NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Report 32-1305

Mariner IV and V Disturbance Torques and Limit Cycles

Daniel A. Prelewicz

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

October 1, 1968

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Report 32-1305

Mariner IV and V Disturbance Torques and Limit Cycles

Daniel A. Prelewicz

Approved by:

W E Bachman for G E Sweetnam

G. E. Sweetnam, Manager
Spacecraft Control Section

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

October 1, 1968

TECHNICAL REPORT 32-1305

Prepared Under Contract No. NAS 7-100
National Aeronautics & Space Administration

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 8⅓ 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 (~¼ 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.

Mariner IV and V Disturbance Torques and Limit Cycles

I. 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

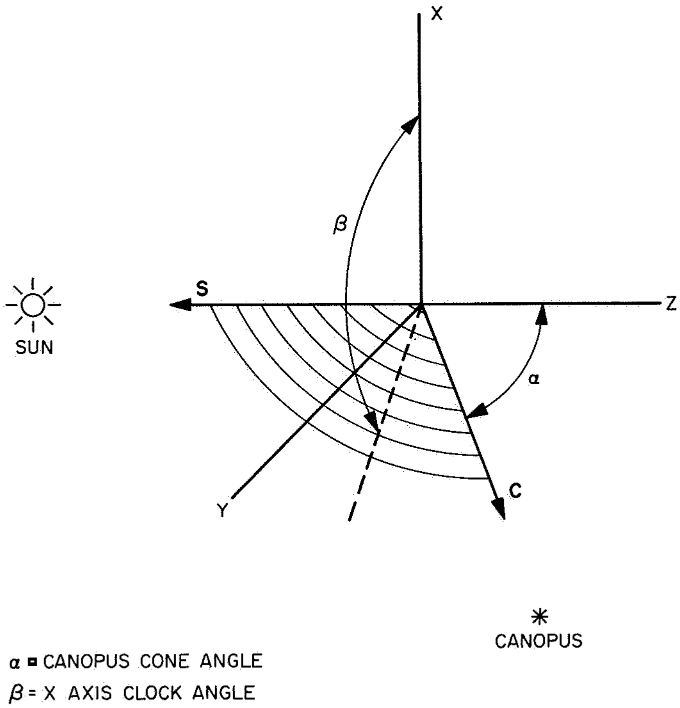


Fig. 1. *Mariner* coordinate reference system

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 (33⅓ bits/s). Approximately 40 days into the mission, the rate is cut to 8⅓ bits/s. During cruise mode operation, the output of the position sensors is sampled every 12.6 (50.4) s when transmission is at 33⅓ (8⅓) 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 X_1, X_2, X_3 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]

$$\left.\begin{aligned}
M_x &= \dot{\omega}_x J_{xx} + \omega_y \omega_z (J_{zz} - J_{yy}) - J_{xy}(\omega_z \omega_x - \dot{\omega}_y) \\
&\quad + J_{xz}(\dot{\omega}_z + \omega_y \omega_x) + J_{yz}(\omega_y^2 - \omega_z^2) \\
M_y &= \dot{\omega}_y J_{yy} + \omega_z \omega_x (J_{xx} - J_{zz}) - J_{yz}(\omega_x \omega_y - \dot{\omega}_z) \\
&\quad + J_{yx}(\dot{\omega}_x + \omega_z \omega_y) + J_{zx}(\omega_z^2 - \omega_x^2) \\
M_z &= \dot{\omega}_z J_{zz} + \omega_x \omega_y (J_{yy} - J_{zz}) - J_{zx}(\omega_y \omega_z - \dot{\omega}_x) \\
&\quad + J_{zy}(\dot{\omega}_y + \omega_x \omega_z) - J_{xy}(\omega_x^2 - \omega_y^2)
\end{aligned}\right\} \quad (1)$$

or, in vector form,

$$\mathbf{T} = J\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times J\boldsymbol{\omega} \quad (2)$$

where $\mathbf{T}$ is the vector torque, $\boldsymbol{\omega}$ is the angular velocity about the center of mass as seen from an inertial reference, and J is the inertia matrix. That is,

$$\left.\begin{aligned}
&\mathbf{T} = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} \qquad \boldsymbol{\omega} = \begin{bmatrix} \dfrac{d\theta_1}{dt} \\ \dfrac{d\theta_2}{dt} \\ \dfrac{d\theta_3}{dt} \end{bmatrix} \qquad \dot{\boldsymbol{\omega}} = \begin{bmatrix} \dfrac{d^2\theta_1}{dt^2} \\ \dfrac{d^2\theta_2}{dt^2} \\ \dfrac{d^2\theta_3}{dt^2} \end{bmatrix} \\
&J_{ij} = \textstyle\int_{mass} (\delta_{ij} r^2 - x_i x_j)\, dm \\
&r^2 = x_1^2 + x_2^2 + x_3^2
\end{aligned}\right\} \quad (3)$$

If $X_1 X_2 X_3$ are taken parallel to XYZ respectively,[2] then the rotational motion in the X_1 X_2 X_3 system is the same as that in the XYZ system.

[1]The 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).

[2]Since 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 $X_1 X_2 X_3$) 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

$$\mathbf{T} = J\dot{\omega} = J\ddot{\theta} \tag{4}$$

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

In Section II, $\dot{\omega}$ 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., $\overline{\mathbf{T}}$, $\bar{a}$) 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 mrad, -0.1114 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.

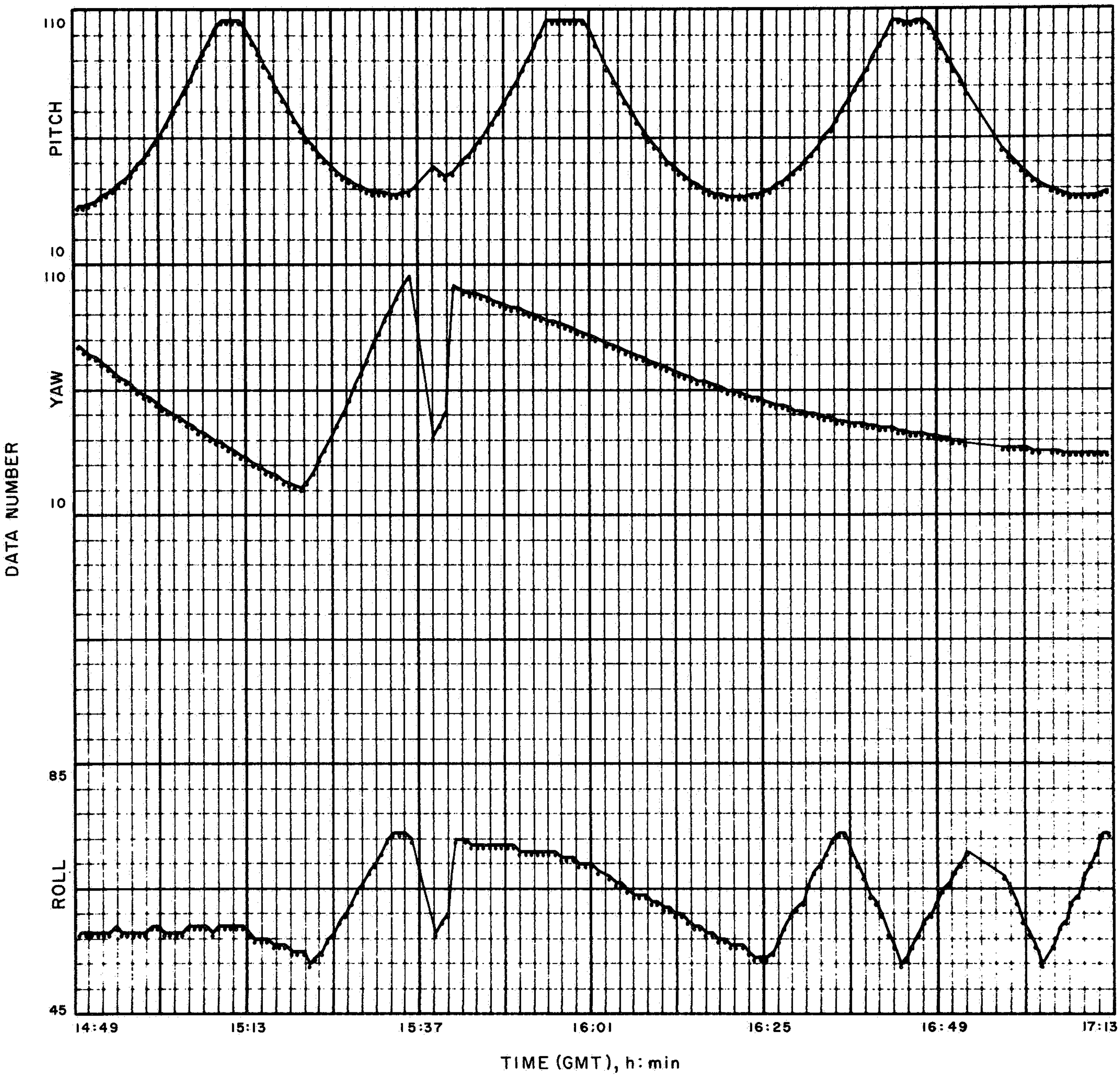


Fig. 2. Edplot data for pitch, roll, and yaw, *Mariner IV*, day 193, 1965

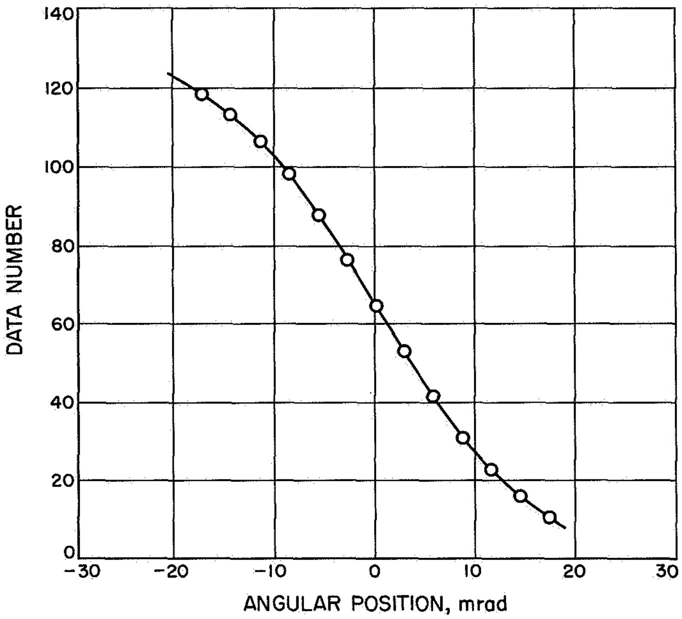


Fig. 3. *Mariner V* pitch sun sensor calibration curve

(2) Because curves are fitted exactly by interval analysis, no further errors are introduced in this step.

Henceforth, computations are done by both single-value and interval analysis. In each case, angular displacements are determined from the raw data by the applicable method.

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.

$$\left.\begin{aligned} \theta_x &= \theta_x \\ \theta_y &= \theta_y \\ \theta_z &= \begin{cases} \dfrac{\theta_v - \cos\alpha\,(\theta_y \sin\beta + \theta_x \cos p)}{\sin\alpha} & \text{(Mars mission)} \\ \dfrac{\theta_v - \cos\alpha\,(\theta_y \sin\beta - \theta_x \cos p)}{\sin\alpha} & \text{(Venus mission)} \end{cases} \end{aligned}\right\} \quad (5)$$

where α is the Canopus cone angle and β is the X axis clock angle. These equations are accurate to first order in the small angles θ_x, θ_y, and θ_v and are easily derived by geometrical arguments.

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 8⅓ bits/s or less than 64 points for 33⅓ bits/s, the torque interval was so large (around 5 dyn-cm) that it was not worth the effort of processing the data.

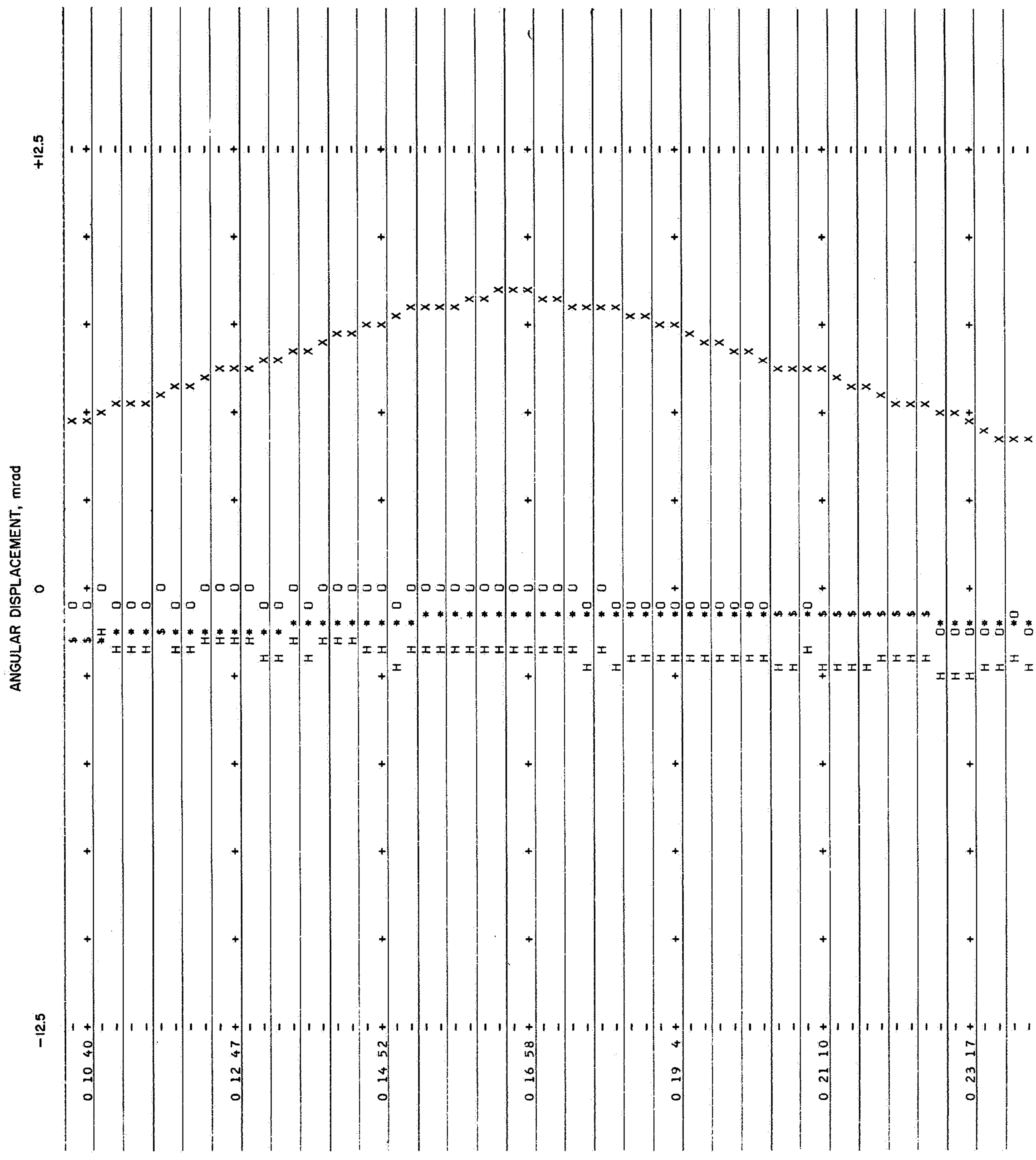


Fig. 4. Reduced position data vs time. Time is in hours, minutes, and seconds (GMT). Symbols: * = pitch; X = yaw; O = V-axis roll; H = Z-axis or true roll; $ = superimposed data points

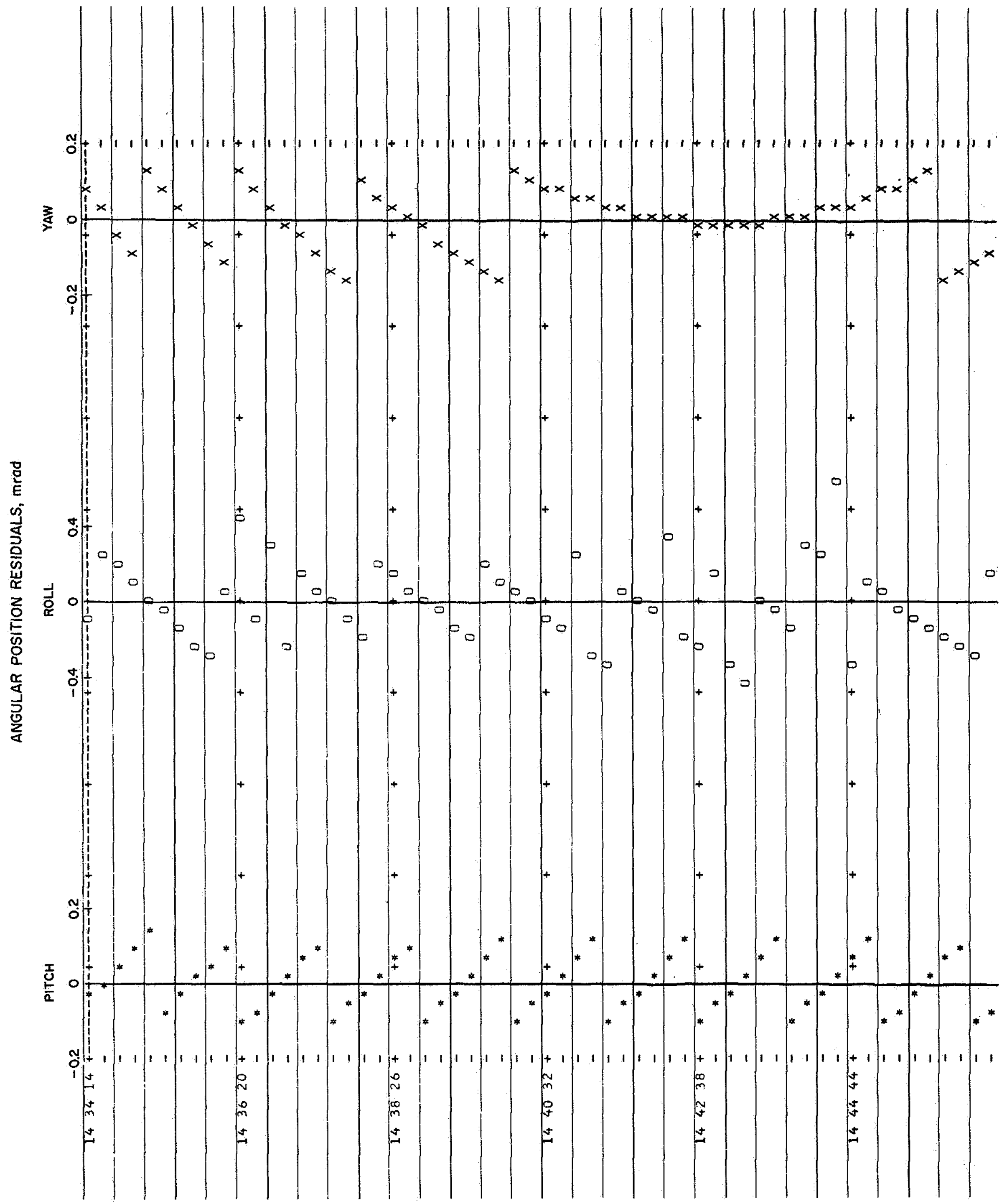


Fig. 5. Pitch, roll, and yaw least-squares fit residuals, *Mariner V*, day 177, 1967. Time is in hours, minutes, and seconds (GMT). Symbols: * = pitch; X = yaw; O = V-axis roll

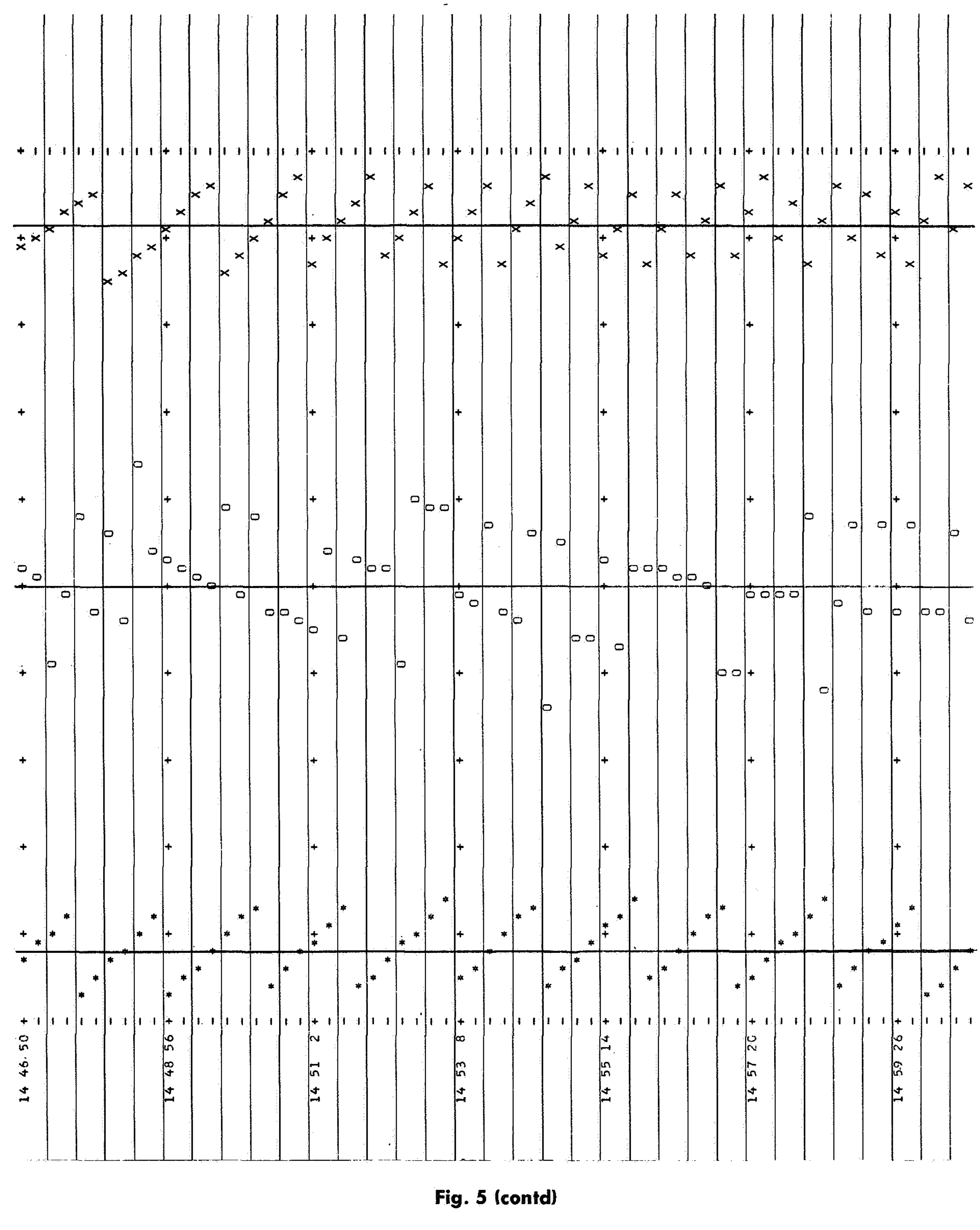


Fig. 5 (contd)

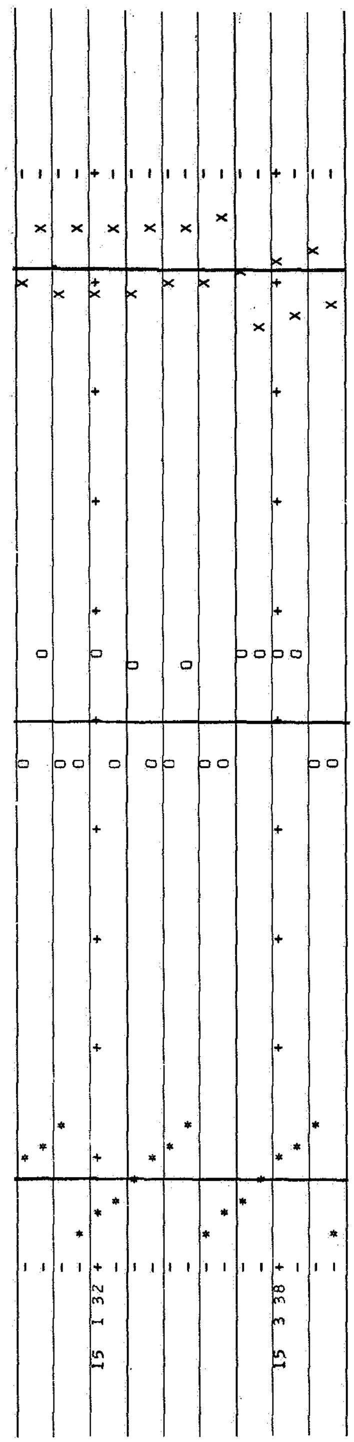


Fig. 5 (contd)

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.

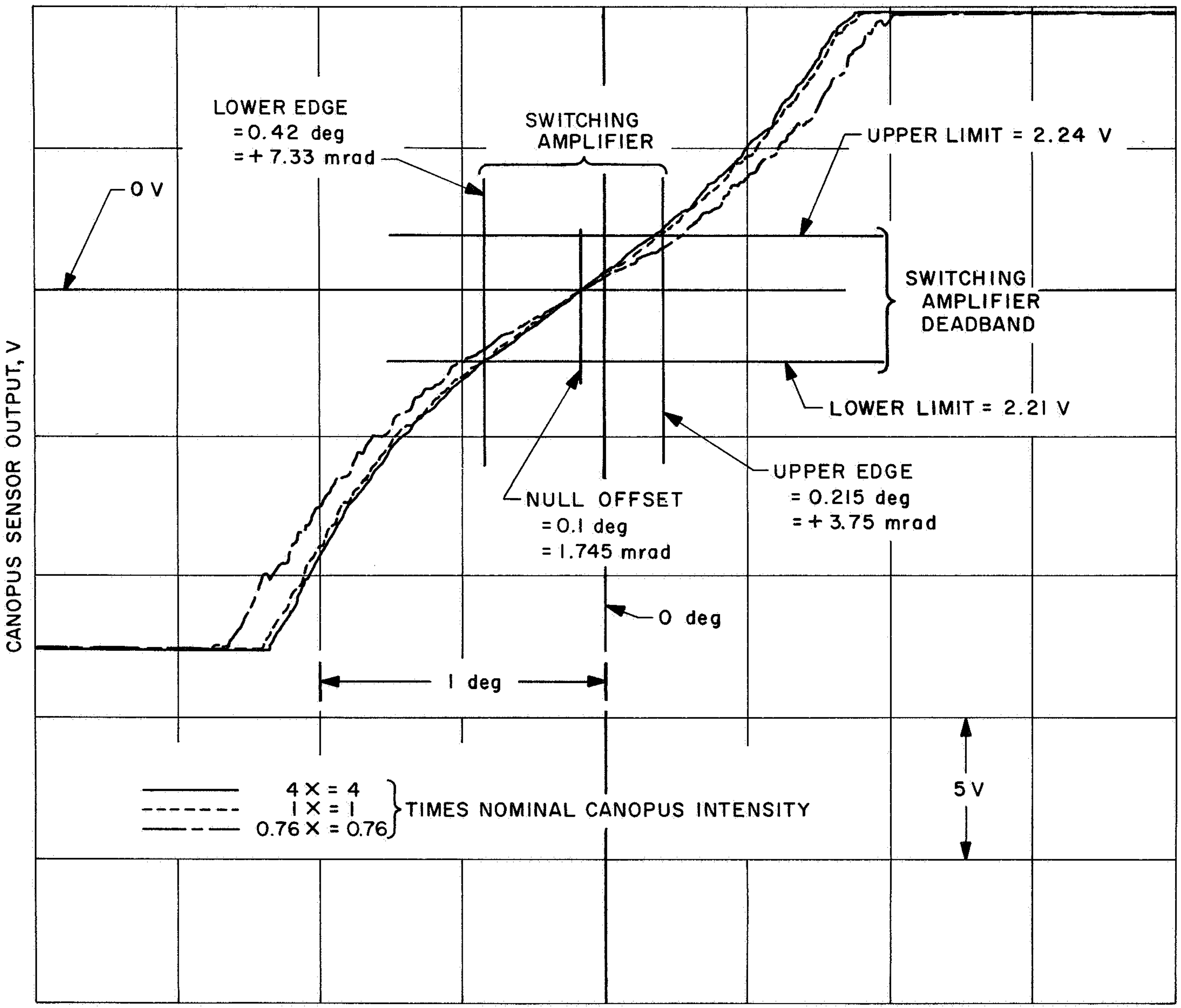


Fig. 6. Typical plot of Canopus sensor output vs angular displacement

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 33⅓ bits/s.

(2) 28 days (days 102–129 of 1965) of the *Mariner IV* mission at 8⅓ bits/s.

(3) 12 days (days 172–183 of 1967) of the *Mariner V* mission at 33⅓ bits/s.

(4) The changeover from 33⅓ to 8⅓ 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 33⅓ 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 33⅓ (8⅓) 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 ¼ 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

1. *Mariner Mars 1964 Project Report: Mission and Spacecraft Development: Vol. I. From Project Inception Through Midcourse Maneuver*, Technical Report 32-740, pp. 214–272. Jet Propulsion Laboratory, Pasadena, Calif., Mar. 1, 1968.
2. Shames, I. H., *Engineering Mechanics: Dynamics*, First Edition, p. 500. Prentice-Hall, Inc., Englewood, Cliffs, N. J., 1962.
3. Moore, R. E., *Interval Analysis*. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1966.

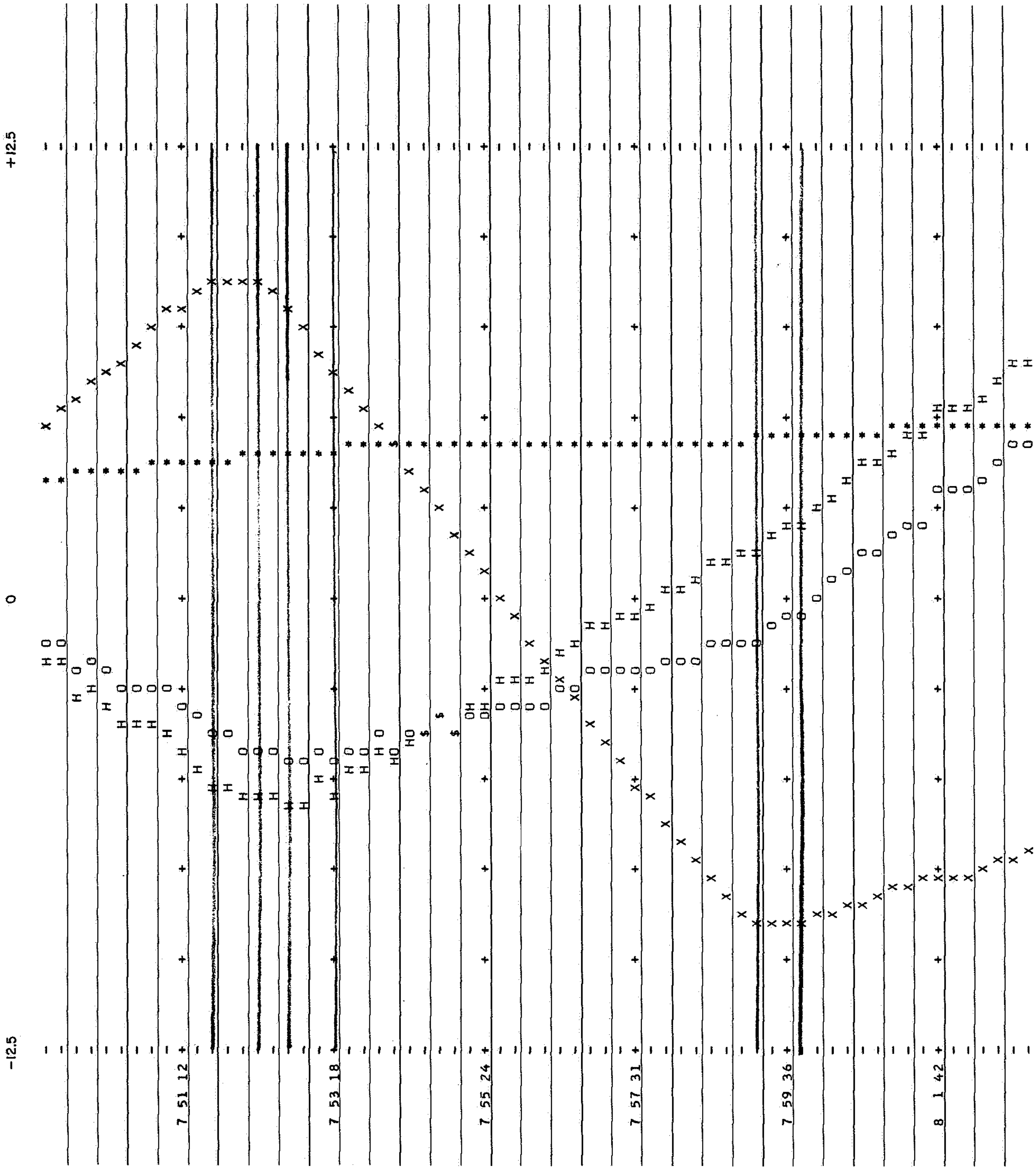


Fig. 7. Ambiguous data points in the limit cycles, *Mariner IV*, day 362, 1964. Time is in hours, minutes, and seconds (GMT). Symbols: * = pitch; X = yaw; O = V-axis roll; H = Z-axis or true roll; $ = superimposed *data* points

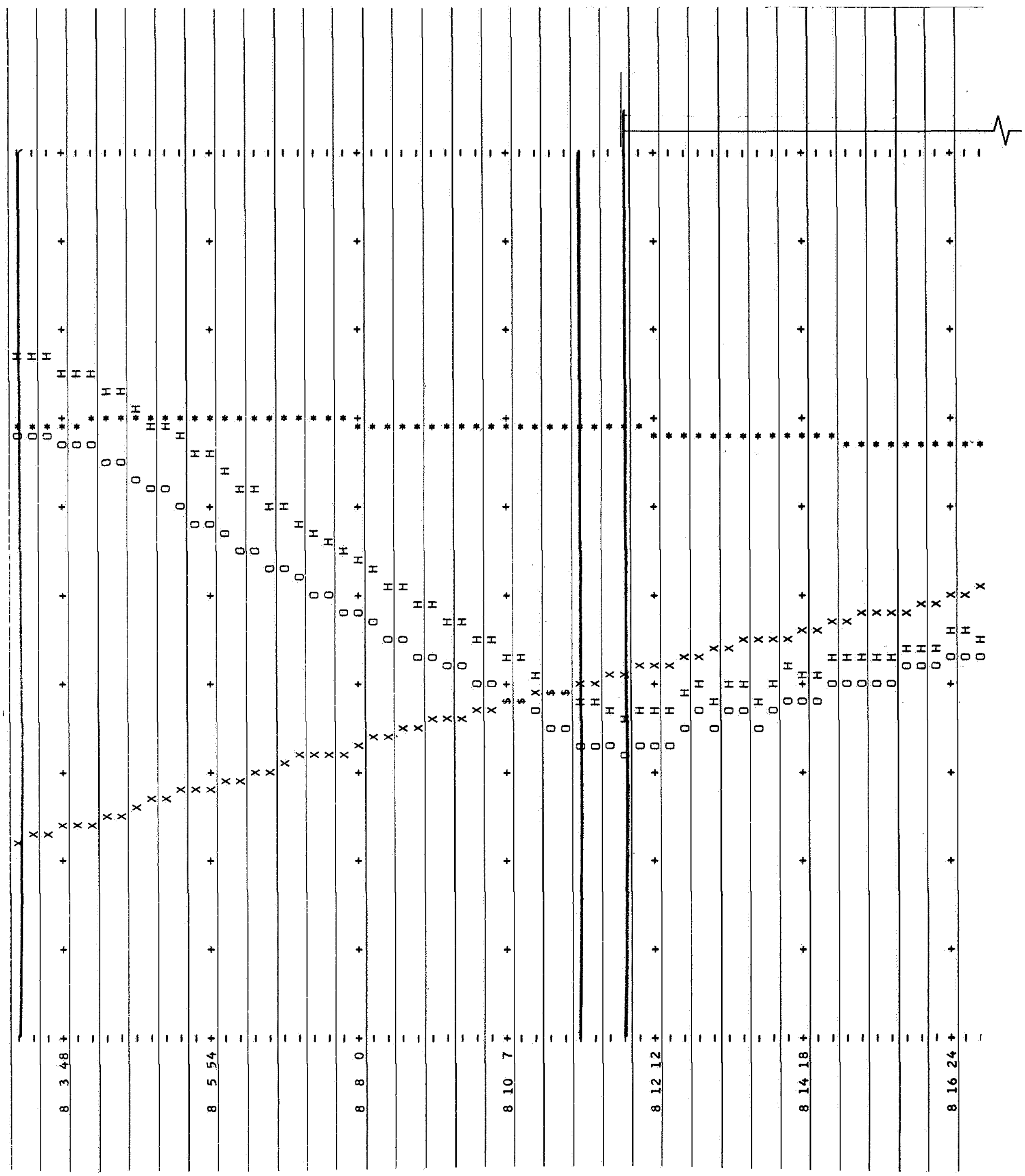


Fig. 7 (contd)

	PITCH	YAW	ROLL
TORQUE LIMITS ARE--UPPER	-0.36012650E 01	-0.43489794E 01	0.16078145E 02
LOWER	-0.64972879E 01	-0.62520215E 01	0.45471035E 01
TORQUE BY LEAST SQUARES	-0.48980332E 01	-0.53501711E 01	0.10291881E 02
RMS ERROR OF PARAB FIT	0.54473304E-01	0.52334471E-01	0.16465428E 00

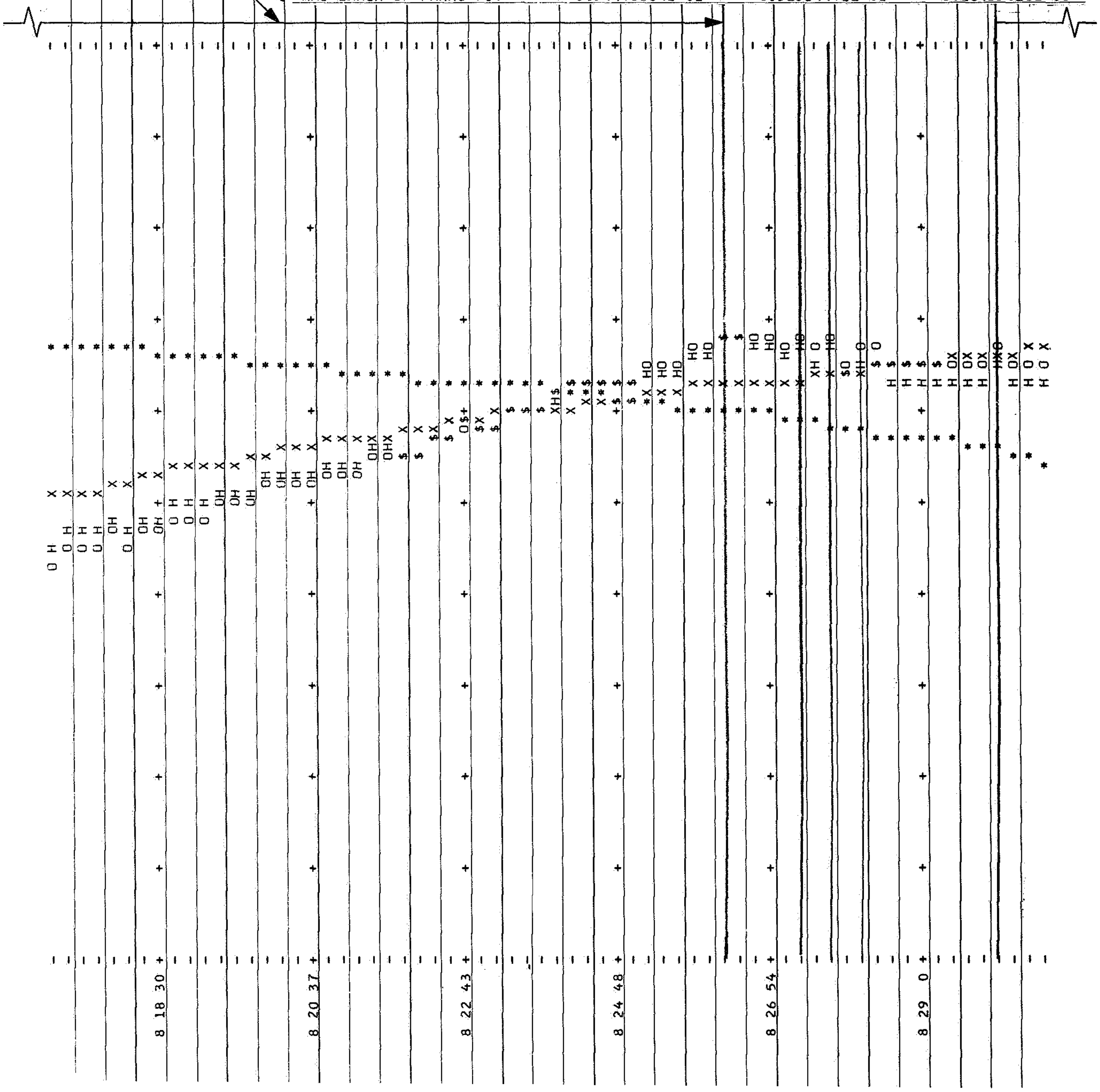


Fig. 7 (contd)

	PITCH	YAW	ROLL
TORQUE LIMITS ARE--UPPER	-0.25919197E 01	-0.52105136E 01	0.10707671E 02
LOWER	-0.38377493E 01	-0.59298938E 01	0.76458724E 01
TORQUE BY LEAST SQUARES	-0.32983256E 01	-0.57002417E 01	0.61572836E 01
RMS ERROR OF PARAB FIT	0.58357663E-01	0.59399258E-01	0.17392722E 00

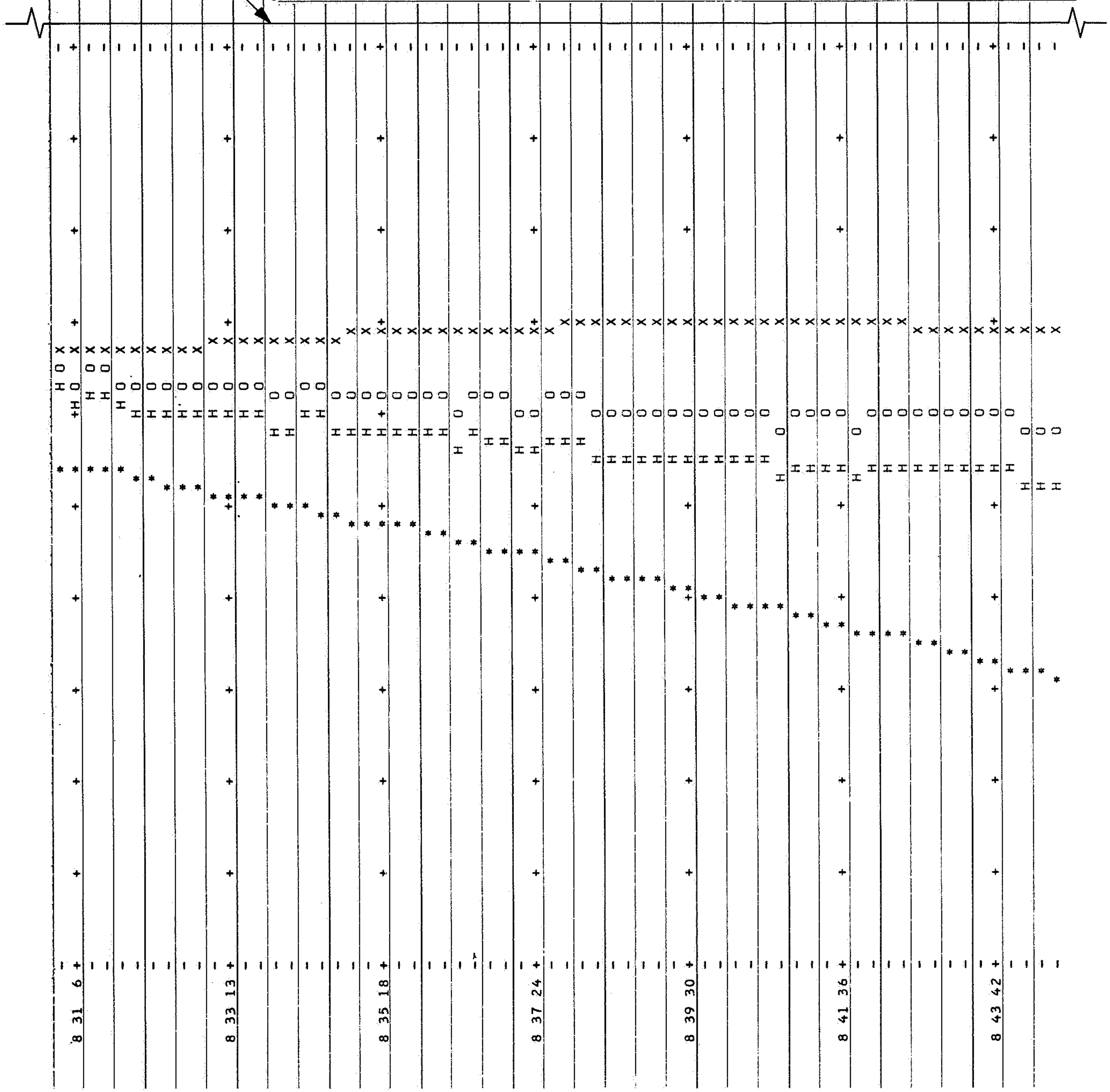


Fig. 7 (contd)

	PITCH	YAW	ROLL
TORQUE LIMITS ARE--UPPER	-0.39839311E 01	-0.46797724E 01	0.94632511E 01
LOWER	-0.43831413E 01	-0.51269926E 01	0.79584640E 01
TORQUE BY LEAST SQUARES	-0.41153100E 01	-0.49121233E 01	0.68773236E 01
RMS ERROR OF PARAB FIT	0.60752930E-01	0.60543466E-01	0.18004663E 00

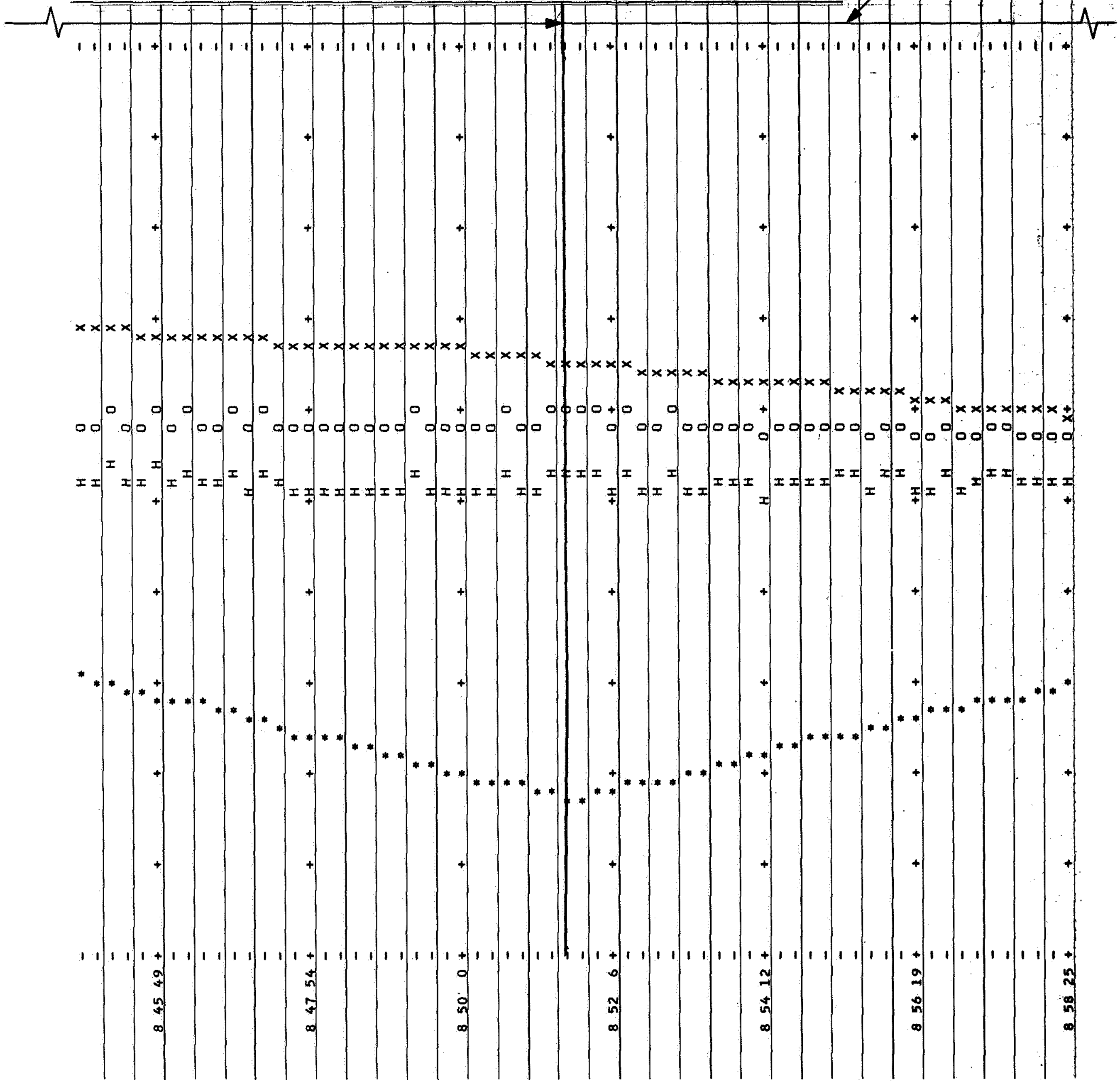


Fig. 7 (contd)

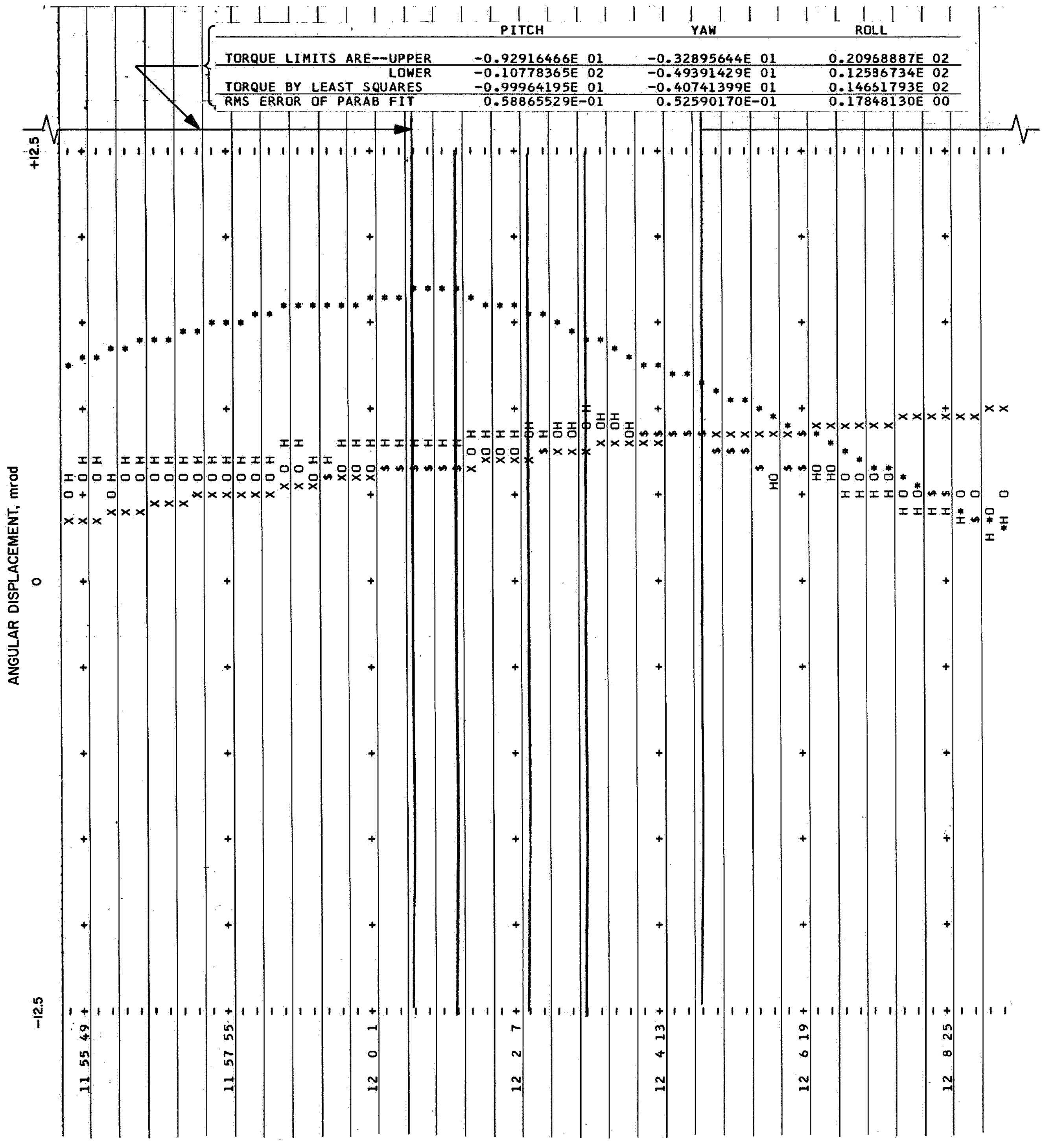


Fig. 8. Improper detection of attitude control thruster firing, *Mariner IV*, day 366, 1964. Time is in hours, minutes, and seconds (GMT). Symbols: * = pitch; X = yaw; O = V-axis roll; H = Z-axis or true roll; $ = superimposed data points

	PITCH	YAW	ROLL
TORQUE LIMITS ARE--UPPER	-0.53306663E 01	-0.18565242E 01	0.12324791E 02
LOWER	-0.54978200E 01	-0.54815487E 01	0.66425580E 01
TORQUE BY LEAST SQUARES	-0.60459032E 01	-0.37794736E 01	0.10573254E 02
RMS ERROR OF PARAB FIT	0.92255428E-01	0.61329259E-01	0.17942433E 00

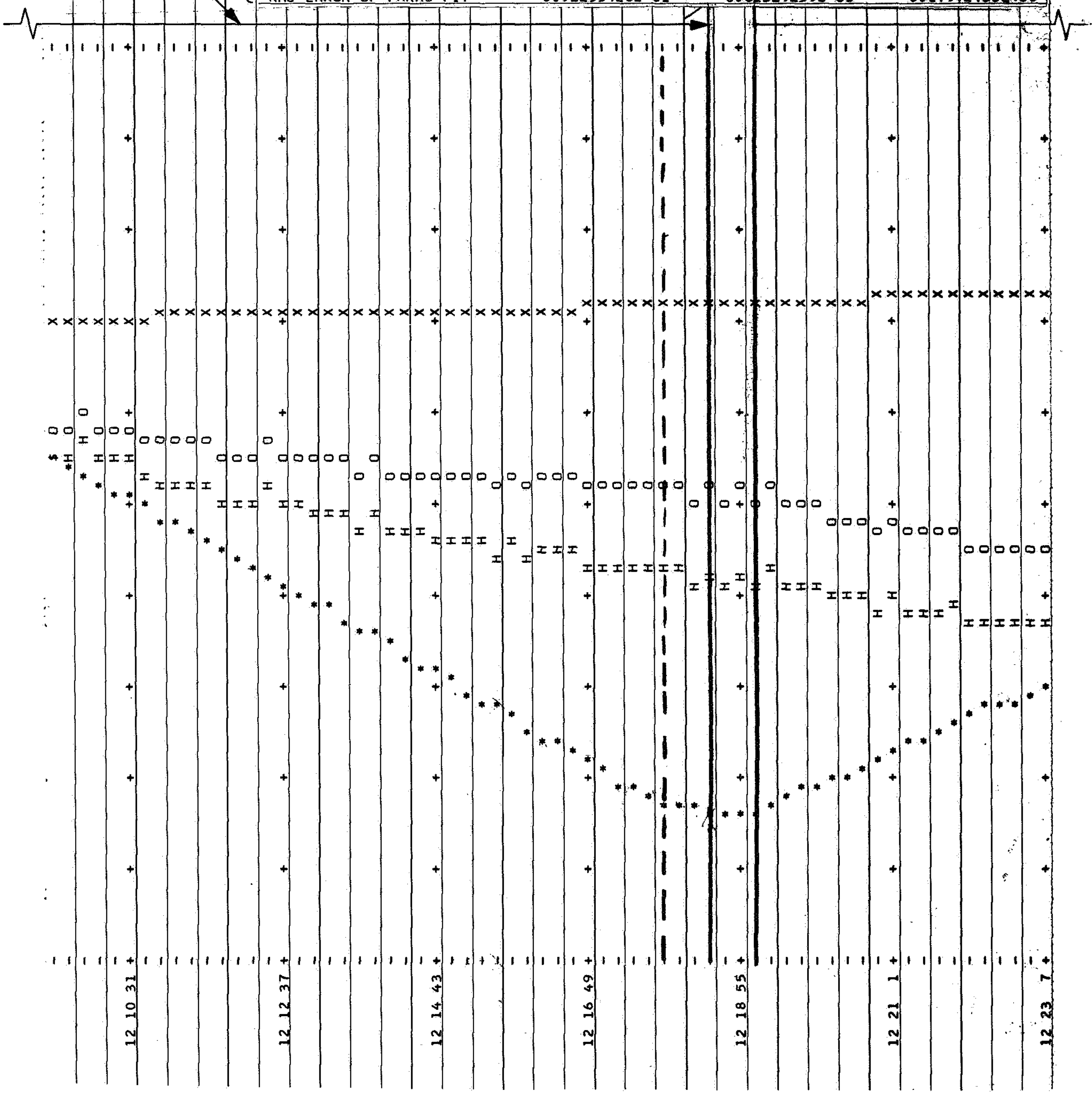


Fig. 8 (contd)

	PITCH	YAW	ROLL
TORQUE LIMITS ARE--UPPER	-0.88982958E 01	-0.39986955E 01	0.15799747E 02
LOWER	-0.93378593E 01	-0.45052115E 01	0.15401840E 02
TORQUE BY LEAST SQUARES	-0.90046277E 01	-0.41434072E 01	0.11648132E 02
RMS ERROR OF PARAB FIT	0.70858079E-01	0.57509277E-01	0.17679893E 00

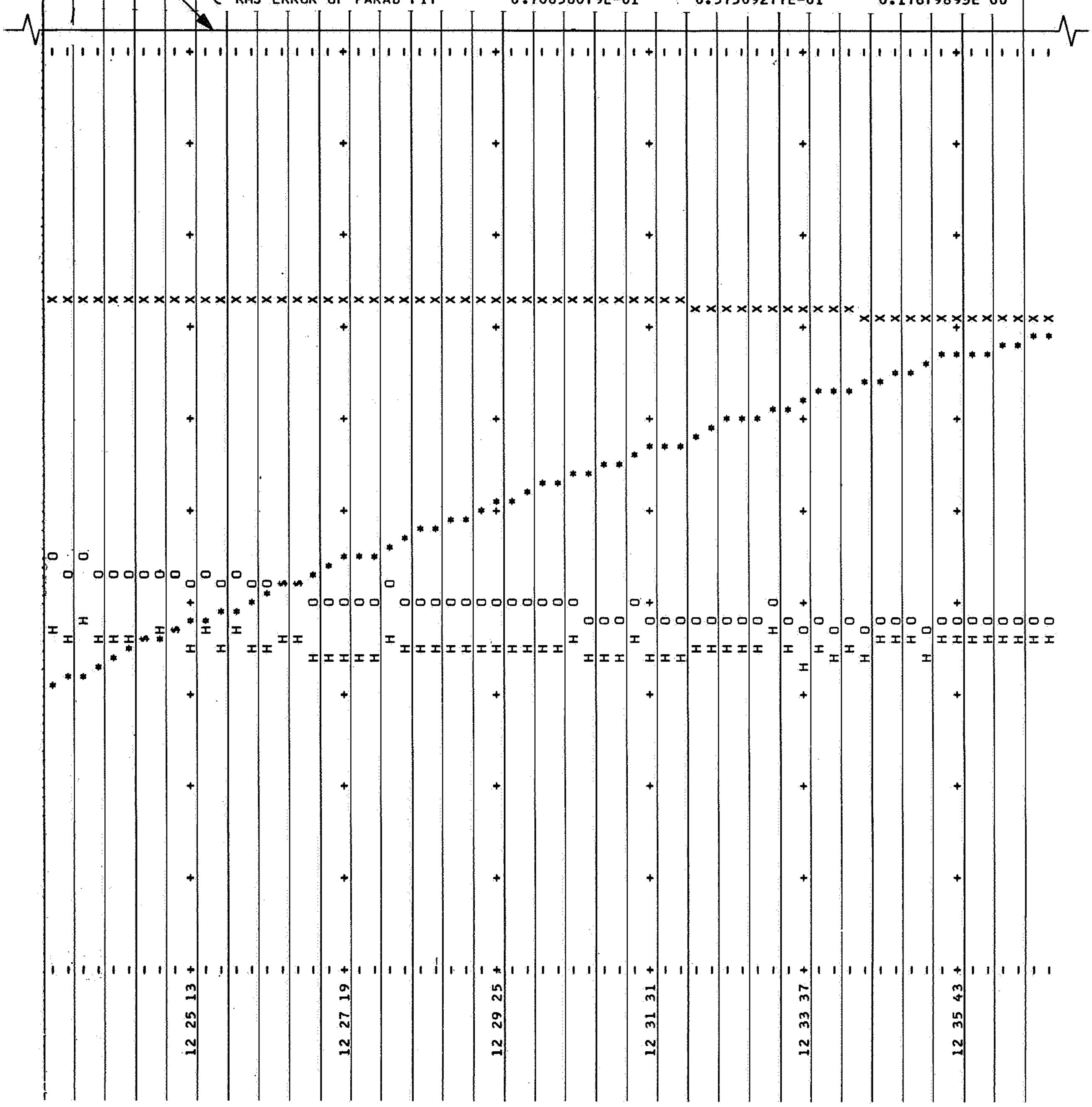


Fig. 8 (contd)

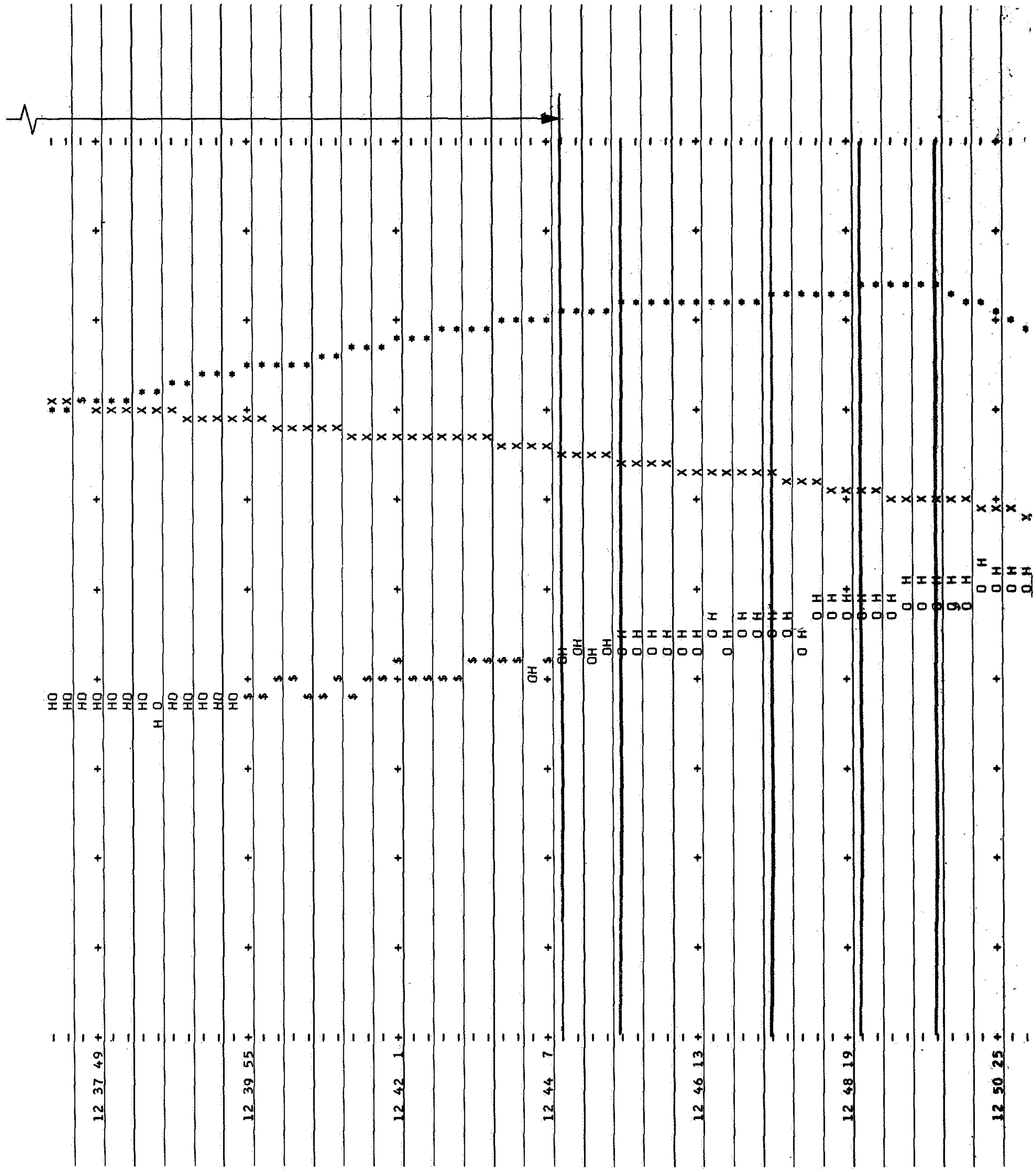


Fig. 8 (contd)

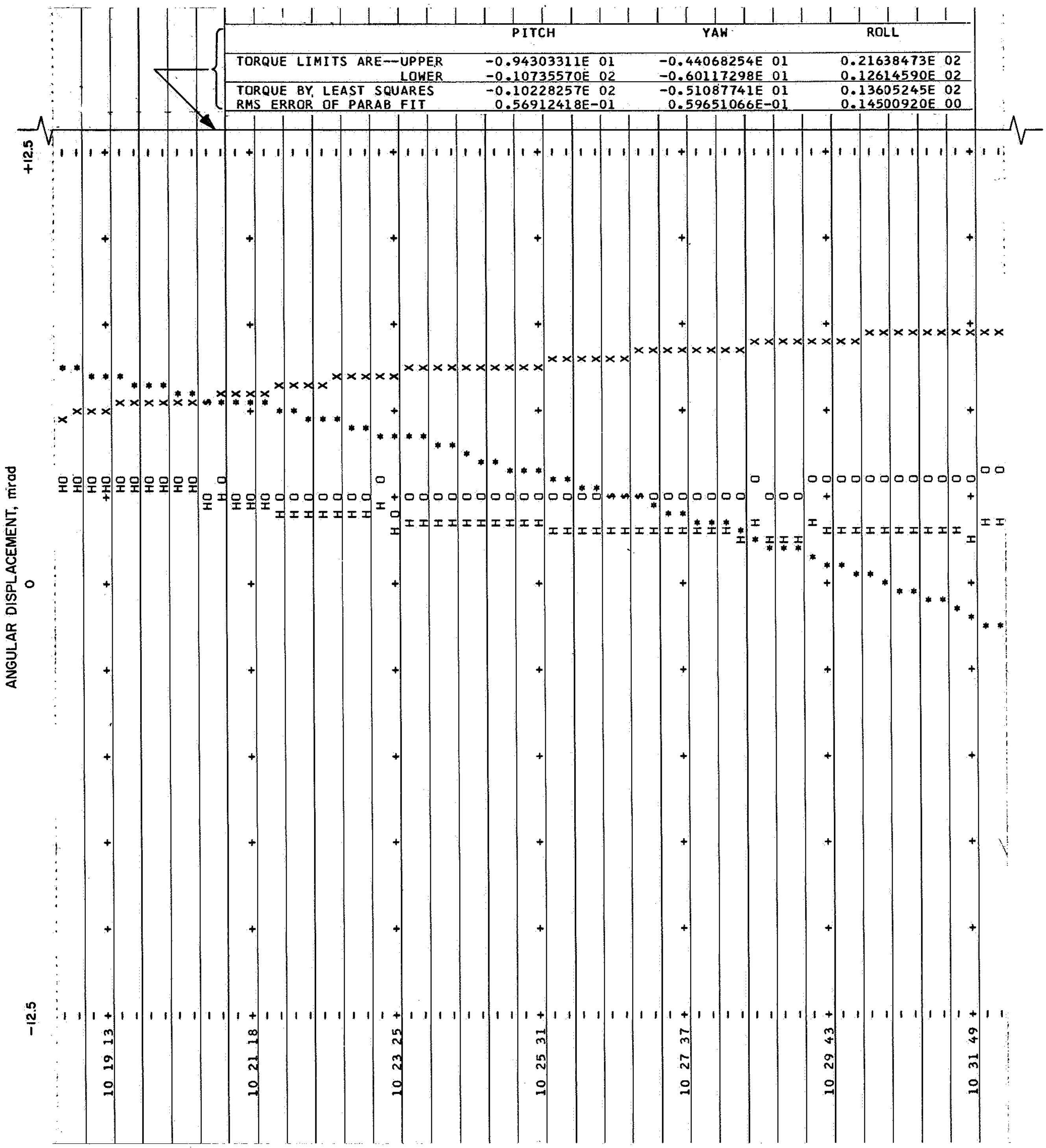


Fig. 9. Angular velocity change for a single thruster firing, *Mariner IV*, day 366, 1964. Time is in hours, minutes, and seconds (GMT). Symbols: * = pitch; X = yaw; O = V-axis roll; H = Z-axis roll; $ = superimposed data points

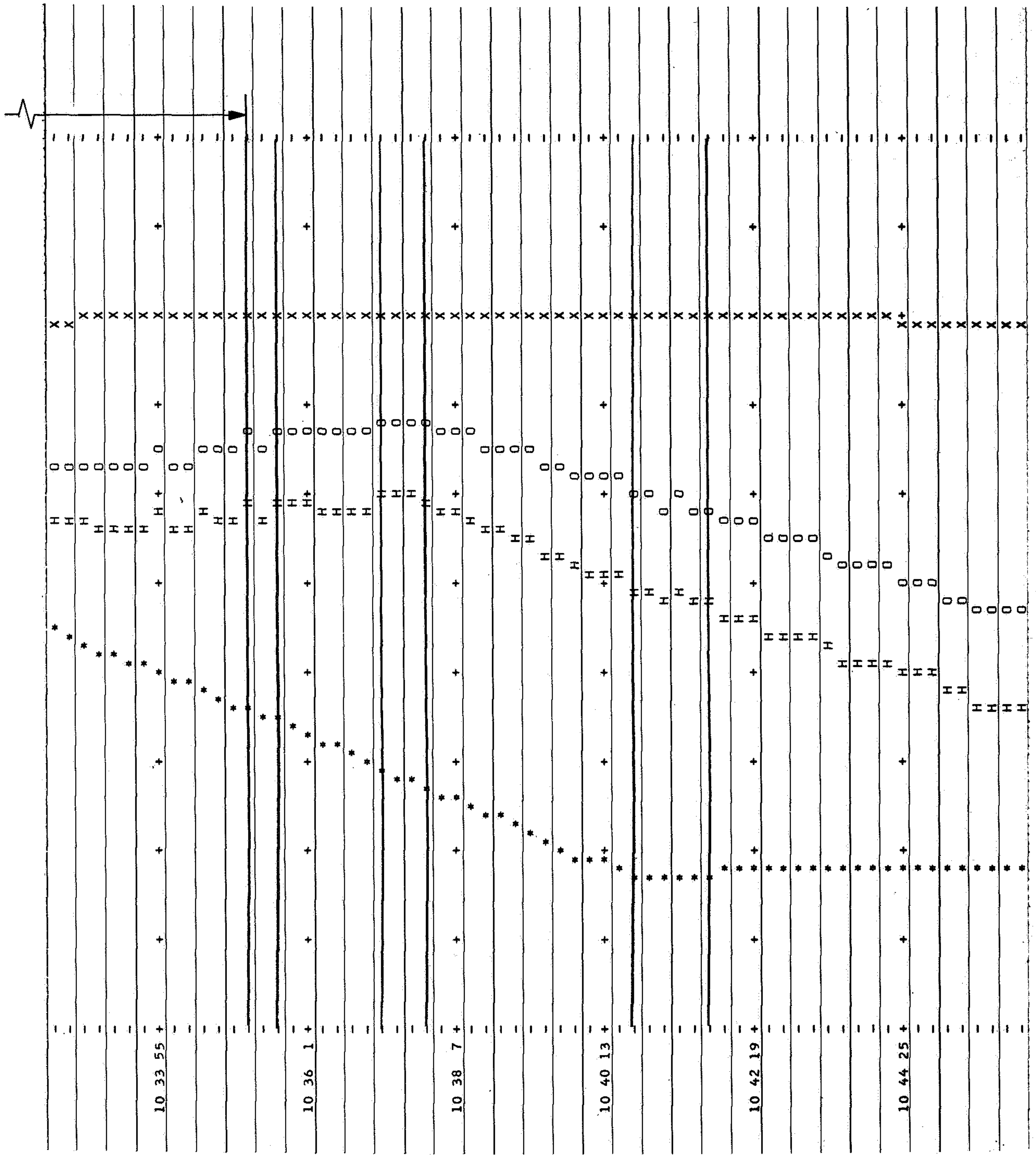


Fig. 9 (contd)

	PITCH	YAW	ROLL
TORQUE LIMITS ARE--UPPER	-0.80093015E 01	-0.46997963E 01	0.19787506E 02
LOWER	-0.10292994E 02	-0.60644476E 01	0.12446967E 02
TORQUE BY LEAST SQUARES	-0.87859795E 01	-0.51049834E 01	0.12010608E 02
RMS ERROR OF PARAB FIT	0.57123836E-01	0.57253269E-01	0.14765694E 00

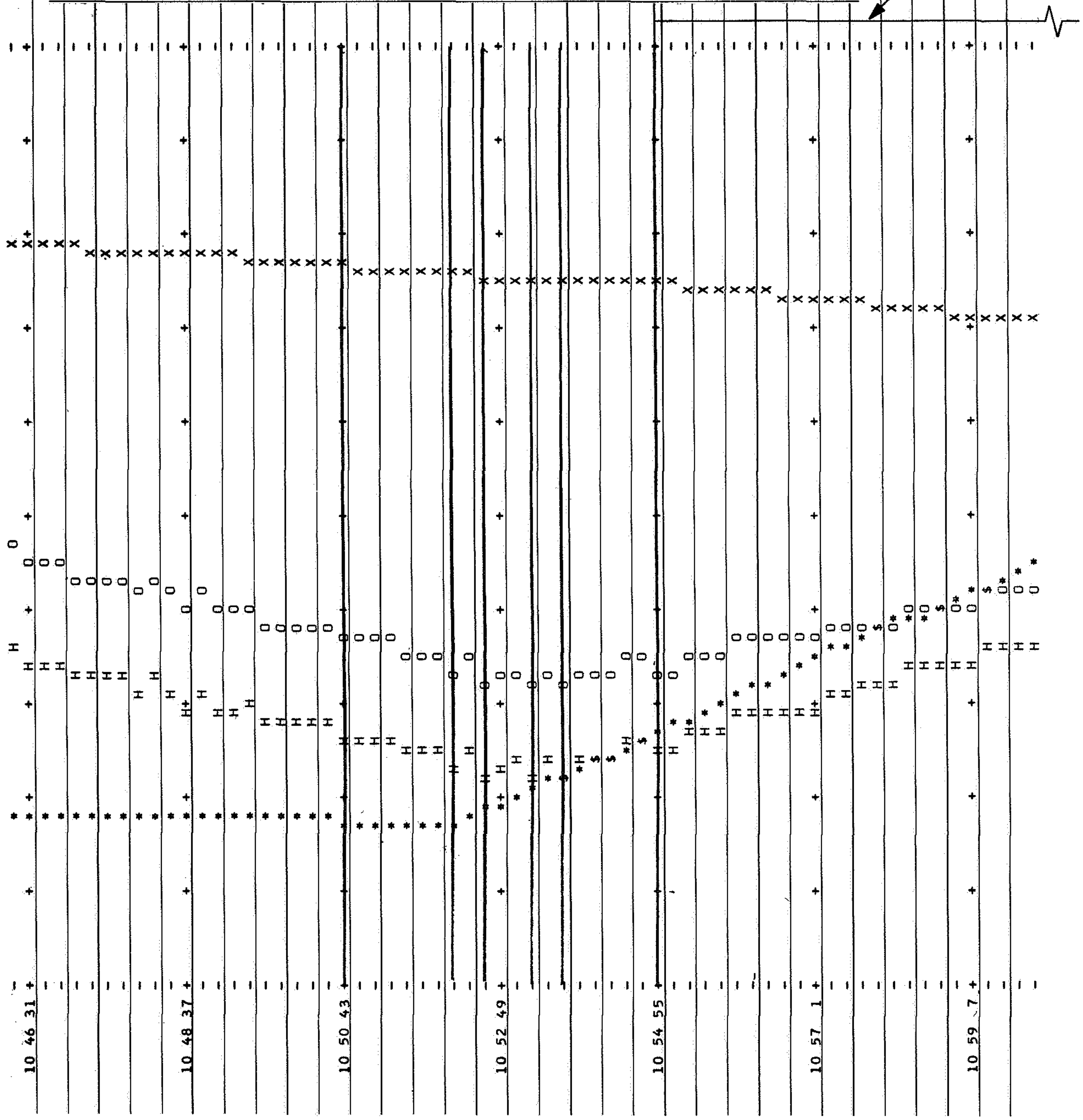


Fig. 9 (contd)

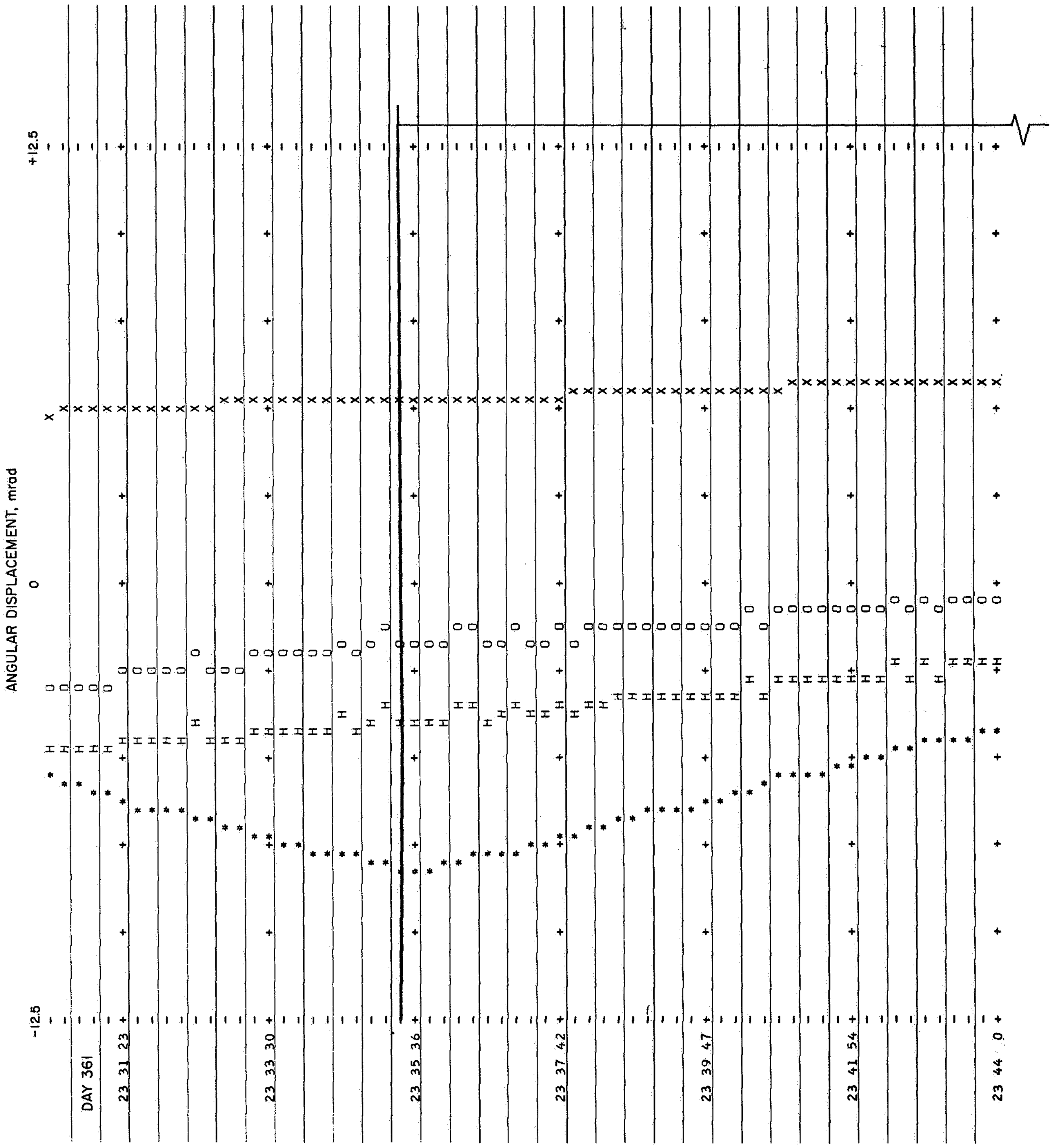


Fig. 10. Evidence of restoring torque, *Mariner IV*, days 361–362, 1964. Time is in hours, minutes, and seconds (GMT). Symbols: * = pitch; X = yaw; O = V-axis roll; H = Z-axis or true roll; $ = superimposed *data points*

	PITCH	YAW	ROLL
TORQUE LIMITS ARE--UPPER	-0.18449577E 01	-0.12632936E 01	0.10105822E 02
LOWER	-0.30045656E 01	-0.32073513E 01	0.65374747E 01
TORQUE BY LEAST SQUARES	-0.24363333E 01	-0.22901845E 01	0.65701225E 01
RMS ERROR OF PARAB FIT	0.56803676E-01	0.56711606E-01	0.15252318E 00

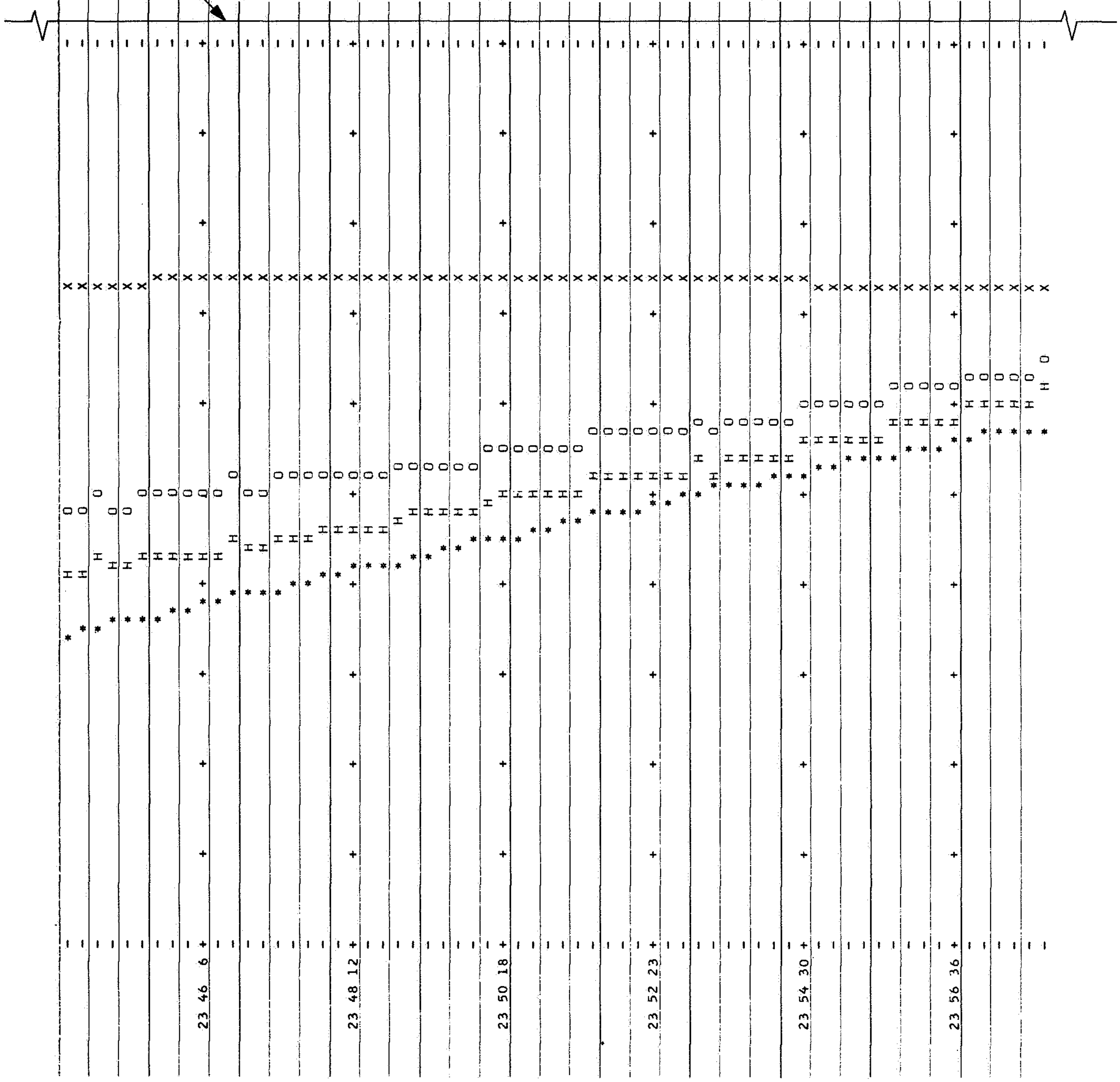


Fig. 10 (contd)

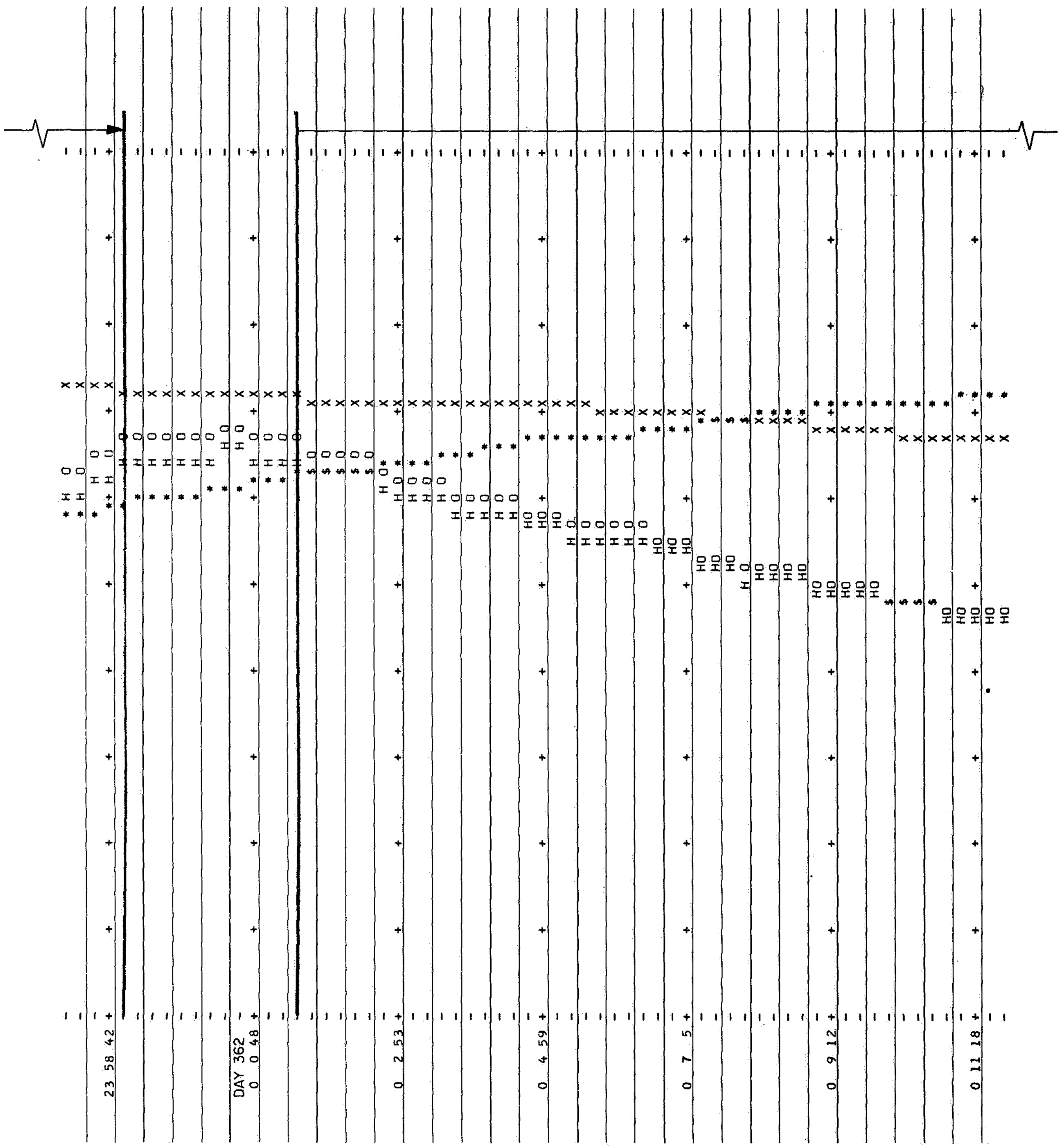


Fig. 10 (contd)

	PITCH	YAW	ROLL
TORQUE LIMITS ARE--UPPER	-0.29540440E 01	-0.19023840E 01	0.10882467E 02
LOWER	-0.40543171E 01	-0.30569675E 01	0.63311093E 01
TORQUE BY LEAST SQUARES	-0.34821788E 01	-0.24284536E 01	0.66958079E 01
RMS ERRUR OF PARAB FIT	0.58864983E-01	0.56122658E-01	0.17087475E 00

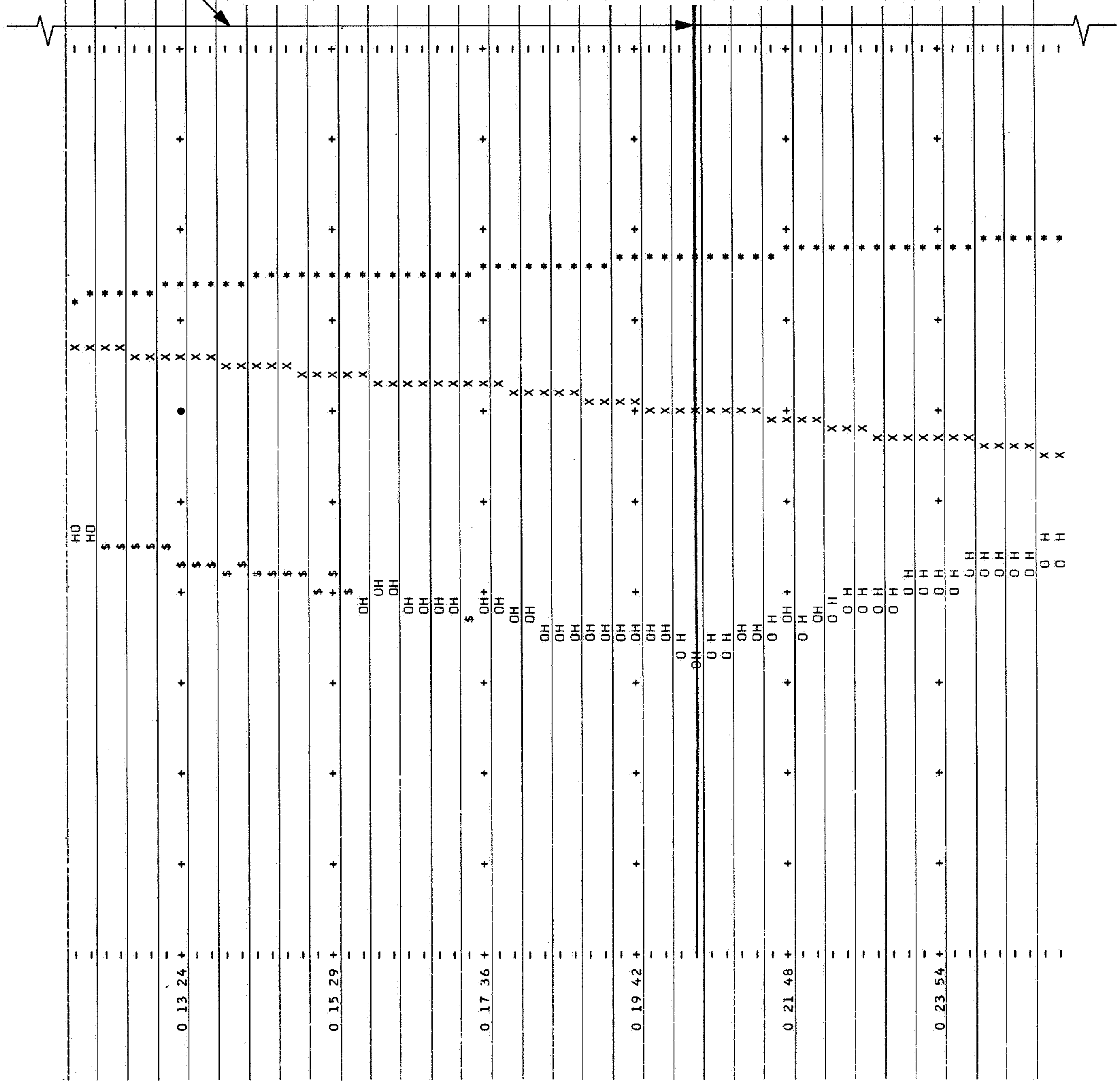


Fig. 10 (contd)

	PITCH	YAW	ROLL
TURQUE LIMITS ARE--UPPER	-0.68567010E-01	-0.95117522E 00	0.11258779E 02
LOWER	-0.51172686E 01	-0.30344956E 01	0.85287110E 01
TURQUE BY LEAST SQUARES	-0.28755690E 01	-0.19639487E 01	0.94653196E 01
RMS ERROR OF PARAB FIT	0.49003376E-01	0.53135793E-01	0.18249243E 00

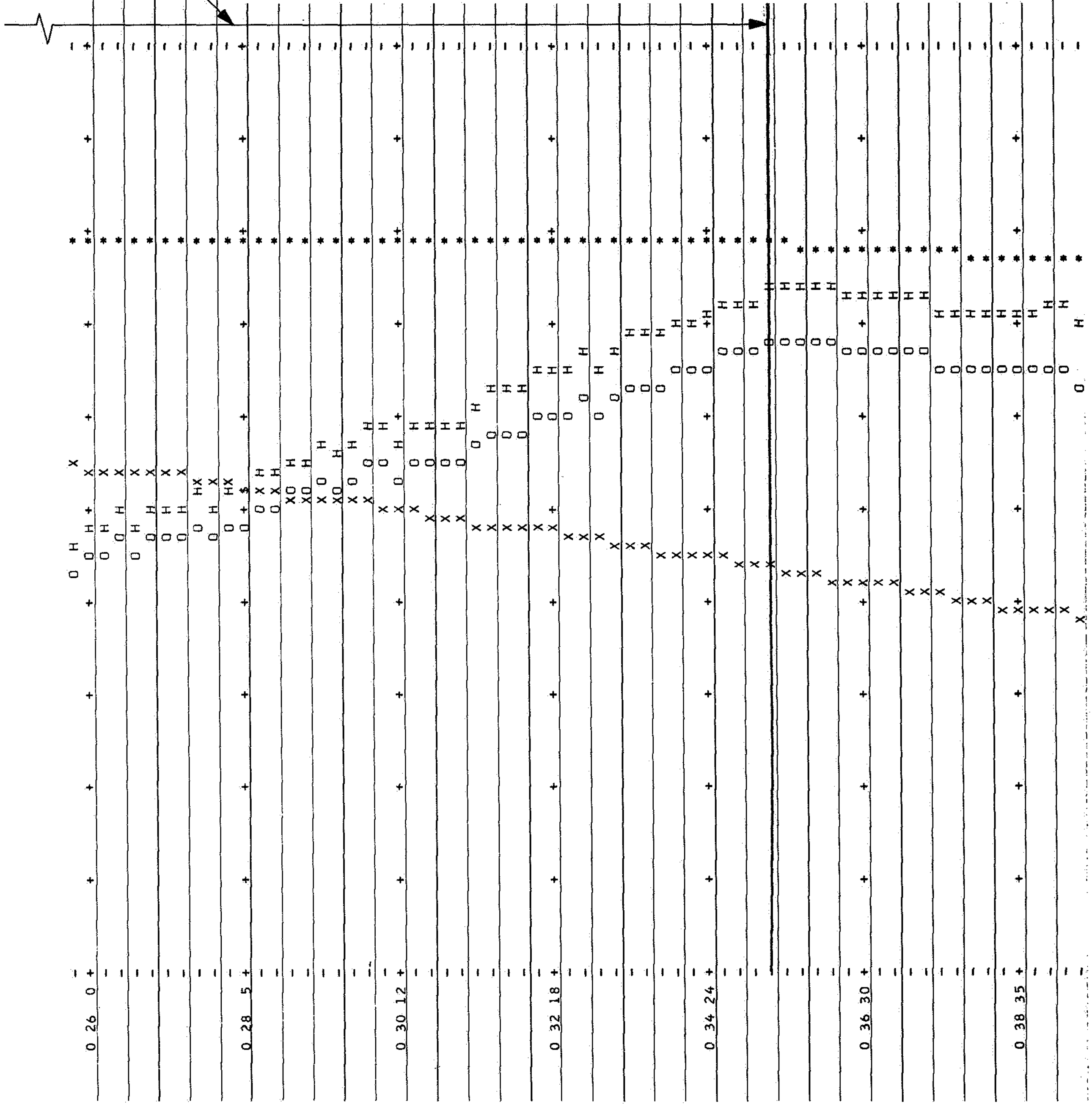


Fig. 10 (contd)

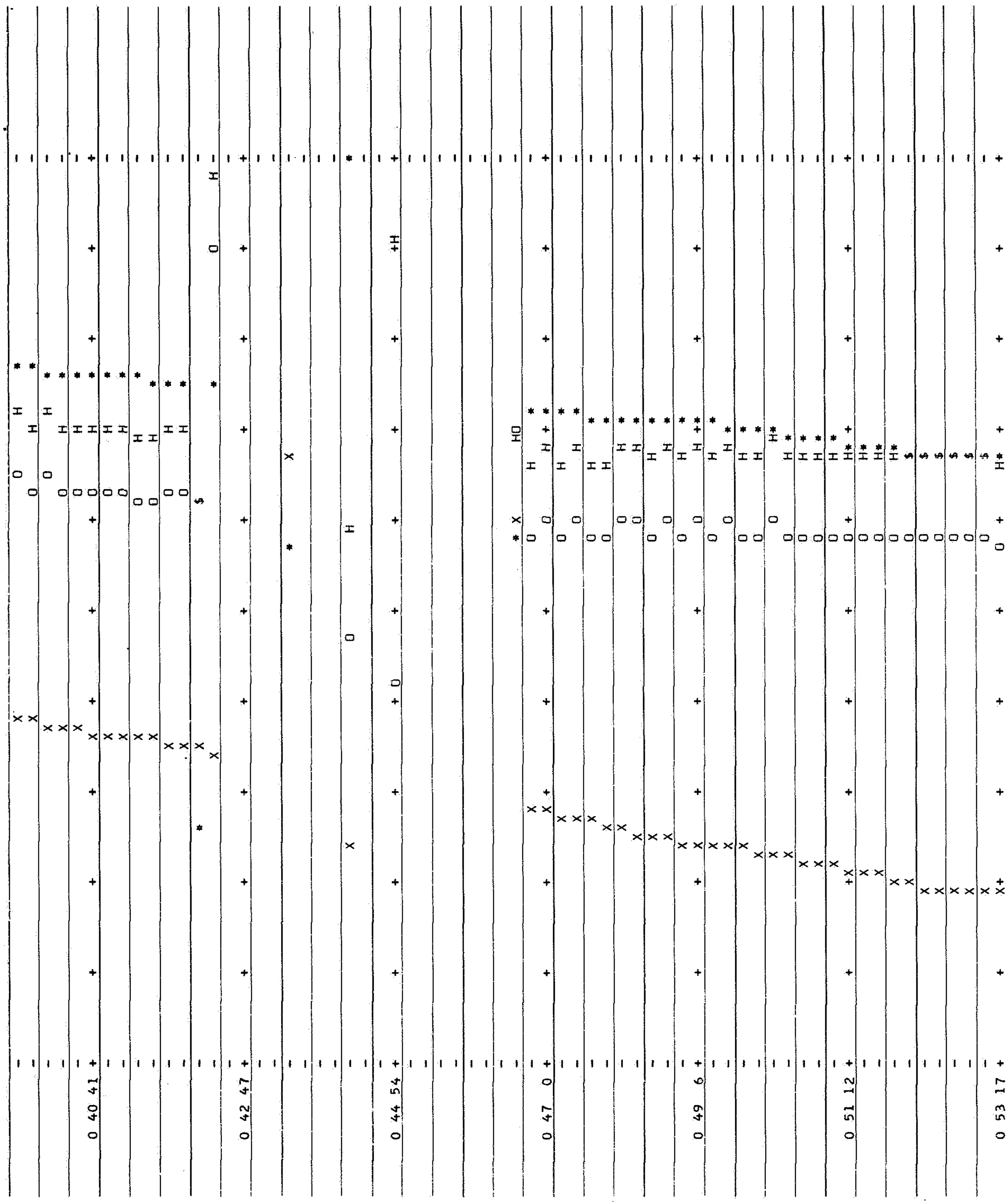


Fig. 10 (contd)

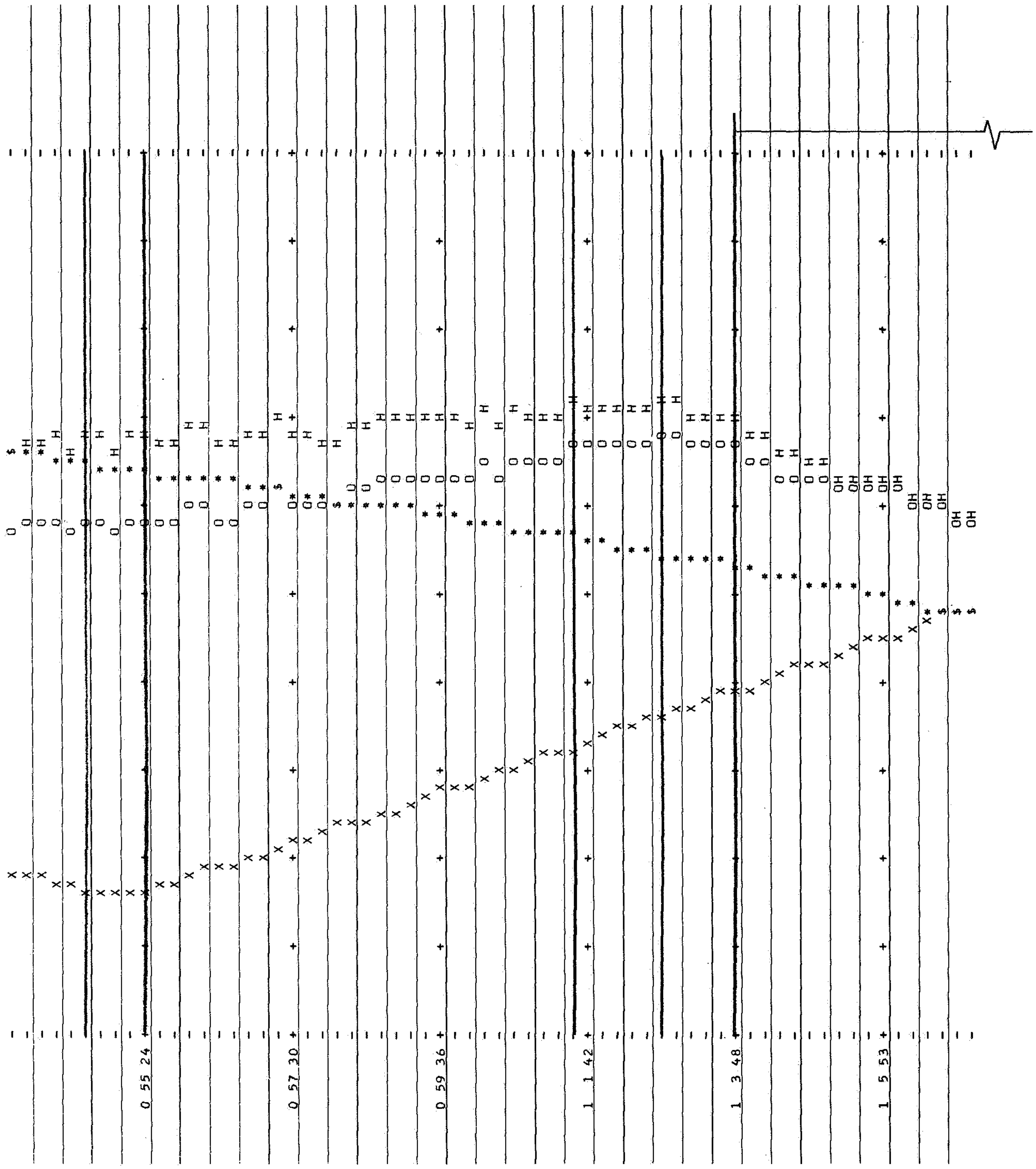


Fig. 10 (contd)

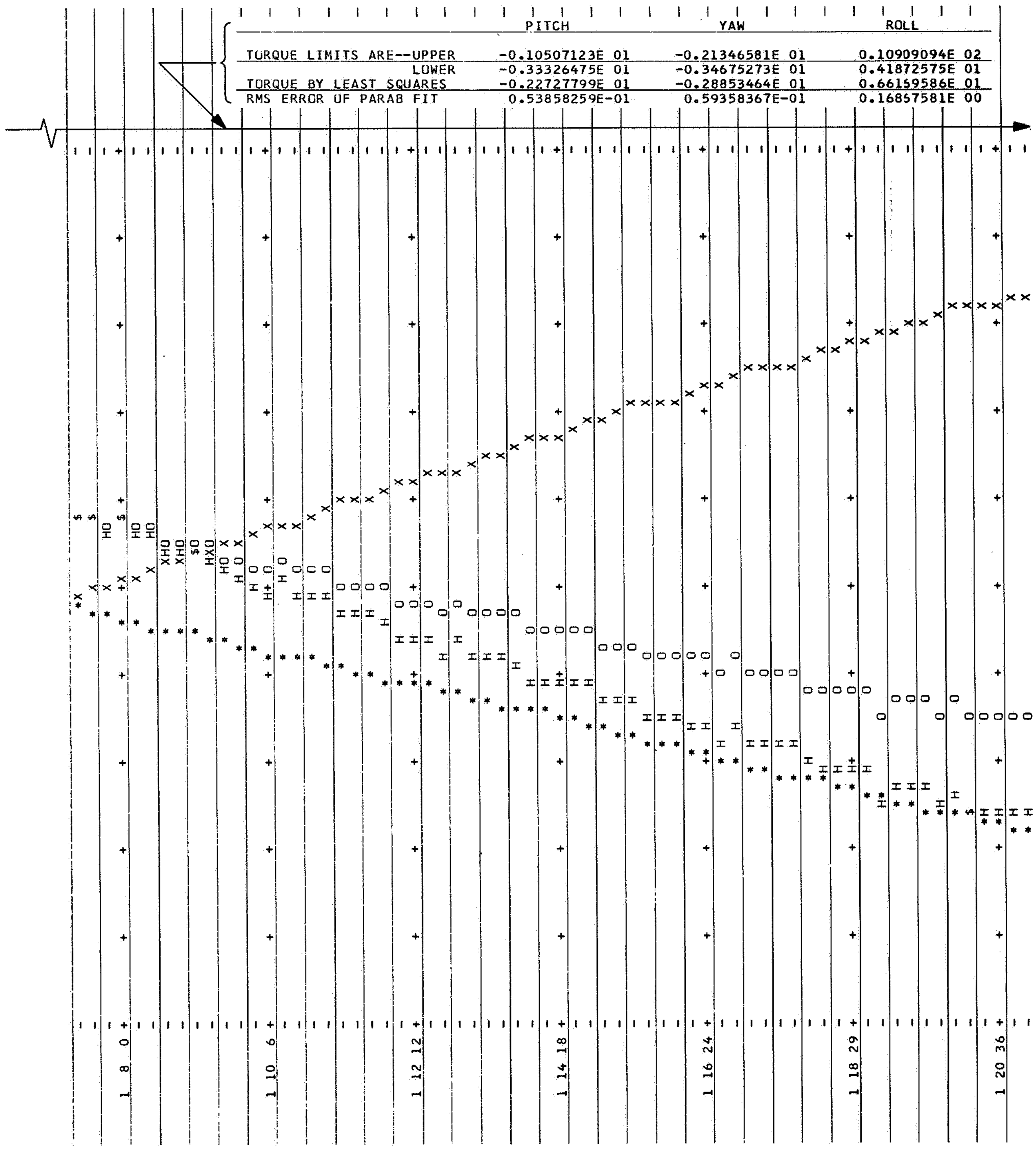

PITCH
YAW
ROLL
TORQUE LIMITS ARE--UPPER -0.10507123E 01 -0.21346581E 01 0.10909094E 02
LOWER -0.33326475E 01 -0.34675273E 01 0.41872575E 01
TORQUE BY LEAST SQUARES -0.22727799E 01 -0.28853464E 01 0.66159586E 01
RMS ERROR OF PARAB FIT 0.53858259E-01 0.59358367E-01 0.16857581E 00
1 8 0
1 10 6
1 12 12
1 14 18
1 16 24
1 18 29
1 20 36

Fig. 10 (contd)

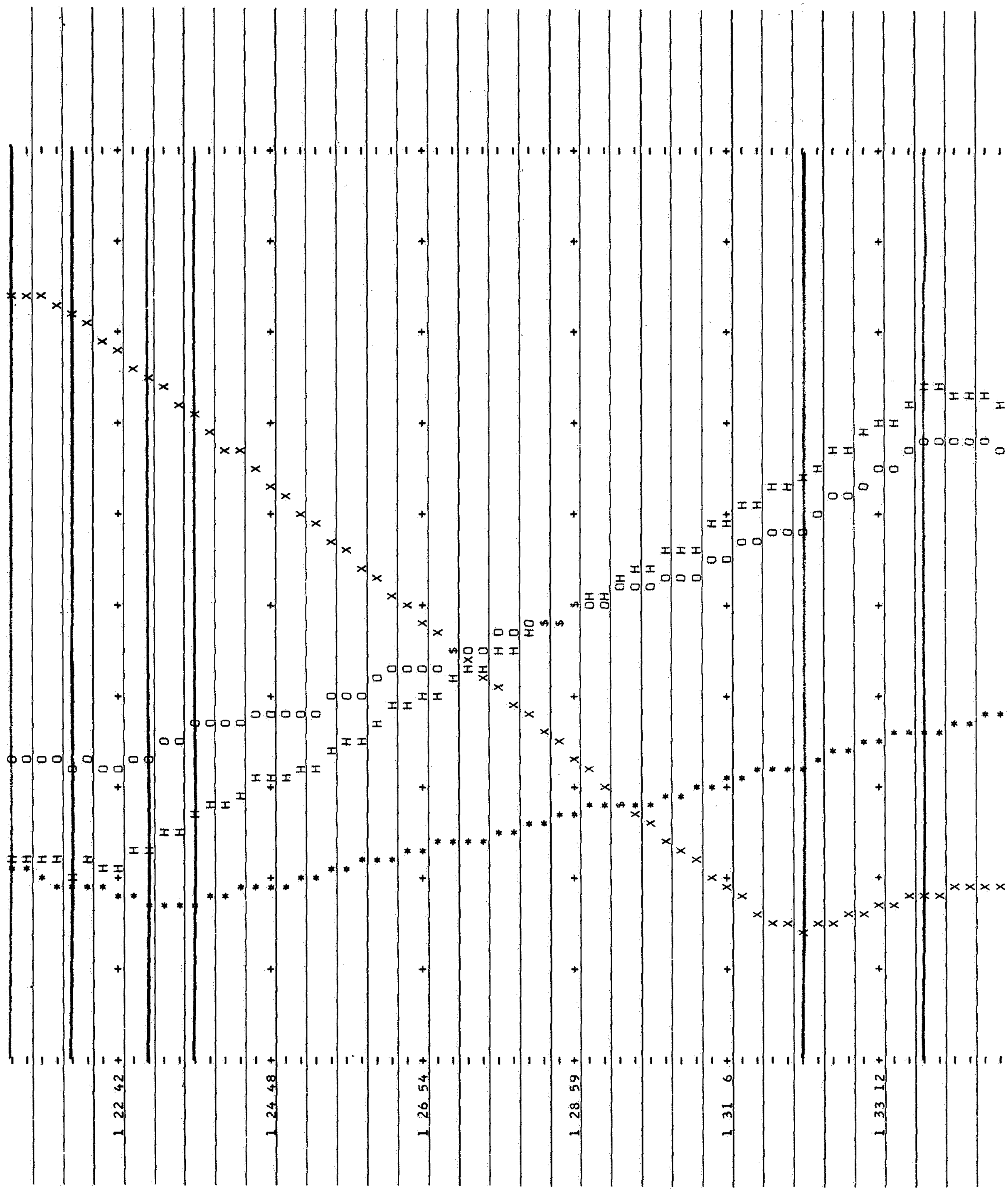


Fig. 10 (contd)

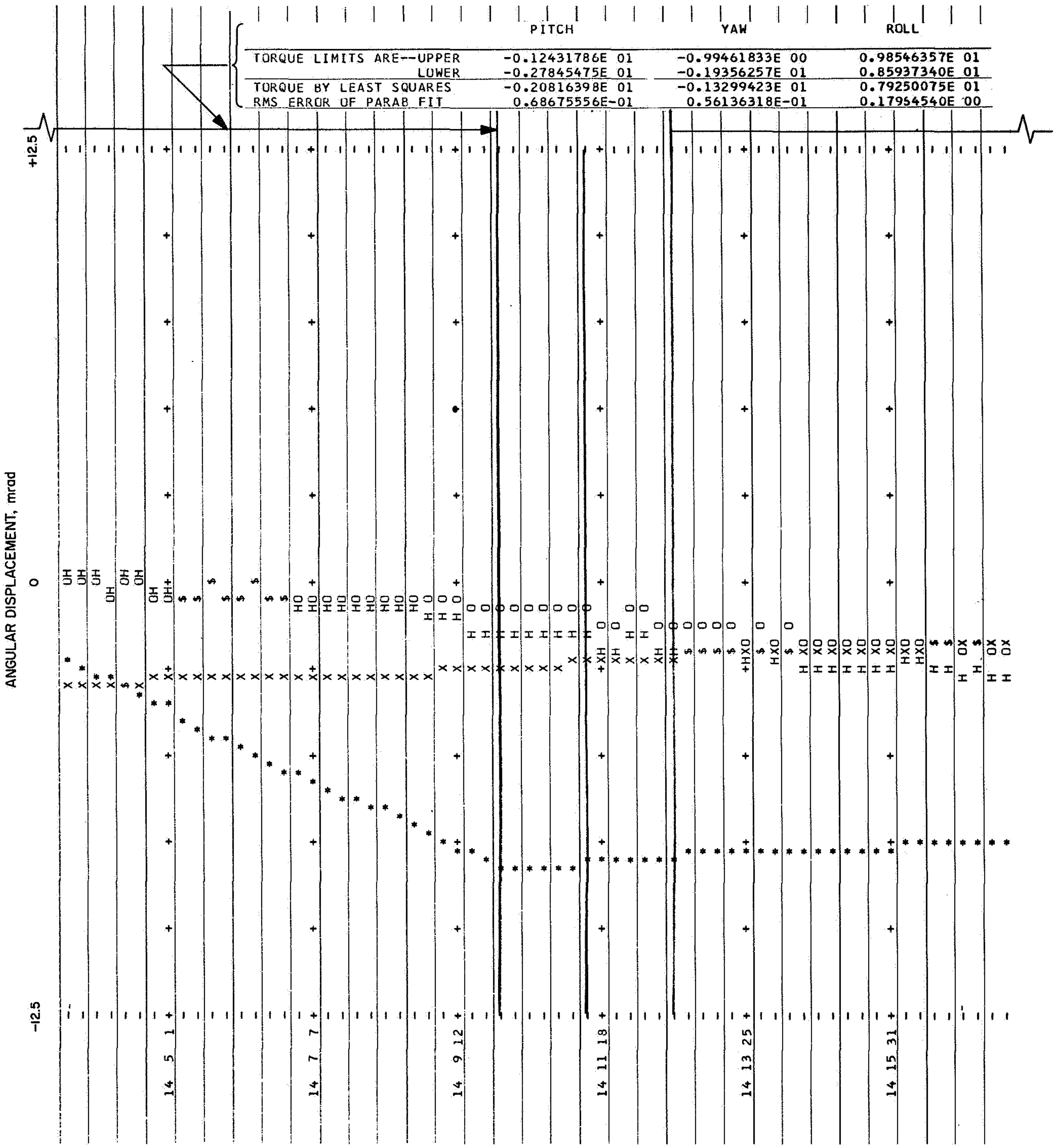


Fig. 11. Torque change accompanying valve firing, *Mariner IV*, day 362, 1964. Time is in hours, minutes, and seconds (GMT). Symbols: * = pitch; X = yaw; O = V-axis roll; H = Z-axis or true roll; $ = superimposed data points

	PITCH	YAW	ROLL
TORQUE LIMITS ARE--UPPER	-0.19330177E 01	-0.91554321E 00	0.82245616E 01
LOWER	-0.23650571E 01	-0.11534468E 01	0.71201401E 01
TORQUE BY LEAST SQUARES	-0.21804808E 01	-0.10590380E 01	0.67421949E 01
RMS ERROR OF PARAB FIT	0.56645254E-01	0.46956542E-01	0.16305994E 00

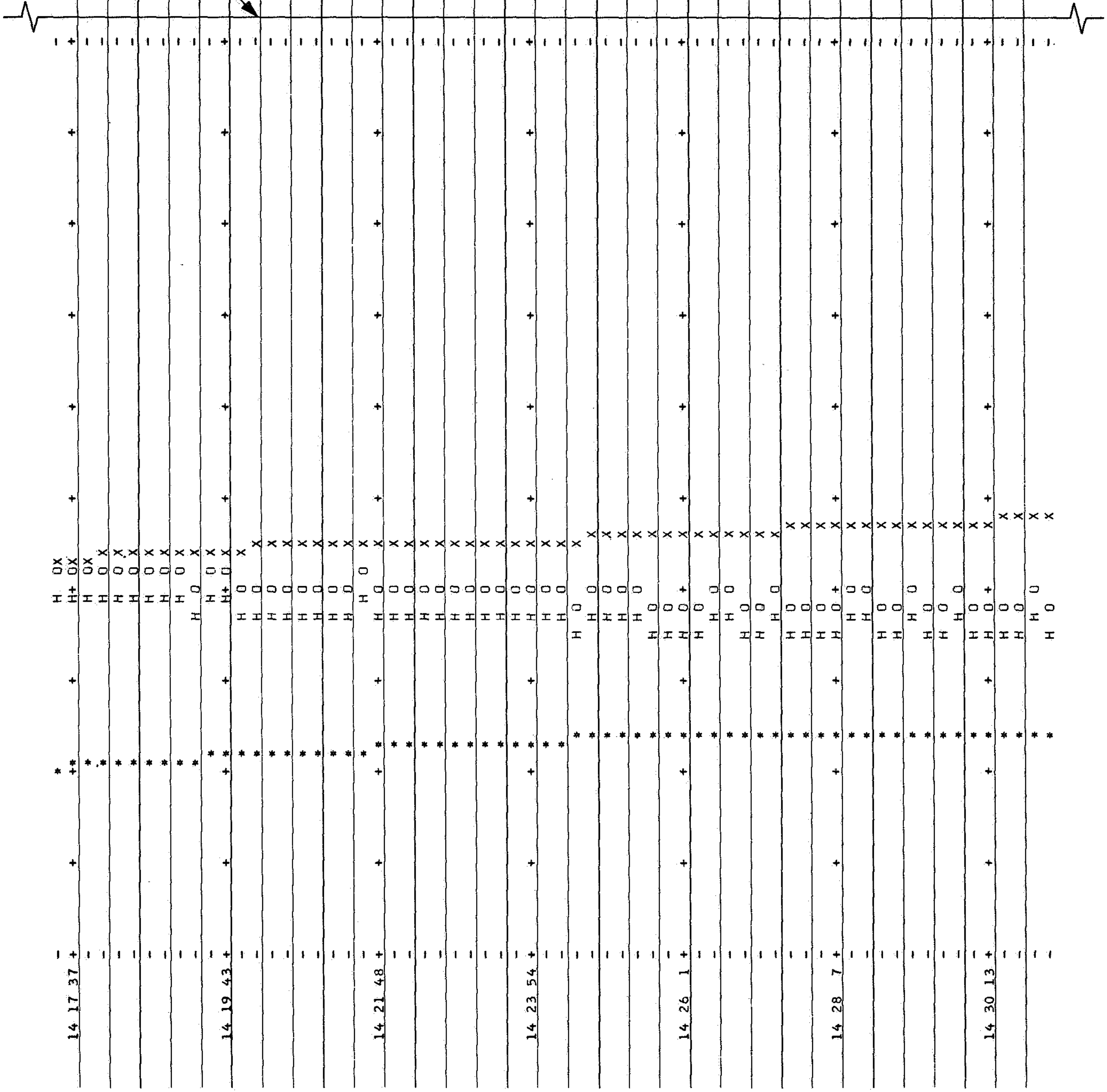


Fig. 11 (contd)

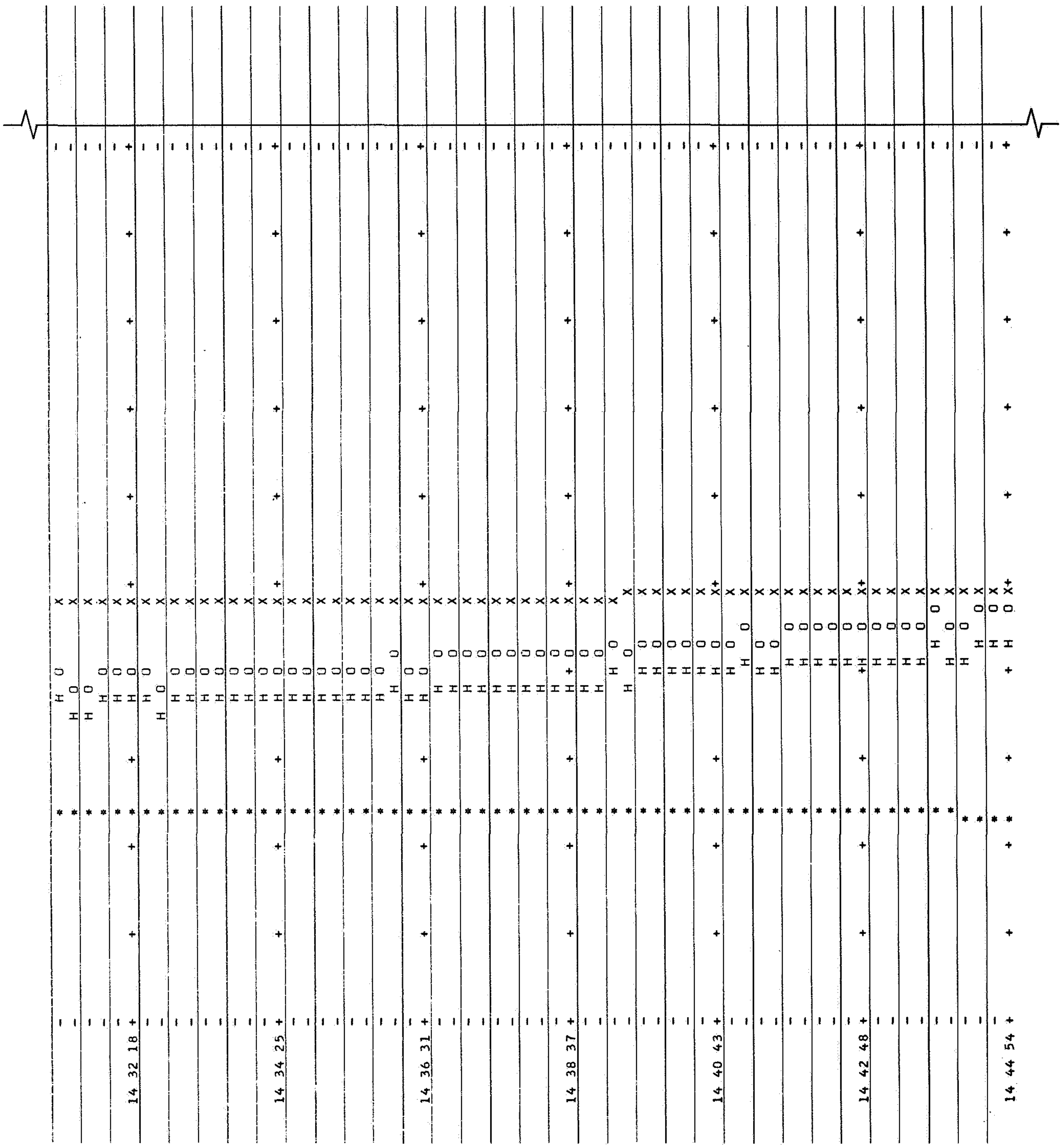


Fig. 11 (contd)

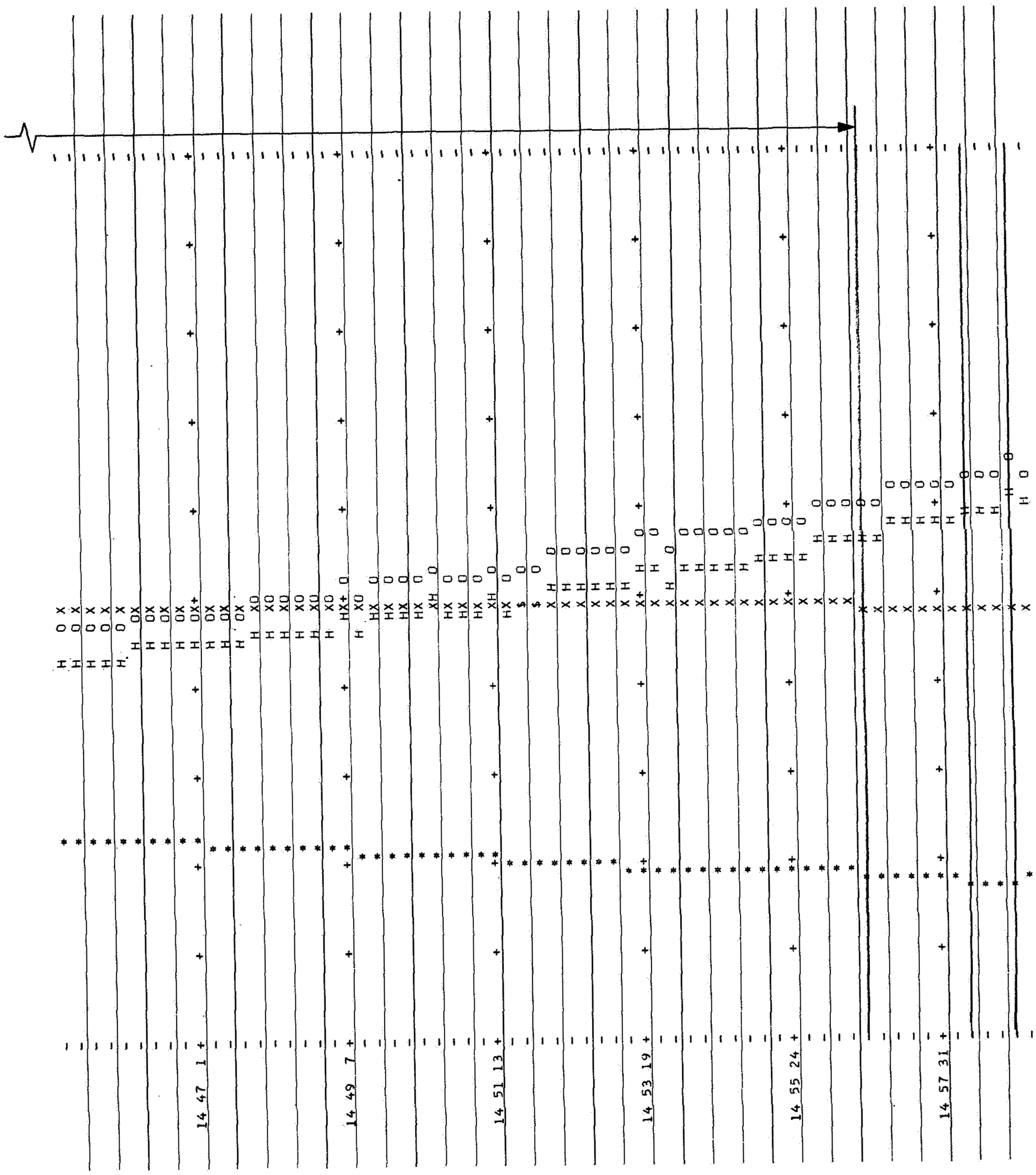


Fig. 11 (contd)

	PITCH	YAW	ROLL
TORQUE LIMITS ARE--UPPER	-0.41470082E 01	-0.20875617E 00	0.14712944E 02
LOWER	-0.63894132E 01	-0.28524495E 01	0.62353487E 01
TORQUE BY LEAST SQUARES	-0.54117630E 01	-0.13963041E 01	0.86817721E 01
RMS ERROR OF PARAB FIT	0.49526629E-01	0.53729685E-01	0.16289875E 00

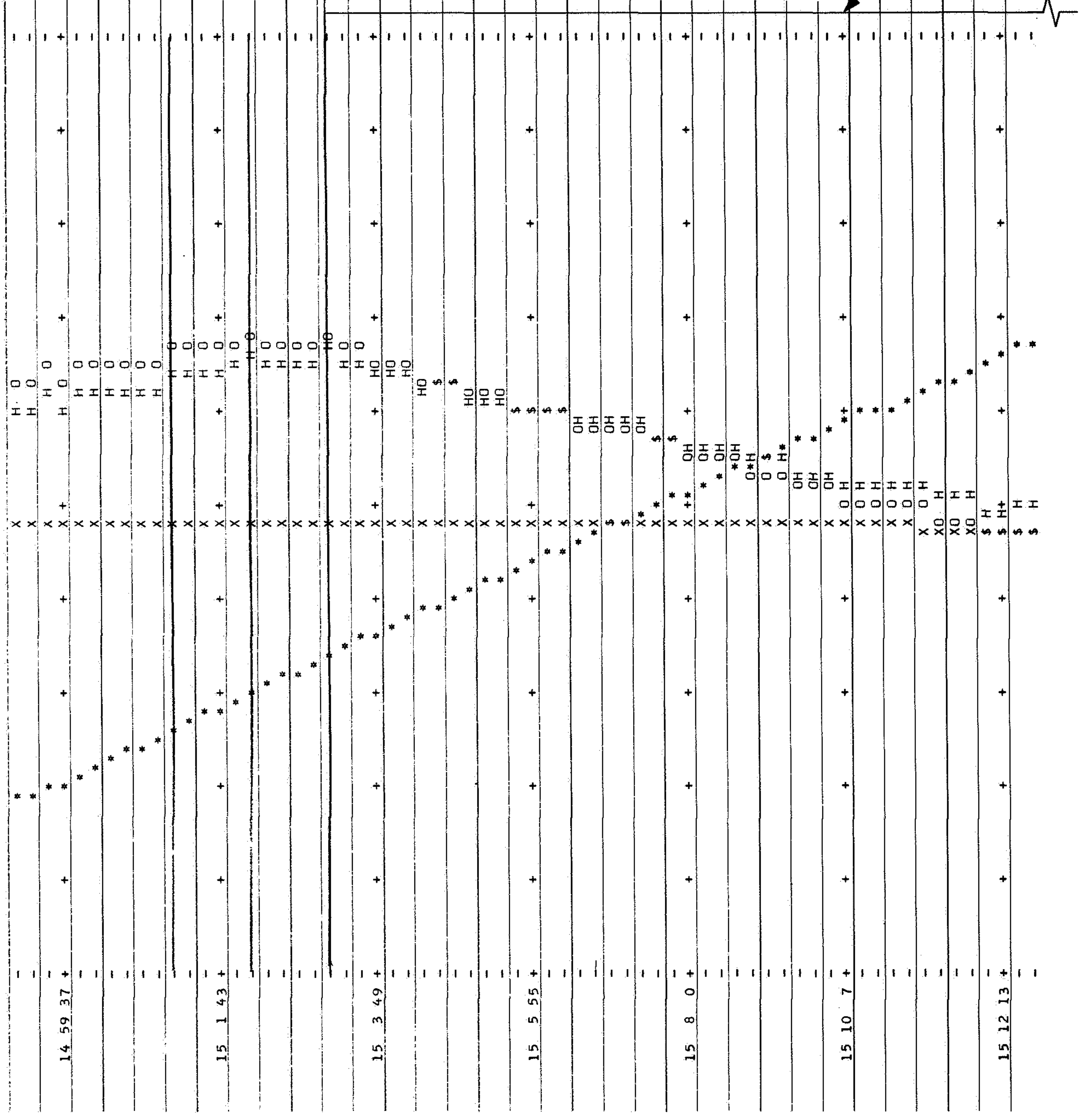
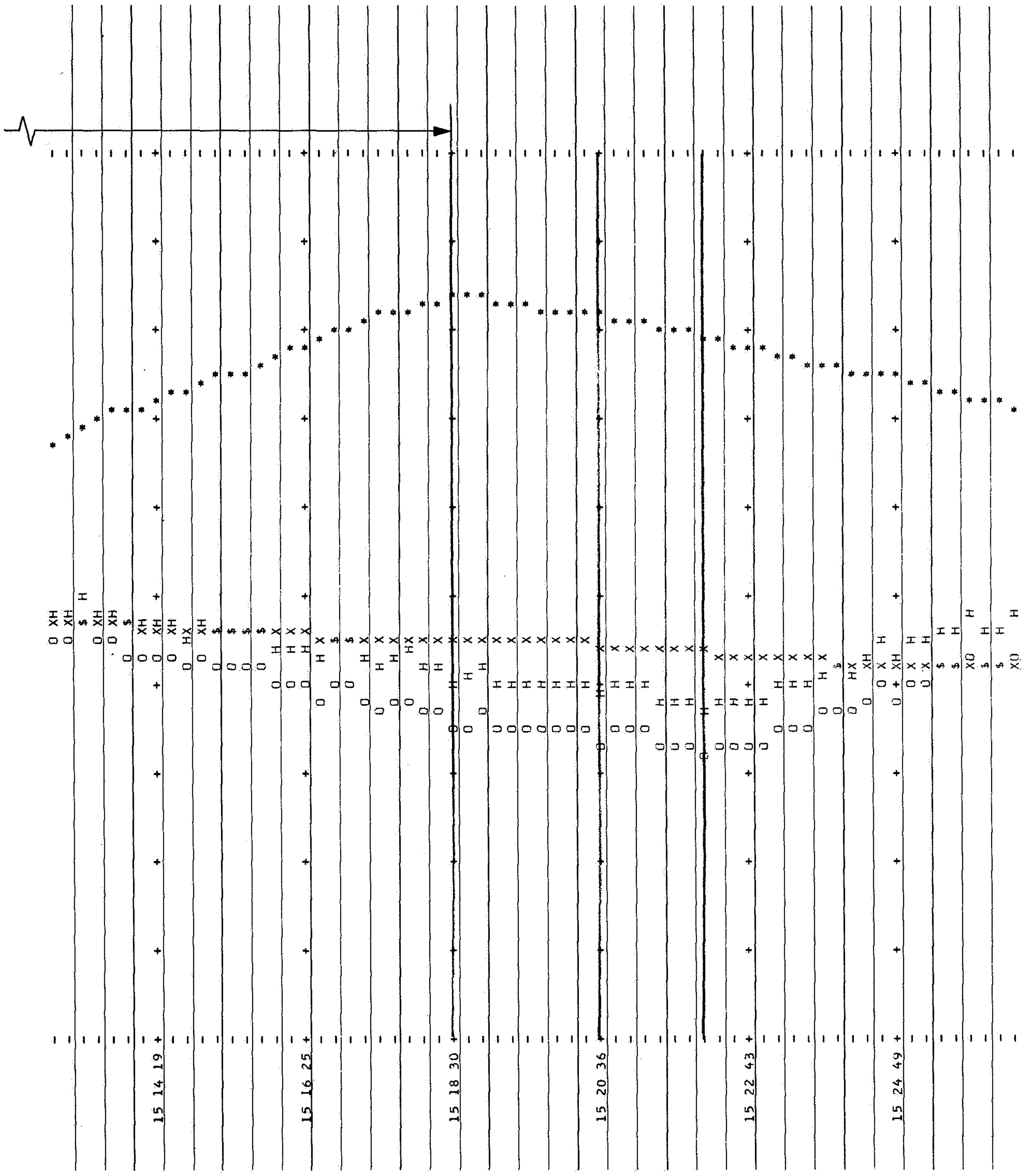


Fig. 11 (contd)


15 14 19
15 16 25
15 18 30
15 20 36
15 22 43
15 24 49

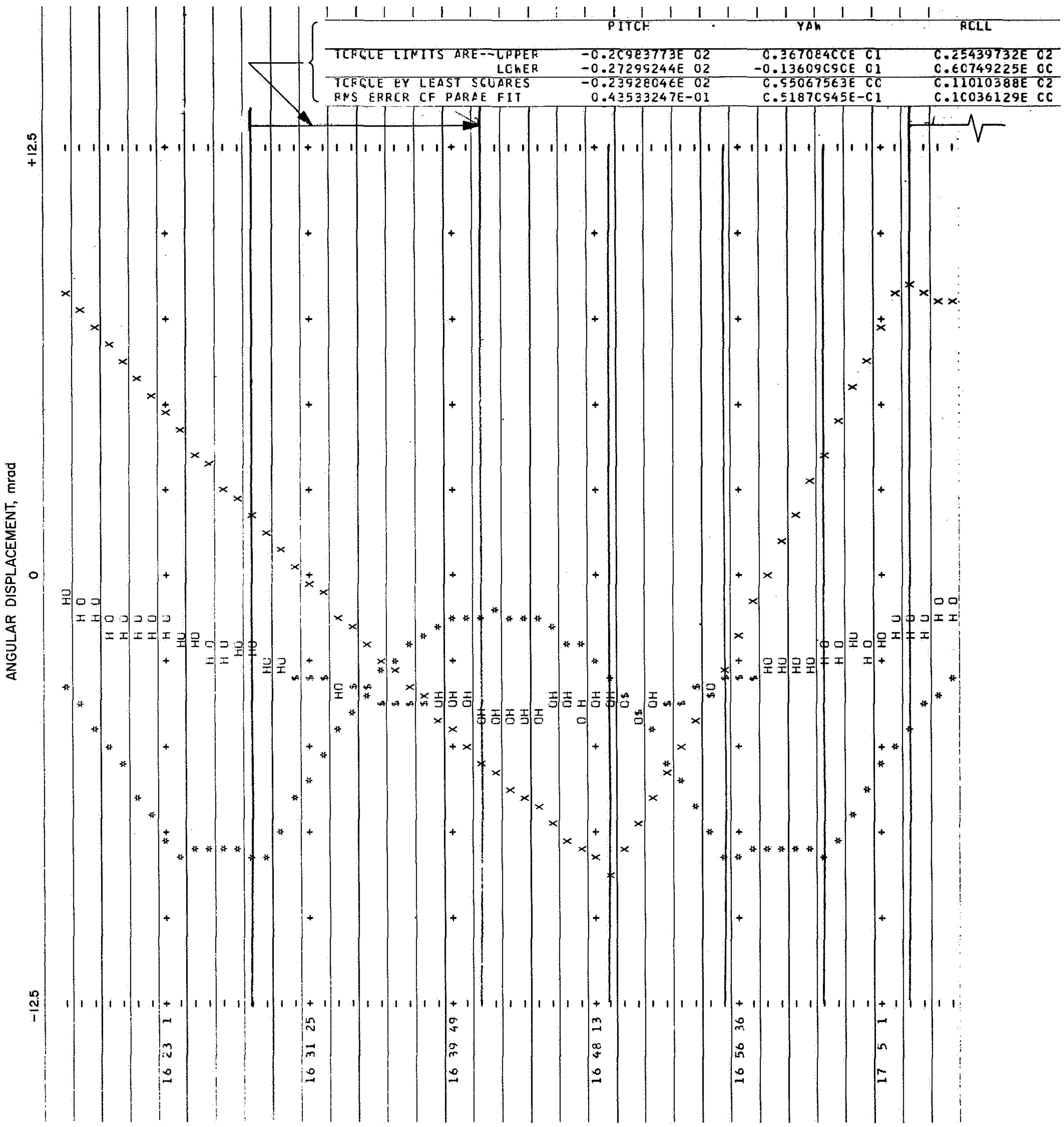
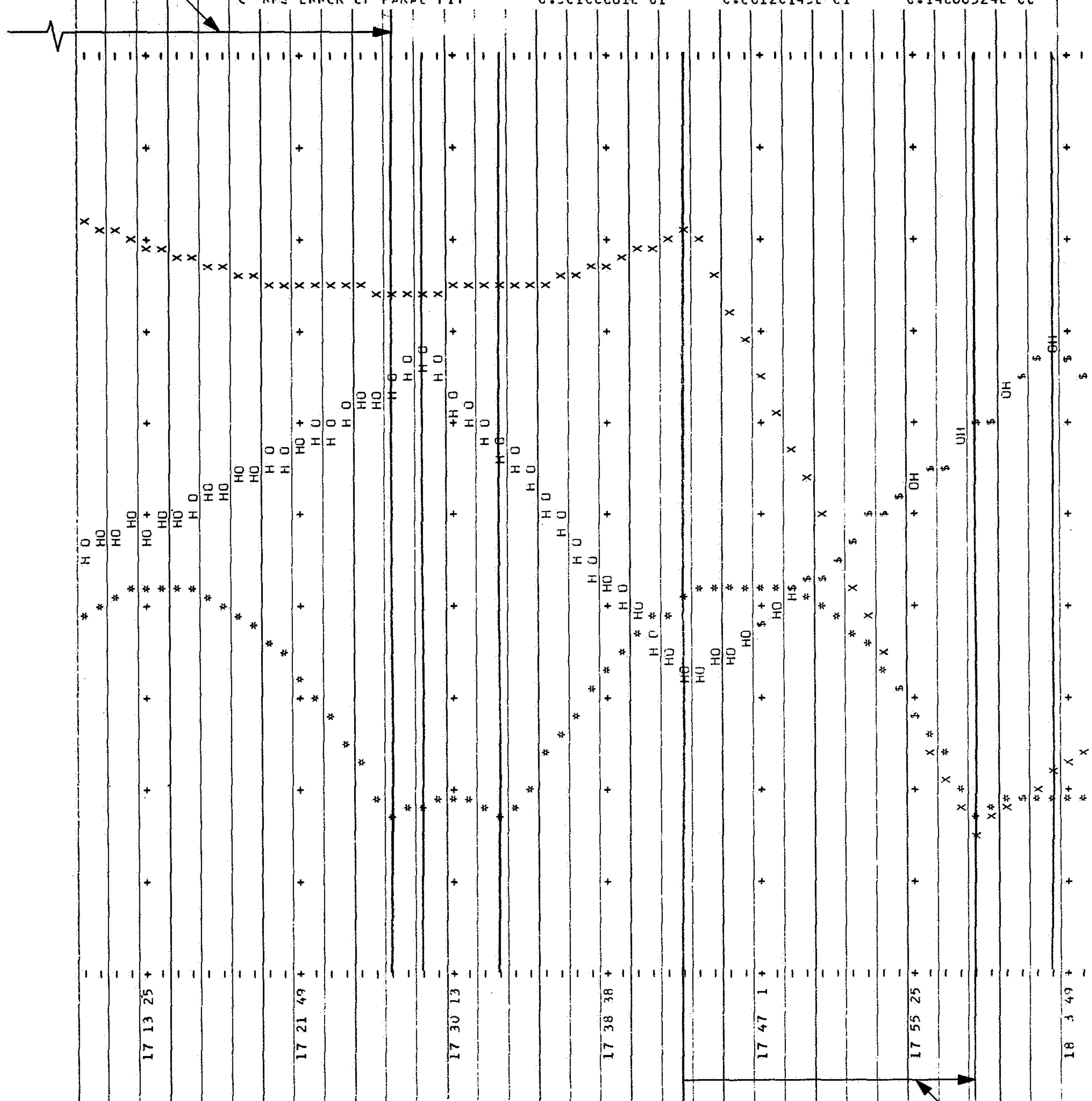


Fig. 12. Limit cycles at the low bit rate of 8⅓ bits/s, *Mariner IV*, day 105, 1965. Time is in hours, minutes, and seconds (GMT). Symbols: * = pitch; X = yaw; O = V-axis roll; H = Z-axis or true roll; $ = superimposed data points

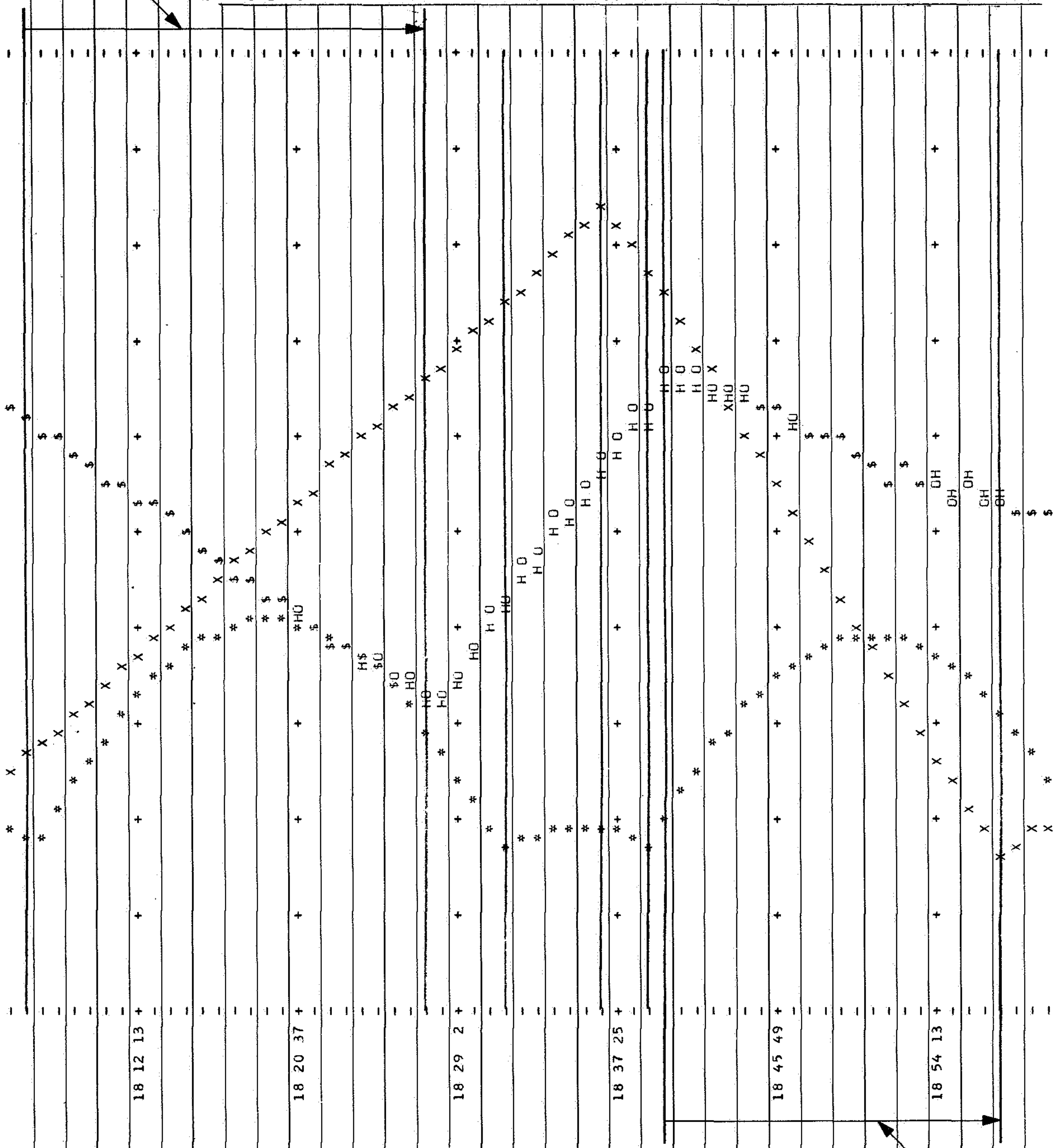
	PITCH	YAW	RCLL
TCRQUE LIMITS ARE--UPPER	-0.23990973E 02	0.27455252E C1	C.11141083E C2
LCWER	-0.26233066E C2	C.7584546CE CC	C.48752540E C1
TCRQUE BY LEAST SQUARES	-0.25165261E 02	C.1814C993E C1	C.56145326E C1
RMS ERRCR CF PARAE FIT	C.5C1C8661E-01	C.6C12C145E-C1	C.14866524E CC



	PITCH	YAW	RCLL
TCRQUE LIMITS ARE--UPPER	-0.23237058E 02	C.43651755E C1	C.34515358E C1
LCWER	-0.27816508E 02	C.19494458E C1	-C.16298233E C1
TCRQUE BY LEAST SQUARES	-0.25175231E 02	C.283252C2E C1	C.2C981733E 01
RMS ERRCR CF PARAE FIT	0.48345726E-01	C.78441752E-C1	C.15447221E CC

Fig. 12 (contd)

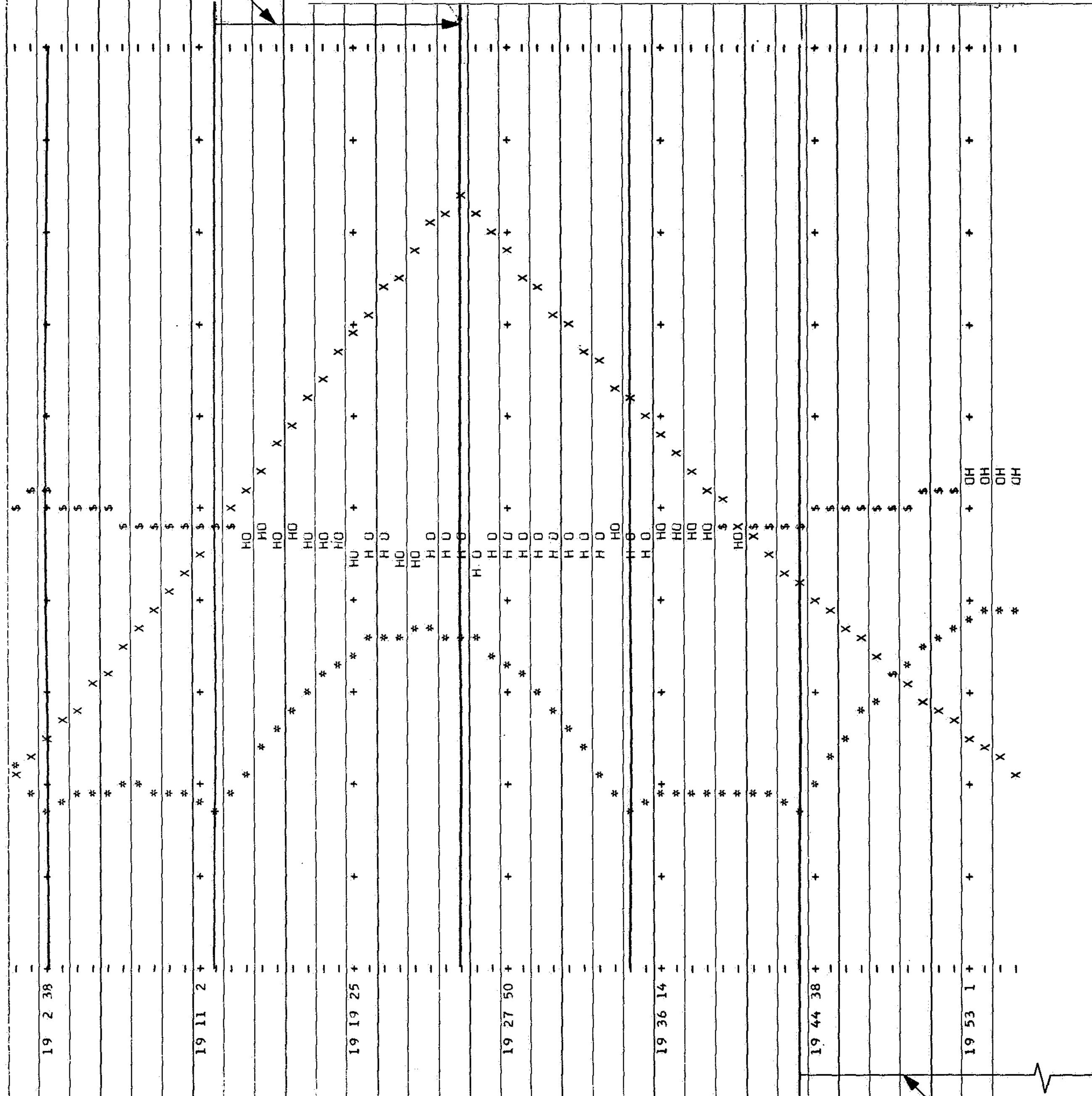
	PITCH	YAW	RCLL
TCRQLE LIMITS ARE--UPPER	-0.24292406E 02	C.55995215E CC	C.96664487E C1
LCWER	-0.26074711E 02	-C.72175769E CC	C.15124317E C1
TCRQUE BY LEAST SQUARES	-0.25165279E 02	C.7120C498E-C2	C.17090128E 01
RMS ERRCR CF PARAE FIT	0.5C468938E-01	C.53745978E-C1	0.15590722E CC



	PITCH	YAW	RCLL
TCRQUE LIMITS ARE--UPPER	-0.23222260E 02	C.25273796E C1	C.78949143E 01
LOWER	-0.26502334E 02	-C.50887105E CC	C.17766599E C1
TCRQLE BY LEAST SQUARES	-0.24864442E C2	C.55240127E CC	C.25565872E C1
RMS ERRCR CF PARAE FIT	0.45899128E-01	C.59190374E-C1	C.19934629E CC

Fig. 12 (contd)

	PITCH	YAW	ROLL
TORQUE LIMITS ARE--UPPER	-0.23023096E 02	0.12829929E 01	0.19843036E 02
LOWER	-0.27632921E 02	-0.62062448E 01	0.28180337E 01
TORQUE BY LEAST SQUARES	-0.25675080E 02	-0.23451067E 01	0.55039123E 01
RMS ERROR OF PARAB FIT	0.53093935E-01	0.60307803E-01	0.19680787E 00



	PITCH	YAW	ROLL
TORQUE LIMITS ARE--UPPER	-0.22053348E 02	0.47493753E 01	0.10578750E 02
LOWER	-0.26773264E 02	0.97549298E 00	-0.58567119E 01
TORQUE BY LEAST SQUARES	-0.24240897E 02	0.30698596E 01	0.22217621E 01
RMS ERROR OF PARAB FIT	0.50972448E-01	0.48327527E-01	0.14573887E 00

Fig. 12 (contd)

	PITCH	YAW	ROLL
TORQUE LIMITS ARE--UPPER	-0.21582146E 02	-0.99118437E 00	0.10155702E 02
LOWER	-0.25793787E 02	-0.43440720E 01	-0.16864140E 01
TORQUE BY LEAST SQUARES	-0.23740541E 02	-0.28032863E 01	0.34036550E 01
RMS ERROR OF PARAB FIT	0.45011786E-01	0.43321976E-01	0.17067071E 00

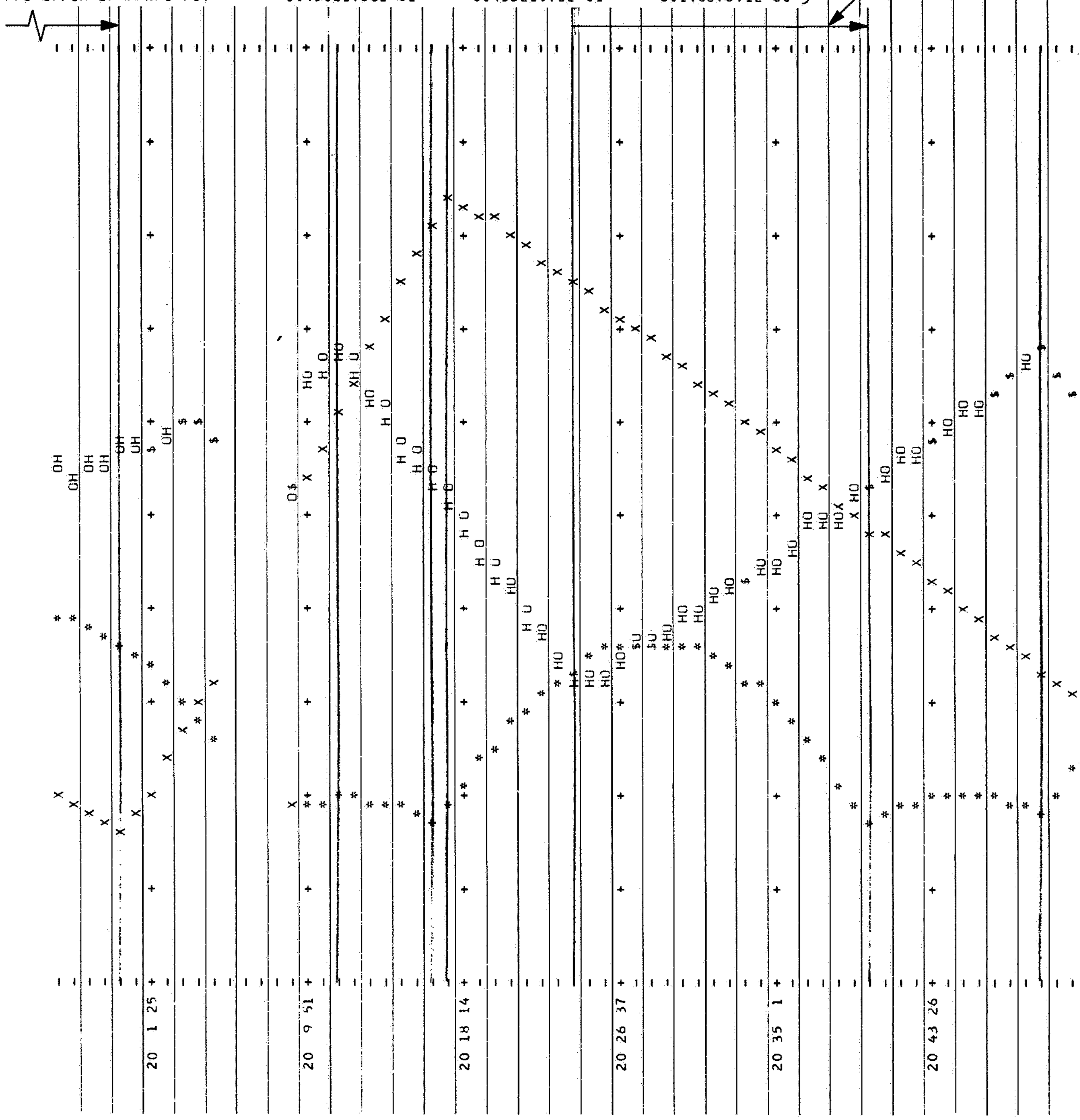


Fig. 12 (contd)

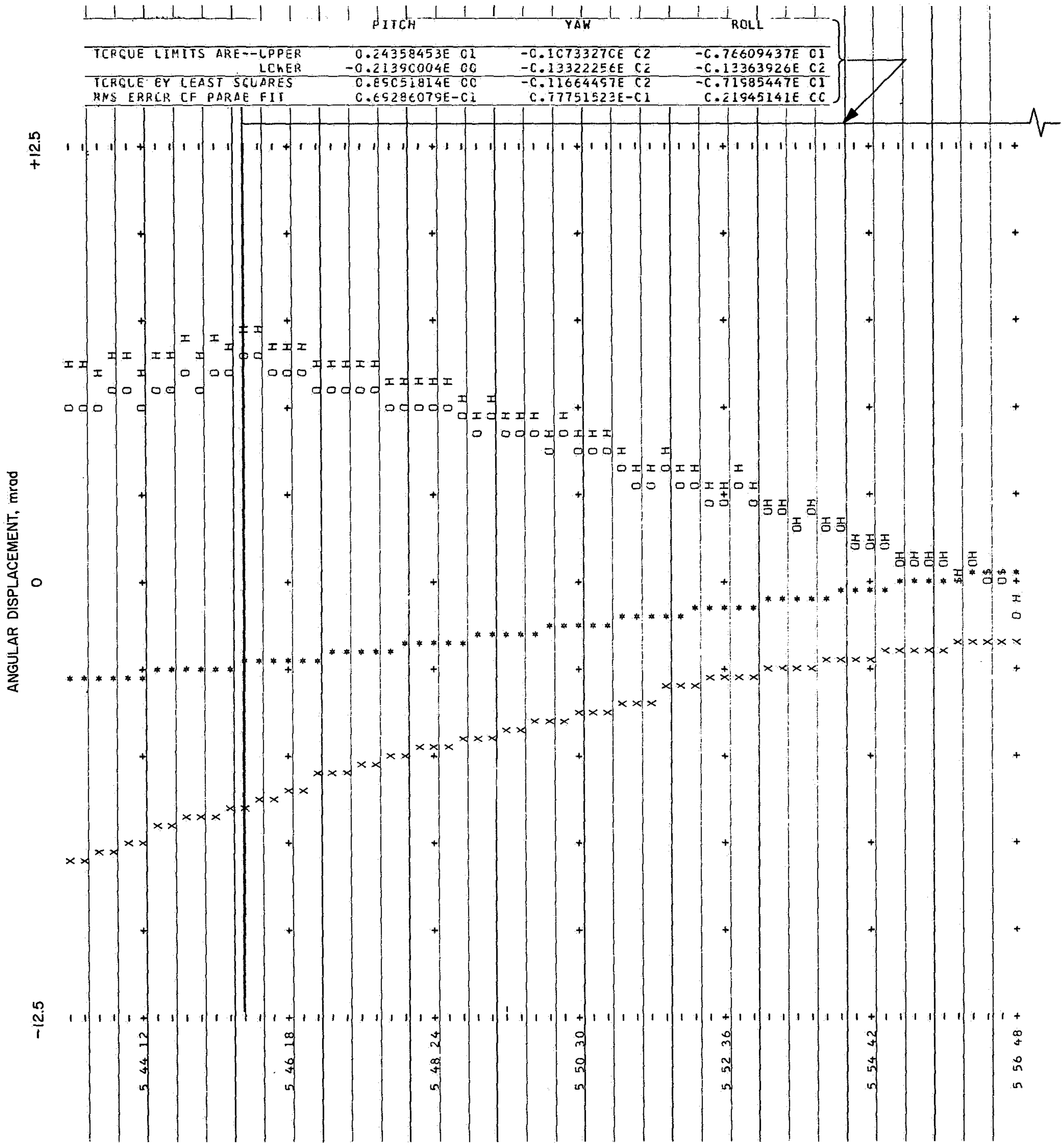


Fig. 13. Torque characteristics, *Mariner V*, day 183, 1967. Time is in hours, minutes, and seconds (GMT). Symbols: * = pitch; X = yaw; O = V-axis roll; H = Z-axis or true roll; $ = superimposed data points

	PITCH	YAW	ROLL
TORQUE LIMITS ARE--UPPER	0.1238694CE C1	-C.1C822C11E C2	C.43166106E CC
LOWER	-0.11450643E 01	-C.13561491E C2	-C.16794262E C2
TORQUE BY LEAST SQUARES	0.28973831E OC	-C.11924713E C2	-C.66551048E C1
RMS ERROR OF PARAB FIT	0.74547760E-C1	C.7681483CE-C1	C.2C766808E CC

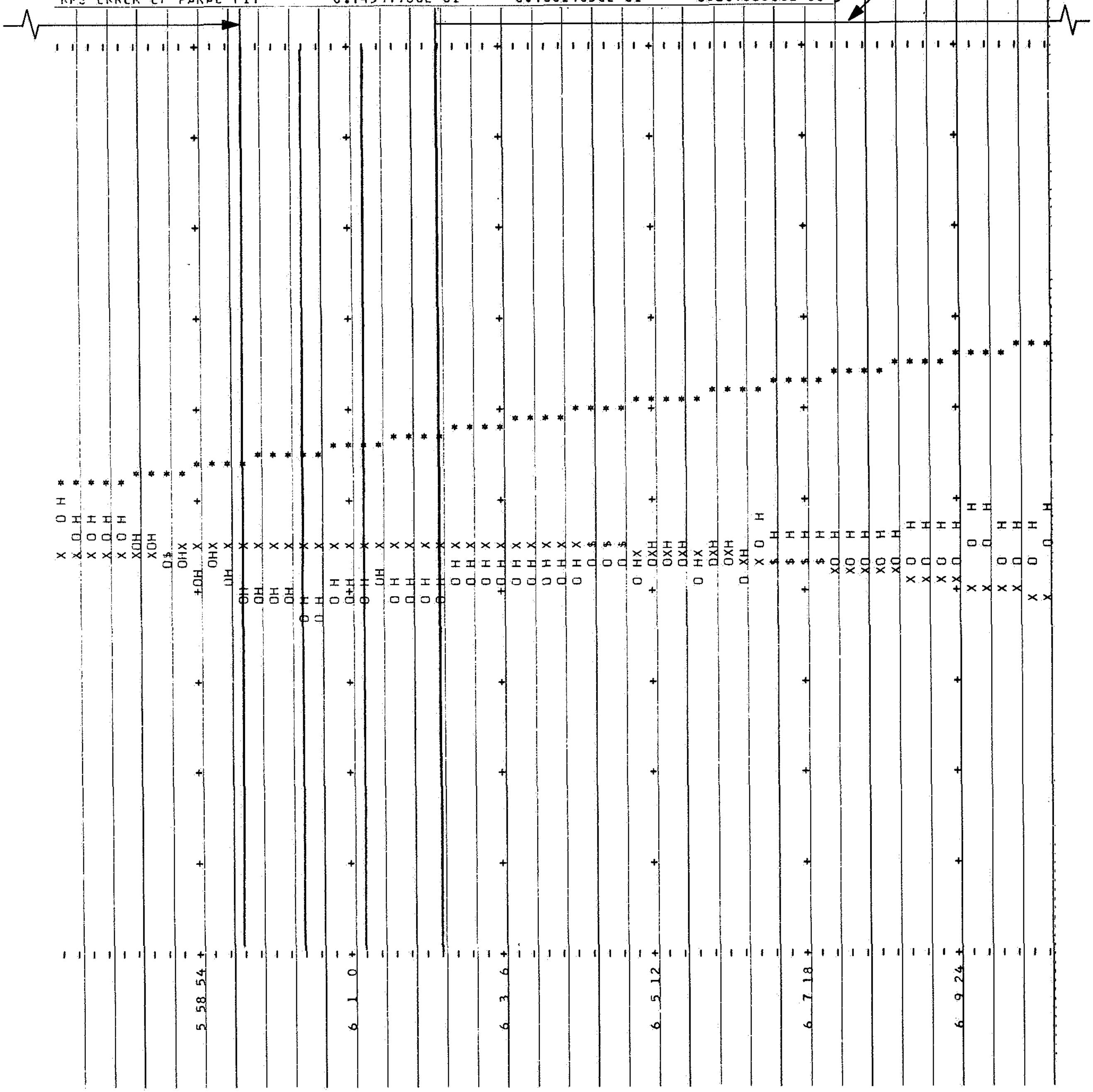


Fig. 13 (contd)

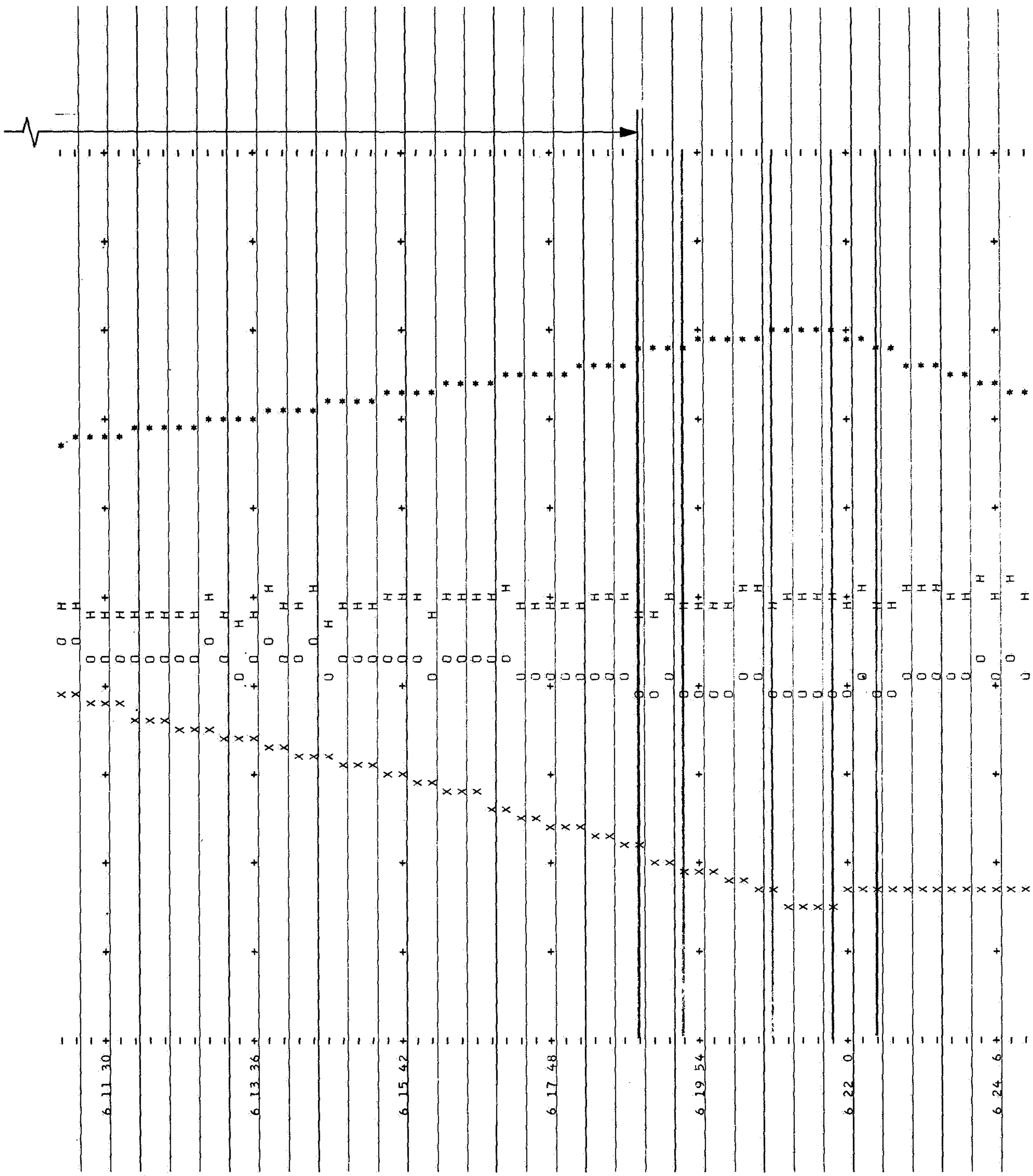


Fig. 13 (contd)

	PITCH	YAW	ROLL
TORQUE LIMITS ARE--UPPER	0.34313705E 01	-0.11360811E 02	-0.76499617E 01
LOWER	0.27851415E 01	-0.11863807E 02	-0.77937392E 01
TORQUE BY LEAST SQUARES	0.31332366E 01	-0.11651734E 02	-0.73048396E 01
RMS ERROR OF PARAB FIT	0.74443196E-01	0.76301528E-01	0.24941518E 00

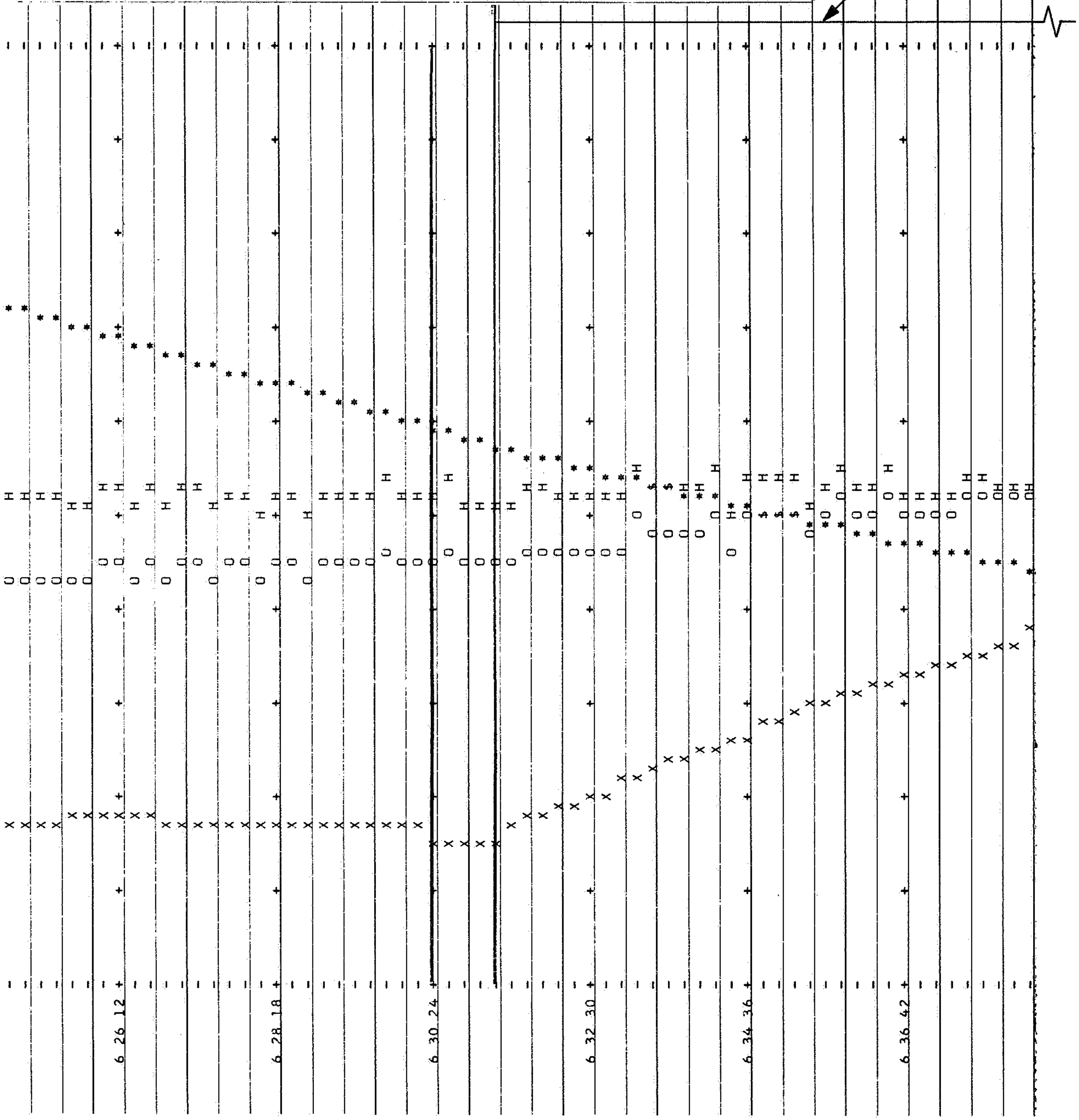


Fig. 13 (contd)

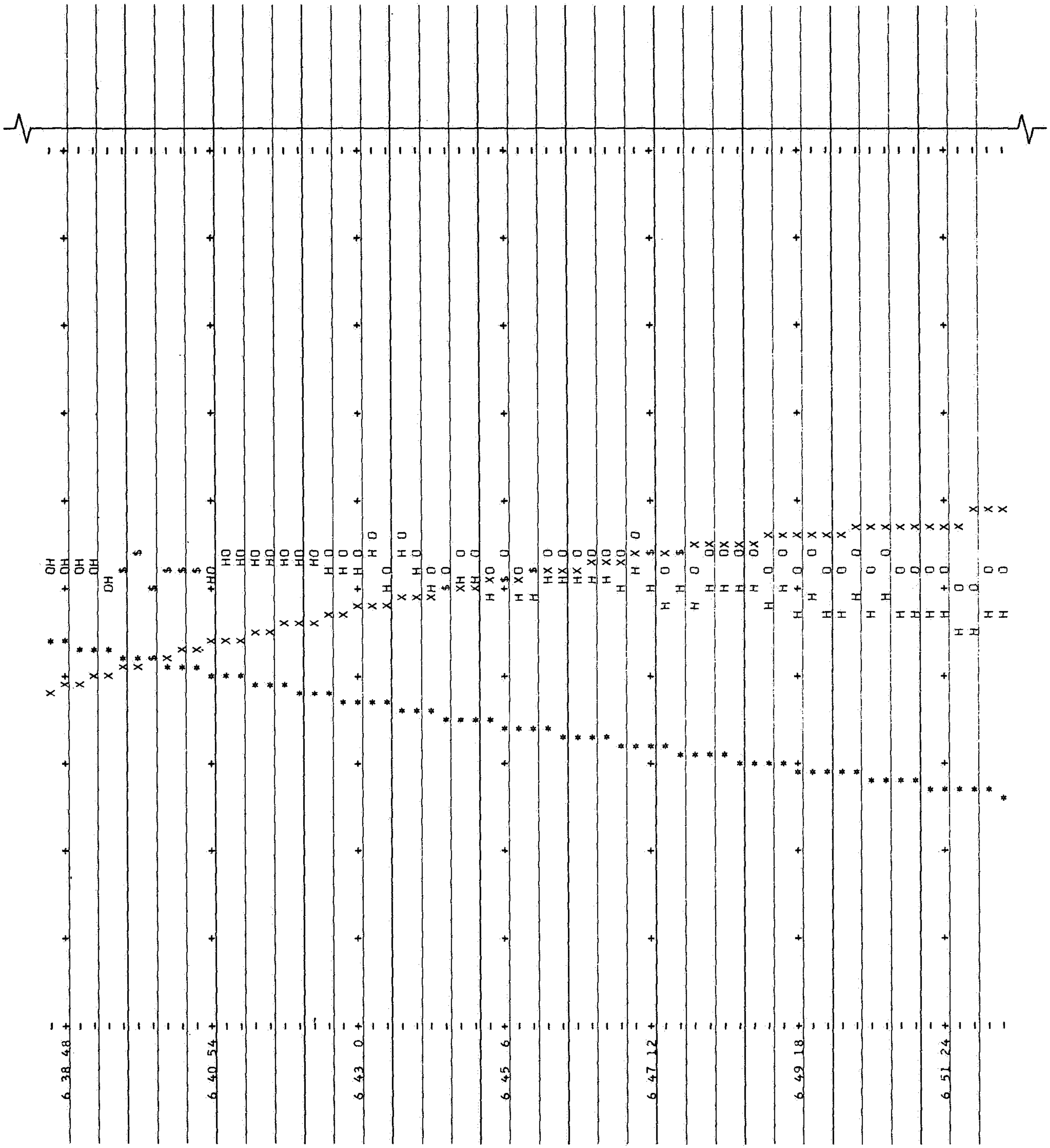


Fig. 13 (contd)

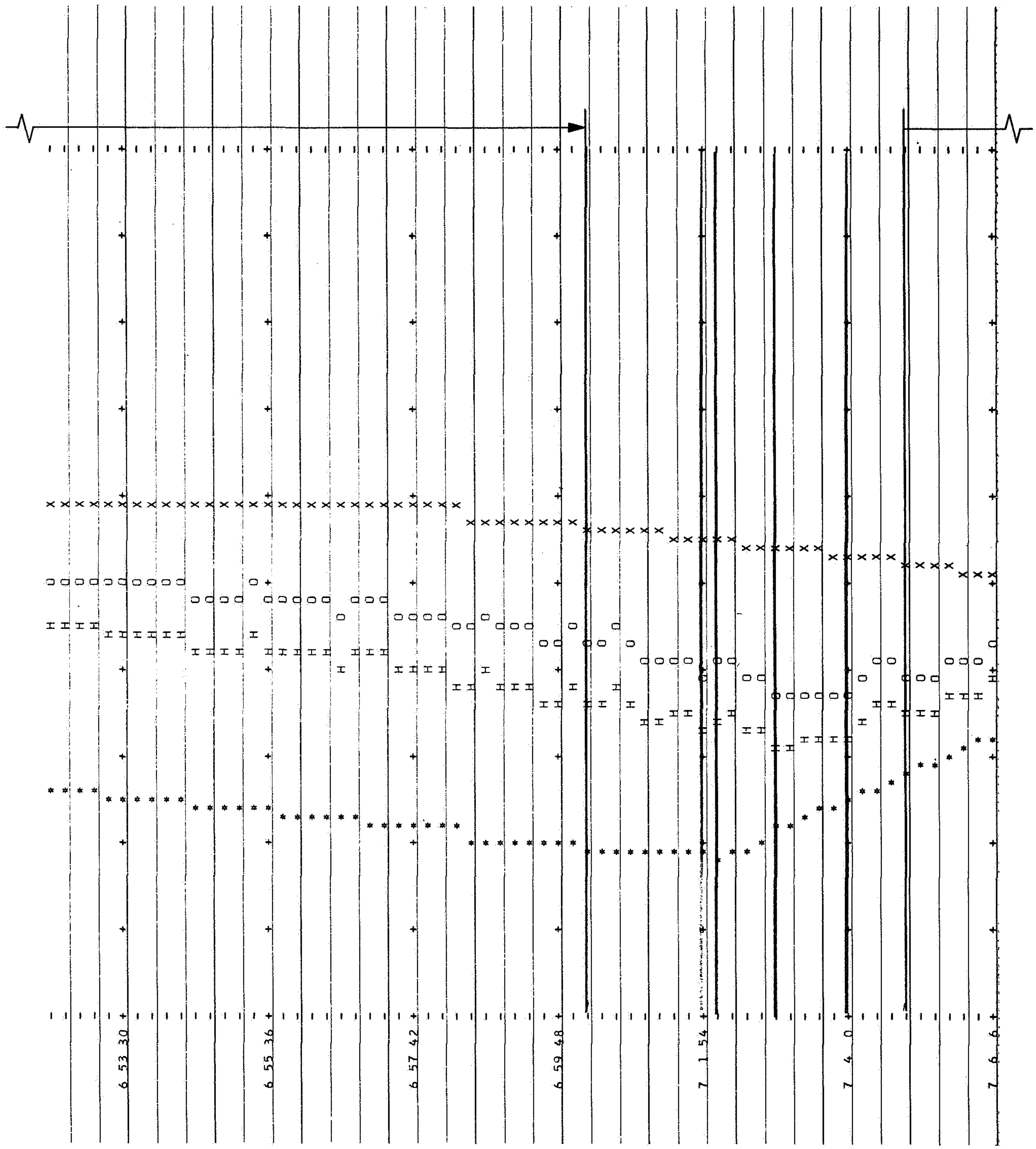


Fig. 13 (contd)

	PITCH	YAh	RCLL
TCRQUE LIMITS ARE--UPPER	-0.19657025E 0C	-C.9C347778E C1	-C.92408660E CC
LCWER	-0.28596989E 01	-C.13076499E C2	-C.18478362E C2
TCRQUE BY LEAST SQUARES	-0.13761098E 01	-C.11586342E C2	-C.53514749E C1
RMS ERRCR CF PARAE FIT	0.78375692E-01	C.753C2217E-C1	C.21581901E CC

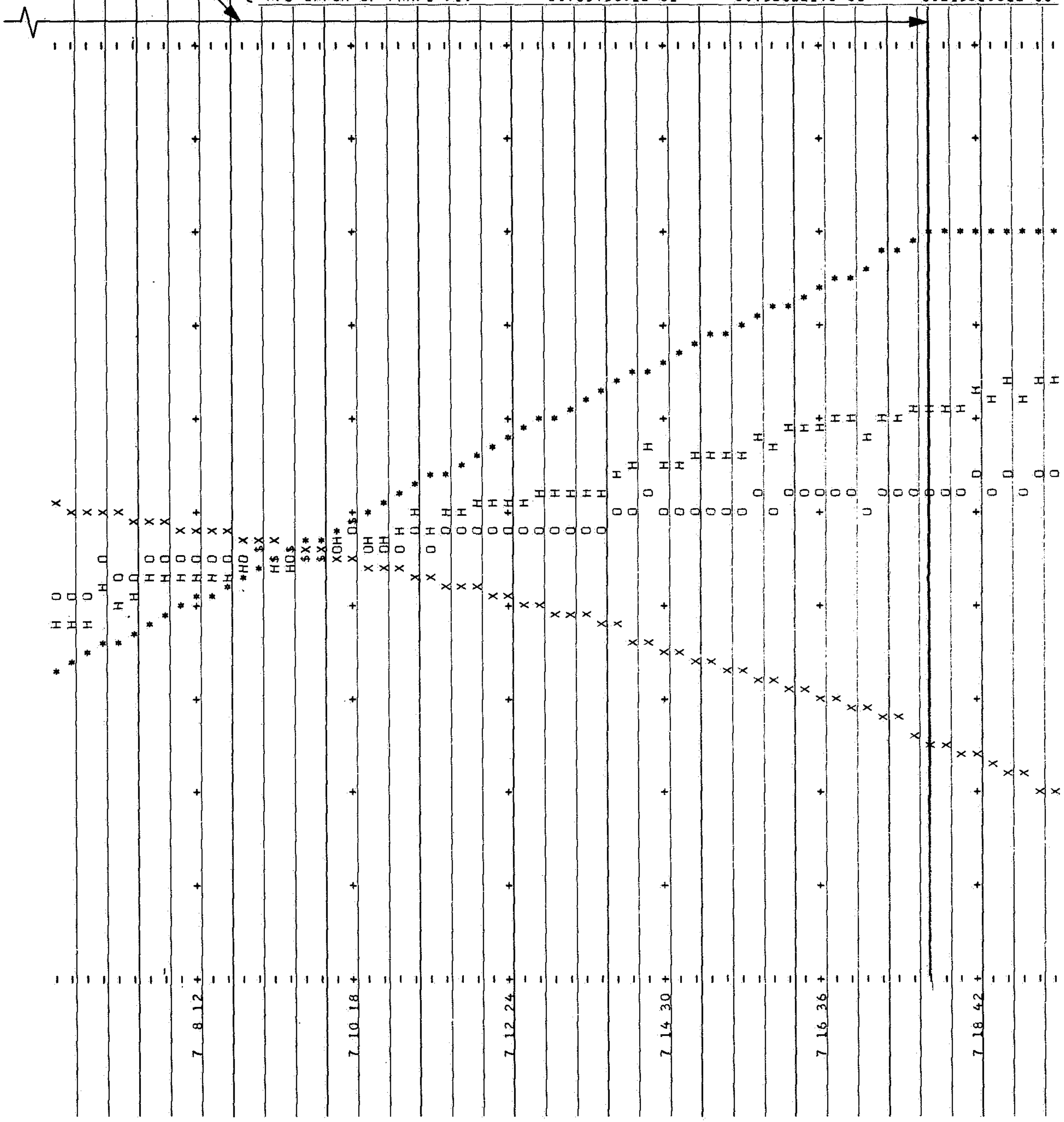


Fig. 13 (contd)

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

$$\bar{\theta} = \bar{a}t^2 + \bar{b}t + \bar{c} \tag{A-1}$$

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

$$\bar{\theta} = \bar{x}t^2 + \bar{y}t + \bar{z} \tag{A-2}$$

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:

$$\left.\begin{aligned} \bar{x}t_i^2 + \bar{y}t_i + \bar{z} &= \bar{\theta}_i \\ \bar{x}t_j^2 + \bar{y}t_j + \bar{z} &= \bar{\theta}_j \\ \bar{x}t_k^2 + \bar{y}t_k + \bar{z} &= \bar{\theta}_k \end{aligned}\right\} \tag{A-3}$$

where

$$i, j, k = 1, N$$

$$i \neq j \neq k$$

That is,

$$\begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} = \begin{bmatrix} t_i^2 & t_i & 1 \\ t_j^2 & t_j & 1 \\ t_k^2 & t_k & 1 \end{bmatrix}^{-1} \begin{bmatrix} \bar{\theta}_i \\ \bar{\theta}_j \\ \bar{\theta}_k \end{bmatrix} \tag{A-4}$$

This equation is solved for every combination of N sample points taken three at a time. The intersection of all solutions yields $\bar{a}, \bar{b}, \bar{c}$. 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:

$$\text{Number of equations} = \binom{N}{3} = \frac{N!}{(N-3)!\,3!} \tag{A-5}$$

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 8⅓ (33⅓) 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 8⅓ 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 ANGLE—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 3×3 matrix by a 3×1 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 3×3 matrix.

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

(13) $IBFTC TRANSF—Multiplies a 3×3 matrix by a 3×1 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

```
$IBFTC ACL
C
C        MARINER ATTITUDE CONTROL LISTER
C
001      DIMENSION IA(500),K(5)
         DIMENSION TH(3)
002      DATA IEH,MH,IEOF,IEND/6H100003,6H100004,6HINTEOF,6HENDTAP/
003      LOGICAL OK
004      DATA (K(I),I=1,5) / 6H200005,6H200006,6H200007,6H200008,6H400000/
006      INTEGER GYRO,VDN(3)
007      REAL V(4)
008      NDEX=0
009      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(NDEX.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. IEH .OR. IA(2) .EQ. MH) GO TO 28
031      IF(IA(2) .NE. IEOF) GO TO 35
034      GO TO 28
035      IF(IA(2) .EQ. IEND) GO TO 28
036      DO 39 I=1,5
037      JQ=I
038      IF(IA(2) .EQ. K(I)) GO TO 42
039      CONTINUE
040      CALL CR
041      STOP
042      IF(JQ .EQ. 4) GO TO 28
04201    IF(NDEX.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,I30,V30,OK,I22,V22,GYRO,ID,IHR,IM,IS)
062      DO 117 IJ=1,10
C
C        FIND AND STORE DATA FROM DESIRED CHANNELS
C
063      CALL GETTM(IA,VDN,I30,V30,OK,I22,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)
      CALL ANGPOS(TH,V,ID)
068   IF(.NOT.OK)GO TO 80
078   IF(I30.EQ.0)GO TO 80
079   CAN=V30
080   IF(NDEX.NE.0)GO TO 102
C
C     DATA OUTPUT
C
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 JPLT3(12.5,-12.5,0.,0.,4,IQ,V,IHR,IM,IS)
105   IQ=1
117   CONTINUE
      KC=KC+1
      IF(KC.NE.30)GO TO 118
      NDEX=2
118   CALL CLOCK(T)
119   T=T-TL
120   IF(60000.*TIME - T) 121,121,25
121   CALL CR
122   CALL STOP9
123   WRITE(6,138)ID,IHR,IM,IS,V(1),V(2),V(3),CAN
      END FILE 17
125   STOP
126   FORMAT(1H1, 35X,46HATTITUDE CONTROL MARINER MDL LISTER    --   PAGE,
     $ I4)
127   FORMAT(4I5)
128   FORMAT(/45X,3HDAY,I4,5H HOUR,I3,4H MIN,I3,4H SEC,I3)
129   FORMAT(/10X,68HDAY    HR     MIN    SEC    PITCH     YAW     ROLL     TRU
     $ ROLL      CAN CONE)
130   FORMAT(10X,I3,2X,I3,4X,I3,3X,I3,2X,3F7.2,4X,F7.2,4X,F7.2)
133   FORMAT(1H1/)
134   FORMAT(80I5)
137   FORMAT(1H1/15X,51HPLOT OF MARINER PITCH YAW AND ROLL CHANNELS IN M
     $RAD,5X,12HSTARTING DAY,I4///
     $25X,9H* = PITCH //25X,7HX = YAW //25X,8HO = ROLL //
     $25X,18HH = CORRECTED ROLL //)
138   FORMAT(10X,20HLAST DATA ON TAPE IS,4I7,5X,4F7.2)
      END
$IBMAP SPLT
*
*                 CALL SPLIT(WORD,DN1,DN2)
*
        ENTRY     SPLIT
SPLIT   SXA       END,4
        CLA       4,4
        STA       SXA1
        CLA       5,4
        STA       SXA2
        CAL*      3,4
        LRS       18
        ANA       MASK
        SLW*      4,4
        LLS       18
        ANA       MASK
        SLW*      5,4
        CAL*      3,4
```

```
         LXA      PZE,4
         COM
         PBT
SXA1     SXA      **,4
         LRS      17
         LBT
SXA2     SXA      **,4
END      AXT      **,4
         TRA      1,4
PZE      PZE      333
MASK     OCT      000000000177
         END
$IBMAP   CLOK2
*
*        DSL/90 SIMULATOR CLOCK
*
         ENTRY    CLOCK
         ENTRY    CLOK3
CLOCK    ZET      FLAG
         TRA      ZERO
         CAL      5
         ORA      EXP
         FAD      ZIP
         XCA
         FMP      SCALE
         STO*     3,4
         TRA      1,4
ZERO     STZ*     3,4
         STZ      5
         STZ      FLAG
         TRA      1,4
CLOK3    ZAC
         NZT      FLAG
         TRA      OK
         STO      FLAG
         STO      5
OK       LDQ      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
$IBMAP   TIMR
*
*                 TIME CONVERTER
*
         ENTRY    TIMER
TIMER    ZAC                                   CALL TIMER(ITIME,IDAY,IMIN,ISEC)
         LDQ*     3,4
         DVH      DAY
         XCA
         ADD      =1
         STO*     4,4
```

```
         ZAC
         DVH     HOUR
         STO*    5,4
         LRS     35
         DVH     MIN
         STO*    6,4
         LRS     35
         DVH     SEC
         STO*    7,4
         TRA     1,4
DAY      DEC     2764800
HOUR     DEC     115200
MIN      DEC     1920
SEC      DEC     32
         END
$IBMAP RDR-
*                MDL TAPE READER   (SETUP ON CK1)
*
INFILE FILE      MDL INPUT,CK1,INPUT,BLOCK=330,MXBCD
*
         ENTRY   READER                                 CALL READER(AREA,NOWDS)
         ENTRY   OR                                     OPN REWD
         ENTRY   ONR                                    OPN NO REWD
         ENTRY   CR                                     CLSE REWD
         ENTRY   CNR                                    CLSE NO REWD
*
OR       SXA     EOR,4
         TSX     .OPEN,4
         PZE     INFILE
EOR      AXT     **,4
         TRA     1,4
ONR      SXA     EONR,4
         TSX     .OPEN,4
         MZE     INFILE
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
         MZE     INFILE
ECNR     AXT     **,4
         TRA     1,4
READER SXA       END,4                                  CALL READER(AREA,NOWDS)
         CLA     3,4
         STA     READ+3
READ     TSX     .READ,4
         PZE     INFILE,,EOB
         PZE     EOF,,ERR
         IORT    **,,**
         LXD     *-1,4
         PXA     0,4
END      AXT     **,4
         STO*    4,4
         TRA     1,4
EOB      HTR     0
EOF      LXA     END,4
         CLA     =-1
```

```
        STO*      4,4
        TRA       1,4
 ERR    TSX       .MWR,4
        PZE       1
        PZE       ERMSG,,6
        LXA       END,4
        CLA       =-2
        STO*      4,4
        TRA       1,4
 ERMSG  BCI       6,MARINER MDL INPUT TAPE READ ERROR
        END
 $IBFTC DECOM-
 C
 C      THIS IS FOR MM-64
 C      DECOMMUTATE MDL DATA--CHANNELS  105,106, AND 114
 C
 1      SUBROUTINE DECOM(BUF,V,K30,V30,OK,K22,V22,GYRO,DAY,HR,MIN,SEC)
 2      LOGICAL OK
 3      INTEGER DN1,DN2,GYRO,V(3)
 4      DIMENSION BUF(330),LP(40),LOPOS(3,40)
 C
 C      HIGH RATE
 C
 5      OK=.TRUE.
 6      CALL TIMER(BUF(3),DAY,HR,MIN,SEC)
 C
 C      FIND DK200 SYNC
 C
 9      KS=-7
 10     DO 15 I=1,10
 11     KS=KS+32
 12     N=I-1
 13     CALL SPLIT(BUF(KS),DN1,DN2)
 14     IF(DN1 .EQ. 127 .AND. DN2 .EQ. 127) GO TO 18
 15     CONTINUE
 16     OK=.FALSE.
 01601  ICNTR=0
 01602  KC=-7
 17     RETURN
 C
 C      SYNC MEDIUM,LOW, AND LOW-LOW
 C
 18     IF(KS .EQ. 313) GO TO 81
 1801   CALL SPLIT(BUF(KS+32),DN1,DN2)
 C
 C      LOW POSITION INDICATOR
 C
 19     DATA (LP(J),J=1,40)/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/
 20     DO 23 I=1,40
 21     ILP=I
 22     IF(DN2 .EQ. LP(I)) GO TO 28
 23     CONTINUE
 24     ILP=2
 25     GO TO 28
 27     DATA (LOPOS(I,1),I=1,120)/
       $                                411,301,431,412,302,432,413,303,433,414,
       $304,434,415,305,435,416,306,436,417,307,437,418,308,438,419,309,
       $439,410,300,430,401,301,421,402,302,422,403,303,423,404,304,424,
       $405,305,425,406,306,426,407,307,427,408,308,428,409,309,429,
       $400,300,420,411,301,431,412,302,432,413,303,433,414,304,434,
```

```
      $415,305,435,416,306,436,417,307,437,418,308,438,419,309,439,
      $410,300,430,401,301,421,402,302,422,403,303,423,404,304,424,
      $405,305,425,406,306,426,407,307,427,408,308,428,409,309,429,
      $400,300,420/
 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
 3209  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
 C     CHANNELS 105,106 AND 114
 C
 40    CALL SPLIT(BUF(KC+2),DN1,DN2)
 41    V(1)=DN2
 42    CALL SPLIT(BUF(KC+3),DN1,DN2)
 43    V(2)=DN1
 51    CALL SPLIT(BUF(KC+7),DN1,DN2)
 52    V(3)=DN1
 59    IF(I22.NE.221)GO TO 63
 5901  CALL SPLIT(BUF(KC+5),DN1,DN2)
 5902  K22=I22
 5903  V22=DN1
 63    IF(I20.NE.202)GO TO 79
 64    I30=LOPOS(2,ILP)
 65    IF(I30.NE.303)GO TO 79
 6601  CALL SPLIT(BUF(KC),DN1,DN2)
       XX=DN2
       II=XX/12.-0.5
       V30=II
 68    K30=I30
 79    CALL TIMER(BUF(KC-22),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
 $IBFTC DECOM-
 C
```

```
C       THIS IS FOR MV-67
C       DECOMMUTATE MDL DATA--CHANNELS  103,104, AND 114
C
1       SUBROUTINE DECOM(BUF,V,K30,V30,OK,K22,V22,GYRO,DAY,HR,MIN,SEC)
2       LOGICAL OK
3       INTEGER DN1,DN2,GYRO,V(3)
4       DIMENSION BUF(330),LP(40),LOPOS(3,40)
C
C       HIGH RATE
C
5       OK=.TRUE.
6       CALL TIMER(BUF(3),DAY,HR,MIN,SEC)
C
C       FIND DK200 SYNC
C
9       KS=-7
10      DO 15 I=1,10
11      KS=KS+32
12      N=I-1
13      CALL SPLIT(BUF(KS),DN1,DN2)
14      IF(DN1 .EQ. 127 .AND. DN2 .EQ. 127) GO TO 18
15      CONTINUE
16      OK=.FALSE.
01601 ICNTR=0
01602 KC=-7
17      RETURN
C
C       SYNC MEDIUM,LOW, AND LOW-LOW
C
18      IF(KS .EQ. 313) GO TO 81
1801    CALL SPLIT(BUF(KS+32),DN1,DN2)
C
C       LOW POSITION INDICATOR
C
19      DATA (LP(J),J=1,40)/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/
20      DO 23 I=1,40
21      ILP=I
22      IF(DN2 .EQ. LP(I)) GO TO 28
23      CONTINUF
24      ILP=2
25      GO TO 28
27      DATA (LOPOS(I,1),I=1,120)/
       $                              411,301,431,412,302,432,413,303,433,414,
       $304,434,415,305,435,416,306,436,417,307,437,418,308,438,419,309,
       $439,410,300,430,401,301,421,402,302,422,403,303,423,404,304,424,
       $405,305,425,406,306,426,407,307,427,408,308,428,409,309,429,
       $400,300,420,411,301,431,412,302,432,413,303,433,414,304,434,
       $415,305,435,416,306,436,417,307,437,418,308,438,419,309,439,
       $410,300,430,401,301,421,402,302,422,403,303,423,404,304,424,
       $405,305,425,406,306,426,407,307,427,408,308,428,409,309,429,
       $400,300,420/
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
3209   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
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
51     CALL SPLIT(BUF(KC+7),DN1,DN2)
52     V(3)=DN1
59     IF(I22.NE.221)GO TO 63
5901   CALL SPLIT(BUF(KC+5),DN1,DN2)
5902   K22=I22
5903   V22=DN1
63     IF(I20.NE.202)GO TO 79
64     I30=LOPOS(2,ILP)
65     IF(I30.NE.303)GO TO 79
6601   CALL SPLIT(BUF(KC),DN1,DN2)
       XX=DN2
       II=XX/12.-0.5
       V30=II
68     K30=I30
79     CALL TIMER(BUF(KC-22),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
$IBFTC CANPUS
C
C      CANOPUS 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
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
```

```
C
C                    K = TIME
C        OUTPUT      C
C                    C(1) = PITCH IN MILLI-RADIANS
C                    C(2) = YAW   IN MILLI-RADIANS
C                    C(3) = ROLL  IN MILLI-RADIANS
C                    C(4) = CORRECTED ROLL IN MILLI-RADIANS
C
      DIMENSION B(3),C(4)
      COMMON/COEFF/AA(4,3)
      COMMON/CANOP1/CLOCK/CANOP2/CONE(275)
C
C                    CLOCK   =X-AXIS CLOCK ANGLE IN DEGREES
C                    CONE(K)=CANOPUS CONE ANGLE IN DEGREES
C
      IF (K.GT.330) J=K-331
      IF (K.LT.330) J=K+34
      THETA1=CLOCK*.17453293E-01
      THETA2=CONE(J)*.17453293E-01
C
C     CALIBRATION OF ANGULAR POSITION FROM DATA NUMBER
C
      C(1)=(AA(1,1)+AA(2,1)*B(1)+AA(3,1)*B(1)**2+AA(4,1)*B(1)**3)
      C(2)=(AA(1,2)+AA(2,2)*B(2)+AA(3,2)*B(2)**2+AA(4,2)*B(2)**3)
      C(3)=(AA(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)*COS(THETA1)+C(2)*SIN(THETA1)))/
     $ SIN(THETA2)
      RETURN
      END
$IBFTC CANPUS
C     CANOPUS MEASUREMENT CORRECTION - FOR MV-67
C
      SUBROUTINE ANGPOS(B,C,K)
C
C     THIS IS FOR MV-67
C     YOU ALSO NEED SUBPROGRAM BLOCK DATA
C
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
C                    K = TIME
C        OUTPUT      C
C                    C(1) = PITCH IN MILLI-RADIANS
C                    C(2) = YAW   IN MILLI-RADIANS
C                    C(3) = ROLL  IN MILLI-RADIANS
C                    C(4) = CORRECTED ROLL IN MILLI-RADIANS
C
      DIMENSION B(3),C(4)
      COMMON/COEFF/A(6,3)
      COMMON/CANOP1/CLOCK/CANOP2/CONE(135)
C
C                    CLOCK   =X-AXIS CLOCK ANGLE IN DEGREES
C                    CONE(K)=CANOPUS CONE ANGLE IN DEGREES
C
      J=K-164
      THETA1=CLOCK*.17453293E-01
      THETA2=CONE(J)*.17453293E-01
C
```

```
C       CALIBRATION OF ANGULAR POSITION FROM DATA NUMBER
C
        C(1)=(A(1,1)+A(2,1)*B(1)+A(3,1)*B(1)**2+A(4,1)*B(1)**3
       1+A(5,1)*B(1)**4+A(6,1)*B(1)**5)
        C(2)=(A(1,2)+A(2,2)*B(2)+A(3,2)*B(2)**2+A(4,2)*B(2)**3
       1+A(5,2)*B(2)**4+A(6,2)*B(2)**5)
        C(3)=(A(1,3)+A(2,3)*B(3)+A(3,3)*B(3)**2+A(4,3)*B(3)**3
       1+A(5,3)*B(3)**4+A(6,3)*B(3)**5)
        C(4)=(C(3)-COS(THETA2)*(-C(1)*COS(THETA1)+C(2)*SIN(THETA1)))/
       $ SIN(THETA2)
        RETURN
        END
$IBFTC ANGLE
C                                                                       ANGLE4
C       THIS IS FOR MM-64                                               ANGLE4
C                                                                       ANGLE4
        BLOCK DATA                                                      ANGLE4
        COMMON/CANOP1/CLOCK                                             ANGLE4
        COMMON/CANOP2/CONE(275)                                         ANGLE4
        COMMON/COEFF/AA(4,3)                                            ANGLE4
        DATA CLOCK/-.56E02/                                             ANGLE4
        DATA (CONE(I),I=1,75)/                                          ANGLE4
       1    101.80,   101.95,   102.09,   102.23,   102.36,             ANGLE4
       2    102.49,   102.61,   102.72,   102.83,   102.93,             ANGLE4
       3    103.03,   103.13,   103.21,   103.30,   103.78,             ANGLE4
       4    103.45,   103.52,   103.58,   103.64,   103.70,             ANGLE4
       5    103.75,   103.79,   103.83,   103.87,   103.90,             ANGLE4
       6    103.93,   103.95,   103.97,   103.98,   103.99,             ANGLE4
       7    104.00,   104.00,   104.00,   104.00,   103.99,             ANGLE4
       8    103.97,   103.96,   103.94,   103.91,   103.89,             ANGLE4
       9    103.85,   103.82,   103.78,   103.74,   103.70,             ANGLE4
       1    103.65,   103.60,   103.55,   103.49,   103.43,             ANGLE4
       2    103.37,   103.30,   103.24,   103.16,   103.09,             ANGLE4
       3    103.01,   102.94,   102.86,   102.77,   102.69,             ANGLE4
       4    102.60,   102.51,   102.42,   102.33,   102.24,             ANGLE4
       5    102.15,   102.05,   101.95,   101.85,   101.74,             ANGLE4
       6    101.64,   101.53,   101.42,   101.31,   101.20/             ANGLE4
        DATA (CONE(I),I=76,150)/                                        ANGLE4
       7    101.09,   100.97,   100.85,   100.74,   100.62,             ANGLE4
       8    100.49,   100.37,   100.25,   100.12,    99.99,             ANGLE4
       9     99.86,    99.73,    99.60,    99.47,    99.34,             ANGLE4
       1     99.20,    99.06,    98.92,    98.78,    98.64,             ANGLE4
       2     98.50,    98.36,    98.21,    98.07,    97.92,             ANGLE4
       3     97.78,    97.63,    97.48,    97.33,    97.19,             ANGLE4
       4     97.04,    96.89,    96.74,    96.59,    96.44,             ANGLE4
       5     96.21,    96.14,    95.99,    95.84,    95.70,             ANGLE4
       6     95.54,    95.39,    95.25,    95.10,    94.95,             ANGLE4
       7     94.80,    94.66,    94.52,    94.38,    94.24,             ANGLE4
       8     94.10,    93.97,    93.83,    93.69,    93.56,             ANGLE4
       9     93.42,    93.28,    93.15,    93.02,    92.88,             ANGLE4
       1     92.75,    92.62,    92.48,    92.35,    92.22,             ANGLE4
       2     92.09,    91.96,    91.83,    91.70,    91.57,             ANGLE4
       3     91.44,    91.31,    91.18,    91.05,    90.93/             ANGLE4
        DATA (CONE(I),I=151,225)/                                       ANGLE4
       4     90.80,    90.68,    90.55,    90.43,    90.30,             ANGLE4
       5     90.18,    90.05,    89.93,    89.81,    89.69,             ANGLE4
       6     89.57,    89.44,    89.32,    89.20,    89.08,             ANGLE4
       7     88.96,    88.84,    88.72,    88.60,    88.49,             ANGLE4
       8     88.37,    88.25,    88.13,    88.02,    87.90,             ANGLE4
       9     87.78,    87.66,    87.55,    87.43,    87.32,             ANGLE4
       1     87.20,    87.08,    86.97,    86.85,    86.74,             ANGLE4
```

```
     2   86.62,    86.50,    86.39,    86.27,    86.16,          ANGLE4
     3   86.04,    85.93,    85.82,    85.70,    85.59,          ANGLE4
     4   85.48,    85.36,    85.30,    85.14,    85.02,          ANGLE4
     5   84.91,    84.80,    84.69,    84.58,    84.47,          ANGLE4
     6   84.35,    84.24,    84.13,    84.02,    83.91,          ANGLE4
     7   83.83,    83.69,    83.58,    83.47,    83.36,          ANGLE4
     8   83.25,    83.14,    83.04,    82.93,    82.82,          ANGLE4
     9   82.71,    82.60,    82.50,    82.39,    82.28/          ANGLE4
      DATA (CONE(I),I=226,275)/                                ANGLE4
     1   82.18,    82.07,    81.96,    81.86,    81.75,          ANGLE4
     2   81.64,    81.54,    81.43,    81.33,    81.22,          ANGLE4
     3   81.12,    81.02,    80.91,    80.81,    80.70,          ANGLE4
     4   80.60,    80.50,    80.39,    80.29,    80.19,          ANGLE4
     5   80.09,    79.98,    79.88,    79.78,    79.68,          ANGLE4
     6   79.58,    79.48,    79.38,    79.28,    79.18,          ANGLE4
     7   79.08,    78.98,    78.88,    78.78,    78.68,          ANGLE4
     8   78.58,    78.48,    78.38,    78.28,    78.18,          ANGLE4
     9   78.09,    77.99,    77.89,    77.79,    77.70,          ANGLE4
     1   77.60,    77.50,    77.40,    77.30,    77.20/          ANGLE4
      DATA(AA(I,1),I=1,12)/13.816884,-.27638703,.13834448E-02,-.70307050 ANGLE4
     $E-05,13.816884,-.27638703,.13834448E-02,-.70307050E-05,26.5,-.414, ANGLE4
     $0.,0./                                                   ANGLE4
      END                                                      ANGLE4
$IBFTC ANGLE
C                                                              ANGLE7
C     THIS IS FOR MV-67                                        ANGLE7.
C                                                              ANGLE7
      BLOCK DATA                                               ANGLE7
      COMMON/CANOP1/CLOCK/CANOP2/CONE(135)                     ANGLE7
      COMMON/COEFF/A(6,3)                                      ANGLE7
      DATA CLOCK/ .45E02/                                      ANGLE7
      DATA(CONE(I),I=5,75 )/ .7671E02,      .7670E02,      .7668E02,  ANGLE7
     8  .7666E02,      .7665E02,      .7664E02,      .7663E02,    ANGLE7
     8  .7663E02,      .7663E02,      .7663E02,      .7664E02,    ANGLE7
     8  .7665E02,      .7666E02,      .7668E02,      .7670E02,    ANGLE7
     8  .7672E02,      .7675E02,      .7678E02,      .7681E02,    ANGLE7
     8  .7685E02,      .7689E02,      .7694E02,      .7699E02,    ANGLE7
     8  .7704E02,      .7710E02,      .7716E02,      .7722E02,    ANGLE7
     8  .7729E02,      .7736E02,      .7744E02,      .7752E02,    ANGLE7
     8  .7760E02,      .7769E02,      .7778E02,      .7787E02,    ANGLE7
     8  .7797E02,      .7808E02,      .7819E02,      .7830E02,    ANGLE7
     8  .7842E02,      .7854E02,      .7866E02,      .7879E02,    ANGLE7
     8  .7892E02,      .7906E02,      .7920E02,      .7935E02,    ANGLE7
     8  .7950E02,      .7966E02,      .7982E02,      .7998E02,    ANGLE7
     8  .8015E02,      .8033E02,      .8051E02,      .8069E02,    ANGLE7
     8  .8088E02,      .8107E02,      .8127E02,      .8147E02,    ANGLE7
     8  .8168E02,      .8189E02,      .8210E02,      .8232E02,    ANGLE7
     8  .8255E02,      .8278E02,      .8302E02,      .8326E02,    ANGLE7
     8  .8350E02,      .8375E02,      .8400E02,      .8426E02/    ANGLE7
      DATA(CONE(I),I=76,135)/.8452E02,      .8479E02,      .8506E02,  ANGLE7
     8  .8534E02,      .8562E02,      .8590E02,      .8619E02,    ANGLE7
     8  .8649E02,      .8678E02,      .8708E02,      .8739E02,    ANGLE7
     8  .8770E02,      .8801E02,      .8833E02,      .8865E02,    ANGLE7
     8  .8897E02,      .8929E02,      .8962E02,      .8995E02,    ANGLE7
     8  .9029E02,      .9062E02,      .9096E02,      .9130E02,    ANGLE7
     8  .9165E02,      .9199E02,      .9233E02,      .9268E02,    ANGLE7
     8  .9303E02,      .9337E02,      .9372E02,      .9406E02,    ANGLE7
     8  .9441E02,      .9475E02,      .9510E02,      .9544E02,    ANGLE7
     8  .9578E02,      .9612E02,      .9645E02,      .9678E02,    ANGLE7
     8  .9711E02,      .9743E02,      .9775E02,      .9807E02,    ANGLE7
     8  .9837E02,      .9868E02,      .9898E02,      .9927E02,    ANGLE7
```

```
     8      .9955E02,          .9983E02,          .10010E03,         .10037E03,            ANGLE7
     8      .10064E03,         .10090E03,         .10116E03,         .10141E03,            ANGLE7
     8      .10165E03,         .10188E03,         .10210E03,         .10231E03,            ANGLE7
     8      .10252E03/                                                                      ANGLE7
      DATA(A(I,1),I=1,18)/25.326053,-.92093034,.19287685E-01,-.28223686E  ANGLE7
     $-03,0.21335972E-05,-.66782966E-08,28.046241,-1.0178037,.20517513E-  ANGLE7
     $01,-.28572125E-03,.20782001E-05,-.63998052E-08,21.414102,-.3237947  ANGLE7
     $1,.11165038E-01,-.34020915E-03,0.33585269E-05,-.10896906E-07/        ANGLE7
      END                                                                   ANGLE7
```

III. Data Reduction Program

```
$IBFTC MARDAT
C
C MARINER DATA REDUCTION
C
001     DIMENSION XJ(3,3),R(3),C(3,2),T1(3),TORQ(3),A(3,2),B(3,2),TIN(3,2)   MRT001
        DIMENSION T(300),P(3),Q(3),WD(3),TAN(3,2),W(3),WP(3)                 MRT002
        DIMENSION THIN(3),THOUT(3),TP(300),TY(300),TR(300)                   MRT003
        REAL INVAL(3),MJJ(4)                                                 MRT004
        COMMON AR(7,306),CKLIM(6),DEL                                        MRT005
        COMMON/LVT/LEVOUT                                                    MRT006
003     LOGICAL START,OK,RERED,BEGIN                                         MRT007
004     INTEGER TH(3),REALT(4)                                               MRT008
005     READ(5,112)DEL,NB,NA,(CKLIM(I),I=1,6)                                MRT009
006     READ(5,113)(XJ(I,1),I=1,9)                                           MRT010
00601   DELT=DEL/60.                                                         MRT011
        MZ=1                                                                 MRT012
        KDX=1                                                                MRT013
        KDY=NB-6                                                             MRT014
        IF(DEL.GT.25.)KDY=NB-4                                               MRT015
        LO=4                                                                 MRT016
        IF(DEL.GT.25.)LO=2                                                   MRT017
        LEVOUT=4                                                             MRT018
        IF(DEL .GT. 25.)LEVOUT=2                                             MRT019
        I1OUT=1                                                              MRT020
00602   IQ=0                                                                 MRT021
00603   CALL START9                                                          MRT022
00605   INX=0                                                                MRT023
007     RERED=.FALSE.                                                        MRT024
008     OK=.TRUE.                                                            MRT025
00801   BEGIN=.TRUE.                                                         MRT026
009     START=.TRUE.                                                         MRT027
010     JSART=4                                                              MRT028
011     II=4                                                                 MRT029
012     JJ=303                                                               MRT030
013     KL=0                                                                 MRT031
014     DO 43 I=II,JJ                                                        MRT032
015     IF(START)KL=KL+1                                                     MRT033
016     READ (17) ID,IHR,IM,IS,(TH(J),J=1,3),CAN                             MRT034
017     AR(1,I)=TH(1)                                                        MRT035
018     AR(2,I)=TH(2)                                                        MRT036
019     AR(3,I)=TH(3)                                                        MRT037
020     AR(4,I)=ID                                                           MRT038
021     AR(5,I)=IHR                                                          MRT039
022     AR(6,I)=IM                                                           MRT040
023     AR(7,I)=IS                                                           MRT041
024     IF(I.EQ.6)GO TO 27                                                   MRT042
025     IF(I.EQ.303)GO TO 31                                                 MRT043
026     GO TO 34                                                             MRT044
027     DO 291 L=1,7                                                         MRT045
028     DO 29 LI=1,3                                                         MRT046
029     AR(L,LI+303)=AR(L,LI+3)                                              MRT047
0291    CONTINUE                                                             MRT048
030     GO TO 34                                                             MRT049
031     DO 331 L=1,7                                                         MRT050
032     DO 33 LI=1,3                                                         MRT051
033     AR(L,LI)=AR(L,LI+300)                                                MRT052
0331    CONTINUE                                                             MRT053
034     IF(.NOT.START)GO TO 43                                               MRT054
035     IF(KL.NE.4)GO TO 43                                                  MRT055
036     START=.FALSE.                                                        MRT056
037     DO 42 L=1,3                                                          MRT057
```

```
 038    DO 39 LI=1,2                                                   MRT058
 03801  MI=II-LI                                                       MRT059
 03802  NI=II+LI                                                       MRT060
 039    AR(L,MI)=2.*AR(L,II)-AR(L,NI)                                  MRT061
 042    CONTINUE                                                       MRT062
 043    CONTINUE                                                       MRT063
 044    IF(.NOT.RERED)GO TO 49                                         MRT064
 045    II=4                                                           MRT065
 046    JJ=JJI                                                         MRT066
 047    RERED=.FALSE.                                                  MRT067
 048    GO TO 14                                                       MRT068
C
C       DATA STORED IN AR==NOW LOOK FOR LIMIT CYCLES
C
 049    INXPRE=INX                                                     MRT069
        IF(INXPRE.EQ.1)JSART=JSART+1                                   MRT070
        IF(INXPRE.EQ.3)JSART=JSART+2                                   MRT071
        CALL DETECT(JSART,JEND,N,INX,NAXIS)                            MRT072
 050    IF(N-NB)51,53,53                                               MRT073
 051    WRITE(6,100)                                                   MRT074
        IF(MZ.EQ.1.OR.LFVOUT.EQ.2) GO TO 5101                          MRT075
        IF(IAXIS.NE.0.AND.NAXIS.EQ.IAXIS) GO TO 5102                   MRT076
 05101  IAXIS=0                                                        MRT077
        GO TO 5103                                                     MRT078
 05102  MZ=1                                                           MRT079
        NJJ=NJJ+N-1                                                    MRT080
 05103  MDX=KDX                                                        MRT081
        KDX=KDX+N-1                                                    MRT082
        IF(KDX.LT.NB)GO TO 80                                          MRT083
        KDX=KDX-NB-1                                                   MRT084
        TORQ(1)=60.                                                    MRT085
        TORQ(2)=60.                                                    MRT086
        TORQ(3)=60.                                                    MRT087
        MDX=NB+1-MDX                                                   MRT088
        DO 5109 LI=1,N                                                 MRT089
        IF(LI.NE.MDX) GO TO 5109                                       MRT090
        MJ=JSART+LI-1                                                  MRT091
        IF(MJ.GT.303)MJ=MJ-300                                         MRT092
        JD=AR(4,MJ)                                                    MRT093
        JH=AR(5,MJ)                                                    MRT094
        JM=AR(6,MJ)                                                    MRT095
        JS=AR(7,MJ)                                                    MRT096
        IF(IQ.EQ.0)WRITE(9,107)JD                                      MRT097
        CALL JPLT3(25.,-25.,0.,0.,3,IQ,TORQ,JH,JM,JS)                  MRT098
        IQ=1                                                           MRT099
 05109  CONTINUE                                                       MRT100
 052    GO TO 80                                                       MRT101
 053    CALL PRIN(N,JSART,DELT,A,B,C,INY,NA)                           MRT102
        IF(INY.EQ.0)GO TO 54                                           MRT103
        WRITE(6,114)INY                                                MRT104
 054    NM=N-1                                                         MRT105
        NO=N-4                                                         MRT106
        IF(DEL.GT.25.)NO=N-2                                           MRT107
        DO 55 LI=LO,NM                                                 MRT108
 05401  MJ=JSART+LI-1                                                  MRT109
 05402  IF(MJ.GT.303)MJ=MJ-300                                         MRT110
 05403  LJ=LI-LO+1                                                     MRT111
        JD=AR(4,MJ)                                                    MRT112
        DO 5404 L=1,3                                                  MRT113
 05404  THIN(L)=AR(L,MJ)                                               MRT114
        CALL ANGPOS(THIN,THOUT,JD)                                     MRT115
```

```
         TP(LJ)=THOUT(1)                                                                        MRT116
         TY(LJ)=THOUT(2)                                                                        MRT117
         TR(LJ)=THOUT(3)                                                                        MRT118
 055     CONTINUE                                                                               MRT119
 056     CALL PARFIT(TP,NO,P,DELT,ER1)                                                          MRT120
 057     CALL PARFIT(TY,NO,Q,DELT,ER2)                                                          MRT121
 058     CALL PARFIT(TR,NO,R,DELT,ER3)                                                          MRT122
         CALL MAVT(XJ,A,TAN)                                                                    MRT123
 059     DO 7904 LI=LO,NM                                                                       MRT124
 05901   IF(KDX.NE.NB)GO TO 7903                                                                MRT125
 05902   KDX=1                                                                                  MRT126
         MJ=JSART+LI-1                                                                          MRT127
         IF(MJ.GT.303)MJ=MJ-300                                                                 MRT128
 065     WD(3)=2.*R(2)                                                                          MRT129
 068     WD(2)=2.*Q(2)                                                                          MRT130
 071     WD(1)=2.*P(2)                                                                          MRT131
C
C        COMPUTE TORQUE
C
 075     CALL MATVEC(XJ,WD,T1)                                                                  MRT132
 076     DO 77 L=1,3                                                                            MRT133
 07601   TIN(L,1)=13.56*TAN(L,1)/1.8                                                            MRT134
 07602   TIN(L,2)=13.56*TAN(L,2)/1.8                                                            MRT135
 077     TORQ(L)=13.56*T1(L)/3.6                                                                MRT136
 07701   TORMAG=SQRT(TORQ(1)**2+TORQ(2)**2+TORQ(3)**2)                                          MRT137
         IF(ER1.GT.0.3)TORQ(1)=60.0                                                             MRT138
         IF(ER2.GT.0.3)TORQ(2)=60.0                                                             MRT139
         IF(ER3.GT.0.3)TORQ(3)=60.0                                                             MRT140
         JD=AR(4,MJ)                                                                            MRT141
         JH=AR(5,MJ)                                                                            MRT142
         JM=AR(6,MJ)                                                                            MRT143
         JS=AR(7,MJ)                                                                            MRT144
 07703   IF(IQ.EQ.0)WRITE(9,107)JD                                                              MRT145
 07704   CALL JPLT3(25.,-25.,0.,0.,3,IQ,TORQ,JH,JM,JS)                                          MRT146
 07705   IQ=1                                                                                   MRT147
         IF(I1OUT.EQ.0) GO TO 7903                                                              MRT148
         IF(IAXIS.EQ.0)GO TO 7902                                                               MRT149
C
C        FIND MINIMUM RATE INCREMENT
C
         IF(IAXIS.NE.1)GO TO 7801                                                               MRT150
         W(1)=P(2)                                                                              MRT151
         W(2)=-2.*P(2)*P(3)                                                                     MRT152
         W(3)=P(1)+P(2)*P(3)**2                                                                 MRT153
         GO TO 7803                                                                             MRT154
 07801   IF(IAXIS.NE.2)GO TO 7802                                                               MRT155
         W(1)=Q(2)                                                                              MRT156
         W(2)=-2.*Q(2)*Q(3)                                                                     MRT157
         W(3)=Q(1)+Q(2)*Q(3)**2                                                                 MRT158
         GO TO 7803                                                                             MRT159
 07802   IF(IAXIS.NE.3)GO TO 7803                                                               MRT160
         W(1)=R(2)                                                                              MRT161
         W(2)=-2.*R(2)*R(3)                                                                     MRT162
         W(3)=R(1)+R(2)*R(3)**2                                                                 MRT163
 07803   CALL MINRAT(W,WP,NJJ,DELT,RATINC,REALT,DEDZON,INDCAT,MJJ)                              MRT164
         IF(INDCAT.EQ.0)GO TO 79                                                                MRT165
C
C        DATA OUTPUT
C
         WRITE(6,202)IAXIS                                                                      MRT166
         GO TO 7902                                                                             MRT167
```

```
079     WRITE(6,203)IAXIS,DEDZON,RATINC                                    MRT168
        WRITE(6,204) (REALT(I),I=1,4)                                      MRT169
07902   WRITE(6,99)                                                        MRT170
        WRITE(6,101)JD,JH,JM,JS                                            MRT171
        WRITE(6,201)LI,N                                                   MRT172
        WRITE(6,102)                                                       MRT173
        WRITE(6,103) (TIN(I,1),I=1,3)                                      MRT174
        WRITE(6,104) (TIN(I,2),I=1,3)                                      MRT175
        WRITE(6,105) (TORQ(I), I=1,3),TORMAG                               MRT176
        WRITE(6,106) ER1,ER2,ER3                                           MRT177
        DO 17902 I=1,3                                                     MRT178
17902   INVAL(I)=TIN(I,1)-TIN(I,2)                                         MRT179
        NV=N-LEVOUT                                                        MRT180
        WRITE(6,120)NV,(INVAL(I),I=1,3)                                    MRT181
        I1OUT=0                                                            MRT182
072     MQ=JSART+LO-1                                                      MRT183
07201   IF(MQ.GT.303)MQ=MQ-300                                             MRT184
07301   MJJ(1)=AR(4,MQ)                                                    MRT185
07202   MJJ(2)=AR(5,MQ)                                                    MRT186
07203   MJJ(3)=AR(6,MQ)                                                    MRT187
07204   MJJ(4)=AR(7,MQ)                                                    MRT188
07903   KDX=KDX+1                                                          MRT189
        IF(KDX.EQ.KDY)KDX=KDX+LEVOUT                                       MRT190
07904   CONTINUE                                                           MRT191
        MZ=0                                                               MRT192
        I1OUT=1                                                            MRT193
        IAXIS=NAXIS                                                        MRT194
C
C       DATA CONDITION OUTPUT PACKAGE
C
        IF(IAXIS.NE.1)GO TO 7906                                           MRT195
        WP(1)=P(2)                                                         MRT196
        WP(2)=-2.*P(2)*P(3)                                                MRT197
        WP(3)=P(1)+P(2)*P(3)**2                                            MRT198
        GO TO 7908                                                         MRT199
07906   IF(IAXIS.NE.2)GO TO 7907                                           MRT200
        WP(1)=Q(2)                                                         MRT201
        WP(2)=-2.*Q(2)*Q(3)                                                MRT202
        WP(3)=Q(1)+Q(2)*Q(3)**2                                            MRT203
        GO TO 7908                                                         MRT204
07907   IF(IAXIS.NE.3)GO TO 7908                                           MRT205
        WP(1)=R(2)                                                         MRT206
        WP(2)=-2.*R(2)*R(3)                                                MRT207
        WP(3)=R(1)+R(2)*R(3)**2                                            MRT208
07908   MJJ=N                                                              MRT209
080     IF(INXPRE.EQ.1)JSART=JSART-1                                       MRT210
        IF(INXPRE.EQ.3)JSART=JSART-2                                       MRT211
08001   IF(INX.EQ.1)WRITE(6,109)                                           MRT212
        IF(INX.EQ.1)WRITE(9,109)                                           MRT213
        IF(INX.EQ.2)WRITE(6,110)                                           MRT214
        IF(INX.EQ.2)WRITE(9,110)                                           MRT215
        IF(INX.EQ.3)WRITE(6,111)                                           MRT216
        IF(INX.EQ.3)WRITE(9,111)                                           MRT217
        II=JSART-2                                                         MRT218
081     IF(JEND.GT.303)GO TO 85                                            MRT219
082     JJ=JEND-3                                                          MRT220
083     JSART=JEND                                                         MRT221
084     GO TO 93                                                           MRT222
085     IF(JEND.GT.306)GO TO 89                                            MRT223
086     JJ=JEND-3                                                          MRT224
087     JSART=JEND-300                                                     MRT225
```

```
088     GO TO 93                                                                   MRT226
089     JJ=303                                                                     MRT227
090     RERED=.TRUE.                                                               MRT228
091     JJI=JEND-303                                                               MRT229
092     JSART=JEND-300                                                             MRT230
093     IF(II.GT.3)GO TO 14                                                        MRT231
09301   IF(.NOT.BEGIN)GO TO 94                                                     MRT232
09302   BEGIN=.FALSE.                                                              MRT233
09303   II=4                                                                       MRT234
        IF(JJ.LE.4)GO TO 49                                                        MRT235
09304   GO TO 14                                                                   MRT236
094     II=300+II                                                                  MRT237
09401   IF(JJ.GT.300)GO TO 14                                                      MRT238
09402   IF(JJ.LE.3)GO TO 9408                                                      MRT239
09403   JJ=303                                                                     MRT240
09404   IF(RERED)GO TO 9406                                                        MRT241
09405   JJI=JEND-3                                                                 MRT242
09406   RERED=.TRUE.                                                               MRT243
09407   GO TO 14                                                                   MRT244
09408   JJ=300+JJ                                                                  MRT245
095     IF(OK)GO TO 14                                                             MRT246
096     CALL STOP9                                                                 MRT247
09601   STOP                                                                       MRT248
099     FORMAT(1H0//)                                                              MRT249
0100    FORMAT(10X,18H NOT ENOUGH POINTS)                                          MRT250
101     FORMAT(1H0,9X,3HDAY,I4,5X,4HHOUR,I4,5X,3HMIN,I4,5X,3HSEC,I4,5X)           MRT251
102     FORMAT(1H0,41X,5HPITCH,17X,3HYAW,16X,4HROLL,11X,9HMAGNITUDE)              MRT252
103     FORMAT(1H0,9X,24HTORQUE LIMITS ARE--UPPER,3E20.8)                         MRT253
104     FORMAT(29X,5HLOWER,3E20.8)                                                 MRT254
105     FORMAT(10X,24HTORQUE BY LEAST SQUARES ,4E20.8)                             MRT255
106     FORMAT(10X,24HRMS ERROR OF PARAB FIT   ,3E20.8//)                          MRT256
0107    FORMAT(1H1/15X,69HPLOT OF MARINER PITCH YAW AND ROLL TORQUES IN DY       MRT257
       $NE-CM -- STARTING DAY,3X,I5)                                               MRT258
109     FORMAT(32X,28HOUTAGE IN DATA AT THIS POINT)                                MRT259
110     FORMAT(25X,45HBIT ERROR AVERAGED OUT IN PREVIOUS DATA BATCH)               MRT260
111     FORMAT(40X,8HBAD DATA)                                                     MRT261
112     FORMAT(F4.1,I3,I3,6F5.2)                                                   MRT262
113     FORMAT(9F6.2)                                                              MRT263
114     FORMAT(1H0,9X,34HNULL INTERSECTION INTERVAL ON AXIS,I5)                    MRT264
120     FORMAT(1H0,2X,21HNUMBER OF POINTS USED,I4,24H  TORQUE INTERVALS-PI       MRT265
       $TCH,E16.8,5H  YAW,E16.8,6H  ROLL,E16.8///)                                 MRT266
201     FORMAT(1H0,9X,9HOUTPUT AT,I4,23H POINT OF PARABOLA WITH,I4,7H POIN       MRT267
       $TS)                                                                        MRT268
202     FORMAT(1H0,9X,24HMINIMUM RATE ERROR  AXIS,I4)                              MRT269
203     FORMAT(1H0,2X,14HFIRING ON AXIS,I3,2X,10HDEADBAND =E17.8,5X,               MRT270
       $ 16HRATE INCREMENT =E17.8)                                                 MRT271
204     FORMAT(29X,6HAT DAY,I4,6H  HOUR,I4,5H  MIN,I4,5H  SEC,I4)                  MRT272
301     FORMAT(1H ,6E20.8)                                                         MRT273
097     END                                                                        MRT274
$IBFTC PARINT
C                                                                                  PARINT
C       INTERVAL ARITHMETIC PARABOLA FIT                                           PARINT
C                                                                                  PARINT
001     SUBROUTINE PRIN(N,JSART,DELT,A,B,C,INX,NA)                                 PARINT
C                                                                                  PARINT
C       PRIN - 8 ARGUMENTS                                                         PARINT
C           INPUT                                                                  PARINT
C                 N          NUMBER OF POINTS                                      PARINT
C                 JSART      STARTING POINT OF PARABOLA                            PARINT
C                 DELT       TIME BETWEEN 2 POINTS IN MIN                          PARINT
C                 NA                                                               PARINT
```

```
C          OUTPUT                                                    PARINT
C               A(3,2)     COEFFICIENTS OF INTERVAL PARABOLA FIT     PARINT
C               B(3,2)     COEFFICIENTS OF INTERVAL PARABOLA FIT     PARINT
C               C(3,2)     COEFFICIENTS OF INTERVAL PARABOLA FIT     PARINT
C                                A(I,2),B(I,2),C(I,2)                PARINT
C                                     PITCH I=1                      PARINT
C                                     YAW   I=2                      PARINT
C                                     ROLL  I=3                      PARINT
C               INX        INDEX    =1  NULL INTERSECTION ON PITCH AXIS PARINT
C                                   =2  NULL INTERSECTION ON YAW AXIS   PARINT
C                                   =3  NULL INTERSECTION ON ROLL AXIS  PARINT
C                                                                    PARINT
002    COMMON AR(7,306),CKLIM(6),DEL                                 PARINT
003    DOUBLE PRECISION ERT(3,3),FRT(3,3)                            PARINT
004    DIMENSION A(3,2),B(3,2),C(3,2),QRT(3,2),SRT(3,3),RRT(3,2),G(2) PARINT
00401  DIMENSION FL(3,2),F(2),P(2),TH(2)                             PARINT
005    NHAF=N/2                                                      PARINT
00501  INX=0                                                         PARINT
00502  IND=1                                                         PARINT
00503  II=NA/5                                                       PARINT
00504  KK=NA/3                                                       PARINT
00505  ISKIP=2                                                       PARINT
00506  IF(DEL.GT.25.)ISKIP=0                                         PARINT
00507  JD=AR(4,JSART)                                                PARINT
00508  DO 64 L=1,3                                                   PARINT
006    DO 57 I=1,II                                                  PARINT
007    MI=JSART+I+ISKIP                                              PARINT
C                                                                    PARINT
C      SETUP OF E AND Q                                              PARINT
C                                                                    PARINT
008    IF(MI.GT.303)MI=MI-300                                        PARINT
009    CALL IANGPS(L,MI,TH,JD)                                       PARINT
010    QRT(1,2)=TH(2)                                                PARINT
011    QRT(1,1)=TH(1)                                                PARINT
012    X=I+ISKIP                                                     PARINT
013    DO 15 J=1,3                                                   PARINT
01301  JI=3-J                                                        PARINT
01302  IF(JI.EQ.0)GO TO 1402                                         PARINT
014    ERT(1,J)=DBLE((X*DELT)**JI)                                   PARINT
01401  GO TO 15                                                      PARINT
01402  ERT(1,J)=1.00D 00                                             PARINT
015    CONTINUE                                                      PARINT
016    DO 57 K=1,KK                                                  PARINT
017    MJ=JSART+NHAF+K-3                                             PARINT
018    IF(MJ.GT.303)MJ=MJ-300                                        PARINT
019    CALL IANGPS(L,MJ,TH,JD)                                       PARINT
020    QRT(2,2)=TH(2)                                                PARINT
021    QRT(2,1)=TH(1)                                                PARINT
022    X=NHAF+K-3                                                    PARINT
023    DO 26 J=1,3                                                   PARINT
024    JI=3-J                                                        PARINT
02401  IF(JI.EQ.0)GO TO 2502                                         PARINT
025    ERT(2,J)=DBLE((X*DELT)**JI)                                   PARINT
02501  GO TO 26                                                      PARINT
02502  ERT(2,J)=1.00D 00                                             PARINT
026    CONTINUE                                                      PARINT
027    DO 57 M=1,II                                                  PARINT
028    MK=JSART+N-M-1                                                PARINT
029    IF(MK.GT.303)MK=MK-300                                        PARINT
030    CALL IANGPS(L,MK,TH,JD)                                       PARINT
031    QRT(3,2)=TH(2)                                                PARINT
```

```
                                                                     CANCOR7
032     QRT(3,1)=TH(1)                                               PARINT
033     X=N-M-1                                                      PARINT
034     DO 37 J=1,3                                                  PARINT
035     JI=3-J                                                       PARINT
03501   IF(JI.EQ.0)GO TO 3602                                        PARINT
036     ERT(3,J)=DBLE((X*DELT)**JI)                                  PARINT
03601   GO TO 37                                                     PARINT
03602   ERT(3,J)=1.00D 00                                            PARINT
037     CONTINUE                                                     PARINT
C                                                                    PARINT
C       FIT CURVE AND SORT OUT DATA                                  PARINT
C                                                                    PARINT
038     CALL MATIN3(ERT,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,QRT,RRT)                                       PARINT
042     IF(I.NE.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(1)=RRT(J,1)                                                PARINT
046     G(2)=RRT(J,2)                                                PARINT
047     F(1)=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)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(1,1)                                               PARINT
059     A(L,2)=EL(1,2)                                               PARINT
060     B(L,1)=EL(2,1)                                               PARINT
061     B(L,2)=EL(2,2)                                               PARINT
062     C(L,1)=EL(3,1)                                               PARINT
063     C(L,2)=EL(3,2)                                               PARINT
064     CONTINUE                                                     PARINT
065     RETURN                                                       PARINT
066     END                                                          PARINT
$IBFTC MAXVT
001     SUBROUTINE MAVT(A,B,C)                                       MAXVT
C                                                                    MAXVT
C       MULTIPLICATION OF MATRIX BY INTERVAL VECTOR                  MAXVT
C       MAVT - 3 ARGUMENTS                                           MAXVT
C              INPUT                                                 MAXVT
C                    A(3,3)     3X3 MATRIC                           MAXVT
C                    B(3,2)     3 DIMENSIONAL VECTOR INTERVAL        MAXVT
C                                 B(I,1) UPPER                       MAXVT
C                                 B(I,2) LOWER                       MAXVT
C             OUTPUT                                                 MAXVT
C                    C(3,2)     3 DIMENSIONAL VECTOR INTERVAL   C=A*B MAXVT
C                                                                    MAXVT
002     DIMENSION A(3,3),B(3,2),C(3,2),S(2),D(2),F(2),G(2)           MAXVT
003     DO 18 I=1,3                                                  MAXVT
004     S(1)=0.                                                      MAXVT
```

```
      S(2)=0.                                                    MAXVT
      DO 15 J=1,3                                                MAXVT
      D(1)=B(J,1)                                                MAXVT
      D(2)=B(J,2)                                                MAXVT
      E=A(I,J)                                                   MAXVT
      CALL ISMULT(D,E,F)                                         MAXVT
      CALL IADD(F,S,G)                                           MAXVT
      S(1)=G(1)                                                  MAXVT
      S(2)=G(2)                                                  MAXVT
      CONTINUE                                                   MAXVT
      C(I,1)=S(1)                                                MAXVT
      C(I,2)=S(2)                                                MAXVT
      CONTINUE                                                   MAXVT
      RETURN                                                     MAXVT
      END                                                        MAXVT
$IBFTC IDMT
      SUBROUTINE ISMULT(B,C,F)                                   IDMT
C                                                                IDMT
C     MULTIPLICATION OF INTERVAL BY SCALAR CONSTANT              IDMT
C          INPUT                                                 IDMT
C                 B(2)         INTERVAL                          IDMT
C                 C            CONSTANT                          IDMT
C          OUTPUT                                                IDMT
C                 F(2)         INTERVAL, F = B*C                 IDMT
C                                                                IDMT
      DIMENSION B(2),F(2)                                        IDMT
      X=C*B(1)                                                   IDMT
      Y=C*B(2)                                                   IDMT
      F(1)=AMAX1(X,Y)                                            IDMT
      F(2)=AMIN1(X,Y)                                            IDMT
      RETURN                                                     IDMT
      END                                                        IDMT
$IBFTC ISUBTR
      SUBROUTINE ISUB(A,B,C)                                     ISUBTR
C                                                                ISUBTR
C     INTERVAL SUBTRACTION                                       ISUBTR
C          INPUT       A,B                                       ISUBTR
C          OUTPUT      C,    C = A-B                             ISUBTR
C                                                                ISUBTR
      DIMENSION A(2),B(2),C(2),D(2)                              ISUBTR
      D(1)=-B(1)                                                 ISUBTR
      D(2)=-B(2)                                                 ISUBTR
      CALL IADD(A,D,C)                                           ISUBTR
      RETURN                                                     ISUBTR
      END                                                        ISUBTR
$IBFTC IMULTP
      SUBROUTINE IMULT(A,B,C)                                    IMULTP
C                                                                IMULTP
C     INTERVAL MULTIPLICATION                                    IMULTP
C          INPUT       A,B                                       IMULTP
C          OUTPUT      C,    C = AXB                             IMULTP
C                                                                IMULTP
      DIMENSION A(2),B(2),C(2)                                   IMULTP
      W=B(1)*A(1)                                                IMULTP
      X=B(1)*A(2)                                                IMULTP
      Y=B(2)*A(1)                                                IMULTP
      Z=B(2)*A(2)                                                IMULTP
      C(1)=AMAX1(W,X,Y,Z)                                        IMULTP
      C(2)=AMIN1(W,X,Y,Z)                                        IMULTP
      RETURN                                                     IMULTP
      END                                                        IMULTP
```

```
$IBFTC PARF-
C                                                                       PARF-
C       LEAST SQUARES FIT A PARABOLA                                    PARF-
C                                                                       PARF-
1       SUBROUTINE PARFIT(T,N,P,DELT,ER)                                PARF-
C                                                                       PARF-
C       PARFIT - 5 ARGUMENTS                                            PARF-
C           INPUT                                                       PARF-
C                 T           POINTS                                    PARF-
C                 N           NUMBER OF POINTS                          PARF-
C           OUTPUT                                                      PARF-
C                 P(3)        LEAST SQUARE FIT COEFFICIENT              PARF-
C                 DELT        TIME BETWEEN 2 POINTS IN MIN              PARF-
C                 ER          MEAN LEAST SQUARE ERROR                   PARF-
C                                                                       PARF-
2       REAL T(1),Q(500,3),P(3),A(3,3),B(3),C(3)                        PARF-
3       DOUBLE PRECISION M(3,3),MINV(3,3),DET                           PARF-
C                                                                       PARF-
C       GENERATE Q                                                      PARF-
C                                                                       PARF-
4       DO 7 I=1,N                                                      PARF-
5       Q(I,1)=1.                                                       PARF-
6       Q(I,2)=FLOAT(I-1)*DELT                                          PARF-
7       Q(I,3)=(FLOAT(I-1)*DELT)**2                                     PARF-
C                                                                       PARF-
C       GENERATE Q' * Q                                                 PARF-
C                                                                       PARF-
8       DO 12 I=1,3                                                     PARF-
9       DO 12 J=1,3                                                     PARF-
10      M(I,J)=0.D0                                                     PARF-
11      DO 12 K=1,N                                                     PARF-
12      M(I,J)=M(I,J)+DBLE(Q(K,I)*Q(K,J))                               PARF-
C                                                                       PARF-
C       GENERATE Q' * T                                                 PARF-
C                                                                       PARF-
13      DO 16 I=1,3                                                     PARF-
14      B(I)=0.                                                         PARF-
15      DO 16 J=1,N                                                     PARF-
16      B(I)=B(I)+Q(J,I)*T(J)                                           PARF-
C                                                                       PARF-
C       SOLVE LINEAR EQNS                                               PARF-
C                                                                       PARF-
17      CALL MATIN3(M,MINV,ID)                                          PARF-
1701    DO 1703 I=1,3                                                   PARF-
1702    DO 1703 J=1,3                                                   PARF-
1703    A(I,J)=SNGL(MINV(I,J))                                          PARF-
1704    CALL MATVEC(A,B,C)                                              PARF-
18      P(1)=C(1)-0.25*C(2)*C(2)/C(3)                                   PARF-
19      P(2)=C(3)                                                       PARF-
20      P(3)=-0.5*C(2)/C(3)                                             PARF-
21      ER=0.                                                           PARF-
22      DO 24 I=1,N                                                     PARF-
23      F=T(I)-P(1)-P(2)*(FLOAT(I-1)*DELT-P(3))**2                      PARF-
24      ER=ER+F**2                                                      PARF-
25      ER=SQRT(ER/FLOAT(N))                                            PARF-
26      RETURN                                                          PARF-
        END                                                             PARF-
$IBFTC DECT
C                                                                       DECT
C       PULSE DETECT SUBROUTINE                                         DECT
C                                                                       DECT
```

```
001     SUBROUTINE DETECT(JSART,JFND,N,INX,IAXIS)                       DECT
C                                                                       DECT
C       DETECT - 5 ARGUMENTS                                            DECT
C           INPUT                                                       DECT
C                 JSART       STARTING POINT                            DECT
C           OUTPUT                                                      DECT
C                 JFNT        ENDING POINT                              DECT
C                 N           NUMBER OF POINTS                          DECT
C                 INX         INDEX    =1  OUTAGE IN DATA               DECT
C                                      =2  ERROR AVERAGED OUT           DECT
C                                      =3  BAD DATA                     DECT
C                 IAXIS       INDEX    =0  NO FIRING                    DECT
C                                      =1  FIRING IN PITCH              DECT
C                                      =2  FIRING IN YAW                DECT
C                                      =3  FIRING IN ROLL               DECT
C                                                                       DECT
002     DIMENSION K(3),DB(3)                                            DECT
003     LOGICAL TRUFAL                                                  DECT
004     COMMON AR(7,306),CKLIM(6),DEL                                   DECT
005     TRUFAL=.FALSE.                                                  DECT
00501 IND=0                                                             DECT
00502 INX=0                                                             DECT
00503 IAXIS=0                                                           DECT
C                                                                       DECT
C       CHECK TO SEE IF POINTS ARE NEAR DEADBAND                        DECT
C                                                                       DECT
006     L=JSART+1                                                       DECT
007     DO 15 I=L,600                                                   DECT
008     K(1)=0                                                          DECT
009     K(2)=0                                                          DECT
010     K(3)=0                                                          DECT
011     M=I                                                             DECT
012     IF(M.GT.303)M=M-300                                             DECT
01201 DO 1205 J=1,3                                                     DECT
01202 DB(J)=ABS(CKLIM(2*J)+.5)                                          DECT
01203 YY=POLY(J,AR(J,M))                                                DECT
01204 IF(YY.GT.0.)DB(J)=ABS(CKLIM(2*J-1)-.5)                            DECT
01205 CONTINUE                                                          DECT
01206 CALL CHEK(M,IND,DB)                                               DECT
01207 IF(IND.EQ.2)INX=2                                                 DECT
01208 IF(IND.EQ.1.OR.IND.EQ.3)GO TO 1501                                DECT
013     DO 15 J=1,3                                                     DECT
01301 JJ=J                                                              DECT
01302 YY=POLY(J,AR(J,M))                                                DECT
014     IF(ABS(YY).GT.DB(J))CALL PATRN(M,J,TRUFAL)                      DECT
015     IF(TRUFAL)GO TO 16                                              DECT
01501 JEND=I-1                                                          DECT
01502 INX=IND                                                           DECT
01503 GO TO 17                                                          DECT
016     JEND=I                                                          DECT
01601 IAXIS=JJ                                                          DECT
017     N=JEND-JSART+1                                                  DECT
018     RETURN                                                          DECT
019     END                                                             DECT
$IBFTC PAT
C                                                                       PAT
C       CHECK PATTERN FOR POSSIBLE FIRING                               PAT
C                                                                       PAT
001     SUBROUTINE PATRN(I,J,TRFL)                                      PAT
C                                                                       PAT
C       PAT - 3 ARGUMENTS                                               PAT
```

```
C          INPUT                                                         PAT
C                I            INDEX     PITCH I=1,  YAW I=2,  ROLL I=3   PAT
C                J            SUBSCRIPT FOR ARRAY AR                     PAT
C          OUTPUT                                                        PAT
C                TRFL         =.TRUE.  FOR PARTICULAR PATTERN            PAT
C                             =.FALSE. FOR OTHERS                        PAT
C                                                                        PAT
002   COMMON AR(7,306),CKLIM(6),DEL                                      PAT
00201  DIMENSION A(7)                                                    PAT
003   LOGICAL TRFL                                                       PAT
00301 TRFL=.FALSE.                                                       PAT
00302 A(1)=ABS(POLY(J,AR(J,I-3)))                                        PAT
00303 A(2)=ABS(POLY(J,AR(J,I-2)))                                        PAT
00304 A(3)=ABS(POLY(J,AR(J,I-1)))                                        PAT
00305 A(4)=ABS(POLY(J,AR(J,I)))                                          PAT
00306 A(5)=ABS(POLY(J,AR(J,I+1)))                                        PAT
00307 A(6)=ABS(POLY(J,AR(J,I+2)))                                        PAT
004   A(7)=ABS(POLY(J,AR(J,I+3)))                                        PAT
00401 IF(A(1).NE.A(4))GO TO 5                                            PAT
00402 IF(A(2).NE.A(4))GO TO 5                                            PAT
00403 IF(A(3).NE.A(4))GO TO 5                                            PAT
00404 IF(A(5).GE.A(4))GO TO 5                                            PAT
00405 IF(A(6).GE.A(4))GO TO 5                                            PAT
00406 IF(A(7).GE.A(4))GO TO 5                                            PAT
00407 GO TO 8                                                            PAT
005   IF(A(3).GE.A(4).OR.A(4).LT.A(5))RETURN                             PAT
006   IF(A(1).GT.A(4))RETURN                                             PAT
00601 IF(A(2).GT.A(4))RETURN                                             PAT
00602 IF(A(6).GT.A(4))RETURN                                             PAT
007   IF(A(7).GT.A(4))RETURN                                             PAT
008   TRFL=.TRUE.                                                        PAT
009   RETURN                                                             PAT
010   END                                                                PAT
$IBFTC INTX
C                                                                        INTX
C     INTERVAL INTERSECTION                                              INTX
C                                                                        INTX
001   SUBROUTINE INTERX(F,G,P,IND)                                       INTX
C                                                                        INTX
C          INPUT                                                         INTX
C                F(2)         INTERVAL                                   INTX
C                G(2)         INTERVAL                                   INTX
C          OUTPUT                                                        INTX
C                P(2)         = F(2) IF NO INTERSECTION                  INTX
C                             = F(2) INTERSECTION G(2)                   INTX
C                IND          INDEX     =0  IF NO INTERSECTION           INTX
C                                       =1  IF INTERSECTION              INTX
C                                                                        INTX
002   DIMENSION F(2),G(2),P(2)                                           INTX
003   IND=1                                                              INTX
004   IF(F(2).LT.F(1))GO TO 8                                            INTX
005   E=F(2)                                                             INTX
006   F(2)=F(1)                                                          INTX
007   F(1)=E                                                             INTX
008   IF(G(2).LT.G(1))GO TO 12                                           INTX
009   E=G(2)                                                             INTX
010   G(2)=G(1)                                                          INTX
011   G(1)=E                                                             INTX
012   IF(F(2).GT.G(1).OR.G(2).GT.F(1))GO TO 20                           INTX
013   P(1)=G(1)                                                          INTX
014   P(2)=G(2)                                                          INTX
```

```
015   IF(G(1).GT.F(1))P(1)=F(1)                                          INTX
016   IF(F(2).GT.G(2))P(2)=F(2)                                          INTX
017   RETURN                                                             INTX
020   IND=0                                                              INTX
021   P(1)=F(1)                                                          INTX
022   P(2)=F(2)                                                          INTX
023   RETURN                                                             INTX
024   END                                                                INTX
$IBFTC IDA
C                                                                        IDA
C     INTERVAL ADDITION                                                  IDA
C                                                                        IDA
001   SUBROUTINE IADD(F,G,SM)                                            IDA
C                                                                        IDA
C          INPUT      F(2),G(2)                                          IDA
C          OUTPUT     SM(2)      SM=F+G                                  IDA
C                                                                        IDA
002   DIMENSION F(2),G(2),SM(2)                                          IDA
003   IF(F(1).GT.F(2))GO TO 7                                            IDA
004   E=F(2)                                                             IDA
005   F(2)=F(1)                                                          IDA
006   F(1)=E                                                             IDA
007   IF(G(1).GT.G(2))GO TO 11                                           IDA
008   E=G(2)                                                             IDA
009   G(2)=G(1)                                                          IDA
010   G(1)=E                                                             IDA
011   SM(1)=F(1)+G(1)                                                    IDA
012   SM(2)=F(2)+G(2)                                                    IDA
013   RETURN                                                             IDA
014   END                                                                IDA
$IBFTC MATT
C                                                                        MATT
C     3X3 MATRIX INVERSION                                               MATT
C                                                                        MATT
001   SUBROUTINE MATIN3(A,B,I)                                           MATT
C                                                                        MATT
C          INPUT      A(3,3)      GIVEN MATRIX                           MATT
C          OUTPUT     B(3,3)      INVERSE MATRIX OF A                    MATT
C                     I           INDEX     =1 FOR NON SINGULAR MATRIX   MATT
C                                           =0 FOR SINGULAR MATRIX       MATT
C                                                                        MATT
002   DOUBLE PRECISION A(3,3),B(3,3),DET                                 MATT
00201 I=1                                                                MATT
003   DET=A(1,1)*A(2,2)*A(3,3)+A(1,2)*A(2,3)*A(3,1)+A(1,3)*A(2,1)*A(3,2) MATT
00301$-A(1,3)*A(2,2)*A(3,1)-A(1,2)*A(2,1)*A(3,3)-A(1,1)*A(3,2)*A(2,3)    MATT
004   IF(DABS(DET).LT.1.D-15)GO TO 15                                    MATT
005   B(1,1)=(A(2,2)*A(3,3)-A(2,3)*A(3,2))/DET                           MATT
006   B(1,2)=-(A(1,2)*A(3,3)-A(1,3)*A(3,2))/DET                          MATT
007   B(1,3)=(A(1,2)*A(2,3)-A(1,3)*A(2,2))/DET                           MATT
008   B(2,1)=-(A(2,1)*A(3,3)-A(2,3)*A(3,1))/DET                          MATT
009   B(2,2)=(A(1,1)*A(3,3)-A(1,3)*A(3,1))/DET                           MATT
010   B(2,3)=-(A(1,1)*A(2,3)-A(2,1)*A(1,3))/DET                          MATT
011   B(3,1)=(A(2,1)*A(3,2)-A(3,1)*A(2,2))/DET                           MATT
012   B(3,2)=-(A(1,1)*A(3,2)-A(1,2)*A(3,1))/DET                          MATT
013   B(3,3)=(A(1,1)*A(2,2)-A(1,2)*A(2,1))/DET                           MATT
014   RETURN                                                             MATT
015   I=0                                                                MATT
016   RETURN                                                             MATT
017   END                                                                MATT
$IBFTC CHECK
C                                                                        CHECK
```

```
C       DATA CHECK                                                      CHECK
C                                                                       CHECK
001     SUBROUTINE CHEK(M,N,DB)                                         CHECK
C                                                                       CHECK
C           INPUT    M       SUBSCRIPT FOR ARRAY AR                     CHECK
C           OUTPUT   N       INDEX    =1  OUTAGE IN DATA                CHECK
C                                     =2  ERROR AVERAGED OUT            CHECK
C                                     =3  BAD DATA                      CHECK
C                    DB(3)   DEADBAND,   PITCH-YAW-ROLL                 CHECK
C                                                                       CHECK
002     COMMON AR(7,306),CKLIM(6),DEL                                   CHECK
00201   DIMENSION B(3),DB(3)                                            CHECK
00202   DO 203 I=1,3                                                    CHECK
00203   B(I)=DB(I)+5.00                                                 CHECK
003     IF(ABS(AR(5,M)-AR(5,M-1)).LT.0.5)GO TO 6                        CHECK
004     TMIN=60.0                                                       CHECK
005     GO TO 7                                                         CHECK
006     TMIN=AR(6,M)                                                    CHECK
007     TIM1=AR(7,M-1)+60.0*AR(6,M-1)+DEL                               CHECK
008     TIM2=AR(7,M)+60.0*TMIN                                          CHECK
009     IF(ABS(TIM1-TIM2).LT.10.0)GO TO 19                              CHECK
010     N=1                                                             CHECK
011     RETURN                                                          CHECK
C                                                                       CHECK
C       CHECK FOR BIT ERROR                                             CHECK
C                                                                       CHECK
019     DO 20 J=1,3                                                     CHECK
01901   A1=POLY(J,AR(J,M))                                              CHECK
01902   A2=POLY(J,AR(J,M-1))                                            CHECK
020     IF(ABS(A2-A1).GT.2.0)GO TO 22                                   CHECK
021     RETURN                                                          CHECK
022     N=2                                                             CHECK
023     DO 24 K=1,3                                                     CHECK
02301   A1=POLY(K,AR(K,M))                                              CHECK
024     IF(ABS(A1).GT.B(K))GO TO 31                                     CHECK
C                                                                       CHECK
C       AVERAGE OUT BIT ERRORS                                          CHECK
C                                                                       CHECK
025     DO 26 L=1,3                                                     CHECK
02501   A1=POLY(L,AR(L,M+1))                                            CHECK
026     IF(ABS(A1).GT.B(K))GO TO 35                                     CHECK
027     DO 29 LI=1,3                                                    CHECK
02701   A1=POLY(LI,AR(LI,M))                                            CHECK
02702   A2=POLY(LI,AR(LI,M-1))                                          CHECK
028     IF(ABS(A2-A1).GT.2.)AR(LI,M)=.5*(AR(LI,M-1)+AR(LI,M+1))         CHECK
029     CONTINUE                                                        CHECK
030     RETURN                                                          CHECK
031     DO 33 LK=1,3                                                    CHECK
03101   A1=POLY(LK,AR(LK,M+1))                                          CHECK
032     IF(ABS(A1).GT.B(LK))GO TO 35                                    CHECK
033     AR(LK,M)=.5*(AR(LK,M-1)+AR(LK,M+1))                             CHECK
034     RETURN                                                          CHECK
035     N=3                                                             CHECK
036     RETURN                                                          CHECK
038     END                                                             CHECK
$IBFTC TRANSF
C                                                                       TRANSF
C       SUBROUTINE--MULTIPLY VECTOR BY A SQUARE MATRIX--LINEAR TRANSFORMAT
001     SUBROUTINE MATVEC(Y,W,R)
002     DIMENSION Y(3,3),W(3),R(3)
003     R(1)=Y(1,1)*W(1)+Y(1,2)*W(2)+Y(1,3)*W(3)
```

```
004    R(2)=Y(2,1)*W(1)+Y(2,2)*W(2)+Y(2,3)*W(3)
005    R(3)=Y(3,1)*W(1)+Y(3,2)*W(2)+Y(3,3)*W(3)
006    RETURN
007    END
$IBFTC RELT
C                                                                      RELT
C      ADDITION OF TIME                                                RELT
C                                                                      RELT
001    SUBROUTINE RELTIM(T,M,REALT)                                    RELT
C                                                                      RELT
C      REALTIM - 3 ARGUMENTS                                           RELT
C          INPUT                                                       RELT
C               T           TIME IN MIN                                RELT
C               M(4)        TIME,DAY-HR-MIN-SEC                        RELT
C          OUTPUT                                                      RELT
C               REALT(4)    TIME,DAY-HR-MIN-SEC,  (REALT) = (T)+(M)    RELT
C                                                                      RELT
002    INTEGER REALT(4)                                                RELT
003    REAL M(4)                                                       RELT
005    J=T                                                             RELT
006    XJ=J                                                            RELT
007    X=M(3)+XJ                                                       RELT
008    Y=(T-XJ)*60.                                                    RELT
009    Z=M(4)+Y                                                        RELT
010    IF(Z.LT.60.)GO TO 14                                            RELT
011    REALT(4)=Z-60.                                                  RELT
012    X=M(3)+XJ+1.                                                    RELT
013    GO TO 15                                                        RELT
014    REALT(4)=Z                                                      RELT
015    IF(X.LT.60.)GO TO 19                                            RELT
016    REALT(3)=X-60.                                                  RELT
017    Y=M(2)+1.                                                       RELT
018    GO TO 21                                                        RELT
019    REALT(3)=X                                                      RELT
020    Y=M(2)                                                          RELT
021    IF(Y.LT.24.)GO TO 25                                            RELT
022    REALT(2)=Y-24.                                                  RELT
023    REALT(1)=M(1)+1.                                                RELT
024    RETURN                                                          RELT
025    REALT(2)=Y                                                      RELT
026    REALT(1)=M(1)                                                   RELT
028    RETURN                                                          RELT
029    END                                                             RELT
$IBFTC IANG
C                                                                      IANG64
C      COMPUTATION OF INTERVALS FOR ANGULAR POSITION                   IANG64
C                                                                      IANG64
001    SUBROUTINE IANGPS(L,M,TH,ID)                                    IANG64
C                                                                      IANG64
C      IANGPS - 4 ARGUMENTS                                            IANG64
C          INPUT                                                       IANG64
C               L           INDEX      PITCH FOR L=1                   IANG64
C                                      YAW   FOR L=2                   IANG64
C                                      ROLL  FOR L=3                   IANG64
C               L,M         SUBSCRIPTS FOR ARRAY AR                    IANG64
C               ID          TIME IN DAYS FROM JAN 1                    IANG64
C          OUTPUT                                                      IANG64
C               TH(2)       INTERVAL OF ANGULAR POSITION, (UPPER,LOWER)  IANG64
C                                                                      IANG64
002    DIMENSION TH(2),THX(2),THY(2),THZ(2),T1(2),T2(2),T3(2)          IANG64
00201  COMMON AR(7,306),CKLIM(6),DEL                                   IANG64
```

```
003    COMMON/CANOP1/CLOCK/CANOP2/CONE(275)                              IANG64
C                                                                        IANG64
C      JD = TIME AFTER LANCH IN DAYS                                     IANG64
C                                                                        IANG64
004    IF(ID.LT.330) JD=ID+34                                            IANG64
00401  IF(ID.GT.330) JD=ID-331                                           IANG64
005    THETA1=CLOCK*.17453293E-01                                        IANG64
006    THETA2=CONE(JD)*.17453293E-01                                     IANG64
007    C1=-COS(THETA1)                                                   IANG64
008    C2=-COS(THETA2)                                                   IANG64
009    S1=SIN(THETA1)                                                    IANG64
010    S2=SIN(THETA2)                                                    IANG64
011    IF(L.EQ.3)GO TO 16                                                IANG64
012    DN=AR(L,M)                                                        IANG64
C                                                                        IANG64
C      POLY IS CALIBRATION FUNCTION                                      IANG64
C                                                                        IANG64
013    TH(1)=POLY(L,DN-.57)                                              IANG64
014    TH(2)=POLY(L,DN+.57)                                              IANG64
015    RETURN                                                            IANG64
016    DN=AR(1,M)                                                        IANG64
C                                                                        IANG64
C      NEED TO CALCULATE TH FOR PITCH AND YAW TO OBTAIN TH FOR ROLL      IANG64
C                                                                        IANG64
017    THX(1)=POLY(1,DN-.57)                                             IANG64
018    THX(2)=POLY(1,DN+.57)                                             IANG64
019    DN=AR(2,M)                                                        IANG64
020    THY(1)=POLY(2,DN-.57)                                             IANG64
021    THY(2)=POLY(2,DN+.57)                                             IANG64
022    DN=AR(3,M)                                                        IANG64
023    THZ(1)=POLY(3,DN-.70)                                             IANG64
024    THZ(2)=POLY(3,DN+.70)                                             IANG64
025    CALL ISMULT(THX,C1,T1)                                            IANG64
026    CALL ISMULT(THY,S1,T2)                                            IANG64
027    CALL IADD(T1,T2,T3)                                               IANG64
028    CALL ISMULT(T3,C2,T1)                                             IANG64
029    CALL IADD(THZ,T1,T2)                                              IANG64
030    A=1./S2                                                           IANG64
031    CALL ISMULT(T2,A,T3)                                              IANG64
032    TH(1)=T3(1)                                                       IANG64
033    TH(2)=T3(2)                                                       IANG64
034    RETURN                                                            IANG64
035    END                                                               IANG64
$IBFTC IANG
C                                                                        IANG67
C      COMPUTATION OF INTERVALS FOR ANGULAR POSITION                     IANG67
C                                                                        IANG67
001    SUBROUTINE IANGPS(L,M,TH,ID)                                      IANG67
C                                                                        IANG67
C      IANGPS - 4 ARGUMENTS                                              IANG67
C              INPUT                                                     IANG67
C                 L            INDEX      PITCH FOR L=1                  IANG67
C                                         YAW   FOR L=2                  IANG67
C                                         ROLL  FOR L=3                  IANG67
C                 L,M          SUBSCRIPTS FOR ARRAY AR                   IANG67
C                 ID           TIME IN DAYS FROM JAN 1                   IANG67
C              OUTPUT                                                    IANG67
C                 TH(2)        INTERVAL OF ANGULAR POSITION, (UPPER,LOWER)  IANG67
C                                                                        IANG67
002    DIMENSION TH(2),THX(2),THY(2),THZ(2),T1(2),T2(2),T3(2)            IANG67
00201  COMMON AR(7,306),CKLIM(6),DEL                                     IANG67
```

```
      COMMON/CANOP1/CLOCK/CANOP2/CONE(135)                              IANG67
C                                                                       IANG67
C     JD = TIME AFTER LANCH IN DAYS                                     IANG67
C                                                                       IANG67
      JD=ID-164                                                         IANG67
      THETA1=CLOCK*.17453293E-01                                        IANG67
      THETA2=CONE(JD)*.17453293E-01                                     IANG67
      C1=-COS(THETA1)                                                   IANG67
      C2=-COS(THETA2)                                                   IANG67
      S1=SIN(THETA1)                                                    IANG67
      S2=SIN(THETA2)                                                    IANG67
      IF(L.EQ.3)GO TO 16                                                IANG67
      DN=AR(L,M)                                                        IANG67
C                                                                       IANG67
C     POLY IS CALIBRATION FUNCTION                                      IANG67
C                                                                       IANG67
      TH(1)=POLY(L,DN-.57)                                              IANG67
      TH(2)=POLY(L,DN+.57)                                              IANG67
      RETURN                                                            IANG67
   16 DN=AR(1,M)                                                        IANG67
C                                                                       IANG67
C     NEED TO CALCULATE TH FOR PITCH AND YAW TO OBTAIN TH FOR ROLL      IANG67
C                                                                       IANG67
      THX(1)=POLY(1,DN-.57)                                             IANG67
      THX(2)=POLY(1,DN+.57)                                             IANG67
      DN=AR(2,M)                                                        IANG67
      THY(1)=POLY(2,DN-.57)                                             IANG67
      THY(2)=POLY(2,DN+.57)                                             IANG67
      DN=AR(3,M)                                                        IANG67
      THZ(1)=POLY(3,DN-.75)                                             IANG67
      THZ(2)=POLY(3,DN+.75)                                             IANG67
      CALL ISMULT(THX,C1,T1)                                            IANG67
      CALL ISMULT(THY,S1,T2)                                            IANG67
      CALL IADD(T1,T2,T3)                                               IANG67
      CALL ISMULT(T3,C2,T1)                                             IANG67
      CALL IADD(THZ,T1,T2)                                              IANG67
      A=1./S2                                                           IANG67
      CALL ISMULT(T2,A,T3)                                              IANG67
      TH(1)=T3(1)                                                       IANG67
      TH(2)=T3(2)                                                       IANG67
      RETURN                                                            IANG67
      END                                                               IANG67
$IBFTC MNRT
C                                                                       MINRAT
C     COMPUTATION OF MINIMUM RATE INCREMENT                             MINRAT
C                                                                       MINRAT
      SUBROUTINE MINRAT(W,WP,NJJ,DELT,RATINC,REALT,DEDZON,INDCAT,MJJ)   MINRAT
C                                                                       MINRAT
C     MINRAT = 9 ARGUMENTS                                              MINRAT
C            INPUT                                                      MINRAT
C                 W(3)        COEFFICIENTS OF PRESENT PARABOLA          MINRAT
C                 WP(3)       COEFFICIENTS OF LAST PARABOLA             MINRAT
C                 NJJ         NUMBER OF POINTS USED IN LAST PARABOLA    MINRAT
C                 DELT        TIME BETWEEN 2 POINTS IN MIN              MINRAT
C                 MJJ(4)      STARTING TIME OF LAST PARABOLA,DAY-HR-MIN-SEC MINRAT
C            OUTPUT                                                     MINRAT
C                 RATINC      RATE INCREMENTS IN DYNE-CM*CM             MINRAT
C                 REALT       TIME OF FIRING,DAY-HR-MIN-SEC             MINRAT
C                 DEDZON      DEADBAND                                  MINRAT
C                 INDCAT      INDEX    =1  COMPUTATIONAL ERROR          MINRAT
C                                      =0  NO COMPUTATIONAL ERROR       MINRAT
```

```
C                                                                       MINRAT
00101 COMMON/LVT/LEVOUT                                                 MINRAT
002   DIMENSION W(3),WP(3)                                              MINRAT
00201 INTEGER REALT(4)                                                  MINRAT
00202 REAL MJJ(4)                                                       MINRAT
00203 FL=14.                                                            MINRAT
00204 IF(LEVOUT.EQ.2) FL=2.                                             MINRAT
003   INDCAT=0                                                          MINRAT
004   X=NJJ-1                                                           MINRAT
C                                                                       MINRAT
C     SHIFT PRESENT PARABOLA AND SOLVE QUADRADIC EQUATION IN T          MINRAT
C                                                                       MINRAT
005   SHIFT=X*DELT                                                      MINRAT
006   A=WP(1)-W(1)                                                      MINRAT
007   B=WP(2)+2.*W(1)*SHIFT-W(2)                                        MINRAT
008   C=WP(3)-(SHIFT**2)*W(1)+W(2)*SHIFT-W(3)                           MINRAT
00801 D=B**2-4.*A*C                                                     MINRAT
00802 IF(D)19,9,9                                                       MINRAT
009   DIS=SQRT(D)                                                       MINRAT
010   T=(-B-DIS)/(2.*A)                                                 MINRAT
011   IF(ABS(SHIFT-T).LT.FL*DELT)GO TO 14                               MINRAT
012   T=(-B+DIS)/(2.*A)                                                 MINRAT
013   IF(ABS(SHIFT-T).GT.FL*DELT)GO TO 19                               MINRAT
014   RT1=2.*WP(1)*T+WP(2)                                              MINRAT
015   RT2=2.*W(1)*(T-SHIFT)+W(2)                                        MINRAT
016   RATINC=(RT2-RT1)*100./6.                                          MINRAT
017   DEDZON=WP(1)*T**2+WP(2)*T+WP(3)                                   MINRAT
01701 CALL RELTIM(T,MJJ,REALT)                                          MINRAT
018   RETURN                                                            MINRAT
019   INDCAT=1                                                          MINRAT
020   RETURN                                                            MINRAT
021   END                                                               MINRAT
$IBFTC MRVSDN
C                                                                       MRVDSN
C     ANGULAR POSITION VS DATA NUMBER--CALIBRATION FOR MV-67            MRVDSN
C                                                                       MRVDSN
001   FUNCTION POLY(L,DN)                                               MRVDSN
002   DIMENSION A(6,3)                                                  MRVDSN
003   DATA(A(I,1),I=1,18)/25.326053,-.92093034,.19287685E-01,-.28223686E MRVDSN
0031 $-03,0.21335972E-05,-.66782966E-08,28.046241,-1.0178037,.20517513E- MRVDSN
0032 $01,-.28572125E-03,.20782001E-05,-.63998052E-08,21.414102,-.3237947 MRVDSN
0033 $1,.11165038E-01,-.34020915E-03,0.33585269E-05,-.10896906E-07/     MRVDSN
004   POLY=A(1,L)+A(2,L)*DN+A(3,L)*DN**2+A(4,L)*DN**3+A(5,L)*DN**4+A(6,L MRVDSN
0041 $)*DN**5                                                           MRVDSN
005   RETURN                                                            MRVDSN
006   END                                                               MRVDSN
$IBFTC MRVSDN
C                                                                       MRVDSN
C     ANGULAR POSITION VS DATA NUMBER--CALIBRATION FOR MM-64            MRVDSN
C                                                                       MRVDSN
001   FUNCTION POLY(L,DN)                                               MRVDSN
002   DIMENSION A(4,3)                                                  MRVDSN
003   DATA (A(I,1),I=1,12)/13.816884,-.27638703,.13834448E-02,-.70307050 MRVDSN
0031 $E-05,13.816884,-.27638703,.13834448E-02,-.70307050E-05,26.5,-.414, MRVDSN
0032 $0.,0./                                                            MRVDSN
004   POLY=A(1,L)+A(2,L)*DN+A(3,L)*DN**2+A(4,L)*DN**3                   MRVDSN
005   RETURN                                                            MRVDSN
006   END                                                               MRVDSN
$IBFTC ANGLE
C                                                                       ANGLE4
C     THIS IS FOR MM-64                                                 ANGLE4
```

```
C                                                                       ANGLE4
      BLOCK DATA                                                        ANGLE4
      COMMON/CANOP1/CLOCK                                               ANGLE4
      COMMON/CANOP2/CONE(275)                                           ANGLE4
      COMMON/COEFF/AA(4,3)                                              ANGLE4
      DATA CLOCK/-.56E02/                                               ANGLE4
      DATA (CONE(I),I=1,75)/                                            ANGLE4
     1    101.80,    101.95,    102.09,    102.23,    102.36,            ANGLE4
     2    102.49,    102.61,    102.72,    102.83,    102.93,            ANGLE4
     3    103.03,    103.13,    103.21,    103.30,    103.78,            ANGLE4
     4    103.45,    103.52,    103.58,    103.64,    103.70,            ANGLE4
     5    103.75,    103.79,    103.83,    103.87,    103.90,            ANGLE4
     6    103.93,    103.95,    103.97,    103.98,    103.99,            ANGLE4
     7    104.00,    104.00,    104.00,    104.00,    103.99,            ANGLE4
     8    103.97,    103.96,    103.94,    103.91,    103.89,            ANGLE4
     9    103.85,    103.82,    103.78,    103.74,    103.70,            ANGLE4
     1    103.65,    103.60,    103.55,    103.49,    103.43,            ANGLE4
     2    103.37,    103.30,    103.24,    103.16,    103.09,            ANGLE4
     3    103.01,    102.94,    102.86,    102.77,    102.69,            ANGLE4
     4    102.60,    102.51,    102.42,    102.33,    102.24,            ANGLE4
     5    102.15,    102.05,    101.95,    101.85,    101.74,            ANGLE4
     6    101.64,    101.53,    101.42,    101.31,    101.20/            ANGLE4
      DATA (CONE(I),I=76,150)/                                          ANGLE4
     7    101.09,    100.97,    100.85,    100.74,    100.62,            ANGLE4
     8    100.49,    100.37,    100.25,    100.12,     99.99,            ANGLE4
     9     99.86,     99.73,     99.60,     99.47,     99.34,            ANGLE4
     1     99.20,     99.06,     98.92,     98.78,     98.64,            ANGLE4
     2     98.50,     98.36,     98.21,     98.07,     97.92,            ANGLE4
     3     97.78,     97.63,     97.48,     97.33,     97.19,            ANGLE4
     4     97.04,     96.89,     96.74,     96.59,     96.44,            ANGLE4
     5     96.21,     96.14,     95.99,     95.84,     95.70,            ANGLE4
     6     95.54,     95.39,     95.25,     95.10,     94.95,            ANGLE4
     7     94.80,     94.66,     94.52,     94.38,     94.24,            ANGLE4
     8     94.10,     93.97,     93.83,     93.69,     93.56,            ANGLE4
     9     93.42,     93.28,     93.15,     93.02,     92.88,            ANGLE4
     1     92.75,     92.62,     92.48,     92.35,     92.22,            ANGLE4
     2     92.09,     91.96,     91.83,     91.70,     91.57,            ANGLE4
     3     91.44,     91.31,     91.18,     91.05,     90.93/            ANGLE4
      DATA (CONE(I),I=151,225)/                                         ANGLE4
     4     90.80,     90.68,     90.55,     90.43,     90.30,            ANGLE4
     5     90.18,     90.05,     89.93,     89.81,     89.69,            ANGLE4
     6     89.57,     89.44,     89.32,     89.20,     89.08,            ANGLE4
     7     88.96,     88.84,     88.72,     88.60,     88.49,            ANGLE4
     8     88.37,     88.25,     88.13,     88.02,     87.90,            ANGLE4
     9     87.78,     87.66,     87.55,     87.43,     87.32,            ANGLE4
     1     87.20,     87.08,     86.97,     86.85,     86.74,            ANGLE4
     2     86.62,     86.50,     86.39,     86.27,     86.16,            ANGLE4
     3     86.04,     85.93,     85.82,     85.70,     85.59,            ANGLE4
     4     85.48,     85.36,     85.30,     85.14,     85.02,            ANGLE4
     5     84.91,     84.80,     84.69,     84.58,     84.47,            ANGLE4
     6     84.35,     84.24,     84.13,     84.02,     83.91,            ANGLE4
     7     83.83,     83.69,     83.58,     83.47,     83.36,            ANGLE4
     8     83.25,     83.14,     83.04,     82.93,     82.82,            ANGLE4
     9     82.71,     82.60,     82.50,     82.39,     82.28/            ANGLE4
      DATA (CONE(I),I=226,275)/                                         ANGLE4
     1     82.18,     82.07,     81.96,     81.86,     81.75,            ANGLE4
     2     81.64,     81.54,     81.43,     81.33,     81.22,            ANGLE4
     3     81.12,     81.02,     80.91,     80.81,     80.70,            ANGLE4
     4     80.60,     80.50,     80.39,     80.29,     80.19,            ANGLE4
     5     80.09,     79.98,     79.88,     79.78,     79.68,            ANGLE4
     6     79.58,     79.48,     79.38,     79.28,     79.18,            ANGLE4
```

```
      7      79.08,      78.98,      78.88,      78.78,      78.68,             ANGLE4
      8      78.58,      78.48,      78.38,      78.28,      78.18,             ANGLE4
      9      78.09,      77.99,      77.89,      77.79,      77.70,             ANGLE4
      1      77.60,      77.50,      77.40,      77.30,      77.20/             ANGLE4
       DATA(AA(I,1),I=1,12)/13.816884,-.27638703,.13834448E-02,-.70307050 ANGLE4
      $E-05,13.816884,-.27638703,.13834448E-02,-.70307050E-05,26.5,-.414,  ANGLE4
      $0.,0./                                                             ANGLE4
       END                                                                ANGLE4
$IBFTC ANGLE
C                                                                         ANGLE7
C      THIS IS FOR MV-67                                                  ANGLE7
C                                                                         ANGLE7
       BLOCK DATA                                                         ANGLE7
       COMMON/CANOP1/CLOCK/CANOP2/CONE(135)                               ANGLE7
       COMMON/COEFF/A(6,3)                                                ANGLE7
       DATA CLOCK/ .45E02/                                                ANGLE7
       DATA(CONE(I),I=5,75 )/ .7671E02,      .7670E02,       .7668E02,    ANGLE7
      8     .7666E02,       .7665E02,       .7664E02,       .7663E02,     ANGLE7
      8     .7663E02,       .7663E02,       .7663E02,       .7664E02,     ANGLE7
      8     .7665E02,       .7666E02,       .7668E02,       .7670E02,     ANGLE7
      8     .7672E02,       .7675E02,       .7678E02,       .7681E02,     ANGLE7
      8     .7685E02,       .7689E02,       .7694E02,       .7699E02,     ANGLE7
      8     .7704E02,       .7710E02,       .7716E02,       .7722E02,     ANGLE7
      8     .7729E02,       .7736E02,       .7744E02,       .7752E02,     ANGLE7
      8     .7760E02,       .7769E02,       .7778E02,       .7787E02,     ANGLE7
      8     .7797E02,       .7808E02,       .7819E02,       .7830E02,     ANGLE7
      8     .7842E02,       .7854E02,       .7866E02,       .7879E02,     ANGLE7
      8     .7892E02,       .7906E02,       .7920E02,       .7935E02,     ANGLE7
      8     .7950E02,       .7966E02,       .7982E02,       .7998E02,     ANGLE7
      8     .8015E02,       .8033E02,       .8051E02,       .8069E02,     ANGLE7
      8     .8088E02,       .8107E02,       .8127E02,       .8147E02,     ANGLE7
      8     .8168E02,       .8189E02,       .8210E02,       .8232E02,     ANGLE7
      8     .8255E02,       .8278E02,       .8302E02,       .8326E02,     ANGLE7
      8     .8350E02,       .8375E02,       .8400E02,       .8426E02/     ANGLE7
       DATA(CONE(I),I=76,135)/.8452E02,      .8479E02,       .8506E02,    ANGLE7
      8     .8534E02,       .8562E02,       .8590E02,       .8619E02,     ANGLE7
      8     .8649E02,       .8678E02,       .8708E02,       .8739E02,     ANGLE7
      8     .8770E02,       .8801E02,       .8833E02,       .8865E02,     ANGLE7
      8     .8897E02,       .8929E02,       .8962E02,       .8995E02,     ANGLE7
      8     .9029E02,       .9062E02,       .9096E02,       .9130E02,     ANGLE7
      8     .9165E02,       .9199E02,       .9233E02,       .9268E02,     ANGLE7
      8     .9303E02,       .9337E02,       .9372E02,       .9406E02,     ANGLE7
      8     .9441E02,       .9475E02,       .9510E02,       .9544E02,     ANGLE7
      8     .9578E02,       .9612E02,       .9645E02,       .9678E02,     ANGLE7
      8     .9711E02,       .9743E02,       .9775E02,       .9807E02,     ANGLE7
      8     .9837E02,       .9868E02,       .9898E02,       .9927E02,     ANGLE7
      8     .9955E02,       .9983E02,       .10010E03,      .10037E03,    ANGLE7
      8     .10064E03,      .10090E03,      .10116E03,      .10141E03,    ANGLE7
      8     .10165E03,      .10188E03,      .10210E03,      .10231E03,    ANGLE7
      8     .10252E03/                                                    ANGLE7
       DATA(A(I,1),I=1,18)/25.326053,-.92093034,.19287685E-01,-.28223686E ANGLE7
      $-03,0.21335972E-05,-.66782966E-08,28.046241,-1.0178037,.20517513E-  ANGLE7
      $01,-.28572125E-03,.20782001E-05,-.63998052E-08,21.414102,-.3237947  ANGLE7
      $1,.11165038E-01,-.34020915E-03,0.33585269E-05,-.10896906E-07/      ANGLE7
       END                                                                ANGLE7
$IBFTC CANCOR
C                                                                        CANCOR4
C      THIS IS FOR MM-64                                                 CANCOR4
C      YOU ALSO NEED SUBPROGRAM BLOCK DATA                               CANCOR4
C                                                                        CANCOR4
       SUBROUTINE ANGPOS(B,C,K)                                          CANCOR4
```

```
C                                                                       CANCOR4
C        SUBROUTINE ANGPOS HAS 3 ARGUMENTS                              CANCOR4
C              INPUT         B,K                                        CANCOR4
C                            B(1) = PITCH MEASUREMENT                   CANCOR4
C                            B(2) = YAW   MEASUREMENT                   CANCOR4
C                            B(3) = ROLL  MEASUREMENT                   CANCOR4
C                            K = TIME                                   CANCOR4
C              OUTPUT        C                                          CANCOR4
C                            C(1) = PITCH IN MILLI-RADIANS              CANCOR4
C                            C(2) = YAW   IN MILLI-RADIANS              CANCOR4
C                            C(3) = ROLL  IN MILLI-RADIANS              CANCOR4
C                                                                       CANCOR4
02       DIMENSION B(3),C(3)                                            CANCOR4
03       COMMON/COEFF/AA(4,3)                                           CANCOR4
04       COMMON/CANOP1/CLOCK/CANOP2/CONE(275)                           CANCOR4
C                                                                       CANCOR4
C        J IS TIME IN DAYS  FROM LANCH                                  CANCOR4
C        K IS TIME IN DAYS  FROM JAN 1                                  CANCOR4
C                            CLOCK   =X-AXIS CLOCK ANGLE IN DEGREES     CANCOR4
C                            CONE(K)=CANOPUS CONE ANGLE IN DEGREES      CANCOR4
C                                                                       CANCOR4
05       IF (K.GT.330) J=K-331                                          CANCOR4
06       IF (K.LT.330) J=K+34                                           CANCOR4
07       THETA1=CLOCK*.17453293E-01                                     CANCOR4
08       THETA2=CONE(J)*.17453293E-01                                   CANCOR4
C                                                                       CANCOR4
C        CALIBRATION OF ANGULAR POSITION FROM DATA NUMBER               CANCOR4
C                            P = PITCH MEASUREMENT IN MILLI-RADIANS     CANCOR4
C                            Y = YAW   MEASUREMENT IN MILLI-RADIANS     CANCOR4
C                            R = ROLL  MEASUREMENT IN MILLI-RADIANS     CANCOR4
C                                                                       CANCOR4
09       P=(AA(1,1)+AA(2,1)*B(1)+AA(3,1)*B(1)**2+AA(4,1)*B(1)**3)       CANCOR4
10       Y=(AA(1,2)+AA(2,2)*B(2)+AA(3,2)*B(2)**2+AA(4,2)*B(2)**3)       CANCOR4
11       R=(AA(1,3)+AA(2,3)*B(3)+AA(3,3)*B(3)**2+AA(4,3)*B(3)**3)       CANCOR4
12       C1=COS(THETA1)                                                 CANCOR4
13       C2=COS(THETA2)                                                 CANCOR4
14       S1=SIN(THETA1)                                                 CANCOR4
15       S2=SIN(THETA2)                                                 CANCOR4
16       C(1)=P                                                         CANCOR4
17       C(2)=Y                                                         CANCOR4
C                                                                       CANCOR4
C        THIS IS SMALL ANGLE APPROXIMATION - SECOND TERMS NEGLECTED     CANCOR4
C                                                                       CANCOR4
18       C(3)=(R-C2*(+P*C1+Y*S1))/S2                                    CANCOR4
19       RETURN                                                         CANCOR4
20       END                                                            CANCOR4
$IBFTC CANCOR
C                                                                       CANCOR7
C        THIS IS FOR MV-67                                              CANCOR7
C        YOU ALSO NEED SUBPROGRAM BLOCK DATA                            CANCOR7
C                                                                       CANCOR7
01       SUBROUTINE ANGPOS(B,C,K)                                       CANCOR7
C                                                                       CANCOR7
C        SUBROUTINE ANGPOS HAS 3 ARGUMENTS                              CANCOR7
C              INPUT         B,K                                        CANCOR7
C                            B(1) = PITCH MEASUREMENT                   CANCOR7
C                            B(2) = YAW   MEASUREMENT                   CANCOR7
C                            B(3) = ROLL  MEASUREMENT                   CANCOR7
C                            K = TIME                                   CANCOR7
C              OUTPUT        C                                          CANCOR7
C                            C(1) = PITCH IN MILLI-RADIANS              CANCOR7
```

```
C                  C(2) = YAW    IN MILLI-RADIANS                     CANCOR7
C                  C(3) = ROLL   IN MILLI-RADIANS                     CANCOR7
C                                                                     CANCOR7
      DIMENSION B(3),C(3)                                             CANCOR7
      COMMON/COEFF/A(6,3)                                             CANCOR7
      COMMON/CANOP1/CLOCK/CANOP2/CONE(135)                            CANCOR7
C                                                                     CANCOR7
C     J IS TIME IN DAYS  FROM LANCH                                   CANCOR7
C     K IS TIME IN DAYS  FROM JAN 1                                   CANCOR7
C                  CLOCK  =X-AXIS CLOCK ANGLE IN DEGREES              CANCOR7
C                  CONE(K)=CANOPUS CONE ANGLE IN DEGREES              CANCOR7
C                                                                     CANCOR7
      J=K-164                                                         CANCOR7
      THETA1=CLOCK*.17453293E-01                                      CANCOR7
      THETA2=CONE(J)*.17453293E-01                                    CANCOR7
C                                                                     CANCOR7
C     CALIBRATION OF ANGULAR POSITION FROM DATA NUMBER                CANCOR7
C                  P = PITCH MEASUREMENT IN MILLI-RADIANS             CANCOR7
C                  Y = YAW   MEASUREMENT IN MILLI-RADIANS             CANCOR7
C                  R = ROLL  MEASUREMENT IN MILLI-RADIANS             CANCOR7
C                                                                     CANCOR7
      P=(A(1,1)+A(2,1)*B(1)+A(3,1)*B(1)**2+A(4,1)*B(1)**3             CANCOR7
     1+A(5,1)*B(1)**4+A(6,1)*B(1)**5)                                 CANCOR7
      Y=(A(1,2)+A(2,2)*B(2)+A(3,2)*B(2)**2+A(4,2)*B(2)**3             CANCOR7
     1+A(5,2)*B(2)**4+A(6,2)*B(2)**5)                                 CANCOR7
      R=(A(1,3)+A(2,3)*B(3)+A(3,3)*B(3)**2+A(4,3)*B(3)**3             CANCOR7
     1+A(5,3)*B(3)**4+A(6,3)*B(3)**5)                                 CANCOR7
      C1=COS(THETA1)                                                  CANCOR7
      C2=COS(THETA2)                                                  CANCOR7
      S1=SIN(THETA1)                                                  CANCOR7
      S2=SIN(THETA2)                                                  CANCOR7
C                                                                     CANCOR7
C     THIS IS SMALL ANGLE APPROXIMATION - SECOND TERMS NEGLECTED      CANCOR7
C                                                                     CANCOR7
      C(1)=P                                                          CANCOR7
      C(2)=Y                                                          CANCOR7
      C(3)=(R-C2*(-P*C1+Y*S1))/S2                                     CANCOR7
      RETURN                                                          CANCOR7
      END                                                             CANCOR7
C
C     A PLOTTER ROUTINE IS ALSO NECESSARY
C     JPLT3 WAS USED WITH THIS PROGRAM
```