CORRECTION - FOR MM-64
C
SUBROUTINE ANGPOS(B,C,K)
C
C THIS IS FOR MM-64
C
YOU ALSO NEED SUBPROGRAM BLOCK DATA
C
C
SUBROUTINE ANGPOS HAS 3 ARGUMENTS
C
INPUT B,K
C
B(1) = PITCH MEASUREMENT
C
B(2) = YAW MEASUREMENT
C
B(3) = ROLL MEASUREMENT

JPL TECHNICAL REPORT 32-1305
C
C K = TIME
C
C(1) = PITCH IN MILLI-RADIANS
C(2) = YAW IN MILLI-RADIANS
C(3) = ROLL IN MILLI-RADIANS
C(4) = CORRECTED ROLL IN MILLI-RADIANS

DIMENSION B(3), C(4)
COMMON/COEFF/AA(4,3)
COMMON/CANOP1/CLOCK/CANOP2/CONE(275)

CLOCK = X-AXIS CLOCK ANGLE IN DEGREES
CONE(K) = CANOPUS CONE ANGLE IN DEGREES

IF (K*GT.330) J=K-331
IF (K*LT.330) J=K+36
THETA1=CLOCK*.17453293E-01
THETA2=CONE(J)*.17453293E-01

CALIBRATION OF ANGULAR POSITION FROM DATA NUMBER
C(3) = EAA(1,3) + AA(2,3)*B(3) + AA(3,3)*B(3)^2 + AA(4,3)*B(3)^3
C(4) = (C(3) - COS(THETA2) * (C(1) + C(2) * SIN(THETA1))) / SIN(THETA2)
RETURN
END

$IBFTC CANPUS
C CANPUS MEASUREMENT CORRECTION - FOR MV-67
C
SUBROUTINE ANGPOS(B,C,K)
C
THIS IS FOR MV-67
C
YOU ALSO NEED SUBPROGRAM BLOCK DATA
C

SUBROUTINE ANGPOS HAS 3 ARGUMENTS
C
INPUT
B,K
B(1) = PITCH MEASUREMENT
B(2) = YAW MEASUREMENT
B(3) = ROLL MEASUREMENT
C
K = TIME
C
OUTPUT
C
C(1) = PITCH IN MILLI-RADIANS
C(2) = YAW IN MILLI-RADIANS
C(3) = ROLL IN MILLI-RADIANS
C(4) = CORRECTED ROLL IN MILLI-RADIANS

DIMENSION B(3), C(4)
COMMON/COEFF/AA(6,3)
COMMON/CANOP1/CLOCK/CANOP2/CONE(135)

CLOCK = X-AXIS CLOCK ANGLE IN DEGREES
CONE(K) = CANOPUS CONE ANGLE IN DEGREES

J=K-164
THETA1=CLOCK*.17453293E-01
THETA2=CONE(J)*.17453293E-01
CALIBRATION OF ANGULAR POSITION FROM DATA NUMBER

\[
C(1) = (A(1,1) + A(2,1)*R(1)) + A(3,1)*R(1) + A(4,1)*B(1)**3
\]
\[
1 + A(5,1)*R(1)**4 + A(6,1)*B(1)**5
\]
\[
C(2) = (A(1,2) + A(2,2)*R(2)) + A(3,2)*R(2)**2 + A(4,2)*B(2)**3
\]
\[
1 + A(5,2)*R(2)**4 + A(6,2)*B(2)**5
\]
\[
C(3) = (A(1,3) + A(2,3)*R(3)) + A(3,3)*R(3)**2 + A(4,3)*B(3)**3
\]
\[
1 + A(5,3)*R(3)**4 + A(6,3)*B(3)**5
\]
\[
C(4) = (C(3) - \cos(\Theta_2) \cdot -(C(1) \cdot \cos(\Theta_1) + C(2) \cdot \sin(\Theta_1)))
\]
\[
\text{return}
\]

CIRCF CT ANGCE

C

THIS IS FOR MM-64

C

01 BLOCK DATA

02 COMMON/CANON1/CLOCK

03 COMMON/CANON2/CONE(275)

04 COMMON/COEFF/A(4,3)

05 DATA CLOCK/#5602/

06 DATA (CONE(I),I=1,75)/

07 1 101.80, 101.95, 102.09, 102.23, 102.36

08 2 102.49, 102.61, 102.72, 102.83, 102.92

09 3 103.03, 103.13, 103.21, 103.30, 103.78

10 4 103.45, 103.52, 103.58, 103.64, 103.70

11 5 103.75, 103.79, 103.83, 103.87, 103.90

12 6 103.93, 103.95, 103.97, 103.98, 103.99

13 7 104.00, 104.00, 104.00, 104.00, 103.99

14 8 103.97, 103.96, 103.94, 103.91, 103.89

15 9 103.85, 103.82, 103.78, 103.74, 103.70

16 1 103.65, 103.60, 103.55, 103.49, 103.43

17 2 103.37, 103.30, 103.24, 103.16, 103.09

18 3 103.01, 102.94, 102.86, 102.77, 102.69

19 4 102.60, 102.51, 102.42, 102.33, 102.24

20 5 102.15, 102.05, 101.95, 101.85, 101.74


22 DATA (CONE(I),I=76,150)/

23 7 101.09, 100.97, 100.85, 100.74, 100.62

24 8 100.49, 100.37, 100.25, 100.12, 100.00

25 9 99.86, 99.73, 99.60, 99.47, 99.34

26 1 99.20, 99.06, 98.92, 98.78, 98.64

27 2 98.50, 98.35, 98.21, 98.07, 97.92

28 3 97.78, 97.63, 97.48, 97.33, 97.19

29 4 97.04, 96.89, 96.74, 96.59, 96.44

30 5 96.21, 96.14, 96.04, 95.95, 95.83

31 6 95.54, 95.39, 95.25, 95.10, 94.95

32 7 94.80, 94.66, 94.52, 94.38, 94.24

33 8 94.10, 93.97, 93.83, 93.69, 93.56

34 9 93.42, 93.28, 93.15, 93.02, 92.88

35 1 92.75, 92.62, 92.48, 92.35, 92.22

36 2 92.09, 91.96, 91.83, 91.70, 91.57

37 3 91.44, 91.31, 91.18, 91.05, 90.93

38 DATA (CONE(I),I=151,225)/

39 4 90.80, 90.68, 90.55, 90.43, 90.30

40 5 90.18, 90.05, 89.93, 89.81, 89.69

41 6 89.57, 89.44, 89.32, 89.20, 89.08

42 7 88.96, 88.84, 88.72, 88.60, 88.49

43 8 88.37, 88.25, 88.13, 88.02, 87.90

44 9 87.78, 87.66, 87.55, 87.43, 87.32

45 1 87.20, 87.08, 86.97, 86.85, 86.74

JPL TECHNICAL REPORT 32-1305

65
| DATA (cone(1), I = 2269275) |
|---|---|---|---|---|---|---|---|
| 46 | 2 | 86.62 | 86.50 | 86.39 | 86.27 | 86.16 | ANGLE4 |
| 47 | 3 | 86.04 | 85.93 | 85.82 | 85.70 | 85.59 | ANGLE4 |
| 48 | 4 | 85.48 | 85.36 | 85.30 | 85.14 | 85.02 | ANGLE4 |
| 49 | 5 | 84.91 | 84.80 | 84.69 | 84.58 | 84.47 | ANGLE4 |
| 50 | 6 | 84.35 | 84.24 | 84.13 | 84.02 | 83.91 | ANGLE4 |
| 51 | 7 | 83.83 | 83.69 | 83.58 | 83.47 | 83.36 | ANGLE4 |
| 52 | 8 | 83.25 | 83.14 | 83.04 | 82.93 | 82.82 | ANGLE4 |
| 53 | 9 | 82.71 | 82.60 | 82.50 | 82.39 | 82.28 | ANGLE4 |

...
35  R  9956F02,      9983E02,      10010E03,      10037E03,      ANGLE7
36  R  10064F03,      10090E03,      10116E03,      10141E03,      ANGLE7
37  R  10165F03,      10188E03,      10210E03,      10231E03,      ANGLE7
38  R  10252E03/      ANGLE7
39   DATA(A(I),I=1,18)/25.326053,-.92093034,.19287685E-01,.28223686E-01/. ANGLE7
40  $-03,.01335972E-05,.66782966E-08,.046241,-.10178037,.20517513E-01/. ANGLE7
41  $01,-.28572125E-03,.20792001E-05,.63928052E-08,.144102,.3237947/. ANGLE7
42  $1,-.1165038E-01,.34020915E-03,.33585269E-03,.10896906E-07/. ANGLE7
43  END/ ANGLE7
III. Data Reduction Program

```plaintext
SIBFTC MARDAT
C
C MARINER DATA REDUCTION
C
001 DIMENSION XJ(3*3),T(3),Q(3),W(3),TAN(3),W3(3),WP(3)
002 COMMON ART(7),S061,CKLIMP61,DEL
003 COMMON LVTS,LEVOUT
004 INTEGER K,OK,ERED,BEGIN
005 READ(5,112)DNL,NB,NA,(CKLIM(I),I=1,6)
006 READ(5,113)(XJ(I,1),I=1,9)
007 DNL=DEL/6.0
008 MZ1
009 KDK=1
010 KDY=NB-6
011 IF(DNL.GT.25.0)KDY=NB-4
012 L0=4
013 IF(DNL.GT.25.0)L0=2
014 LEVOUT=4
015 IF(DNL.GT.25.0)LEVOUT=2
016 I1OUT=1
017 CALL START5
018 INX=7
019 RERED=.FALSE.
020 OK=.TRUE.
021 BEGIN=.TRUE.
022 START=.TRUE.
023 JSART=4
024 I1=4
025 JJ=303
026 KLE=0
027 IF(JJ.LE.49.0)KJ=II,JJ
028 READ(17)ID,IM,IS,(TH(I),I=1,3),CAN
029 AR(I1,I)=TH(1)
030 AR(I2,I)=TH(2)
031 AR(I3,I)=TH(3)
032 AR(I4,I)=ID
033 AR(I5,I)=IM
034 AR(I6,I)=IS
035 IF(I1.EQ.6)GO TO 27
036 IF(I1.EQ.9)GO TO 31
037 GO TO 34
038 DO 29 L=1,3
039 DO 29 LI=1,3
040 AR(L1,LI)=AR(L,LI+3)
041 CONTINUE
042 GO TO 34
043 DO 31 L=1,7
044 DO 33 L=1,3
045 AR(L1,LI)=AR(L,LI+300)
046 CONTINUE
047 IF(.NOT.START)GO TO 43
048 IF(KL.EQ.410)GO TO 43
049 START=.FALSE.
050 DO 47 LI=1,3
051 END
```
C  DATA STORED IN AR==NOW LOOK FOR LIMIT CYCLES
C
049 INXPREF=INX
IF(INXPREF.EQ.1)JSART=JSART+1
IF(INXPREF.EQ.3)JSART=JSART+2
CALL DETECT(JSART,END,N,INX,NAXIS)
050 WRITE(6,100)
051 WRITE(6,100)
IF(INX.EQ.1.OR.LEV(MI,EQ.1)) GO TO 5101
IF(INX.EQ.0.AND.NAXIS.EQ.EQ.1) GO TO 5102
05101 IAX=I
GO TO 5103
05102 MZ=1
NJ= NJ+ N-1
05103 MDX=KDX
KDX=KDX+N-1
IF(KDX.LT.MP) GO TO 30
KDX=KDX-NB-1
TORG(11)=60.
TORG(7)=60.
TORG(1)=60.
MDX=NR-1-MDX
DO 5109 LI=1N
IF(LI.NE.MD) GO TO 5109
5109 CONTINUE
MJE=JSART+LI-1
IF(MJ,GT,300)MJE=MJ-300
JD=ART4(MJ)
JH=ART5(MJ)
JM=ART6(MJ)
JS=ART7(MJ)
IF(JG.EQ.0)WRITE(19,107)JD
CALL JPLT3(25,A,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
1=1
05109 CONTINUE
052 GO TO 30
053 CALL PRIM(N,JSART,DELT,A,B,C,INX,NA)
IF(INX.EQ.NM)GO TO 54
WRITE(6,114)INX
054 NM=1
N0=N-4
IF(DELT.GT.25)NM=NM-2
DO 55 LI=LO,NM
55 CONTINUE
MJE=JSART+LI-1
0541 MJ=ART6(MJ)
0542 IF(MJ.GT.300)MJ=MJ-300
0543 LJ=L+LO+1
JD=ART4(MJ)
DO 5404 L=1,3
0544 IF(THIN.LT.AR(L,MJ))
CALL *N8POS(THIN,THOUT,JD)
LCOVPIFT TOROF

70 JPL TECHNICAL REPORT 32-1305

I

C

C COMPUTE TORQUE

C

CALL MATVEC(XJ,WD,T1)

0.75 CALL MATVEC(XJ,WD,T1)

DO 177 L=1,3

0.76 DO 177 L=1,3

0.77 TIN(L)=13.56*TAN(L-1)/1.8

0.78 TIN(L)=13.56*TAN(L-1)/1.8

0.79 TORO(L)=13.56*TI(L)/3.6

0.80 TORO(L)=13.56*TI(L)/3.6

0.81 TORMG=SORIT(TORO(1)**2+TORO(2)**2+TORO(3)**2)

0.82 TORMG=SORIT(TORO(1)**2+TORO(2)**2+TORO(3)**2)

IF(ER(1),GT,0.9) TORO(1)=60.0

IF(ER(1),GT,0.9) TORO(1)=60.0

IF(ER(2),GT,0.3) TORO(2)=60.0

IF(ER(2),GT,0.3) TORO(2)=60.0

IF(ER(3),GT,0.3) TORO(3)=60.0

IF(ER(3),GT,0.3) TORO(3)=60.0

JD=AR(4,MJ)

JD=AR(4,MJ)

JH=AR(5,MJ)

JH=AR(5,MJ)

JM=AR(6,MJ)

JM=AR(6,MJ)

JS=AR(7,MJ)

JS=AR(7,MJ)

C IF(IOUT.EQ.0) GO TO 7903

C IF(IAXIS.EQ.0) GO TO 7902

C FIND MINIMUM RATE INCREMENT

C

IF(IAXIS.NE.1) GO TO 7801

W(1)=P(2)

W(1)=P(2)

W(2)=-2.5*P(2)*P(3)

W(2)=-2.5*P(2)*P(3)

W(3)=P(1)+P(2)*P(3)**2

W(3)=P(1)+P(2)*P(3)**2

GO TO 7803

GO TO 7803

IF(IAXIS.NE.2) GO TO 7802

W(1)=Q(2)

W(1)=Q(2)

W(2)=-2.5*Q(2)*Q(3)

W(2)=-2.5*Q(2)*Q(3)

W(3)=Q(1)+Q(2)*Q(3)**2

W(3)=Q(1)+Q(2)*Q(3)**2

GO TO 7803

GO TO 7803

IF(IAXIS.NE.3) GO TO 7803

W(1)=R(2)

W(1)=R(2)

W(2)=-2.5*R(2)*R(3)

W(2)=-2.5*R(2)*R(3)

W(3)=R(1)+R(2)*R(3)**2

W(3)=R(1)+R(2)*R(3)**2

C CALL MINRAT(W,P,JH,DEL,RAID,REALT,DEDZON,INDCAT,MJ)

C CALL MINRAT(W,P,JH,DEL,RAID,REALT,DEDZON,INDCAT,MJ)

IF(INDCAT.EQ.1) GO TO 79

IF(INDCAT.EQ.0) GO TO 79

C DATA OUTPUT

C

WRITE(6,207) IAXIS

WRITE(6,207) IAXIS

GO TO 7902

GO TO 7902
079   WRITE(6,204)IAXIS,DFDZOM,RATINC
      WRITE(6,204)REALT(I),I=1,4
07902  WRITE(6,991)
      WRITE(6,101)J0,JH,JM,JS
      WRITE(6,201)LIN
      WRITE(6,102)
      WRITE(6,103)TIN(I),I=1,3
      WRITE(6,104)TIN(I),I=1,3
      WRITE(6,105)TORMAG
      WRITE(6,106)FR1,FR2,FR3
      DO 17902 I=1,3
17902  INV(I)=TIN(I)+TIN(I,2)
      N=NEVOUT
      WRITE(6,120)N,INV(I),I=1,3
      I10UT=0
072   MQ=JSART+LT-1
07201  IF(MQ.GT.300)MG=MG-300
07301  MJJ(1)=AR(4,MG)
07402  MJJ(2)=AR(5,MG)
07503  MJJ(3)=AR(6,MG)
07606  MJJ(4)=AR(7,MG)
07903  KDX=KDX+1
      IF(KDX.EQ.KDY)KDX=KDX+LEVOUT
07904  CONTINUE
      NE=0
      I10UT=1
C    DATA CONDITION OUTPUT PACKAGE
C
     IF(IAXIS.NE.1)GO TO 7906
     WP(1)=P(2)
     WP(2)=2.*P(2)+P(3)
     WP(3)=P(1)+P(2)+P(3)**2
     GO TO 7908
07906  IF(IAXIS.NE.2)GO TO 7907
     WP(1)=Q(2)
     WP(2)=2.*Q(2)+Q(3)
     WP(3)=Q(1)+Q(2)+Q(3)**2
     GO TO 7908
07907  IF(IAXIS.NE.3)GO TO 7908
     WP(1)=R(2)
     WP(2)=2.*R(2)+R(3)
     WP(3)=R(1)+R(2)+R(3)**2
07908  MJJ=N
080   IF(INXPRF.EQ.1)JSART=JSART-1
     IF(INXPRF.EQ.3)JSART=JSART-2
08001  IF(INX.EQ.1)WRITE(6,109)
     IF(INX.EQ.1)WRITE(6,109)
     IF(INX.EQ.2)WRITE(6,110)
     IF(INX.EQ.2)WRITE(6,110)
     IF(INX.EQ.3)WRITE(6,111)
     IF(INX.EQ.3)WRITE(6,111)
     II=JSART-2
081   IF(JEND.GT.3)GO TO 85
082   JJ=JEND-3
089   JSART=JEND
094   GO TO 93
095   IF(JEND.GT.3)GO TO 89
096   JJ=JEND-3
087   JSART=JEND-300
088 GO TO 91
089 JJ=303
090 RERED=.TRUE.
091 JJI=JEND-303
092 JSART=JEND-300
093 IF(I1.GT.3)GO TO 14
094 IF (.NOT. BEGIN) GO TO 94
095 BEGIN=.FALSE.
096 IF(JJ.LE.4)GO TO 49
097 GO TO 14
098 II=300+II
099 IF(JJ.GT.300)GO TO 14
090 II=II+1
091 JJ=303
092 IF(RERED)GO TO 94
093 JJI=JEND-3
094 RERED=.TRUE.
095 GO TO 14
096 JJ=300+JJ
097 IF(I1.EQ.14)GO TO 14
098 CALL STOP
099 STOP
100 FORMAT(1H/1)
101 FORMAT(1X,18H NOT ENOUGH POINTS)
102 FORMAT(1H9,3H,3DAY,14,5X,4HOUR,14,5X,3HMIN,14,5X,3HSEC,14,5X)
103 FORMAT(1H9,24HTORQUE LIMITS ARE--UPPER,3E20.8)
104 FORMAT(1H9,24HTORQUE BY LEAST SQUARES,4E20.8)
105 FORMAT(1H9,24HRMS ERROR OF PARAB FIT,4E20.8/)
106 FORMAT(1H9,69HPLOT OF MARINER PITCH YAW AND ROLL TORQUES IN DY)
107 **NEWCM = STARTING DAY,3X,i5**
108 FORMAT(1H9,28HOUTAGE IN DATA AT THIS POINT)
109 FORMAT(1H9,45H MINIMUM RATE ERROR AVERAGED OUT IN PREVIOUS DATA BATCH)
110 FORMAT(1H9,8HRAD DATA)
111 FORMAT(1H9,6F5.2)
112 FORMAT(1H9,24HNULL INTERSECTION INTERVAL ON AXIS,i5)
113 FORMAT(1H9,24HNUMER OF POINTS USED,14,24H TORQUE INTERVALS-PI)
114 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
115 FORMAT(1H9,2X,i2,H,SHOUTPUT AT,14,23H POINT OF PARABOLA WITH,14,7H POIN)
116 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
117 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
118 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
119 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
120 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
121 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
122 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
123 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
124 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
125 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
126 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
127 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
128 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
129 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
130 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
131 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
132 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
133 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
134 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
135 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
136 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
137 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
138 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
139 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
140 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
141 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
142 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
143 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
144 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
145 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
146 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
147 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
148 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
149 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
150 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
151 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
152 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
153 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
154 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
155 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
156 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
157 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
158 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
159 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
160 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
161 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
162 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
163 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
164 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
165 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
166 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
167 FORMAT(1H9,2X,i2,H,15E16.8,6H ROLL,15E16.8/)
168 FORMA
C   OUTPUT
C   A(3,2) COEFFICIENTS OF INTERVAL PARABOLA FIT
C   B(3,2) COEFFICIENTS OF INTERVAL PARABOLA FIT
C   C(3,2) COEFFICIENTS OF INTERVAL PARABOLA FIT
C   AT1,21,B1,21,C1,21
C   PITCH I=1
C   TAN I=2
C   ROLL I=3
C   INX INDEX =1 NULL INTERSECTION ON PITCH AXIS
C   =2 NULL INTERSECTION ON YAW AXIS
C   =3 NULL INTERSECTION ON ROLL AXIS
C
002 COMMON ART7,3061,CKLIM(61,DEL
003 DOUBLE PRECISION ERT(3,3),FRT(3,3)
0040 DIMENSION A(3,2),B(3,2),C(3,2),ORT(3,2),SRT(3,2),RRT(3,2),G(2)
005 NHAFF=N/2
006 INX=0
007 ND=1
008 N=NA/N
009 KM=NA/N
0010 ISKIP=2
0011 IF(DEL GT 2.5) ISKIP=0
0012 JD=AR(4,JSART)
0013 DO 64 L=1,N
0014 DO 57 I=1,N
0015 MI=JSART+1+ISKIP
C
C   SETUP OF E AND Q
C
0016 IF(MI GT 300)MI=M-300
0017 CALL IANGPS(L,VI,TH,JD)
0018 ORT(1,2)=TH(2)
0019 ORT(1,1)=TH(1)
0020 X=1+ISKIP
0021 DO 15 J=1,3
0022 JI=J+3
0023 IF(JI EQ 0)GO TO 1402
0024 ERT(1,J)=DALE((X*DELT)**J)
0025 GO TO 15
0026 CONTINUE
0027 DO 57 K=1,3
0028 M=JSART+NHAFF+K-3
0029 IF(MM GT 300)MM=M-300
0030 CALL IANGPS(L,MM,TH,JD)
0031 ORT(2,2)=TH(2)
0032 ORT(2,1)=TH(1)
0033 X=NHAFF+K-3
0034 DO 26 J=1,3
0035 J=J+4
0036 IF(J EQ 0)GO TO 2502
0037 ERT(2,J)=DALE((X*DELT)**J)
0038 GO TO 26
0039 CONTINUE
0040 DO 57 M=1,11
0041 MK=JSART+N-M-1
0042 IF(MK GT 300)MK=M-300
0043 CALL IANGPS(L,MK,TH,JD)
0044 ORT(3,2)=TH(2)

JPL TECHNICAL REPORT 32-1305
032. ORT(3,1) = TH(1)          PARINT
033. X = N + M - 1          PARINT
034. DO 37 J = 1, 3          PARINT
035. PJJ = J          PARINT
03501 IF (.NOT. G(J,J)) GO TO 3602  PARINT
036. FRT(3, J) = DALE1((X*DELT)**J))  PARINT
03601 GO TO 37          PARINT
03602 FRT(3, J) = 1.0000  PARINT
037. CONTINUE          PARINT

C FIT CURVE AND SORT OUT DATA
C
038. CALL MATIN3(FRT, FRT, IMD)  PARINT
03801 IF (IMD.EQ.0) GO TO 57          PARINT
03802 DO 40 MM = 1, 3          PARINT
03803 DO 39 NN = 1, 3          PARINT
039. SRT(MM, NN) = SNGL(FRT(MM, NN))  PARINT
040. CONTINUE          PARINT
041. CALL MAVT(SRT, ORT, RRT)  PARINT
042. IF(K.EQ.1) GO TO 44          PARINT
043. IF(K.EQ.1 .AND. M.EQ.1) GO TO 54          PARINT
044. DO 52 J = 1, 3          PARINT
045. G(0) = VRT(J, 1)          PARINT
046. G21 = RRT(J, 2)          PARINT
047. F(0) = EL(J, 1)          PARINT
048. F(2) = EL(J, 2)          PARINT
050. CALL INTERX(F, G, P, IND)  PARINT
051. EL(J, 1) = P(1)          PARINT
052. EL(J, 2) = P(2)          PARINT
05201 IF(IND*EQ.0)*IF(INX*L  PARINT
053. GO TO 57          PARINT
054. DO 56 J = 1, 3          PARINT
055. EL(J, 1) = RRT(J, 1)          PARINT
056. EL(J, 2) = RRT(J, 2)          PARINT
057. CONTINUE          PARINT
058. A(L, 1) = EL(I, 1)          PARINT
059. A(L, 2) = EL(I, 2)          PARINT
060. B(L, 1) = EL(I, 2)          PARINT
061. B(L, 2) = EL(I, 2)          PARINT
062. C(L, 1) = EL(I, 1)          PARINT
063. C(L, 2) = EL(I, 1)          PARINT
064. CONTINUE          PARINT
065. RETURN          PARINT

END          PARINT

$IRFTC MAXVT
001. SUBROUTINE MAVT(A, R, C) MAXVT
002. MULTIPICATION OF MATRIX BY INTERVAL VECTOR MAXVT
003. MAVT - 3 ARGUMENTS MAXVT
004. INPUT MAXVT
005. A(3,3) 3X3 MATRIC MAXVT
006. B(3,2) 3-DIMENSIONAL VECTOR-INTERVAL MAXVT
007. C(3,2) 3-DIMENSIONAL VECTOR-INTERVAL MAXVT
008. R(I,1) UPPER MAXVT
009. B(I,2) LOWER MAXVT
010. OUTPUT MAXVT
011. C(I,2) 3-DIMENSIONAL VECTOR-INTERVAL MAXVT
012. DIMENSION A(3,3), B(3,2), C(3,2), S(21), D(2), F(2), G(2) MAXVT
013. DO 18 I = 1, 3 MAXVT
014. S(I) = 0 MAXVT

4
C CONSTANT IDMT

C(2) INTERVAL, F = B*C

DIMENSION B(2), F(2)

X = C*R(1)

Y = C*B(2)

F(1) = AMAX1(X, Y)

F(2) = AMIN1(X, Y)

RETURN

END

$IBFTC ISUBTR

SUBROUTINE ISUB(A, B, C)

INPUT A, B

OUTPUT C, C = A - B

DIMENSION A(2), B(2), C(2), D(2)

D(1) = -B(1)

D(2) = -B(2)

CALL IADDA(D, A, C)

RETURN

END

$IBFTC IMULT

SUBROUTINE IMULT(A, B, C)

INPUT A, B

OUTPUT C, C = AXB

DIMENSION A(2), B(2), C(2)

W = A(1)*B(1)

X = A(1)*B(2)

Y = A(2)*B(1)

Z = A(2)*B(2)

C(1) = AMAX1(W, X, Y, Z)

C(2) = AMIN1(W, X, Y, Z)

RETURN

END
C
C LEAST SQUARES FIT A PARABOLA
C
1 SUBROUTINE PARFIT(T(N),P,D,DELTA,ER)
C
C PARFIT = 5 ARGUMENTS
C
2 INPUT
3 T POINTS
N NUMBER OF POINTS
4 OUTPUT
C
P(3) LEAST SQUARE FIT COEFFICIENT
Delt TIME BETWEEN 2 POINTS IN MIN
ER MEAN LEAST SQUARE ERROR

2 REAL T(N),Q(200,3),P(3),A(3,3),B(3),C(3)
3 DOUBLE PRECISION M(3,3),MINV(3,3),DET

C GENERATE Q

4 DO 7 I=1,N
5 Q(I,1)=1.
6 Q(I,2)=FLOAT(I-1)*DELTA
7 Q(I,3)=(FLOAT(I-1)*DELTA)**2

C GENERATE Q* T

8 DO 12 I=1,3
9 DO 12 J=1,3
10 M(I,J)=Q(I,J)*Q(1,1)
11 DO 12 K=1,N
12 M(I,J)=M(I,J)+Q(K,1)*Q(K,J)

C GENERATE Q* T

13 DO 16 I=1,3
14 R(I,1)=0.
15 DO 16 J=1,N
16 R(I,J)=A(I,J)*Q(J,1)*T(J)

C SOLVE LINEAR EQUATIONS

17 CALL MATINV(M,MINV,ID)
18 DO 1703 I=1,3
19 DO 1703 J=1,3
20 A(I,J)=SNGL(MINV(I,J))
21 CALL MATVEC(A,B,C)
22 P(1)=C(1)-0.25*C(2)*C(2)/C(3)
23 P(2)=C(3)
24 P(3)=-0.5*C(2)/C(3)
25 ER=ER+F**2
26 RETURN
27 END

C PULSE DETECT SUBROUTINE

C
SUBROUTINE DETECT(JSART, JFND, N, INX, IAXIS)

C DETECT = 5 ARGUMENTS
C INPUT
C JSART STARTING POINT
C OUTPUT
C JFND ENDING POINT
C N NUMBER OF POINTS
C INX INDEX = 1 OUTAGE IN DATA
C = 2 ERROR AVERAGED OUT
C = 3 BAD DATA
C IAXIS INDEX = 0 NO FIRING
C = 1 FIRING IN PITCH
C = 2 FIRING IN YAW
C = 3 FIRING IN ROLL

DIMENSION K(31, DB(3))
LOGICAL TRUFAL
COMMON ART(30, 6), CKLIM(6, 6)
TRUFAL, FALSE
IND = 0
INX = 0
IAXIS = 0

C CHECK TO SEE IF POINTS ARE NEAR DEADBAND

L = JSART + 1
DO 15 I = L, 600
K(I) = 0
K(2) = 0
K(3) = 0
M = I
12 IF (K(I) .GT. 300) M = M - 300
1201 DO 1205 J = 1, 3
1202 DB(J) = APS(CKLIM(2 * J + 1))
1203 YY = POLY(J, AR(J, M))
1204 IF (YY .LT. 0.) DB(J) = APS(CKLIM(2 * J - 2))
1205 CONTINUE
1206 CALL CHECK(M, IND, DB)
1207 IF (IND .EQ. 2) INX = 2
1208 IF (IND .EQ. 1 OR IND .EQ. 3) GO TO 1501
13 DO 15 J = 1, 3
1301 J = J
1302 YY = POLY(J, AR(J, M))
14 IF (ABS(YY) .GT. 10 * DB(J)) CALL PATRN(M, J, TRUFAL)
15 IF (TRUFAL) GO TO 16
1501 JEND = I
1502 INX = IND
1503 GO TO 17
16 JEND = I
1601 IAXIS = JJ
17 N = JEND - JSART + 1
18 RETURN
19 FND
SUBFTE PAT
C
C CHECK PATTERN FOR POSSIBLE FIRING
C
C SUBROUTINE PATRN(I, J, TRUFAL)
C P - 3 ARGUMENTS
C INPUT

C I INDEX PITCH I=1, YAW I=2, ROLL I=3

C

C OUTPUT

C TRFL = TRUE FOR PARTICULAR PATTERN

C = FALSE FOR OTHERS

C

002 COMMON AR(7,306),CKLIM(6),DEL

00201 DIMENSION A(7)

003 LOGICAL TRFL

00301 TRFL = .TRUE.

00302 A(1) = ABS(POLY(J,AR(J,I-1)))

00303 A(2) = ABS(POLY(J,AR(J,I-2)))

00304 A(3) = ABS(POLY(J,AR(J,I-1)))

00305 A(4) = ABS(POLY(J,AR(J,I+1)))

00306 A(5) = ABS(POLY(J,AR(J,I+1)))

00307 A(6) = ABS(POLY(J,AR(J,I+2)))

004 A(7) = ABS(POLY(J,AR(J,I+3)))

00401 IF(A(1).NE.A(4)) GO TO 5

00402 IF(A(2).NE.A(4)) GO TO 5

00403 IF(A(3).NE.A(4)) GO TO 5

00404 IF(A(5).GE.A(4)) GO TO 5

00405 IF(A(6).GE.A(4)) GO TO 5

00406 IF(A(7).GE.A(4)) GO TO 5

00407 GO TO 8

005 IF(A(3).GE.A(4).OR.A(4).LT.A(5)) RETURN

006 IF(A(1).GT.A(4)) RETURN

00601 IF(A(2).GT.A(4)) RETURN

00602 IF(A(5).GT.A(4)) RETURN

00607 IF(A(7).GT.A(4)) RETURN

007 TRFL = .TRUE.

009 RETURN

010 END

*TRFC INTX

C INTERVAL INTERSECTION

C

001 SUBROUTINE INTERX(F,G,P,IND)

C

C INPUT

C F(2) INTERVAL

C G(2) INTERVAL

C OUTPUT

C P(2) = F(2) IF NO INTERSECTION

C = F(2) INTERSECTION G(2)

C IND INDEX = 0 IF NO INTERSECTION

C = 1 IF INTERSECTION

C

002 DIMENSION F(2),G(2),P(2)

003 IND=1

004 IF(F(2).LT.F(1)) GO TO 8

005 F=F(2)

006 F(2)=F(1)

007 F(1)=E

008 IF(G(2).LT.G(1)) GO TO 12

009 E=G(2)

010 G(2)=G(1)

011 G(1)=F

012 IF(F(2).GT.G(2).OR.G(1).GT.F(1)) GO TO 20

013 P(1)=G(1)

014 P(2)=G(2)
IF (G(1) * GT. * G(2)) P(2) = F(2)
RETURN
IND = * 
P(1) = F(1)
P(2) = F(2)
RETURN
END

SUBROUTINE IADD(F, G, SM)

INPUT F(2), G(2)
OUTPUT SM(2) 
SM = F + G

DIMENSION F(2), G(2), SM(2)
IF (F(1) * GT. * F(2)) GO TO 7
E = F(2)
F(2) = F(1)
F(1) = E
IF (F(1) * GT. * G(2)) GO TO 11
E = G(2)
G(2) = G(1)
F(1) = E
SM(2) = F(2) * G(2)
RETURN
END

SUBROUTINE MATT

3X3 MATRIX INVERSION

SUBROUTINE MATINV(A, S, I)

INPUT A(3, 3) 
OUTPUT B(3, 3) 
INVERSE MATRIX OF A

I INDEX #1 FOR NON SINGULAR MATRIX
= 0 FOR SINGULAR MATRIX

DOUBLE PRECISION A(3, 3), B(3, 3), DET

IF (DARS(1) * LT. * 1.0 - 15) GO TO 15
B(1, 1) = (A(2, 2) * A(3, 3) - A(2, 3) * A(3, 2)) / DET
B(1, 2) = -(A(1, 2) * A(3, 3) - A(1, 3) * A(3, 2)) / DET
B(1, 3) = (A(1, 2) * A(2, 3) - A(1, 3) * A(2, 2)) / DET
B(2, 1) = -(A(2, 1) * A(3, 3) - A(2, 3) * A(3, 1)) / DET
B(2, 2) = (A(1, 1) * A(3, 3) - A(1, 3) * A(3, 1)) / DET
B(2, 3) = -(A(1, 1) * A(2, 3) - A(1, 3) * A(2, 1)) / DET
B(3, 1) = -(A(2, 1) * A(3, 2) - A(2, 2) * A(3, 1)) / DET
B(3, 2) = (A(1, 1) * A(3, 2) - A(1, 2) * A(3, 1)) / DET
B(3, 3) = (A(1, 1) * A(2, 2) - A(1, 2) * A(2, 1)) / DET
RETURN
END

RETURN
END

SUBROUTINE CHECK

JPL TECHNICAL REPORT 32-1305
C DATA CHECK
C
C SUBROUTINE CHECK(M,N,DD)
C
C INPUT M SUBSCRIPT FOR ARRAY AR
C CHECK
C OUTPUT N INDEX =1 OUTAGE IN DATA
C CHECK
C =2 ERROR AVERAGED OUT
C CHECK
C =3 BAD DATA
C CHECK
C DB(3) DEADBAND, PITCH YAW ROLL
C CHECK
C
C COMMON AR(7,306),CKLIM(61)*DEL
C
C DIMENSION B(I),DB(3)
C
C DO 20 I=1,3
C CHECK
C B(I)=DB(I)+5.00
C CHECK
C
C IF(ABS(AR(5,M)-AR(5,M-1))*.LT.0.5)GO TO 6
C CHECK
C TMN=60.0
C CHECK
C 20 GO TO 7
C CHECK
C
C TM2=AR(6,M)
C CHECK
C
C TIM1=AR(7,M)-60.0*AR(6,M-1)+DEL
C CHECK
C TIM2=AR(7,M)+60.0*TMN
C CHECK
C
C IF(ABS(TIM1-TIM2)*.LT.10.0)GO TO 10
C CHECK
C N=1
C CHECK
C RETURN
C CHECK
C
C CHECK FOR BIT ERROR
C CHECK
C
C 10 DO 20 J=1,3
C CHECK
C A1=POLY(J,AR(J,M))
C CHECK
C A2=POLY(J,AR(J,M-1))
C CHECK
C 20 IF(ABS(A2-A1)*.GT.2.0)GO TO 22
C CHECK
C RETURN
C CHECK
C 22 N=2
C CHECK
C GO TO 24
C CHECK
C
C DO 24 K=1,3
C CHECK
C A1=POLY(K,AR(K,M))
C CHECK
C 24 IF(ABS(A1)*.GT.11.0)GO TO 31
C CHECK
C AVERAGE OUT BIT ERRORS
C CHECK
C
C 29 DO 26 L=1,3
C CHECK
C A1=POLY(L,AR(L,M+1))
C CHECK
C 26 IF(ABS(A1)*.GT.3.0)GO TO 35
C CHECK
C 27 DO 29 LI=1,3
C CHECK
C A1=POLY(LI,AR(LI,M))
C CHECK
C 29 CONTINUE
C CHECK
C 28 RETURN
C CHECK
C
C 31 DO 33 LK=1,3
C CHECK
C A1=POLY(LK,AR(LK,M+1))
C CHECK
C 33 IF(ABS(A1)*.GT.3.0)GO TO 35
C CHECK
C 34 RETURN
C CHECK
C 35 N=3
C CHECK
C 36 RETURN
C CHECK
C END
C STOP
C TRANSF

C SUBROUTINE--MULTIPLY VECTOR BY A SQUARE MATRIX--LINEAR TRANSFORMATION

C
C SUBROUTINE MATVEC(Y,W,R)
C
C DIMENSION Y(3,3),M(3),R(3)
C
C R(1)=Y(1,1)*M(1)+Y(1,2)*M(2)+Y(1,3)*M(3)
C
C JPL TECHNICAL REPORT 32-1305
ANGPS & COVPUTATION OF INTERVALS F38 AVERAGE POSITION

ANGPS - 4 ARGUMENTS

INPUT

L INDEX YAW FOR L=1

M YAW FOR L=2

ROLL FOR L=3

TH(2) INTERVAL OF ANGULAR POSITION, (UPPER,LOWER)

C OUTER

REAL(4) TIME, DAY-HR-MIN-SEC, (REALT) = (T) + (M)

INTEGER REALT(4)

REAL M(4)

J = T

XJ = J

X = M(3) + XJ

Y = (T - XJ) * 60.

Z = M(4) + Y

IF (Z < 60.) GO TO 14

REALT(4) = Z - 60.

12 X = M(3) + XJ + 1.

GO TO 15

REALT(4) = 2

IF (X < 60.) GO TO 19

REALT(9) = X - 60.

17 Y = M(7) + I.

GO TO 21

REALT(9) = X

20 Y = M(2)

IF (Y < 24.) GO TO 25

REALT(2) = Y - 24.

23 REALT(1) = M(1) + 1.

RETURN

25 REALT(2) = Y

26 REALT(1) = M(1)

RETURN

END

ANG64
 COMMON/CANT/CLK/CANT2/CONE(275)

 JD = TIME AFTER LAUNCH IN DAYS

 IF(IDLTF399) JD=ID+94
 IF(IDLT330) JD=ID-331
 THE1=CLK*17453293E-01
 THE2=CONE(JD)*17453293E-01

 C1=COS(THE1)
 C2=COS(THE2)
 S1=SIN(THE1)
 S2=SIN(THE2)
 IF(L=EQ.31) GO TO 16
 D1=AR(L,M)

 POLY IS CALIBRATION FUNCTION

 TH(1)=POLY(LDN-.57)
 TH(2)=POLY(LDN+.57)
 RETURN

 DN=AR(1,M)

 NEED TO CALCULATE TH FOR PITCH AND YAW TO OBTAIN TH FOR ROLL

 THX(1)=POLY(1,DN-.57)
 THX(2)=POLY(1,DN+.57)
 DN1=AR(2,M)
 THY(1)=POLY(2,DN-.57)
 THY(2)=POLY(2,DN+.57)
 DN2=AR(3,M)
 THZ(1)=POLY(3,DN-.70)
 THZ(2)=POLY(3,DN+.70)

 CALL ISMULT(THX,C1,T1)
 CALL ISMULT(THY,S1,T2)
 CALL ISMULT(T1,T2,T9)
 CALL ISMULT(T3,C2,T1)
 CALL ISMULT(T3,T1,T2)
 A=1./S2
 CALL ISMULT(T2,A,T3)
 TH(1)=T3(1)
 TH(2)=T9(2)
 RETURN

 END

 COMputation of INTERVALS for Angular Position

 SUBROUTINE IANGPS(L,M,TH,ID)

 IANGPS - 4 arguments

 INPUT

 L INDEX PITCH FOR L=1
 YAW FOR L=2
 ROLL FOR L=3
 L,M SUBSCRIPTS FOR ARRAY AR
 ID TIME IN DAYS FROM JAN 1
 OUTPUT

 TH(2) INTERVAL OF ANGULAR POSITION (UPPER,LOWER)

 DIMENSION TH(2),THX(2),THY(2),THZ(2),T1(2),T2(2),T3(2)
 COMMON AR(7,306),CKLM(6),DEL
COMMON/CANOP1/CLOCK/CANOP2/CONE(135)

C JD = TIME AFTER LANCE IN DAYS

C 504 JD=JD-164

C 505 THF1A=CLOCK*1.7453293E-01

C 506 THF2A=CONE(JDJ)*1.7453293E-01

C 507 C1=COS(THF1A)

C 508 C2=COS(THF2A)

C 509 S1=SIN(THF1A)

C 510 S2=SIN(THF2A)

C 511 IF(L.EQ.3)GO TO 16

C 512 DN=AR(L+11)

C POLY IS CALIBRATION FUNCTION

C 513 TH(1)=POLY(L*DN+.57)

C 514 TH(2)=POLY(L*DN+.57)

C 515 RETURN

C 516 DN=AR(L+11)

C NEED TO CALCULATE TH FOR PITCH AND YAW TO OBTAIN TH FOR ROLL

C 517 THX(1)=POLY(L*DN+.57)

C 518 THX(2)=POLY(L*DN+.57)

C 519 DN=AR(L+11)

C 520 THY(1)=POLY(L*DN+.57)

C 521 THY(2)=POLY(L*DN+.57)

C 522 DN=AR(L+11)

C 523 THZ(1)=POLY(L*DN+.57)

C 524 THZ(2)=POLY(L*DN+.57)

C 525 CALL ISMUL(THX+C1*T1)

C 526 CALL ISMUL(THY+C2*T2)

C 527 CALL IADD(T1,T2,T3)

C 528 CALL IADD(T3+C2,T1)

C 529 CALL IADD(T2+A,T3)

C 530 TH(1)=T3(1)

C 531 TH(2)=T3(2)

C 532 RETURN

C 533 END

STFTC: MNRT

C COMPUTATION OF MINIMUM RATE INCREMENT

C 534 SUBROUTINE MNRT(WP,YYJ,DT,RTING,REALT,DEDZON,INDCAT,MJ)

C MNRT — 9 ARGUMENTS

C INPUT

C W(3) COEFFICIENTS OF PRESENT PARABOLA

C N(3) NUMBER OF POINTS USED IN LAST PARABOLA

C DFLT TIME BETWEEN 2 POINTS IN MIN

C MJ(4) STARTING TIME OF LAST PARABOLA, DAY-HR-MIN-SEC

C OUTPUT

C RTING RATE INCREMENTS IN DYNE CM/CM

C REALT TIME OF FIRING, DAY-HR-MIN-SEC

C DEDZON DEADBAND

C INDCAT INDEX = 1 COMPUTATIONAL ERROR

C INDEX = 0 NO COMPUTATIONAL ERROR
C COMMON/LVT/LEVOUT
00101 COMMON/LVT/LEVOUT
002 DIMENSION W(3),WP(3)
003 INTEGER REALT(4)
004 REAL MJ Ji(4)
005 FL=14.
006 FPLEVOUT*EQ*.2, FL=2
007 INDICAT=0
008 X=MJ Ji+1
C
C SHIFT PRESENT PARABOLA AND SOLVE QUADRATIC EQUATION IN T
C
009 SHIFT=X*DELT
010 A=WP(1)-W(1)
011 B=WP(2)+2.*W(1)*SHIFT-W(2)
012 C=WP(3)-(SHIFT**2)*W(1)+%2(2)*SHIFT-W(3)
013 D=N**2-4.*A*C
014 IF(D)19,9
015 DIS=SQR(T(D))
016 T=(-B-DIS)/(2.*A)
017 IF(CAPS(SHIFT=))LT+DEL T +GO TO 14
018 T=(-B+DIS)/(2.*A)
019 IF(CAPS(SHIFT=))LT+DEL T +GO TO 19
020 RT1=2.*WP(1)*T+WP(2)
021 RT2=2.*W(1)+T-SHIFT+W(2)
022 RATINC=(RT2-RT1)*100*.6.
023 IFDIS=WP(1)**2+WP(2)*T+WP(3)
024 CALL RELTIM(T,MJJ,REALT)
025 RETURN
026 INDICAT=1
027 RETURN
028 END

ANGULAR POSITION VS DATA NUMBER--CALIBRATION FOR MV-67

FUNCTION POLYL(BDN)
031 FUNCTION POLYL(BDN)
032 DIMENSION A(6,3)
033 DATA A(I,1) = 1.187/275.267653,-192093094,-192876455-E01,-202236066-E00
034 DATA A(I,2) = 0.2315772E-05,-667829606-E08,2842241,-1.0178037,-20517513E-0
035 DATA A(I,3) = -2857125E-09,-20782091E-05,-63998052E-08,-21141927-3237947
036 DATA A(I,4) = 31,1169738E-11,-34020915E-07,0.33583269E-05,-10896960E-07
037 DATA A(I,5) = 0*DN*5
038 RETURN
039 END

ANGULAR POSITION VS DATA NUMBER--CALIBRATION FOR MV-64

FUNCTION POLYL(BDN)
041 FUNCTION POLYL(BDN)
042 DIMENSION A(4,3)
043 DATA A(I,1) = 1.121/276.818863,-27638703,-1383448E-02,-76307050
044 DATA A(I,2) = 0.13816884,-27638703,-1383448E-02,-76307050
045 DATA A(I,3) = 0*DN*3
046 RETURN
047 END

ANGULAR POSITION VS DATA NUMBER--CALIBRATION FOR MM-64

FUNCTION POLYL(BDN)
049 FUNCTION POLYL(BDN)
050 DIMENSION A(4,3)
051 DATA A(I,1) = 2.127/276.818863,-27638703,-1383448E-02,-76307050
052 DATA A(I,2) = 0.13816884,-27638703,-1383448E-02,-76307050
053 DATA A(I,3) = 0*DN*3
054 RETURN
055 END

IMPFCT ANGLE

THIS IS FOR MM-64

ANGLE4
<table>
<thead>
<tr>
<th>C</th>
<th>BLOCK DATA</th>
<th>ANGLE4</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>BLOCK DATA</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>02</td>
<td>COMMON/CANOPY/CLOCK</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>03</td>
<td>COMMON/CANOPY2/CONE(275)</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>04</td>
<td>COMMON/CPFE/ATT(93)</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>05</td>
<td>DATA CLOCK/56F07/</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>06</td>
<td>DATA (CONET:1=1751/)</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>07</td>
<td>1 101.50, 101.95, 102.09, 102.23, 102.36</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>08</td>
<td>2 102.49, 102.63, 102.72, 102.82, 102.99</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>09</td>
<td>3 103.73, 103.13, 103.21, 103.30, 103.78</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>10</td>
<td>4 103.65, 103.52, 103.58, 103.64, 103.70</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>11</td>
<td>5 103.75, 103.79, 103.83, 103.87, 103.90</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>12</td>
<td>6 103.95, 103.99, 103.97, 103.99, 103.99</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>13</td>
<td>7 104.00, 104.00, 104.00, 104.00, 104.00</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>14</td>
<td>8 103.97, 103.97, 103.97, 103.97, 103.97</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>15</td>
<td>9 103.85, 103.82, 103.78, 103.74, 103.70</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>16</td>
<td>1 103.65, 103.60, 103.55, 103.49, 103.43</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>17</td>
<td>2 103.37, 103.30, 103.24, 103.17, 103.09</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>18</td>
<td>3 103.21, 102.94, 102.86, 102.77, 102.69</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>19</td>
<td>4 102.60, 102.51, 102.42, 102.33, 102.24</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>20</td>
<td>5 102.15, 102.04, 102.05, 101.85, 101.74</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>22</td>
<td>DATA (CONET:1=76.150/)</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>23</td>
<td>7 101.09, 100.97, 100.85, 100.74, 100.62</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>24</td>
<td>8 100.69, 100.37, 100.25, 100.12, 99.99</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>25</td>
<td>9 99.86, 99.73, 99.60, 99.47, 99.34</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>26</td>
<td>1 99.20, 99.06, 98.92, 98.78, 98.64</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>27</td>
<td>2 98.50, 98.36, 98.21, 98.07, 97.92</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>28</td>
<td>3 97.78, 97.65, 97.52, 97.39, 97.19</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>29</td>
<td>4 97.04, 96.89, 96.74, 96.59, 96.44</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>30</td>
<td>5 96.71, 96.19, 95.59, 95.06, 95.00</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>31</td>
<td>6 95.94, 95.39, 95.25, 95.10, 94.95</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>32</td>
<td>7 95.45, 94.66, 94.22, 94.06, 93.70</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>33</td>
<td>8 94.10, 93.97, 93.83, 93.69, 93.56</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>34</td>
<td>9 93.47, 93.28, 93.15, 93.02, 92.88</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>35</td>
<td>1 92.75, 92.62, 92.48, 92.35, 92.22</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>36</td>
<td>2 92.59, 92.38, 92.17, 91.95, 91.74</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>37</td>
<td>3 91.44, 91.31, 91.18, 91.05, 90.93</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>38</td>
<td>DATA (CONET:1=51.225/)</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>39</td>
<td>4 90.90, 90.68, 90.55, 90.43, 90.30</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>40</td>
<td>5 90.18, 90.06, 89.93, 89.79, 89.63</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>41</td>
<td>6 89.57, 89.44, 89.32, 89.20, 89.08</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>42</td>
<td>7 88.96, 88.83, 88.71, 88.60, 88.49</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>43</td>
<td>8 88.37, 88.25, 88.13, 88.02, 87.90</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>44</td>
<td>9 87.78, 87.66, 87.55, 87.43, 87.32</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>45</td>
<td>1 87.20, 87.08, 86.97, 86.85, 86.74</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>46</td>
<td>2 86.62, 86.50, 86.39, 86.27, 86.16</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>47</td>
<td>3 86.04, 85.93, 85.82, 85.70, 85.59</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>48</td>
<td>4 85.48, 85.36, 85.25, 85.14, 85.02</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>49</td>
<td>5 84.91, 84.80, 84.69, 84.58, 84.47</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>50</td>
<td>6 84.35, 84.24, 84.13, 84.02, 83.91</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>51</td>
<td>7 83.83, 83.69, 83.58, 83.47, 83.36</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>52</td>
<td>8 83.25, 83.14, 83.04, 82.93, 82.82</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>53</td>
<td>9 82.71, 82.60, 82.50, 82.39, 82.28</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>54</td>
<td>DATA (CONET:1=276.275/)</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>55</td>
<td>1 82.18, 82.07, 81.96, 81.86, 81.75</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>56</td>
<td>2 81.64, 81.54, 81.43, 81.33, 81.22</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>57</td>
<td>3 81.12, 81.02, 80.91, 80.81, 80.70</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>58</td>
<td>4 80.60, 80.50, 80.40, 80.30, 80.20</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>59</td>
<td>5 80.09, 79.98, 79.88, 79.78, 79.68</td>
<td>ANGLE4</td>
</tr>
<tr>
<td>60</td>
<td>6 79.58, 79.48, 79.38, 79.29, 79.20</td>
<td>ANGLE4</td>
</tr>
</tbody>
</table>
THIS IS FOR MM-67

C

BLOCK DATA

COMMON/CANPOL/CLOCK/CANPOL2/CONE(135)

DATA CLOCK/A45502/

DATA/CONF1=1175/1=7671E02, 7670E02, 7668E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

R 7655E02, 7664E02, 7664E02, 7664E02/

C THIS IS FOR MM-64

C YOU ALSO NEED SUBPROGRAM BLOCK DATA

SUBROUTINE ANGPOST(B,C,K)
SUBROUTINE ANGPOS HAS 3 ARGUMENTS

INPUT B*K
B(1) = PITCH MEASUREMENT
B(2) = YAW MEASUREMENT
B(3) = ROLL MEASUREMENT

OUTPUT C
K = TIME

C(1) = PITCH IN MILLI-RADIANS
C(2) = YAW IN MILLI-RADIANS
C(3) = ROLL IN MILLI-RADIANS

DIMENSION B(3), C(3)
COMMON/COEFF/AA(4,3)
COMMON/CANOP1/CLOCK/CANOP2/CONE(279)

K IS TIME IN DAYS FROM LANCHE
K IS TIME IN DAYS FROM JAN 1
CLOCK = X-AXIS CLOCK ANGLE IN DEGREES
CONE(K) = CANOPUS CONE ANGLE IN DEGREES

IF (KGT.330) J=K-331
IF (KLT.330) J=K+36

IF (KLT.330) J=K+36
THETA2=CONE(J)*17453293E-01
THETA2=CONE(J)*17453293E-01

CALIBRATION OF ANGULAR POSITION FROM DATA NUMBER

P = PITCH MEASUREMENT IN MILLI-RADIANS
Y = YAW MEASUREMENT IN MILLI-RADIANS
R = ROLL MEASUREMENT IN MILLI-RADIANS

P=(AA(1,1)+AA(2,1)*B(1)+AA(3,1))*B(1)**3+AA(4,1)*B(1)**3
Y=(AA(1,2)+AA(2,2)*B(2)+AA(3,2))*B(2)**3+AA(4,2)*B(2)**3
R=(AA(1,3)+AA(2,3)*B(3)+AA(3,3))*B(3)**3+AA(4,3)*B(3)**3

C1=COS(THETAB1)
C2=COS(THETAB2)
S1=SIN(THETAB1)
S2=SIN(THETAB2)

C(1)=P
C(2)=Y
C(3)=(R-C2*+(P*C1+Y*S1))/S2

RETURN

END

THIS IS SMALL ANGLE APPROXIMATION - SECOND TERMS NEGLECTED

C = 1
C1 = (R-C2*+(P*C1+Y*S1))/S2

RETURN

END

SUBROUTINE ANGPOS(B,K)

SUBROUTINE ANGPOS HAS 3 ARGUMENTS

INPUT B*K
B(1) = PITCH MEASUREMENT
B(2) = YAW MEASUREMENT
B(3) = ROLL MEASUREMENT

OUTPUT C
K = TIME

C(1) = PITCH IN MILLI-RADIANS
C C(2) = YAW IN MILLI-RADIANS
C C(3) = ROLL IN MILLI-RADIANS

02 DIMENSION B(3),C(3)
03 COMMON/Coeff/A(16,3)
04 COMMON/CANOP1/CLOCK/CANOP2/CONE(135)
05 J IS TIME IN DAYS FROM LANCH
06 K IS TIME IN DAYS FROM JAN 1
07 J=K-164
08 THETA1=CLOCK*.17453293E-01
09 THETA2=CONE(J)**17453293E-01

C CALIBRATION OF ANGULAR POSITION FROM DATA NUMBER
C P = PITCH MEASUREMENT IN MILLI-RADIANS
C Y = YAW MEASUREMENT IN MILLI-RADIANS
C R = ROLL MEASUREMENT IN MILLI-RADIANS

08 P=(A(1,1)+A(2,1)*B(1)+A(3,1)*B(1)**2+A(4,1)*B(1)**3
09 1*A(5,1)*B(1)**4+A(6,1)*B(1)**5
10 Y=(A(1,2)+A(2,2)*B(2)+A(3,2)*B(2)**2+A(4,2)*B(2)**3
11 1*A(5,2)*B(2)**4+A(6,2)*B(2)**5
12 R=(A(1,3)+A(2,3)*B(3)+A(3,3)*B(3)**2+A(4,3)*B(3)**3
13 1*A(5,3)*B(3)**4+A(6,3)*B(3)**5)
14 C1=COS(THETA1)
15 C2=COS(THETA2)
16 S1=SIN(THETA1)
17 S2=SIN(THETA2)

C THIS IS SMALL ANGLE APPROXIMATION - SECOND TERMS NEGLECTED

16 C(1)=P
17 C(2)=Y
18 C(3)=(R-C2*(1-P*C1+Y*S1))/S2
21 RETURN
22 END

C A PLOTTER ROUTINE IS ALSO NECESSARY
C JPLT3 WAS USED WITH THIS PROGRAM