COMPUTER PROGRAM FOR POST-FLIGHT EVALUATION OF THE CONTROL SURFACE RESPONSE FOR AN ATTITUDE CONTROLLED MISSILE

R. N. KNAUBER

VOUGHT CORPORATION
P. O. Box 225907
Dallas, Texas 75265

NASA Contract NAS1-15000
November 1982

Because of its significant early commercial potential, this information, which has been developed under a U.S. Government Program, is being disseminated within the United States in advance of general publication. This information may be duplicated and used by the recipient with the express limitation that it not be published. Release of this information to other domestic parties by the recipient shall be made subject to the same conditions.

Foreign release may be made only with prior NASA approval and appropriate export licenses. This legend shall be marked on any reproduction of this information in whole or in part.

Review for general release November 30, 1985
COMPUTER PROGRAM FOR POST-FLIGHT EVALUATION OF THE CONTROL SURFACE RESPONSE FOR AN ATTITUDE CONTROLLED MISSILE

SUMMARY

A FORTRAN IV coded computer program is presented for post-flight analysis of a missile's control surface response. It includes preprocessing of digitized telemetry data for time lags, biases, non-linear calibration changes and filtering. Measurements include autopilot attitude rate and displacement gyro output and four control surface deflections. Simple first order lags are assumed for the pitch, yaw and roll axes of control. Each actuator is also assumed to be represented by a first order lag. Mixing of pitch, yaw and roll commands to four control surfaces is assumed. A pseudo-inverse technique is used to obtain the pitch, yaw and roll components from the four measured deflections.

This program has been used for over 10 years on the NASA/SCOUT launch vehicle for post-flight analysis and was helpful in detecting incipient actuator stall due to excessive hinge moments.

The program is currently set up for a CDC CYBER 175 computer system. It requires 34K words of memory and contains 675 cards. A sample problem presented herein including the optional plotting requires eleven (11) seconds of central processor time.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY</td>
<td>i</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF ILLUSTRATIONS</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF SYMBOLS</td>
<td>v</td>
</tr>
<tr>
<td>1.0 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2.0 METHODOLOGY</td>
<td>2</td>
</tr>
<tr>
<td>2.1 Assumptions</td>
<td>2</td>
</tr>
<tr>
<td>2.2 Equations</td>
<td>3</td>
</tr>
<tr>
<td>2.2.1 Control Surface Response</td>
<td>3</td>
</tr>
<tr>
<td>2.2.2 Pseudo-Inverse</td>
<td>4</td>
</tr>
<tr>
<td>2.2.3 Telemetered Data Adjustments</td>
<td>5</td>
</tr>
<tr>
<td>3.0 PROGRAM DESCRIPTION</td>
<td></td>
</tr>
<tr>
<td>3.1 General</td>
<td>8</td>
</tr>
<tr>
<td>3.2 Program Flow</td>
<td>8</td>
</tr>
<tr>
<td>3.3 Subroutine Description</td>
<td>9</td>
</tr>
<tr>
<td>3.4 Input Data Description</td>
<td>14</td>
</tr>
<tr>
<td>3.4.1 Title and Control Integers</td>
<td>14</td>
</tr>
<tr>
<td>3.4.2 Telemetry Adjustments</td>
<td>15</td>
</tr>
<tr>
<td>3.4.3 Control System Constants</td>
<td>16</td>
</tr>
<tr>
<td>3.4.4 Non-Linear Calibration Adjustment Tables</td>
<td>18</td>
</tr>
<tr>
<td>3.4.5 Plot Constants</td>
<td>18</td>
</tr>
<tr>
<td>3.4.6 Telemetered Data Time Histories</td>
<td>19</td>
</tr>
<tr>
<td>3.5 Output Data Description</td>
<td>19</td>
</tr>
<tr>
<td>3.5.1 Individual Control Surface Response</td>
<td>20</td>
</tr>
<tr>
<td>3.5.2 Pitch, Yaw and Roll Component Response</td>
<td>20</td>
</tr>
<tr>
<td>3.5.3 CALCOMP Plots</td>
<td>20</td>
</tr>
<tr>
<td>Appendix A - FORTRAN Program Listing</td>
<td>A-1</td>
</tr>
<tr>
<td>Figure No.</td>
<td>Descriptions</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>1</td>
<td>Control System Block Diagram</td>
</tr>
<tr>
<td>2</td>
<td>Program Subroutine Interaction Map</td>
</tr>
<tr>
<td>3</td>
<td>Flow Chart of FINRES</td>
</tr>
<tr>
<td>4</td>
<td>Sample Problem Input Data</td>
</tr>
<tr>
<td>5</td>
<td>Sample Problem Printed Output Data - Individual Control Surfaces</td>
</tr>
<tr>
<td>6</td>
<td>Sample Problem Printed Output Data - Pitch, Yaw, and Roll Components</td>
</tr>
<tr>
<td>7</td>
<td>Sample Problem CALCOMP Plots</td>
</tr>
</tbody>
</table>
**LIST OF SYMBOLS**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[A]$</td>
<td>transformation matrix for control surface commands</td>
<td></td>
</tr>
<tr>
<td>$a_{ij}$</td>
<td>elements of control surface transformation matrix $[A]$, 'i' denotes row, 'j' denotes column</td>
<td></td>
</tr>
<tr>
<td>$a_\theta, a_\psi, a_\phi$</td>
<td>pitch, yaw, and roll displacement cross-coupling of rate coefficient in telemetered data</td>
<td>sec</td>
</tr>
<tr>
<td>$\bar{b}$</td>
<td>state equation coefficient vector of input</td>
<td></td>
</tr>
<tr>
<td>$e$</td>
<td>control error signal</td>
<td>degree</td>
</tr>
<tr>
<td>$[P]$</td>
<td>state system matrix for channel response</td>
<td></td>
</tr>
<tr>
<td>$K_P$</td>
<td>autopilot attitude displacement gain</td>
<td>deg/deg</td>
</tr>
<tr>
<td>$K_R$</td>
<td>autopilot attitude rate gain</td>
<td>sec</td>
</tr>
<tr>
<td>$p$</td>
<td>roll rate gyro output</td>
<td>deg/sec</td>
</tr>
<tr>
<td>$q$</td>
<td>pitch rate gyro output</td>
<td>deg/sec</td>
</tr>
<tr>
<td>$r$</td>
<td>yaw rate gyro output</td>
<td>deg/sec</td>
</tr>
<tr>
<td>$s$</td>
<td>Laplacian operator</td>
<td></td>
</tr>
<tr>
<td>$T$</td>
<td>transfer function</td>
<td></td>
</tr>
<tr>
<td>$t$</td>
<td>time</td>
<td>sec</td>
</tr>
<tr>
<td>$u$</td>
<td>state equation input</td>
<td></td>
</tr>
<tr>
<td>$x$</td>
<td>state variables</td>
<td></td>
</tr>
<tr>
<td>$y$</td>
<td>general nomenclature for telemetered parameters</td>
<td></td>
</tr>
<tr>
<td>$\delta$</td>
<td>control surface deflection</td>
<td>degrees</td>
</tr>
<tr>
<td>$\theta_e$</td>
<td>pitch attitude displacement gyro output</td>
<td>degrees</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Butterworth filter parameter</td>
<td>sec</td>
</tr>
<tr>
<td>$\phi_e$</td>
<td>roll attitude displacement gyro output</td>
<td>degrees</td>
</tr>
</tbody>
</table>
LIST OF SYMBOLS (Cont.)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \psi_e )</td>
<td>yaw attitude displacement gyro output ( \ldots \ldots ) degrees</td>
</tr>
<tr>
<td>( \omega )</td>
<td>characteristic break frequency ( \ldots \ldots \ldots \ldots ) rad/sec</td>
</tr>
<tr>
<td>( \omega_{co} )</td>
<td>Butterworth cutoff frequency ( \ldots \ldots \ldots \ldots ) rad/sec</td>
</tr>
</tbody>
</table>

Prefix

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta )</td>
<td>incremental value</td>
</tr>
</tbody>
</table>

Subscripts

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{act} )</td>
<td>actuator of control surface</td>
</tr>
<tr>
<td>( \text{bias} )</td>
<td>telemetry bias value</td>
</tr>
<tr>
<td>( c )</td>
<td>commanded or calculated</td>
</tr>
<tr>
<td>( m )</td>
<td>measured value</td>
</tr>
<tr>
<td>( \delta )</td>
<td>control surface</td>
</tr>
<tr>
<td>( \theta )</td>
<td>pitch channel</td>
</tr>
<tr>
<td>( \phi )</td>
<td>roll channel</td>
</tr>
<tr>
<td>( \psi )</td>
<td>yaw channel</td>
</tr>
</tbody>
</table>

Special Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>dots above denote time derivative</td>
</tr>
<tr>
<td>-</td>
<td>dashes above denote a vector</td>
</tr>
<tr>
<td>[ ]</td>
<td>matrix</td>
</tr>
<tr>
<td>[ ](^T)</td>
<td>transpose of a matrix</td>
</tr>
<tr>
<td>[ ](^{-1})</td>
<td>inverse of a matrix</td>
</tr>
<tr>
<td>#</td>
<td>matrix pseudo-inverse</td>
</tr>
<tr>
<td>'</td>
<td>superscript or primes denote a modified or adjusted parameter</td>
</tr>
</tbody>
</table>
Post-flight analysis of a missile autopilot and control system should include comparison of measured parameters for consistency with preflight mathematical models. For a proportional control system telemetered data may include the autopilot gyro outputs and the measured control surface deflections. Preflight gains applied to the gyro outputs and the model should yield the control surface deflections. Differences between the reconstructed values and measured values may be indicative of anomalies or incipient failures. This report presents a computer program used for post-flight analysis of the NASA/SCOUT Launch Vehicle first stage proportional control system. It has been used to identify anomalous behavior.

The missile autopilot is assumed to contain three axes of information each of which includes a displacement and rate term such as, pitch, yaw and roll axes having angular displacement and rate included in the control law. A block diagram of the system is presented in Figure 1.

In post-flight data reduction and analysis there are sometimes small deviations in time for each parameter, differences between assumed linear calibration for data reduction and actual non-linear calibrations, cross-coupling between on-board telemetry channels, and, higher frequency data and noise. All of these can have a significant effect on the post-flight reconstruction process. Therefore, a large part of the methodology herein involves shifting, adjusting and smoothing of the digitized reduced telemetry data.

The assumptions, methodology, program description and running instructions are presented in the following sections.
2.0 METHODOLOGY

This section contains the methodology and equations which are used to adjust the telemetry data and to reconstruct the control surface response from the measured data. Telemetry data for the autopilot pitch, yaw, and roll channel attitude rates and displacements and four control surface deflections are required. Estimates or preflight measurements of autopilot gains are also required. A proportional control system represented by the block diagram of Figure 1 is assumed. The assumptions and equations are presented in the following paragraphs.

2.1 Assumptions

Major assumptions and approximations are:

- autopilot and control system as presented in Figure 1,
- control gains are constant in time,
- pitch, yaw, and roll channels frequency response can be represented as a first order lag (single break frequency),
- actuator response for each of the four control surfaces can be represented by a first order lag,
- mixing of the pitch, yaw, and roll error signals for commands to the four control surfaces is represented by a constant matrix transformation,
- frequency response of the telemetry data is greater than the autopilot model
- phase shifts in the telemetry data can be represented by an equivalent time shift for each of the parameters,
- telemetry system cross-coupling is limited to the attitude displacement gyro output and is proportional to vehicle rate about that axis
- non-linearities in calibration of the control surfaces are not time dependent
2.2 Equations

2.2.1 Control Surface Response

The control surface response is represented by the block diagram of Figure 1. The pitch, yaw, and roll channel error signals are,

\begin{align*}
(2.1) \quad & e_\theta = \left( K_{D\theta} \theta_e + K_{R\theta} \varphi \right) \left( \frac{\omega_\theta}{s + \omega_\theta} \right) \\
(2.2) \quad & e_\psi = \left( K_{D\psi} \psi_e + K_{R\psi} \varphi \right) \left( \frac{\omega_\psi}{s + \omega_\psi} \right) \\
(2.3) \quad & e_\phi = \left( K_{D\phi} \phi_e + K_{R\phi} p \right) \left( \frac{\omega_\phi}{s + \omega_\phi} \right)
\end{align*}

These error signals are processed through a linear transformation to mix the signals to each of four control surfaces. This transformation is,

\begin{align*}
(2.4) \quad & \delta_c = [A] \bar{e}
\end{align*}

where, \( \delta_c \) is the four element vector,

\begin{align*}
(2.5) \quad & \delta_c = \begin{bmatrix} \delta_{c1} \\ \delta_{c2} \\ \delta_{c3} \\ \delta_{c4} \end{bmatrix}
\end{align*}

\([A]\) is the 4 by 3 transformation matrix specified by the autopilot design

\begin{align*}
(2.6) \quad & [A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix}
\end{align*}

\( \bar{e} \) is the three element error vector,

\begin{align*}
\bar{e} = \begin{bmatrix} e_\theta \\ e_\psi \\ e_\phi \end{bmatrix}
\end{align*}

According to Figure 1 each actuator responds to commands as a simple first order lag, i.e.,

\begin{align*}
(2.7) \quad & \delta = \delta_c \left\{ \frac{\omega_{act}}{s + \omega_{act}} \right\}
\end{align*}
Up to this point the system is linear. It is assumed that each actuator responds with the same break frequency. Therefore, the transfer function of equation (2.7) can be moved and included in the transfer function of equations (2.1), (2.2), and (2.3). This is done in the program to reduce computer time. The commanded pitch, yaw, and roll surface deflections become,

\[
\delta_c = \begin{bmatrix} A \end{bmatrix} \begin{bmatrix} \omega_{\text{act}} \\ s + \omega_{\text{act}} \end{bmatrix}
\]

This will be followed through the derivation for the pitch channel; yaw and roll channel equations are similar. Prior to the mixing of channel error signals via the \([A]\) transformation, the equivalent pitch component of commanded surface deflections is,

\[
\delta_{c\theta} = \left( K_d \theta_e + K_R q \right) \begin{bmatrix} \omega_{\theta} \\ s + \omega_{\theta} \end{bmatrix} \begin{bmatrix} \omega_{\text{act}} \\ s + \omega_{\text{act}} \end{bmatrix}
\]

This can be put into the state variable form,

\[
\dot{x} = \begin{bmatrix} A \end{bmatrix} \begin{bmatrix} \omega_{\theta} \\ s + \omega_{\theta} \end{bmatrix} \begin{bmatrix} \omega_{\text{act}} \\ s + \omega_{\text{act}} \end{bmatrix}
\]

where the input,

\[
u = \begin{bmatrix} K_d \theta_e + K_R q \end{bmatrix}
\]

\[
x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}
\]

(two state vector)

\[
\begin{bmatrix} F \end{bmatrix} = \begin{bmatrix} -\omega_{\theta} & 0 \\ \omega_{\text{act}} & -\omega_{\text{act}} \end{bmatrix}
\]

(system matrix)

\[
b = \begin{bmatrix} \omega_{\theta} \\ 0 \end{bmatrix}
\]

(input coefficient vector)

and the output is,

\[
\delta_{c\theta} = x_2
\]

In the program these equations are solved by a linear system time response subroutine (TRESP) which uses a fourth-order RUNGE-KUTTA integration method. After the pitch, yaw, and roll commanded deflections are computed they are transformed by the matrix \([A]\) (Equation 2.6) to provide the calculated or reconstructed deflections of each of the four control surfaces.

2.2.2 Pseudo-Inverse

Commanded individual control surfaces represented by equation 2.4 involves a linear transformation \([A]\) of a 'three-vector' into a 'four-vector'. The
pitch, yaw, and roll components of the four control surfaces can be calculated from the measured individual control surface deflections. This is easily done using the pseudo-inverse transformation of \([A]\). A matrix inverse does not exist for non-square matrices. The pseudo-inverse provides a reverse transformation in a least-squares sense. The pseudo-inverse is,

\[
(2.16) \quad [A]# = ([A]^T[A])^{-1} [A]^T
\]

for the overdetermined case where \([A]\) has more rows than columns.


Actual pitch, yaw, and roll control surface deflection components can be computed,

\[
(2.17) \begin{bmatrix}
\delta_{\text{act}\theta} \\
\delta_{\text{act}\psi} \\
\delta_{\text{act}\phi}
\end{bmatrix} = [A]^# \begin{bmatrix}
\delta_{\text{act}1} \\
\delta_{\text{act}2} \\
\delta_{\text{act}3} \\
\delta_{\text{act}4}
\end{bmatrix}
\]

2.2.3 Telemetered Data Adjustments

Reduced telemetry data for the pitch, yaw, and roll displacements and rates and the four control surface deflections may require further adjustments and filtering. These adjustments include biases, time shifts, non-linearities, cross-coupling, and additional filtering.

BIASES

Each parameter is assumed to have a simple bias error which can be estimated from a quiet period of flight such as prior to vehicle liftoff. These are entered in the input to the routine and added to the reduced telemetry data, i.e.,

\[
(2.18) \quad y'_m(t) = y_m(t) + \Delta y
\]

where, \(y_m(t)\) is the reduced measured telemetry parameter
\(y'_m(t)\) is the adjusted parameter
\(\Delta y\) is the bias shift required

TIME SHIFTS

Each measured parameter is assumed to have a different time delay due to the telemetry system and ground station playbacks. In order to be consistent each parameter must be shifted to a common time base. This is done by a table lookup versus time such that the parameter \(y\) is shifted back in time by its peculiar lag, i.e.,

\[
(2.19) \quad y''_m(t) = y'_m(t+\Delta t)
\]

where, \(y''_m(t)\) is the measured parameter adjusted for time and bias shifts
\(\Delta t\) is the required time shift for parameter 'y'
NON-LINEARITIES

The four control surface deflections may have a parabolic calibration curve (this is the case for a measurement using a linear potentiometer attached to a bellcrank). In data reduction the calibration applied is usually a straight line or a series of straight lines (see the sketch below).

This can lead to subtle errors in reconstructing the control surface deflections. Therefore, an adjustment is made to the data via a table lookup in the computer program. This adjustment is,

\[ (2.20) \Delta y = \Delta y_{bias} + \Delta y(\delta) \]

where, \( \Delta y_{bias} \) is an average bias value

\( \Delta y(\delta) \) is a function of the deflection based on the sketches above

The bias shown in Equation (2.20) is inserted into Equation (2.18).

CROSS-COUPLING

Peculiarities in an attitude gyro telemetry pickoff may induce a voltage proportional to the vehicle rate about the input axis (this is peculiar to the SCOUT vehicle using miniature integrating rate gyros to measure attitude displacements). The true attitude displacement is therefore,

\[ (2.21) \theta_e = \theta_{em} - a_\theta q \]
\[ \psi_e = \psi_{em} - a_\psi r \]
\[ \phi_e = \phi_{em} - a_\phi p \]

where the coefficients \( a_\theta, a_\psi, \) and \( a_\phi \) are determined by preflight test of the telemetry and autopilot subsystems.

ADDITIONAL FILTERING

Since the telemetry data may have noise or data at frequencies well above the bandwidth of interest, additional filtering capability is afforded by the computer program. A third order Butterworth filter is supplied. The cutoff frequency is an input variable. Each parameter is filtered with the same algorithm and cutoff frequency. The transfer function is,

\[ (2.22) T(s) = \frac{1}{1 + 2 rs + 2 r^2 s^2 + r^3 s^3} \]
where,

\[(2.23) \quad r = \frac{0.707}{\omega_c} \]

and,

\[ \omega_c \text{ is the cutoff frequency in radians per second} \]

The equivalent state-space filter equation for this transfer function is,

\[(2.24) \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -2/r & -2/r^2 & -1/r^3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} u \]

where, \( x_1, x_2, \) and \( x_3 \) are the filter states.

\( u \) is the filter input

\( x_3 \) state is the filter output

These equations are solved in the computer program using the time response subroutine (TRESP).

Application of this filter adds a low frequency time delay to the data. This delay is,

\[(2.25) \quad t = \frac{1.418}{\omega_c} \text{ (seconds)} \]

This time increment is added to the individual telemetry parameter time delays before shifting the data with Equation (2.19).
3.0 PROGRAM DESCRIPTION

3.1 General

This computer program is programmed in FORTRAN IV for a CDC CYBER 175 system. The coding is in the most part compatible with ANSI standards. Non-ANSI statements include the PROGRAM card and the use of EOF (end-of-file) test for transfer from reading input. Another area of limited portability is the use of ten letter words for labeling information. Consult your computer department for changes in these areas.

The computer routine is arranged to operate with standard card input and line printer output. Optional plotting is based on standard CALCOMP plotters and software. The CALTEKA Tektronix terminal plot can also be used without program modification.

A main routine (FINRES) and twelve (12) subroutines require approximately 34K words of computer memory. Input and output is stored in arrays to facilitate time adjustments, filtering, and a well formatted printed and plotted output.

Program flow and user instructions are presented in the following paragraphs. Input and output of a sample problem is illustrated along with detailed descriptions.

3.2 Program Flow

Program flow is straightforward in eight basic parts.

- input data
- telemetry data adjust for calibration and filtering
- reconstruction of pitch, yaw and roll commands
- comparison of individual control surfaces with reconstructed commands
- output of individual surface data
- optional reconstruction of pitch, yaw and roll components of control surface response and output
- plotted output

The interdependence of the main routine and the subroutines is presented in Figure 2. A flow chart of the main routine (FINRES) is presented in Figure 3. A complete listing of the FORTRAN program and subroutines other than the standard CALCOMP library subroutines are presented in Appendix A.

Descriptions of the twelve subroutines are presented in the following paragraphs.
3.3 Subroutine Description

Twelve subroutines are used to support the FINRES main program; CURVE, DASH, ERSIG, FILFIL, FIN, PSEUDO, RUNGE, SIMEQ, TBLN, TRESP, XMULT, and YDOT. A brief description of each is presented below.

**CURVE**

This subroutine sets up the calcomp plot for one frame of a single control surface deflection comparison. It includes the graph paper description (CAL22) which has a 10 by 16 grid size with 20 divisions per inch. Paper performance size is 11 by 17. All data to be plotted by this routine enters in the argument list. The actual curve plotting is made through calls to the DASH subroutine. The call statement for CURVE is,

```
CALL CURVE (T, CALC, ACT, NP, NTIT, NM, XS, YS, DS)
```

where,

- **T** - input array name of time abscissas
- **CALC** - input array name of computed commanded surface deflections (reconstructed)
- **ACT** - input array name of measured control surface deflection
- **NP** - is the number of time points in T, CALC, and ACT arrays to be plotted
- **NTIT** - input array of eight (8) ten-letter words contained 80 character title
- **NM** - input ten letter word variable to identify frame (this is output on second line of title)
- **XS** - abscissa (time) scale factor (units/inch)
- **YS** - ordinate deflection scale factor (units/inch)
- **DS** - ordinate scale factor for difference (CALC-ACT) deflection (units/inch)

Care must be taken in selecting scale factors so that plotted data falls on grid. Limiting of plotted data is automatically invoked in CURVE through the call statements to DASH.

**DASH**

This subroutine plots a curve on a CALCOMP plotter for a set of ordinates and abscissas. The style and type of line drawn is selected by the user. Note that the CALCOMP plot is specified in inches; plotting on metric paper requires appropriate scaling change before entering this subroutine.

The call statement is,

```
CALL DASH (X, Y, NP, Z1, Z2, SPACE, XSCALE, YSCALE, LSYMB, XLIM, YLIM)
```

where,

- **X** - input array of abscissa values
- **Y** - input array of ordinate values
- **NP** - number of points in X and Y to be plotted
for dashed-dot lines this is length of long line measured in inches (see sketch below)

for dashed-dot lines this is length of short line measured in inches (see sketch)

for dashed style lines this is the length of the space between lines measured in inches.

SPACE = 0 gives a solid line plot

SPACE = negative gives special CALCOMP symbols at each point

abscissa plot scale factor (units per inch)

ordinate plot scale factor (units per inch)

special CALCOMP symbol code number used if SPACE is negative (see code below)

(+ ) LSYMB gives straight solid lines between symbol points

(- ) LSYMB gives only symbols at each point without lines

plot limiting of the abscissa (inches) points out of range, range will appear at this limit

plot range of ordinate (inches)

For ease in use, the following styles are typically possible,

<table>
<thead>
<tr>
<th>LINE</th>
<th>TYPE</th>
<th>Z1</th>
<th>Z2</th>
<th>SPACE</th>
<th>LSYMB</th>
</tr>
</thead>
<tbody>
<tr>
<td>__</td>
<td>Solid</td>
<td>--</td>
<td>--</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>---</td>
<td>Dashed</td>
<td>0.25</td>
<td>0.25</td>
<td>0.10</td>
<td>0.00</td>
</tr>
<tr>
<td>----</td>
<td>Dashed</td>
<td>0.07</td>
<td>0.07</td>
<td>0.07</td>
<td>0.00</td>
</tr>
<tr>
<td>--</td>
<td>Dashed</td>
<td>0.50</td>
<td>0.03</td>
<td>0.07</td>
<td>0.00</td>
</tr>
<tr>
<td>△△△</td>
<td>Dot</td>
<td>--</td>
<td>--</td>
<td>-0.1</td>
<td>+2</td>
</tr>
<tr>
<td>△△△</td>
<td>Symbols</td>
<td>--</td>
<td>--</td>
<td>-0.1</td>
<td>-2</td>
</tr>
</tbody>
</table>

ERSIG

This subroutine computes the single axis error signal commanded deflection (pitch, yaw, or roll). It includes adjustment for rate crosscoupling into the displacement telemetry data, the channel break frequency and actuator break frequency time response. The call statement is,

CALL ERSIG (FILTER, B, T, Q, TH, NP, NINT, CTH, CTHD, DT, DTM, DTMR, AD, W, WACT)

where,

FILTER input third order coefficient array for the Butterworth filter states (see Paragraph 2.2.3)
B telemetry filtering input coefficient vector (Paragraph 2.2.3)
T input array of time points for telemetry data containing NP points
Q input array of NP points of telemetered rate data corresponding to (T) times. It is changed to output the filtered error signal.
input array of NP points of telemetered displacement data corresponding to (T) times

number of time points in arrays T, Q, and TH

number of integration steps per value of time in (T) array to be used in filtering and response histories (e.g., if NINT = 2) the integration step size is one-half of the time between points in (T) array

attitude displacement gain \( K_D \)
rate gain \( K_R \)
time delay of third order filter (FILTER) and (B) to be used in time corrections
time delay of attitude error telemetry data
time delay of rate telemetry data
telemetry cross-coupling coefficient of attitude displacement due to rate

break frequency of the control channel

actuator break frequency

Note that the array (Q) is destroyed by the subroutine and used to return the computed filtered error signal.

**FILFIL**

This subroutine fills the third order Butterworth filter coefficient arrays and computes the low frequency time lag of this filter as presented in Paragraph 2.2.3. The call statement is,

```
CALL FILFIL (WCO, A, B, DT)
```

where,

WCO

input cutoff frequency of the Butterworth filter (radians per second)

A

output coefficient matrix of the filter (3 by 3)

B

output coefficient vector for filter state inputs

DT

output effective low frequency time lag associated with the filter (seconds)

**FIN**

This subroutine filters a telemetered control surface deflection time history and adjusts it for time shifts. The call statement is,

```
CALL FIN (FILTER, B, T, D, NP, DTFIL, TC, NINT)
```

where,

FILTER

input (3 by 3) matrix of third order filter coefficient matrix

B

input (3 by 1) coefficient input vector for third order filter

T

input array of NP times for deflection array D

D

input array of NP control surface deflections (also the output filtered time adjusted deflection)

NP

number of time points in T, and D arrays

DTFIL

input low frequency time lag of filter in seconds

TC

input time lag of telemetered control surface deflection for use in adjustment
NINT  number of integration steps to be used between time points in \((T)\) array

**PSEUDO**

This subroutine computes the pseudo-inverse of the coefficient matrix as described in Paragraph 2.2.2. The call statement is,

\[
\text{CALL PSEUDO (B, A, N, M, NER)}
\]

where,

- **B** output \(M\) by \(N\) pseudo-inverse matrix of \(A\)
- **A** input matrix having dimensions \(N\) by \(M\)
- **N** is number of rows of \((A)\) and number of columns of \((B)\)
- **M** is number of columns of \((A)\) and number of rows of \((B)\)
- **NER** is an error indicator
  - \(NER = 1\) normal execution
  - \(NER = 0\) abnormal condition because of a submatrix singularity (pseudo inverse cannot be computed)

**RUNGE**

This subroutine aids in the RUNGE-KUTTA integration of the time response \((TRESP)\) subroutine. The call statement is,

\[
\text{CALL RUNGE (N, FN, H, X, Y, L, I)}
\]

where,

- **N** input system order
- **FN** first derivatives of the state
- **H** integration step size
- **X** time variable
- **Y** state vector
- **L** control integer
  - \(L = 1\) indicates incomplete integration process
  - \(L = 2\) indicates completed integration step
- **I** number of times that RUNGE has been entered on the current integration step (when \(I = 5\) the final answer is computed)

**SIMEQ**

This subroutine is used to compute the inverse of a square matrix by diagonalization. With modification it can be used to solve a set of simultaneous linear equations. The call statement is,

\[
\text{CALL SIMEQ (A, KC, AINV, IERR)}
\]

where,

- **A** input \(KC\) by \(KC\) matrix to be inverted
- **KC** input order of the matrix
- **AINV** output inverted matrix if computed
- **IERR** output error variable
  - \(IERR = 1\) normal computation
  - \(IERR = 0\) abnormal (A matrix is singular)
TBLN

This is a single table lookup subroutine using linear interpolation between points. This subroutine requires separate consistent arrays of abscissas and ordinates. The abscissas must be in ascending order. The call statement is,

CALL TBLN (Y, X, T, A, NT, M)

where,

Y     output ordinate to be found
X     input abscissa value
T     array of abscissas
A     array of ordinates
NT    number of values in (T) and (A) arrays
M     input index 1 to NT to begin the table search. After locating the ordinate the nearest location is returned for future use.

TRESP

This subroutine used in conjunction with subroutines RUNGE and YDOT solve a time response for a set of up to three first order linear differential equations by a fourth order RUNGE-KUTTA integration procedure. The call statement is,

CALL TRESP (A, B, T, Y, Z, NP, N, K, NINT)

where,

A     input coefficient matrix
B     input coefficient vector for state equations
T     input array of (NP) forcing function time values
Y     input array of NP forcing function values corresponding to times in (T) array
Z     output array of 'Kth' state variable values corresponding to time in (T) array
NP    input number of values in T, Y, and Z array
N     input order of the system
K     input designation of state variable corresponding to the desired output
NINT  input number of integration steps between time points in (T) array

XMULT

This subroutine multiplies two matrices. The call statement is,

CALL XMULT (A, B, C, N)

where,

A     input premultiplier matrix
B     input postmultiplier matrix
This subroutine computes the first derivative of the system state vector used by TRESP to integrate a set of first order linear differential equations. The call statement is,

```
CALL YDOT (A, Y, XDOT, B, U, N)
```

where,

- **A**: input system coefficient matrix of order N
- **Y**: input state vector
- **XDOT**: input derivative of the state vector and returned updated derivative of the state vector
- **B**: input coefficient vector of the forcing function
- **U**: input value of the forcing function
- **N**: input order of the system

### 3.4 Input Data Description

Input data descriptions are presented in the following subparagraphs. A sample problem input data listing is presented in Figure 4 for reference. Input data can be separated into the following categories:

1. title card and control options
2. telemetry adjustments
3. control system constants
4. control surface deflection adjustment tables for non-linearities
5. plot scale factors
6. time histories of measured telemetry variables

#### 3.4.1 Title and Control Constants

The first card of input contains 80 columns of arbitrary title information. This information is printed at the head of each page of output and at the top of each frame of plotted data. It is input using an array (NTIT) having eight words containing ten characters each.

The second card contains four (4) integer constants right justified without a decimal point. It is input with a format of (4I5). The description is,
3.4.2 Telemetry Adjustments

This group contains twenty-four (24) constants used for adjustment of the telemetry data. These are input eight numbers per card in fields of ten columns (format **8E10.3**). The descriptions are,

<table>
<thead>
<tr>
<th>FORTRAN NAME</th>
<th>CARD</th>
<th>COLUMNS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DL(1)</td>
<td>3</td>
<td>1-10</td>
<td>bias adjustment to be added to pitch rate</td>
</tr>
<tr>
<td>DL(2)</td>
<td>3</td>
<td>11-20</td>
<td>bias adjustment to be added to yaw rate</td>
</tr>
<tr>
<td>DL(3)</td>
<td>3</td>
<td>21-30</td>
<td>bias adjustment to be added to roll rate</td>
</tr>
<tr>
<td>DL(4)</td>
<td>3</td>
<td>31-40</td>
<td>bias adjustment to be added to pitch displacement</td>
</tr>
<tr>
<td>DL(5)</td>
<td>3</td>
<td>41-50</td>
<td>bias adjustment to be added to yaw displacement</td>
</tr>
<tr>
<td>DL(6)</td>
<td>3</td>
<td>51-60</td>
<td>bias adjustment to be added to roll displacement</td>
</tr>
<tr>
<td>DL(7)</td>
<td>3</td>
<td>61-70</td>
<td>bias adjustment to be added to fin 1 control surface</td>
</tr>
<tr>
<td>FORTRAN NAME</td>
<td>CARD</td>
<td>COLUMNS</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>DL(8)</td>
<td>3</td>
<td>71-80</td>
<td>bias adjustment to be added to fin 2 control surface</td>
</tr>
<tr>
<td>DL(9)</td>
<td>4</td>
<td>1-10</td>
<td>bias adjustment to be added to fin 3 control surface</td>
</tr>
<tr>
<td>DL(10)</td>
<td>4</td>
<td>11-20</td>
<td>bias adjustment to be added to fin 4 control surface</td>
</tr>
<tr>
<td>TCQ</td>
<td>4</td>
<td>21-30</td>
<td>time lag in pitch rate data</td>
</tr>
<tr>
<td>TCR</td>
<td>4</td>
<td>31-40</td>
<td>time lag in yaw rate data</td>
</tr>
<tr>
<td>TCP</td>
<td>4</td>
<td>41-50</td>
<td>time lag in roll rate data</td>
</tr>
<tr>
<td>TCTH</td>
<td>4</td>
<td>51-60</td>
<td>time lag in pitch displacement data</td>
</tr>
<tr>
<td>TCPH</td>
<td>4</td>
<td>61-70</td>
<td>time lag in yaw displacement data</td>
</tr>
<tr>
<td>TC1</td>
<td>5</td>
<td>1-10</td>
<td>time lag in fin 1 control surface data</td>
</tr>
<tr>
<td>TC2</td>
<td>5</td>
<td>11-20</td>
<td>time lag in fin 2 control surface data</td>
</tr>
<tr>
<td>TC3</td>
<td>5</td>
<td>21-30</td>
<td>time lag in fin 3 control surface data</td>
</tr>
<tr>
<td>TC4</td>
<td>5</td>
<td>31-40</td>
<td>time lag in fin 4 control surface data</td>
</tr>
<tr>
<td>AKTH</td>
<td>5</td>
<td>41-50</td>
<td>pitch displacement to rate telemetry cross coupling coefficient</td>
</tr>
<tr>
<td>AKPS</td>
<td>5</td>
<td>51-60</td>
<td>yaw displacement to rate telemetry cross coupling coefficient</td>
</tr>
<tr>
<td>AKPH</td>
<td>5</td>
<td>61-70</td>
<td>roll displacement to rate telemetry cross coupling coefficient</td>
</tr>
<tr>
<td>WCO</td>
<td>5</td>
<td>71-80</td>
<td>cutoff frequency of third order Butterworth filter to be used on all input time histories (hertz)</td>
</tr>
</tbody>
</table>

3.4.3 Control System Constants

This group of input data includes the autopilot control system gains, break frequencies and error signal mixing matrix for the individual control surface commands. Gains and break frequencies are input in fields of ten with format (6E10.3). They are,
<table>
<thead>
<tr>
<th>FORTRAN NAME</th>
<th>CARD</th>
<th>COLUMN</th>
<th>SYMBOL</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTH</td>
<td>6</td>
<td>1-10</td>
<td>K_Dθ</td>
<td></td>
<td>pitch attitude displacement gain</td>
</tr>
<tr>
<td>CTHD</td>
<td>6</td>
<td>11-20</td>
<td>K_Rθ</td>
<td>sec</td>
<td>pitch rate gain</td>
</tr>
<tr>
<td>CPS</td>
<td>6</td>
<td>21-30</td>
<td>K_Dψ</td>
<td></td>
<td>yaw attitude displacement gain</td>
</tr>
<tr>
<td>CPSD</td>
<td>6</td>
<td>31-40</td>
<td>K_Rψ</td>
<td>sec</td>
<td>yaw rate gain</td>
</tr>
<tr>
<td>CPH</td>
<td>6</td>
<td>41-50</td>
<td>K_Dφ</td>
<td></td>
<td>roll attitude displacement gain</td>
</tr>
<tr>
<td>CPHD</td>
<td>6</td>
<td>51-60</td>
<td>K_Rφ</td>
<td>sec</td>
<td>roll rate gain</td>
</tr>
<tr>
<td>W1Q</td>
<td>6</td>
<td>61-70</td>
<td>ωθ</td>
<td>rad/sec</td>
<td>pitch channel break frequency</td>
</tr>
<tr>
<td>W1R</td>
<td>6</td>
<td>71-80</td>
<td>ωψ</td>
<td>rad/sec</td>
<td>yaw channel break frequency</td>
</tr>
<tr>
<td>W1P</td>
<td>7</td>
<td>1-10</td>
<td>ωφ</td>
<td>rad/sec</td>
<td>roll channel break frequency</td>
</tr>
<tr>
<td>WACT</td>
<td>7</td>
<td>11-20</td>
<td>ω_act</td>
<td>rad/sec</td>
<td>control surface actuator break frequency</td>
</tr>
</tbody>
</table>

The next four cards contain the transfer matrix describing the relationship of the four control surfaces to the pitch, yaw and roll error signals. These are input with one card for each surface with three constants relating the components of pitch, yaw and roll error signal to each surface. Input is with format (4E10.3).

<table>
<thead>
<tr>
<th>FORTRAN NAME</th>
<th>CARD</th>
<th>COLUMNS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A(1,1)</td>
<td>8</td>
<td>1-10</td>
<td>fraction of pitch error for surface no. 1</td>
</tr>
<tr>
<td>A(1,2)</td>
<td>8</td>
<td>11-20</td>
<td>fraction of yaw error for surface no. 1</td>
</tr>
<tr>
<td>A(2,1)</td>
<td>9</td>
<td>1-10</td>
<td>fraction of roll error for surface no. 1</td>
</tr>
<tr>
<td>A(2,2)</td>
<td>9</td>
<td>11-20</td>
<td>fraction of pitch error for surface no. 2</td>
</tr>
<tr>
<td>A(2,3)</td>
<td>9</td>
<td>21-30</td>
<td>fraction of yaw error for surface no. 2</td>
</tr>
<tr>
<td>A(3,1)</td>
<td>10</td>
<td>1-10</td>
<td>fraction of pitch error for surface no. 3</td>
</tr>
<tr>
<td>A(3,2)</td>
<td>10</td>
<td>11-20</td>
<td>fraction of yaw error for surface no. 3</td>
</tr>
<tr>
<td>A(3,3)</td>
<td>10</td>
<td>21-30</td>
<td>fraction of roll error for surface no. 3</td>
</tr>
<tr>
<td>A(4,1)</td>
<td>11</td>
<td>1-10</td>
<td>fraction of pitch error for surface no. 4</td>
</tr>
<tr>
<td>A(4,2)</td>
<td>11</td>
<td>11-20</td>
<td>fraction of yaw error for surface no. 4</td>
</tr>
<tr>
<td>A(4,3)</td>
<td>11</td>
<td>21-30</td>
<td>fraction of roll error for surface no. 4</td>
</tr>
</tbody>
</table>
### 3.4.4 Control Surface Non-Linear Calibration Adjustment Tables

This group of input includes a table of adjustments for each control surface which allows for non-linearities in calibration not included in the data reduction process. These are read with format \((I5/,8E10.3)\). The first card of each table contains the number of abscissa-ordinate pairs. The abscissa (control surface deflection) must be in ascending order. Four tables, one for each fin control surface includes,

<table>
<thead>
<tr>
<th>FORTRAN NAME</th>
<th>COLUMN</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NT1</td>
<td>1-5</td>
<td>number of pairs of abscissas and ordinates in surface no. 1 adjustment table</td>
</tr>
<tr>
<td>CD1(I), ED1(I),</td>
<td>1-10, 11-20, etc</td>
<td>CD1 contains abscissa value of surface no. 1 deflection ED1 contains ordinate value of surface no. 1 to be added to input telemetry data</td>
</tr>
<tr>
<td>NT1</td>
<td>1-5</td>
<td>(similar to above description for control surface no. 2)</td>
</tr>
<tr>
<td>CD2(I), ED2(I),</td>
<td>1-10, 11-20, etc</td>
<td>(similar to above description for control surface no. 3)</td>
</tr>
<tr>
<td>NT4</td>
<td>1-5</td>
<td>(similar to above description for control surface no. 4)</td>
</tr>
</tbody>
</table>

### 3.4.5 Plot Constants

This includes a single card of scale factors for the CALCOMP type plots. This card is input only when \((IOP = 1)\) on the second card of input (Paragraph 3.4.1). Input uses format \((8E10.3)\). The descriptions are,

<table>
<thead>
<tr>
<th>FORTRAN NAME</th>
<th>COLUMNS</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSCALE</td>
<td>1-10</td>
<td>sec/inch</td>
<td>time abscissa scale factor per inch of paper</td>
</tr>
<tr>
<td>YSCALE</td>
<td>11-20</td>
<td>deg/inch</td>
<td>control surface ordinate scale factor per inch of plot paper</td>
</tr>
<tr>
<td>DSSCALE</td>
<td>21-30</td>
<td>deg/inch</td>
<td>control surface deflection difference scale factor per inch of paper</td>
</tr>
</tbody>
</table>
3.4.6 Telemetered Data Time Histories

This group of data is entered with format (11F7.3); each card contains all of the telemetered parameters for a given time point. The time points must be equally spaced. The analysis uses all the input data time points and stops reading input when an 'End-of-File' card is read in the input data stream. The description of this data follows (refer to the sample problem input of Figure 4).

<table>
<thead>
<tr>
<th>FORTRAN NAME</th>
<th>COLUMN</th>
<th>SYMBOL</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>T(J)</td>
<td>1-7</td>
<td>t</td>
<td>seconds</td>
<td>time</td>
</tr>
<tr>
<td>Q(J)</td>
<td>8-14</td>
<td>p</td>
<td>deg/sec</td>
<td>telemetered pitch rate</td>
</tr>
<tr>
<td>R(J)</td>
<td>15-21</td>
<td>q</td>
<td>deg/sec</td>
<td>telemetered yaw rate</td>
</tr>
<tr>
<td>P(J)</td>
<td>22-28</td>
<td>r</td>
<td>deg/sec</td>
<td>telemetered roll rate</td>
</tr>
<tr>
<td>TH(J)</td>
<td>29-35</td>
<td>\theta_e</td>
<td>degrees</td>
<td>telemetered pitch displacement error</td>
</tr>
<tr>
<td>PS(J)</td>
<td>36-42</td>
<td>\psi_e</td>
<td>degrees</td>
<td>telemetered yaw displacement error</td>
</tr>
<tr>
<td>PH(J)</td>
<td>43-49</td>
<td>\phi_e</td>
<td>degrees</td>
<td>telemetered roll displacement error</td>
</tr>
<tr>
<td>D1(J)</td>
<td>50-56</td>
<td>\delta_1</td>
<td>degrees</td>
<td>telemetered fin 1 control surface deflection</td>
</tr>
<tr>
<td>D2(J)</td>
<td>57-63</td>
<td>\delta_2</td>
<td>degrees</td>
<td>telemetered fin 2 control surface deflection</td>
</tr>
<tr>
<td>D3(J)</td>
<td>64-70</td>
<td>\delta_3</td>
<td>degrees</td>
<td>telemetered fin 3 control surface deflection</td>
</tr>
<tr>
<td>D4(J)</td>
<td>71-77</td>
<td>\delta_4</td>
<td>degrees</td>
<td>telemetered fin 4 control surface deflection</td>
</tr>
</tbody>
</table>

3.5 Output Data Description

Output includes printed data and optional CALCOMP type plots (if IOP = 1). A detailed description of the output is presented in the following paragraphs with a sample problem for reference. The printed output includes three parts depending on option.

1) individual control surface response
2) pitch yaw and roll control surface response (if IOC = 1)
3) CALCOMP plots (if IOP = 1)
3.5.1 Individual Control Surface Response

Printed output for a sample problem is presented in Figure 5. The first part of the output includes time histories of these parameters for each of the four control surfaces. These parameters are,

<table>
<thead>
<tr>
<th>OUTPUT LABEL</th>
<th>SYMBOLS</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND</td>
<td>$\delta_c$</td>
<td>deg</td>
<td>reconstructed control surface deflection based on the telemetered autopilot data</td>
</tr>
<tr>
<td>ACTUAL</td>
<td>$\delta_{act}$</td>
<td>deg</td>
<td>filtered smoothed and adjusted telemetry data for the control surface deflection</td>
</tr>
<tr>
<td>DELTA</td>
<td>$\delta_c - \delta_{act}$</td>
<td>deg</td>
<td>difference between the reconstructed and actual deflection data</td>
</tr>
</tbody>
</table>

3.5.2 Pitch, Yaw, and Roll Component Response

If the option (IOC = 1) is chosen the pseudo-inverse of the control surface mixing gain matrix is computed. The pitch, yaw, and roll average components based on the four measured control surfaces are then computed. These are presented in Figure 6 for the sample problem. The parameters for the pitch, yaw, and roll channels are,

<table>
<thead>
<tr>
<th>OUTPUT LABEL</th>
<th>SYMBOLS</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND</td>
<td>$\delta_c$</td>
<td>deg</td>
<td>reconstructed pitch, yaw, and roll control error signal based on the telemetered autopilot data</td>
</tr>
<tr>
<td>ACTUAL</td>
<td>$\delta_{act}$</td>
<td>deg</td>
<td>pitch, yaw, and roll component of the control surface deflection based on the telemetered control surface deflection and the pseudo-inverse gain matrix</td>
</tr>
<tr>
<td>DELTA</td>
<td>$\delta_c - \delta_{act}$</td>
<td>deg</td>
<td>difference between the above values</td>
</tr>
</tbody>
</table>

3.5.3 CALCOMP Plots

If (IOP = 1) CALCOMP type plots are generated corresponding to the data printed. However, all computed points are plotted, whereas printed output can be suppressed by (NPRT) on the second input card. Sample problem plots are presented in Figure 7 and correspond to the descriptions presented in paragraphs 3.5.1 and 3.5.2.
FIGURE 1
Control System Block Diagram

\[ \vec{\delta} = [A] \vec{\theta} \]

\[ \delta_1 = \frac{\omega_{act}}{s + \omega_{act}} \]

\[ \delta_2 = \frac{\omega_{act}}{s + \omega_{act}} \]

\[ \delta_3 = \frac{\omega_{act}}{s + \omega_{act}} \]

\[ \delta_4 = \frac{\omega_{act}}{s + \omega_{act}} \]
**FIGURE 2**

**PROGRAM SUBROUTINE INTERACTION MAP**

<table>
<thead>
<tr>
<th></th>
<th>CURVE</th>
<th>DASH</th>
<th>ERSIG</th>
<th>FILFIL</th>
<th>FIN</th>
<th>PSEUDO</th>
<th>RUNGE</th>
<th>SIMEQ</th>
<th>TBLN</th>
<th>TRESP</th>
<th>XMULT</th>
<th>YDOT</th>
<th>CALCOMP LIBRARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>FINRES (MAIN)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CURVE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DASH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERSIG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FILFIL</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSEUDO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RUNGE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIMEQ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TBLN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRESP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XMULT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YDOT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 3
Flow Chart of FINRES

START

Read title and option constants IOP, IOC, NPRT, NINT

Read telemetry adjustments

Read control gain constants

Read control transformation matrix [A]

Read control surface non-linear adjustment tables

test for plot

yes

IOP.EQ.0

no

Read plot scale factors

20

yes

IOC.EQ.0

no

Compute pseudo-inverse [A#]

yes

NER.EQ.0

no

IOC=0

Write error message

A

40
Figure 3 (continued)
Flow Chart of FINRES

A

40

Rewind 9

NP=0

50

Read 1 time point of telemetry data

test for End-of-Files

EOF(5).NE.0

yes

Rewind 9

70

Rewind 9

80

Read times, rates and displacements from Tape 9

Filter, adjust & compute pitch, yaw and roll error signals $e_\theta, e_\psi, e_\phi$

Rewind 9

100

Read control surface deflection from Tape 9

B

Filter four surface deflections $\delta_1, \delta_2, \delta_3, \delta_4$

110

Compute commanded deflections and difference from actuals

140

Print Individual control surface deflection data
Figure 3 (concluded)
Flow Chart of FINRES

B

test for plot

no

yes (no plot)

Plot individual control surface deflection data

160

test for pseudo-inverse data

no

yes (none)

IOC.EQ.0

165

Compute $\delta_1, \delta_2, \delta_3, \delta_4$ from $\delta_1, \delta_2, \delta_3, \delta_4$ and [A#]

170

Print output of pitch, yaw and roll components

test for plot

no

yes (no plot)

IOP.EQ.0

Plot pitch, yaw, and roll components

STOP
<table>
<thead>
<tr>
<th>Problem Input Data</th>
</tr>
</thead>
</table>

Figure 4 (continued)

- 27 -
<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
<th>Column 4</th>
<th>Column 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.555</td>
<td>.4.49</td>
<td>.282</td>
<td>.018</td>
<td>1.001</td>
</tr>
<tr>
<td>5.655</td>
<td>.5.45</td>
<td>.282</td>
<td>.018</td>
<td>1.001</td>
</tr>
<tr>
<td>5.705</td>
<td>.5.60</td>
<td>.282</td>
<td>.018</td>
<td>1.001</td>
</tr>
<tr>
<td>5.805</td>
<td>.5.95</td>
<td>.282</td>
<td>.018</td>
<td>1.001</td>
</tr>
<tr>
<td>5.855</td>
<td>.5.95</td>
<td>.282</td>
<td>.018</td>
<td>1.001</td>
</tr>
<tr>
<td>5.905</td>
<td>.5.95</td>
<td>.282</td>
<td>.018</td>
<td>1.001</td>
</tr>
<tr>
<td>5.955</td>
<td>.5.95</td>
<td>.282</td>
<td>.018</td>
<td>1.001</td>
</tr>
<tr>
<td>6.005</td>
<td>.5.95</td>
<td>.282</td>
<td>.018</td>
<td>1.001</td>
</tr>
</tbody>
</table>

Figure 4 (continued)
Sample Problem Input Data
| 11.055 | - 0.598 | 0.061 | 0.033 | 1.010 | 1.166 | 0.615 | 0.778 | 6.828 | 3.150 | 5.978 |
| 11.105 | - 0.596 | 0.042 | 0.030 | 1.012 | 1.148 | 0.540 | 0.760 | 6.128 | 3.257 | 6.043 |
| 11.155 | - 0.594 | 0.039 | 0.029 | 0.998 | 1.189 | 0.529 | 0.769 | 6.176 | 3.268 | 6.053 |
| 11.205 | - 0.592 | 0.035 | 0.027 | 0.988 | 1.136 | 0.529 | 0.769 | 6.176 | 3.268 | 6.053 |
| 11.254 | - 0.587 | 0.030 | 0.025 | 0.958 | 1.203 | 0.591 | 0.760 | 6.164 | 3.150 | 6.139 |
| 11.305 | - 0.582 | 0.026 | 0.023 | 0.909 | 1.233 | 0.553 | 0.673 | 6.242 | 2.953 | 6.219 |
| 11.355 | - 0.577 | 0.022 | 0.020 | 0.859 | 1.267 | 0.496 | 0.684 | 6.281 | 2.810 | 6.235 |
| 11.405 | - 0.572 | 0.019 | 0.017 | 0.809 | 1.310 | 0.516 | 0.661 | 6.128 | 2.810 | 6.235 |
| 11.455 | - 0.567 | 0.015 | 0.014 | 0.759 | 1.353 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |
| 11.505 | - 0.562 | 0.012 | 0.011 | 0.709 | 1.396 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |
| 11.555 | - 0.557 | 0.009 | 0.008 | 0.659 | 1.439 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |
| 11.605 | - 0.552 | 0.006 | 0.006 | 0.609 | 1.483 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |
| 11.655 | - 0.547 | 0.003 | 0.003 | 0.559 | 1.526 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |
| 11.705 | - 0.542 | - 0.001 | 0.001 | 0.509 | 1.568 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |
| 11.755 | - 0.537 | - 0.003 | 0.003 | 0.460 | 1.610 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |
| 11.805 | - 0.532 | - 0.006 | 0.006 | 0.411 | 1.652 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |
| 11.855 | - 0.527 | - 0.009 | 0.009 | 0.362 | 1.694 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |
| 11.905 | - 0.522 | - 0.011 | 0.011 | 0.313 | 1.736 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |
| 11.955 | - 0.517 | - 0.014 | 0.014 | 0.264 | 1.777 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |
| 12.005 | - 0.512 | - 0.016 | 0.016 | 0.215 | 1.819 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |
| 12.055 | - 0.507 | - 0.018 | 0.018 | 0.166 | 1.860 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |
| 12.105 | - 0.502 | - 0.021 | 0.021 | 0.116 | 1.901 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |
| 12.155 | - 0.497 | - 0.023 | 0.023 | 0.067 | 1.942 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |
| 12.205 | - 0.492 | - 0.026 | 0.026 | 0.017 | 1.982 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |
| 12.255 | - 0.487 | - 0.028 | 0.028 | - 0.001 | 2.022 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |
| 12.305 | - 0.482 | - 0.031 | 0.031 | - 0.05 | 2.062 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |
| 12.355 | - 0.477 | - 0.034 | 0.034 | - 0.10 | 2.101 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |
| 12.405 | - 0.472 | - 0.036 | 0.036 | - 0.15 | 2.141 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |
| 12.455 | - 0.467 | - 0.039 | 0.039 | - 0.20 | 2.181 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |
| 12.505 | - 0.462 | - 0.042 | 0.042 | - 0.25 | 2.220 | 0.517 | 0.661 | 6.128 | 2.810 | 6.235 |

Figure 4 (continued)
Sample Problem Input Data
Sample Problem Input Data
<table>
<thead>
<tr>
<th>TIME</th>
<th>F1M ONE</th>
<th>F1M TWO</th>
<th>F1M THREE</th>
<th>F1M FOUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.45</td>
<td>1.52</td>
<td>-0.81</td>
<td>0.29</td>
<td>-0.87</td>
</tr>
<tr>
<td>-0.20</td>
<td>1.52</td>
<td>-0.81</td>
<td>0.29</td>
<td>-0.87</td>
</tr>
<tr>
<td>-1.21</td>
<td>1.52</td>
<td>-0.81</td>
<td>0.29</td>
<td>-0.87</td>
</tr>
<tr>
<td>-1.13</td>
<td>1.52</td>
<td>-0.81</td>
<td>0.29</td>
<td>-0.87</td>
</tr>
<tr>
<td>-2.38</td>
<td>-2.52</td>
<td>1.14</td>
<td>-1.22</td>
<td>-0.90</td>
</tr>
<tr>
<td>-2.38</td>
<td>-2.52</td>
<td>1.14</td>
<td>-1.22</td>
<td>-0.90</td>
</tr>
<tr>
<td>1.06</td>
<td>-2.17</td>
<td>-0.83</td>
<td>-0.41</td>
<td>-0.83</td>
</tr>
<tr>
<td>1.06</td>
<td>-2.17</td>
<td>-0.83</td>
<td>-0.41</td>
<td>-0.83</td>
</tr>
<tr>
<td>1.56</td>
<td>-1.87</td>
<td>-0.32</td>
<td>0.89</td>
<td>-0.89</td>
</tr>
<tr>
<td>1.56</td>
<td>-1.87</td>
<td>-0.32</td>
<td>0.89</td>
<td>-0.89</td>
</tr>
<tr>
<td>2.06</td>
<td>-0.21</td>
<td>-0.59</td>
<td>0.39</td>
<td>-0.59</td>
</tr>
<tr>
<td>2.06</td>
<td>-0.21</td>
<td>-0.59</td>
<td>0.39</td>
<td>-0.59</td>
</tr>
<tr>
<td>2.56</td>
<td>-0.37</td>
<td>-0.37</td>
<td>0.37</td>
<td>-0.37</td>
</tr>
<tr>
<td>2.56</td>
<td>-0.37</td>
<td>-0.37</td>
<td>0.37</td>
<td>-0.37</td>
</tr>
<tr>
<td>3.06</td>
<td>-0.67</td>
<td>-0.67</td>
<td>0.67</td>
<td>-0.67</td>
</tr>
<tr>
<td>3.06</td>
<td>-0.67</td>
<td>-0.67</td>
<td>0.67</td>
<td>-0.67</td>
</tr>
<tr>
<td>3.56</td>
<td>-0.27</td>
<td>-0.27</td>
<td>0.27</td>
<td>-0.27</td>
</tr>
<tr>
<td>3.56</td>
<td>-0.27</td>
<td>-0.27</td>
<td>0.27</td>
<td>-0.27</td>
</tr>
<tr>
<td>4.06</td>
<td>-0.44</td>
<td>-0.44</td>
<td>0.44</td>
<td>-0.44</td>
</tr>
<tr>
<td>4.06</td>
<td>-0.44</td>
<td>-0.44</td>
<td>0.44</td>
<td>-0.44</td>
</tr>
<tr>
<td>4.56</td>
<td>-0.68</td>
<td>-0.68</td>
<td>0.68</td>
<td>-0.68</td>
</tr>
<tr>
<td>4.56</td>
<td>-0.68</td>
<td>-0.68</td>
<td>0.68</td>
<td>-0.68</td>
</tr>
<tr>
<td>5.06</td>
<td>-0.83</td>
<td>-0.83</td>
<td>0.83</td>
<td>-0.83</td>
</tr>
<tr>
<td>5.06</td>
<td>-0.83</td>
<td>-0.83</td>
<td>0.83</td>
<td>-0.83</td>
</tr>
<tr>
<td>5.56</td>
<td>0.04</td>
<td>-0.60</td>
<td>0.60</td>
<td>-0.60</td>
</tr>
<tr>
<td>5.56</td>
<td>0.04</td>
<td>-0.60</td>
<td>0.60</td>
<td>-0.60</td>
</tr>
<tr>
<td>6.06</td>
<td>0.12</td>
<td>-0.76</td>
<td>0.76</td>
<td>-0.76</td>
</tr>
<tr>
<td>6.06</td>
<td>0.12</td>
<td>-0.76</td>
<td>0.76</td>
<td>-0.76</td>
</tr>
<tr>
<td>6.56</td>
<td>0.20</td>
<td>-0.90</td>
<td>0.90</td>
<td>-0.90</td>
</tr>
<tr>
<td>6.56</td>
<td>0.20</td>
<td>-0.90</td>
<td>0.90</td>
<td>-0.90</td>
</tr>
<tr>
<td>7.06</td>
<td>0.28</td>
<td>-1.06</td>
<td>1.06</td>
<td>-1.06</td>
</tr>
<tr>
<td>7.06</td>
<td>0.28</td>
<td>-1.06</td>
<td>1.06</td>
<td>-1.06</td>
</tr>
<tr>
<td>7.56</td>
<td>0.37</td>
<td>-1.22</td>
<td>1.22</td>
<td>-1.22</td>
</tr>
<tr>
<td>7.56</td>
<td>0.37</td>
<td>-1.22</td>
<td>1.22</td>
<td>-1.22</td>
</tr>
<tr>
<td>8.06</td>
<td>0.46</td>
<td>-1.37</td>
<td>1.37</td>
<td>-1.37</td>
</tr>
<tr>
<td>8.06</td>
<td>0.46</td>
<td>-1.37</td>
<td>1.37</td>
<td>-1.37</td>
</tr>
</tbody>
</table>
### Figure 6
Sample Problem Printed Output Data
Pitch, Yaw and Roll Components

#### PAGE NO. 3
PITCH, YAW, ROLL COMMANDS BASED ON MEASURED FINS

<table>
<thead>
<tr>
<th>TIME</th>
<th>COMMAND PITCH</th>
<th>ACTUAL DELTA</th>
<th>COMMAND YAW</th>
<th>ACTUAL DELTA</th>
<th>COMMAND ROLL</th>
<th>ACTUAL DELTA</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.0</td>
<td>.21</td>
<td>.43</td>
<td>.15</td>
<td>.55</td>
<td>.21</td>
<td>.12</td>
</tr>
<tr>
<td>-1.1</td>
<td>.22</td>
<td>.41</td>
<td>.16</td>
<td>.53</td>
<td>.20</td>
<td>.11</td>
</tr>
<tr>
<td>-1.2</td>
<td>.23</td>
<td>.45</td>
<td>.18</td>
<td>.59</td>
<td>.23</td>
<td>.13</td>
</tr>
<tr>
<td>-1.3</td>
<td>.24</td>
<td>.43</td>
<td>.20</td>
<td>.57</td>
<td>.22</td>
<td>.12</td>
</tr>
<tr>
<td>-1.4</td>
<td>.25</td>
<td>.44</td>
<td>.21</td>
<td>.58</td>
<td>.23</td>
<td>.11</td>
</tr>
<tr>
<td>-1.5</td>
<td>.26</td>
<td>.45</td>
<td>.23</td>
<td>.59</td>
<td>.24</td>
<td>.10</td>
</tr>
<tr>
<td>-1.6</td>
<td>.27</td>
<td>.43</td>
<td>.24</td>
<td>.58</td>
<td>.22</td>
<td>.10</td>
</tr>
<tr>
<td>-1.7</td>
<td>.28</td>
<td>.45</td>
<td>.25</td>
<td>.59</td>
<td>.23</td>
<td>.10</td>
</tr>
<tr>
<td>-1.8</td>
<td>.29</td>
<td>.44</td>
<td>.26</td>
<td>.58</td>
<td>.22</td>
<td>.10</td>
</tr>
<tr>
<td>-1.9</td>
<td>.30</td>
<td>.45</td>
<td>.27</td>
<td>.59</td>
<td>.23</td>
<td>.10</td>
</tr>
</tbody>
</table>

#### PAGE NO. 4
PITCH, YAW, ROLL COMMANDS BASED ON MEASURED FINS

<table>
<thead>
<tr>
<th>TIME</th>
<th>COMMAND PITCH</th>
<th>ACTUAL DELTA</th>
<th>COMMAND YAW</th>
<th>ACTUAL DELTA</th>
<th>COMMAND ROLL</th>
<th>ACTUAL DELTA</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.1</td>
<td>4.36</td>
<td>5.11</td>
<td>.75</td>
<td>1.34</td>
<td>1.39</td>
<td>.05</td>
</tr>
<tr>
<td>12.2</td>
<td>4.16</td>
<td>5.05</td>
<td>.75</td>
<td>1.01</td>
<td>1.09</td>
<td>.03</td>
</tr>
<tr>
<td>12.3</td>
<td>4.51</td>
<td>4.99</td>
<td>.99</td>
<td>1.09</td>
<td>1.09</td>
<td>.03</td>
</tr>
<tr>
<td>12.4</td>
<td>4.35</td>
<td>4.95</td>
<td>.95</td>
<td>1.09</td>
<td>1.09</td>
<td>.03</td>
</tr>
<tr>
<td>12.5</td>
<td>4.52</td>
<td>4.99</td>
<td>.99</td>
<td>1.09</td>
<td>1.09</td>
<td>.03</td>
</tr>
<tr>
<td>12.6</td>
<td>4.41</td>
<td>4.96</td>
<td>.96</td>
<td>1.09</td>
<td>1.09</td>
<td>.03</td>
</tr>
<tr>
<td>12.7</td>
<td>4.35</td>
<td>4.94</td>
<td>.94</td>
<td>1.09</td>
<td>1.09</td>
<td>.03</td>
</tr>
<tr>
<td>12.8</td>
<td>4.42</td>
<td>4.96</td>
<td>.96</td>
<td>1.09</td>
<td>1.09</td>
<td>.03</td>
</tr>
<tr>
<td>12.9</td>
<td>4.36</td>
<td>4.95</td>
<td>.95</td>
<td>1.09</td>
<td>1.09</td>
<td>.03</td>
</tr>
</tbody>
</table>

- 32 -
Figure 7
Sample Problem CALCOMP Plots
Figure 7 (continued)
Sample Problem CALCOMP Plots
APPENDIX A

FORTRAN PROGRAM LISTING

A complete FORTRAN source program listing is presented in the following pages. It starts with the main routine (FINRES) and is followed by the subroutines arranged in alphabetical order. There are a total of 182 cards in FINRES. The total program including subroutines (less CALCOMP library) contains 675 cards.
PROGRAM FINRES(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE9)
C THIS PROGRAM ANALYZES POST-FLIGHT MEASURED CONTROL SURFACE
C DEFLECTIONS AND AUTOPILOT PARAMETERS. IT PROVIDES AN ADJUST-
CMENT CAPABILITY AND DIGITAL FILTERING OF THE REDUCED TELEMETRY
C DATA INCLUDING TIME SHIFTS.
DIMENSION CD1(20),ED1(20),CD2(20),ED2(20),CD3(20),ED3(20),CD4(20),
ED4(20),A(4,4),APS(4,4),FILTER(3,3),B(3),DATA(11),DL(10),
DATA NM1,NM2,NM3,NM4,NM5,NM6,NM7/10HFIN NO. 1 ,10HFIN NO. 2 ,
10HFIN NO. 3 ,10HFIN NO. 4 ,10HPITCH FINS ,10HYAU FINS ,10HROLL FINS /
C READ TITLE CARD
READ(5,199) (NTIT(J),J=1,8)
199 FORMAT(8A10)
C READ INTEGER OPTIONS AND CONSTANTS
READ(5,220) IOC,IOP,NPRT,NINT
220 FORMAT(4IS)
C READ TELEMETRY ADJUSTMENT FACTORS AND TIME FACTORS
READ(5,230) (DL(I),I=1,10),TCQ,TCR,TCP,TCTH,TCPS,TCPH,
TC1,TC2,TC3,TC4,AKTH,AKPS,AKPH,U10-6.2832iUC0
230 FORMAT(8E10.3)
C READ AUTOPILOT GAINS AND BREAK FREQUENCIES
READ(5,230) CTH,CTHD,CPS,CPSD,CPH,CPHD,U1Q,U1R,U1P,UACT
C READ FIN GAIN MATRIX
DO 10 J=1,4
10 READ(5,240) (A(J,I),I=1,3)
240 FORMAT(6E10.3)
C READ MEASURED FIN CALIBRATION ADJUSTMENT TABLES
READ(5,250) NT1,(CD1(J),ED1(J),J=1,NT1)
READ(5,250) NT2,(CD2(J),ED2(J),J=1,NT2)
READ(5,250) NT3,(CD3(J),ED3(J),J=1,NT3)
READ(5,250) NT4,(CD4(J),ED4(J),J=1,NT4)
250 FORMAT(5I,/(8E10.3))
C TEST FOR PLOT OPTION
IF(IOP.EQ.0) GO TO 20
C CALCULATE PLOT. READ IN SCALE FACTORS (UNITS PER INCH)
READ(5,230) XSCALE, YSCALE, DSSCALE
IF(XSCALE.LE.0.) IOP=0
IF(YSCALE.LE.0.) IOP=0
IF(DSCALE.LE.0.) IOP=0
C TEST FOR PSEUDO-INVERSE CALCULATION OF PITCH, YAW, ROLL DEFLECTION
20 IF(I0C.EQ.0) GO TO 40
C COMPUTE PSEUDO-INVERSE OF FIN GAIN MATRIX
N=4
M=3
CALL PSEUDO(APS,A,N,M,NER)
IF(NER.EQ.1) GO TO 40
IOC=0
WRITE(6,200)
200 FORMAT(/1,5X,34HMATRIX IS SINGULAR, CANNOT COMPUTE ,
1 1X,14HPSEUDO-INVERSE ,/) 40 NP=0
REWIND 9
M1=1
M2=1
M3=1
M4=1
C BEGIN READING THE TELEMETRY DATA
50 READ(5,104) (DATA(I),I=1,11)
104 FORMAT(11F7.3)
IF(EOF(5).NE.0) GO TO 70
NP=NP+1
C ADJUST TELEMETRY DATA FOR BIAS AND FIN CAL NON-LINEARITY
DO 60 I=1,6
60 DATA(I+1)=DATA(I+1)+DL(I)
X=DATA(8)
CALL TBLN(Y,X,CD1,ED1,NT1,M1)
DATA(8)=DATA(8)+Y+DL(7)
X=DATA(9)
CALL TBLN(Y,X,CD2,ED2,NT2,M2)
DATA(9)=DATA(9)+Y+DL(8)
X=DATA(10)
CALL TBLN(Y,X,CD3,ED3,NT3,M3)
DATA(10)=DATA(10)+Y+DL(9)
X=DATA(11)
CALL TBLN(Y,X,CD4,ED4,NT4,M4)
DATA(11)=DATA(11)+Y+DL(10)

WRITE ADJUSTED DATA ON TAPE 9
WRITE(9,104) (DATA(I),I=1,11)
GO TO 50

70 CONTINUE
C READ TAPE 9 DATA INTO TABLES
REWIND 9
DO 80 J=1,NP
80 READ(9,104) T(J),Q(J),R(J),P(J),TH(J),PS(J),PH(J)
C INITIALIZE FILTER AND FILTER THESE PARAMETERS
CALL FILFIL(WCO,FILTER,B,DTFIL)
C COMPUTE PITCH ERROR SIGNAL AND RETURN IN Q ARRAY
CALL ERSIG(FILTER,B,T,Q,TH,NP,NINT,CTH,CTHD,DTFIL,
  TCTH,TCQ,AKTH,U1Q,WACT)
C COMPUTE YAW ERROR SIGNAL AND RETURN IN R ARRAY
CALL ERSIG(FILTER,B,T,R,PS,NP,NINT,CPS,CPSD,DTFIL,
  TCP,R,AKPS,U1R,WACT)
C COMPUTE ROLL ERROR SIGNAL AND RETURN IN P ARRAY
CALL ERSIG(FILTER,B,T,P,PH,NP,NINT,CPH,CPHD,DTFIL,
  TCPHP,TCR,AKPH,U1P,UACT)
C COMPUTE FOUR CONTROL SURFACE DEFLECTIONS AND FILTER
REWIND 9
DO 100 J=1,NP
100 READ(9,105) T(J),D1(J),D2(J),D3(J),D4(J)
105 FORMAT(F7.3,42X,4F7.3)
C FILTER AND ADJUST CONTROL SURFACES
CALL FIN(FILTER,B,T,D1,NP,DTFIL,TCP,NINT)
CALL FIN(FILTER,B,T,D2,DP,DTFIL,TC2,NINT)
CALL FIN(FILTER,B,T,D3,DP,DTFIL,TC3,NINT)
CALL FIN(FILTER,B,T,D4,DP,DTFIL,TC4,NINT)
THE PITCH, yaw, and roll reconstructed and filtered
DEFLECTIONS ARE AVAILABLE. NEXT COMPUTE THE COMMANDED
INDIVIDUAL FINS AND STORE IN ARRAYS TH,PS,PH,DUM
DO 110 J=1,NP
TH(J)=A(1,1)*Q(J)+A(1,2)*R(J)+A(1,3)*P(J)
PS(J)=A(2,1)*Q(J)+A(2,2)*R(J)+A(2,3)*P(J)
PH(J)=A(3,1)*Q(J)+A(3,2)*R(J)+A(3,3)*P(J)
110 DUM(J)=A(4,1)*Q(J)+A(4,2)*R(J)+A(4,3)*P(J)
C PREPARE AND PRINT OUTPUT OF INDIVIDUAL FINS
NPAGE=1
NLINE=59
KPRT=NPRT
DO 140 J=1,NP
IF(NLINE.LT.59) GO TO 130
WRITE(6,205) NPAGE,(NTIT(I),I=1,8)
NLINE=8
NPAGE=NPAGE+1
205 FORMAT(1H1,2X,8HPAGE NO.,I3,/,1X,8A10,/,9X,16H* FIN ONE
1,8X,16H* FIN TWO,8X,19H* FIN THREE,
2 5X,18H FIN FOUR,/,3X,6HTIME,
3 4(24H COMMAND ACTUAL DELTA ),/)
130 IF(KPRT.LT.NPRT) GO TO 140
KPRT=0
DLT1=TH(J)-D1(J)
DLT2=PS(J)-D2(J)
DLT3=PH(J)-D3(J)
DLT4=DUM(J)-D4(J)
WRITE(6,206) T(J),TH(J),D1(J),DLT1,PS(J),D2(J),DLT2,PH(J),D3(J),
1 DLT3,DUM(J),D4(J),DLT4
NLINE=NLINE+1
140 KPRT=KPRT+1
206 FORMAT(1X,13F8.2)
TEST FOR PLOT

IF(CIOC.EQ.0) GO TO 160

CALCOMP PLOT OF INDIVIDUAL CONTROL SURFACES
CALL CURVE(T,TH,D1,NP,NTIT,NM1,XSCALE,YSCALE,DSCALE)
CALL CURVE(T,PS,D2,NP,NTIT,NM2,XSCALE,YSCALE,DSCALE)
CALL CURVE(T,PH,D3,NP,NTIT,NM3,XSCALE,YSCALE,DSCALE)
CALL CURVE(T,DUM,D4,NP,NTIT,NM4,XSCALE,YSCALE,DSCALE)

TEST FOR ERROR SIGNAL RECONSTRUCTION BASED ON MEASURED FINS

160 IF(CIOC.EQ.0) GO TO 180

RECONSTRUCT PITCH, YAW, AND ROLL ERROR SIGNAL FROM MEASURED
FIN DEFLECTIONS AND OUTPUT. PITCH, YAW, AND ROLL DEFLECTIONS
BASED ON FINS 1 THRU 4 WILL BE IN ARRAYS TH,PS, AND PH

NLINE=59
DO 170 J=1,NP
IF(NLINE.LT.59) GO TO 165
WRITE(6,207) NPAGE,(NTIT(I),I=1,8)
NLINE=8
NPAGE=NPAGE+1
KPRT=NPRT

165 TH(J)=APS(1,1)*D1(J)+APS(1,2)*D2(J)+APS(1,3)*D3(J)+APS(1,4)*D4(J)
PS(J)=APS(2,1)*D1(J)+APS(2,2)*D2(J)+APS(2,3)*D3(J)+APS(2,4)*D4(J)
PH(J)=APS(3,1)*D1(J)+APS(3,2)*D2(J)+APS(3,3)*D3(J)+APS(3,4)*D4(J)

COMPUTE DIFFERENCES
DLT1=Q(J)-TH(J)
DLT2=R(J)-PS(J)
DLT3=P(J)-PH(J)
IF(KPRT.LT.NPRT) GO TO 170
KPRT=0
WRITE(6,206) T(J),Q(J),TH(J),DLT1,R(J),PS(J),DLT2,P(J),PH(J),DLT3
NLINE=NLINE+1

170 KPRT=KPRT+1

207 FORMAT(1H1,2X,8HPAGE NO.,I3,10X,48HPITCH, YAW, ROLL COMMANDS BASED
1 ON MEASURED FINS ,/,,8A10,,8X,1H*,7X,
2 9HP I T C H ,7X,1H*,9X,5HY A W ,9X,1H*,7X,7HR O L L ,9X,
3 1H*,13X,6HTIME ,3(24H COMMAND ACTUAL DELTA ),/)

141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175
IF(IOP.EQ.0) GO TO 180
C
PLOT CALCOMP PLOT OF PITCH, YAW, AND ROLL FINS
CALL CURVE(T,Q,TH,NP,NTIT,NM5,XSCALE,YSCALE,DSCALE)
CALL CURVE(T,R,PS,NP,NTIT,NM6,XSCALE,YSCALE,DSCALE)
CALL CURVE(T,P,PH,NP,NTIT,NM7,XSCALE,YSCALE,DSCALE)
180 STOP
END
SUBROUTINE CURVE(T,CALC,ACT,NP,NTIT,NM,XS,YS,DS)
C
THIS SUBROUTINE PLOTS A CALCOMP TYPE PLOT OF THE COMMANDED
C
ACTUAL, AND DIFFERENCE BETWEEN THE COMMANDED AND ACTUAL
C
CONTROL SURFACE DEFLECTION.
DIMENSION T(2000),CALC(2000),ACT(2000),NTIT(8),X(2),Y(2)
DATA X(1),X(2),Y(1),Y(2),NP,ZL,ZK,SPACE,YMAX,DTMAX/5.0,6.2,
1         2.75,2.75,2.0,0.07,0.0,0.04,3.15/
DM=-2.*XS
DD=-DS
TF=T(NP)/XS + 1.
NTF=TF
TF=NTF
CALL PLOTS(5,HCAL22,0,4,PLOT)
CALL PLOT(1.,6.,-3)
CALL SYMBOL(2.,3.75,0.14,NTIT,0.,80)
CALL SYMBOL(2.,3.5,0.14,NTM,0.,10)
CALL AXIS(0.,-2.,30,HCONTROL SURFACE DEFLECTION DEG,30,6.,90.,DM,
1         YS)
CALL AXIS(0.,-2.,13,HTIME SECONDS, -13,TF,0.,0.,YS)
CALL DASH(T,ACT,NP,ZL,SPACE,ZK,XS,YS,1,TF,YMAX)
CALL DASH(T,CALC,NP,ZL,ZK,SPACE,XS,YS,1,TF,YMAX)
CALL PLOT(5.,2.95,3)
CALL PLOT(6.2,2.95,2)
CALL SYMBOL(6.3,2.95,0.1,12,HMEASURED FIN,0.,12)
CALL SYMBOL(6.3,2.75,0.1,13,HRECONSTRUCTED,0.,13)
CALL PLOT(5.,2.75,3)
CALL DASH(X,Y,NPP,ZL,ZK,SPACE,1.,1.,1,TF,4.)
CALL PLOT(0.,-4.,-3)
CALL AXIS(0.,-2.,13,HTIME SECONDS, -13,TF,0.,0.,XS)
CALL AXIS(0.,-1.,20,HDIFFERENCE CALC-ACT,20,2.,90.,DD,DS)
C
CALCULATE AND PLOT THE DIFFERENCE (RECONSTRUCTED MINUS ACTUAL)
DO 10 J=1,NP
10 CALC(J)=(CALC(J)-ACT(J))/DS
CALL DASH(T,CALC,NP,ZL,SPACE,ZK,XS,1.,1,TF,DTMAX)
CALL PLOT(16.,-2.,-3)
CALL PLOT(0.,0.,999)
RETURN
END
DECK DASH

SUBROUTINE DASH (X,Y,NP,Z1,Z2,SPACE,XSCALE,YSCALE,LSYM,XLIM,YLIM)

C SYMBOLS, DASHED, DASHED-DOT LINES OR SOLID LINES WITH OR WITHOUT
C SYMBOLS
C BASED ON A SET OF SEQUENTIAL POINTS GIVEN IN
C THE INPUT 'X' ABSCISSA ARRAY AND THE 'Y' ORDINATE ARRAY

DIMENSION X(1),Y(1)

DO 10 I=1,NP
   XA=X(I)/XSCALE
   YA=Y(I)/YSCALE
   IF (ABS(XA).GT.XLIM) GO TO 10
   IF (ABS(YA).GT.YLIM) GO TO 10
   CALL PLOT (XA,YA,3)
   GO TO 20
10 CONTINUE

20 IF (SPACE) 330,310,30
C PLOT A BROKEN LINE
30 K=0
   PI2=1.5708
   Z=Z1
   ZB=Z2
   IF (Z2.GT.0.) GO TO 40
   ZB=Z1
40 ZD=Z
   LZ=0
   SL=0.
   NF-NP-1
   DO 300 J=1,NF
      XA=X(J)/XSCALE
      IF (ABS(XA)-XLIM .GT. 0.) GO TO 300
      XB=X(J+1)/XSCALE
      IF (ABS(XB)-XLIM .GT. 0.) GO TO 300
      YA=Y(J)/YSCALE
      IF (ABS(YA)-YLIM .GT. 0.) GO TO 300
      YB=Y(J+1)/YSCALE
      IF (ABS(YB)-YLIM .GT. 0.) GO TO 300
   300 CONTINUE
DY=YG-YA
DX=XB-XA
IF (DX .GE. 0.) GO TO 80
IF (DY) 50,60,70
50 TH=-PI2
GO TO 90
60 TH=0.
GO TO 90
70 TH=PI2
GO TO 90
80 TH=ATAN(DY/DX)
90 DX=XB-XA
DY=YG-YA
DZ=SQR(DX+DY)
C TEST TO SEE WHAT IS GOING ON
IF (K) 100,180,220
100 K=1
SL=SPACE
IF (DZ-SPACE) 110,120,150
C SPACE IS LARGER THAN DZ
110 SL=SL-DZ
CALL PLOT (XB,VB,3)
GO TO 300
C NEXT POINT IS EXACTLY ONE SPACE
120 K=0
IF (LZ .GE. 0) GO TO 130
ZD=ZB
LZ=1
GO TO 140
130 ZD=Z
LZ=0
140 SL=0.
CALL PLOT (XB,VB,3)
GO TO 300
C NEXT POINT MORE THAN ONE SPACE AWAY
150 XA=XA+SPACE*COS(TH)
    YA=YA+SPACE*SIN(TH)
    IF (ABS(XA)-XLIM .GE. 0.) GO TO 300
    IF (ABS(YA)-YLIM .GE. 0.) GO TO 300
    K=0
    IF (LZ .NE. 0 ) GO TO 160
    ZD=ZB
    LZ=1
    GO TO 170
160 ZD=Z
    LZ=0
170 SL=0.
    CALL PLOT (XA,YA,3)
    GO TO 90
C K=0 LINE BEING DRAWN ZD LENGTH NOT DRAWN RESUME AS IS LINE STARTING
180 IF (DZ-ZD) 190,200,210
C LINE GOES AT LEAST TO NEXT POINT
190 K=0
    ZD=ZD-DZ
    CALL PLOT (XB,YB,2)
    GO TO 300
C LINE ENDS AT NEXT POINT
200 K=-1
    SL=SPACE
    ZD=0.
    CALL PLOT (XB,YB,2)
    GO TO 300
C LINE ENDS BEFORE NEXT POINT
210 K=1
    SL=SPACE
    XA=XA+ZD*COS(TH)
    YA=YA+ZD*SIN(TH)
    IF (ABS(XA)-XLIM .GE. 0.) GO TO 300
    IF (ABS(YA)-YLIM .GE. 0.) GO TO 300
    CALL PLOT (XA,YA,2)
ZD = 0.
GO TO 90
C K = 1 IS IN SPACE
220 ZD = 0.
   IF (DZ - SL) 230, 240, 270
230 K = 1
   SL = SL - DZ
   CALL PLOT (XB, YB, 3)
   GO TO 300
C SL = DZ
240 K = 0
   IF (LZ .NE. 0) GO TO 250
   ZD = ZB
   LZ = 1
   GO TO 260
250 ZD = Z
   LZ = 0
   CALL PLOT (XB, YB, 3)
   GO TO 300
C SL IS LESS THAN DZ
270 K = 0
   IF (LZ .NE. 0) GO TO 280
   ZD = ZB
   LZ = 1
   GO TO 290
280 ZD = Z
   LZ = 0
   XA = XA + SL * COS(TH)
   YA = YA + SL * SIN(TH)
   IF (ABS(XA) - XLIM .GE. 0.) GO TO 300
   IF (ABS(YA) - YLIM .GE. 0.) GO TO 300
   SL = 0.
   CALL PLOT (XA, YA, 3)
   GO TO 90
300 CONTINUE
GO TO 370
C STRAIGHT LINE PLOT OPTION
310 DO 320 J=I,NP
XA=X(J)/XSCALE
YA=Y(J)/YSCALE
IF (ABS(XA)-XLI".GT. 0.) GO TO 320
IF (ABS(YA)-YLIM .GT. 0.) GO TO 320
CALL PLOT (XA,YA,2)
320 CONTINUE
GO TO 370
C PLOT SYMBOLS ON LINE NO LINE IF LYSMB IS NEGATIVE
330 NSM=IABS(LYSMB)
IF (LYSMB .LT. 0 ) GO TO 340
K=-2
GO TO 350
340 K=-1
350 DO 360 J=1,NP
XA=X(J)/XSCALE
YA=Y(J)/YSCALE
IF (ABS(XA)-XLI".GT. 0.) GO TO 360
IF (ABS(YA)-YLIM .GT. 0.) GO TO 360
CALL SYMBOL (XA,YA,0.07,NSM,0.0,K)
360 CONTINUE
370 CALL PLOT (0.,0.,3)
RETURN
END
SUBROUTINE ERSIG(Filter,B,T,Q,TH,NP,NINT,CTH,CTHD,DT, 1  DTT,DTTM,AD,U,WACT)  
* This subroutine adjusts the attitude rate and displacement  
for telemetry cross coupling, time lags, and then computes  
the error signal. The displacement and rate are entered  
in arrays TH and Q. The error signal is returned in (Q).  
FILTER THE DISPLACEMENT TIME HISTORY  
N=3  
K=3  
CT=CTHD-AD*CTH  
CALL TRESP(Filter,B,T,TH,DUM,NP,N,K,NINT)  
CALL TRESP(Filter,B,T,Q,TH,NP,N,K,NINT)  
M=1  
DTT=DT+DTTM  
DTR=DT+DTTMR  
AT THIS POINT THE FILTERED DISPLACEMENT IS IN (DUM) AND THE  
FILTERED RATE IS IN (TH)  
DO 10 J=1,NP  
TA=T(J)+DTT  
CALL TBLN(DISp,TA,T,DUM,NP,M)  
TB=T(J)+DTR  
CALL TBLN(RATE,TB,T,TH,NP,M)  
C COMPUTE THE ERROR SIGNAL AND PLACE IN (Q)  
10 Q(J)=CTH*DISP+CT*RATe  
NEXT COMPUTE THE FIRST ORDER AUTOPILOT LAG (W) AND THE FIRST  
ORDER ACTUATOR LAG (WACT) RESPONSE.  
AB(2)=0.  
A(1,2)=0.  
N=2  
K=2  
AB(1)=W  
A(1,1)=-U  
A(2,1)=WACT  
A(2,2)=-WACT  
CALL TRESP(A,AB,T,Q,DUM,NP,N,K,NINT)  
C PUT RESULTANT ERROR SIGNAL IN (Q) ARRAY.  
DO 30 J=1,NP  
30 Q(J)=DUM(J)  
RETURN  
END
*DECK FILFIL
SUBROUTINE FILFIL(UCO, A, B, DT)
C THIS SUBROUTINE FILLS THE FILTER COEFFICIENTS FOR A THIRD
C ORDER BUTTERWORTH FILTER HAVING A CUTOFF FREQUENCY OF
C (UCO) RADIANS PER SECOND.
DIMENSION A(3,3), B(3)
C ZERO OUT THE COEFFICIENT ARRAYS
DO 10 J=1,3
   B(J)=0.
   DO 10 K=1,3
      A(J,K)=0.
10      
C FILL REMAINING CONSTANTS
TAU=0.7071/UCO
A(2,1)=1.0
A(3,2)=1.0
A(1,1)=-2./TAU
A(1,2)=A(1,1)/TAU
A(1,3)=-1./(TAU*TAU*TAU)
B(1)=-A(1,3)
C COMPUTE THE LOW FREQUENCY TIME LAG FOR THIS FILTER
DT=2.*TAU
RETURN
END
*DECK FIN

SUBROUTINE FIN(FILTER,B,T,D,NP,DTFIL,TC,NINT)
C THIS SUBROUTINE FILTERS THE CONTROL SURFACE DEFLECTION AND
C ADJUSTS FOR TIME DELAYS.
N=3
K=3
DT=DTFIL+TC
CALL TRESP(FILTER,B,T,D,DUM,NP,N,K,NINT)
M=1
C THE FILTERED DEFLECTION IS NOW IN (DUM)
DO 10 J=1,NP
   TA=T(J)+DT
   CALL TBLN(FN,TA,T,DUM,NP,M)
10   D(J)=FN
RETURN
END
*DECK PSEUDO
SUBROUTINE PSEUDO(B,A,N,M,NER)
C THIS SUBROUTINE COMPUTES THE PSEUDO INVERSE MATRIX B FROM THE
C N BY M MATRIX (A). (B)=A*B*INVB(AT,A)*AT
DIMENSION AS(4,4),A(4,4),B(4,4),AINV(4,4)
L=M
LS=N
IF(N.GT.M) LS=M
IF(N.GT.M) L=N
C SET MATRIX ELEMENTS TO ZERO
DO 10 J=1,L
DO 10 K=1,L
B(J,K)=0.
10 AS(J,K)=0.
C COMPUTE THE TRANSPOSE OF A
DO 20 J=1,L
DO 20 K=1,L
20 AS(J,K)=A(K,J)
C MULTIPLY A TRANSPOSE TIMES A, AND STORE IN B
CALL XMULT(AS,A,B,L)
C COMPUTE INVERSE OF B AND STORE IN AINV
CALL SIMEQ(B,LS,AINV,NER)
IF(NER.NE.0) GO TO 30
C MATRIX IS SINGULAR RETURN TO CALLING ROUTINE
RETURN
C COMPUTE THE PSEUDO INVERSE
30 CALL XMULT(AINV,AS,B,L)
RETURN
END
*DECK RUNGE
SUBROUTINE RUNGE (N,F, H, X, Y, L,I)
C THIS SUBROUTINE PERFORMS THE RUNGE-KUTTA INTEGRATION
C UPDATES FOR THE TRESP SUBROUTINE.
DIMENSION Y(1),F(1),SV(3),FF(3)
I = I + 1
GO TO ( 1, 2, 3, 4, 5), I
1 L = 1
RETURN
2 DO 6 J=1,N
   SV(J) = Y(J)
   FF(J) = F(J)
6 Y(J) = SV(J) + .5*H*F(J)
   X = X + .5*H
   L = 1
RETURN
3 DO 7 J=1,N
   FF(J) = FF(J) + 2.*F(J)
7 Y(J) = SV(J) + .5*H*F(J)
   L = 1
RETURN
4 DO 8 J=1,N
   FF(J) = FF(J) + 2.*F(J)
8 Y(J) = SV(J) + H*F(J)
   X = X + .5*H
   L = 1
RETURN
5 DO 9 J=1,N
9 Y(J) = SV(J) + (H/6.)*(FF(J) + F(J))
   L = 2
   I = 0
RETURN
END
SUBROUTINE SIMEQ (A,KC,AINU,IERR)
C THIS SUBROUTINE FINDS THE INVERSE OF MATRIX (A) USING
C DIAGONALIZATION PROCEDURES.
N=1
IERR=1
DO 10 I=1,KC
  DO 10 J=1,KC
    AINU(I,J)=0.
10  B(I,J)=A(I,J)
  DO 20 I=1,KC
    AINU(I,I)=1.
20  X(I)=XDOT(I)
  DO 110 I=1,KC
    COMP=0.
    K=I
110 IF (ABS(B(K,I))-ABS(COMP) .LE. 0.) GO TO 40
    COMP=B(K,I)
    N=K
40  K=K+1
    IF (K-KC .LE. 0 ) GO TO 30
    IF (B(N,I) .EQ. 0.) GO TO 120
    IF (N-I) 120,70,50
50  DO 60 M=1,KC
    TEMP=B(I,M)
    B(I,M)=B(N,M)
    B(N,M)=TEMP
    TEMP=AINU(I,M)
    AINU(I,M)=AINU(N,M)
60  AINU(N,M)=TEMP
    TEMP=X(I)
    X(I)=X(N)
    X(N)=TEMP
70  X(I)=X(I)/B(I,I)
TEMP = B(I, I)
DO 80 M = 1, KC
AINV(I, M) = AINV(I, M) / TEMP
80 B(I, M) = B(I, M) / TEMP
DO 100 J = 1, KC
IF (J - I .EQ. 0) GO TO 100
IF (B(J, I) .EQ. 0.) GO TO 100
X(J) = X(J) - B(J, I) * X(I)
TEMP = B(J, I)
DO 90 N = 1, KC
AINV(J, N) = AINV(J, N) - TEMP * AINV(I, N)
90 B(J, N) = B(J, N) - TEMP * B(I, N)
100 CONTINUE
110 CONTINUE
RETURN
120 WRITE( 6, 130)
IERR = 0
RETURN
130 FORMAT (6X, 'THE MATRIX IS SINGULAR')
END
*DEC OPT
SUBROUTINE TBLN (Y,X,T,A,NT,M)
\* THIS SUBROUTINE IS A TABLE LOOKUP FROM ABSCISSA TABLE 'T'
AND ORDNATE TABLE'A'. 'Y' IS ORDNATE AT GIVEN ABSCISSA 'X'.
\* 'NT' IS LENGTH OF TABLES'T'AND'A'. 'M' IS LOCATION OF LAST VALUE
DIMENSION T(1),A(1)
10 IF (T(M)-X) 50,20,30
20 Y=A(M)
RETURN
30 IF (T(1)-X.LT.0.) GO TO 40
   M=1
   GO TO 20
40 M=M-1
   GO TO 10
50 MM=M+1
   IF (MM-NT.LE.0) GO TO 60
   M=NT
   GO TO 20
60 IF (T(MM)-X.GT.0.) GO TO 70
   M=MM
   GO TO 50
70 M=MM-1
   DT=T(MM)-T(M)
   IF (DT.NE.0.) GO TO 80
   Y=A(M)
   RETURN
80 DY=A(MM)-A(M)
   DDT=X-T(M)
   Y=A(M)+DY*DDT/DT
   RETURN
END
SUBROUTINE TRESP(A,B,T,Y,Z,NP,N,K,NINT)
C THIS SUBROUTINE PERFORMS A RUNGE-KUTTA INTEGRATION OF
C A SET OF (N) LINEAR FIRST ORDER DIFFERENTIAL EQUATIONS.
C THE OUTPUT IS THE (K)TH STATE VARIABLE.
STP=NINT
Z(1)=Y(1)
M=1
DO 1 J=1,3
XDOT(J)=0.
1 X(J)=0.
DO 20 J=2,NP
TA=T(J-1)
DT=T(J)-TA
H=DT/STP
DO 10 I=1,NINT
II=0
5 CALL RUNGE(N,XDOT,H,TA,X,L,II)
IF(L.EQ.2) GO TO 10
CALL TBLN(U,TA,T,Y,NP,M)
CALL YDOT(A,X,XDOT,B,U,N)
GO TO 5
10 CONTINUE
Z(J)=X(K)
20 CONTINUE
RETURN
END
*DECK XMULT
SUBROUTINE XMULT(A,B,C,N)  
C THIS SUBROUTINE COMPUTES THE PRODUCT OF TWO MATRICES A AND B  
C HAVING DIMENSIONS N BY N. THE RESULT IS C  
DIMENSION A(4,4),B(4,4),C(4,4)  
C SET WORKING MATRIX ELEMENTS TO ZERO  
DO 10 J=1,N  
DO 10 K=1,N  
10 C(J,K)=0.  
DO 20 J=1,N  
DO 20 K=1,N  
DO 20 JK=1,N  
20 C(J,K)=C(J,K)+A(J,JK)*B(JK,K)  
RETURN  
END

*DECK YDOT
SUBROUTINE YDOT(A,Y,XDOT,B,U,N)  
C THIS SUBROUTINE UPDATES THE STATE VARIABLE EQUATIONS FOR  
C THE TRESP RUNGE-KUTTA INTEGRATION SUBROUTINE.  
DIMENSION Y(3),A(3,3),XDOT(3),B(3)  
DO 2 I=1,N  
XDOT(I)=0.  
DO 1 J=1,N  
XDOT(I)=XDOT(I)+A(I,J)*Y(J)  
1 CONTINUE  
XDOT(I)=XDOT(I)+B(I)*U  
2 CONTINUE  
RETURN  
END
A FORTRAN IV coded computer program is presented for post-flight analysis of a missile's control surface response. It includes preprocessing of digitized telemetry data for time lags, biases, non-linear calibration changes and filtering. Measurements include autopilot attitude rate and displacement gyro output and four control surface deflections. Simple first order lags are assumed for the pitch, yaw and roll axes of control. Each actuator is also assumed to be represented by a first order lag. Mixing of pitch, yaw and roll commands to four control surfaces is assumed. A pseudo-inverse technique is used to obtain the pitch, yaw and roll components from the four measured deflections.

This program has been used for over 10 years on the NASA/SCOUT launch vehicle for post-flight analysis and was helpful in detecting incipient actuator stall due to excessive hinge moments.

The program is currently set up for a CDC CYBER 175 computer system. It requires 34K words of memory and contains 675 cards. A sample problem presented herein including the optional plotting requires eleven (11) seconds of central processor time.
End of Document