The High Resolution Accelerometer Package (HIRAP) Flight Experiment Summary for the First 10 Flights
The High Resolution Accelerometer Package (HiRAP) Flight Experiment Summary for the First 10 Flights

Robert C. Blanchard
Langley Research Center
Hampton, Virginia

K. T. Larman and M. Barrett
Lockheed Engineering & Sciences Company
Hampton, Virginia
| Appendix H | Source Codes                  | 47 |
| References |                             | 165|
| Tables     |                             | 166|
| Figures    |                             | 167|
Summary

The High Resolution Accelerometer Package (HiRAP) instrument is a triaxial, orthogonal system of gas-damped accelerometers with a resolution of $1 \times 10^{-6}$ (1 $\mu g$). The purpose of HiRAP is to measure the low-frequency component of the total acceleration along the orbiter vehicle (OV) body axes to a resolution of 1 $\mu g$ while the OV descends through the rarefied-flow flight regime. Two HiRAP instruments have flown on a total of 10 Space Transportation System (STS) missions. The aerodynamic component of the acceleration measurements was separated from the total acceleration by a data processing system that included removing OV rotationally induced linear accelerations, reaction control system impulses, effects of orbiter mechanical systems, and instrument bias. Instrument bias and orbiter mechanical system acceleration effects were incorporated into one bulk bias. The rate of change of instrument bias with increasing temperature was evaluated. Both the bulk bias and the trend of increasing bias with temperature were subtracted from the acceleration measurements to produce aerodynamic descent data sets for all 10 flights. This document describes the detailed methods of converting the raw data set into the triaxial reentry aerodynamic acceleration data set. This includes algorithms, discussions of the processes, and plots of the data set. The aerodynamic acceleration data sets were input to an aerodynamic coefficient model. The components of the coefficient model are described in the form of both algorithms and performance envelope plots. The aerodynamic acceleration data and coefficient model are used to estimate the atmospheric density for the altitude range of 140 to 60 km and a downrange distance of about 600 km. A density ratio model is developed and presented to verify the analysis techniques. For 8 of 10 flights results from this model agree with expected results. For the results that do not agree with expected results, instrument malfunction and misalignment, inaccuracies in the processing of the data, and aerodynamic model assumption have been explored as possible sources of errors.

Introduction

The primary aim of the High Resolution Accelerometer Package (HiRAP) experiment is to measure the aerodynamic accelerations along the body axes of the orbiter vehicle (OV) while the OV descends through the rarefied-flow flight regime. These measurements are used to determine the aerodynamic performance coefficients of the OV in an atmospheric region that cannot be duplicated in ground-based research.

This report documents the analysis of in-situ aerodynamic acceleration flight measurements from the first 10 HiRAP missions. This research project is part of the NASA Orbiter Experimental (OEX) Program.

The purpose of this report is to present the 10 aerodynamic acceleration data sets of the HiRAP flight experiment, document the procedure used to produce the reentry aerodynamic accelerations from the HiRAP measurements, and summarize the aerodynamic flight results. Included is a description of the aerodynamic performance model used to compare with the HiRAP measurements in view of anticipated atmospheric density results. This report is intended to serve as a reference document for analysis of future HiRAP and other acceleration experiment flights and therefore includes source code listings, data file names, and constants used in these codes.

Symbols and Abbreviations

- $A_x$, $A_y$, $A_z$: $X$-, $Y$-, and $Z$-body-axis components of acceleration
- $C_a$: axial-force coefficient
- $\overline{C}_a$: normalized axial-force coefficient
- $C_n$: normal-force coefficient
- $\overline{C}_n$: normalized normal-force coefficient
- $d$: orbiter mean chord length, m
- $g$: Earth acceleration of gravity, 9.8 m/sec$^2$
- $L/D$: lift-drag force ratio
- $l$: molecular mean free path constant, m
- $M$: orbiter mass, kg
- $MW_{76}$: molecular weight estimate from 1976 U.S. Standard Atmosphere (ref. 1)
- $N_{Kn}$: Knudsen number
- $p$, $q$, $r$: orbiter pitch, yaw, and roll rates, deg/sec
- $\dot{p}$, $\dot{q}$, $\dot{r}$: orbiter pitch, yaw, and roll rates of change, deg/sec$^2$
- $S$: orbiter reference area, 249.91 m$^2$
The HiRAP accelerometer package was designed to measure high-altitude aerodynamic acceleration on the Space Shuttle orbiter vehicle (OV) during atmospheric reentry. The general approach is to use the HiRAP experiment to measure the accelerations on the OV during the unpowered gliding reentry and descent to estimate the aerodynamic performance coefficients.

During the descent period of each mission, data acquisition begins just prior to the deorbit burn, when the orbiter is at an altitude between 250 to 300 km. Data are obtained until the X- and Z-axis channels become and remain saturated at approximately 95 to 100 km and 80 to 85 km, respectively. HiRAP, therefore, has a limited lower altitude range. The Y-axis channel saturates intermittently as a function of aerodynamic maneuvers. The shuttle inertial measurement unit (IMU) instrument is also a set of triaxial orthogonal accelerometers whose axes are aligned with the OV body axes and is used for shuttle guidance and control. The IMU measurements of acceleration have a resolution in the range of 1000 µg and are used in this analysis when the HiRAP sensors saturate.

To date, the OEX HiRAP project has flown two instrument packages, S/N 001 and S/N 002, on two orbiter vehicles, OV-099 (Challenger) and OV-102 (Columbia). Table 1 lists the 10 STS missions on which the HiRAP instruments have flown and the instrument serial numbers, entry dates, and cross-referencing data file numbers and STS mission numbers to allow correlation between this report, prior HiRAP publications, and Johnson Space Center (JSC) publications. HiRAP instrument S/N 002 was lost with Challenger; OV-099, on flight STS-51L and therefore is not available for additional flights. Figure 2 shows the descent trajectories at altitudes from 160 to 60 km and the dates of each of the HiRAP missions.

Numerous HiRAP measurements have been made and analyses have been completed and documented (refs. 2 to 8). The generalized analysis procedure outlined in this document relies on many of the conclusions of these more specific analyses.
The accelerometer voltages for each channel are digitized and recorded as a function of time in a pulse code modulated (PCM) data stream of 14-bit analog binary data words.

The PCM format is used for all data collected from all OEX experiments. The format can be described as a two-dimensional array of 8-bit words that form one data cycle. (The PCM format is described in a document entitled “ACIP-PCM Data Format Control Document,” revision C of specification 2359217 produced by the Aerospace Systems Division of Bendix Corp., Ann Arbor, Michigan, May 12, 1981.) This data cycle is comprised of the encoded data from all OEX experiments on the data bus. The HiRAP data are subcommutated within this array.

The source code HRPSTRP is used to read the OEX flight tapes. Appendix A contains flight data tape volume serial number (VSN) identifiers and file names. The HRPSTRP code writes a time interface file (TIF) science data file containing HiRAP acceleration data. (A description of the TIF format is given in appendix A of an internal Langley Research Center document by Karen D. Brender entitled “STS Post-flight Output Files,” which was produced in February 1982.) In the TIF format, the header contains the serial number of the file, the number of the data channels, the data label and units of each data channel, and an 80-character title. Each subsequent record contains the HiRAP flight data in the same order as described by the data label and units in the header.

The times of the science measurements on the flight tapes may be skewed, that is, show time reversals or duplications or be unsynchronized with other simultaneously sampled data sets, as a result of anomalies in initial recording quality and merging of the various instrument data time lines. The source code SCIREAD is used to read science data files, remove these time errors, and write correct science data files. These science data files are labeled SCIXXY, where XX refers to data file number (table 1) and Y refers to the segment of the flight. For example, file SCI326 holds science data from segment 6 of the HiRAP measurement data set for flight STS-61C. This labeling convention is used on all data files and source codes of this analysis. The science files that are output from SCIREAD are also formatted in TIF. These files begin with a header followed by four-channel science data records of time, X-axis counts, Z-axis counts, and Y-axis counts.

Although the HiRAP data sets begin at approximately the deorbit burn for each flight, the focus of this analysis is on HiRAP measurements during the reentry and descent portions of the orbiter trajectory. Therefore, the HiRAP measurements used in this analysis begin approximately 2000 sec after the deorbit burn, when any atmospheric effects are first measured, and continue through sensor saturation. The saturation times are tabulated in appendix B. The HiRAP sensor accelerometer count and temperature data are shown for the X-, Y-, and Z-axes in figures 3 to 12 for all 10 flights. In these and subsequent plots the time histories extend to approximately 200 sec after saturation of the X-channel. These plots are used to verify that the data are continuous and exhibit expected characteristics.

**Instrument Sensor Temperature Data**

HiRAP sensor temperature data are time-tagged records of the temperature of each accelerometer during flight. The sensor temperature directly affects the acceleration measurement. These temperature data are used in the determination of the accelerometer bias. A synopsis of the temperature conversion algorithms is that a rough measurement of sensor temperature is first determined from the coarse temperature count. The temperature is then further resolved within 0.06°F using one of the eight ranges of the fine temperature count measurement.

Temperatures are measured at each of the accelerometer sensors by a thermistor. The output from the thermistor is a ±5-V signal that is digitized by an 8-bit analog-to-digital converter (ADC) placed in the PCM data stream. Two temperature ranges are monitored for each accelerometer sensor, fine and coarse.

Coarse and fine temperature count data are measured along with ±5-V power supply voltage as a function of time. The temperature count data rate is 2.7 Hz for all flights except STS-61C, which is recorded at a rate of 1.6 Hz. These temperature and voltage measurements are referred to as housekeeping data.

As with the acceleration data, the time tags of the housekeeping measurements on the flight tapes may also exhibit dropouts or reversals. The source code HSKPRED is used to read the housekeeping data stripped from the flight tapes, remove any time errors, and write housekeeping data files HSKPXXY. (See appendix A for housekeeping data file names.) These housekeeping files are TIF formatted. Each data record consists of a nine-channel record of time, fine and coarse temperature counts for each of the three axes, and measurement of positive and negative power supply voltages.
The source code TCALIB is used to read housekeeping data files HSKPXXXY, convert the coarse and fine temperature counts to degrees Fahrenheit, and write the temperature files TCVXXXY. Appendix C gives the algorithms used to convert the temperature counts to voltages and degrees Fahrenheit.

The temperature of the HiRAP instrument increases with time during orbit and for the early portion of descent prior to convective cooling. The increase in temperature is generally linear with time. When the orbiter has descended to the altitude where cooling by atmospheric venting is effective, the HiRAP temperature stabilizes and then decreases with time until touchdown. Part d of figures 3 to 12 shows plots of temperature versus time for each flight and axis for the concurrent times of the HiRAP acceleration measurements. The temperature histories shown extend only to the time at which temperature first begins to decrease because of convective cooling.

For all flights except STS-09, the temperature histories show an expected continuous linear increase with time. For flight STS-09, figure 6(d) shows an interruption in temperature on the Y-axis sensor temperature profile between 83,000 and 83,100 sec GMT. Figure 6(e) presents the triaxial temperature profiles for flight STS-09 for an earlier time phase that shows that an interruption in temperature occurs on all three axes.

There is no known orbiter event or instrument response that could explain the instantaneous rise in temperature these plots for flight STS-09 display. Therefore, the possibility of errors in processing the temperature count data was investigated.

An examination of the fine and coarse temperature counts (figs. 6(f) and 6(g)) shows that no discontinuity exists in the coarse temperature count profile. The discontinuity in temperature for flight STS-09 is traced to an improper ranging between fine temperature count ranges. This improper ranging does not appear to affect the current calibration of any of the axes of the HiRAP data set because the calibration does not incorporate the discontinuity. The exploration and correction of this problem have been relegated to a future investigation.

Trajectory and Orientation Data

To identify the various effects on the aerodynamic acceleration data sets, acceleration measurements must be correlated with vehicle trajectory and orientation data. The vehicle trajectory and orientation data include orbiter altitude, angle of attack, body flap deflection, elevon deflection, velocity, and ground track. These trajectory data are compiled along with the orbiter control surface data and are written to TIF-formatted files.

Higher altitude trajectory data are recorded on files labeled XBETXX, for Extended Best Estimate Trajectory. Lower altitude trajectory data are recorded on files labeled ABETXX, for Aerodynamic Best Estimate Trajectory. These two data sets overlap to some extent. However, the ABET and XBET are determined independently, which for some flights leads to an altitude discontinuity between the two data sets. The differences for each flight are accounted for in the present analysis.

Altitude and time histories of the angle of attack, body flap angle, and elevon angle are shown on figures 13 to 22 for the altitude regions corresponding to the descent portion of the 10 HiRAP mission trajectories. These figures are used to locate times of orbiter attitude maneuvers, which may correlate with signal changes in the accelerometry data sets. Appendix B lists the GMT times and altitudes of the first point in the ABET and XBET trajectory data sets for each of the 10 orbiter flights analyzed herein.

Data Reduction Procedures

The systems aboard the orbiter vehicle used in the orientation and control of the vehicle during descent produce accelerations on the vehicle. The HiRAP instrument measures these accelerations. The HiRAP instrument measurements also show a bias related to instrument temperature. The following sections describe the procedures to reduce the HiRAP measurements of the orbiter total acceleration along each axis, including the temperature acceleration, to produce the aerodynamic components of the orbiter acceleration.

Reentry Time-Line Events

An initial step in the reduction procedures is to check the acceleration measurements for expected characteristic signals resulting from routine events in the orbiter reentry time line. Possible anomalies in the acceleration histories can be identified by a quick look at the raw accelerometer counts with time (figs. 3 to 12). In addition, instrument power supply voltages and temperature profiles are checked to determine the instrument status.

A listing of time-line events follows. Figure 23 shows X-, Y-, and Z-axis acceleration histories for flight STS-61C, with each of these time-line events and their characteristic signals labeled. It is important to note that these characteristic signals are described in units of $g (1 \times 10^{-6}g)$ in order to provide a quick analogy to the physics of the events producing
the acceleration signals. The procedure of converting the raw measurements from counts to units of $\mu g$ is described in a subsequent section.

Each acceleration history is checked for 10 timeline events as follows:

1. Thermal stabilization after power is supplied to the instrument. The HiRAP sensor requires about 30 minutes after power up before its electronic elements become thermally stabilized (temperature rise with time is linear). Once the instrument is thermally stabilized, each HiRAP sensor indicates a nonzero signal that is a temperature-related bias. The temperature bias value is unique for each sensor and each flight and varies in absolute magnitude from approximately 10 to 2500 $\mu g$. Figure 23 shows the temperature bias after power up for flight STS-61C to be approximately $-1850$, $760$, and $-1740 $ $\mu g$ for the $X$-, $Y$-, and $Z$-axis, respectively. Figure 23 also shows the constant slope of the average acceleration signal over time (until onset of drag and lift, which is discussed later). This slope is a measure of the increase of temperature bias due to the increase of temperature over time.

2. Electronic HiRAP system self check. This appears as a series of symmetric positive and negative impulses following application of power to the instrument. These positive and negative impulses are the responses to a predetermined electronic stimulus and are not a measure of acceleration. The self check signal is visible on all three axes during the same time interval.

3. Ignition of the first auxiliary power unit (APU). This appears only on the $Z$-axis as a positive shift of, on average, about $10 \mu g$. Following the initial jump in acceleration due to the ignition of the first APU, the signal appears as a 1-Hz sine wave with a magnitude of approximately $100 \mu g$.

4. Deorbit burn. This signal appears as a gap in acceleration on the $X$- and $Z$-axes (saturating these two channels) for the duration of the deorbit burn. On the $Y$-axis, the deorbit burn signal appears as a roughly symmetrical but noisy change in the acceleration signal of approximately $\pm 200 \mu g$. The deorbit burn lasts between 160 and 290 sec. For flight STS-61C it is about 232 sec. On all axes these signals appear between the first APU ignition and the reentry pitch maneuver.

5. Pitch maneuver to set the OV reentry attitude. This maneuver results in a step-function-shaped change in acceleration of about $30 \mu g$ on the $X$-axis and $-30 \mu g$ on the $Z$-axis. This signal does not appear on the $Y$-axis. The pitch maneuver occurs about 60 sec after the deorbit burn.

6. Dumping of fuel from the forward RCS pod. This fuel dump results in a step-function-shaped shift of approximately $-600 \mu g$ on the $X$-axis and of approximately $100 \mu g$ on the $Z$-axis. The fuel dump does not impact the $Y$-axis and does not occur on every flight.

7. Ignition of the second and third APU's. This event results in approximately a 50-$\mu g$ shift on the $Z$-axis only. During their operation, the APU's add a noisy low-frequency signal to the HiRAP measurements, with a magnitude ranging between $\pm 300 \mu g$.

8. Onset of atmospheric axial-, normal- and side-force components. As the orbiter descends, atmospheric axial and normal forces produce a steadily increasing magnitude of acceleration measured on the $X$- and the $Z$-axis, respectively. On the $Y$-axis, the large variation in signal magnitude and sign results from a combination of side force and cross-range steering ($\pm 5000 \mu g$).

9. Instrument saturation. When the accelerations exceed $-8000 \mu g$ the $X$- and $Z$-axis channels become saturated. The $Z$-axis channel saturates at an altitude between 110 and 95 km. The $X$-axis channel saturates at between 95 and 80 km. Below these altitudes the $X$- and $Z$-axis channels remain saturated except for an occasional saw tooth-shaped signal resulting from a large control surface change. The $Y$-axis sensor signal ranges between $\pm 8000 \mu g$ during reentry but does not saturate for extended periods of time.

10. Reaction control system (RCS) vernier and primary thruster firing. The activation of these thrusters results in spike-shaped acceleration signals. The magnitude of the acceleration depends on the cant and type of thruster. For the reentry and descent portions of the 10 flights analyzed herein, there is no record of instances of vernier thruster firing. For the purposes of future analysis of flights when vernier thruster firing does occur during reentry and descent, the maximum signal magnitude is expected to be approximately 120 $\mu g$. In the case of the primary thrusters, the maximum signal magnitude is approximately 4000 $\mu g$. The primary thrusters are used to control the orbiter attitude until aerodynamic surfaces become effective. Therefore, primary thruster activation occurs frequently during descent. The signal induced by these thrusters appears as a distinct spike followed by a roughly sinusoidal dampening lasting a few seconds. These signals are a smaller percentage of the total signal as the magnitude of the acceleration due to lift and drag increases. These
signals are not shown in figure 23 but are presented subsequently.

Appendix B lists the GMT and altitude of the orbiter at the times of the APU shift, the deorbit burn, the pitch maneuver, and the X- and Z-axis saturations for each flight.

**Corrections Applied to the Acceleration Measurements**

All the HiRAP data sets had to be corrected to account for the nonaerodynamic signals measured by the HiRAP. Nonaerodynamic acceleration measurements include the electronic self check, RCS thruster firings, APU operation, and linear accelerations induced by orbiter rotational motion. Although crew motions and operation of onboard machinery produce accelerations that are measurable by HiRAP, no time line is available of crew motions or machinery operation (exclusive of the APU’s). However, because the crew are strapped into their seats during reentry, their motion-induced accelerations should be negligible. Therefore, it is assumed in this analysis that the vector sum acceleration of all crew activities and machinery other than the APU’s onboard the orbiter is random.

The HiRAP instrument measurements include an acceleration bias that depends on temperature. This temperature bias is evident in the average nonzero acceleration level measured by the instrument after the instrument has thermally stabilized at an altitude region of little or no aerodynamic acceleration. As temperature increases steadily during most of the descent portion of flight, this temperature-induced bias also increases. This change of bias with temperature is referred to as the bias slope. The temperature bias and bias slope must be removed from the acceleration measurements. The following sections detail the procedure of accounting for any nonaerodynamic signals, temperature bias, and bias slope in the HiRAP acceleration measurement data sets.

**Removal of thruster effects.** The reaction control system (RCS) thrusters provide attitude control for the OV at or near orbital altitudes and during the early portion of descent where control surfaces are ineffective. The RCS is composed of 38 primary thrusters and 6 vernier thrusters, which are grouped in three locations on the orbiter. One RCS thruster group is in the forward nose section and the other two are located on the left and right aft thruster pods. When the primary thrusters are activated, the resulting acceleration signals vary in magnitude up to approximately 4000 μg. The resulting signal can be greater when several thrusters fire simultaneously or less when only a thrust component is measured. When activated, each thruster fires in bursts of 80 msec separated by gaps of 80 msec.

It is not practical to separate the effect of each RCS thruster firing from the aerodynamic signal because the magnitude of the acceleration signal of each thruster can vary from one occurrence to another. Thus, sections of acceleration measurements that occur during the thruster firing must be removed from the measurement data set. During each flight, the thruster firing histories are recorded on the OEX flight tape. By reading the times of the thruster firings from the OEX flight tape, the thrust component acceleration measurements can be identified and removed.

Source code ZPRESS reads the RCS chamber pressures from data tapes JHXX and outputs the number of occurrences of firing for each thruster and the reference pressure of each firing. The minimum reference pressure of each thruster is identified. This is called the zero reference pressure. Source code THRUST reads the chamber pressures and removes X- and Z-axis acceleration measurements that occur when any chamber pressure exceeds its zero reference pressure. Source code GPREMXX removes Y-axis acceleration during periods when thruster pressure exceeds its respective zero reference value. Refer to appendix D for the VSN identifier of the RCS chamber pressure tapes found in the tape library.

In addition, the interval of RCS activity is expanded to compensate for synchronizing errors that result in differences between the acceleration response and the thruster chamber pressure readings. This results in a lag of up to 1 sec between the thruster firing time and measured discrete acceleration. Within the source code THRUST, this time difference is accounted for by decreasing the initial thrust firing time by a lag time called TLAG. Therefore, the interval of data to be removed starts prior to the time recorded for the thruster firing.

A second expansion accounts for thrust-induced structural ringing. This ringing signal occurs after all chamber pressures have returned to their zero reference values following a firing sequence. Within the THRUST code, this second expansion occurs by increasing the thrust firing interval time by a time called TLAG1.

Often thruster acceleration signals overlap. This leads to a complete masking of the desired aerodynamic acceleration signal because so much of the acceleration data are removed with the thrust spike and thrust ringing. A study was performed to determine
the minimum amount of data to remove while the
thrust ringing and time synchronization problems are
still accounted for. The results are that the value of
TLAG is 0.04 sec for flights STS-06, STS-07, STS-
08, STS-09, STS-41B, STS-41C, STS-51B, and STS-
51F. For flights STS-61A and STS-61C, TLAG is 0.84
and 0.08 sec, respectively. The value of TLAG1 is
0.80 sec for flights STS-06, STS-07, STS-08, STS-
09, STS-41B, STS-41C, STS-51B, and STS-51F. For
flights STS-61A and STS-61C TLAG1 is 0.96 sec.

The acceleration data for the expanded time scale
shown in figure 24 clearly show the thruster firing and
the ringing for flight STS-61C. The thrust signal is
indicated by the large, spike-shaped signal followed by
damping in the X- and Z-axis acceleration
histories. In this case, the Y-axis is not impacted
significantly by the thrust signal, but it is for other
thruster firings.

Figure 25 shows an example of the expanded scale
effects of thrust signal removal for flight STS-61C.
The greater variation of the Z-axis data is due to
the accelerations induced by the APU activity. In
some cases, spike-shaped signals remain in the data
following the thrust removal analysis. The reason for
this is not currently known but may be related to
the quality of the RCS data tapes. These spikes are
removed later.

**Conversion of counts to engineering units.**
The accelerometer count data are converted to engi-
neering units with a temperature-independent scale
factor for flights STS-06, STS-07, STS-08, STS-09,
STS-41B, STS-41C, STS-51B, and STS-61C. In these
cases when scale factors are assumed constant, the
dependent terms for scale factor is 8019 µg. The
value of acceleration after conversion, disregarding
the temperature-dependent terms for scale factor, is
7999 µg. Figures 26 to 35 show the time histories of
reentry and descent acceleration measurements after
the conversion from counts to engineering units and
after thrust spikes have been removed.

The temperature dependency of the scale fac-
tors of the modified instrument S/N 002 was evalu-
ted to determine its effect on the acceleration
measurements. For a typical change in monitor volt-
age over the descent period, the scale factor change
(and subsequently the acceleration change) is approx-
imately 0.25 percent. For example, at an accelerom-
eter reading of 16,383 counts (full scale), the value of
acceleration after conversion with the temperature-
dependent terms for scale factor is 8019 µg. The
value of acceleration after conversion, disregarding
the temperature-dependent terms for scale factor, is
7999 µg. Figures 26 to 35 show the time histories of
reentry and descent acceleration measurements after
the conversion from counts to engineering units and
after thrust spikes have been removed.

**Correction to account for instrument offset
from center of gravity.** The HiRAP instruments
are not mounted at the orbiter centers of gravity. Be-
cause of this offset, HiRAP measures linear acceler-
ations that are induced by orbiter rotational motions.
Once the conversion to engineering units is made,
the X- and Z-axis acceleration histories are corrected
with the program JTRATES to remove these induced
linear accelerations. This procedure does not include
removing induced accelerations from the Y-axis ac-
celeration histories because the offset of the Y-axis
sensor from the center of gravity is so small that the
error due to induced accelerations on the Y-axis is
insignificant.

The induced accelerations are calculated with the
distance between the accelerometer mounting loca-
tions and the flight-dependent location of the cen-
ter of gravity at approximately 122,000 m (entry
interface). Center-of-gravity locations and reentry
OV mass values are tabulated in appendix E for all
10 flights. The XBETXX files hold orbiter rotational
rates and rates of change. These files are input to
program JTRATES along with files NTCXX (or, in
the cases when a temperature-dependent scale fac-
tor is used in the conversion process, files MGXX).
Program JRATES reads the 1-Hz rotational rates,
calculates the resulting induced accelerations, inter-
polates the induced accelerations to the HiRAP data
rate, and subtracts the induced linear accelerations
from the HiRAP measurements. The corrected ac-
celerometer data are written to file CGXX.
The complete induced acceleration matrix is as follows:

\[
\begin{bmatrix}
\Delta A_x \\
\Delta A_y \\
\Delta A_z
\end{bmatrix} =
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
\begin{bmatrix}
-(q^2 + r^2)(pq - \dot{r})(pr + \dot{q}) \\
(pq + \dot{r}) - (p^2 + r^2)(qr - \dot{p}) \\
(pr - \dot{q})(qr + \dot{p}) - (p^2 + q^2)
\end{bmatrix}
\]

where

- \( \Delta A_x \) induced linear acceleration along X-axis
- \( \Delta A_y \) induced linear acceleration along Y-axis
- \( \Delta A_z \) induced linear acceleration along Z-axis
- \( X \) distance along X-axis of HiRAP to orbiter center of gravity
- \( Y \) distance along Y-axis of HiRAP to orbiter center of gravity
- \( Z \) distance along Z-axis of HiRAP to orbiter center of gravity
- \( p \) pitch rate
- \( q \) yaw rate
- \( r \) roll rate
- \( \dot{p} \) pitch rate of change
- \( \dot{q} \) yaw rate of change
- \( \dot{r} \) roll rate of change

Noise is introduced to the acceleration data by including rotational rates of change in the calculation of induced linear accelerations. This is due to numerical differentiation of the gyro data. The error in excluding the effect of rotational rates of change is on the order of 1 \( \mu g \) (except for data segments during the deorbit maneuver, which are not analyzed herein). Therefore, after removing the Y-axis correction and the rates of change terms, the algorithm to correct for induced accelerations reduces to

\[
\begin{bmatrix}
\Delta A_x \\
\Delta A_z
\end{bmatrix} =
\begin{bmatrix}
X \\
Z
\end{bmatrix}
\begin{bmatrix}
-(q^2 + r^2)(pq)(pr) \\
(pr)(qr) - (p^2 + q^2)
\end{bmatrix}
\]

With improvement in the resolution of the rotational data, it may be possible to more precisely account for the induced linear accelerations due to rotational rates of change.

It is important to note that the corrected accelerations in the CGXX file begin at the time the XBETXX file begins. This time occurs prior to the aerodynamic region of study, as the XBETXX files begin at approximately deorbit burn. Should higher altitude orbiter angular velocity data be required, the data given by the Aerodynamic Coefficient Identification Package (ACIP) experiment may be used.

**Removal of random data spike.** This analysis is performed to remove from the X- and Z-axis acceleration histories any remaining random data spikes that were not identified on the RCS tapes. The name of the source code used is THFIT, and it removes data from both the X-axis and the Z-axis that exceed a bandwidth around the mean. The magnitude of the bandwidth depends on altitude for the Z-axis. The bandwidth is 75 \( \mu g \) at times prior to the start of the APU’s and 225 \( \mu g \) after this time. For the X-axis, the magnitude of the bandwidth is constant at 45 \( \mu g \). This step of the analysis is not done for the Y-axis because of the highly variable nature of acceleration along this axis.

The input to source code THFIT is file CGXX. The output files are FITXX(X) and FITXX(Z) for the X- and the Z-axis, respectively. These files hold data from which all random data spikes have been removed. Code RCOMBIN reads files FITXX(X) and FITXX(Z) and writes the recombined X- and Z-axis data in file FITXX. Figures 26 to 35 show the acceleration data after the data spikes have been removed.

**Filling in data gaps.** Data gaps created by the thrust and spike removal processes are filled so that acceleration histories are continuous with time. Source code FILLSQR is used for the X- and Z-axis accelerations and source code FILLCDE is used for the Y-axis acceleration. Both programs calculate fill data in a similar procedure. First, the mean, slope, and standard error \( \sigma \) of data adjacent to gaps are evaluated with linear regression. Then data that fall outside \( \pm 3\sigma \) of the line are culled. The standard error of the remaining data is then evaluated, and data outside the \( \pm 3\sigma \) fit found by the regression are again removed. The fill data are then calculated to replicate the standard error, slope, and mean of these remaining data. The fill data then replace all points missing because of thrust removal. Plots of the resulting X-, Y-, and Z-axis acceleration data are shown in figures 36 to 45. Comparing these filled data sets with the FITXX files in figures 26 to 34 shows how the time continuity is maintained without significantly altering the aerodynamic acceleration data set.
Effect on instrument measurements due to misalignment within preflight tolerances.
Alignment of the internally orthogonal HiRAP axes relative to the orbiter body axes is checked when the instrument is installed. For each HiRAP installation on the OV, the alignment check indicated the HiRAP axes were aligned within the preflight tolerance of 5 arc minutes.

The error due to misalignment of the HiRAP instrument when alignment is within tolerances would be greatest on the X- and Y-axes. This is because of larger forces on the Z-axis. To illustrate, the corrected acceleration along the X-axis \(A_{x,c}\) can be represented as a function of the measured acceleration in the X- and Z-directions as follows:

\[ A_{x,c} = A_{x,m} \cos \theta + A_{z,m} \sin \theta \]

where

- \(A_{x,c}\) corrected acceleration on X-axis
- \(A_{x,m}\) HiRAP X-axis channel acceleration measurement
- \(A_{z,m}\) HiRAP Z-axis channel acceleration measurement
- \(\theta\) misalignment angle, deg

For \(\theta = 5\) arc minutes, at an altitude of about 100 km, the above relation gives the error due to misalignment to be approximately 1.5 percent of the measured force along the X-axis. This is equivalent to an absolute error of approximately 5 \(\mu g\). As there is no information to define the alignment angles to an accuracy greater than the alignment tolerances, no attempt is made to account for any misalignment within the preflight tolerances.

Calculating temperature bias, bias slope, and APU effects. The temperature bias of each sensor was evaluated for a large range of temperatures and varying temperature rates of change in the laboratory before installation and after any modifications. The previous baseline analysis of HiRAP flight data used results of these laboratory calibrations to calculate temperature bias and bias slopes. (See appendix F for an explanation of the ground calibration procedure.) However, the laboratory calibrations consist of a limited set of accelerometer data performed in a laboratory environment of \(1g\) and do not simulate the acceleration environment of reentry. As a result, it was decided to evaluate temperature bias and bias slope for each flight using acceleration measurements. In the free-molecular-flow flight regime (above an altitude of approximately 160 km), the aerodynamic accelerations on the orbiter are less than 1 \(\mu g\). Therefore, if all thruster, APU, and other orbiter environmental effects can be accounted for, the bias and bias slope of the HiRAP measurements due to temperature change can be evaluated.

Laboratory results show that for HiRAP S/N 001, the bias slope changes with temperature. For the full range of temperatures in the laboratory test (from 30° to 120°F), bias slope can change by up to 40 percent (ref. 9). However, for instrument S/N 001, laboratory results show that for the more limited temperature changes during descent (approximately 4°F change), the effect of neglecting the change in bias slope in the calibration of the acceleration data is on the order of 1 \(\mu g\). Therefore, in this analysis, for each flight, bias slope is assumed to be constant over the period of descent.

The three APU’s on the orbiter are started after the deorbit burn but prior to occurrence of atmospheric effects. The APU’s idle in standby, prepared to provide power for the hydraulic flight control systems during reentry. The exhaust from each APU produces an acceleration signal at a frequency of about 1 Hz. This exhaust signal strongly impacts acceleration along the Z-axis because of the alignment of the APU exhaust ports. The exhaust signal varies over less than 1 sec because the fuel feed is pulsed during operation. The magnitude of the signal varies from \(\pm 300\) \(\mu g\) on the Z-axis. The average magnitude of the signal over time scales greater than 2 sec is approximately constant. Therefore, for the purposes of this analysis, the APU signal is treated as a bias in the acceleration measurements.

There is some error associated with treating the APU signal as a bias. This error is due to the asymmetry of the APU signal. The APU signal is greatest in the negative direction on the Z-axis. The extent of asymmetry was calculated for one flight. In this case, the arithmetic difference between the 1-sec mean and the 1500-point median of a segment of Z-axis data (after APU start and before reentry) is approximately 10 \(\mu g\). Until more accurate methods of removing the effect of the APU are developed, the value of the bias will be in error by approximately this amount.

The three APU’s are running after thermal gradients of the HiRAP have stabilized. Thus, a bulk bias, made up of APU bias and temperature bias, can be evaluated from HiRAP measurements when all APU’s are running. This bulk bias must also be evaluated before the HiRAP measures aerodynamic accelerations.
The start time of the calibration period is set for at least 10 sec after all APU's have been started. The total length of time for the calibration period is chosen to maximize the amount of data and thus ensure the APU bias will be approximately constant. The end times of the calibration period for each flight are adjusted so that the total calibration period lasts 400 sec but excludes data containing atmospheric effects.

The acceleration data sets used to evaluate the bulk bias and bias slope are data sets for which thruster effects and induced linear accelerations have been removed. With the use of these corrected measurements, data that exceed the mean by 3σ are removed. The bulk bias is calculated by evaluating the resulting mean value of acceleration for each axis. The bias slope is calculated by evaluating the change of the mean value of acceleration with temperature over the 400-sec calibration period. This method was applied to all 10 flights.

An alternate method is used to evaluate the bulk bias. This method starts with acceleration data sets that still contain RCS thrust. To eliminate the RCS thrust signal, only those data points that vary in magnitude from adjacent points by greater than the exhaust thrust signal from one APU are removed. Also, to account for lead and lag times to thruster activity, data just prior to and following these periods are also removed. The remaining data are fit to a line and the intercept is calculated. The biases calculated with this method are within 4 of the biases calculated with the first method for RCS thrust removal.

Appendix G lists the bias and bias slope results found with both the thrust removal method and the alternate statistical method for the Z-axis. The comparative values given by the ground calibration are also shown. The biases and bias slopes for the X- and the Y-axis are calculated only with the thrust removal method.

The method used to evaluate bulk bias that does not require processing of RCS thrust data is approximately as accurate as the method that does. In the event that thrust data are not available at the time HiRAP acceleration data sets are processed, this method provides a reasonably valid means of evaluating the bulk bias and bias slope.

Subtracting bulk bias and bias slope to produce aerodynamic accelerations. The bulk bias for each axis is subtracted from the acceleration data starting at the beginning of the 400-sec calibration period and continuing to saturation or, in the case of the Y-axis, to the time when the sensor temperature begins to decrease. The bias proportional to increasing temperatures is then subtracted from the acceleration data. The resulting acceleration data are considered to be the best measurement of the aerodynamic component accelerations of the orbiter during reentry.

These full rate acceleration histories are averaged over 1 sec for use in another phase of analysis. Source code TYMAVG is used to average X- and Z-axis data, while source code INTTIM is used to average Y-axis data. Figures 46 to 55 show the 1-sec averaged reentry aerodynamic acceleration data sets for each flight and axis.

The source codes used in the data reduction discussed herein are given in appendix H.

Analysis of Aerodynamic Acceleration Measurements

The HiRAP acceleration data set has been modified to remove or account for all recognized non-aerodynamic forces. As these data represent only the aerodynamic forces on the OV during unpow ered reentry through the rarefied-flow regime, performance and state analyses may be performed. Two of these analyses, characterization of the aerodynamic force coefficients and an estimation of the atmospheric density, are performed and the results are presented in this document.

Aerodynamic Coefficient Analysis

The reentry aerodynamic acceleration measurements represent the atmospheric effects on the orbiter as it descends through varying regions of flow conditions. Wind tunnel tests are used to provide estimates of orbiter aerodynamic coefficients at lower altitudes approaching the hypersonic-continuum-flow regime (less than about 60 km). Computer simulations are used to estimate OV aerodynamic performance coefficients at orbital altitudes in the free-molecular-flow regime (greater than about 160 km).

A previous analysis (ref. 5) used HiRAP flight measurements to develop an aerodynamic model that provides estimates of the orbiter aerodynamic coefficients in the transitional-flow regime between the continuum- and free-molecular-flow regimes:

\[ \hat{C}_a = \exp \left[ -A \left( B - \log_{10} N_{Kn} \right)^C \right] \]  
\[ \hat{C}_n = \exp \left[ -D \left( E - \log_{10} N_{Kn} \right)^F \right] \]

where

- \( \hat{C}_a \) normalized axial-force coefficient
\( \bar{C}_n \)  
normalized normal-force coefficient

\[
A = 0.2262
\]

\[
B = 1.2042
\]

\[
C = 1.8410
\]

\[
D = 0.2998
\]

\[
E = 1.3849
\]

\[
F = 1.7120
\]

This model uses input axial and normal components of aerodynamic acceleration to calculate the aerodynamic coefficients of the orbiter along its descent path. It includes the effects of orbiter attitude and control surfaces. The results of the model are compared with the expected results as a measure of the accuracy of the input accelerations.

The HiRAP X- and Z-axis aerodynamic acceleration histories presented herein are input to the model, along with orbiter orientation and control surface deflections. The model is run for all altitudes between the highest altitude for which atmospheric effects are sensed by the HiRAP instrument and 60 km. For the purposes of this analysis, the highest altitude is the altitude at which simultaneous 1-sec averages of the X- and Z-axis accelerations are negative (an indication of atmospheric drag).

The following sections describe how the model works and present the results from the model used with the aerodynamic acceleration histories presented in this report.

**Inputs to the Aerodynamic Model**

To create continuous aerodynamic acceleration histories for altitudes above 60 km, where the HiRAP instrument saturates, accelerations measured by the IMU's (ref. 10) are used. The IMU-derived accelerations are at a 1-Hz data rate. Source code MERG replaces saturation values of 1-sec averages of the HiRAP aerodynamic accelerations with the IMU-derived accelerations. The XBETXX and ABETXX are input files to MERG. The result is a continuous record of the X- and Z-axis accelerations with simultaneous velocity, attitude, and control surface deflection data written to file HKDATXX.

**The Aerodynamic Coefficient Model**

Source code MTEST88 contains the algorithms of the aerodynamic coefficient model. The model provides parameterizations of the axial and normal coefficients of the orbiter as functions of Knudsen number \( Kn \) as shown in equations (1) and (2).

Figure 56 shows a plot of the normalized coefficients \( \bar{C}_a \) and \( \bar{C}_n \) as functions of Knudsen number. The values of \( C_a \) and \( C_n \) can be calculated from the normalized values given by equations (1) and (2) and the values of these coefficients in the free-molecular- and continuum-flow regions, as shown below:

\[
C_a = C_{a,c} + (C_{a,f} - C_{a,c}) \bar{C}_a
\]

\[
C_n = C_{n,c} + (C_{n,f} - C_{n,c}) \bar{C}_n
\]

where \( c \) refers to the continuum-flow coefficient value and \( f \) refers to the free-molecular-flow coefficient value. The continuum- and free-molecular-flow coefficient values are functions of angle of attack, body flap, and elevon. The functions that define the changes of these coefficients with control surfaces are compiled from the results of a previous analysis of HiRAP flight \( L/D \) measurements (ref. 5) and from the L-7 Orbiter Aerodynamic Data Book (OADB, ref. 11). Figures 57 to 59 show the hypersonic-continuum-flow value for the OV normal- and axial-force coefficients with angle of attack, body flap, and elevon.

Before equations (1) or (2) can be evaluated, the Knudsen number must be known. Knudsen number and atmospheric density are related by

\[
N_{Kn} = \frac{(MW_{76} \rho d)}{\rho d} \quad (3)
\]

where

- \( MW_{76} \) mean molecular weight estimate from 1976 U.S. Standard Atmosphere (ref. 1)
- \( l \) molecular mean free path constant
- \( d \) mean chord of orbiter
- \( \rho \) atmospheric density

As there is no measurement of density along the descent path, density must be implicitly derived with an iterative procedure. The MTEST88 program solves for a value of Knudsen number that satisfies

\[
C_{i,model} - C_{i,m} = 0 \pm 0.001 \quad (4)
\]

where \( i \) represents axial or normal coefficient.

The definition of the measured aerodynamic coefficients \( C_{i,m} \) is

\[
C_{i,m} = A_{i,m} \left( \frac{1}{\rho V^2 S} \right)^{-1} \quad (5)
\]

where

- \( C_{i,m} \) axial or normal coefficient
$A_{i,m}$ 1-sec average of measured axial or normal acceleration

$S$ orbiter reference area

$M$ orbiter mass

$V$ orbiter velocity

These definitions show that the accuracy of the result for a density that satisfies equation (4) depends partly on the accuracy of the measured axial or normal accelerations $A_{i,m}$ for a given aerodynamic model.

**Atmospheric Density Analysis**

An initial value of density is required to start the iteration. The initial density estimate is calculated by

$$\rho_i = A_{z,m} \left( \frac{1}{2} V^2 \frac{S}{M} C_n \right)^{-1}$$

where $\rho_i$ is the initial value of density, $C_n$ is the average of the OADB free-molecular-flow and continuum-flow values of $C_n$, and $A_{z,m}$ is the normal acceleration measurement. Because $C_n$ varies only about 17 percent in the transition from the free-molecular-flow regime to the continuum-flow regime, this initial estimate has an error of about 8.5 percent.

The program first converges on a value of density using normal acceleration. To start the iteration, $C_n$ is calculated from the estimate of density and equation (4) is evaluated. For each cycle of the iteration procedure, the program changes the estimate of density by increments. These increments are determined by the Newton-Raphson method and are proportional to the difference between $C_n$ and $C_{n,m}$, where $C_{n,m}$ is a coefficient formed by the measurement of normal acceleration and the current iterated value of density. The iteration continues until the difference between consecutive density estimates is less than 0.1 percent (indicating a satisfactory solution has been found). The program repeats the above procedure to converge on a value of density using normal acceleration (i.e., the program converges on a value of density that satisfies the relation in eq. (4)).

Because the axial coefficient varies by approximately 100 percent between the free-molecular-flow and the hypersonic-continuum-flow regime, the initial density estimate used in the axial density calculation is the same as that in the normal acceleration iteration procedure.

**Summary of Atmospheric Density Analyses**

The MTEST88 program calculates a density derived from normal accelerations, and a density derived from axial accelerations, for each 1-sec average of the reentry and descent acceleration histories used in the aerodynamic analysis. The expected result is that these densities derived from separate measurements are equal.

Parts a of figures 60 to 69 show profiles of the ratio of the density derived from the normal acceleration to the density derived from the axial acceleration. The expected result is that density ratio profiles vary less than 1 percent in the altitude region of 60 to 120 km. Within this region, variations of greater magnitude are expected to occur, but these occurrences should generally be short-term. The density ratio profile at altitudes above 120 km is expected to show greater variations because of the varying APU signal.

For 8 of the 10 flights, density ratio results match expected results. However, for flights STS-51F and STS-61A, the density derived from the normal acceleration differs from the density derived from the axial acceleration by more than 15 percent for an extended portion of the profile (at altitudes of 95 to 110 km).

For the eight flights for which density ratio results do not match expected results, the density profile results are compared with the 1976 U.S. Standard Atmosphere (ref. 1) density profiles. These results are shown in parts b of figures 60 to 69, where calculated density is normalized against the 1976 U.S. Standard Atmosphere value. In this comparison, the density used is derived from the HiRAP axial acceleration measurements from the highest altitude of the aerodynamic analysis to that altitude at which the HiRAP axial channel saturates. Below this saturation altitude, the density profiles are derived with IMU normal axis acceleration measurements. For these flights, the calculated densities differ from those of the 1976 U.S. Standard Atmosphere by -50 to 20 percent at higher altitudes. These variations may in part be due to the origin of the Standard Atmosphere assumptions, particularly the uncertainties at high altitudes.

Density ratio results of flights STS-51F and STS-61A indicate the possibility of errors in the aerodynamic component accelerations. Also, density ratio results for these flights could indicate possible errors either in the parameterizations of the aerodynamic coefficients in the transition-flow regime or in the assumptions of atmospheric state in the iteration procedure. Each of these areas was investigated and the results are presented below.

**Possible Error Sources in Flights STS-51F and STS-61A Component Accelerations**

Errors in the density ratio results of the MTEST88 program occur if the $C_{i,m}$ parameters of
equation (4) are inaccurate. The definition of \( C_{i,m} \) given by equation (5) shows that the accuracy of this parameter is directly dependent on the measured aerodynamic acceleration components \( A_{i,m} \).

Errors in the measurement or processing of the aerodynamic acceleration data sets could occur at a number of the stages in the experiment and in the analysis. Sensor malfunction seems to be a probable source of error because flights STS-51F and STS-61A were both instrumented with the modified version of HiRAP S/N 002. For example, the instrument on these two flights had unique characteristics associated with alignment at installation, sensor range, scale factor, and instrument performance. In the processing of the data sets, errors in the calculation of the bias and bias slopes would produce errors in the results. Each of these error sources was investigated and the results are described in the following paragraphs.

**Alignment at installation.** If the HiRAP instrument is misaligned with the orbiter body axes at installation or knocked from its original alignment later, its measurements will not be representative of accelerations along the orbiter body axes. In this case, if we assume the IMU instrument is aligned along the orbiter body axes, simultaneous HiRAP and IMU measurements will differ. To investigate how well HiRAP and IMU measurements agree, the average differences between HiRAP acceleration measurements and IMU acceleration measurements in an altitude region just prior to HiRAP saturation are evaluated. The average differences for all flights are 171, 391, and 184 \( \mu \)g for the X-, Y-, and Z-axis acceleration, respectively. For flights STS-51F and STS-61A, differences for each axis are less than the average differences calculated with results for all flights. Thus, based upon the agreement between IMU and HiRAP data it appears that misalignment is not a source of error.

However, it was decided to evaluate to what extent compensating for misalignment would affect density ratio results. To do this, various misalignment configurations were modeled and applied to the measured acceleration data \( A_{i,m} \). For \( \theta \) degrees of misalignment in the X-Z plane, the corrected accelerations \( A_{i,c} \) would be

\[
A_{x,c} = A_{x,m} \cos \theta + A_{z,m} \sin \theta
\]

\[
A_{z,c} = A_{z,m} \cos \theta - A_{x,m} \sin \theta
\]

Because the magnitude of the Z-axis signal is approximately 10 times that of the X-axis signal at altitudes of 95 to 110 km, relatively small angles of misalignment would change axial acceleration greatly if some part of the normal signal were impacting the axial measurement. The input data of X- and Z-axis HiRAP accelerations are adjusted to simulate the effect of correcting for misalignment. For a 1° misalignment in the X-Z plane (\( \theta = -1° \)), the results of the density ratio profile are shown in figure 70. These results show much improvement over the original results in the altitude region of 95 to 110 km. However, the average difference between IMU and HiRAP accelerations is recalculated for each axis and is much greater than the difference for the original accelerations. Thus the analysis of alignment errors and their effects on the density ratio results does not resolve the anomaly in the results for flights STS-51F and STS-61A. In addition, the introduction of alignment errors produces an IMU-HiRAP mismatch.

**Sensor range modification.** As part of the measurement range modification to the HiRAP S/N 002 instrument, a large positive bias was applied to the instrument. This results in approximately twice the range capability for the modified S/N 002 than for the S/N 001 or the unmodified S/N 002. However, the results of the laboratory calibration of the modified S/N 002 (ref. 11) present a value for scale factor that is approximately equal to that for the S/N 001 (ref. 10) and for the unmodified S/N 002 (ref. 12). Initially this result was unexpected because of the large differences in range capability.

The laboratory calibrations of the sensors were checked to ensure that an incorrect value of scale factor is not being applied to the measurements. Subsequently it was found that the sensor scale factor does not change because of the range modification (private communication from Doug Thomas, KMS Fusion, Inc., Ann Arbor, Michigan).

Also as part of the modification procedure, scale factor was evaluated as a function of temperature monitor voltage. An incorrectly compensated scale factor of the modified HiRAP S/N 002 in the acceleration data sets was investigated. However, it was found that the temperature dependency of scale factor has no significant impact on the results. The scale factor used for flights STS-51F and STS-61A acceleration data sets does not appear to be in error.

**Faulty instrument operation.** Laboratory calibration results for the unmodified HiRAP S/N 002 show that the instrument failed at certain temperatures. Part of the purpose of modifying the HiRAP S/N 002 is to fix these failure points. Although the laboratory calibration results for the
modified HiRAP S/N 002 instrument do not indicate any instrument malfunction, it is unlikely but possible that a failure could still occur at certain temperatures. If a failure does occur, it could be associated with internal synchronization within the instrument, that is, certain elements of the electronics become out of phase with other component elements during flight (ref. 8). This could result in errors in acceleration on the order of 100 µg, and is most likely to occur in the range of approximately 95°F. From the ignition of the three APU's to landing, sensor temperatures change from 74° to 79°F and from 96° to 102°F for flights STS-51F and STS-61A, respectively. With the loss of HiRAP S/N 002 on Space Shuttle Challenger there is no way to determine if an instrument failure did occur. This remains a possible source of error.

Calibration. The flight post-APU (i.e., after all APU initiations) calibration of the bias and temperature bias slopes of flights STS-51F and STS-61A could be incorrect. These parameters are compared with laboratory results for the modified HiRAP S/N 002 instrument. For the X- and the Z-axis on both flights, the greatest difference between the calculated result and the laboratory result for acceleration bias is approximately 1 percent (or approximately 70 µg). As instrument bias is expected to drift with time, this difference is considered to be within a normal range.

For flights STS-51F and STS-61A, the greatest difference between the calculated result and laboratory result for bias slope occurs for the X-axis for the STS-61A acceleration history and is approximately 20 percent (or approximately 4 µg/°F). The dynamic laboratory calibration of HiRAP S/N 001 (ref. 13) shows that changes of bias slope with temperature of approximately 30 percent occur over the full temperature range of laboratory calibration. However, the only calibration of the modified HiRAP S/N 002 instrument was a static calibration, so that bias slopes for this instrument are available only for a limited number of temperatures. Therefore, as the bias of HiRAP S/N 001 instrument is shown to change by 5 percent in the laboratory calibration, there is no reason to conclude that the calculated bias slope difference of 20 percent from the laboratory calibration value for the modified S/N 002 is abnormal.

As a final check of the calibration of the acceleration histories for flights STS-51F and STS-61A, it was decided to apply the laboratory results for bias and bias slope in the calibration of these data sets to see if the aerodynamic analysis results would improve. However, the density ratio results for both flight STS-51F and flight STS-61A with these recalibrated data sets are very similar to the results with acceleration data calibrated from the post-APU procedure. It should be noted that the post-APU calibration worked on eight flights. Therefore, the post-APU procedure for calibrating the acceleration data sets appears to be acceptable.

Adjustment to scale factor. If the magnitude of axial acceleration were increased and/or the magnitude of the normal acceleration were decreased in the acceleration histories of flights STS-51F and STS-61A, the density ratios would more closely approach 1.0 in this region. To test this, a new set of acceleration histories was generated for both flight STS-51F and flight STS-61A. For the new set, the scale factor used on the X-axis for each flight was decreased by 5 percent over the laboratory value, the result being an increase in X-axis acceleration. Also, the scale factor of the Z-axis was increased by 5 percent, the result being a decrease in Z-axis acceleration. The MTEST88 program was run with the new data sets as input. The density ratio results did improve for each flight. However, the agreement between IMU and HiRAP acceleration measurements is considerably worse than it was before scale factor was changed. Thus an adjustment to scale factor is not an acceptable remedy to the HiRAP acceleration data sets.

Possible Errors in Estimates of Aerodynamic Coefficients

As described in the section explaining the aerodynamic performance model, the purpose of the MTEST88 source code is to converge on a value of density that satisfies equation (4). From this equation, it can be seen that the density results would be in error if the value of $C_i^{model}$ were in error.

The aerodynamic model includes the effects of orbiter attitude changes. However, the model could be in error for only certain attitude configurations. For this case, the error in the results would be limited only to flights during which this attitude occurred.

Flights STS-51F and STS-61A have very similar attitude histories. To determine if the errors in the density ratio results are correlated with attitude, the density ratio results are plotted along with normal coefficient versus altitude for flight STS-61A in figure 71. Any short-term variation of normal coefficient is due to attitude change. From figure 71, there does not appear to be a correlation between the short-term variation in normal coefficient and the 17-percent error in the density ratio results at altitudes of 95 to 110 km. As short-term variation in the
normal coefficient is predominantly due to changes in angle of attack, the error in the density ratio results does not appear to be linked to changes in angle of attack. However, the density ratio results may be linked with other functions of the model, such as the compensation of body flap and elevon. These have not been evaluated.

Possible Errors in Assumptions of Atmospheric State

The MTEST88 program results for density ratio are affected by the assumed molecular weight profile because the value of Knudsen number used in the iteration depends on molecular weight, as shown in equation (3). Presently, the assumed molecular weight profile of the MTEST88 program is the 1976 U.S. Standard Atmosphere (ref. 1) profile for molecular weight. This model atmosphere represents a best estimate of the average atmospheric state over all latitudes, longitudes, and solar activity. Therefore, this model provides a value of atmospheric state as a function of a single variable, altitude.

Below the turbopause, at approximately 90 km, constituents of the atmosphere are completely mixed. Above the turbopause, molecular weight varies with latitude, longitude, and solar activity because the constituents are diffuse enough to react independently to solar activity. Therefore, at any altitude above the turbopause, the actual atmospheric molecular weight at the position of the orbiter trajectory may vary considerably from that value given by the 1976 U.S. Standard Atmosphere. Also, adjustments to the height of the turbopause of up to 20 km from its 1976 U.S. Standard Atmosphere value of 88 km may be realistic.

The impact of changing the assumed molecular weight on the density ratio results of the MTEST88 program was investigated for the results of flight STS-61A. For the alternate profile, the altitude of the turbopause is decreased and the rate at which molecular weight drops off with altitude above the turbopause is increased relative to the 1976 U.S. Standard Atmosphere value. For example, at an altitude of 140 km, the molecular weight given by this alternate profile is approximately 20 percent lower than the 1976 U.S. Standard Atmosphere value. Figure 72 shows density ratio results for flight STS-61A with an alternate molecular weight profile. The density ratio results do show improvement with this alternate molecular weight profile. However, these density ratio results are still not satisfactory, and for further improvement, the molecular weight profile approaches unrealistic values. Therefore, the approach of changing the assumed molecular weight does not appear to resolve density ratio discrepancies.

Concluding Remarks

This report presents the data analysis procedure for obtaining orbiter vehicle (OV) reentry aerodynamic acceleration data sets from High Resolution Accelerometer Package (HiRAP) and inertial measurement unit (IMU) measurements made as the OV descends through the free-molecular-, transition-, and hypersonic-continuum-flow flight regimes. The experimental data, analysis procedure, and results from the first 10 Space Transportation System (STS) HiRAP missions are presented and discussed. The results of the data analysis on the acceleration measurements are presented graphically for each step of the process from raw data to atmospheric density as a function of aerodynamic coefficient component.

The purpose of the data reduction and calibration procedures is to produce aerodynamic acceleration component histories along the OV body axes from the HiRAP and IMU measurements of the total acceleration. The data reduction and calibration procedures include correcting for the effects of orbiter rotationally induced linear accelerations, reaction control system impulses, auxiliary power units, and instrument temperatures. The details of the data calibration and reduction procedures are described in this document, and all source codes, flight parameters, and constants used in the procedures are included.

Results of an aerodynamic analysis using the aerodynamic acceleration components from each of these 10 flights agree with expected results for 8 of the flights. For the two flights for which results do not agree with expected results (STS-51F and STS-61A), possible sources of errors in the measurement and processing of acceleration histories and in the aerodynamic analysis were investigated. The conclusions from this error investigation show that instrument misalignment, calibration scale factor, post auxiliary power unit calibration procedures, and sensor range modification are not responsible for the density ratio discrepancies for these two flights. However, a malfunction of the modified version of the instrument that flew on only these two flights remains a probable source of error.

NASA Langley Research Center
Hampton, VA 23665-5225
January 2, 1992
Appendix A

Summary of Flight Data Files

HiRAP S/N 001

<table>
<thead>
<tr>
<th>OEX FLIGHT TAPES</th>
<th>SDC DATA TAPES</th>
<th>SCIENCE TIMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST4859</td>
<td>NU1174</td>
<td>DAY 94, ASCENT</td>
</tr>
<tr>
<td></td>
<td>SCI061</td>
<td>66301-67660 SEC.</td>
</tr>
<tr>
<td></td>
<td>NU1229</td>
<td>18:25:01-18:47:40</td>
</tr>
<tr>
<td></td>
<td>HSKP061</td>
<td></td>
</tr>
<tr>
<td>ST4875</td>
<td>NE0520</td>
<td>DAY 96, ORBIT</td>
</tr>
<tr>
<td></td>
<td>SCI062</td>
<td>71881-74662 SEC.</td>
</tr>
<tr>
<td></td>
<td>NE0535</td>
<td>19:58:01-20:44:22</td>
</tr>
<tr>
<td></td>
<td>HSKP062</td>
<td></td>
</tr>
<tr>
<td>ST4860</td>
<td>NG0279</td>
<td>DAY 96, ORBIT</td>
</tr>
<tr>
<td></td>
<td>SCI063</td>
<td>72001-74164 SEC.</td>
</tr>
<tr>
<td></td>
<td>NG0280</td>
<td>20:00:01-20:36:04</td>
</tr>
<tr>
<td></td>
<td>HSKP063</td>
<td></td>
</tr>
<tr>
<td>ST4876</td>
<td>NE0609</td>
<td>DAY 98, ORBIT</td>
</tr>
<tr>
<td></td>
<td>SCI064</td>
<td>76561-78734 SEC.</td>
</tr>
<tr>
<td></td>
<td>NE0658</td>
<td>21:16:01-21:52:14</td>
</tr>
<tr>
<td></td>
<td>HSKP064</td>
<td></td>
</tr>
<tr>
<td>ST4853</td>
<td>NG0633</td>
<td>DAY 99, DESCENT</td>
</tr>
<tr>
<td></td>
<td>SCI065</td>
<td>64492-68192 SEC.</td>
</tr>
<tr>
<td></td>
<td>NG1066</td>
<td>17:54:52-18:56:32</td>
</tr>
<tr>
<td></td>
<td>HSKP065</td>
<td></td>
</tr>
</tbody>
</table>

+++

<table>
<thead>
<tr>
<th>DIRECT ACCESS FILE</th>
<th>NUMBER OF POINTS</th>
<th>ACCELEROMETER COUNTS</th>
<th>DIRECT ACCESS FILE</th>
<th>NUMBER OF POINTS</th>
<th>COARSE TEMPERATURE COUNTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCI061</td>
<td>236000</td>
<td>X:0 16383</td>
<td>HSKP061</td>
<td>3689</td>
<td>X:75 104</td>
</tr>
<tr>
<td>ASCENT</td>
<td></td>
<td>Y:0 16383</td>
<td></td>
<td></td>
<td>Y:28 249</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0 16383</td>
<td></td>
<td></td>
<td>Z:77 105</td>
</tr>
<tr>
<td>SCI062</td>
<td>440000</td>
<td>X:0 16017</td>
<td>HSKP062</td>
<td>6870</td>
<td>X:97 242</td>
</tr>
<tr>
<td>ORBIT</td>
<td></td>
<td>Y:38 16322</td>
<td></td>
<td></td>
<td>Y:69 128</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0 14919</td>
<td></td>
<td></td>
<td>Z:97 239</td>
</tr>
<tr>
<td>SCI063</td>
<td>376000</td>
<td>X:0 15787</td>
<td>HSKP063</td>
<td>5869</td>
<td>X:98 242</td>
</tr>
<tr>
<td>ORBIT</td>
<td></td>
<td>Y:94 16256</td>
<td></td>
<td></td>
<td>Y:60 123</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0 14919</td>
<td></td>
<td></td>
<td>Z:97 239</td>
</tr>
<tr>
<td>SCI064</td>
<td>377791</td>
<td>X:5177</td>
<td>9688</td>
<td>HSKP064</td>
<td>5902</td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>------</td>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>Y:3356</td>
<td>16116</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z:1990</td>
<td>16383</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SCI065</th>
<th>643453</th>
<th>X:0</th>
<th>16383</th>
<th>HSKP064</th>
<th>10053</th>
<th>X:5</th>
<th>209</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y:0</td>
<td>16383</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y:6</td>
<td>214</td>
</tr>
<tr>
<td>Z:0</td>
<td>16383</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Z:6</td>
<td>209</td>
</tr>
</tbody>
</table>

**DAY**

**EXTENDED BET SOURCE FILE**

**AEROBET SOURCE FILE**

<table>
<thead>
<tr>
<th>DAY</th>
<th>EXTENDED BET SOURCE FILE</th>
<th>AEROBET SOURCE FILE</th>
</tr>
</thead>
<tbody>
<tr>
<td>99</td>
<td>NC0709</td>
<td>NK0917</td>
</tr>
</tbody>
</table>

**HIRAP S/N 001**

**OEX FLIGHT TAPES**

<table>
<thead>
<tr>
<th>SDC DATA TAPES</th>
<th>SCIENCE</th>
<th>HOUSEKEEP</th>
<th>SCIENCE TIMES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ST5075</td>
<td>1. NA0210 SCI071</td>
<td>NA0216 HSKP071</td>
<td>DAY 169, ASCENT 41041-42362 SEC. 11:24:01-11:46:02</td>
</tr>
<tr>
<td></td>
<td>2. NA0254 SCI072</td>
<td>NA0280 HSKP072</td>
<td>DAY 175, ORBIT 45656-47789 SEC. 12:40:56-13:16:29</td>
</tr>
<tr>
<td></td>
<td>3. ND0379 SCI073</td>
<td>ND0571 HSKP073</td>
<td>DAY 175, DESCENT 47794-50492 SEC. 13:16:34-14:01:32</td>
</tr>
</tbody>
</table>

**DIRECT ACCESS FILE**

<table>
<thead>
<tr>
<th>NUMBER</th>
<th>ACCELEROMETER</th>
<th>DIRECT ACCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIN.</td>
<td>MAX.</td>
<td>MIN.</td>
</tr>
<tr>
<td>POINTS</td>
<td>COUNTS</td>
<td>POINTS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SCI071</th>
<th>224000</th>
<th>X:0</th>
<th>16383</th>
<th>HSKP071</th>
<th>3504</th>
<th>X:85</th>
<th>114</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCENT</td>
<td></td>
<td>Y:0</td>
<td>16383</td>
<td></td>
<td></td>
<td>Y:84</td>
<td>116</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0</td>
<td>16383</td>
<td></td>
<td></td>
<td>Z:87</td>
<td>207</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SCI072</th>
<th>370801</th>
<th>X:0</th>
<th>15825</th>
<th>HSKP072</th>
<th>5793</th>
<th>X:61</th>
<th>177</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORBIT</td>
<td></td>
<td>Y:0</td>
<td>16342</td>
<td></td>
<td></td>
<td>Y:63</td>
<td>230</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0</td>
<td>16383</td>
<td></td>
<td></td>
<td>Z:62</td>
<td>207</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SCI073</th>
<th>468871</th>
<th>X:5627</th>
<th>16383</th>
<th>HSKP073</th>
<th>7328</th>
<th>X:90</th>
<th>104</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESCENT</td>
<td></td>
<td>Y:0</td>
<td>16383</td>
<td></td>
<td></td>
<td>Y:96</td>
<td>228</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0</td>
<td>16383</td>
<td></td>
<td></td>
<td>Z:90</td>
<td>207</td>
</tr>
<tr>
<td>OEX FLIGHT TAPES</td>
<td>SDC DATA TAPES</td>
<td>SCIENCE TIMES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>----------------</td>
<td>---------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ST5237</td>
<td>1. NU0229 SCI081 NU0271 HSKP081</td>
<td>DAY 242, ASCENT 22921-24321 SEC. 06:22:01-06:45:21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. NU0279 SCI082 NU0632 HSKP082</td>
<td>DAY 244, ORBIT 25921-28297 SEC. 07:12:01-07:51:37</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. NU0705 SCI083 NV0276 HSKP083</td>
<td>DAY 248, DESCENT 23581-27765 SEC. 06:33:01-07:42:45</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DIRECT ACCESS FILE</th>
<th>NUMBER OF POINTS</th>
<th>ACCELEROMETER COUNTS</th>
<th>DIRECT ACCESS FILE</th>
<th>NUMBER OF POINTS</th>
<th>COARSE TEMPERATURE COUNTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCI081</td>
<td>238000</td>
<td>X:0 16383</td>
<td>HSKP081</td>
<td>3719</td>
<td>X:80 109</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Y:0 16383</td>
<td></td>
<td></td>
<td>Y:80 110</td>
</tr>
<tr>
<td>ASCENT</td>
<td></td>
<td>Z:0 16383</td>
<td></td>
<td></td>
<td>Z:82 110</td>
</tr>
<tr>
<td>SCI082</td>
<td>405763</td>
<td>X:0 8573</td>
<td>HSKP082</td>
<td>6339</td>
<td>X:76 173</td>
</tr>
<tr>
<td>ORBIT</td>
<td></td>
<td>Y:256 16270</td>
<td></td>
<td></td>
<td>Y:82 125</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0 9142</td>
<td></td>
<td></td>
<td>Z:75 207</td>
</tr>
<tr>
<td>SCI083</td>
<td>723581</td>
<td>X:0 16383</td>
<td>HSKP083</td>
<td>11305</td>
<td>X:28 131</td>
</tr>
<tr>
<td>DESCENT</td>
<td></td>
<td>Y:0 16383</td>
<td></td>
<td></td>
<td>Y:27 116</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0 16383</td>
<td></td>
<td></td>
<td>Z:29 207</td>
</tr>
</tbody>
</table>

---

HiRap S/N 001

HiRap S/N 002 (BEFORE RECALIBRATION)
## OEX Flight Tapes

<table>
<thead>
<tr>
<th>OEX Flight Tapes</th>
<th>SDC Data Tapes</th>
<th>Housekeep</th>
<th>Science Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST5340</td>
<td>1. NT0838 SCI091</td>
<td>NT010 HSKP091</td>
<td>DAY 342, Orbit 9021-10802 SEC. 02:30:21-03:00:02</td>
</tr>
<tr>
<td></td>
<td>2. S092 SCI092</td>
<td>NU0141 HSKP092</td>
<td>DAY 342, Orbit 11701-15012 SEC. 03:15:01-04:10:12</td>
</tr>
<tr>
<td></td>
<td>4. NU0231 SCI094</td>
<td>NU0340 HSKP094</td>
<td>DAY 342, Descent 82381-85802 SEC. 22:53:01-23:50:02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DIRECT ACCESS FILE</th>
<th>NUMBER OF POINTS</th>
<th>ACCELEROMETER COUNTS</th>
<th>DIRECT ACCESS FILE</th>
<th>NUMBER OF POINTS</th>
<th>COARSE TEMPERATURE COUNTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCI091</td>
<td>307374</td>
<td>X:7537 12324</td>
<td>HSKP091</td>
<td>4809</td>
<td>X:42 247</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Y:4594 16383</td>
<td></td>
<td></td>
<td>Y:42 234</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:263 10804</td>
<td></td>
<td></td>
<td>Z:42 237</td>
</tr>
<tr>
<td>SCI092</td>
<td>573000</td>
<td>X:9782 12615</td>
<td>HSKP092</td>
<td>8946</td>
<td>X:99 135</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Y:6348 16383</td>
<td></td>
<td></td>
<td>Y:97 134</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:2055 16383</td>
<td></td>
<td></td>
<td>Z:98 135</td>
</tr>
<tr>
<td>SCI093</td>
<td>171391</td>
<td>X:0 13706</td>
<td>HSKP093</td>
<td>2677</td>
<td>X:91 119</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Y:4413 11841</td>
<td></td>
<td></td>
<td>Y:91 118</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0 11742</td>
<td></td>
<td></td>
<td>Z:91 118</td>
</tr>
<tr>
<td>SCI094</td>
<td>592449</td>
<td>X:0 16383</td>
<td>HSKP094</td>
<td>9258</td>
<td>X:119 146</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Y:0 16383</td>
<td></td>
<td></td>
<td>Y:118 146</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0 16383</td>
<td></td>
<td></td>
<td>Z:118 146</td>
</tr>
</tbody>
</table>

---

**HiRAP S/N 001**

---

**DAY 342**

**Extended BET Source File** NC0709

**Aerobet Source File** NL0701

---

**STS-41B**
OEX FLIGHT TAPES

<table>
<thead>
<tr>
<th>SDC DATA TAPES</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCIENCE</td>
</tr>
<tr>
<td>SCIENCE TIMES</td>
</tr>
</tbody>
</table>

**ST5370**
1. NF0156 NF0158 SCI111 HSKP111
   **DAY 34, ASCENT** 45901-48002 SEC. 12:45:01-13:20:02

**ST5371**
2. NF0203 NF0205 SCI112 HSKP112
   **DAY 37, ORBIT** 53722-55132 SEC. 14:55:22-15:18:52

**ST5372**
3. NF0206 NF0234 SCI113 HSKP113
   **DAY 42, DESCENT** 40501-44512 SEC. 11:15:01-12:21:52

+++++++++++++++++++++++++++++++++++++++++++++++++++++++

**DIRECT ACCESS NUMBER ACCELEROMETER ACCESS OF COUNTS ACCESS OF COUNTS**

<table>
<thead>
<tr>
<th>DIRECT ACCESS</th>
<th>NUMBER OF POINTS</th>
<th>ACCELEROMETER COUNTS</th>
<th>DIRECT ACCESS</th>
<th>NUMBER OF POINTS</th>
<th>COARSE TEMPERATURE COUNTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCESS FILE</td>
<td>MIN. MAX.</td>
<td>X:0 Y:0 Z:0</td>
<td>MIN. MAX.</td>
<td>X:0 Y:0 Z:0</td>
<td>X:0 Y:0 Z:0</td>
</tr>
<tr>
<td>SC111</td>
<td>365000</td>
<td>16383 16383 16383</td>
<td>HSKP111</td>
<td>5705</td>
<td>76 75 78</td>
</tr>
<tr>
<td>ASCENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCI112</td>
<td>240000</td>
<td>1036 16376 16383 Y:12</td>
<td>HSKP112</td>
<td>3823</td>
<td>18 18 20</td>
</tr>
<tr>
<td>ORBIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCI113</td>
<td>697104</td>
<td>28 72 73</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DESCENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DAY 42**

**EXTENDED BET SOURCE FILE** NC0709
**AEROBET SOURCE FILE** NF0349

----------------------------------------------------------------------------------------------------------------------

**STS-41C**

**HiRAP S/N 001**

**OEX FLIGHT TAPES**

<table>
<thead>
<tr>
<th>SDC DATA TAPES</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCIENCE</td>
</tr>
<tr>
<td>SCIENCE TIMES</td>
</tr>
</tbody>
</table>

**ST5630**
1. NS0671 NS0672 SCI131 HSKP131
   **DAY 97, ASCENT** 49801-51422 SEC. 13:50:01-14:17:02
<table>
<thead>
<tr>
<th>DIRECT ACCESS FILE</th>
<th>NUMBER OF POINTS</th>
<th>ACCELEROMETER COUNTS</th>
<th>DIRECT ACCESS FILE</th>
<th>NUMBER OF POINTS</th>
<th>TEMPERATURE COUNTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCI131</td>
<td>282000</td>
<td>X:0 11513</td>
<td>HSKP131</td>
<td>4403</td>
<td>X:75 104</td>
</tr>
<tr>
<td>ASCENT</td>
<td></td>
<td>Y:0 16383</td>
<td></td>
<td></td>
<td>Y:75 107</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0 16383</td>
<td></td>
<td></td>
<td>Z:76 209</td>
</tr>
<tr>
<td>SCI132</td>
<td>516000</td>
<td>X:65 13970</td>
<td>HSKP132</td>
<td>8070</td>
<td>X:71 108</td>
</tr>
<tr>
<td>ORBIT</td>
<td></td>
<td>Y:3 16328</td>
<td></td>
<td></td>
<td>Y:76 114</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0 16383</td>
<td></td>
<td></td>
<td>Z:72 107</td>
</tr>
<tr>
<td>SCI133</td>
<td>24447</td>
<td>X:5286 11950</td>
<td>HSKP133</td>
<td>381</td>
<td>X:49 53</td>
</tr>
<tr>
<td>ORBIT</td>
<td></td>
<td>Y:5070 10962</td>
<td></td>
<td></td>
<td>Y:51 56</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0 12747</td>
<td></td>
<td></td>
<td>Z:50 54</td>
</tr>
<tr>
<td>SCI134</td>
<td>23551</td>
<td>X:0 11282</td>
<td>HSKP134</td>
<td>367</td>
<td>X:91 94</td>
</tr>
<tr>
<td>ORBIT</td>
<td></td>
<td>Y:5252 11046</td>
<td></td>
<td></td>
<td>Y:96 99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:2802 11719</td>
<td></td>
<td></td>
<td>Z:91 94</td>
</tr>
<tr>
<td>SCI135</td>
<td>911020</td>
<td>X:0 16383</td>
<td>HSKP135</td>
<td>14236</td>
<td>X:59 243</td>
</tr>
<tr>
<td>DESCENT</td>
<td></td>
<td>Y:0 16383</td>
<td></td>
<td></td>
<td>Y:61 205</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:1 16383</td>
<td></td>
<td></td>
<td>Z:59 207</td>
</tr>
</tbody>
</table>

DAY 104
EXTENDED BET SOURCE FILE NC0709
AEROBET SOURCE FILE NC0740

HiRAP S/N 001
<table>
<thead>
<tr>
<th>OEX FLIGHT TAPES</th>
<th>SDC DATA TAPES</th>
<th>SCIENCE TIMES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2. NC0533 SCI242</td>
<td>NC0805 HSKP242</td>
</tr>
<tr>
<td>JH5B10</td>
<td>4. ND1219 SCI244</td>
<td>ND1237 SCI244</td>
</tr>
<tr>
<td>JH5B11</td>
<td>5. NK0659 SCI245</td>
<td>NL0257 HSKP245</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DIRECT ACCESS FILE</th>
<th>NUMBER OF POINTS</th>
<th>ACCELEROMETER COUNTS</th>
<th>DIRECT ACCESS FILE</th>
<th>NUMBER OF POINTS</th>
<th>ACCELEROMETER COUNTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCI241</td>
<td>264371</td>
<td>X:0 1040</td>
<td>HSKP241</td>
<td>4130</td>
<td>X:86 117</td>
</tr>
<tr>
<td>ASCENT</td>
<td></td>
<td>Y:0 16383</td>
<td></td>
<td></td>
<td>Y:86 120</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0 16383</td>
<td></td>
<td></td>
<td>Z:88 117</td>
</tr>
<tr>
<td>SCI242</td>
<td>71422</td>
<td>X:6208 11736</td>
<td>HSKP242</td>
<td>1115</td>
<td>X:127 135</td>
</tr>
<tr>
<td>ORBIT</td>
<td></td>
<td>Y:234 11213</td>
<td></td>
<td></td>
<td>Y:131 139</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:570 12244</td>
<td></td>
<td></td>
<td>Z:128 135</td>
</tr>
<tr>
<td>SCI243</td>
<td>81599</td>
<td>X:9020 9142</td>
<td>HSKP243</td>
<td>1274</td>
<td>X:137 142</td>
</tr>
<tr>
<td>ORBIT</td>
<td></td>
<td>Y:8265 8559</td>
<td></td>
<td></td>
<td>Y:143 148</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:9394 9683</td>
<td></td>
<td></td>
<td>Z:136 141</td>
</tr>
<tr>
<td>SCI244</td>
<td>593479</td>
<td>X:0 16383</td>
<td>HSKP244</td>
<td>9412</td>
<td>X:28 85</td>
</tr>
<tr>
<td>DESCENT</td>
<td></td>
<td>Y:0 16383</td>
<td></td>
<td></td>
<td>Y:28 90</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0 16383</td>
<td></td>
<td></td>
<td>Z:30 204</td>
</tr>
<tr>
<td>SCI245</td>
<td>182591</td>
<td>X:16383 16383</td>
<td>HSKP245</td>
<td></td>
<td>X:</td>
</tr>
<tr>
<td>DESCENT</td>
<td></td>
<td>Y:16383</td>
<td></td>
<td></td>
<td>Y:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:16383</td>
<td></td>
<td></td>
<td>Z:</td>
</tr>
<tr>
<td>DAY</td>
<td>EXTENDED BET SOURCE FILE</td>
<td>218</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------</td>
<td>-----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AEROBET SOURCE FILE</td>
<td>NC0709</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**HiRAP S/N 002 (RECALIBRATED)**

<table>
<thead>
<tr>
<th>OEX FLIGHT TAPES</th>
<th>SDC DATA TAPES</th>
<th>SCIENCE TIMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>JH5F14</td>
<td>1. NY0338 SCI261</td>
<td>NY0449 HSKP261</td>
</tr>
<tr>
<td></td>
<td>2. NY0340 SCI262</td>
<td>NY0341 HSKP262</td>
</tr>
<tr>
<td></td>
<td>3. NY0342 SCI263</td>
<td>NY0343 HSKP263</td>
</tr>
<tr>
<td></td>
<td>4. NY0344 SCI264</td>
<td>NY0345 HSKP264</td>
</tr>
</tbody>
</table>

+++ COARSE DIRECT NUMBER ACCELEROMETER ACCESS OF COUNTS FILE POINTS MIN. MAX. ++

<table>
<thead>
<tr>
<th>DIRECT ACCESS FILE</th>
<th>NUMBER OF POINTS</th>
<th>ACCELEROMETER COUNTS</th>
<th>DIRECT ACCESS FILE</th>
<th>NUMBER OF POINTS</th>
<th>COARSE TEMPERATURE COUNTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCI261</td>
<td>241000</td>
<td>X:0 16383</td>
<td>HSKP261</td>
<td>3774</td>
<td>X:96 130</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Y:0 16383</td>
<td></td>
<td></td>
<td>Y:51 129</td>
</tr>
<tr>
<td>ASCENT</td>
<td></td>
<td>Z:0 16383</td>
<td></td>
<td></td>
<td>Z:104 216</td>
</tr>
<tr>
<td>SCI262</td>
<td>110590</td>
<td>X:0 12190</td>
<td>HSKP262</td>
<td>1727</td>
<td>X:108 116</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Y:6845 16064</td>
<td></td>
<td></td>
<td>Y:106 114</td>
</tr>
<tr>
<td>ORBIT</td>
<td></td>
<td>Z:0 10216</td>
<td></td>
<td></td>
<td>Z:107 116</td>
</tr>
<tr>
<td>SCI263</td>
<td>290532</td>
<td>X:0 16094</td>
<td>HSKP263</td>
<td>4550</td>
<td>X:83 100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Y:1552 16383</td>
<td></td>
<td></td>
<td>Y:81 98</td>
</tr>
<tr>
<td>DESCENT</td>
<td></td>
<td>Z:0 15356</td>
<td></td>
<td></td>
<td>Z:82 215</td>
</tr>
<tr>
<td>SCI264</td>
<td>156087</td>
<td>X:0 16383</td>
<td>HSKP264</td>
<td>2454</td>
<td>X:99 107</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Y:0 16383</td>
<td></td>
<td></td>
<td>Y:97 106</td>
</tr>
<tr>
<td>DESCENT</td>
<td></td>
<td>Z:0 16383</td>
<td></td>
<td></td>
<td>Z:98 107</td>
</tr>
</tbody>
</table>
STS-61A

OEX FLIGHT TAPES

<table>
<thead>
<tr>
<th>OEX FLIGHT TAPES</th>
<th>SDC DATA TAPES</th>
<th>SDC DATA TAPES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SCIENCE</td>
<td>HOUSEKEEP</td>
</tr>
<tr>
<td>JH6A15</td>
<td>1. NM1180</td>
<td>NM1267</td>
</tr>
<tr>
<td></td>
<td>SCI301</td>
<td>HSKP301</td>
</tr>
<tr>
<td></td>
<td>DAY 303, ASCENT</td>
<td>60661-64137 SEC.</td>
</tr>
<tr>
<td></td>
<td>16:51:01-17:47:57</td>
<td></td>
</tr>
<tr>
<td>JH6A16</td>
<td>2. NM1271</td>
<td>NN0129</td>
</tr>
<tr>
<td></td>
<td>SCI302</td>
<td>HSKP302</td>
</tr>
<tr>
<td></td>
<td>DAY 303, ORBIT</td>
<td>85021-492 SEC.</td>
</tr>
<tr>
<td></td>
<td>23:37:01-00:08:12</td>
<td></td>
</tr>
<tr>
<td>JH6A17</td>
<td>3. NM1065</td>
<td>NS0628</td>
</tr>
<tr>
<td></td>
<td>SCI303</td>
<td>HSKP303</td>
</tr>
<tr>
<td></td>
<td>DAY 304, ORBIT</td>
<td>23161-25552 SEC.</td>
</tr>
<tr>
<td></td>
<td>06:26:01-07:05:52</td>
<td></td>
</tr>
<tr>
<td>JH6A18</td>
<td>4. NS0818</td>
<td>NS0844</td>
</tr>
<tr>
<td></td>
<td>SCI304</td>
<td>HSKP304</td>
</tr>
<tr>
<td></td>
<td>DAY 309, ORBIT</td>
<td>78301-88027 SEC.</td>
</tr>
<tr>
<td></td>
<td>21:45:01-22:17:52</td>
<td></td>
</tr>
<tr>
<td>JH6A19</td>
<td>5. NA0160</td>
<td>NA0260</td>
</tr>
<tr>
<td></td>
<td>SCI305</td>
<td>HSKP305</td>
</tr>
<tr>
<td></td>
<td>DAY 310, ORBIT</td>
<td>27601-30022 SEC.</td>
</tr>
<tr>
<td></td>
<td>07:40:01-08:20:22</td>
<td></td>
</tr>
<tr>
<td>JH6A20</td>
<td>6. NH1067</td>
<td>NH1155</td>
</tr>
<tr>
<td></td>
<td>SCI306</td>
<td>HSKP306</td>
</tr>
<tr>
<td></td>
<td>DAY 310, DESCENT</td>
<td>59122-62482 SEC.</td>
</tr>
<tr>
<td></td>
<td>16:25:22-17:21:52</td>
<td></td>
</tr>
<tr>
<td>JH6A21</td>
<td>7. NH1215</td>
<td>NM1216</td>
</tr>
<tr>
<td></td>
<td>SCI307</td>
<td>HSKP307</td>
</tr>
<tr>
<td></td>
<td>DAY 310, DESCENT</td>
<td>62461-64132 SEC.</td>
</tr>
<tr>
<td></td>
<td>17:21:01-17:48:52</td>
<td></td>
</tr>
</tbody>
</table>

Axes X and Z are pegged for this entire channel.

+----------------------------------------+----------------------------------------+----------------------------------------+----------------------------------------+----------------------------------------+----------------------------------------+----------------------------------------+----------------------------------------+----------------------------------------+
<p>| DIRECT ACCESS                          | NUMBER OF POINTS                      | ACCELEROMETER COUNTS                  | DIRECT ACCESS                          | NUMBER OF POINTS                      | ACCELEROMETER COUNTS                  | COARSE TEMPERATURE COUNTS          |
| FILE                                   | MIN.                                    | MAX.                                    | FILE                                   | MIN.                                    | MAX.                                    |
| SCI301                                 | X:0                                     | 13556                                   | HSKP301                                | X:79                                     | 138                                     |
| ASCENT                                 | Y:0                                     | 16383                                   |                                         | Y:79                                     | 215                                     |
|                                        | Z:0                                     | 16383                                   |                                         | Z:79                                     | 217                                     |
| SCI302                                 | X:2184                                  | 4253                                    | HSKP302                                | X:143                                    | 178                                     |
| ORBIT                                  | Y:7117                                  | 12832                                   |                                         | Y:142                                    | 186                                     |
|                                        | Z:2087                                  | 4129                                    |                                         | Z:142                                    | 217                                     |</p>
<table>
<thead>
<tr>
<th>SCI303</th>
<th>408000</th>
<th>X:3769 9084</th>
<th>HSKP303</th>
<th>6376</th>
<th>X:249 254</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORBIT</td>
<td></td>
<td>Y:5734 13944</td>
<td></td>
<td>Y:248</td>
<td>254</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:2915 8174</td>
<td></td>
<td>Z:249</td>
<td>254</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SCI304</th>
<th>340991</th>
<th>X:0 3635</th>
<th>HSKP304</th>
<th>5327</th>
<th>X:79 101</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORBIT</td>
<td></td>
<td>Y:2020 12108</td>
<td></td>
<td>Y:77</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:1048 4455</td>
<td></td>
<td>Z:78</td>
<td>101</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SCI305</th>
<th>420799</th>
<th>X:2009 3931</th>
<th>HSKP305</th>
<th>6574</th>
<th>X:102 128</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORBIT</td>
<td></td>
<td>Y:6742 9997</td>
<td></td>
<td>Y:100</td>
<td>127</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:101 128</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SCI306</th>
<th>584239</th>
<th>X:0 16383</th>
<th>HSKP306</th>
<th>9129</th>
<th>X:52 153</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESCENT</td>
<td></td>
<td>Y:0 16383</td>
<td></td>
<td>Y:59</td>
<td>153</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0 16383</td>
<td></td>
<td>Z:121</td>
<td>215</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SCI307</th>
<th>287871</th>
<th>X:0 16383</th>
<th>HSKP307</th>
<th>4497</th>
<th>X:148 154</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESCENT</td>
<td></td>
<td>Y:0 16383</td>
<td></td>
<td>Y:148</td>
<td>153</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:16383 16383</td>
<td></td>
<td>Z:147</td>
<td>154</td>
</tr>
</tbody>
</table>

---

**DAY**

**310**

**EXTENDED BET SOURCE FILE**

NC0709

**AEROBET SOURCE FILE**

NY1721

---

**HiRAP S/N 001**

(RECALIBRATED)

**OEX FLIGHT TAPES**

**SDC DATA TAPES**

**SCIENCE HOUSEKEEP**

**SCIENCE TIMES**

---

Files 32xS and 32xH have no header and are archived on 882817C.

**JHDT01**

1. 321S

   SCI321 HSKP321

   **DAY 15, ORBIT**

   30311-32851 SEC.

   08:25:11-09:07:31

**JHDT02**

2. 322S

   SCI322 HSKP322

   **DAY 16, ORBIT**

   59363-61017 SEC.

   16:29:23-16:56:57

**JHDT03**

3. 323S

   SCI323 HSKP323

   **DAY 17, ORBIT**

   55246-56473 SEC.

   15:20:46-15:41:13

**JH6C13**

4. 324S

   SCI324 HSKP324

   **DAY 18, DESCENT**

   45818-47737 SEC.


---

25
<table>
<thead>
<tr>
<th>DATE</th>
<th>TIME</th>
<th>DAY</th>
<th>ACCESS</th>
<th>COUNTS</th>
<th>MIN.</th>
<th>MAX.</th>
<th>COUNTS</th>
<th>MIN.</th>
<th>MAX.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>DIRECT ACCESS FILE</th>
<th>NUMBER OF POINTS</th>
<th>ACCELEROMETER COUNTS</th>
<th>DIRECT ACCESS FILE</th>
<th>NUMBER OF POINTS</th>
<th>TEMPERATURE COUNTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCI1321</td>
<td>280975</td>
<td>X:6462 16383</td>
<td>HSKP321</td>
<td>4454</td>
<td>X:29 82</td>
</tr>
<tr>
<td>ORBIT</td>
<td></td>
<td>Y:5715 16383</td>
<td></td>
<td></td>
<td>Y:28 88</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0 16383</td>
<td></td>
<td></td>
<td>Z:31 83</td>
</tr>
<tr>
<td>SCI1322</td>
<td>183615</td>
<td>X:0 16383</td>
<td>HSKP322</td>
<td>2910</td>
<td>X:5* 101</td>
</tr>
<tr>
<td>ORBIT</td>
<td></td>
<td>Y:0 16383</td>
<td></td>
<td></td>
<td>Y:5* 108</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0 16383</td>
<td></td>
<td></td>
<td>Z:5* 102</td>
</tr>
<tr>
<td>SCI1323</td>
<td>136995</td>
<td>X:0 16383</td>
<td>HSKP323</td>
<td>2157</td>
<td>X:5* 132</td>
</tr>
<tr>
<td>ORBIT</td>
<td></td>
<td>Y:0 16383</td>
<td></td>
<td></td>
<td>Y:5* 139</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0 16383</td>
<td></td>
<td></td>
<td>Z:5* 132</td>
</tr>
<tr>
<td>SCI1324</td>
<td>213263</td>
<td>X:0 12803</td>
<td>HSKP324</td>
<td>3375</td>
<td>X:144 160</td>
</tr>
<tr>
<td>DESCENT</td>
<td></td>
<td>Y:3334 11835</td>
<td></td>
<td></td>
<td>Y:151 167</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0 16383</td>
<td></td>
<td></td>
<td>Z:145 160</td>
</tr>
<tr>
<td>SCI1325</td>
<td>77000</td>
<td>X:6703 12613</td>
<td>HSKP325</td>
<td>1199</td>
<td>X:159 165</td>
</tr>
<tr>
<td>DESCENT</td>
<td></td>
<td>Y:0 16383</td>
<td></td>
<td></td>
<td>Y:166 172</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0 16383</td>
<td></td>
<td></td>
<td>Z:159 166</td>
</tr>
<tr>
<td>SCI1326</td>
<td>71000</td>
<td>X:8762 16383</td>
<td>HSKP326</td>
<td>1104</td>
<td>X:164 169</td>
</tr>
<tr>
<td>DESCENT</td>
<td></td>
<td>Y:2059 16383</td>
<td></td>
<td></td>
<td>Y:171 175</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z:0 16383</td>
<td></td>
<td></td>
<td>Z:164 170</td>
</tr>
</tbody>
</table>

*HSKP322: THESE LOW COUNTS OCCUR DURING LAST 30 SECONDS OF FILE ONLY. OTHERWISE: XMIN=81, YMIN=87, ZMIN=83 COUNTS.

*HSKP323: LAST 10 SECONDS ONLY. OTHERWISE: XMIN=118, YMIN=124, XMIN=119 COUNTS.
## Appendix B

### Orbiter Descent Event Times

<table>
<thead>
<tr>
<th>Event</th>
<th>Altitude, km</th>
<th>Time, sec</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>STS-06</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time blanked out for APU shift</td>
<td>249 to 239</td>
<td>65,990 to 65,170</td>
</tr>
<tr>
<td>Deorbit burn</td>
<td>249 to 291</td>
<td>64,910 to 64,650</td>
</tr>
<tr>
<td>Pitch maneuver</td>
<td>291 to 284</td>
<td>25 to 65,000</td>
</tr>
<tr>
<td>X-axis saturation</td>
<td>85</td>
<td>66,147</td>
</tr>
<tr>
<td>Z-axis saturation</td>
<td>100</td>
<td>66,314</td>
</tr>
<tr>
<td>XBET epoch</td>
<td>289</td>
<td>64,510</td>
</tr>
<tr>
<td>ABET epoch</td>
<td>123</td>
<td>66,200</td>
</tr>
<tr>
<td><strong>STS-07</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time blanked out for APU shift</td>
<td>253 to 239</td>
<td>47,520 to 47,625</td>
</tr>
<tr>
<td>Deorbit burn</td>
<td>295 to 297</td>
<td>46,560 to 46,730</td>
</tr>
<tr>
<td>Pitch maneuver</td>
<td>298 to 295</td>
<td>46,800 to 47,060</td>
</tr>
<tr>
<td>X-axis saturation</td>
<td>84</td>
<td>48,605</td>
</tr>
<tr>
<td>Z-axis saturation</td>
<td>97</td>
<td>48,520</td>
</tr>
<tr>
<td>XBET epoch</td>
<td>295</td>
<td>46,560</td>
</tr>
<tr>
<td>ABET epoch</td>
<td>208</td>
<td>47,840</td>
</tr>
<tr>
<td><strong>STS-08</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time blanked out for APU shift</td>
<td>210 to 207</td>
<td>25,035 to 25,085</td>
</tr>
<tr>
<td>Deorbit burn</td>
<td>219 to 220</td>
<td>24,440 to 24,610</td>
</tr>
<tr>
<td>Pitch maneuver</td>
<td>220 to 211</td>
<td>24,700 to 25,020</td>
</tr>
<tr>
<td>X-axis saturation</td>
<td>81</td>
<td>26,086</td>
</tr>
<tr>
<td>Z-axis saturation</td>
<td>98</td>
<td>25,888</td>
</tr>
<tr>
<td>XBET epoch</td>
<td>219</td>
<td>24,450</td>
</tr>
<tr>
<td>ABET epoch</td>
<td>187</td>
<td>25,310</td>
</tr>
<tr>
<td><strong>STS-09</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time blanked out for APU shift</td>
<td>217 to 214</td>
<td>82,950 to 82,990</td>
</tr>
<tr>
<td>Deorbit burn</td>
<td>240 to 238</td>
<td>82,320 to 82,480</td>
</tr>
<tr>
<td>Pitch maneuver</td>
<td>236 to 214</td>
<td>82,600 to 82,990</td>
</tr>
<tr>
<td>X-axis saturation</td>
<td>85</td>
<td>84,016</td>
</tr>
<tr>
<td>Z-axis saturation</td>
<td>97</td>
<td>83,935</td>
</tr>
<tr>
<td>XBET epoch</td>
<td>240</td>
<td>82,320</td>
</tr>
<tr>
<td>ABET epoch</td>
<td>161</td>
<td>83,843</td>
</tr>
<tr>
<td><strong>STS-41B</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time blanked out for APU shift</td>
<td>236 to 233</td>
<td>41,534 to 41,559</td>
</tr>
<tr>
<td>Deorbit burn</td>
<td>276 to 277</td>
<td>40,560 to 40,750</td>
</tr>
<tr>
<td>Pitch maneuver</td>
<td>278 to 271</td>
<td>40,850 to 41,120</td>
</tr>
<tr>
<td>X-axis saturation</td>
<td>82</td>
<td>42,574</td>
</tr>
<tr>
<td>Z-axis saturation</td>
<td>97</td>
<td>42,474</td>
</tr>
<tr>
<td>XBET epoch</td>
<td>276</td>
<td>40,570</td>
</tr>
<tr>
<td>ABET epoch</td>
<td>252</td>
<td>41,380</td>
</tr>
<tr>
<td>Event</td>
<td>Altitude, km</td>
<td>Time, sec</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>STS-41C</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time blanked out for APU shift</td>
<td>319 to 313</td>
<td>46495 to 46520</td>
</tr>
<tr>
<td>Deorbit burn</td>
<td>500 to 505</td>
<td>44960 to 45220</td>
</tr>
<tr>
<td>Pitch maneuver</td>
<td>506 to 500</td>
<td>45300 to 45500</td>
</tr>
<tr>
<td>X-axis saturation</td>
<td>90</td>
<td>47480</td>
</tr>
<tr>
<td>Z-axis saturation</td>
<td>99</td>
<td>47396</td>
</tr>
<tr>
<td>XBET epoch</td>
<td>500</td>
<td>44600</td>
</tr>
<tr>
<td>ABET epoch</td>
<td>213</td>
<td>46890</td>
</tr>
<tr>
<td><strong>ST-51B</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time blanked out for APU shift</td>
<td>288 to 260</td>
<td>55575 to 55725</td>
</tr>
<tr>
<td>Deorbit burn</td>
<td>357 to 364</td>
<td>54280 to 54570</td>
</tr>
<tr>
<td>Pitch maneuver</td>
<td>365 to 362</td>
<td>54658 to 54887</td>
</tr>
<tr>
<td>X-axis saturation</td>
<td>85</td>
<td>56627</td>
</tr>
<tr>
<td>Z-axis saturation</td>
<td>98</td>
<td>56540</td>
</tr>
<tr>
<td>XBET epoch</td>
<td>357</td>
<td>54282</td>
</tr>
<tr>
<td>ABET epoch</td>
<td>123</td>
<td>56400</td>
</tr>
<tr>
<td><strong>ST-51F</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time blanked out for APU shift</td>
<td>255 to 250</td>
<td>68525 to 68555</td>
</tr>
<tr>
<td>Deorbit burn</td>
<td>321 to 324</td>
<td>67382 to 67552</td>
</tr>
<tr>
<td>Pitch maneuver</td>
<td>325 to 322</td>
<td>67690 to 67851</td>
</tr>
<tr>
<td>X-axis saturation</td>
<td>79</td>
<td>69553</td>
</tr>
<tr>
<td>Z-axis saturation</td>
<td>92</td>
<td>69440</td>
</tr>
<tr>
<td>XBET epoch</td>
<td>321</td>
<td>67372</td>
</tr>
<tr>
<td>ABET epoch</td>
<td>123</td>
<td>69260</td>
</tr>
<tr>
<td><strong>ST-61A</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time blanked out for APU shift</td>
<td>267 to 264</td>
<td>61220 to 61240</td>
</tr>
<tr>
<td>Deorbit burn</td>
<td>334 to 338</td>
<td>60030 to 60200</td>
</tr>
<tr>
<td>Pitch maneuver</td>
<td>340 to 339</td>
<td>60269 to 60504</td>
</tr>
<tr>
<td>X-axis saturation</td>
<td>79</td>
<td>62276</td>
</tr>
<tr>
<td>Z-axis saturation</td>
<td>93</td>
<td>62167</td>
</tr>
<tr>
<td>XBET epoch</td>
<td>334</td>
<td>60022</td>
</tr>
<tr>
<td>ABET epoch</td>
<td>121</td>
<td>62000</td>
</tr>
<tr>
<td><strong>ST-61C</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time blanked out for APU shift</td>
<td>261 to 257</td>
<td>47700 to 47725</td>
</tr>
<tr>
<td>Deorbit burn</td>
<td>328 to 332</td>
<td>46472 to 46704</td>
</tr>
<tr>
<td>Pitch maneuver</td>
<td>334 to 329</td>
<td>46815 to 47065</td>
</tr>
<tr>
<td>X-axis saturation</td>
<td>84</td>
<td>48711</td>
</tr>
<tr>
<td>Z-axis saturation</td>
<td>97</td>
<td>48625</td>
</tr>
<tr>
<td>XBET epoch</td>
<td>328</td>
<td>46462</td>
</tr>
<tr>
<td>ABET epoch</td>
<td>210</td>
<td>48000</td>
</tr>
</tbody>
</table>
Appendix C

Conversion of Temperature Counts to Temperature

The algorithms used to convert temperature counts into temperature for each sensor are presented in this appendix. Temperature constants of the algorithms are unique for every sensor and for every calibration of the sensor. Also, some of the methods vary between calibrations. The procedure that pertains to S/N 001 and S/N 002 (prior to modification) is based on reference 14. Refer to each subsequent section for additional information pertaining to HiRAP S/N 001 (prior to modification) and to HiRAP S/N 001 and S/N 002 (after modification). The rest of this appendix is extracted from references 9, 14, 15, and 16 with modifications as necessary.

From reference 14: The algorithms and lookup table given in the first part of this appendix are derived from data supplied by Bell Aerospace Textron for HiRAP S/N 002. It is assumed that the characteristics of the HiRAP S/N 001 sensors will be roughly similar to those of the S/N 002 sensors and that the same algorithms will be used with appropriate new entries in the lookup tables.

The coarse ranges of the HiRAP coarse-fine temperature monitors cover approximately 23°F to 152°F, with small variations between sensors. The eight fine ranges of each sensor each cover about 17.65°F, with overlaps of 1.6°F to 2.4°F. At some temperatures two fine outputs are possible, depending upon which fine range has been selected. There is no indication in the HiRAP housekeeping channels as to which fine range to use, but reference to the coarse output can resolve the ambiguity. The simplest approach is to compute the two possible fine temperatures and then select whichever is closest to the coarse temperature. Since the correct fine temperature should always be within about ±0.5°F of the coarse temperature, and the incorrect fine temperature should always be about ±17°F different, there is no possibility of selecting the wrong value.

Let

\[ V_C \]  coarse temperature monitor voltage
\[ V_F \]  fine temperature monitor voltage
\[ T_C \]  coarse temperature, °F
\[ T_F \]  final (correct) fine temperature, °F
\[ T_M, T_{M+1} \]  candidate fine temperatures, °F
\[ M \]  fine range serial number, 1 to 8
\[ K_C \]  coarse monitor scale factor (from table C1 for each sensor), V/°F
\[ K_F \]  fine monitor scale factor (from table C1 for each sensor), V/°F
\[ \theta_M \]  temperature for zero volts in fine range M (from table C1 for each sensor), °F

\[ \text{INT} \]  integral part of

Then

\[ T_C = \theta_1 + V_C / K_C \]
\[ M = \text{INT}(0.5 + 1.6 V_C) \]

Fine range number is either \( M \) or \( M + 1 \). If \( M = 0 \), use \( 1 \). If \( M + 1 = 9 \), use \( B \). Compute

\[ T_M = \theta_M + V_F / K_F \]

and

\[ T_{M+1} = \theta_{M+1} + V_F / K_F \]

Compare \( T_M \) and \( T_{M+1} \) with \( T_C \) and select whichever is within about ±0.5°F of \( T_C \) as the correct value of \( T_F \). A minimum difference significantly greater than ±0.5°F should be noted as an indication of possible changes in the coarse and/or fine temperature calibrations.

Table C1. Temperature Correction Constants for HiRAP S/N 002

(a) X-axis

| \( K_C, V/°F \) | 0.03910 |
| \( K_F, V/°F \) | 0.28257 |
| \( \theta_M, °F \): |
| \( M = 1 \) | 23.68 |
| \( M = 2 \) | 39.45 |
| \( M = 3 \) | 55.22 |
| \( M = 4 \) | 71.10 |
| \( M = 5 \) | 86.97 |
| \( M = 6 \) | 102.44 |
| \( M = 7 \) | 117.90 |
| \( M = 8 \) | 133.71 |

(b) Y-axis

| \( K_C, V/°F \) | 0.03891 |
| \( K_F, V/°F \) | 0.28347 |
| \( \theta_M, °F \): |
| \( M = 1 \) | 23.57 |
| \( M = 2 \) | 38.83 |
| \( M = 3 \) | 54.10 |
| \( M = 4 \) | 70.31 |
| \( M = 5 \) | 86.52 |
| \( M = 6 \) | 102.32 |
| \( M = 7 \) | 118.13 |
| \( M = 8 \) | 134.17 |
Table C1. Concluded

(c) Z-axis

\[
K_C, V/°F \quad 0.03904 \\
K_F, V/°F \quad 0.28254 \\
θ_M, °F: \\
M = 1 \quad 23.86 \\
M = 2 \quad 39.50 \\
M = 3 \quad 55.15 \\
M = 4 \quad 70.84 \\
M = 5 \quad 86.53 \\
M = 6 \quad 102.36 \\
M = 7 \quad 118.20 \\
M = 8 \quad 134.15
\]

From reference 9: The method of calculating temperatures [for S/N 001 (prior to modification)] is the same as that described in reference 14, except for one minor difference. From reference 14 the equation for calculating the coarse temperature is

\[
T_C = θ_1 + V_C/K_C
\]

This equation is modified to

\[
T_C = θ_C + V_C/K_C
\]

where the relevant values of θ_C are given along with K_C, K_F, and θ_M in tables C2. The reason for the change is that the constant in one best-fit equation for T_C has been found to differ from θ_1 by more than 0.1°F, although in a perfect system they should be identical.

Table C2. Correction Constants for HiRAP S/N 001

(a) X-axis

\[
K_C, V/°F \quad 0.04096 \\
K_F, V/°F \quad 0.29174 \\
θ_C, °F \quad 30.70 \\
θ_M, °F: \\
M = 1 \quad 30.56 \\
M = 2 \quad 45.74 \\
M = 3 (“halfway” range) \quad 60.75 \\
M = 4 \quad 75.76 \\
M = 5 \quad 90.83 \\
M = 6 \quad 105.56 \\
M = 7 (“halfway” range) \quad 120.57 \\
M = 8 \quad 135.57
\]

(b) Y-axis

\[
K_C, V/°F \quad 0.04098 \\
K_F, V/°F \quad 0.29420 \\
θ_C, °F \quad 29.98
\]

From reference 12: In reference 15, bias was treated as a function of the corrected coarse temperature monitor (CTM) voltage rather than as a function of temperature, as was the case in all previous calibrations of HiRAP’s. The same procedure is followed here.

A best-fit temperature versus CTM voltage (or corrected CTM voltage) function is included in the data sheet for each axis, but it is not required for the computation of bias.

The following is the method of calculating effective temperature monitor voltage. This is unchanged from reference 15. Because each of the eight fine temperature ranges overlaps its neighbor’s there may be an ambiguity to be resolved. Let

\[
V_C \quad \text{coarse temperature monitor voltage} \\
V_F \quad \text{fine temperature monitor voltage} \\
V_C \quad \text{final (corrected) temperature monitor voltage} \\
M \quad \text{fine range serial number, 1 to 8} \\
G \quad \text{slope (gain) of } V_F \text{ relative to } V_C \\
V_M \quad \text{value of } V_C \text{ corresponding to } 0 \text{ V in fine range } M
\]

\[
\text{INT} \quad \text{integral part of}
\]
Then

\[ M = \text{INT}(0.5 + 1.6V_C) \]

Fine range number is either \( M \) or \((M + 1)\). If \( M = 0 \), use 1. If \((M + 1) = 9\), use 8. Compute

\[ V_{c1} = V_M + V_F/G \]

and

\[ V_{c2} = V_{M+1} + V_F/G \]

Compare \( V_{c1} \) and \( V_{c2} \) with \( V_C \), and choose whichever is the closest as the value of \( V_c \) to be used in computing bias— one value will always be much closer than the other, so there will be no possibility of an incorrect choice.

The appropriate values of \( G \) and \( V_M \) are given in tables C3 and C4. For each sensor axis the value of \( G \) is given as a constant, since there were no significant variations with temperature. The worst-case deviations from the mean values would produce an error of less than 0.2 \( \mu \text{g} \) in the estimated bias.

Table C3. Temperature Monitor Constants for S/N 001 After Recalibration

(a) X-axis

[From pp. 28, 60, 61, and 62 of ref. 15; \( G = 7.181 \pm 0.005 \)]

<table>
<thead>
<tr>
<th>( M )</th>
<th>( V_M ), V</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.0014</td>
</tr>
<tr>
<td>2</td>
<td>0.6127</td>
</tr>
<tr>
<td>3</td>
<td>1.2268</td>
</tr>
<tr>
<td>4</td>
<td>1.8407</td>
</tr>
<tr>
<td>5</td>
<td>2.4547</td>
</tr>
<tr>
<td>6</td>
<td>3.0685</td>
</tr>
<tr>
<td>7</td>
<td>3.6823</td>
</tr>
<tr>
<td>8</td>
<td>4.2963</td>
</tr>
</tbody>
</table>

(b) Y-axis

[From pp. 31, 64, 65, and 66 of ref. 15; \( G = 7.186 \pm 0.003 \)]

<table>
<thead>
<tr>
<th>( M )</th>
<th>( V_M ), V</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000</td>
</tr>
<tr>
<td>2</td>
<td>0.6142</td>
</tr>
<tr>
<td>3</td>
<td>1.2283</td>
</tr>
<tr>
<td>4</td>
<td>1.8424</td>
</tr>
<tr>
<td>5</td>
<td>2.4564</td>
</tr>
<tr>
<td>6</td>
<td>3.0704</td>
</tr>
<tr>
<td>7</td>
<td>3.6843</td>
</tr>
<tr>
<td>8</td>
<td>4.2981</td>
</tr>
</tbody>
</table>

(c) Z-axis

[From pp. 34, 68, 69, and 70 of ref. 15; \( G = 7.192 \pm 0.004 \)]

<table>
<thead>
<tr>
<th>( M )</th>
<th>( V_M ), V</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.0002</td>
</tr>
<tr>
<td>2</td>
<td>0.6140</td>
</tr>
<tr>
<td>3</td>
<td>1.2281</td>
</tr>
<tr>
<td>4</td>
<td>1.8424</td>
</tr>
<tr>
<td>5</td>
<td>2.4566</td>
</tr>
<tr>
<td>6</td>
<td>3.0705</td>
</tr>
<tr>
<td>7</td>
<td>3.6845</td>
</tr>
<tr>
<td>8</td>
<td>4.2984</td>
</tr>
</tbody>
</table>

Table C4. Temperature Monitor Constants for S/N 002 After Recalibration

(a) X-axis

[From ref. 16; \( G = 7.221 \pm 0.020 \)]

<table>
<thead>
<tr>
<th>( M )</th>
<th>( V_M ), V</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.6158</td>
</tr>
<tr>
<td>2</td>
<td>1.2302</td>
</tr>
<tr>
<td>3</td>
<td>1.8463</td>
</tr>
<tr>
<td>4</td>
<td>2.4609</td>
</tr>
<tr>
<td>5</td>
<td>3.0759</td>
</tr>
<tr>
<td>6</td>
<td>3.6912</td>
</tr>
<tr>
<td>7</td>
<td>4.3047</td>
</tr>
<tr>
<td>8</td>
<td>5.0121</td>
</tr>
</tbody>
</table>

(b) Y-axis

[From ref. 16; \( G = 7.1787 \pm 0.0038 \)]

<table>
<thead>
<tr>
<th>( M )</th>
<th>( V_M ), V</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000</td>
</tr>
<tr>
<td>2</td>
<td>0.6142</td>
</tr>
<tr>
<td>3</td>
<td>1.2283</td>
</tr>
<tr>
<td>4</td>
<td>1.8424</td>
</tr>
<tr>
<td>5</td>
<td>2.4564</td>
</tr>
<tr>
<td>6</td>
<td>3.0704</td>
</tr>
<tr>
<td>7</td>
<td>3.6843</td>
</tr>
<tr>
<td>8</td>
<td>4.2981</td>
</tr>
</tbody>
</table>

31
Table C4. Concluded

(c) Z-axis

[From ref. 16; $G = 7.1798 \pm 0.0019$]

<table>
<thead>
<tr>
<th>$M$</th>
<th>$V_M$, V</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.6141</td>
</tr>
<tr>
<td>2</td>
<td>0.6141</td>
</tr>
<tr>
<td>3</td>
<td>2.4579</td>
</tr>
<tr>
<td>4</td>
<td>3.0726</td>
</tr>
<tr>
<td>5</td>
<td>3.6874</td>
</tr>
<tr>
<td>6</td>
<td>4.3019</td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>
Appendix D
Data Tape Volume Serial Number (VSN) Identifiers of RCS Chamber Pressure Tapes
and RCS Zero Reference Values

STS-06 Thrust Removal
RCS Pressure Data Tape: VSN = NE0927

Zero Reference Values used in program THRUST to determine
thruster firing times:

ZERO(5) = 1.6
ZERO(6) = 0.8
ZERO(7) = 0.8
ZERO(10) = 0.8
ZERO(12) = 0.8
ZERO(22) = 0.8
ZERO(28) = 0.8
ZERO(31) = 2.4
ZERO(38) = 0.8
ZERO(40) = 0.8
ZERO(42) = 0.8
ZERO(44) = 0.8

Lag Times used in Thrust:

TLAG = 0.04
TLAG1 = 0.80

Data intervals of thruster activity removed:

Thrust Start = Firing Time - TLAG
Thrust Stop = Firing Time + TLAG1
STS-07 Thrust Removal
RCS Pressure Data Tape: VSN = NL1187

Zero Reference Values used in program THRUST to determine thrustor firing times:

ZERO(5) = 1.6
ZERO(10) = 0.8
ZERO(12) = 1.6
ZERO(14) = 0.8
ZERO(21) = 0.8
ZERO(22) = 0.8
ZERO(31) = 2.4
ZERO(38) = 1.6
ZERO(40) = 0.8
ZERO(42) = 1.6
ZERO(44) = 0.8

Lag Times used in Thrust:

TLAG = 0.04
TLAG1 = 0.80

Data intervals of thrustor activity removed:

Thrust Start = Firing Time - TLAG
Thrust Stop = Firing Time + TLAG1
STS-08 Thrust Removal
RCS Pressure Data Tape: VSN = NH0732

Zero Reference Values used in program THRUST to determine thrustor firing times:

ZERO(5) = 1.7
ZERO(6) = 0.8
ZERO(7) = 0.8
ZERO(8) = 0.8
ZERO(9) = 0.8
ZERO(10) = 0.8
ZERO(12) = 0.8
ZERO(14) = 0.8
ZERO(21) = 0.8
ZERO(22) = 0.8
ZERO(28) = 0.8
ZERO(31) = 2.4
ZERO(38) = 0.8
ZERO(40) = 0.8
ZERO(42) = 0.8
ZERO(44) = 0.8

Lag Times used in Thrust:

TLAG = 0.04
TLAG1 = 0.80

Data intervals of thrustor activity removed:

Thrust Start = Firing Time - TLAG
Thrust Stop = Firing Time + TLAG1
Zero Reference Values used in program THRUST to determine thrustor firing times:

ZERO( 4) = 0.8  
ZERO( 5) = 0.8  
ZERO(10) = 0.8  
ZERO(13) = 0.8  
ZERO(16) = 0.8  
ZERO(20) = 0.8  
ZERO(21) = 0.8  
ZERO(23) = 1.6  
ZERO(24) = 1.6  
ZERO(27) = 0.8  
ZERO(29) = 0.8  
ZERO(30) = 1.6  
ZERO(32) = 0.8  
ZERO(33) = 0.8  
ZERO(35) = 0.8  
ZERO(37) = 0.8  
ZERO(38) = 0.8  
ZERO(40) = 1.6  
ZERO(41) = 0.8  
ZERO(42) = 1.6  
ZERO(44) = 0.8

Lag Times used in Thrust:

TLAG = 0.04  
TLAG1 = 0.80

Data intervals of thrustor activity removed:

Thrust Start = Firing Time - TLAG  
Thrust Stop = Firing Time + TLAG1
STS-41B Thrust Removal
RCS Pressure Data Tape: VSN = NT0106

Zero Reference Values used in program THRUST to determine
thrustor firing times:

ZERO( 5) = 1.6
ZERO(10) = 0.8
ZERO(12) = 1.6
ZERO(14) = 0.8
ZERO(21) = 0.8
ZERO(22) = 0.8
ZERO(31) = 2.4
ZERO(38) = 1.6
ZERO(40) = 0.8
ZERO(42) = 1.6
ZERO(44) = 0.8

Lag Times used in Thrust:

TLAG = 0.04
TLAG1 = 0.80

Data intervals of thrustor activity removed:

Thrust Start = Firing Time - TLAG

Thrust Stop = Firing Time + TLAG1
STSW-41C Thrust Removal
RCS Pressure Data Tape: VSN = NS1149

Zero Reference Values used in program THRUST to determine thrustor firing times:

ZERO(5) = 1.6
ZERO(10) = 0.8
ZERO(12) = 0.8
ZERO(21) = 0.8
ZERO(26) = 0.8
ZERO(27) = 0.8
ZERO(28) = 0.8
ZERO(30) = 0.8
ZERO(38) = 0.8
ZERO(40) = 0.8
ZERO(42) = 1.6
ZERO(44) = 0.8

Lag Times used in Thrust:

TLAG = 0.04
TLAG1 = 0.80

Data intervals of thrustor activity removed:

Thrust Start = Firing Time - TLAG
Thrust Stop = Firing Time + TLAG1
STS-51B Thrust Removal
RCS Pressure Data Tape: VSN = NE1020

Zero Reference Values used in program THRUST to determine thrustor firing times:

ZERO( 5) = 1.6
ZERO( 6) = 0.8
ZERO(10) = 0.8
ZERO(12) = 0.8
ZERO(21) = 0.8
ZERO(22) = 1.6
ZERO(25) = 0.8
ZERO(31) = 2.4
ZERO(39) = 0.8
ZERO(42) = 1.6

Lag Times used in Thrust:

TLAG = 0.04
TLAG1 = 0.80

Data intervals of thrustor activity removed:

Thrust Start = Firing Time - TLAG
Thrust Stop = Firing Time + TLAG1
STS-51F Thrust Removal  
RCS Pressure Data Tape: VSN = NF0565

Zero Reference Values used in program THRUST to determine thrustor firing times:

ZERO(5) = 1.6  
ZERO(6) = 0.8  
ZERO(7) = 0.8  
ZERO(10) = 0.8 
ZERO(12) = 1.6 
ZERO(21) = 0.8 
ZERO(22) = 1.6 
ZERO(23) = 0.8 
ZERO(25) = 0.8 
ZERO(31) = 2.4 
ZERO(42) = 0.8

Lag Times used in Thrust:

TLAG = 0.04  
TLAG1 = 0.80

Data intervals of thrustor activity removed:

Thrust Start = Firing Time - TLAG 
Thrust Stop = Firing Time + TLAG1
STS-61A Thrust Removal
RCS Pressure Data Tape: VSN = NJ0978

Zero Reference Values used in program THRUST to determine thrustor firing times:

ZERO( 5) = 1.6
ZERO( 6) = 0.8
ZERO( 8) = 0.8
ZERO(10) = 0.8
ZERO(12) = 1.6
ZERO(21) = 0.8
ZERO(22) = 0.8
ZERO(23) = 0.8
ZERO(25) = 0.8
ZERO(31) = 2.4
ZERO(35) = 1.6
ZERO(36) = 0.8
ZERO(39) = 0.8
ZERO(42) = 0.8
ZERO(43) = 1.6

Lag Times used in Thrust:

TLAG = 0.84
TLAG1 = 0.96

Data intervals of thrustor activity removed:

Thrust Start = Firing Time - TLAG
Thrust Stop = Firing Time + TLAG1
STS-61C Thrust Removal
RCS Pressure Data Tape: VSN = NH0153

Zero Reference Values used in program THRUST to determine thruster firing times:

ZERO( 5) = 0.8
ZERO( 6) = 0.8
ZERO( 8) = 0.8
ZERO( 9) = 0.8
ZERO(12) = 0.8
ZERO(13) = 0.8
ZERO(14) = 0.8
ZERO(15) = 0.8
ZERO(16) = 0.8
ZERO(20) = 0.8
ZERO(21) = 0.8
ZERO(24) = 0.8
ZERO(25) = 0.8
ZERO(41) = 0.8
ZERO(42) = 0.8

Lag Times used in Thrust:

TLAG = 0.08
TLAG1 = 0.96

Data intervals of thruster activity removed:

Thrust Start = Firing Time - TLAG

Thrust Stop = Firing Time + TLAG1
Appendix E
Orbiter Weight at Entry and Center-of-Gravity Locations

<table>
<thead>
<tr>
<th>Flight</th>
<th>Weight at entry interface, lb</th>
<th>X-axis</th>
<th>Y-axis</th>
<th>Z-axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>STS-06</td>
<td>191384.0</td>
<td>1101.2</td>
<td>0.3</td>
<td>371.5</td>
</tr>
<tr>
<td>STS-07</td>
<td>204983.0</td>
<td>1091.3</td>
<td>-.6</td>
<td>373.3</td>
</tr>
<tr>
<td>STS-08</td>
<td>205020.0</td>
<td>1091.5</td>
<td>-.1</td>
<td>373.5</td>
</tr>
<tr>
<td>STS-09</td>
<td>221143.4</td>
<td>1087.3</td>
<td>-.1</td>
<td>373.7</td>
</tr>
<tr>
<td>STS-41B</td>
<td>202966.5</td>
<td>1090.7</td>
<td>1.3</td>
<td>372.6</td>
</tr>
<tr>
<td>STS-41C</td>
<td>198152.8</td>
<td>1101.5</td>
<td>-.1</td>
<td>371.6</td>
</tr>
<tr>
<td>STS-51B</td>
<td>214787.4</td>
<td>1085.7</td>
<td>-.3</td>
<td>373.4</td>
</tr>
<tr>
<td>STS-51F</td>
<td>218227.4</td>
<td>1082.3</td>
<td>-.6</td>
<td>373.4</td>
</tr>
<tr>
<td>STS-61A</td>
<td>215255.4</td>
<td>1085.5</td>
<td>-.4</td>
<td>374.2</td>
</tr>
<tr>
<td>STS-61C</td>
<td>211194.4</td>
<td>1085.2</td>
<td>.4</td>
<td>371.4</td>
</tr>
</tbody>
</table>
Appendix F  
Ground Calibration Procedure

Each sensor is calibrated before its delivery, prior to installation, and again after the sensor is repaired or modified. The purpose of the calibration is to evaluate the bias and bias slope of the instrument with change in temperature. The calibration also evaluates the change of scale factor with temperature. The basic procedure of the calibration is described in the following sections, which are from reference 14 with some modification.

Temperature Correction of Accelerometer Scale Factor

The scale factors for each HiRAP sensor are determined at five temperatures, nominally 30°F, 60°F, 90°F, 120°F, and 150°F. In practice, the actual calibration temperatures may differ by up to ±5°F from the nominals. The method of finding the scale factor used in processing HiRAP data is to calculate the fine temperature $T_F$, as described in appendix B. Let

\[ r \quad \text{reference temperature serial number, from 1 to 5} \]
\[ T_r \quad \text{reference temperature at serial } r, \, ^\circ F \]
\[ K_r \quad \text{scale factor at serial } r, \, V/mg \]
\[ K_{T,r} \quad \text{scale factor temperature coefficient from } r \text{ to } r+1, \, (V/mg)/^\circ F \]

All these quantities appear in the lookup tables given in the calibration reports for each particular sensor (refs. 9, 12, 15, and 17).

The procedure is to step $r$ from 1 to 4 until $T_F$ lies between $T_r$ and $T_{r+1}$. The corrected scale factor $K$ is then given by

\[ K = K_r + (T_F - T_r) K_{T,r} \, V/mg \]

Temperature Correction of Accelerometer Bias

The bias of each HiRAP sensor can be as much as ±1 mg, with slow drifts over time as well as over temperature. The only way of obtaining measurements during entry that are accurate to ±10 μg or better is to record the sensor's output and its temperature during a quiet period in orbit shortly before entry and then, with this treated as the datum, compute the bias shifts due to subsequent changes in temperature.

The bias of each sensor is measured at the same five temperatures as the scale factor, but the absolute values are of no interest, only the differences over each temperature interval. To correct bias during flight data processing, the difference in calibration bias from the nominal 30°F value is recorded in the lookup tables given in the calibration reports for each sensor (refs. 9, 12, 15, and 17), along with the bias temperature coefficient over each temperature interval. The bias changes relative to the 30°F value are computed for the datum on-orbit temperature and the particular entry temperature, the difference between these two being the required bias correction for temperature.

Let

\[ r \quad \text{reference temperature serial number, 1 to 5} \]
\[ T_r \quad \text{reference temperature at serial } r, \, ^\circ F \]
\[ B_r \quad \text{bias at serial } r \text{ relative to nominal 30°F value, } \mu g \]
\[ B_{T,r} \quad \text{bias temperature coefficient over interval from serial } r \text{ to } r+1, \, (B_{r+1} - B_r)(T_{r+1} - T_r), \mu g \]

The above quantities appear in the lookup tables given in the calibration reports for each particular sensor.

Also define the following:

\[ T_{F,0} \quad \text{datum on-orbit fine temperature, } ^\circ F \]
\[ T_{F,1} \quad \text{entry fine temperature, } ^\circ F \]
\[ B_0 \quad \text{bias at } T_{F,0} \text{ relative to nominal 30°F value, } \mu g \]
\[ B_1 \quad \text{bias at } T_{F,1} \text{ relative to nominal 30°F value, } \mu g \]
\[ \Delta B_0 \quad \text{bias change from datum, } \mu g \]

Step $r$ from 1 to 4 until $T_{F,0}$ lies between $T_r$ and $T_{r+1}$. Then

\[ B_0 = B_r + (T_{F,0} - T_r) B_{T,r} \, \mu g \]

(This need be computed only once.) Step $r'$ from 1 to 4 until $T_{F,1}$ lies between $T_{r'}$ and $T_{r+1}$. Then

\[ B_1 = B_{r'} + (T_{F,1} - T_{r'}) B_{T,r'} \, \mu g \]

and

\[ \Delta B_0 = B_1 - B_0 \, \mu g \]

For an example of this procedure, see reference 14.
Appendix G

Temperature Biases and Bias Slopes

The following table shows the value of bias and bias slope for each flight of the HiRAP experiment. The results given for the post-APU procedure of bias calibration are evaluated at the midpoint of the 400-sec section of data used for this method of calibration. The values of bias and bias slope calculated with the laboratory-derived relation of bias versus temperature (or voltage) are evaluated at the temperature (or voltage) given at the midpoint of the 400-sec calibration period. These results are given as ground calibration in the table below. For the Z-axis an alternate method of bias evaluation is applied, as described in the main text. The results with this method used to evaluate bias on the Z-axis are given in the table below as an alternate procedure.

### X-axis

<table>
<thead>
<tr>
<th>Flight</th>
<th>Bias slope, ( \mu g/\circ F ), from...</th>
<th>Bias, ( \mu g ), from...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Post-APU procedure</td>
<td>Ground calibration procedure</td>
</tr>
<tr>
<td>STS-06</td>
<td>-30.6</td>
<td>-26.0</td>
</tr>
<tr>
<td>STS-07</td>
<td>-22.3</td>
<td>-20.0</td>
</tr>
<tr>
<td>STS-08</td>
<td>-26.7</td>
<td>-20.4</td>
</tr>
<tr>
<td>STS-09</td>
<td>-23.3</td>
<td>-16.4</td>
</tr>
<tr>
<td>STS-41B</td>
<td>-25.5</td>
<td>-21.1</td>
</tr>
<tr>
<td>STS-41C</td>
<td>-22.8</td>
<td>-20.0</td>
</tr>
<tr>
<td>STS-51B</td>
<td>-23.8</td>
<td>-20.0</td>
</tr>
<tr>
<td>STS-51F</td>
<td>-18.2</td>
<td>-18.8</td>
</tr>
<tr>
<td>STS-61A</td>
<td>-21.4</td>
<td>-17.3</td>
</tr>
<tr>
<td>STS-61C</td>
<td>-27.9</td>
<td>-25.3</td>
</tr>
</tbody>
</table>

### Y-axis

<table>
<thead>
<tr>
<th>Flight</th>
<th>Bias slope, ( \mu g/\circ F ), from...</th>
<th>Bias, ( \mu g ), from...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Post-APU procedure</td>
<td>Ground calibration procedure</td>
</tr>
<tr>
<td>STS-06</td>
<td>10.5</td>
<td>15.0</td>
</tr>
<tr>
<td>STS-07</td>
<td>4.1</td>
<td>10.0</td>
</tr>
<tr>
<td>STS-08</td>
<td>8.9</td>
<td>10.6</td>
</tr>
<tr>
<td>STS-09</td>
<td>29.7</td>
<td>23.6</td>
</tr>
<tr>
<td>STS-41B</td>
<td>4.1</td>
<td>10.3</td>
</tr>
<tr>
<td>STS-41C</td>
<td>27.8</td>
<td>9.1</td>
</tr>
<tr>
<td>STS-51B</td>
<td>6.1</td>
<td>10.9</td>
</tr>
<tr>
<td>STS-51F</td>
<td>29.0</td>
<td>29.9</td>
</tr>
<tr>
<td>STS-61A</td>
<td>27.7</td>
<td>28.9</td>
</tr>
<tr>
<td>STS-61C</td>
<td>15.3</td>
<td>13.5</td>
</tr>
</tbody>
</table>
### Z-axis

<table>
<thead>
<tr>
<th>Flight</th>
<th>Post-APU procedure</th>
<th>Ground calibration procedure</th>
<th>Alternate procedure</th>
<th>Post-APU procedure</th>
<th>Ground calibration procedure</th>
<th>Alternate procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>STS-06</td>
<td>-21.2</td>
<td>-15.8</td>
<td>(a)</td>
<td>-1587</td>
<td>-1666</td>
<td>-1586</td>
</tr>
<tr>
<td>STS-07</td>
<td>-19.3</td>
<td>-20.0</td>
<td></td>
<td>-617</td>
<td>-722</td>
<td>-617</td>
</tr>
<tr>
<td>STS-08</td>
<td>-28.3</td>
<td>-21.6</td>
<td></td>
<td>-389</td>
<td>-494</td>
<td>-389</td>
</tr>
<tr>
<td>STS-09</td>
<td>-13.0</td>
<td>-8.7</td>
<td></td>
<td>-224</td>
<td>-216</td>
<td>-223</td>
</tr>
<tr>
<td>STS-41B</td>
<td>-25.3</td>
<td>-19.9</td>
<td></td>
<td>-212</td>
<td>-362</td>
<td>-213</td>
</tr>
<tr>
<td>STS-41C</td>
<td>-19.7</td>
<td>-19.9</td>
<td></td>
<td>-659</td>
<td>-742</td>
<td>-660</td>
</tr>
<tr>
<td>STS-51B</td>
<td>-22.1</td>
<td>-21.0</td>
<td></td>
<td>-615</td>
<td>-579</td>
<td>-614</td>
</tr>
<tr>
<td>STS-51F</td>
<td>-12.4</td>
<td>-12.0</td>
<td></td>
<td>5449</td>
<td>5404</td>
<td>5449</td>
</tr>
<tr>
<td>STS-61A</td>
<td>-9.9</td>
<td>-10.7</td>
<td></td>
<td>5195</td>
<td>5126</td>
<td>5198</td>
</tr>
<tr>
<td>STS-61C</td>
<td>-18.5</td>
<td>-18.0</td>
<td></td>
<td>-1806</td>
<td>-1977</td>
<td>-1804</td>
</tr>
</tbody>
</table>

*(a) Not available.*
**Appendix H**

**Source Codes**

```plaintext
PROGRAM HRPSTRP (INPUT, OUTPUT, TAPE1, TAPE2)
DIMENSION INBUF(615)

C*************THIS IS THE ORIGINAL, DON'T MUCK ABOUT WITH IT**********
COMMON/TBLK/BTIME, ETIME, NCYC
PRINT 2
2 FORMAT (1H1)
   ISN=1
   NFILE=0
   NCYC=0
10 DO 15 I=1, 615
   INBUF(I)=0
15 CONTINUE
BUFFER IN(I,1) (INBUF(1), INBUF(615))
IF(UNIT(I)) 20, 22, 25
20 I LEN=LENGTH(1)
   IF(NCYC .GT. I000) STOP
   CALL BYTE8(ILEN, INBUF)
   GO TO 10
22 NFILE=NFILE+I
ENDFILE 2
PRINT I00, NFILE, BTIME, ETIME, NCYC
   NCYC=0
   BUFFER IN(I,1) (INBUF(1), INBUF(615))
   IF(UNIT(I)) 20, 23, 25
23 ENDFILE 2
   STOP
25 PRINT 250, ETIME
250 FORMAT (///IX, 'PARITY ERROR --- TIME OF LAST GOOD DATA FRAME = ', IF9.3)
30 STOP
END
SUBROUTINE BYTE8 (ILEN, INBUF)
DIMENSION INBUF(ILEN)
COMMON/Master/IDATA(4608)
IF(ILEN .LT. 615) THEN
   IFLAG=208
   K=I
ELSE
   IFLAG=4608
   K=17
END IF
DO 10 I=1, IFLAG
   IDATA(I)=0
10 CONTINUE
   J=0
   DO 20 I=1, ILEN
   DO 15 L=K, 60, 8
   J=J+1
   IF(J .GT. IFLAG) GO TO 20
   CALL STRING(INBUF(I), L, IDATA(J), 53, 8)
15 CONTINUE
   IF(L .EQ. 65) K=5
```

47
IF(L .EQ. 61) K=1
CONTINUE
IF(IFLAG .EQ. 208) THEN
CALL RECHDR
ELSE
CALL RECDAT
END IF
RETURN
END
SUBROUTINE RECDAT
COMMON/MASTER/IDATA(4608)
COMMON/TBLK/BTIME,ETIME,NCYC
DIMENSION ISEC(65),IMSEC(65),IXCNT(65),IYCNT(65),IZCNT(65),IENG(8)
ISEC(65)=IMSEC(65)=IXCNT(65)=IYCNT(65)=IZCNT(65)=0
NCYC=NCYC+1
K=0
DO 25 J=1,64
ID=IDATA(K+3)
ITIME=0
CALL STRING(IDATA(K+6),53,ITIME,37,8)
CALL STRING(IDATA(K+5),53,ITIME,45,8)
CALL STRING(IDATA(K+8),53,ITIME,53,8)
CALL BCDCOD(ID,ITIME,NDAY,NHR,MIN,NSEC,MSEC)
IF(ID .EQ. 0) THEN
HM=FLOAT(NHR)*3600.+FLOAT(MIN)*60.
ELSE
ISEC(J)=NSEC
IMSEC(J)=MSEC
SEC=FLOAT(NSEC)+FLOAT(MSEC)/1000.
END IF
IF(NCYC .EQ. 1.AND. ID .EQ. 1) BTIME=HM+SEC=0.008875
IXCNT(J)=0
CALL STRING(IDATA(K+9),53,IXCNT(J),53,8)
CALL STRING(IDATA(K+10),55,IXCNT(J),47,6)
IYCNT(J)=0
CALL STRING(IDATA(K+33),53,IYCNT(J),53,8)
CALL STRING(IDATA(K+34),55,IYCNT(J),47,6)
IZCNT(J)=0
CALL STRING(IDATA(K+15),53,IZCNT(J),53,8)
CALL STRING(IDATA(K+16),55,IZCNT(J),47,6)
IF(J .LT. 34 .OR. J .GT. 41) GO TO 20
IENG(J-33)=IDATA(K+7)
K=K+72
20 CONTINUE
ETIME=HM+SEC
I=64
30 IF(I .EQ. 2) GO TO 40
ITIM=(ISEC(I)-ISEC(I-1))*1000+IMSEC(I)-IMSEC(I-1)
IF(ITIM .LT. 0) GO TO 50
I=I-1
GO TO 30
40 IF(IMSEC(2) .LT. 8 .AND. ISEC(2) .EQ. 0) GO TO 50
GO TO 70
50 DO 60 L=I,64
60


```
60  ISEC(L) = ISEC(L) + 60
70  ISEC(1) = HM + ISEC(2) + 1
     IMSEC(1) = IMSEC(2) - 9
     IF(IMSEC(1) .GE. 0) GO TO 79
     IMSEC(1) = IMSEC(1) + 1000
79  DO 80 I = 2, 64
80  ISEC(I) = ISEC(I) + HM + 1
     WRITE (2) ISEC(34), IMSEC(34), (IENG(J), J = 1, 8)
     DO 90 I = 1, 65
90  WRITE (2) ISEC(I), IMSEC(I), IXCNT(I), IYCNT(I), IZCNT(I)
     RETURN
END
SUBROUTINE BCDCOD (ID, ITIME, NDAY, NHR, MIN, NSEC, MSEC)
   IF (ID .EQ. 0) THEN
     NDAY = NHR = MIN = 0
     CALL STRING (ITIME, 37, NDAY, 51, I0)
     CALL BCDOCT (NDAY)
     CALL STRING (ITIME, 47, NHR, 55, 6)
     CALL BCDOCT (NHR)
     CALL STRING (ITIME, 53, MIN, 54, 7)
     CALL BCDOCT (MIN)
   ELSE
     NSEC = MSEC = 0
     CALL STRING (ITIME, 53, NSEC, 54, 7)
     CALL BCDOCT (NSEC)
     CALL STRING (ITIME, 43, MSEC, 51, I0)
     CALL BCDOCT (MSEC)
   END IF
   RETURN
END
SUBROUTINE BCDOCT (I)
   IUN = 0
   ITEN = 0
   IHUN = 0
   CALL STRING (I, 49, IHUN, 57, 4)
   CALL STRING (I, 53, ITEN, 57, 4)
   CALL STRING (I, 57, IUN, 57, 4)
   I = (IHUN*I0 + ITEN) *I0 + IUN
   RETURN
END
SUBROUTINE RECHDR
   DIMENSION IDIS(9)
   COMMON/MASTER/IDATA (208), IDUM (3888)
   FORID = IDATA(1)
   SCPU = IDATA(113)
   FRSZ = IDATA(115)
   FRNO = IDATA(117)
   RANGE = IDATA(119)
   CALL ASC2DC (IDIS)
   PRINT 5015, FORID, SCPU, FRSZ, FRNO, RANGE, IDIS
5015  FORMAT (IHI, 'FORMAT ID = ', F6.2, /IX, 'CPU WORD SIZE = ',
     21X, 'DATA RANGE = ', F6.2, /IX, 9A10//)
   RETURN
END
```
SUBROUTINE ASC2DC(IDIS)
DIMENSION IDIS(9)
COMMON/MASTER/IDATA(208), IDUM(3888)
INTEGER ASCII(17), Dcly(17)
DATA ASCII/O"53", O"55", O"52", O"57", O"50", O"44", O"75", O"40",
O"54", O"56", O"43", O"133", O"135", O"45", O"73", O"72"
O"56", O"57", O"60", O"61", O"62", O"63", O"77", O"00"
DO 10 I=1, 9
IDIS(I)=10H
10 CONTINUE
DCBIT=1
IDWD=1
DO 50 I=1, 88
IF((IDATA(120+I) .LT. O"101") .OR. (IDATA(120+I) .GT. O"132")) GO TO 15
IDATA(120+I)=IDATA(120+I) - O"100"
GO TO 40
15 IF((IDATA(120+I) .LT. O"60") .OR. (IDATA(120+I) .GT. O"71")) GO TO 20
IDATA(120+I)=IDATA(120+I) - O"25"
GO TO 40
20 DO 30 J=1, 17
IF(IDATA(120+I) .NE. ASCII(J)) GO TO 30
IDATA(120+I)=DSPLY(J)
GO TO 40
30 CONTINUE
IDATA(120+I)=O"55"
40 CALL STRING(IDATA(120+I), 55, IDIS(IDWD), DCBIT, 6)
DCBIT=DCBIT+6
IF(DCBIT.LE. 60) GO TO 50
IDWD=IDWD+1
DCBIT=DCBIT-60
50 CONTINUE
RETURN
END
PROGRAM SCIREAD (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE9, TAPE1, * TAPE2)

THIS ROUTINE READS AND PRINTS RAW DATA FROM HIRAP TAPES

WRITTEN BY: J. W. RUSSELL, DEC 1985

DIMENSION DATAPT(4), NAMES(4), XMAX(4), XMIN(4)

CHARACTER*8 NAMES
DATA NAMES/'TIME-SEC','X-COUNTS','Z-COUNTS','Y-COUNTS'/

WRITE (6, 800)
800 FORMAT (/5X, 'JTREAD STARTED',/) 
START = 0.0
KMAX = 225
IREV = 0
IGAP = 0
NCHAN = 4

READ TAPE HEADER
READ (9, END=600) ISN, NCHAN, (NAMES(I),I=1,NCHAN),
* (IUNITS(I),I=1,NCHAN), (HDR(I),I=1,8)
READ (9, END=600) IDAY, IHOUR, IMIN
WRITE (6, 123) IDAY, IHOUR, IMIN
123 FORMAT (2X, 'IDAY, IHOUR, IMIN ON HEADER = ', 3 (2X, I6),/) 
600 IF (EOF(9).NE. 0) THEN
WRITE (6, 802)
802 FORMAT (5X, 40HEOF FOUND WHEN ATTEMPTING TO READ HEADER )
GO TO 900
ENDIF

WRITE (6, 104) ISN, NCHAN, (HDR(I),I=1,8)
C104 FORMAT (1H1, 5X, 7HFILE = ,I7, 5X, 8HNCHAN = ,112, 5X, 15HFOUND HEADER =
* ,8A10 )
WRITE (6, 303) (NAMES(I),I=1,NCHAN)
WRITE (6, 303) (IUNITS(I),I=1,NCHAN)
303 FORMAT (1X, 10(A10,2X))

READ (9, END=91) (DATAPT(I),I=1,NCHAN)
WRITE (1) DATAPT(1), DATAPT(2), DATAPT(3), DATAPT(4)

SET MINIMUM AND MAXIMUM VALUES

DO 1 I = 2, NCHAN
XMIN(I) = DATAPT(I)
XMAX(I) = DATAPT(I)
1 CONTINUE
ITIME = 0
START = DATAPT(1)
TIMENEW = START - 0.1
KCNT = 0
NUMPTS = 0

MOVE TO DESIRED START AND PRINT RAW DATA UNTIL END OF FILE
IS REACHED ELIMINATING BAD POINTS AND TIME REVERSALS

READ (9, END=91) (DATAPT(I), I=1,NCHAN)
IF (DATAPT(1).LT.TIMENEW) THEN
   IF (TIMENEW.GT.86390.0.AND.DATAPT(1).LT.0.0) THEN
      ITIME = 1
   ELSE
      IF (IREV.EQ.1) THEN
         TIME2 = DATAPT(1)
      ELSE
         TIME1 = DATAPT(1)
         TIME2 = DATAPT(1)
         IREV = 1
      ENDIF
   GO TO 90
ENDIF
ENDIF
IF (IREV.EQ.1) WRITE(6,40) TIME1, TIME2
FORMAT (5X, ' TIME REVERSAL BETWEEN TIME =', F12.3, ' AND TIME =', *F12.3 )
IREV = 0

IF (DATAPT(1).GT.(TIMENEW+.01)) THEN
   IF (IGAP.EQ.0) THEN
      WRITE (6,113) DATAPT(1), TIMENEW
      FORMAT(2X, 'TIME GAP BETWEEN TIMES ',F12.3,' AND ',F12.3)
      IGAP = 1
   END IF
   IF (DATAPT(1).GT.(TIMENEW+1.0).AND.
      DATAPT(1).LT.(TIMENEW+3.0)) GO TO 90
   *       IF (DATAPT(1).GT.(TIMENEW+15.).AND.
      *          DATAPT(1).NE.31361.197) GO TO 90

END IF

TIMENEW = DATAPT(1)
IGAP = 0

NUMPTS = NUMPTS + 1
KCNT = KCNT + 1
DATAPT(1) = DATAPT(1) + ITIME * 86400.0
DO 3 I = 2,NCHAN
   IF (DATAPT(I).LT.XMIN(I)) XMIN(I) = DATAPT(I)
   IF (DATAPT(I).GT.XMAX(I)) XMAX(I) = DATAPT(I)
   CONTINUE
WRITE(1) DATAPT(1),DATAPT(2),DATAPT(3),DATAPT(4)
IF (KCNT.EQ.KMAX) THEN
WRITE(6,5) DATAPT(1),DATAPT(2),DATAPT(3),DATAPT(4)
5 FORMAT(1F12.3,3E12.5)
KCNT = 0
ENDIF
C
GO TO 90
91 IF (EOF(9) .NE. 0.) THEN
STOP = DATAPT(1) - 86400.0 * ITIME
WRITE(6,92) START,STOP
92 FORMAT( / 5X,'START TIME =',1E12.5,5X,'STOP TIME =',1E12.5 /)
DO 4 I = 2,NCHAN
WRITE(6,93) I,NAMES(I),XMIN(I),XMAX(I)
93 FORMAT(5X,'CHANNEL',1I2,5X,'NAMES =',1A10,5X,
'MIN VALUE =',1E12.5,5X,'MAX VALUE =',1E12.5 )
4 CONTINUE
ENDIF
WRITE(6,245) NUMPTS
245 FORMAT(5X,'NUMBER OF POINTS ON TAPE 1 =',I8)
C
900 CONTINUE
WRITE(6,801)
801 FORMAT( / 5X,'JTREAD COMPLETED')
STOP
END
PROGRAM HSKPRED (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, *
* TAPE9, TAPE1, TAPE2, TAPE3, TAPE4)

THIS ROUTINE READS AND PRINTS RAW DATA FROM HIRAP TAPES
WRITTEN BY HIRAP PROJECT GROUP - JANUARY 1986

DIMENSION DATAPT(9), NAMES(9), IUNITS(9), HDR(9), XMAX(9), *
* XMIN(9)

CHARACTER*8 NAMES
DATA NAMES/'TIME-SEC','X-FINE ', 'X-COARSE', 'Y-FINE ', *
* 'Y-COARSE', 'Z-FINE ', 'Z-COARSE', 'POS-V ', *
* 'NEG-V '/

THE FOLLOWING IS A LIST OF INPUTS TO THIS PROGRAM

KMAX COUNT AT WHICH EACH POINT IS WRITTEN TO OUTPUT

KMAX = 4
NCHAN = 9
JREV = 0
IGAP = 0
STPTIM = 49255.

READ(9) IDAY, IHOUR, IMIN
WRITE(6,123) IDAY, IHOUR, IMIN
FORMAT(2X, 'IDAY, IHOUR, IMIN = ', 3 (2X, I5)/)

READ(9, END=1) (DATAPT(I), I=1,NCHAN)
WRITE(1) (DATAPT(I), I=1,NCHAN)

DEFINE HEADER IF NECESSARY

CONTINUE

IF (EOF(9) .NE. 0) THEN
WRITE(6,10)
GO TO 7
ENDIF

SET MINIMUM AND MAXIMUM VALUES

DO 2 I = 2,NCHAN
XMIN(I) = DATAPT(I)
XMAX(I) = DATAPT(I)
CONTINUE
ITIME = 0
START = DATAPT(1)
TIMENEW = START - 0.1
KCNT = 0
NUMPTS = 0

MOVE TO DESIRED START AND PRINT RAW DATA UNTIL END OF FILE
IS REACHED ELIMINATING BAD POINTS AND TIME REVERSALS

CONTINUE
READ(9,END=5) (DATAPT(I),I=1,NCHAN)

IF (DATAPT(1).GT.STPTIM) GO TO 108

IF (DATAPT(1) .LT. TIMENEW) THEN
  IF (TIMENEW > 86390.0 .AND. DATAPT(1) .LT. 10.0) THEN
    ITIME = 1
  ELSE
    IF (JREV.EQ.0) THEN
      WRITE(6,13) DATAPT(I)
      JREV = 1
    END IF
    GO TO 3
  ENDIF
ELSE IF (DATAPT(1). EQ. TIMENEW) THEN
  GO TO 3
END IF

IF (DATAPT(1).GT.(TIMENEW+.7)) THEN
  IF (IGAP.EQ.0) THEN
    WRITE(6,113) DATAPT(I), TIMENEW
    IGAP = 1
  END IF
  IF (DATAPT(1) .GT. (TIMENEW+50.) .AND. DATAPT(1) .LT. (TIMENEW+100.)) GOTO 3
END IF

JREV = 0
TIMENEW = DATAPT(1)
IGAP = 0
NUMPTS = NUMPTS + 1
KCNT = KCNT + 1
DATAPT(1) = DATAPT(1) + ITIME * 86400.0
DO 4 I = 2,NCHAN
   IF (DATAPT(I) .LT. XMIN(I)) XMIN(I) = DATAPT(I)
   IF (DATAPT(I) .GT. XMAX(I)) XMAX(I) = DATAPT(I)
   CONTINUE
C
   WRITE(1) (DATAPT(I),I=1,NCHAN)
C
   IF (KCNT .EQ. KMAX) THEN
      WRITE(6,14) (DATAPT(I),I=1,NCHAN)
      KCNT = 0
   ENDIF
C
   GO TO 3
C
5   IF (EOF(9).NE.0) THEN
      STOP = DATAPT(1) - 86400.0 * ITIME
      WRITE (6,15) START,STOP
      DO 6 I = 2,NCHAN
         WRITE(6,16) I,NAMES(I),XMIN(I),XMAX(I)
      CONTINUE
   ENDIF
C
   WRITE (6,17) NUMPTS
C
   CONTINUE
   WRITE(6,18)
   STOP
108  WRITE(6,119) STPTIM
119  FORMAT(2X,'RAN INTO STOP TIME',2X,F12.3)
C
   STOP
10  FORMAT(5X,'EOF FOUND WHEN ATTEMPTING TO READ HEADER')
11  FORMAT(1HL1,4X,'ISN =',I7,5X,'NCHAN =',112,5X,
      '*FOUND HEADER ','8A10')
12  FORMAT(9(2X,A10))
13  FORMAT(5X,'TIME REVERSAL AT TIME =',1E12.5)
14  FORMAT(5F12.3,/,4F12.3)
15  FORMAT(/ 5X,'START TIME =',1E12.5,5X,'STOP TIME =',
      * 1E12.5 /)
16  FORMAT(5X,'CHANNEL',I2,5X,'NAMES =',1A10,5X,
      '*MIN VALUE =',1E12.5,5X,'MAX VALUE =',1E12.5)
17  FORMAT(5X,'NUMBER OF POINTS ON TAPE 1 =',I8)
18  FORMAT(// 5X,'HSKPRED COMPLETED')
19  FORMAT(// 5X,'TIME GAPS')/
20  FORMAT(5X,'TIME GAP BETWEEN TIME =',1F12.3,
      * ' AND TIME =',1F12.3)
21  FORMAT(// 5X,'TIME REVERSALS')/
22  FORMAT(9(2X,A10)/)
END
PROGRAM TкалIB (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE7, TAPE1)

DIMENSION NAMES (10), IUNITS (10), TITLE (8), DATAPT (7), TF (3), TC (3),
* VC(3), TEMP (3)

KMAX1 = 4
KMAX2 = 20
NP = 0
REWIND (7)
READ (5, 52, END=1) SN, T1, T2

CONTINUE
IF (EOF (5) .NE. 0) THEN
WRITE (6, 58)
GO TO 1
ELSE
ISN = SN
WRITE (6, 60) ISN, T1, T2
WRITE (6, 53)
ENDIF
IF (EOF (7) .NE. 0) THEN
WRITE (6, 57)
GO TO 1
ENDIF

KCNT1 = 3
KCNT2 = 19
READ (7, END=5) (DATAPT(J), J=1, 7)
TIME = DATAPT (1)
IF (TIME .LT. T1) GO TO 4
IF (TIME .GT. T2) GO TO 10
KCNT1 = KCNT1 + 1
KCNT2 = KCNT2 + 1
IF (KCNT1 .EQ. KMAX1) THEN
NP = NP + 1
TF (1) = DATAPT (2)
TC (1) = DATAPT (3)
TF (2) = DATAPT (4)
TC (2) = DATAPT (5)
TF (3) = DATAPT (6)
TC (3) = DATAPT (7)
CALL TVCCALC TO GET VOLTAGES AND TEMPERATURES
ENDIF
IF (SN .EQ. 1.0) CALL TVCCAL1 (TF, TC, VC, TEMP)
IF (SN .EQ. 2.0) CALL TVCCAL2 (TF, TC, VC, TEMP)
WRITE (1) TIME, (VC(I), I=1, 3), (TEMP(I), I=1, 3)
KCNT1 = 0
IF (KCNT2 .GE. KMAX2) THEN
WRITE (6, 54) TIME, (VC(I), I=1, 3), (TEMP(I), I=1, 3)
KCNT2 = 0
ENDIF
ENDIF

GO TO 4
CONTINUE
WRITE (6, 57)
CONTINUE
WRITE (6, 59) NP
C
STOP
C
FORMAT (3F8.0)
C
FORMAT (/ 4X, 8HTIME, SEC, 6X, 3HVCX, 9X, 3HVCY, 9X, 3HVCZ, 7X, 6HTEMP-X, *
6X, 6HTEMP-Y, 6X, 6HTEMP-Z, /)
C
FORMAT (1F12.3, 6F12.5)
C
FORMAT (/ 5X, 42HEND OF FILE FOUND ON TAPE 7 - PROGRAM STOP /)
C
FORMAT (/ 5X, 42HEND OF FILE FOUND ON TAPE 5 - PROGRAM STOP /)
C
FORMAT (/ 10X, 25HTOTAL NUMBER OF POINTS = , II6 *
5X, 12HHIRAP S/N 00, II1, :
5X, 7HFILE = , 5HTAPE7, 5X, 5HT1 = , *
1F10.2, 9X, 5HT2 = , 1F10.2 /)
END
SUBROUTINE TVCCALI (TF, TC, VC, TEMP)
C
THIS ROUTINE COMPUTES TEMPERATURE FROM FINE AND COARSE
TEMPERATURE COUNTS OF RECALIBRATED HIRAP SN 001
C
DIMENSION A(3), B(3), G(3), VM(10, 3), TF(3), TC(3), VC(3), TEMP(3)

DATA A / 31.74, 30.92, 30.46 /
DATA B / 24.46, 24.51, 24.66 /
DATA G / 7.181, 7.186, 7.192 /
DATA VM / -0.0014, -0.0014, 0.6127, 1.2268, 1.8407, 2.4547, *
3.0685, 3.6823, 4.2963, 4.2963, 0.0000, 0.0000, 0.6142, 1.2283,
* 1.8424, 2.4564, 3.0704, 3.6843, 4.2981, 4.2981, -0.0002,
* -0.0002, 0.6140, 1.2281, 1.8424, 2.4566, 3.0705, 3.6845,
* 4.2984, 4.2984 /

MAX TEMPERATURE RANGE = 5000 MILIVOLTS = 5 VOLTS
MAX TEMPERATURE COUNTS RANGE = 250
SCALE FACTOR, SF = 5000/(2 ** 8 - 1) = 5000/255
IN ACCORDANCE WITH INITIAL HIRAP CALIBRATIONS, BOTH THE
FINE AND COARSE TEMPERATURE COUNTS SHOULD BE REDUCED BY 3.
SF = 5.0/250.0

COMPUTE VOLTAGES AND TEMPERATURE FOR X, Y, AND Z AXES

DO 1 I = 1, 3
TFNEW = (TF(I) - 3.0) * SF
TCNEW = (TC(I) - 3.0) * SF
M = 0.5 + (1.6 * TCNEW) + 1.0
IF (M.LT.1) M = 1
IF (M.GT.9) M = 9
VC1 = VM(M, I) + TFNEW/G(I)
VC2 = VM(M+1, I) + TFNEW/G(I)
A1 = ABS (VC1 - TCNEW)
A2 = ABS (VC2 - TCNEW)
VC(I) = VC1
IF(A2.LT.AI) VC(I) = VC2
TEMP(I) = A(I) + B(I) * VC(I)
CONTINUE
RETURN
END
SUBROUTINE TVCCAL2 (TF, TC, VC, TEMP)

THIS ROUTINE COMPUTES TEMPERATURE FROM FINE AND COURSE TEMPERATURE COUNTS OF RECALIBRATED HIRAP S/N 002

DIMENSION A(3), B(3), G(3), VM(10, 3), TF(3), TC(3), VC(3), TEMP(3)

DATA A / 24.60, 24.53, 24.60 / 
DATA B / 25.96, 25.72, 25.72 / 
DATA G / 7.221, 7.1798, 7.187 / 
DATA VM / 0.0014, 0.0014, 0.6158, 1.2302, 1.8463, 2.4609, 3.0759, * 3.6192, 4.3047, 4.3047, -0.0005, -0.0005, 0.6141, 1.2287, 1.8434, * 2.4579, 3.0726, 3.6874, 4.3019, 4.3019, -0.0001, -0.0001, 0.6145, * 1.2291, 1.8437, 2.4582, 3.0729, 3.6875, 4.3021, 4.3021 / 

MAX TEMPERATURE RANGE = 5000 MILLIVOLTS = 5 VOLTS
MAX TEMPERATURE COUNTS RANGE = 250
SCALE FACTOR, SF = 5000/(2 ** 8 - 1) = 5000/255
IN ACCORDANCE WITH INITIAL HIRAP CALIBRATIONS, BOTH THE FINE AND COURSE TEMPERATURE COUNTS SHOULD BE REDUCED BY 3.

SF = 5.0/250.0

COMPUTE VOLTAGES AND TEMPERATURES FOR X, Y, AND Z AXES

DO 1 I = 1, 3
TFNEW = (TF(I) - 3.0) * SF
TCNEW = (TC(I) - 3.0) * SF
M = 0.5 + 1.6 * TCNEW + 1
IF (M .LT. 1) M = 1
IF (M .GT. 9) M = 9
VC1 = VM(M, I) + TFNEW/G(I)
VC2 = VM(M+1, I) + TFNEW/G(I)
A1 = ABS(VC1 - TCNEW)
A2 = ABS(VC2 - TCNEW)
VC(I) = VC1
IF (A2 .LT. A1) VC(I) = VC2
TEMP(I) = A(I) + B(I) * VC(I)
1 CONTINUE

RETURN
END

5401.0 6057.0
PROGRAM JTRATES (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE7, TAPE8, TAPE9, TAPE10, TAPE1, TAPE2)

THIS IS THE HIRAP PROGRAM THAT CALCULATES ACCELERATIONS DUE TO ANGULAR VELOCITIES AND THEN INTERPOLATES TO MATCH THE HIRAP SCIENCE DATA INTERATION RATE. THE INDUCED ACCELERATIONS ARE THEN ALGEBRAICALLY SUBTRACTED FROM THE RAW MICRO-G FILE. THE CG CORRECTED MICRO-GS ARE PLOTTED IF SO DESIRED.

WRITTEN BY: JOSEPH S. ROWLEY, JUNE 1983
AND MODIFIED BY HIRAP PROJECT GROUP, SEPT 1986

DIMENSION DATA (100), DATA7 (100),
* NAMES (100), IUNITS (100), HDR (8),
* NAMES7 (100), IUNITS7 (100), HDR7 (8),
* XMOUNT (3), YMOUNT (3), ZMOUNT (3), XBAR (3), YBAR (3), ZBAR (3)

REAL NEWTIM, NEWX, NEWY, NEWZ

THE FOLLOWING IS A LIST OF INPUTS TO THIS PROGRAM

IFLT STS FLIGHT NUMBER
IPLTMG = 1 TO PLOT MICRO-GS ON +/- 1000 SCALE
= 0 NO PLOTTING
T1 START TIME FOR DATA CALCULATION (SEC)
T2 STOP TIME FOR DATA CALCULATION
DAY DAY OF DESCENT
XEPOCH START TIME OF RATE DATA (XBET FILE)
ALTSTRT START ALT FOR MICRO-G PLOT (KM)
ALTSTOP STOP ALT FOR PLOT (KM)
XLENG LENGTH OF X-AXIS PLOT GRID (IN)
XCG VEHICLE CENTER OF GRAVITY LOCATION IN THE ORBITER
YCG STRUCTURAL REFERENCE SYSTEM (IN)
ZCG
XMOUNT HIRAP LOCATION ON ORBITER (IN)
YMOUNT
ZMOUNT

YLENG          LENGTH OF Y-AXIS PLOT GRID

TAPES USED IN THIS CODE:

INPUT:

TAPE 7    UNCORRECTED MICRO-GS (T,A,X,Z,Y)
TAPE 9    XBET FILE

OUTPUT:

TAPE 8    CORRECTED MICRO-GS (T,A,X,Z)

PRINT 800

800 FORMAT(1H1,4X,'RATES STARTED')

810 FORMAT(2I10)
811 FORMAT(4F10.0)
820 FORMAT(// 5X,34HEOF FOUND ON TAPE 5 - PROGRAM STOP)
821 FORMAT(// 5X,10HIFLT = STS ,1I2,5X,9HIPLTMG = ,1I2)
822 FORMAT(// 5X,10HALTSTRT = ,1F10.2,5X,10HALTSTOP = ,1F10.2)
823 FORMAT(// 5X,5HT1 = ,1F10.2,/ 5X,5HT2 = ,1F10.2,
*     / 5X,13HXBET EPOCH = ,1F10.2,/ 5X,6HDAY = ,1F4.0)
824 FORMAT(// 5X,6HXCG = ,1F10.2,/ 5X,6HYCG = ,1F10.2,
*     / 5X,6HZCG = ,1F10.2 //)
825 FORMAT(// 5X,10HIFLT = STS ,1I2,5X,9HIPLTMG = ,1I2)
826 FORMAT(// 5X,10HALTSTRT = ,1F10.2,5X,10HALTSTOP = ,1F10.2)
827 FORMAT(// 5X,5HT1 = ,1F10.2,/ 5X,5HT2 = ,1F10.2,
*     / 5X,13HXBET EPOCH = ,1F10.2,/ 5X,6HDAY = ,1F4.0)
828 FORMAT(// 5X,6HXCG = ,1F10.2,/ 5X,6HYCG = ,1F10.2,
*     / 5X,6HZCG = ,1F10.2 //)
DEGRAD = 180./3.1415927
IF (FLT.EQ.32) THEN
   KMAX = 350
ELSE
   KMAX = 350
ENDIF
KCNT = KMAX - 1

USE EXACT HIRAP LOCATION AND CG LOCATION AT ENTRY INTERFACE TO OBTAIN THE MOMENT ARMS IN FEET.

X-AXIS ACCELEROMETER

XMOUNT(1) = 1216.97
YMOUNT(1) = - 5.50
ZMOUNT(1) = 298.55

Y-AXIS ACCELEROMETER

XMOUNT(2) = 1218.27
YMOUNT(2) = - 5.50
ZMOUNT(2) = 298.55

Z-AXIS ACCELEROMETER

XMOUNT(3) = 1218.27
YMOUNT(3) = - 5.50
ZMOUNT(3) = 296.12

DO 111 I=1,3
   XBAR(I) = -(XMOUNT(I) - XCG)/12.
   YBAR(I) = (YMOUNT(I) - YCG)/12.
   ZBAR(I) = -(ZMOUNT(I) - ZCG)/12.
   WRITE(6,826) I, XBAR(I), YBAR(I), ZBAR(I)
111 CONTINUE

826 FORMAT(/ 5X, 5HXBAR(,II1,4H) = ,1F10.2,
*       / 5X, 5HYBAR(,II1,4H) = ,1F10.2,
*       / 5X, 5HZBAR(,II1,4H) = ,1F10.2 //)

READ TAPE HEADERS

IFILE = 9
READ(9, END=600) ISN, NCHAN, (NAMES(I), I=1,NCHAN),
WRITE (6,887) IFILE
WRITE (6,888) ISN, NCHAN, HDR

IF (EOF (9) .NE. 0) THEN
WRITE (6,802) IFILE
FORMAT (/// 5X,38HEOF FOUND WHEN READING HEADER ON FILE ,1I2)
GO TO 999
ENDIF

IFILE = 7
READ (7, END=601) ISN7, NCHAN7, (NAMES7(I), I=1, NCHAN7),
(IUNITS7(I), I=1, NCHAN7), (HDR7(I), I=1, 8)

ISN7 = 1
NCHAN7 = 5
WRITE (6,887) IFILE
WRITE (6,888) ISN7, NCHAN7, HDR7
WRITE (6,889) (NAMES7(I), I=1, NCHAN7)
WRITE (6,890) (IUNITS7(I), I=1, NCHAN7)

FORMAT (/// 5X,5HTAPE ,1I2)
FORMAT (///5X,6HISN = ,1I5,5X,8HNCHAN = ,1I5, / 5X,
*        15HFOUND HEADER : ,8A10 /)
FORMAT (4(2X,A10))
FORMAT (4(2X,A10) //)

IF (EOF (7) .NE. 0) THEN
WRITE (6,802) IFILE
GO TO 999
ENDIF

WRITE (8) ISN7, NCHAN7, (NAMES7(I), I=1, NCHAN7),
*        (IUNITS7(I), I=1, NCHAN7), (HDR7(I), I=1, 8)

READ ANGULAR VELOCITIES AND ACCELERATIONS

READ (9, END=450) (DATA(J), J=1, NCHAN)
IF (EOF (9) .NE. 0) THEN
PRINT 465, TIME
FORMAT (/// 3X,33HEOF FOUND ON TAPE 9 AFTER TIME = ,1F12.3 //)
GO TO 200
ENDIF
TIME = DATA(1) + XEPOCH
IF (TIME.LT.T1) GO TO 40

P = DATA(49)
Q = DATA(50)
R = DATA(51)
PDOT = 0
QDOT = 0
RDOT = 0

CONVERT P, Q, R TO RAD/SEC IF NEEDED

P = P/DEGRAD
Q = Q/DEGRAD
R = R/DEGRAD

ANGACLX = XBAR(1) * (Q * Q + R * R) - YBAR(1) * (P * Q - RDOT) -
          ZBAR(1) * (P * R + QDOT)

ANGACLY =-XBAR(2) * (P * Q + RDOT) + YBAR(2) * (P * P + R * R) -
          ZBAR(2) * (Q * R - PDOT)

ANGACLZ =-XBAR(3) * (P * R - QDOT) - YBAR(3) * (Q * R + PDOT) +
          ZBAR(3) * (P * P + Q * Q)

CONVERT FT/SEC2 TO MICRO-GS

DX = ANGACLX / 32.174 * 1E06
DY = ANGACLY / 32.174 * 1E06
DZ = ANGACLZ / 32.174 * 1E06

WRITE DATA TO TAPE10

WRITE(10) TIME, DX, DZ, DY

IF (TIME.GT.T2) GO TO 200
GO TO 40

CONTINUE

REWIND 10
TIMSCI = 0

NEWTIM = NEWX = NEWY = NEWZ = 0
READ(10,END=66) TIME,DX,DZ,DY
IF(EOF(10).NE.0) THEN
  PRINT 67,TIME
  FORMAT ('/5X,34HEOF FOUND ON TAPE 10 AFTER TIME = ,1F12.3/')
  GO TO 900
ENDIF

OLDX = NEWX
OLDY = NEWY
OLDZ = NEWZ
OLDTIM = NEWTIM
NEWX = DX
NEWY = DY
NEWZ = DZ
NEWTIM = TIME

IF((TIMSCI - OLDTIM).GT.0.995) GO TO 65

READ(7,END=51) (DATA7(J),J=1,NCHAN7)

IF(EOF(7).NE.0) THEN
  PRINT 466,DATA7(1)
  FORMAT ('/5X,33HEOF FOUND ON TAPE 7 AFTER TIME = ,1F12.3/')
  GO TO 900
ENDIF

TIMSCI = DATA7(1)
ALT = DATA7(2)
XMG = DATA7(3)
ZMG = DATA7(5)

IF(TIMSCI.LT.OLDTIM) GO TO 50
IF((TIMSCI - OLDTIM).GT.0.995) GO TO 65

STRAIGHT LINE INTERPOLATION TO MATCH HIRAP DATA RATE

XANGA = ((TIMSCI - OLDTIM) / (NEWTIM - OLDTIM))
  * (NEWX - OLDX) + OLDX

YANGA = ((TIMSCI - OLDTIM) / (NEWTIM - OLDTIM))
  * (NEWY - OLDY) + OLDY

ZANGA = ((TIMSCI - OLDTIM) / (NEWTIM - OLDTIM))
* (NEWZ - OLDZ) + OLDZ

CORRECT MICRO-GS TO VEHICLE CG

XMG = XMG + XANGA
ZMG = ZMG + ZANGA

DX = -XANGA
DZ = -ZANGA

WRITE CALCULATED DATA TO TAPE8

WRITE(8) TIMSCI, ALT, XMG, ZMG
WRITE(1) TIMSCI, ALT, DX, DZ
KCNT = KCNT + 1
IF (KCNT.GE.KMAX) THEN
  ALT = ALT/1000.
  WRITE(6,777) TIMSCI, ALT, XMG, ZMG, DX, DZ
  KCNT = 0
ENDIF

777 FORMAT(6F12.3)

IF (TIMSCI.GT.T2) GO TO 900
GO TO 50

900 CONTINUE

IF (IPLTMG.EQ.1) THEN
  CALL PSEUDO
RESTART(1)

PLOT X-AXIS INDUCED ACCELERATIONS

IY = 1
CALL CALPLT(2.0,2.0,-3)
CALL DRAW(ALTSTR, ALTSTOP, IY, FLT, SN, DAY, XLENG, YLENG)
READ(1,END=129) TIME, ALT, DX, DZ
ALT = ALT/1000.
IF (ALT.GT.ALTSTR) GO TO 128
TPlot = (ALTSTR - ALT) * (XLENG/DELAT)
YPlot = (DX + 10.0) * (YLENG/20.0)
IF (YPlot.GT.0.AND.YPlot.LT.YLENG) THEN
  CALL PNTPLT(TPLOT, YPLOT, 21, 1)
ENDIF
129 IF (EOF(1).NE.0.OR.TIME.GT.T2) GO TO 130
GO TO 128
CONTINUE
GO TO 133
REWIND(1)

PLOT Y-AXIS INDUCED ACCELERATIONS

IY = 2
CALL NFRAME
CALL CALPLT(2.0, 2.0, -3)
CALL DRAW(ALTSTRT, ALTSTOP, IY, FLT, SN, DAY, XLENG, YLENG)
READ(1, END=132) TIME, ALT, DX, DZ
ALT = ALT/1000.
IF (ALT.GT. ALTSTRT) GO TO 131
T PLOT = (ALTSTRT - ALT) * (XLENG/DELALT)
YPLOT = (DY + 10.0) * (YLENG/20.0)
IF (YPLOT.GT.0 AND YPLOT.LT.YLENG) THEN
CALL PNTPLT(TPLOT, YPLOT, 21, 1)
ENDIF
IF(EOF(1).NE.0.OR.TIME.GT.T2) GO TO 133
GO TO 131
CONTINUE
REWIND(1)

PLOT Z-AXIS INDUCED ACCELERATIONS

IY = 3
CALL NFRAME
CALL CALPLT(2.0, 2.0, -3)
CALL DRAW(ALTSTRT, ALTSTOP, IY, FLT, SN, DAY, XLENG, YLENG)
READ(1, END=135) TIME, ALT, DX, DZ
ALT = ALT/1000.
IF (ALT.GT. ALTSTRT) GO TO 134
T PLOT = (ALTSTRT - ALT) * (XLENG/DELALT)
YPLOT = (DZ + 10.0) * (YLENG/20.0)
IF (YPLOT.GT.0 AND YPLOT.LT.YLENG) THEN
CALL PNTPLT(TPLOT, YPLOT, 21, 1)
ENDIF
IF(EOF(1).NE.0.OR.TIME.GT.T2) GO TO 136
GO TO 134
CONTINUE
GO TO 36
REWIND(1)
READ(1)

PLOT X-AXIS MICROGS

IY = 1
CALL NFRAME
CALL CALPLT(2.0, 2.0, -3)
CALL DRAW(ALTSTRT, ALTSTOP, IY, FLT, SN, DAY, XLENG, YLENG)
READ(1, END=29) TIME, XMG, ZMG, YMG
IF (ALT.GT.ALTSTRT) GO TO 28

TPLOT = (TIME - ALTSTRT) * (XLENG/DELAULT)
YPLOT = (XMG + 1000.0) * (YLENG/2000.0)

IF (YPLOT.GT.0.AND.YPLOT.LT.YLENG) THEN
   CALL PNTPLT(TPLOT,YPLOT,21,1)
ENDIF

IF (EOF(1).NE.0.OR.TIME.GT.T2) GO TO 30
GO TO 28

CONTINUE
REWIND(1)
READ(1)

C

PLOT Y-AXIS MICRO-GS
C

IY = 2
CALL NFRAME
CALL CALPLT(2.0,2.0,-3)
CALL DRAW1(ALTSTRT,ALTSTOP,IY,FLT,SN,DAY,XLENG,YLENG)

READ(1,END=32) TIME,XMG,ZMG,YMG

IF (ALT.GT.ALTSTRT) GO TO 31
TPLOT = (TIME - ALTSTRT) * (XLENG/DELAULT)
YPLOT = (YMG + 1000.0) * (YLENG/2000.0)

IF (YPLOT.GT.0.AND.YPLOT.LT.YLENG) THEN
   CALL PNTPLT(TPLOT,YPLOT,21,1)
ENDIF

IF (EOF(1).NE.0.OR.TIME.GT.T2) GO TO 33
GO TO 31

CONTINUE
REWIND(1)
READ(1)

C

PLOT Z-AXIS MICRO-GS
C

IY = 3
CALL NFRAME
CALL CALPLT(2.0,2.0,-3)
CALL DRAW1(ALTSTRT,ALTSTOP,IY,FLT,SN,DAY,XLENG,YLENG)

READ(1,END=35) TIME,XMG,ZMG

IF (ALT.GT.ALTSTRT) GO TO 34
TPLOT = (TIME - ALTSTRT) * (XLENG/DELAULT)
YPLOT = (ZMG + 1000.0) * (YLENG/2000.0)

IF (YPLOT.GT.0.AND.YPLOT.LT.YLENG) THEN
   CALL PNTPLT(TPLOT,YPLOT,21,1)
ENDIF

IF (EOF(1).NE.0.OR.TIME.GT.T2) GO TO 36
GO TO 34

CONTINUE
REWIND(1)
READ(1)

C

END

CONTINUE
PRINT 801

68
FORMAT(// 5X, 'RATES COMPLETED')
STOP
END

SUBROUTINE DRAW(ALTSTRT, ALTSTOP, IY, FLT, SN, DAY, XLENG, YLENG)

THIS ROUTINE DRAWS GRID FOR HIRAP ACCELEROMETER VS TIME PLOTS

IY = 1 FOR X AXIS
     = 2 FOR Y AXIS
     = 3 FOR Z AXIS

DIMENSION TITLE2 (4), TITLE3 (3), YCHAR (3), LABEL1 (6),
     *      LABEL2 (2)

DATA TITLE2 / 10H HIRAP S/N, 10H00 ACCELEROMETER DATA,
     *      10H TA-STS /
DATA TITLE3 / 10H INDUCTED ACCELERATION, 10H S /
DATA YCHAR / 1HX, 1HY, 1HZ /
DATA LABEL1 / 10H AXIS MIC, 10H RO-GS AS A, 10H FUNCTION ,
     *      10H O F ALTITUDE, 10H E, KM - DA, 10H Y /
DATA LABEL2 / 10H FILE: CGI, 10H3II /

DELALT = (ALTSTRT - ALTSTOP)
Y2 = YLENG/2.
CALL CALPLT(0.0, YLENG, 2)
CALL CALPLT(XLENG, 0.0, 3)
CALL CALPLT(0.0, 0.0, 2)
CALL CALPLT(0.0, Y2, 3)
CALL CALPLT(XLENG, Y2, 2)

DRAW TICK MARKS ON X AXIS

NX = DELALT/2
XBAR = 0.0
KCNT = 0
DELX = XLENG/NX
IEND = NX
DO 1 J = 1, IEND
    KCNT = KCNT + 1
    XBAR = DELX * J
    IF (KCNT.EQ.10) THEN
        DELY = 0.14
        KCNT = 0
    ELSE
        DELY = 0.09
    ENDIF
    CALL CALPLT(XBAR, DELY, 2)
1 CONTINUE

DRAW TICK MARKS ON Y AXIS

NY = 20
KCNT = 0
XBAR = 0.0
DELY = YLENG/NY
IEND = NY
DO 3 J = 1, IEND
KCNT = KCNT + 1
IF (KCNT.EQ.2) THEN
   DX = 0.14
   KCNT = 0
ELSE
   DX = 0.09
ENDIF
YBAR = DELY * J
CALL CALPLT(XBAR, YBAR, 3)
CALL CALPLT(DX, YBAR, 2)
CONTINUE

LABEL X AXIS

DELX = DELX * 10
YBAR = -0.25
XVAL = ALTSTRT + 20.0
IEND = NX/10 + 1
DO 5 I = 1, IEND
XST = -0.20
XBAR = XST + DELX * (I-1)
CALL NUMBER(XBAR, YBAR, 0.10, XVAL, 0.0, -1)
CONTINUE

LABEL Y AXIS

YST = -0.05
DELY = DELY * 2
IEND = NY/2 + 1
DO 6 I = 1, IEND
YVAL = -10.0 + 2.0 * (I-1)
XST = -0.45
IF(YVAL.LE.-1000) XST = -0.55
IF(YVAL.EQ.0.0) XST = -0.17
IF(YVAL.GT.0.0) XST = -0.39
IF(YVAL.GE.1000) XST = -0.49
CALL NUMBER(XST, YBAR, 0.10, YVAL, 0.0, -1)
CONTINUE

XBAR = 1.0
YBAR = YLENG + 1.0
CALL CHARACT(XBAR, YBAR, 0.10, TITLE2, 0.0, 40)
XBAR = XBAR + 1.12
CALL NUMBER(XBAR, YBAR, 0.10, SN, 0.0, -1)
XBAR = XBAR + 2.48
CALL NUMBER(XBAR, YBAR, 0.10, FLT, 0.0, -1)
XBAR = 1.0
YBAR = YBAR - 0.25
CALL CHARACT(XBAR, YBAR, 0.10, TITLE3, 0.0, 30)
YBAR = YBAR - 0.25
CALL CHARACT(XBAR,YBAR,0.10,YCHAR(IY),0.0,1)
CALL CHARACT(XBAR,YBAR,0.10,LABEL1,0.0,60)
XBAR = XBAR + 4.80
CALL NUMBER(XBAR,YBAR,0.10,DAY,0.0,-1)
XBAR = 1.0
YBAR = YBAR - 0.25
CALL CHARACT(XBAR,YBAR,0.10,LABEL2,0.0,20)
RETURN
END

SUBROUTINE DRAW1(ALTSTRT, ALTSTOP, IY, FLT, SN, DAY, XLENG, YLENG)

THIS ROUTINE DRAWS GRID FOR HIRAP ACCELEROMETER VS TIME PLOTS

IY = 1 FOR X AXIS
   = 2 FOR Y AXIS
   = 3 FOR Z AXIS

DIMENSION TITLE2(4), TITLE3(3), YCHAR(3), LABEL1(6),
*  LABEL2 (2)

DATA TITLE2 / 10HHIRAP S/N ,10HO0 ACCELE,10HROMETER DA,  
*  10HTA - STS /
DATA TITLE3 / 10HCORRECTED ,10HTO VEHICLE,10H CG  /
DATA YCHAR / 1HX, 1HY, 1HZ /
DATA LABEL1 / 10H AXIS MIC,10HRO-GS AS A,10H FUNCTION ,  
*  10HOF SECONDS,10H, GMT - DA,10HY /
DATA LABEL2 / 10HFILE : CGI,10H3II /

DELALT = (ALTSTRT - ALTSTOP)
CALL CALPLT(0.0,YLENG,2)
CALL CALPLT(XLENG,0.0,3)
CALL CALPLT(0.0,0.0,2)

DRAW TICK MARKS ON X AXIS

NX = DELALT/100
YBAR = 0.0
KCNT = 0
DELX = XLENG/NX
IEND = NX
DO 1 J = 1,IEND
   KCNT = KCNT + 1
   XBAR = DELX * J
   IF(KCNT.EQ.5) THEN
      DELY = 0.14
      KCNT = 0
   ELSE
      DELY = 0.09
   ENDIF
   CALL CALPLT(XBAR,YBAR,3)
   CALL CALPLT(XBAR,DELY,2)
1 CONTINUE
DRAW TICK MARKS ON Y AXIS

NY = 40
KCNT = 0
XBAR = 0.0
DELY = YLENG/NY
IEND = NY
DO 3 J = 1, IEND
  KCNT = KCNT + 1
  IF (KCNT.EQ.4) THEN
    DX = 0.14
    KCNT = 0
  ELSE
    DX = 0.09
  ENDIF
  YBAR = DELY * J
  CALL CALPLT (XBAR, YBAR, 3)
  CALL CALPLT (DX, YBAR, 2)
CONTINUE

LABEL X AXIS

DELX = DELX * 5
YBAR = -0.25
XVAL = ALTSTRT - 500.0
IEND = NX/5 + 1
DO 5 I = 1, IEND
  XST = -0.20
  XBAR = XST + DELX * (I-1)
  CALL NUMBER (XBAR, YBAR, 0.10, XVAL, 0.0, -1)
CONTINUE

LABEL Y AXIS

YST = -0.05
DELY = DELY * 4
IEND = NY/4 + 1
DO 6 I = 1, IEND
  YBAR = YST + DELY * (I-1)
  YVAL = -1000.0 + 200.0 * (I-1)
  XST = -0.45
  IF (YVAL.LE.-1000) XST = -0.55
  IF (YVAL.EQ.0.0) XST = -0.17
  IF (YVAL.GT.0.0) XST = -0.39
  IF (YVAL.GE.10000) XST = -0.49
  IF (YVAL.LE.10000) XST = -0.49
  CALL NUMBER (XST, YBAR, 0.10, YVAL, 0.0, -1)
CONTINUE

XBAR = 1.0
YBAR = YLENG + 1.0
CALL CHARACT (XBAR, YBAR, 0.10, TITLE2, 0.0, 40)
XBAR = XBAR + 1.12
CALL NUMBER(XBAR, YBAR, 0.10, SN, 0.0, -1)
XBAR = XBAR + 2.48
CALL NUMBER(XBAR, YBAR, 0.10, FLT, 0.0, -1)
XBAR = 1.0
YBAR = YBAR - 0.25
CALL CHARACT(XBAR, YBAR, 0.10, TITLE3, 0.0, 30)
YBAR = YBAR - 0.25
CALL CHARACT(XBAR, YBAR, 0.10, YCHAR(IY), 0.0, 1)
CALL CHARACT(XBAR, YBAR, 0.10, LABEL1, 0.0, 60)
XBAR = XBAR + 4.80
CALL NUMBER(XBAR, YBAR, 0.10, DAY, 0.0, -1)
XBAR = 1.0
YBAR = YBAR - 0.25
CALL CHARACT(XBAR, YBAR, 0.10, LABEL2, 0.0, 40)
RETURN
END

6      0
290.0  85.0  8.0  5.0
65400.0 68200.0 64600.0 99.0
1101.2 0.3 371.5
ROUTE (OUTPUT, DEF, DC=LP, ST=RHR, UN=101385)
PROGRAM ZPRESS (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE9)

THIS PROGRAM FINDS THE ZERO REFERENCE VALUES FOR THE 44 CHANNELS
CONTAINED ON THE REACTION CONTROL SYSTEM THRUST CHAMBERS DATA TAPE.
THESE VALUES ARE USED AS INPUTS TO THE PROGRAM THRUST.

WRITTEN BY: ROBIN WADDELL    FEBRUARY 1984
MODIFIED BY: JOSEPH S. ROWLEY    MARCH 1985

DIMENSION DATAPT(100), DIFVAL(100,100), SAMVAL(100,100)
INTEGER I,L,NAMES(100),IUNITS(100), HDR(8), NDIFPTS(100), FLAG, .FAKE(100)

THE FOLLOWING ARE INPUTS TO THIS PROGRAM
STOP       X-AXIS SATURATION TIME

INPUTS GO HERE
STOP=69553.000

FLAG=0
DATAPT(1) = 0.0

DO 10 I=1,50
  NDIFPTS(I)=0
10 CONTINUE

PRINT 105
105 FORMAT(1H1, "ZPRESS PROCEDURE STARTED")
PRINT 2, STOP
2 FORMAT(IX, "STOP=", F6.0)

READ HEADER AND PRINT NAMES AND IUNITS
READ (9, END=120) ISN, NCHAN, (NAMES(I), I=1, NCHAN),
   .(IUNITS(I), I=1, NCHAN), (HDR(I), I=1, 8)
120   IF (EOF(9) .NE. 0) THEN
   PRINT 125
125   FORMAT(/,"EOF FOUND WHEN ATTEMPTING TO READ HEADER")
   GO TO 210
ENDIF

READ (9, END=135) (DATAPT(K), K=1, NCHAN)
135   IF (EOF(9) .NE. 0) THEN
   PRINT 15
15    FORMAT(/,"END OF FILE WHEN READING DATA")
   IF (STOP .GT. DATAPT(1)) GO TO 110
   GO TO 210
ENDIF
   IF (STOP .LE. DATAPT(1)) GO TO 210

IF FIRST PASS INCLUDE ALL VAULES
   IF (FLAG .EQ. 0) THEN
   DO 101 I=3,46
   DIFVAL(I,1)=DATAPT(I)
   SAMVAL(I,1)=1
   NDIFPTS(I)= NDIFPTS(I)+1
101      CONTINUE
   FLAG=FLAG+1
READ PRESSURES FOR ALL CHANNELS

WRITE (6, 126) ISN, NCHAN, HDR
PRINT 303, (NAMES(I), I=1,10)
PRINT 303, (NAMES(I), I=11,20)
PRINT 303, (NAMES(I), I=21,30)
PRINT 303, (NAMES(I), I=31,40)
PRINT 303, (NAMES(I), I=41,NCHAN)
PRINT 304
PRINT 303, (IUNITS(I), I=1,10)
PRINT 303, (IUNITS(I), I=11,20)
PRINT 303, (IUNITS(I), I=21,30)
PRINT 303, (IUNITS(I), I=31,40)
PRINT 303, (IUNITS(I), I=41,NCHAN)
PRINT 304
126   FORMAT(1X,6HISN = ,1I4,5X,8HNCHAN = ,1I4,/ 1X,
   * 15HFOUND HEADER : ,8A10 //)
303   FORMAT(1X,10(A10))
304   FORMAT(/)
C C C C C
C C C C
ERROR CODES (1E+37 OR GREATER) AND PressURES GREATER
THAN 5 PSIA NOT CONSIDERED AS REALISTIC ZERO REFERENCE VALUES
C
50 DO 102 I=3,46
FAKE(I)=NDIFPTS(I)
DO 100 J=1,FAKE(I)
IF(DATAPT(I) .GE. 1.E+37) GO TO 30
IF(DATAPT(I) .GT. 5) GO TO 102
30 IF(DATAPT(I) .EQ. DIFVAL(I,J)) THEN
SAMVAL(I,J)=SAMVAL(I,J)+1
GO TO 102
ELSE
IF(J .EQ. NDIFPTS(I)) THEN
NDIFPTS(I)=NDIFPTS(I)+1
DIFVAL(I,NDIFPTS(I))=DATAPT(I)
SAMVAL(I,NDIFPTS(I))=1
ENDIF
ENDIF
CONTINUE
CONTINUE
GO TO 40
C C C
210 DO 300 I=3,46
PRINT 9,I
9 FORMAT(1X,"CHANNEL NUMBER=",I4)
DO 301 L=1,NDIFPTS(I)
PRINT *,DIFVAL(I,L),SAMVAL(I,L)
301 CONTINUE
PRINT 302
302 FORMAT(/)
300 CONTINUE
C C
PRINT 400
400 FORMAT(/,1X,"ZPRESS PROCEDURE COMPLETED")
STOP
END
PROGRAM THRUST (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE9, TAPE7, TAPE10, TAPE11)

THIS ROUTINE FINDS FIRING TIMES FOR ORBITER RCS THRUSTERS NEO1020 WITH TCOD924, THRUS924
WRITTEN BY: JOSEPH S. ROWLEY, JUNE 1983

REAL STOP, PRESS (100), ZERO (50), START, LAG, TCLAG

INTEGER NAMES1 (100), IUNITS1 (100), HDR (8), NCHAN1, NAMES (100), IUNITS (100), HDR1 (8), NCHAN

THE FOLLOWING IS A LIST OF INPUTS TO THIS PROGRAM

START      START TIME FOR DATA CALCULATION
STOP       STOP TIME FOR DATA CALCULATION
LAG        LAG TIME REQUIRED TO ENLARGE FIRING INTERVAL
TCLAG      TIME THAT LAG VALUE IS CHANGED
ZERO       ZERO FOR EACH CHAMBER

PRINT 800
800 FORMAT (/,,1X,'THRUST STARTED')

DO 22 I=1,50
ZERO (I) = 0
22 CONTINUE

INPUTS GO HERE

START=00000
STOP=56714.
LAG=10.
TCLAG=66280
ZERO (5) = 1.6
ZERO (10) = .8
ZERO(12) = .8
ZERO(22) = .8
ZERO(28) = .8
ZERO(31) = .9
ZERO(38) = 1E37
ZERO(40) = 1E37
ZERO(42) = .8

WRITE TAPE HEADER ON TAPE8 AND TAPE10

ISNI=1
NCHAN=2
NAMESI(1)=4HTIME
NAMESI(2)=10HRCS FIRING
IUNITSI(1)=7HSECONDS
IUNITSI(2)=1H
HDR1(1)=10H OEX-HIRAP
HDR1(2)=10H RCS
HDR1(3)=10H THRUSTING
HDR1(4)=6H TIMES
HDR1(5)=8H DAY 248
HDR1(6)=1H
HDR1(7)=1H
HDR1(8)=1H

WRITE (8) ISNI, NCHAN, (NAMESI(I), I=1, NCHAN),
          (IUNITSI(I), I=1, NCHAN), (HDR1(I), I=1, 8),
          (HDR1(I), I=1, 8)
PRINT*, ISNI, NCHAN, (NAMESI(I), I=1,NCHAN),
     (IUNITSI(I), I=1, NCHAN)
FORMAT (4X, 5HISNI=, I5, 2X, 7HNCHAN=, I3, 2X, 2A7, 5X, 8A10)

NAMESI(2)=8HRCS CODE
WRITE (10) ISNI, NCHAN, (NAMESI(I), I=1, NCHAN),
           (IUNITSI(I), I=1, NCHAN), (HDR1(I), I=1, 8)

READ TAPE HEADER

101 READ (9, END=600) ISN, NCHAN, (NAMES(I), I=1, NCHAN),
       (IUNITS(I), I=1, NCHAN), (HDR(I), I=1, 8)

600 IF (EOF(9) .NE. 0) THEN
    PRINT 802
    802 FORMAT (1X,"EOF FOUND WHEN ATTEMPTING TO READ HEADER")
    GO TO 900
END IF

40 READ(9,END=450) TIME, DAY, (PRESS(J), J=3, 46)
C** PRINT*, TIME, DAY, (PRESS(I), I=3, 10)
C*400 FORMAT(12(3X,F10.3))
450 IF(EOF(9) .NE. 0 .AND. TIME .GT. STOP) THEN
PRINT 465, TIME
465 FORMAT (1X,'EOF FOUND WHEN READING DATA ON TAPE9
. AFTER TIME= ',F16.9)
IF (TIME .LT. STOP) GO TO 101
PRINT 467, TIME
467 FORMAT (1X,4X,36HSTOP TIME LESS THAN LAST TIME FOUND =,F16.9)
GO TO 900
END IF

IF (START .GT. TIME) GO TO 40

DO 57 I=3,46
IF (PRESS(I) .GT. 1E+18 .OR. PRESS(I) .LT. ZERO(I)) GO TO 57
IF (I .LE. 18) THEN
CODE=I
ELSE IF (I .LE. 32) THEN
CODE=2
ELSE
CODE=3
END IF
WRITE (10) TIME, CODE
57 CONTINUE

IF (STOP .LE. TIME) GO TO 300
GO TO 40
300 PRINT 245,TIME
245 FORMAT (1X,4X,46HSTOP TIME DESIRED IS LESS THAN LAST TIME READ=, 
. F16.9)
LTIME=TIME
REWIND(10)
PTIME=0
LFSTOP=0
60 READ(10,END=61) TIME, CODE
61 IF(EOF(10) .NE. 0) THEN
   PRINT 62, TIME
   62 FORMAT(1X,'EOF FOUND WHEN READING DATA ON TAPE10
                            AFTER TIME= ',F16.9)
   PRINT 63, TIME
   63 FORMAT(1X,4X,38HSTOP TIME LESS THAN LAST TIME FOUND =,F16.9)
   GO TO 900
   END IF

30 IF (TIME - PTIME .GT. 0.041) THEN
   IF (LFSTOP .EQ. 0) THEN
      STIME=TIME - 0.04
      LFSTOP=1
   ELSE
      IF (TIME .LT. TCLAG) THEN
         STPTIME=PTIME + LAG
      ELSE
         STPTIME=PTIME + .08
      END IF
      WRITE (8) STIME, STPTIME
      PRINT 88, STIME, STPTIME
      88 FORMAT (5X, F12.5, 5X, F12.5)
      LFSTOP=0
      GO TO 30
   ELSE
      END IF
   END IF
   END IF

PTIME=TIME

IF (LTIME .GT. TIME) GO TO 60

900 CONTINUE
   PRINT 801
   801 FORMAT (1X,'THRUST COMPLETED')
   STOP
   END
PROGRAM THFIT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
        .          TAPE9,TAPE10,TAPE11,TAPE12,TAPE13)

THIS ROUTINE REMOVES ODD DATA SPIKES THAT ARE LEFT AFTER OTHER
THRUST REMOVING EFFORTS.

WRITTEN BY: JOSEPH S. ROWLEY, JAN 1985

DIMENSION X(10000),Y(10000),
        .          NAMES(10),HDR(8),IUNITS(10),DATAPT(10),BUF(65)

REAL LOWER

THE FOLLOWING IS A LIST OF INPUTS TO THIS PROGRAM

CHANIV     CHANNEL TO BE FIT AS INDEPENDENT VARIABLE
CHANDV     CHANNEL TO BE FIT AS DEPENDENT VARIABLE
START      START TIME
STOP       STOP TIME
SIZE       NUMBER OF POINTS FOR LEAST SQUARES FIT
GAPSTR     TIME FOR START OF APU TRANSITION - Z AXIS ONLY
GAPSTP     TIME FOR STOP OF APU TRANSITION - Z AXIS ONLY
SATTIME    SATURATION TIME FOR DEPENDENT CHANNEL
SERSKIP    NUMBER OF SERIALS TO SKIP TO LOCATE DATA
BURNSTR    START TIME OF DEORBIT BURN
BURNSTP    STOP TIME OF DEORBIT BURN

WRITE(6,777)
777 FORMAT(1H1,13HTHFIT STARTED //)

INPUTS GO HERE
CHANIV=1
CHANIV=4
START=0
STOP=62345.
GAPSTR=61220.
GAPSTP=61245.
SATTIME=100000.
CURVBK=61700.
BURNSTR=60030.
BURNSTP=60220.
SIZE1=1740.
SIZE2=174.
SERSKIP=0
JCNT = 0
JMAX = 350

PRINT*, 'CHANIV= ',CHANIV
PRINT*, 'CHANIV= ',CHANIV
PRINT*, 'START = ',START
PRINT*, 'STOP = ',STOP
PRINT*, 'GAPSTR = ',GAPSTR
PRINT*, 'GAPSTP = ',GAPSTP
PRINT*, 'SATTIME = ',SATTIME
PRINT*, 'CURVBK = ',CURVBK
PRINT*, 'BURNSTR = ',BURNSTR
PRINT*, 'BURNSTP = ',BURNSTP
PRINT*, 'SIZE1 = ',SIZE1
PRINT*, 'SIZE2 = ',SIZE2
PRINT*, 'SERSKIP = ',SERSKIP

SKIP TO DESIRED SERIAL IF NEEDED

IF(SERSKIP .NE. 0) THEN
GENERATE FILE ENVIRONMENT TABLE FOR SPECIFIED DIRECT ACCESS FILE

NAME=L"TAPE9"
CALL GENFET(FET,NAME,BUF,65)

CALL SKIPR (FET,SERSKIP)
END IF
READ, WRITE, AND PRINT TAPE HEADER

READ (9, END=600) ISN, NCHAN, (NAMES (I), I=1, NCHAN), (IUNITS (I), I=1, NCHAN), (HDR (I), I=1, 8)

600 IF (EOF(9) .NE. 0) THEN
   PRINT 802
802 FORMAT (1X, "EOF FOUND WHEN ATTEMPTING TO READ HEADER")
   GO TO 400
END IF

WRITE (12) ISN, NCHAN, (NAMES (I), I=1, NCHAN), (IUNITS (I), I=1, NCHAN), (HDR (I), I=1, 8)

PRINT 104, ISN, HDR
104 FORMAT (/X, 4HFILE, I7, 14H FOUND HEADER=, 8A10)
PRINT 303, (NAMES (I), I=1, NCHAN)
PRINT 303, (IUNITS (I), I=1, NCHAN)
PRINT 302
303 FORMAT (I8 (A10))
302 FORMAT (/)

DETERMINE UPPER, LOWER, UPPER2, LOWER2 FOR DESIRED CHANNEL

IF (CHANDV .EQ. 3) THEN
   UPPER=45
   LOWER=45
   UPPER2=45
   LOWER2=45
ELSE IF (CHANDV .EQ. 4) THEN
   UPPER=75
   LOWER=75
   UPPER2=225
   LOWER2=225
ELSE
   PRINT*, 'PROBLEM WITH CHANDV DEFINITION'
END IF

CURVE FITTING STARTS HERE
DATAPT(1)=0
FAKE=0
SIZE=SIZE1

P=0
SUMX=0
SUMY=0
SUMXSQ=0
SUMXY=0
REWIND (11)

40 READ(9,END=500) (DATAPT(J), J=1,NCHAN)
500 IF(EOF(9) .NE. 0) THEN
   PRINT 165,DATAPT(1)
   165 FORMAT (IX,'EOF FOUND WHEN READING DATA ON TAPE9' 
   . 'AFTER TIME= ',F16.9)
   GO TO 400
END IF

IF (DATAPT(1) .LT. START) GO TO 40
IF (DATAPT(1) .GE. STOP) THEN
   PRINT*, 'STOP TIME FOUND AFTER READ AT LABEL 40=',DATAPT(1)
   GO TO I00
END IF

IF (DATAPT(1) .GT. CURVBK .AND. FAKE .NE. I) THEN
   SIZE=SIZE2
   FAKE=1
   GO TO 34
END IF

IF (DATAPT(1) .GE. BURNSTR .AND. DATAPT(1) .LE. BURNSTP) GO TO 40

IF (DATAPT(1) .GE. GAPSTR .AND. DATAPT(1) .LE. GAPSTP 
   . .AND. CHANDV .EQ. 4) GO TO 40

IF (P .EQ. 0) THEN
   FIRSTX=DATAPT(CHANIV)
END IF
P=P+1
IF (P .GT. SIZE) PRINT*, 'PROBLEM WITH SIZE OF P'
WRITE(11) (DATAPT(J), J=1,NCHAN)
WRITE(13) (DATAPT(J), J=1,NCHAN)

X(P)=DATAPT(CHANIV)-FIRSTX
Y(P)=DATAPT(CHANDV)
SUMX=SUMX+X(P)
SUMY=SUMY+Y(P)
SUMXSQ=SUMXSQ+X(P)*X(P)
SUMXY=SUMXY+X(P)*Y(P)

PRINT*, 'CHECK 3 ', SIZE, P
IF (P .NE. SIZE) GO TO 40
IF (DATAPT(1) .LE. SATTIME) TIMLAS=DATAPT(1)
PRINT*, 'CHECK 1'

AVGX=SUMX/P
AVGY=SUMY/P
SLOPE=(SUMXY-P*AVGX*AVGY)/(SUMXSQ-P*AVGX*AVGX)
YINT = AVGY - SLOPE * AVGX

PRINT*, 'SUMX=', SUMX
PRINT*, 'SUMY=', SUMY
PRINT*, 'SLOPE=', SLOPE
PRINT*, 'YINT=', YINT
PRINT*, 'AVGX=', AVGX
PRINT*, 'AVGY=', AVGY
PRINT*, 'SUMXY=', SUMXY
PRINT*, 'SUMXSQ=', SUMXSQ
PRINT*, 'P=', P

CALCULATE DATA FROM ABOVE EQUATION

REWIND (11)

PASS =0
PASS1=0

30 READ(11,END=31) TIME
31 IF(EOF(11) .NE. 0) THEN
   PRINT 32,TIME
32 FORMAT (IX,'EOF FOUND WHEN READING DATA ON TAPE11 ATER TIME= ',F16.9)
   GO TO 34
END IF
C
C IF (PASS1 .EQ. 0) THEN
    FTIME = TIME
    PASS1 = 1
END IF

C
PASS = PASS + 1

C
YVALUE = SLOPE * ( TIME - FTIME ) + YINT
DELT = TIME - FTIME

C
PRINT*, ' CHECK 2'
WRITE (10) TIME, YVALUE
JCNT = JCNT + 1
IF (JCNT .GE. JMAX) THEN
    WRITE (6, 22) YVALUE, SLOPE, TIME, DELT, YINT
    JCNT = 0
ENDIF

22 FORMAT (1F12.3, 1E12.5, 3F12.3)
GO TO 30

C
100 CONTINUE
C
PRINT*, ' '
PRINT*, ' '
PRINT*, 'DATA SPIKE REMOVING STARTED'

C
TIME = 0

C
REWIND (13)
REWIND (10)

C
TIME = 0

C
55 READ (13, END=550) (DATAPT(J), J=1, NCHAN)
550 IF (EOF (13) .NE. 0) THEN
    PRINT 265, DATAPT(1)
265 FORMAT (1X, 'EOF FOUND WHEN READING DATA ON TAPE13 AFTER TIME= ', F16.9)
    GO TO 400
END IF
IF (DATAPT(1) .LT. START) GO TO 55
IF (DATAPT(1) .GE. STOP) GO TO 400
IF (DATAPT(1) .LT. TIME) GO TO 55
IF (DATAPT(1) .GT. TIME) GO TO 56
GO TO 57

56 READ(10,END=556) TIME,YVALUE
556 IF(EOF(10) .NE. 0) THEN
PRINT 557,TIME
557 FORMAT (IX,'EOF FOUND WHEN READING DATA ON TAPE10 IN 2ND LOCATION
AFTER TIME= ',F16.9)
GO TO 400
END IF

IF (DATAPT(1) .LT. TIME) GO TO 55
IF (DATAPT(1) .GT. TIME) GO TO 56

57 IF ((DATAPT(1) .GE. TIMLAS) .AND. (DATAPT(1) .LE. SATTIME)) GO TO 55

IF (TIME .NE. DATAPT(1)) THEN
PRINT*, 'PROBLEM WITH TIME MATCH AT TIME=', TIME, DATAPT(1)
GO TO 400
END IF

IF (CHANDV .EQ. 4 .AND. DATAPT(1) .GE. GAPSTP) THEN
UPPER=UPPER2
LOWER=LOWER2
END IF

AUPPER= YVALUE + UPPER
ALOWER= YVALUE - LOWER
ELIMINATE SPIKE IF NECESSARY

IF(DATAPT(2).LT.87000.AND.DATAPT(3).GT.-4000.) GO TO 55
IF(DATAPT(2).LT.86000.AND.DATAPT(3).GT.-4700.) GO TO 55

IF (DATAPT(CHANDV) .GT. AUPPER .OR. DATAPT(CHANDV) .LT. ALOWER) GO TO 55

WRITE(12) (DATAPT(I),I=1,NCHAN)
GO TO 55

400 PRINT*, 'THFIT COMPLETED'

STOP
END
PROGRAM RCOMBIN(INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE7, TAPE8, TAPE9, TAPE10, TAPE1)

THIS ROUTINE WAS WRITTEN TO COMPLEMENT JSRSMTH (THE HIRAP DATA SMOOTHING ROUTINE) WHICH IS LIMITED TO SMOOTHING ONLY ONE CHANNEL AT A TIME. THE PURPOSE OF THIS ROUTINE IS TO COMBINE THREE FILES (X, Y, Z-AXIS) INTO ONE FOR ALL THREE.

WRITTEN BY: JOSEPH S. ROWLEY, SEPT 1983
MODIFIED BY: ROBIN WADDELL, JAN 1984

DIMENSION XDATA(10), YDATA(10), ZDATA(10)

INTEGER ISN7, ISN8, ISN9, NCHAN7, NCHAN8, NCHAN9,
       NAMES8(10), NAMES9(10), IUNITS7(10), IUNITS8(10),
       IUNITS9(10), HDR7(8), HDR8(8), HDR9(8), NAMES7(10), COUNT

THE FOLLOWING IS A LIST OF INPUTS TO THIS PROGRAM

COUNT NUMBER OF CHANNELS TO BE COMBINED
START START TIME FOR COMBINING FILES
STOP STOP TIME FOR COMBINING FILES

PRINT 800
800 FORMAT(/,/,1X,'COMBINE STARTED')

INPUTS GO HERE

COUNT=2
START=0
STOP=100000.
READ TAPE HEADERS

IF (COUNT .GE. 1) THEN
  101 READ (7, END=102) ISN7, NCHAN7, (NAMES7 (I), I=1, NCHAN7),
      (IUNITS7 (I), I=1, NCHAN7), (HDR7 (I), I=1, 8)

  102 IF (EOF(7) .NE. 0) THEN
      PRINT 103
      103 FORMAT (1X, "EOF FOUND WHEN READING TAPE7 HEADER")
      GO TO 900
      END IF

      PRINT 104, ISN7, HDR7
  104 FORMAT (/*, 1X, 'FILE ', I7, ' FOUND HEADER=', 8A10)
      PRINT 105, (NAMES7 (I), I=1, NCHAN7)
      PRINT 105, (IUNITS7 (I), I=1, NCHAN7)
      PRINT 106
      106 FORMAT (/*, 8 (AI0))
      END IF

IF (COUNT .GE. 2) THEN
  801 READ (8, END=802) ISN8, NCHAN8, (NAMES8 (I), I=1, NCHAN8),
      (IUNITS8 (I), I=1, NCHAN8), (HDR8 (I), I=1, 8)

  802 IF (EOF(8) .NE. 0) THEN
      PRINT 803
      803 FORMAT (8X, "EOF FOUND WHEN READING TAPE8 HEADER")
      GO TO 900
      END IF

      PRINT 804, ISN8, HDR8
  804 FORMAT (/*, 1X, 'FILE ', I7, ' FOUND HEADER=', 8A10)
      PRINT 805, (NAMES8 (I), I=1, NCHAN8)
      PRINT 805, (IUNITS8 (I), I=1, NCHAN8)
      PRINT 806
      806 FORMAT (/*, 8 (AI0))
      END IF

IF (COUNT .GE. 3) THEN
  901 READ (9, END=902) ISN9, NCHAN9, (NAMES9 (I), I=1, NCHAN9),
      (IUNITS9 (I), I=1, NCHAN9), (HDR9 (I), I=1, 8)

  902 IF (EOF(9) .NE. 0) THEN
      PRINT 903
      903 FORMAT (1X, "EOF FOUND WHEN READING TAPE9 HEADER")
      GO TO 900
      END IF
END IF

C  PRINT 904,ISN9,HDR9
C 904 FORMAT(/,1X,'FILE ',I7,' FOUND HEADER=',8A10)
C  PRINT 905, (NAMES9(I),I=1,NCHAN9)
C  PRINT 905, (IUNITS9(I),I=1,NCHAN9)
C  PRINT 906
C 906 FORMAT(1X,8(A10))
C  END IF

C  NCHAN7=COUNT+1
C  WRITE (10) ISN7,NCHAN7,(NAMES7(I),I=1,NCHAN7),
C  (IUNITS7(I),I=1,NCHAN7), (HDR7(I),I=1,8)
C
C  NCHAN7=COUNT+1
C  MOVE TO DESIRED START AND COMBINE FILES UNITL STOP
C  OR END OF FILE
C
C  XDATA(1)=0
C  YDATA(1)=0
C  ZDATA(1)=0

40 IF(COUNT .GE. 1) THEN
C  READ(7,END=41) (XDATA(J),J=1,NCHAN7)
C 41 IF(EOF(7) .NE. 0) THEN
C  PRINT 42,XDATA(1)
C 42 FORMAT (1X,'EOF FOUND WHEN READING DATA ON TAPE7
C  . AFTER TIME= ',F16.9)
C  GO TO 900
C  END IF
C
C  END IF
C  IF(XDATA(1).LT.ZDATA(1))GO TO 40
C  IF(XDATA(1).GT.ZDATA(1))GO TO 50
C  GO TO 999
C 50 IF(COUNT .GE. 2) THEN
C  READ(8,END=51) (ZDATA(J),J=1,NCHAN8)
C 51 IF(EOF(8) .NE. 0) THEN
C  PRINT 52,ZDATA(1)
C 52 FORMAT (1X,'EOF FOUND WHEN READING DATA ON TAPE8
C  . AFTER TIME= ',F16.9)
C  GO TO 900
C  END IF
C
C  IF(XDATA(1).LT.ZDATA(1))GO TO 40
C  IF(XDATA(1).GT.ZDATA(1))GO TO 50
C  END IF
C
C  IF(XDATA(1).LT.YDATA(1))GO TO 40
C  IF(XDATA(1).GT.YDATA(1))GO TO 60
C  GO TO 999

91
60 IF(COUNT.GE.3)THEN
  READ(9,END=61) (YDATA(J),J=1,NCHAN9)
61 IF(EOF(9) .NE. 0) THEN
  PRINT 62,YDATA(1)
62 FORMAT (1X,'EOF FOUND WHEN READING DATA ON TAPE9
     .  
     .  AFTER TIME= ',F16.9)
  GO TO 900
END IF

C
  IF(XDATA(1).LT.YDATA(1))GO TO 40
  IF(XDATA(1).GT.YDATA(1))GO TO 60
C
END IF
999 CONTINUE
  IF (XDATA(1) - START) 40,240,240
240 IF (STOP - XDATA(1)) 300,300,241
241 CONTINUE
  IF (COUNT .EQ. 2) THEN
    WRITE(10) XDATA(1),XDATA(2),ZDATA(3)
    GO TO 40
  END IF
C
  IF(COUNT.EQ.3) THEN
    WRITE(10) XDATA(1),XDATA(2),ZDATA(3),YDATA(4)
    GO TO 40
END IF
C
PRINT 777,XDATA(1),ZDATA(1),YDATA(1)
777 FORMAT(1X,'TIME MISMATCH PRESENT X=',F16.9,'Z=',F16.9,
     .  'Y=',F16.9)
  GO TO 900
C
C
C
300 PRINT 245,XDATA(1)
245 FORMAT (1X,4X,46HSTOP TIME DESIRED IS LESS THAN LAST TIME READ=,
     .  .F13.4)
C
900 CONTINUE
PRINT 888
888 FORMAT('  COMBINE COMPLETED')
STOP
END

PROGRAM YCONV06 (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE8,
INTEGER OUTTAPE

DIMENSION TIM(15000), TEMPER(15000, 6), TEMP(6)

INPUT BLOCK: CHANGE FOR EACH FLIGHT

T1 = 65500.
T2 = 68100.
I = 0

END INPUT BLOCK

INTAPE = 8
OUTTAPE = 10
NFILE = 9
READ (NFILE)
10 I = I+1
   READ (NFILE, END=20) TIM(I), (TEMPER(I,J), J=I, 6)
   GO TO 10
20 CONTINUE
   NT = I-1
   FINTIM = TIM(NT)

REWIND (INTAPE)
REWIND (OUTTAPE)
30 READ (INTAPE, END=70) TYME, XCOUNT, ZCOUNT, YCOUNT

IF (TYME.LT.T1) GO TO 30
IF (TYME.GT.T2) GO TO 70
IF (TYME.GT.FINTIM) THEN
   DO 40 NN = I, 6
   TEMP(NN) = 9999.
   CONTINUE
ELSE
   CALL IUNI (15000, NT, TIM, 6, TEMPER, 1, TYME, TEMP, -1, IE)
ENDIF

CONVERT FROM COUNTS TO UNITS OF MICROGS

IF (IE.NE.0) THEN
   WRITE (6, 50) IE, TYME
   FORMAT (2X, 'INTERPOLATION ERROR ', I3, 'AT TIME = ', 1X, F12.4)
50
ENDIF

SN 001 SCALE FACTOR VALUES: PRE-MOD

SFX = -1.247237E-3
SFY = 1.253821E-3
SFZ = -1.246810E-3

VX = -10.+((XCOUNT/16383.)*20.
VY = -10.+((YCOUNT/16383.)*20.
VZ = -10.+((ZCOUNT/16383.)*20.

CONVERSION TO MICROG'S WITH SCALE FACTOR

XMG = VX/SFX
YMG = VY/SFY
ZMG = VZ/SFZ

WRITE NON-DETRENDED MICROG TO OUTPUT TAPE10

WRITE (OUTTAPE) Tyme, XMG, YMG, ZMG, (TEMP(K), K=1, 6)

IF (Tyme.LT. (TI+1.)) THEN
    WRITE (6, 60) Tyme, XMG, YMG, ZMG, (TEMP(M), M=1, 6)
60 FORMAT (2X, 'AT TIME = ', F11.4, ' AX AY AZ = ', 3(2X, E11.4), /
     " TEMP = ", 6(1X, F11.4))

ENDIF

GO TO 30

70 CONTINUE

80 FORMAT (2X, 'FINAL VALUES Tyme, AX, AY, AZ, TEMP=', 1X, F12.4,
     * 3(1X, F11.4)
     * , /, 6(1X, F11.4))

WRITE (6, 80) Tyme, XMG, YMG, ZMG, (TEMP(L), L=1, 6)

WRITE (6,90) T1, T2
90 FORMAT (2X, 'REQUESTED TIME PROCESSED T1 - T2 =', 2(2X,F11.4))
STOP
END
PROGRAM FILLSQ( INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, *
  TAPE1,TAPE2)

  HDR(8), *A(3), B(3), C(3), SDEV(3), T(2000), ACC(2000, 3), GAPSTRT(100), *
  GAPSTOP(100), NFIL(100)

EQUIVALENCE (ACC(1,1), X(1)), (ACC(1,2), Z(1))

READ WINDOW SIZE

READ (5, 53) NPTS, IDEV, FREQ
READ (1, END=1) ISN, NCHAN, (NAMES(I), I=1, NCHAN), *
  (IUNITS(I), I=1, NCHAN), (HDR(I), I=1, 8)

DO 5 I = 1, 100
  GAPSTRT(I) = 1.0E+06
  GAPSTOP(I) = 1.0E+06
  NFIL(I) = 0.0
CONTINUE

CONTINUE

IF (EOF(1) .NE. 0) THEN
  WRITE (6, 50)
  IFILE = 1
  GO TO 14
ENDIF

WRITE (2) ISN, NCHAN, (NAMES(I), I=1, NCHAN), (IUNITS(I), *
  I=1, NCHAN), (HDR(I), I=1, 8)
READ (5, 52) FLT, DAY
  IFLT = FLT
WRITE (6, 54) IFLT, NPTS, FREQ, IDEV
WRITE (6, 55) TSTART, TLIMIT
DELTIM = 1.0/FREQ * 5.0
WRITE (6, 56) DELTIM
WRITE (6, 57) NMED = NPTS/2
  IFILE = 0
  IGAP = 0
  IPTS = NPTS
READ (1, END=1) T(1), (ACC(I,J), J=1, 2)
WRITE(2) T(1), (ACC(1,J), J=1,2)
TFIRST = T(1)

CONTINUE
READ(1,END=1) T(1), (ACC(1,J), J=1,2)
WRITE(2) T(1), (ACC(1,J), J=1,2)
IF (T(1) .LT. TSTART) GO TO 4

IF (TSTART .EQ. 0.0) THEN
  TSTART = T(I)
  WRITE(6,60) TSTART
ENDIF

I = 1
CONTINUE
I = I + 1
READ(1,END=1) T(I), (ACC(I,J), J=1,2)
IF (I .EQ. I) GO TO 6
IF ((T(I) - T(I-1)) .LT. DELTIM) THEN
  IF (I .LT. IPTS) GO TO 6
  GO TO 7
ELSE
  IGAP = IGAP + 1
  GAPSTR(T(IP)) = T(I-1)
  GAPSTOP(IP) = T(I)
  GAP = T(I) - T(I-1)
  NFIL(IP) = GAP * FREQ - 0.5
  IF (IGAP .EQ. 100) GO TO 7
  GO TO 6
ENDIF

CONTINUE
IPTS = I
WRITE(6,62) IPTS, IGAP, T(1), IPTS, T(IPTS)
DO 20 L = I, IGAP
  WRITE(6,100) L, GAPSTR(L), L, GAPSTOP(L), L, NFIL(L)
FORMAT(10X,8HGAPSTR(L),112,4H) = ,1F10.3,5X,8HGAPSTOP(L),112,4H) = ,
  *1F10.3,5X,5HNFIL(L),112,4H) = ,1I4)
20 CONTINUE
IF (IGAP .GT. 0) THEN
  CALL POINTS(IPTS, T, ACC, A, B, C, SDEV, TSTART)
ENDIF

ISTOP = NMED
IF (IPTS .LT. NPTS) ISTOP = IPTS/2
IF (IFILE .EQ. 1) ISTOP = IPTS
JCNT = 1
CONTINUE

DO 9 I = 1, ISTOP

96
IF (T(I) .EQ. GAPSTOP(JCNT)) THEN
  T1 = GAPSTRT(JCNT)
  T2 = GAPSTOP(JCNT)
  NFILL = NFIL(JCNT)
  CALL GAPFILL(NFILL, T1, T2, A, B, C, SDEV, FREQ, IDEV, TSTART)
  JCNT = JCNT + 1
ENDIF
C
WRITE(2) T(I), (ACC(I,J), J=1,2)
CONTINUE
C
IF (IFILE .EQ. 1) GO TO 15
ICNT = 0
JCNT = JCNT - 1
C
IF (IGAP .GT. JCNT) THEN
  NCNT = IGAP - JCNT
  DO 10 I = 1, NCNT
      GAPSTRT(I) = GAPSTRT(I + JCNT)
      GAPSTOP(I) = GAPSTOP(I + JCNT)
      NFIL(I) = NFIL(I + JCNT)
  ICNT = ICNT + 1
  CONTINUE
  IGAP = ICNT
  IF (JCNT .GT. 0) THEN
    DO 12 II = ICNT + I, JCNT
        GAPSTRT(I) = 1.0E+06
        GAPSTOP(I) = 1.0E+06
        NFIL(I) = 0.0
    CONTINUE
  ENDIF
  ICNT = 0
DO 13 I = ISTOP + I, IPTS
  ICNT = ICNT + 1
  T(ICNT) = T(I)
  DO 12 J = 1, 2
      ACC(ICNT, J) = ACC(I,J)
  CONTINUE
  I = ICNT
WRITE(6, 101) T(I), ICNT, T(ICNT), IGAP, JCNT
101 FORMAT(10X,7HT(1) = ,1F10.3,5X,2HT(,1I4,4H) = ,1F10.3,5X,
*7HIGAP = ,1I2,5X,7HJCNT = ,1I2 )
IPTS = NPTS
GO TO 6
14 CONTINUE
I = I - 1
GO TO 7
15 CONTINUE
TLAST = T(IPTS)
WRITE(6, 61) TFIRST, TLAST
STOP
C
50 FORMAT(// 5X,40HEOF FOUND WHEN ATTEMPTING TO READ HEADER)
51 FORMAT(5X,7(A10,2X))
52 FORMAT(4F8.0)
53 FORMAT(2I4,1F8.0)
54 FORMAT(1H1,4X,10HFLT = STS ,1I2,5X,1I5,18H POINT WINDOW SIZE,
*5X,12HFREQUENCY = ,1F5.1,3H HZ,5X,7HIDEV = ,1I1 )
55 FORMAT(5X,13HTIME START = ,1F10.3,8H SECONDS,5X,
*13HTIME LIMIT = ,
*1F10.3,8H SECONDS )
56 FORMAT(5X,6HISN = ,1I3,5X,8HNCHAN = ,1I3,5X,
* 15HFOUNDED HEADER : ,8A10 //)
57 FORMAT(5X,1A10 //</)
58 FORMAT(5X,19HMAXIMUM TIME GAP = ,1F8.6,8H SECONDS )
59 FORMAT(1H1 / 26X,23HTIME GAP FILL IN POINTS // 8X,4HTIME,11X,
*6HX-AXIS,11X,6HZ-AXIS,11X,6HY-AXIS /)
60 FORMAT(5X,9HTFIRST = ,1F10.3)
61 FORMAT(10X,9HTLAST = ,1F10.3)
62 FORMAT(// 5X,9HTFIRST = ,1F10.3,5X,9HTLAST = ,1F10.3)
63 FORMAT(/10X,7HIPTS = ,1I5,5X,7HIGAP = ,1I4,5X,7HT(1) = ,1F10.3,
*5X,2HT(1,1I4,4H) = ,1F10.3)
END
SUBROUTINE BANDSQR (X, Y, NX, SDEV, XSQSUM, XSUM, XYSUM, YSUM, NTOT,
*XCUBSUM,XFORSUM, XSQYSUM, A, B, C)
C
* XYSUM(3),YSUM(3),A(3),B(3),C(3),NX(3),XCUBSUM(3),XFORSUM(3),
* XSQYSUM(3)
C
DO 2 J = 1,2
   IEND = NX(J)
   L = 0
C
   DO 1 I = 1,IEND
      YEST = A(J) + B(J) * X(I,J) + C(J) * X(I,J)**2
      YTOP = YEST + SDEV(J)
      YBOT = YEST - SDEV(J)
C
      IF (Y(I,J) .GE. YBOT .AND. Y(I,J) .LE. YTOP) THEN
         L = L + 1
         X(L,J) = X(I,J)
         Y(L,J) = Y(I,J)
         XCUBSUM(J) = XCUBSUM(J) + X(I,J)**3
         XFORSUM(J) = XFORSUM(J) + X(I,J)**4
         XSUM(J) = XSUM(J) + X(I,J)
C
ENDIF
XSQSUM(J) = XSQSUM(J) + X(I,J)**2
XSQYSUM(J) = XSQYSUM(J) + X(I,J)**2
XYSUM(J) = XYSUM(J) + X(I,J) * Y(I,J)
YSUM(J) = YSUM(J) + Y(I,J)

CONTINUE

NTOT(J) = L
CONTINUE
WRITE(6,10)
CONTINUE
DO 3 I = 1,2
WRITE(6,11)I,NX(I),I,XSUM(I),I,XSQSUM(I),I,XYSUM(I),I,
YSUM(I),I,XCUBSUM(I),I,XFORSUM(I),I,XSQYSUM(I),
I,NTOT(I)
CONTINUE

FORMAT(15X,14HBANDSQR OUTPUT )
FORMAT(10X,3HNX(I),111,4H) = ,1I4,5X,5HXSUM(),111,4H) = ,1E12.5,5X,
*7HXSUM(),111,4H) = ,1E12.5, / 10X,6HXYSUM(),111,4H) = ,1E12.5,5X,
*5YSUM(),111,4H) = ,1E12.5,5X,8HXCUBSUM(),111,4H) = ,1E12.5 / 10X,
*8HXFORSUM(),111,4H) = ,1E12.5,5X,8HXYSUM(),111,4H) = ,1E12.5,5X,
*5HNTOT(),111,4H) = ,1I6 / )
RETURN
END
SUBROUTINE GAPFILL(NFILL,TIM1,TIM2,A,B,C,SDEV,FREQ,IDEV,TSTART)
DIMENSION Y(3),A(3),B(3),C(3),SDEV(3),DIST(300),DNORM(300)
DATA (DNORM(I),I=1,100) / .50000,.50399,.50798,.51197,.51595,
* .51994,.52392,.52790,.53188,.53586,.53983,.54380,
* .54776,
* .55172,.55567,.55962,.56356,.56749,.57142,.57535,
* .57926,
* .58317,.58706,.59095,.59483,.59871,.60257,.60642,
* .61026,
* .61409,.61791,.62172,.62552,.62930,.63307,.63683,
* .64058,
* .64431,.64803,.65173,.65542,.65910,.66276,.66640,.67003,
* .67364,.67724,.68082,.68439,.68793,.69146,.69497,.69847,
* .70194,.70540,.70884,.71226,.71566,.71904,.72240,.72575,
* .72907,.73237,.73565,.73891,.74215,.74537,.74857,.75175,
* .75490,.75804,.76115,.76424,.76730,.77035,.77337,.77637,
* .77935,.78230,.78524,.78814,.79103,.79398,.79673,.79955,
* .80234,.80511,.80785,.81057,.81327,.81594,.81859,.82121,
* .82381,.82639,.82894,.83147,.83398,.83646,.83891 /
DATA (DNORM(I),I=101,200) / .84134,.84375,.84614,.84849,.85083,
* .85314,.85543,.85769,.85993,.86214,.86433,.86650,.86864,
* .87076,.87286,.87493,.87698,.87900,.88100,.88298,.88493,
* .88686,.88877,.89065,.89251,.89435,.89617,.89796,.89973,
* .90147,.90320,.90490,.90658,.90824,.90988,.91149,.91309,
* .91466,.91621,.91774,.91924,.92073,.92220,.92364,.92507,
DO 2 I = I,NFILL
KMAX= 20
KCNT = 19
WRITE(6,11)
IF (IDEV .EQ. I) CALL RANSET (N)

C
WRITE(6,11)
DELTIM = 1.0/FREQ
KCNT = 19
KMAX = 20
IF (IDEV .EQ. 1) CALL RANSET(N)
C
DO 2 I = 1,NFILL
TIME = TIM1 + DELTIM * I
IF (TIME .GE. TIM2) GO TO 3
TIMEX = TIME - TSTART

DO 1 J = 1, 2
  IF (IDEV .EQ. 1) Z = RANF()

  IF (IDEV .EQ. 1 .AND. J .NE. 1) THEN
    IF (Z .GE. 0.5) THEN
      Z1 = Z
      CALL IUNI(300, 300, DNORM, 1, DIST, 1, Z1, ANS, -1, IE)
      IF (ANS .GT. 3.0) ANS = 3.0
    ELSE
      Z1 = 1.0 - Z
      CALL IUNI(300, 300, DNORM, 1, DIST, 1, Z1, ANS, -1, IE)
      IF (ANS .GT. 3.0) ANS = 3.0
      ANS = -ANS
    ENDIF
  ELSE
    DELACC = ANS * SDEV(J)
  ELSE
    DELACC = 0.0
  ENDIF

  Y(J) = A(J) + B(J)*TIMEX + C(J)*TIMEX**2 + DELACC
  CONTINUE

WRITE(2) TIME, (Y(J), J=1, 2)
KCNT = KCNT + 1

IF (KCNT .EQ. KMAX) THEN
  KCNT = 0
  WRITE(6, 10) TIME, (Y(J), J=1, 2)
ENDIF

CONTINUE

CONTINUE

IF (KCNT .NE. 0) WRITE(6, 10) TIME, (Y(J), J=1, 2)
RETURN

FORMAT(5X, 1F10.3, 3(5X, 1E12.5))
FORMAT( // 10X, 19H GAPFILL DATA POINTS )
END

SUBROUTINE POINTS(IPTS, T, ACC, A, B, C, SDEV, TSTART)

* C(3), XSUM(3), XSQSUM(3), NTOTAL(3), YSUM(3), XYSUM(3), YBAR(3), NX(3),
* XSQYSUM(3), XCBUSUM(3), XFORSUM(3), TZ(2000, 3), AZ(2000, 3), SUM(3)

C ICNT = 0
ICNTMAX = 3
DO 1 I = 1, 2
   A(I) = 0.0
   B(I) = 0.0
   C(I) = 0.0
   NTOTAL(I) = 0
   NX(I) = IPTS
   SDEV(I) = 1.0E06
   XCUBSUM(I) = 0.0
   XFORSUM(I) = 0.0
   XSUM(I) = 0.0
   XSQSUM(I) = 0.0
   XSQYSUM(I) = 0.0
   XYSUM(I) = 0.0
   YSUM(I) = 0.0

C

DO 7 J = 1, IPTS
   TZ(J, I) = T(J) - TSTART
   AZ(J, I) = ACC(J, I)
   CONTINUE

7 C

CONTINUE

1 C

CONTINUE

2 CONTINUE

CALL BANDSQR(TZ, AZ, NX, SDEV, XSQSUM, XSUM, XYSUM, YSUM, NTOTAL,
   * XCUBSUM, XFORSUM, XSQYSUM, A, B, C)
   ICNT = ICNT + 1

C

DO 4 I = 1, 2
   WRITE (6, 22) I, XSQSUM(I), I, XSUM(I), I, XYSUM(I), I, YSUM(I), I,
   * XCUBSUM(I), I, XFORSUM(I), I, XSQYSUM(I), I, NTOTAL(I)
   N = NTOTAL(I)
   XBAR(I) = XSUM(I)/N
   YBAR(I) = YSUM(I)/N
   VAL1 = XCUBSUM(I) - XBAR(I) * XSQSUM(I)
   VAL2 = XSQSUM(I) - N * XBAR(I)**2
   DENOM = XFORSUM(I) - XSQSUM(I)**2/N - VAL1**2/VAL2
   C(I) = (XSQYSUM(I) - YBAR(I) * XSQSUM(I) + (N * XBAR(I) * YBAR(I) - XYSUM(I) ) * VAL1/VAL2
   * YBAR(I) - XYSUM(I)) / VAL1/VAL2)/DENOM
   B(I) = (XYSUM(I) - N * XBAR(I) * YBAR(I) - C(I) * VAL1)/VAL2
   A(I) = YBAR(I) - B(I) * XBAR(I) - C(I) / N * XSQSUM(I)

COMPUTE VARIANCE

SUM(I) = 0.0

C

DO 3 J = 1, N
   SUM(I) = SUM(I) + (AZ(J, I) - A(I) - B(I) * TZ(J, I) -
   * C(I) * TZ(J, I)**2)**2
   CONTINUE

3 C

VAR(I) = SUM(I)/(N - 3)
SDEV(I) = SQRT(VAR(I))

4 C

CONTINUE
IF (ICNT .EQ. 1) WRITE(6,20)
IF (ICNT .EQ. 2) WRITE(6,21)
IF (ICNT .EQ. 3) WRITE(6,25)
IF (ICNT .EQ. 4) WRITE(6,26)
WRITE(6,23)

IF (ICNT .EQ. ICNTMAX) THEN
  WRITE(6,25)
  WRITE(6,23)
DO 5 I = 1,2
   WRITE(6,24) XBAR(I), YBAR(I), SDEV(I),A(I),B(I),C(I), NTOTAL(I)
   CONTINUE
ENDIF

IF (ICNT .LT. ICNTMAX) THEN
DO 6 I = 1,2
   NX(I) = NTOTAL(I)
   NTOTAL(I) = 0
   XCUBSUM(I) = 0.0
   XFORSUM(I) = 0.0
   XSUM(I) = 0.0
   XSQSUM(I) = 0.0
   XSQYSUM(I) = 0.0
   XYSUM(I) = 0.0
   YSUM(I) = 0.0
   CONTINUE
GOTO2
ENDIF
GO TO 2
ENDIF

RETURN

FORMAT(/ 5X,24HFIRST STANDARD DEVIATION )
FORMAT(/ 5X,25HSECOND STANDARD DEVIATION )
FORMAT(/ 10X,7HXSQYSUM(,11I,4H) = ,1E12.5,5X,5HXSUM(,11I,4H) = ,*1E12.5,5X,6HXYSUM(,11I,4H) = ,1E12.5,5X,5HYSUM(,11I,4H) = ,1E12.5*
/ 10X,8HXSUM(,11I,4H) = ,1E12.5,5X,8HXFORSUM(,11I,4H) = ,
*1E12.5,5X,8HXSUM(,11I,4H) = ,1E12.5,5X,7HNTOTAL(,11I,4H) = ,
*116 / )
FORMAT(3X,4HXBAR,8X,4HYBAR,10X,4HSDEV,10X,1HA,10X,1HB,11X,1HC,11X,
*4HNPTS / )
FORMAT(6E12.5,3X,1I6)
FORMAT(/ 5X,24THIRD STANDARD DEVIATION )
FORMAT(/ 5X,25TH FOURTH STANDARD DEVIATION )
END
PROGRAM GPREM06 (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE8, TAPE9*
, TAPE9)

DIMENSION TEND(5000), TBEG(5000), TEMP(6)
INTEGER INTAPE, OUTTAPE
COMMON TMID, T1, T2, TXMID, TZMID
COMMON AZSLP, AXSLP, AZINT, AXINT
COMMON AZSE, AXSE
COMMON AYSLP, AYINT, AYSE

C
NORMLX = 0
NORMLZ = 0
AVGZ = 0.
AVGX = 0.
NORMLY = 0
AVGY = 0.
NCULL = 0

UPDATE THE INPUTS WITH RESULTS FROM NCULL

TIMSCI = 0.

INPUT BLOCK: CHANGE FOR EACH FLIGHT

T1 = 65500.
T2 = 68000.

WRITE (5, 2005)

END INPUT BLOCK

T1... THE INITIAL TIME OF THE DATA SECTION
T2... THE FINAL TIME OF THE DATA SECTION.

JJ = 1
10 CONTINUE
READ (3, 20, END=30) TEND(JJ), TBEG(JJ)
20 FORMAT (2X, F10.2, 2X, F10.2)

WRITE (6, 602) JJ, TBEG(JJ), TEND(JJ)

JJ = JJ + 1
GO TO 10
30 IGAPS = JJ - 1

WRITE (6, 40) IGAPS
40 FORMAT (2X, 'NUMBER OF GAPS = ', I7)

CALCULATE THE MIDPOINT OF TIME

TMID = (T2 - T1) / 2. + T1

SWITCH INPUT AND OUTPUT DEVICES AFTER RE-WRITING
NGAPS = 0
INTAPE = 8
REWIND (INTAPE)

DO 70 LL = 1, IGAPS
   READ (8, END=110) Tyme, AX, AY, AZ, (TEMP(J), J=1, 6)
   IF (Tyme.LT.TBEG(LL)) THEN
      WRITE (9) Tyme, AX, AY, AZ, (TEMP(I), I=1, 6)
      GO TO 50
   ENDIF
   IF (Tyme.GT.TEND(LL)) GO TO 60
   AX = 9999.
   AY = 9999.
   AZ = 9999.
   NGAPS = NGAPS+1

   WRITE (6,952) LL, TBEG(LL), TEND(LL), Tyme, AX, AY, AZ
   WRITE (9) Tyme, AX, AY, AZ, (TEMP(I), I=1, 6)
   GO TO 50

70 CONTINUE

TRANSFER DATA TO OUTPUT AFTER GAPS ARE EXHAUSTED

READ (8, END=90) Tyme, AX, AY, AZ, (TEMP(J), J=1, 6)
WRITE (9) Tyme, AX, AY, AZ, (TEMP(J), J=1, 6)
GO TO 80

WRITE (6,100) Tyme, AX, AY, AZ, (TEMP(J), J=1, 6)
100 FORMAT (2X,'Tyme, AX, AY, AZ, TEMP(1)-(6) = ',4(1X,F12.4),/,2X,6(1X,*F12.4))
110 CONTINUE

WRITE (6,120) NGAPS
120 FORMAT (2X,'NO. OF DATUM REPLACED WITH 9999= ',1X,I7)

INTAPE = 9
REWIND (INTAPE)

TAPE 9 SHOULD NOW HOLD ALL 9999 IN THRUST PLACES

T1 = 65500.
T2 = 65900.

CALL STATS PROGRAM TO CHECK STATS FOR 400 SECOND PERIOD

CALL FINDSLP TO GET SLOPE FOR USE IN FINDSIG

CALL FINDSLP (INTAPE)

WRITE (6,130) AXSLP, AXINT, AYSLP, AYINT, AZSLP, AZINT
130 FORMAT (2X,'FIRST RESULT, AXSLP, AXINT = ',2X,2(E15.7,2X)/,*' AYSLP, AYINT = ',2X,2(E15.7,2X)/,' AZSLP, AZINT = ',2X,*2(E15.7,2X)/)

C CALCULATE STANDARD ERROR TO TEST VALUES AGAINST
C
C CALL FINDSIG (INTAPE)
C
WRITE (6,140) AXSE,AYSE,AZSE
140 FORMAT (2X,'FIRST STANDARD ERROR
* E15.7)/)

C AX, AY, AZ NO CULLING=: ' ,/,3 (2X,
C END
SUBROUTINE FINDSIG (INTAPE)
INTEGER INTAPE, OUTTAPE, IOSTAT, IOST
DIMENSION TEMP (6)
REAL XYINT
COMMON TMID, T1, T2, TXMID, TZMID
COMMON AZSLP, AXSLP, AZINT, AXINT
COMMON AZSE, AXSE
COMMON AYSLP, AYINT, AYSE
REWIND (INTAPE)

SUMAZSE = 0.
SUMAYSE = 0.
SUMAXSE = 0.
AZSE = 0.
AXSE = 0.
AYSE = 0.
NCOUNT = 0
NRD = 0
RES = 0.
10 READ (INTAPE,END=40) TYME,AX,AY,AZ, (TEMP (N),N=1,6)
IF (TYME.LT.T1) GO TO 10
IF (TYME.GT.T2) GO TO 40
IF (AZ.EQ.9999.) GO TO 10
XX = TYME-TMID
IF (AZ.EQ.9999.) GO TO 30
ACTZ = AZ
ACTY = AY
ACTX = AX

CALZ = AZSLP*(XX)+AZINT
CALY = AYSLP*(XX)+AYINT
CALX = AXSLP*(XX)+AXINT

SUMAXSE = SUMAXSE+(ACTX-CALX)**2
SUMAYSE = SUMAYSE+(ACTY-CALY)**2
SUMAZSE = SUMAZSE+(ACTZ-CALZ)**2
NCOUNT = NCOUNT+1
IF (NCOUNT.EQ.1) THEN
WRITE (6,20) TYME,AX,AY,AZ, (TEMP (K),K=1,6)
20 FORMAT (2X,'FIRST DATA POINT READ = ',4F14.4,/,2X,*' 6 TEMP POINTS = ',6(1X,F10.3))
ENDIF
30 GO TO 10
40 CONTINUE

C WRITE (6, 50) TYME, AX, AY, AZ, (TEMP(J), J=1, 6)
50 FORMAT (2X, 'LAST DATA POINT READ = ', 4F14.4, /, 2X,
* '6 TEMP POINTS = ', 6(1X, 2F12.2))

C

AZSE = SQRT (SUMAZSE/(NCOUNT-2))
AXSE = SQRT (SUMAXSE/(NCOUNT-2))
AYSE = SQRT (SUMAYSE/(NCOUNT-2))
REWRIND (INTAPE)

C

RETURN

END

SUBROUTINE FINDSLP (INTAPE)
REAL XYINT
INTEGER INTAPE, OUTTAPE, IOSTAT, IOST
DIMENSION TEMP(6)
COMMON TMID, T1, T2, TXMID, TZMID
COMMON AZSLP, AXSLP, AZINT, AXINT
COMMON AZSE, AXSE
COMMON AYSLP, AYINT, AYSE
REWRIND (INTAPE)
SUMX=0.
SUMXSQ=0.
SUMZSQ=0.
SUMZ=0.
SUMY=0.
NCOUNT = 0
RES = 0.

C

SUMAZ = 0.
SUMAX = 0.
SUMAZT = 0.
SUMAYT = 0.
SUMAXT = 0.
SUMAZSQ = 0.
SUMAYSQ = 0.
SUMAXSQ = 0.
AZSLP = 0.
AYSLP = 0.
AXSLP = 0.
AYINT = 0.
AZINT = 0.
AXINT = 0.
AZSE = 0.

107
AYSE = 0.
AXSE = 0.
SUMAZSE = 0.
SUMAYSE = 0.
SUMAXSE = 0.

CALCULATE STANDARD ERROR ON AZ AND AX

10 READ (INTAPE,END=50) Tyme,AX,AY,AZ,(TEMP(N),N=1,6)
IF (Tyme.LT.T1) GO TO 10
IF (Tyme.GT.T2) GO TO 50
IF (AZ.EQ.9999.) GO TO 10
IF (AZ.EQ.9999.) GO TO 40
XX = Tyme-TMID

SUMAZ = SUMAZ+AZ
SUMAX = SUMAX+AX
SUMAY = SUMAY+AY
SUMAZT = SUMAZT+(XX*AZ)
SUMAXT = SUMAXT+(XX*AX)
SUMAYT = SUMAYT+(XX*AY)

SUMX = SUMX+XX
SUMXSQ = SUMXSQ+XX*XX
SUMZSQ = SUMZSQ+XX*XX
SUMZ = SUMZ+XX
NCOUNT = NCOUNT+1
IF (NCOUNT.EQ.1) THEN
WRITE (6,20) Tyme,AX,AY,AZ,(TEMP(J),J=1,6)
20 FORMAT (2X,'FIRST DATA POINT READ = ',4FI4.4,/,2X,
* '6 TEMP POINTS = ',6(IX, F10.3))
30 FORMAT (2X,'LAST DATA POINT READ = ',4FI4.4,/,2X,
* '6 TEMP POINTS = ',6(IX, 2FI2.2))
ENDIF

40 GO TO 10

50 WRITE (6,30) Tyme,AX,AY,AZ,(TEMP(K),K=1,6)
AZSLP = (NCOUNT*SUMAZT-SUMX*SUMAZ)/(NCOUNT*SUMXSQ-SUMX*SUMX)
AXSLP = (NCOUNT*SUMAXT-SUMX*SUMAX)/(NCOUNT*SUMXSQ-SUMX*SUMX)
AYSLP = (NCOUNT*SUMAYT-SUMX*SUMAY)/(NCOUNT*SUMXSQ-SUMX*SUMX)

AZINT = SUMAZ/NCOUNT-AZSLP*(SUMX/NCOUNT)
AXINT = SUMAX/NCOUNT-AXSLP*(SUMX/NCOUNT)
AYINT = SUMAY/NCOUNT-AYSLP*(SUMX/NCOUNT)

REWIND (INTAPE)
RETURN
END
PROGRAM FILLCDE (INPUT, OUTPUT, TAPE5=INPUT
*, TAPE6=OUTPUT, TAPE8, TAPE9, TAPE14)
DIMENSION TEND(5000), TBEG(5000), TEMP(6)
DIMENSION ORT(1600)
DIMENSION XT(1600), YT(1600), ZT(1600)

DIMENSION ARAN(10000), XRN(1600)
DIMENSION AAY(1600), AAX(1600), AAZ(1600), AT(1600)

DIMENSION TX(1600), TY(1600), TZ(1600)
DIMENSION AX2T(1600), AYT(1600), AZ2T(1600)
DIMENSION W(1600), WY(1600, 6), WX(1600, 6), WZ(1600, 6)
DIMENSION WAX(6, 6), WAX(6, 6), WAZ(6, 6), BX(6, 1), BY(6, 1), BZ(6, 1)
DIMENSION RX(1600, 1), RY(1600, 1), RZ(1600, 1)

DIMENSION SX(1), SY(1), SZ(1)

REAL FINTIM
INTEGER INTAPE, OUTTAPE
LOGICAL FLG X, FLG Z

NORMLY = 0
NCULL = 0
TIMSCI = 0.
FLGX = .TRUE.
FLGZ = .TRUE.

NPC = 0
NSAT = 0
FINTIM = 68050.

INPUT BLOCK: CHANGE FOR EACH FLIGHT
TI400 = 65500.

END INPUT BLOCK

INTAPE = 9
REWIND(INTAPE)

READ IN CHUNKS OF DATA, 9 SECONDS IN LENGTH

DO 625 L = 1, 1600
625 W(L) = 1.0
NPC = NPC + 1
1250 IF (NPC .EQ. 1) THEN
      TBEG(NPC) = TI400
      TEND(NPC) = TBEG(NPC) + 9.
ELSE
\[ \text{TBEG}(\text{NPC}) = \text{TEND}(\text{NPC}-1) \]
\[ \text{TEND}(\text{NPC}) = \text{TBEG}(\text{NPC}) + 9. \]

\begin{verbatim}
BUILD ORIGINAL DATA SET

DO 626 I = 1, 1600
   RX(I,1) = 0.
   RY(I,1) = 0.
   RZ(I,1) = 0.
   DO 624 J = 1, 6
      WX(I,J) = 0.
      WY(I,J) = 0.
      WZ(I,J) = 0.
   CONTINUE
DO 628 K = 1, 6
   BX(K,1) = 0.
   BY(K,1) = 0.
   BZ(K,1) = 0.
   DO 629 M = 1, 6
      WAX(K,M) = 0.
      WAY(K,M) = 0.
      WAZ(K,M) = 0.
   CONTINUE

ND = 0
NX = 0
NY = 0
NZ = 0
NAD = 0

READ(9, END = 1310) TYME, AX, AY, AZ, (TEMP(I), I=1,6)
IF (TYME.GE.FINTIM) GO TO 1311
IF (TYME.LT.TBEG(NPC)) GO TO 1299
IF (TYME.GT.TEND(NPC)) GOTO 1300

NAD = NAD + 1
AT(NAD) = TYME - TBEG(1)
ORT(NAD) = TYME
TX(NAD) = TEMP(4)
TY(NAD) = TEMP(5)
TZ(NAD) = TEMP(6)

FILTER DATA FOR ERRONEOUS POINTS

AAX(NAD) = AX
AAY(NAD) = AY
AAZ(NAD) = AZ

BUILD INTERIM DATA SET
\end{verbatim}
IF (AX.EQ.9999.) THEN
GO TO 1291
ELSE
NX = NX + 1
XT(NX) = TYME - TBEG(1)
AX2(NX) = AX
END IF

IF (AY.EQ.9999.) THEN
GO TO 1292
ELSE
NY = NY + 1
YT(NY) = TYME - TBEG(1)
AY2(NY) = AY
END IF

IF (AZ.EQ.9999.) THEN
GO TO 1299
ELSE
NZ = NZ + 1
ZT(NZ) = TYME - TBEG(1)
AZ2(NZ) = AZ
END IF

GO TO 1299

CONTINUE

IF (NX.EQ.0) GO TO 1307
IF (NY.EQ.0) GO TO 1307
IF (NZ.EQ.0) GO TO 1307
CALL LSQPOL(1600,NX,XT,1,AX2,W,3,3,RX,SX,WX,WX,IERR)
WRITE(6,681) IERR,NX
FORMAT(2X,'IERR = ',I5,' NUMBER OF DATA = ',I6)
XSE = (SX(1)/NX)**0.5
IF XSE LT 1.0, ASSUME X IS SATURATED AND DO NOT CULL FURTHER
ONLY WRITE OUT ORIGINAL DATA OUT TO THE TAPE
WRITE(6,7009) TBEG(NPC),TEND(NPC)
WRITE(6,7021) BX(1,1),BX(2,1),BX(3,1),XSE,IERR,NX
IF (BX(3,1).LT.-1.0E10) THEN
WRITE(6,17) (AX2T(L),AX2(L),L=1,NDBX)
GO TO 1307
END IF
FORMAT(2X,'ARRAY AX2T AX2=',2(IX,FI2.4)/,1600(2(IX,FI2.4)))
FORMAT(2X,'X 0TH,1ST,2ND = ',3(IX,E12.4),' XSE = ,IERR = ,NX = ',
*E12.4,2(1X,I5))
CONTINUE

CALL LSQPOL(1600, NY, YT, AY2, W, 3, 3, RY, SY, WAY, BY, WY, IERR)

WRITE(6, 681) IERR, NY
YSE = (SY(1)/NY)**0.5
WRITE(6, 7010) BY(1,1), BY(2,1), BY(3,1), YSE, IERR, NY

CALL LSQPOL(1600, NZ, ZT, AZ2, W, 3, 3, RZ, SZ, WAZ, BZ, WZ, IERR)

WRITE(6, 681) IERR, NZ
ZSE = (SZ(1)/NZ)**0.5
WRITE(6, 7020) BZ(1,1), BZ(2,1), BZ(3,1), ZSE, IERR, NZ

SUM SHOULD GIVE SUM OF RESIDUALS FOR EACH POINT

CONTINUE
NCULL = 1

CULL THE ORIGINAL DATA ON THE BASIS OF 1 STANDARD ERROR

CONTINUE
NDBY = 0
NDBZ = 0
NDBX = 0
DO 7030 J = I, NAD

TIM = AT(J)

IF (XSE.LT.1.) GO TO 8002
IF (AAX(J).EQ.9999.) GOTO 8002
CALX = BX(1,1) + BX(2,1)*TIM + BX(3,1)*TIM**2
IF (ABS(AAX(J)-CALX).LE.XSE) THEN
NDBX = NDBX + 1
AX2(NDBX) = AAX(J)
AX2T(NDBX) = TIM
END IF

CONTINUE

IF (YSE.LT.1.) GO TO 8003
IF (AAY(J).EQ.9999.) GO TO 8003
CALY = BY(1,1) + BY(2,1)*TIM + BY(3,1)*TIM**2
IF (ABS(AAY(J)-CALY).LE.YSE) THEN
NDBY = NDBY + 1

112
AY2(NDBY) = AAY(J)
AY2T(NDBY) = TIM
END IF
8003 CONTINUE
C003 IF (ZSE.LT.1.) GO TO 8010
IF (AAZ(J).EQ.9999.) GO TO 8010
CALZ = BZ(1,1) + BZ(2,1)*TIM + BZ(3,1)*TIM**2
IF (ABS(AAZ(J)-CALZ).LE.ZSE) THEN
  NDBZ = NDBZ + 1
  A22(NDBZ) = AAZ(J)
  A22T(NDBZ) = TIM
END IF
8010 CONTINUE
7030 CONTINUE
C
C IF (XSE.LT.1.)GO TO 6240
C
CALL LSQPOL(1600, NDBX, AX2T, 1, AX2, W, 3, 3, RX, SX, WAX, BX, WX, IERR)
C
WRITE(6,681) IERR, NDBX
XSE = (SX(1)/NDBX)**0.5
WRITE(6,7050) BX(1,1), BX(2,1), BX(3,1), XSE, NCULL, IERR, NDBX
7050 FORMAT(2X,'X 0TH, IST, 2ND = ',3(IX, E12.4),' XSE = ',E12.4,
* NCULL, IE,NDBX= ',3(IX, I5))
C
6240 CONTINUE
C240 IF (YSE.LT.1.0)GOTO 6250
C
CALL LSQPOL(1600, NDBY, AY2T, 1, AY2, W, 3, 3, RY, SY, WAY, BY, WY, IERR)
C
WRITE(6,681) IERR, NDBY
YSE = (SY(1)/NDBY)**0.5
WRITE(6,7051) BY(1,1), BY(2,1), BY(3,1), YSE, NCULL, IERR, NDBY
C
6250 CONTINUE
C250 IF (ZSE.LT.1.)GOTO 6260
CALL LSQPOL(1600, NDBZ, A22T, 1, A22, W, 3, 3, RZ, SZ, WAZ, BZ, WZ, IERR)
C
WRITE(6,681) IERR, NDBZ
ZSE = (SZ(1)/NDBZ)**0.5
C
WRITE(6,7052) BZ(1,1), BZ(2,1), BZ(3,1), ZSE, NCULL, IERR, NDBZ
7051 FORMAT(2X,'Y 0TH, IST, 2ND = ',3(IX, E12.4),' YSE = ',
* E12.4, ' NCULL, IE,NDBY= ',3(IX, I5))
7052 FORMAT(2X,'Z 0TH, IST, 2ND = ',3(IX, E12.4),' ZSE = ',E12.4,
* ' NCULL, IE,NDBZ= ',3(IX, I5))
C
6260 CONTINUE
C
C CULL THE DATA SET FURTHER
C
NCULL = NCULL + 1
IF (NCULL.GT.2) GO TO 7026
GO TO 7025
7026 CONTINUE
C
C SHOULD NOW HAVE DATA WHICH DOES NOT REQUIRE FURTHER CALLS
C
7011 FORMAT(2X,'XSE YSE ZSE= ',3(2X,E12.4))
C
C GENERATE RANDOM NUMBERS TO REPLACE 9999'S WITH
C
   X = 0.0
   DO 1400 J = 1,NAD
       XRAN(J) = XNRANF(X)
   X = 0.0
1400
C
C REPLACE 9999 ELEMENTS OF ARRAYS WITH RANDOM NUMBERS
C
   DO 830 J = 1,NAD
       TIM = AT(J)
   C IF(XSE.LT.1.) GO TO 9831
       IF (AAX(J).EQ.9999.) THEN
        AAX(J) = XRAN(J)*XSE + BX(1,1) + BX(2,1)*TIM + BX(3,1)*TIM**2
       END IF
9831 CONTINUE
C831    IF (YSE.LT.1.) GOTO 9833
       IF (AAY(J).EQ.9999.) THEN
        AAY(J) = XRAN(J)*YSE + BY(1,1) + BY(2,1)*TIM + BY(3,1)*TIM**2
       END IF
9833 CONTINUE
C833    IF (ZSE.LT.1.) GO TO 9832
       IF (AAZ(J).EQ.9999.) THEN
        AAZ(J) = XRAN(J)*ZSE + BZ(1,1) + BZ(2,1)*TIM + BZ(3,1)*TIM**2
       END IF
9832 RTIM = AT(J) + TBEG(1)
       WRITE(14) RTIM, AAX(J), AAY(J), AAZ(J), TX(J), TY(J), TZ(J)
C
C 830 CONTINUE
C
   DO 526 I = 1,1600
       RX(I,1) = 0.
       RY(I,1) = 0.
       RZ(I,1) = 0.
   DO 524 J = 1,6
       WX(I,J) = 0.
       WY(I,J) = 0.
   DO 529 M = 1,6
       WAX(K,M) = 0.
529 CONTINUE
C
WAY(K,M) = 0.
WAZ(K,M) = 0.
CONTINUE

CALCULATE STATS ON DATA AFTER IT HAS BEEN FILLED

CALL LSQPOL(1600,NAD,AT,1,AAX,W,3,RX,SX,WAX,BX,WX,IERR)

WRITE(6,681) IERR,NAD
XSE = (SX(1)/NAD)**0.5
WRITE(6,9150) BX(1,1),BX(2,1),BX(3,1),XSE,NCULL,IERR,NAD
9150 FORMAT(2X,'FILLED X 0TH,1ST,2ND',3(1X,E12.4),' XSE = ',E12.4,
  * ' NCULL,IE,NAD= ',3(I4))
9151 FORMAT(2X,'FILLED Y 0TH,1ST,2ND',3(1X,E12.4),' YSE = ',E12.4,
  * ' NCULL,IE,NAD= ',3(I4))
9152 FORMAT(2X,'FILLED Z 0TH,1ST,2ND',3(1X,E12.4),' ZSE = ',E12.4,
  * ' NCULL,IE,NAD= ',3(I4))

CALL LSQPOL(1600,NAD,AT,1,AAY,W,3,RY,SY,WAY,BY,WY,IERR)

WRITE(6,681) IERR,NAD
YSE = (SY(1)/NAD)**0.5
WRITE(6,9151) BY(1,1),BY(2,1),BY(3,1),YSE,NCULL,IERR,NAD

CALL LSQPOL(1600,NAD,AT,1,AAZ,W,3,RZ,SZ,WAZ,BZ,WZ,IERR)

WRITE(6,681) IERR,NAD
ZSE = (SZ(1)/NAD)**0.5
WRITE(6,9152) BZ(1,1),BZ(2,1),BZ(3,1),ZSE,NCULL,IERR,NAD

NPC = NPC + 1

CONTINUE CYCLES

GO TO 1250
1307 WRITE(6,943)
943 FORMAT(2X,'ONE ARRAY HAS NO DATA IN IT')
GO TO 1311

CONTINUE
NSAT = NSAT + 1
IF (NSAT.EQ.1) THEN
WRITE(6,949)
949 FORMAT(2X,'X SATURATION REACHED ,SIMPLY TRANSFER DATA')
END IF
WRITE REMAINING DATA IN BUFFET TO OUTPUT TAPE

DO 1450 J = 1, NAD
WRITE(14) ORT(J),AAX(J),AAY(J),AAZ(J),TX(J),TY(J),TZ(J)
CONTINUE

CONTINUE TO TRANSFER DATA WITH NO CURLLING

WRITE (6, 6079) TYME,AX,AY,AZ
FORMAT (2X, 'LAST TYME,AX,AY,AZ = ', 4(2X, F15.4))

WRITE(14) EOF

END
PROGRAM INTTIM (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE8, TAPE9)

C

INTAPE = 8
REWIND (INTAPE)
TBEG = 61260
TLAST = 62696

ESTABLISH TIME PERIOD OVER WHICH TO AVERAGE

NSEC = 0
NRD = 0
NT = 0
TXAVG = 0.
TYAVG = 0.
TZAVG = 0.
NPRT = 0
NRDX = 0
NRDY = 0
NRDZ = 0
AVGX = 0.
AVGY = 0.
AVGZ = 0.

10 NSEC = NSEC+1
   TST = TBEG+0.507
C
   TEND = TST+1.0

20 READ (INTAPE.END=90) TYME, AX, AY, AZ, TX, TY, TZ
   IF (TYME.GT.TLAST) GO TO 90
   IF (TYME.LT.TST) GO TO 20
   IF (TYME.GT.TEND) GO TO 70
30 NRD = NRD+1
   IF (NRD.EQ.1) T1 = TYME
C
C  NEED TO SKIP OVER 9999 POINTS WHICH WERE LEFT OVER FROM
C  FILTERING IN PROGRAM FILLCDE
C
C  IF (AX.EQ.9999.) GO TO 40
   AVGX = AVGX+AX
NRDX = NRDX+1

40 IF (AY.EQ.9999.) GO TO 50
    AVGY = AVGY+AY
    NRDY = NRDY+1

50 IF (AZ.EQ.9999.) GO TO 60
    AVGZ = AVGZ+AZ
    NRDZ = NRDZ+1

60 CONTINUE
    TXAVG = TXAVG + TX
    TYAVG = TYAVG + TY
    TZAVG = TZAVG + TZ
    NT = NT + 1
    GO TO 20

70 CONTINUE

T2 = Tyme

CALCULATE AVERAGE AND WRITE OUT
FOR THE CASE WHEN SATURATION HAS OCCURED YET THRUSTING
STILL ACCOUNTS FOR MOST OF THE DATA. REPLACE THE AVERAGE
WHICH IS UNDEFINED WITH DATA REPRESENTING SATURATION.

IF (NRDX.EQ.0) THEN
    AVGX =-8000.
    WRITE(6,7890) Tyme
    FORMAT(2X,'FALSE X SATURATION VALUE AT Tyme = ',F12.4)
ELSE
    AVGX = AVGX/NRDX
    END IF

IF (NRDY.EQ.0) THEN
    AVGY = 8000.
    WRITE(6,7891) Tyme
    FORMAT(2X,'FALSE Y SATURATION VALUE AT Tyme = ',F12.4)
ELSE
    AVGY = AVGY/NRDY
    END IF

IF (NRDZ.EQ.0) THEN
    AVGZ = -8000.
    WRITE(6,7892) Tyme
    FORMAT(2X,'FALSE Z SATURATION VALUE AT Tyme = ',F12.4)
ELSE
    AVGZ = AVGZ/NRDZ
    END IF
ELSE
AVGZ = AVGZ/NRDZ
END IF
TXAVG = TXAVG/NT
TYAVG = TYAVG/NT
TZAVG = TZAVG/NT
ITIME = INT((T1+T2)/2)
RTIME = FLOAT( ITIME)
WRITE (9) RTIME, AVGX, AVGY, AVGZ, TXAVG, TYAVG, TZAVG
NPRT = NPRT + 1
IF (NPRT. GT. 500) GO TO 3200
WRITE (6,80) RTIME, TI, T2, NRD, AVGX, AVGY, AVGZ, TXAVG, TYAVG, TZAVG
3200 CONTINUE
80 FORMAT (2X, 'RTIME T1 T2 NRD AVGX AVGY AVGZ ', /, 2X, 1X, F14 5, 2 (1X, *F14 5), I5,3(1X,F14 5), /, 1X, '3 TEMPS = ',3(1X,F14 5))

NRD = 0
NRDX = 0
NRDY = 0
NRDZ = 0
NT = 0
TXAVG = 0
TYAVG = 0
TZAVG = 0
AVGX = 0
AVGY = 0
AVGZ = 0

CALCULATE NEW TIME PERIOD

NSEC = NSEC+ 1
TST = TEND
TEND = TST+ 1.0

GO TO 30

90 WRITE (6, 100)
T2 = TYME
100 FORMAT (2X, 'END OF TAPE 8 ENCONTORED' )
C    AVERAGE ANY DATA IN THE BUFFER
C
AVGX = AVGX/NRDX
AVGY = AVGY/NRDY
AVGZ = AVGZ/NRDZ
TXAVG = TXAVG/NT
TYAVG = TYAVG/NT
TZAVG = TZAVG/NT
ITIME = INT((T1+T2)/2)
RTIME = FLOAT(ITIME)
WRITE (9) RTIME,AVGX,AVGY,AVGZ, TXAVG, TYAVG, TZAVG
WRITE (6,110) RTIME,NRD,AVGX,AVGY, AVGZ, TXAVG, TYAVG, TZAVG
110 FORMAT (2X.'LST TIM NRD AVGX AVGY AVGZ= ',IX, FI4.5, I5,3(IX, FI4.5)
*/,1X,'3 TEMPS = ',3(IX,F14.5))
C
END
PROGRAM ORBLTA (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE7, TAPE8, TAPE1, TAPE3)

WRITTEN BY HIRAP PROJECT GROUP - JANUARY 1986

THIS PROGRAM PLOTS HIRAP ACCELERATION DATA IN MICRO-GS. ALTITUDE IS READ FROM A MERGED HCDAT FILE (IF AVAILABLE) AND IS LINEARLY INTERPOLATED TO MATCH THE HIRAP DATA RATE. ALTITUDE (METERS) IS THEN WRITTEN TO THE MICRO-G FILE AND WILL BE RETAINED FOR THE DATA ANALYSIS PROCESS SO THAT ACCELERATION DATA MAY BE ANALYSED WITH RESPECT TO TIME AND/OR ALTITUDE.

IMG = 1 INPUT DATA IN MICRO-GS (FILE 8)
ICG = 0 MICRO-GS NOT YET CORRECTED TO THE VEHICLE CG
ICG = 1 MICRO-GS CORRECTED TO THE VEHICLE CG
IFLT = STS FLIGHT NUMBER
IPLTMG = 0 ACCELEROMETER MICRO-G DATA NOT PLOTTED
        = 1 PLOTS ACCELEROMETER DATA IN MICRO-GS
IRECAL = 0 OLD (BEFORE MOD OR RECAL) HIRAP CALIBRATIONS
        = 1 RECALIBRATED HIRAP CALIBRATIONS, USING THE CORRECTED TEMPERATURE MONITOR VOLTAGE, VC
INORBIT = 0 GROUND CALIBRATION TEMPERATURE BIAS
         = 1 IN-ORBIT BIAS CALIBRATION I
         = 2 IN-ORBIT BIAS CALIBRATION II
         = 3 IN-ORBIT BIAS CALIBRATION III

**************
* NOTE : CURRENTLY THE IN-ORBIT BIAS CALIBRATIONS ARE FOR THE BEFORE-RECAL S/N 001 HIRAP ONLY
* (STS-6,7,8,11,13,AND 24)
* **************

PLOTTING ROUTINE INPUTS:

STRTALT = X AXIS ORIGIN IN KM
STOPALT = X AXIS FINAL ALT IN KM
STRTY = Y AXIS ORIGIN IN MICRO-GS (OR COUNTS)
STOPY = Y AXIS MAXIMUM MICRO-GS (OR COUNTS)
T1 = START TIME FOR FILE, SECONDS
T2 = STOP TIME FOR FILE, SECONDS
DAY = DAY OF DATA FILE TO BE PLOTTED

ALTTIC = INCREMENT FOR TICK MARKS ON X AXIS, KM

YTIC = INCREMENT FOR TICK MARKS ON Y AXIS, MICRO-GS

ALTLABL = INCREMENT OF LABELED ALT, KM

YLABL = INCREMENT OF LABELED MICRO-GS

DIMENSION T(10000), DATAPT(4), XCOUNT(3), XMICROG(3), X(10000),
* Z(10000), Y(10000), TIM(5000), TF(5000, 3), VC(5000, 3),
* NAMES(10), IUNITS(10), HDR(8),
* HDAT(3), HTIME(5000), HALT(5000)

ISTOP = 0
IPNT = 0
IFILE = 6
NFILE = 0
NP = 0
JPLT = 0

READ(5, 60) IFLT, IMG, ICG, IRECAL, INORBIT, IMEAN, IPLTMG
READ(5, 52) STRTALT, STOPALT, ALTTIC, ALTLABL, XLENG
READ(5, 52) STRTY, STOPY, YTIC, YLABL, YLENG
READ(5, 52, END=7) T1, T2, DAY
DELALT = (STRTALT - STOPALT)
DELMG = (STOPY - STRTY)
FLT = IFLT
SN = 1.0
IF (FLT.EQ.9.OR.FLT.EQ.26.OR.FLT.EQ.30) SN = 2.0
ISN = SN

IF (INORBIT.GT.4) THEN
WRITE(3, 64) INORBIT
ISTOP = 1
ENDIF

IF (SN.EQ.2.AND.INORBIT.NE.0) THEN
WRITE(3, 65) ISN, INORBIT
ISTOP = 1
ENDIF

IF (DELALT.LT.ALTIC) THEN
WRITE(3, 66) DELALT, ALTTIC
ISTOP = 1
ENDIF

IF (DELMG.LT.YTIC) THEN

ENDIF
WRITE (3, 67) DELMG, YTIC
ISTOP = 1
ENDIF

C

XX = DELALT/ALTTIC
NX = DELALT/ALTTIC
IF (NX.NE.XX) THEN
    WRITE (3, 68) DELALT, ALTTIC
    ISTOP = 1
ENDIF

C

YY = DELMG/YTIC
NY = DELMG/YTIC
IF (NY.NE.YY) THEN
    WRITE (3, 69) DELMG, YTIC
    ISTOP = 1
ENDIF

C

IF (ALTLABL.LT.ALTTIC) THEN
    WRITE (3, 70) ALTLABL, ALTTIC
    ISTOP = 1
ENDIF

C

IF (YLABL.LT.YTIC) THEN
    WRITE (3, 71) YLABL, YTIC
    ISTOP = 1
ENDIF

C

XX = ALTLABL/ALTTIC
NX = ALTLABL/ALTTIC
IF (NX.NE.XX) THEN
    WRITE (3, 72) ALTLABL, ALTTIC
    ISTOP = 1
ENDIF

C

YY = YLABL/YTIC
NY = YLABL/YTIC
IF (NY.NE.YY) THEN
    WRITE (3, 73) YLABL, YTIC
    ISTOP = 1
ENDIF

C

IF (ISTOP.EQ.1) STOP

C

IF (IMEAN.EQ.1) THEN
    KMAX = 1
ELSE
    KMAX = 350
    IF (FLT.EQ.32) KMAX = 225
ENDIF

KCNT = KMAX - 1
IF (NFILE.EQ.4) GO TO 15
IFILE = IFILE + 1
NFILE = NFILE + 1
REWIND(IFILE)
REWIND(NFILE)

IF (EOF(5).NE.0) THEN
    WRITE(6,57)
    STOP
ELSE
    WRITE(6,56) IFLT, IMG, ICG, IRECAL, IMEAN, IPLTMG, INORBIT
    WRITE(6,54) STRTALT, STOPALT, STRY, STOPY
    WRITE(6,53) IFILE, NFILE, T1, T2, DAY
ENDIF

READ(IFILE,END=10) ISN,NCHAN, (NAMES(I),I=1,NCHAN),
* (IUNITS(I),I=1,NCHAN), (HDR(I),I=1,8)

IF (EOF(IFILE).NE.0) THEN
    WRITE(6,51) IFILE
    STOP
ENDIF

ISN = 1
NCHAN = 5
NAMES(2) = 3HALT
NAMES(3) = 6HX-AXIS
NAMES(4) = 6HZ-AXIS
NAMES(5) = 6HY-AXIS
IUNITS(2) = 6HMETERS
IUNITS(3) = 6HMICRO-GS
IUNITS(4) = 6HMICRO-GS
IUNITS(5) = 6HMICRO-GS

READ (NFILE)
I = 0
CONTINUE
I = I + 1
READ(NFILE,END=17) (HDAT(J),J=1,3)
HTIME(I) = HDAT(1)
HALT(I) = HDAT(3)

IF (EOF(NFILE).NE.0) THEN
    NT = I - 1
    WRITE(6,62) NFILE,NT
    GO TO 1
ENDIF
GO TO 16
READ DATA FILE
CONTINUE
READ (IFILE, END=2) TIME, XMG, ZMG, YMG

IF (TIME .LT. T1) GO TO 1
IF (TIME .GT. T2) GO TO 15

CALL IUNI (5000, NT, HTIME, 1, HALT, 1, TIME, ALT, -1, IE)
WRITE (8) TIME, ALT, XMG, ZMG, YMG

IF (EOF (IFILE) .NE. 0) THEN
   WRITE (6, 58) TIME
   GO TO 15
ENDIF
GO TO 1

CONTINUE
WRITE (6, 63) (NAMES(I), I=1, NCHAN)
WRITE (6, 36) (IUNITS(I), I=1, NCHAN)

IF (IPLTMG .EQ. 1) THEN
   REWIND (8)
   READ (8)

   PLOT X-AXIS MICROGS AS A FUNCTION OF ALTITUDE
   IY = 1
   CALL PSEUDO
   CALL CALPLT (2.0, 2.0, -3)
   CALL DRAW1 (STRTALT, STOPALT, IY, FLT, SN, ICG, IMEAN, DAY,
            STRTY, STOPY, XLENG, YLENG, ALTTIC, YTIC, ALTLABL, YLABL)

   READ (8, END=21) TIME, ALT, XMG, ZMG, YMG
   ALT = ALT/1000.
   IF (ALT .GT. STRTALT) GO TO 20
   TPLT = (STRTALT - ALT) * (XLENG/DELMG)
   YPLT = (XMG - STRTY) * (YLENG/DELMG)
   IF (YPLT .GE. 0 .AND. YPLT .LT. YLENG) THEN
      CALL PNTPLT (TPLT, YPLT, 21, I)
   ENDIF
   IF (EOF (8) .NE. 0 .OR. TIME .GT. T2) GO TO 22
   IF (IMG .EQ. 1) THEN
      KCNT = KCNT + 1
      IF (KCNT .GE. KMAX) THEN
         WRITE (6, 59) TIME, ALT, XMG, ZMG, YMG
         KCNT = 0
      ENDIF
   ENDIF
GO TO 20

CONTINUE
GO TO 32
REWIND(8)
READ(8)

C PLOT Y-AXIS MICRO-GS AS A FUNCTION OF ALTITUDE

C
C IY = 2
CALL NFRA M
CALL CALPL T(2.0,2.0,-3)
CALL DRAW1 (STRALT,STOPALT,IY,FLT,SN,ICG,IMEAN,DAY,
 STRY ,STOPY,XLENG,YLENG,ALTTIC,YTIC,ALTLABL,YLABL)
READ (8,END=31) TIME,ALT,XMG,ZMG
ALT = ALT/1000.
IF (ALT.GT.STRALT) GO TO 30
TPLOT = (STRALT - ALT) * (XLENG/DELA LT)
YPLOT = (YMG - STRTY) * (YLENG/DELMG)
IF (YPLOT.GE.0.AND.YPLOT.LT.YLENG) THEN
 CALL PNTPL T(TPLOT ,YPLOT,21,1)
ENDIF
IF (EOF(8).NE.0.OR.TIME.GT.T2) GO TO 32
GO TO 30
C CONTINUE
REWIND (8)
READ (8)

C PLOT Z-AXIS MICRO-GS AS A FUNCTION OF ALTITUDE

C
C IY = 3
JPLOT = 0
CALL NFRA M
CALL CALPL T(2.0,2.0,-3)
CALL DRAW1 (STRALT,STOPALT,IY,FLT,SN,ICG,IMEAN,DAY,
 STRY ,STOPY,XLENG,YLENG,ALTTIC,YTIC,ALTLABL,YLABL)
READ (8,END=41) TIME,ALT,XMG,ZMG
ALT = ALT/1000.
IF (ALT.GT.STRALT) GO TO 40
TPLOT = (STRALT - ALT) * (XLENG/DELA LT)
YPLOT = (ZMG - STRTY) * (YLENG/DELMG)
IF (YPLOT.GE.0.AND.YPLOT.LT.YLENG) THEN
 CALL PNTPL T(TPLOT,YPLOT,21,1)
ENDIF
IF (EOF(8).NE.0.OR.TIME.GT.T2) GO TO 42
GO TO 40
C CALL CALPL T(0.0,0.0,999)
ENDIF
C STOP
C
FORMAT (/ 5X,7HFILE = ,115,5X,8HNCHAN = ,1I2,5X,15HFOUND HEADER =
 * ,8A10 / )
FORMAT (/ 5X,49HEOF FOUND WHEN ATTEMPTING TO READ HEADER ON FILE

126
* *II2 *)
52 FORMAT(5F8.0)
53 FORMAT(5X,8HFILE = ,II2,/ 5X,8HFILE = ,II1,/ 5X,5HT1 = ,1F10.2,
* / 5X,5HT2 = ,1F10.2, / 5X,6HDAY = ,1F4.0 /
54 FORMAT( / 5X,10HSTRTALT = ,1F10.2,5X,10HSTOPALT = ,1F10.2
* / 5X,8HSTRTY = ,1F10.2,7X,8HSTOPY = ,1F10.2 /
56 FORMAT(1H1,4X,10HFLT = STS ,1II2,5X,6HTMG = ,II1,5X,6HICG = ,II1,
* 5X,9HIRECAL = ,II1,5X,8HIMEAN = ,II1,5X,9HIXPLTMG = ,II1,
* 5X,10HINORBIT = ,II1 /
57 FORMAT(5X,42HEND OF FILE FOUND ON TAPE 5 - PROGRAM STOP /)
58 FORMAT( / 5X,29HNO POINTS FOUND AFTER TIME = ,1F10.3 /
59 FORMAT(5F12.3)
60 FORMAT(7I4)
61 FORMAT( / 10X,25HTOTAL NUMBER OF POINTS = ,II6 /
62 FORMAT( / 5X,32HNUMBER OF MERGED POINTS ON FILE ,II1,3H = ,
* 1I4 /)
63 FORMAT(1X,5(2X,A10))
64 FORMAT( / 5X,18HWRONG! (INORBIT = ,II3,1H),
* / 5X,46HINPUT VALUE OF INORBIT MUST BE = 0,1,2,3, OR,4,
* / 5X,9HTRY AGAIN )
65 FORMAT( / 5X,13HWRONG! (SN = ,II3,12H, INORBIT = ,II3,1H),
* / 5X,28HFOR S/N 002 INORBIT MUST = 0,
* / 5X,45H(CURRENTLY NO IN-ORBIT BIAS CALS FOR S/N 002)
66 FORMAT( / 5X,17HWRONG! (DELALT = ,1F8.0,11H, ALTTIC = ,1F8.0,1H),
* / 5X,34HDELALT MUST BE GREATER THAN ALTTIC,
* / 5X,9HTRY AGAIN)
67 FORMAT( / 5X,16HWRONG! (DELMG = ,1F8.0,9H, YTIC = ,1F8.0,1H),
* / 5X,31HDELMG MUST BE GREATER THAN YTIC,
* / 5X,9HTRY AGAIN)
68 FORMAT( / 5X,17HWRONG! (DELMG = ,1F8.0,11H, ALTTIC = ,1F8.0,1H),
* / 5X,44HDELMG MUST BE AN INTEGER MULTIPLE OF ALTTIC,
* / 5X,9HTRY AGAIN)
69 FORMAT( / 5X,16HWRONG! (DELMG = ,1F8.0,9H, YTIC = ,1F8.0,1H),
* / 5X,41HDELMG MUST BE AN INTEGER MULTIPLE OF YTIC,
* / 5X,9HTRY AGAIN)
70 FORMAT( / 5X,18HWRONG! (ALTLABL = ,1F8.0,11H, ALTTIC = ,1F8.0,1H),
* / 5X,35HALTLABL MUST BE GREATER THAN ALTTIC,
* / 5X,9HTRY AGAIN)
71 FORMAT( / 5X,16HWRONG! (YLABL = ,1F8.0,9H, YTIC = ,1F8.0,1H),
* / 5X,31HYLABL MUST BE GREATER THAN YTIC,
* / 5X,9HTRY AGAIN)
72 FORMAT( / 5X,18HWRONG! (ALTLABL = ,1F8.0,11H, ALTTIC = ,1F8.0,1H),
* / 5X,45HALTLABL MUST BE AN INTEGER MULTIPLE OF ALTTIC,
* / 5X,9HTRY AGAIN)
73 FORMAT( / 5X,16HWRONG! (YLABL = ,1F8.0,9H, YTIC = ,1F8.0,1H),
* / 5X,41HYLABL MUST BE AN INTEGER MULTIPLE OF YTIC,
* / 5X,9HTRY AGAIN)

C END
SUBROUTINE DRAW1 (STRTALT,STOPALT,IY,FLT,SN,ICG,IMEAN,DAY,
* STRTY,STOPY,XLENG,YLENG,ALTTIC,YTIC,ALTLABL,YLABL)

C THIS ROUTINE DRAWS GRID FOR HIRAP ACCELEROMETER VS TIME PLOTS
IY = 1 FOR X AXIS
      = 2 FOR Y AXIS
      = 3 FOR Z AXIS

DIMENSION TITLE1(5),TITLE2(4),TITLE3(8),YCHAR(3),LABEL1(6),
      * LABEL2(4),LABEL3(2),LABEL4(2)

DATA TITLE1 / 10HHIRAP S/N ,10H00 UNCORR,10HECTED ACCE,
      * 10HDATA, STS- /
DATA TITLE2 / 10HHIRAP S/N ,10H00 ACCELE,10HROMETER DA,
      * 10HTA - STS /
DATA TITLE3 / 10HCORRECTED ,10HTO VEHICLE,10H CG /
DATA TITLE3 / 10HCALCULATED,10H INDUCED A,10HCCELERATIO,
      * 10HN DUE TO H,10HIRAP OFFSET,10HT FROM VEH,
      * 10HICLE CG, ,10HSTS-32 /
DATA YCHAR / 1HX, 1HY, 1HZ /
DATA LABEL1 / 10H AXIS MIC,10HRO-GS AS A,10H FUNCTION ,
      * 10HOF ALTITUD,10HE, KM - DA,10HY /
DATA LABEL2 / 10HFILE : MG0,10H6II IN-OR,10HBIT BIAS C,
      * 10HALIBRATION /
DATA LABEL3 / 10HFILE : MG0,10H6II /
DATA LABEL4 / 10HONE SECOND,10H MEAN /

DEALT = (STRALT - STOPALT)
DELMG = (STOPY - STRTY)
BIGTICX = ALTLABL/ALTTIC
BIGTICY = YLABL/YTIC
Y2 = (-STRTY/DELMG) * YLENG
CALL CALPLT(0.0, YLENG, 2)
CALL CALPLT(XLENG, 0.0, 3)
CALL CALPLT(0.0,0.0,2)
CALL CALPLT(0.0,Y2,3)
CALL CALPLT(XLENG,Y2,2)

DRAW TICK MARKS ON X AXIS

NX = DEALT/ALTTIC
YBAR = 0.0
KCNT = 0
DELX = XLENG/NX
IEND = NX
DO 1 J = 1,IEND
   KCNT = KCNT + 1
   XBAR = DELX * J
   IF(KCNT.EQ.BIGTICX) THEN
      DELY = 0.14
      KCNT = 0
   ELSE
      DELY = 0.09
   ENDIF
   CALL CALPLT(XBAR,YBAR,3)
   CALL CALPLT(XBAR,DELY,2)
1 CONTINUE
DRAW TICK MARKS ON Y AXIS

NY = DELMG/YTIC
KCNT = 0
XBAR = 0.0
DELY = YLENG/NY
IEND = NY
DO 3 J = 1, IEND
KCNT = KCNT + 1
IF (KCNT.EQ.BIGTICY) THEN
  DX = 0.14
  KCNT = 0
ELSE
  DX = 0.09
ENDIF
YBAR = DELY * J
CALL CALPLT(XBAR, YBAR, 3)
CALL CALPLT(DX, YBAR, 2)
CONTINUE

LABEL X AXIS

DELEX = DELEX * BIGTIX
YBAR = -0.25
XVAL = STRTALT + ALTLABL
IEND = NX/BIGTIX + 1
DO 5 I = 1, IEND
XVAL = XVAL - ALTI./KBL
XST = - 0.20
XBAR = XST + DELEX * (I-1)
CALL NUMBER(XBAR, YBAR, 0.10, XVAL, 0.0, -1)
CONTINUE

LABEL Y AXIS

YST = -0.05
DELY = DELY * BIGTICY
IEND = NY/BIGTICY + 1
DO 6 I = 1, IEND
YBAR = YST + DELY * (I-1)
YVAL = STRTY + YLABL * (I-1)
XST = -0.45
IF(YVAL.LE.-1000) XST = -0.55
IF(YVAL.EQ.0.0) XST = -0.17
IF(YVAL.GT.0.0) XST = -0.39
IF(YVAL.GE.1000) XST = -0.49
CALL NUMBER(XST, YBAR, 0.10, YVAL, 0.0, -1)
CONTINUE

XBAR = 1.0
YBAR = YLENG + 1.0
IF(ICG.EQ.I) THEN
  CALL CHARACT(XBAR, YBAR, 0.10, TITLE2, 0.0, 40)
ELSE
SUBROUTINE ORBCALI (XCOUNT, TEMP, XMICROG)


DIMENSION B(3), BCOEF(4,3), SF(3), TEMP(3), XCOUNT(3), XMICROG(3), VX(3)

DATA BCOEF / 2135.0, -58.477, 0.44717, -0.0019150,
            -931.0, 26.091, -0.24936, 0.0011841,
            1653.0, -51.220, 0.30606, -0.00096787 /

SF(1) = -1.247237E-3
SF(1) = -1.247237E-3
SF(2) = 1.253821E-3
SF(3) = -1.246810E-3

COMPUTE ACCELEROMETER VOLTAGES FROM COUNTS DATA

DO 1 I = 1,3
   VX(I) = -10.0 + XCOUNT(I)/16383.0 * 20.0
   CONTINUE

COMPUTE BIAS FOR ALL THREE AXES

DO 2 I = 1,3
   B(I) = ((BCOEF(4,I) * TEMP(I) + BCOEF(3,I)) * TEMP(I) +
   * BCOEF(2,I)) * TEMP(I) + BCOEF(I,I)
   CONTINUE

COMPUTE ACCELERATION IN MICRO-GS

DO 3 I = 1,3
   XMICROG(I) = VX(I)/SF(I) - B(I)
   CONTINUE

RETURN
END

SUBROUTINE ORBCAL2(XCOUNT, TEMP, XMICROG)

THIS ROUTINE COMPUTES THE SCALE FACTOR AND THE BIAS IN THE X, Y,
AND Z ACCELEROMETERS FOR HIRAP S/N 001 USING THE CALCULATED TEMPERATURE
FROM THE OLD (BEFORE MOD) CALIBRATIONS. THE INPUT IS XCOUNT,
AND THE OUTPUT ACCELEROMETER DATA IN MICRO-GS IS XMICROG.
THE MICRO-G CALIBRATION ROUTINE USED HERE IS THE IN-ORBIT
FLIGHT DERIVED SET OF EQUATIONS FOR THE TEMPERATURE DEPENDENT
BIAS. THIS ROUTINE IS AN ATTEMPT AT A MORE ACCURATE BIAS
DETERMINATION THAN CURRENT GROUND CALIBRATION TECHNIQUES.

DIMENSION B(3), BCOEF(4,3), SF(3), TEMP(3), XCOUNT(3), XMICROG(3),
   * VX(3)

DATA BCOEF / 1241., 3.50654,-0.7643, 0.0057957,
   *  217., -39.143, 0.9526, -0.0060287,
   *  953., -2.17931,-0.6482, 0.0050837 /

SF(1) = -1.247237E-3
SF(2) = 1.253821E-3
SF(3) = -1.246810E-3

COMPUTE ACCELEROMETER VOLTAGES FROM COUNTS DATA

DO 1 I = 1,3
   VX(I) = -10.0 + XCOUNT(I)/16383.0 * 20.0
   CONTINUE
COMPUTE BIAS FOR ALL THREE AXES

DO 2 I = 1,3
   B(I) = ((BCOEF(4,I) * TEMP(I) + BCOEF(3,I)) * TEMP(I) +
   * BCOEF(2,I)) * TEMP(I) + BCOEF(1,I)
CONTINUE

COMPUTE ACCELERATION IN MICRO-GS

DO 3 I = 1,3
   XMICROG(I) = VX(I)/SF(I) - B(I)
CONTINUE

RETURN
END

SUBROUTINE ORBCAL3(XCOUNT, TEMP, XMICROG)

THIS ROUTINE COMPUTES THE SCALE FACTOR AND THE BIAS IN THE X, Y,
AND Z ACCELEROMETERS FOR HIRAP S/N 001 USING THE CALCULATED TEMPERATURE
FROM THE OLD (BEFORE MOD) CALIBRATIONS. THE INPUT IS XCOUNT,
AND THE OUTPUT ACCELEROMETER DATA IN MICRO-GS IS XMICROG.
THE MICRO-G CALIBRATION ROUTINE USED HERE IS THE IN-ORBIT
FLIGHT DERIVED SET OF EQUATIONS FOR THE TEMPERATURE DEPENDENT
BIAS. THIS ROUTINE IS AN ATTEMPT AT A MORE ACCURATE BIAS
DETERMINATION THAN CURRENT GROUND CALIBRATION TECHNIQUES.

DIMENSION B(3), BCOEF(4,3), SF(3), TEMP(3), XCOUNT(3), XMICROG(3),
  VX(3)

DATA BCOEF / 2465., -56.2229, 0.25003, 0.0,
  * -916., 22.503, -0.12344, 0.0,
  * 1737., -49.3751, 0.23295, 0.0 /

SF(1) = -1.247237E-3
SF(2) = 1.253821E-3
SF(3) = -1.246810E-3

COMPUTE ACCELEROMETER VOLTAGES FROM COUNTS DATA

DO 1 I = 1,3
   VX(I) = -10.0 + XCOUNT(I)/16383.0 * 20.0
CONTINUE

COMPUTE BIAS FOR ALL THREE AXES

DO 2 I = 1,3
   B(I) = ((BCOEF(4,I) * TEMP(I) + BCOEF(3,I)) * TEMP(I) +
   * BCOEF(2,I)) * TEMP(I) + BCOEF(1,I)
CONTINUE
SUBROUTINE XCALIB1 (XCOUNT, TEMP, XMICROG)

THIS ROUTINE COMPUTES THE SCALE FACTOR AND THE BIAS IN THE X, Y, AND Z ACCELEROMETERS FOR HIRAP S/N 001 USING THE CALCULATED TEMPERATURE FROM THE OLD (BEFORE MOD) CALIBRATIONS. THE INPUT IS XCOUNT, AND THE OUTPUT ACCELEROMETER DATA IN MICRO-GS IS XMICROG.

DIMENSION B(3), BCOEF(4,3), SF(3), TEMP(3), XCOUNT(3), XMICROG(3), * VX(3)

DATA BCOEF / 1598.1124, -33.327639, 0.18039702, -8.1516194E-4, -607.12033, 13.236853, -0.12054064, 6.4796047E-4, 1159.2864, -33.800394, 0.12802022, -2.9966848E-4 /

SF(1) = -1.247237E-3
SF(2) = 1.2538210E-3
SF(3) = -1.246810E-3

COMPUTE ACCELEROMETER VOLTAGES FROM COUNTS DATA

DO 1 I = 1,3
VX(I) = -10.0 + XCOUNT(I)/16383.0 * 20.0
CONTINUE

COMPUTE BIAS FOR ALL THREE AXES

DO 2 I = 1,3
B(I) = ((BCOEF(4, I) * TEMP(I) + BCOEF(3, I)) * TEMP(I) + * BCOEF(2, I)) * TEMP(I) + BCOEF(1, I)
CONTINUE

COMPUTE ACCELERATION IN MICRO-GS

DO 3 I = 1,3
XMICROG(I) = VX(I)/SF(I) - B(I)
CONTINUE

RETURN
END
SUBROUTINE XCALIB2 (XCOUNT, TEMP, XMICROG)

THIS ROUTINE COMPUTES THE SCALE FACTOR AND THE BIAS IN THE X, Y, AND Z ACCELEROMETERS FOR HIRAP S/N 002 USING THE CALCULATED TEMPERATURE FROM THE OLD (BEFORE RECAL) CALIBRATIONS. THE INPUT IS XCOUNT,
AND THE OUTPUT ACCELEROMETER DATA IN MICRO-GS IS XMICROG.

DIMENSION B(3), BCOEF(4,3), SF(3), TEMP(3), XCOUNT(3), XMICROG(3), VX(3)

DATA BCOEF / 1432.0258, -34.821054, 0.18235912, -5.9645121E-4,
           -3485.1197, 45.521179, -0.18902003, 5.8390244E-4,
           1017.3523, -21.741313, 0.12689434, -4.0574082E-4 /

SF(1) = -1.248570E-3
SF(2) = 1.269482E-3
SF(3) = -1.256565E-3

COMPUTE ACCELEROMETER VOLTAGES FROM COUNTS DATA

DO 1 I = 1,3
  VX(I) = -10.0 + XCOUNT(I)/16383.0 * 20.0
CONTINUE

COMPUTE BIAS FOR ALL THREE AXES

DO 2 I = 1,3
  B(I) = ((BCOEF(4, I) * TEMP(I) + BCOEF(3, I)) * TEMP(I) +
          BCOEF(2, I)) * TEMP(I) + BCOEF(1, I)
CONTINUE

COMPUTE ACCELERATION IN MICRO-GS

DO 3 I = 1,3
  XMICROG(I) = VX(I)/SF(I) - B(I)
CONTINUE

RETURN
END

SUBROUTINE RECAL1(XCOUNT, VTEMP, XMICROG)

THIS ROUTINE COMPUTES THE SCALE FACTOR AND THE BIAS IN THE X, Y,
AND Z ACCELEROMETERS FOR THE RECALIBRATED HIRAP S/N 001 USING
THE CORRECTED TEMPERATURE VOLTAGE, VTEMP.
THE INPUT ACCELEROMETER DATA IN COUNTS IS XCOUNT,
AND THE OUTPUT ACCELEROMETER DATA IN MICRO-GS IS XMICROG.

DIMENSION B(3), BCOEF(4,3), SF(3), SFCOEF(4,3), VTEMP(3),
               XCOUNT(3), XMICROG(3), VX(3)

DATA BCOEF / -111.50, -668.65, 11.765, 0.00, 71.90, 186.29, 0.00,
           3.6768, -282.50, -619.206, 26.135, 0.00 /
DATA SFCOEF / -1246.07, 0.00, 0.00, 0.020696, 1252.56, 0.00,
              0.00, 0.00, -1246.0, 5.1906, -2.7447,
              0.33579 /

COMPUTE ACCELEROMETER VOLTAGES FROM COUNTS DATA
DO 1 I = 1,3
VX(I) = -10.0 + XCOUNT(I)/16383.0 * 20.0
CONTINUE

COMPUTE BIAS FOR ALL THREE AXES

DO 2 I = 1,3
B(I) = ((BCOEF(4,I) * VTEMP(I) + BCOEF(3,I)) * VTEMP(I) +
      * BCOEF(2,I)) * VTEMP(I) + BCOEF(1,I)
CONTINUE

COMPUTE SCALE FACTOR FOR ACCELEROMETER

DO 3 I = 1,3
SF(I) = (((SFCOEF(4,1) * VTEMP(I) + SFCOEF(3, I)) * VTEMP(I) +
     * SFCOEF(2,I)) * VTEMP(I) + SFCOEF(1,I))/10.0**6
CONTINUE

COMPUTE ACCELERATION IN MICRO-GS

DO 4 I = 1,3
XMICROG(I) = VX(I)/SF(I) - B(I)
CONTINUE

RETURN
END

SUBROUTINE RECAL2 (XCOUNT, VTEMP, XMICROG)

THIS ROUTINE COMPUTES THE SCALE FACTOR AND THE BIAS IN THE X, Y,
AND Z ACCELEROMETERS FOR THE RECALIBRATED HIRAP S/N 002 USING
THE CORRECTED TEMPERATURE VOLTAGE, VTEMP.
THE INPUT ACCELEROMETER DATA IN COUNTS IS XCOUNT,
AND THE OUTPUT ACCELEROMETER DATA IN MICRO-GS IS XMICROG.

DIMENSION B(3), BCOEF(4,3), SF(3), SFCOEF(4,3), VTEMP(3),
     * XCOUNT(3), XMICROG(3), VX(3)

DATA BCOEF / 6429.8, -567.06, 39.292, -7.5809, -827.8, 1014.78,
     * -97.844, 11.805, 6100.3, -467.94, 79.069, -12.521 /
DATA SFCOEF / -1250.035, 3.372104, -1.283124, 0.1437924, 1271.533,
     * -1.344372, 0.0, 0.04102193, -1253.671, -2.584656,
     * 1.044413, -0.1370558 /

COMPUTE ACCELEROMETER VOLTAGES FROM COUNTS DATA

DO 1 I = 1,3
VX(I) = -10.0 + XCOUNT(I)/16383.0 * 20.0
CONTINUE

COMPUTE BIAS FOR ALL THREE AXES

DO 2 I = 1,3
B(I) = ((BCOEF(4,I) * VTEMP(I) + BCOEF(3,I)) * VTEMP(I) +
      * BCOEF(2,I)) * VTEMP(I) + BCOEF(1,I)
CONTINUE

COMPUTE SCALE FACTOR FOR ACCELEROMETER

DO 3 I = 1,3
SF(I) = (((SFCOEF(4,I) * VTEMP(I) + SFCOEF(3,I)) * VTEMP(I) +
* SFCOEF(2,I)) * VTEMP(I) + SFCOEF(1,I))/10.0**6
CONTINUE

COMPUTE ACCELERATION IN MICRO-GS

DO 4 I = 1,3
XMICROG(I) = VX(I)/SF(I) - B(I)
CONTINUE

RETURN
END

6 1 0 0 4 0 1
160.0 60.0 2.0 20.0 8.0
8000.0 2000.0 250.0 1000.0 5.0
64500.0 66500.0 99.0
PROGRAM TYMAVG (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE7, TAPE8,
* TAPE9)
C
DIMENSION DATA(7), NAMES(10), IUNITS(10), IHDR(8)
C
KOUNT1=0
KOUNT2=0
KNTX = 0
KNTZ = 0
KNTY = 0
SUMXMG=0.
SUMALT=0.
SUMYMG=0.
SUMZMG=0.
JMP=0
KHEAD=1
C
READ(7)
READ(7)
C
IF (KHEAD.EQ.1) GOTO 7
READ(7, END=1000) ISN,NCHAN, (NAMES(I), I=1,NCHAN), (IUNITS(I),
.I=1,NCHAN), (IHDR(I), I=1,8)
C
WRITE(9) ISN,NCHAN, (NAMES(I), I=1,NCHAN), (IUNITS(I),
.I=1,NCHAN), (IHDR(I), I=1,8)
C
NCHAN=4
C
READ(7, END=2000) (DATA(I), I=1,NCHAN)
KOUNT1=KOUNT1+1
TYM=DATA(1)
TYMINT=INT(TYM)
TYMDEC=TYM-TYMINT
HUNS=TYMDEC*100.
   IF (JMP.GT.0) GOTO 15
   IF (KOUNT2.GT.1) GOTO 15
   IF (INT(HUNS)/50..NE.1) GOTO 10
15   IF (INT(HUNS)/50..EQ.1.AND.KOUNT2.GT.1) GOTO 20
   KOUNT2=KOUNT2+1
C
C
C ALTH=DATA(2)
C XMG=DATA(3)
C ZMG=DATA(4)
C YMG=DATA(5)
C
   IF (DATA(2).EQ.99999.) GOTO 18
   SUMXMG=SUMXMG+DATA(2)
   KNTX=KNTX + 1
C
   SUMALT=SUMALT+DATA(3)
18   IF (DATA(3).EQ.99999.) GOTO 19
   SUMZMG=SUMZMG+DATA(3)
   KNTZ = KNTZ + 1
SUMYMG=SUMYMG+DATA(4)
KNTY = KNTY+1
19 CONTINUE
GOTO 10

C
C 20 AVXMG=SUMXMG/KNTX
C AVALT=SUMALT/KOUNT2
CAVZMG=SUMZMG/KNTZ
CAVYMG=SUMYMG/KNTY
AVTYM=TYMINT
IF(KNTX.LT.(KOUNT2*.8)) THEN
   PRINT*, 'TIME, KNTX, KOUNT2 : ', AVTYM, KNTX, KOUNT2
ENDIF
IF(KNTZ.LT.(KOUNT2*.8)) THEN
   PRINT*, 'TIME, KNTZ, KOUNT2 : ', AVTYM, KNTZ, KOUNT2
ENDIF
WRITE (9) AVTYM, AVXMG, AVZMG, AVYMG
WRITE (8,345) AVTYM, AVXMG, AVZMG, AVYMG
345 FORMAT (E12.6,1X,E12.6,1X,E12.6,1X,E12.6)
KOUNT2=0
KNTX = 0
KNTZ = 0
KNTY = 0
SUMXMG=0.
SUMALT=0.
SUMZMG=0.
SUMYMG=0.
JMP=1
GOTO 15
1000 GOTO 10
2000 STOP
END
PROGRAM MERG(INPUT, OUTPUT, TAPE1, TAPE2, TAPE3, TAPE4, TAPE5,
   . TAPE6=OUTPUT, TAPE7)
DIMENSION DAXZ(3,2000), SMOO(7), SMO(7)

TAPE (1) = ATRW       (EPOCH) 201 WORD FILE
  2 = XBET       (EPOCH) 66 WORD FILE
  3 = HIRPB     (DECIMAL TIME, NO HEADER)
  4 = XLAR    (DECIMAL TIME, TWO HEADER RECORDS)

-----------------------------------------------

TAPE (5) = MERGED FILES USING DECIMAL TIME
  6 = PRINT OUT AT TEN SECOND INTERVALS

-----------------------------------------------

DIMENSION DAT(201), UNITS(24), ALPHA(24), TITLES(8), DLAR(29), DXT(66)

TITLES (1)=10H STS-35
TITLES(2)=10H MERGED FI
TITLES (3)=10HLES USED I
TITLES(4)=10HN ANALYSIS
TITLES(5)=10H OF HIRAP
TITLES(6)=10HDATA
TITLES(7)=10H
TITLES(8)=10H

WGT=226613.*.4535924
XBETEP=18100.
ABETEP=1E6
DISJNT= 0.
TTRAN=0.00
AELE= 0.
ABFL= 0.
XTRAN=82300.
ZTRAN=97200.
IXSM=0
IZSM=0
XBIAS=0.0
ZBIAS=0.00
RTRAN=400000.

HIRAP HEADER READ FOR TEST RUN OF STS-24
READ(3,1666)
KVNT = 0

ABRIDGED 24-WORD DATA ARRAY

ALPHA (1)=10HTIME
UNITS (1)=10HSEC

ALPHA (2)=10HVEL
UNITS (2)=10HM/SEC

ALPHA (3)=10HALT
UNITS (3) = 10HMETERS

ALPHA (4) = 10HPLAT
UNITS (4) = 10HDEG

ALPHA (5) = 10HPLONG
UNITS (5) = 10HDEG

ALPHA (6) = 10HALPHA
UNITS (6) = 10HDEG

ALPHA (7) = 10HBETA
UNITS (7) = 10HDEG

ALPHA (8) = 10HP
UNITS (8) = 10HDEG/S

ALPHA (9) = 10HQ
UNITS (9) = 10HDEG/S

ALPHA (10) = 10HR
UNITS (10) = 10HDEG/S

ALPHA (11) = 10HPDOT
UNITS (11) = 10HDEG/S2

ALPHA (12) = 10HQDOT
UNITS (12) = 10HDEG/S2

ALPHA (13) = 10HRDOT
UNITS (13) = 10HDEG/S2

ALPHA (14) = 10HAELE
UNITS (14) = 10HDEG

ALPHA (15) = 10HABFL
UNITS (15) = 10HDEG

ALPHA (16) = 10HWGT
UNITS (16) = 10HKG

ALPHA (17) = 10HDUMY1
UNITS (17) = 10H

ALPHA (18) = 10HRL
UNITS (18) = 10HKG/M3

ALPHA (19) = 10HTIL
UNITS (19) = 10HKELV

ALPHA (20) = 10HPIL
UNITS (20) = 10HN/M2

ALPHA (21) = 10HWML
UNITS(21)=10HKG/KMOL

ALPHA(22)=10HDUMY2
UNITS(22)=10H

ALPHA(23)=10HAX
UNITS(23)=10HM/S2

ALPHA(24)=10HAZ
UNITS(24)=10HM/S2

WRITE HEADERS: TAPE(5) AND PRINT - OUT

ISEQ=1
NWDS=24
WRITE(5) ISEQ, NWDS, (ALPHA(I), I=1, NWDS), (UNITS(I), I=1, NWDS),
.TITLES

WRITE(6,1001)
1001 FORMAT(/2X,*....HIRAP ANALYSIS MERGED FILE HEADER...*/)
WRITE(6,1002) TITLES

WRITE(6,1003) (I, ALPHA(I), UNITS(I), I=1, NWDS)
1003 FORMAT(2X,*....LABELS AND UNITS FOR DATA WORDS .....*/
.(3X,*(*, I3,*)*, 2X, A10, 3X, A10))

WRITE(6,1004)
1004 FORMAT(/4X,*TIME ALT WML AELE ABFL AX
* AZ VEL TIL PIL RL ALPHA*/)

INITIALIZE COUNTERS
TIME=-1.
N6=10
NZP=1

OPTIONAL HIRAP AX , AZ SMOOTHING
IF(IZSM.EQ.1) GOTO81
IF(IYSM.EQ.1) GOTO81
GOTO8

K=0
J=0
READ(3,1666) (DAXZ(I,J), I=1, 3)
IF(EOF(3)) 2, 1
K=K+1
DAXZ(1,K)=DAXZ(1,K+3)
C

IF (IXSM.EQ.1) GOTO 82
DAXZ(2,K)=DAXZ(2,K+3)
GOTO 91

82 DO 83 I=1,7
SMO(I)=DAXZ(2,K-1+I)

83 SMOO(I)=SMO(I)
NS=7

84 DM=SMOO(1)
IM=1

85 DO 85 I=2,NS
IF (SMOO(I).LE.DM) GOTO 85
DM=SMOO(I)
IM=I

85 CONTINUE

86 CONTINUE

IF (NS.LE.4) GOTO 88
ISMO=0

86 DO 86 I=1,NS
IF (I.EQ.IM) GOTO 86
ISMO=ISMO+1
SMO(ISMO)=SMOO(I)

87 CONTINUE

87 CONTINUE
NS=NS-1

88 DAXZ(2,K)=DM

C

91 IF (IZSM.EQ.1) GOTO 92
DAXZ(1,K)=DAXZ(1,K+3)
GOTO 93

92 DO 3 I=1,7
SMO(I)=DAXZ(3,K-1+I)

3 SMOO(I)=SMO(I)
NS=7

4 DM=SMOO(1)
IM=1

5 CONTINUE

5 CONTINUE

IF (NS.LE.4) GOTO 11
ISMO=0

5 CONTINUE

DO 6 I=1,NS
IF (I.EQ.IM) GOTO 6
ISMO=ISMO+1
SMO(ISMO)=SMOO(I)

6 CONTINUE

6 CONTINUE
NS=NS-1

7 SMOO(I)=SMO(I)
GOTO 4

7 SMOO(I)=SMO(I)
GOTO 4

11 DAXZ(3,K)=DM

142
C 93 IF (K.LT.J-7) GOTO2
DHP1=DAXZ (1,1)
DHP2=DAXZ (2,1)
DHP3=DAXZ (3,1)
C
C INITIAL READ OF HEADERS AND RECORDS (ABET, XBET, HIRAP, XLAR)
GOTO9
8 READ (3,1666) DHPI, DHP2, DHP3
   DHP1=DHP1*1.
C
C WRITE (6,1666) DHPI, DHP2, DHP3
9 READ (1)
   READ (2)
   READ (4)
   READ (4)
   READ (4) (DLAR(I), I=1,29)
C
10 IF (TIME.LT.ABETEP+TTRAN-1.) GOTO20
C
C READ ABET RECORDS
12 READ (1) (DAT(I), I=1,201)
   IF (EOF (1)) 999, 15
15 IF (DAT (1)+ABETEP.LE.TIME) GOTO12
   TIME=DAT (1)+ABETEP
   VEL=DAT (2) *.3048
   ALT=DAT (5) *.3048
   PLAT=DAT (6)
   PLONG=DAT (7)
   AALPHA=DAT (10)
   BETA=DAT (9)
   P=DAT (37)
   Q=DAT (38)
   R=DAT (39)
   PDOT=DAT (43)
   QDOT=DAT (44)
   RDOT=DAT (45)
   AELE=DAT (65)
   ABFL=DAT (69)
   AX=DAT (40) *.3048
   AZ=DAT (42) *.3048
   RL=DAT (28)*14.5939/.3048/.3048/.3048
   TIL=DAT (27)*5./9.
   PIL=DAT (26)*4.448222/.3048/.3048
   WML=DAT (22)*DAT (22)*1.4*8314.34*TIL/VEL/VEL
   GOTO30
C
C READ XBET RECORDS
20 READ (2) (DXT(I), I=1,66)
   IF (EOF (2)) 999, 25
25 TIME=DXT (1)+XBETEP
   IF (TIME.LT.18800.) GOTO 20
   VEL=DXT (2) *.3048
   ALT=DXT (5) *.3048-DISJNT
   PLAT=DXT (6)
PLONG=DXT(7)
AALPHA=DXT(10)
BETA=DXT(9)
P=DXT(49)
Q=DXT(50)
R=DXT(51)
PDOT=DXT(64)
QDOT=DXT(65)
RDOT=DXT(66)
AX=DXT(52) *.3048
AZ=DXT(54) *.3048
RL=DXT(45)*14.5939/.3048/.3048/.3048
TIL=DXT(44)*5.9.
PII=DXT(43)*4.448222/.3048/.3048
WML=DXT(41)*DXT(41)*1.4*8314.34*TIL/VEL/VEL

C

30 IF(ALT.LT.XTRAN)GOTO050
C
C FILL-IN HIRAP AX
IF (DHP1.GE.TIME-.5)GOTO40
IF (IZSM.NE.1)GOTO38
NZP=NZP+1
IF (NZP.GT.K)GOTO999
DHP1=DAXZ(1,NZP)
DHP2=DAXZ(2,NZP)
DHP3=DAXZ(3,NZP)
GOTO30

38 READ (3,1666)DHP1,DHP2,DHP3
   DHP1=DHP1*1.
C
WRITE(6,1666)DHP1,DHP2,DHP3
IF (EOF(3)) 997, 30

40 AX=-9999.
   IF (DHP1.LE.TIME+.5)AX=(DHP2+XBIAS)*32.1747/1.0E06*.3048
C
IF ((DHP1.LE.TIME+.5).AND.(DHP2.GT.9000.))AX=DHP2
C
50 IF (ALT.LT.ZTRAN)GOTO070
C
C FILL-IN HIRAP AZ
IF (DHP1.GE.TIME-.5)GOTO60
IF (IZSM.NE.1)GOTO58
NZP=NZP+1
IF (NZP.GT.K)GOTO999
DHP1=DAXZ(1,NZP)
DHP2=DAXZ(2,NZP)
DHP3=DAXZ(3,NZP)
GOTO50

58 READ (3,1666)DHP1,DHP2,DHP3
C
WRITE(6,2323)DHP1,DHP2,DHP3
2323 FORMAT (3(1X,E12.6))
   DHP1=DHP1*1.
   IF (EOF(3)) 999, 50

60 AZ=-9999.
   IF (DHP1.LE.TIME+.5)AZ=(DHP3+ZBIAS)*32.1747/1.0E06*.3048
WRITE(6,2323)DHP1,DHP2,DHP3
IF((DHP1.LE.TIME+.5).AND.(DHP3.GT.-9000.))AZ=DHP3

70 IF(ALT.LT.RTRAN)GOTO90
C

C FILL-IN XLAR ATMOS DATA
IF(DLAR(4).GE.TIME-.5)GOTO80
READ(4)(DLAR(I),I=1,29)
IF(EOF(4))999,70

80 IF(DLAR(4).GT.TIME+.5)GOTO90
RL=DLAR(8)
TIL=DLAR(6)
PIL=DLAR(7)
WML=DLAR(25)

C

C LOAD MERGED ARRAY
C

90 DAT(1)=TIME
DAT(2)=VEL
DAT(3)=ALT
DAT(4)=PLAT
DAT(5)=PLONG
DAT(6)=AALPHA
DAT(7)=BETA
DAT(8)=P
DAT(9)=Q
DAT(10)=R
DAT(11)=PDOT
DAT(12)=QDOT
DAT(13)=RDOT
DAT(14)=AELE
DAT(15)=ABFL
DAT(16)=WGT
DAT(17)=-9999.
DAT(18)=RL
DAT(19)=TIL
DAT(20)=PIL
DAT(21)=WML
DAT(22)=-9999.
DAT(23)=AX
DAT(24)=AZ

C

C WRITE DATA : TAPE(5) AND PRINT - OUT
C

WRITE(5) (DAT(I),I=1,24)
WRITE(7,2999)DAT(1),DAT(3),DAT(23),DAT(24),DAT(6)
FORMAT(5(1X,E12.6))

N6=N6+1
IF(N6.LT.10)GOTO10
WRITE(6,1005)TIME,ALT,WML,AELE,ABFL,AX,AZ,VEL,TIL,PIL,RL,DAT(6)
1005 FORMAT(6X,1F9.3,1X,1F8.1,1F8.2,2F8.1,7E12.5)
    N6=0
    GOTO10
C
997    WRITE(6,1006)TIME,DHP1
1006    FORMAT(2X,*...FOUND EOF ON HIRAP TAPE AT TIME...*/2(F10.5))
1666    FORMAT(F9.3,1X,F9.3,1X,F9.3)
C
999    STOP
    END
PROGRAM MTEST88(INPUT,OUTPUT,TAPE1,TAPE5=INPUT,TAPE6=OUTPUT,TAPE3,
* TAPE4,TAPE7)

C
C THIS PROGRAM PERFORMS ANALYSIS COMPUTATIONS ON THE MERGED HIRAP,
C OI, TRAJECTORY, AND EXTENDED LAIRS FLIGHT DATA FILES.
C
C** THIS PROGRAM ALSO ALLOWS USER TO PLOT ANY ELEMENT OF THE 24-WORD
C MERGED DATA FILE AND THE ANALYSIS COMPUTATIONS WHICH ARE STORED
C IN ACCESS-RECORDS 25-66.
C
C** AGAINST OTHER ELEMENTS OF THE SAME FILE
C
C AN ADDITIONAL UTILITY STORES THE PLOT DATA ON TAPE3
C WITH NEW PLOTS OF THE SAME FORMAT FOR USE IN FUTURE COMPARISONS
C
C C** USER INPUT, IN ADDITION TO ELEMENTS TO BE PLOTTED, INCLUDES
C X-AXIS AND Y-AXIS RANGES, AND AXIS LENGTHS.
C
C DIMENSION DAT(66),ALPHA(66),UNITS(66),TITLES(8)
DIMENSION BCDX(2),BCDY(2),ANS(4)
DIMENSION XWK(500),YWK(500,2),WWX(500),WWZ(500),RWK(500,2)
DIMENSION BAC(4,2),WK(500,4),SWK(2),AWK(4,4)
NAMELIST/OPT/XL,YL,ALTSTRT,ALTEND,GAPSTRT,GAPEND,ISYMB,IGRID,ITPLT
.,ITAG,AFIT,NFIT,XBIAS,ZBIAS,ARHO,IDMP,IATM,QFAC
NWDS=66
C
C...........NAMELIST/OPT/ INPUTS...........................................
C
C..XL.......X-AXIS LENGTH(INCHES),DEFAULT VALUE IS 4.0
C..YL.......Y-AXIS " "
C..ALTSTRT..START ALT. (METERS) FOR SEARCH,DEFAULT IS 180000.
C..ALTEND...STOP " " 60000.
C..GAPSTRT..UPPER ALT GAP , DEF=400000.
C..GAPEND...LOWER ALT GAP , " "
C..ISYMB...SYMBOL OPTION FOR PLOT
C 0-LINE (DEFAULT)
C 1-POINTS
C..IGRID...PLOT GRID OPTION
C 0-NO GRID (DEFAULT)
C 1-GRID
C..ITPLT...AXIS FORMAT
C 0-NORMAL PLOT (DEFAULT)
C 1-T PLOT OPTION
C..ITAG....USE TO OVERLAY FOLLOWING PLOT ( 1-OVERLAY) DEF.=0
C..AFIT....FITS A NFIT ORDER FUNCTION TO ACCEL. DATA , DEF.=400000.
C..NFIT ABOVE ALT=AFIT , WHICH IS THEN USED IN PLACE OF DEF=1 MAX: NFIT=
C 3
C RAW DATA FOR AERO CALCULATIONS.
C..XBIAS...X AND Z-AXIS ACCEL. BIASES , IN ADDITION DEFS. =0.
C..ZBIAS TO INPUT DATA.
C..ARHO....TRANSITION ALTITUDE FOR COMPOSITE AXIAL/NORMAL DENS. CALCS. DEF=890
00.
C..IDMP....DUMPS ALL DATA IF IDMP=1, BYPASSES AERO CALCS. DEF=0
C..IATM....MEAN MOL WT ATM MODEL
C 0-LAIRS (DEFAULT)
C 1-MODEL
C..QFAC....SCALING FACTOR , DEF=1.
C
C............FORMATTED INPUTS...........................................
C..CARD 1---NECESSARY INPUT TO SELECT PLOT PARAMETERS....................
C
C C.... XAXIS (A10) ALPHA FOR INDEPENDENT VARIABLE
C C.... YAXIS (A10) ALPHA FOR DEPENDENT VARIABLE
C.... XMAX (F10.3) MAXIMUM VALUE FOR X
C.... XMIN (F10.3) MINIMUM VALUE FOR X
C.... YMAX (F10.3) MAXIMUM VALUE FOR Y
C.... YMIN (F10.3) MINIMUM VALUE FOR Y
C
C INITIALIZE PLOT UTILITY
CALL PSEUDO
CALL CALPLT(5.,3.,-3)
NTAG=0
L=0
K=1
C
C READ,WRITE HEADER ON INPUT DATA FILE
DO 25 I=1,NWDS
DAT(I)=0.
ALPHA(I)=IOH
25 UNITS(I)=10H
WRITE(6,1000)
1000 FORMAT(1H1)
READ(1) ISEQ,NWSS,(ALPHA(I),I=1,NWSS),(UNITS(I),I=1,NWSS),
TITLES
IF(EOF(1)) 500,5555
5555 CONTINUE
REWIND 1
C
C*******************************
C
C GENERALIZED 66-WORD ARRAY
C
C ALPHA,UNITS( 1 - 24 ) CONTAIN INPUT DATA
C
C ALPHA(25)=IOHHDIST
UNITS(25)=IOHKILOMETERS
C
C UNITS(3)=IOHKILOMETERS
UNITS(1)=IOHSECX100
C
C ALPHA(26)=IOHACCN
UNITS(26)=IOHLOG10MIC-G
C
C ALPHA(27)=IOHACCA
UNITS(27)=IOHLOG10MIC-G
C
C ALPHA(28)=IOHFLOD
C
C ALPHA(29)=IOHFNOA
C
C ALPHA(30)=IOHCDB
C
C ALPHA(31)=IOHCXELFM
UNITS(31)=IOH
C
C ALPHA(32)=IOHCZELFM
UNITS(32)=IOH
C
C ALPHA(33)=IOHPN
UNITS(33)=IOHLOG10
C
C ALPHA(34)=IOHCABI
C
C ALPHA(35)=IOHXXKN2
UNITS(35)=IOHLOG10
C
C ALPHA(36)=IOHTW
UNITS(36)=IOHK

148
ALPHA (37) = 10HAMACH
ALPHA (38) = 10HACCPN
UNITS (38) = 10HLOG1OMIC-G
ALPHA (39) = 10HCN2
ALPHA (40) = 10HCNB1
ALPHA (41) = 10HHRON/RHOA
ALPHA (42) = 10HCA2
ALPHA (43) = 10HGAEL
UNITS (43) = 10HGRAPHUNITS
ALPHA (44) = 10HGABEL
UNITS (44) = 10HGRAPHUNITS
ALPHA (45) = 10HGAEL
UNITS (45) = 10HGRAPHUNITS
ALPHA (46) = 10HCA2
ALPHA (47) = 10HRHO/R76
ALPHA (48) = 10HRHO
UNITS (48) = 10HKG/M3
ALPHA (49) = 10HC1
ALPHA (50) = 10HCN1
ALPHA (51) = 10HFLD1
ALPHA (52) = 10HMW76
UNITS (52) = 10HKG/KMOLE
ALPHA (53) = 10HCXBFM
ALPHA (54) = 10HCZBFM
ALPHA (55) = 10HPDIF
ALPHA (56) = 10HCAB2
ALPHA (57) = 10HCNB2
ALPHA (58) = 10HCDB2
ALPHA (59) = 10HXKN
UNITS (59) = 10H (LOG10)
ALPHA (60) = 10HFLD2
ALPHA (63) = 10HAELE
UNITS (63) = 10HDEG
SET CONSTANTS
SS6 = 2690.* .3048*.3048
PI = 4.* ATAN (1.)
PIS = PI/180.
C KN VS RHO  62ATM  KN=10^A*RHO^B  (RHO LBM/FT3)

AKN=-7.7176662-1.5973
BKKN=-.99057554

EXP (-A2ZE*(B2ZE-LOGI0(KN))^C2ZE)  KN < B2ZE

A2ZE=.2262
B2ZE=1.2042
C2ZE=1.8410
A1ZE=.2998
B1ZE=1.3849
C1ZE=1.7128

C***********************************************************************

WRITE(6,1001)
1001 FORMAT(/2X,*...HIRAP ANALYSIS MERGED FILE HEADER....*/)
WRITE(6,1002)TITLES
1002 FORMAT(2X,8A10/)
WRITE(6,1003) (I,ALPHA(I),UNITS(I),I=1,24)
1003 FORMAT(2X,*...LABELS AND UNITS FOR DATA WORDS ..../*)
     .((3X,*(*,I3,*)*,2X,A10,3X,A10))
WRITE(6,1038)
1038 FORMAT(/4X,*ACCESS RECORDS OF ANALYSIS COMPUTATIONS*/)
WRITE(6,1003) (I,ALPHA(I),UNITS(I),I=25,NWDS)

C READ / WRITE NAMELIST OPT
5 CONTINUE

XL=4.
YL=4.
ALTSTR=180000.
ALTEND=60000.
GAPSTR=400000.
GAPEND=400000.
ISYMB=0
IGRID=0
ITPLT=0
ITAG=0
AFIT=400000.
NFIT=1
XBIAS=0.0
ZBIAS=0.0
ARHO=89000.
IDMP=0
IATM=0
QFAC=1.

C READ OPT
C---------------------------------------------------------------
666 WRITE(6,OPT)
C
C DUMMY HEADER READ
READ(1)
C
C READ FORMATTED USER INPUT
555 CONTINUE
READ(5,6) XAXIS,YAXIS,XMAX,XMIN,YMAX,YMIN
6 FORMAT (2A10, 4F10.3)
   IF (EOF(5)) 260, 7

C IDENTIFY DATA WITH ALPHA INFO.
7    DO 9 I=1, NWDS
     IF (XAXIS.EQ.ALPHA(I)) IX=I
     IF (YAXIS.EQ.ALPHA(I)) IY=I
     IF (IX.EQ.0) GO TO 4009
     IF (IY.EQ.0) GO TO 4010
   9 CONTINUE
   WRITE (6, 1000)
   WRITE (6, 2000) K
   WRITE (6, 1011) XL, YL
   IF (IGRID.EQ.0) WRITE (6, 1017)
   IF (IGRID.EQ.1) WRITE (6, 1018)
   WRITE (6, 1021) ALTSTRT, ALTEND
   WRITE (6, 1004) ALPHA(IX), XMIN, XMAX
   WRITE (6, 1005) ALPHA(IY), YMIN, YMAX
   IF (IX.EQ.IY) GOTO 650

C TIME, ALTITUDE SCALING FOR PLOTS
C
   ITIME=1
   IALT=3
   ALTEND=ALTEND/1000.
   ALTSTRT=ALTSTRT/1000.
   IF (IX.NE.ITIME) GO TO 40
   XMAX=XMAX/100.
   XMIN=XMIN/100.
   40 CONTINUE
   IF (IY.NE.ITIME) GO TO 41
   YMAX=YMAX/100.
   YMIN=YMIN/100.
   41 CONTINUE
   IF (IX.NE.IALT) GO TO 50
   XMAX=XMAX/1000.
   XMIN=XMIN/1000.
   50 CONTINUE
   IF (IY.NE.IALT) GO TO 51
   YMAX=YMAX/1000.
   YMIN=YMIN/1000.
   51 CONTINUE

C PLOT GRID, AXES, LABELS
BCDX(1)=ALPHA(IX)
BCDX(2)=UNITS(IX)
BCDY(1)=ALPHA(IY)
BCDY(2)=UNITS(IY)
XSF=(XMAX-XMIN)/XL
YSF=(YMAX-YMIN)/YL
XFR=2.*XL
YFR=0.
IF (IGRID.EQ.0) GOTO 901
NOX=XL
NOY=YL
CALL GRID(0., 0., XL, YL, NOX, NOY)
GOTO 902
901 CONTINUE
902 CONTINUE
   IF (NTAG.NE.0) GOTO 903
   IF (ITPLT.EQ.1) GOTO 9031
   CALL GRID(0., 0., XL, YL, 1, 1)
   CALL AXES(0., 0., 0., XL, XMIN, XSF, 1., 0., BCDX, 2., -20)
   CALL AXES(0., 0., 90., YL, YMIN, YSF, 1., 0., 1H, 2., 1)
   CALL AXES(XL, 0., 90., YL, YMIN, YSF, 1., 0., 1H, 0., -1)
GOTO932

931 CALL AXES(0.,-.5,0.,XL,XMIN,XSF,1.,0.,BCDX,.2,-20)
CALL AXES(0.,0.,90.,YL,YMIN,YSF,1.,0.,1H ,2,1)
YO0=YL/2.
CALL CALPLT(0.,YO0,3)
CALL CALPLT(XL,YO0,2)

932 CALL CHARWH(W1,ZA,2B,.15,TITLES,9)
YI=-1.5
XI=(XL-W1)/2.0
CALL CHARACT(X1,Y1,.15,TITLES,0.,9)

903 CALL CHARWH(W1,ZA,2B,.15,BCDY,20)
XI=-(.5+W1)
YI=YL+.5-.25*NTAG
CALL CHARACT(X1,Y1,.15,BCDY,0.,20)

WRITE (3) BCDX,BCDY,TITLES(1)
NTCT=0

C C C C
C C C C
C........... READ DATA FROM TAPE1
C C

IF(IDMP.EQ.1)GOTO200

C CURVE FIT UPPER-END ALT > AFIT
IF(ALTSTRT*1000..LE.AFIT)GOTO200
NFT=0
600 READ(1) (DAT(I),I=1,24)
IF(EOF(1))620,610
610 IF(DAT(3)/1000..GT.ALTSTRT)GOTO600
IF(DAT(3).LT.AFIT)GOTO620

C BLANK RECORD INDICATOR
IF(DAT(23).EQ.-9999.)GOTO600
IF(DAT(24).EQ.-9999.)GOTO600

C MICRO-G RESOLUTION FILTER
IF(DAT(23).GT.-9.81E-06)GOTO600
IF(DAT(24).GT.-9.81E-06)GOTO600

C INPUT ADDITIONAL BIASES
AX=DAT(23)+XBIAS*9.81/1.0E06
AZ=DAT(24)+ZBIAS*9.81/1.0E06
IF(AZ.GE.0.)GOTO600
IF(AX.GE.0.)GOTO600

C NFT=NFT+1
WWX(NFT)=(-AX)/9.81*1.0E06
WWZ(NFT)=(-AZ)/9.81*1.0E06
XWK(NFT)=DAT(3)
YWK(NFT,1)=ALOG10(-AX)
YWK(NFT,2)=ALOG10(-AZ)
GOTO600

C

620 BAC(3,1)=0.
BAC(4,1)=0.
BAC(3,2)=0.
BAC(4,2)=0.
NLSQ=NFIT+1
CALL LSQPOL(500,NFT,XWK,2,YWK,WWZ,4,NLSQ,RWK,SWK,AWK,BAC,WK,IERR)
CALL LSQPOL(500,NFT,XWK,1,YWK,WWX,4,NLSQ,RWK,SWK,AWK,BAC,WK,IERR)
REWIND 1
READ(1)

C

C

200 READ(1) (DAT(I),I=1,24)
IF (EOF(1)) 820, 205
C
C SEARCH DATA
205 CONTINUE
C
C INTERPOLATE TIME AT ALT = 190KM
 IF (NTCT.EQ.0) GOTO206
 IF (DAT(3).LT.190000) GOTO206
 T190= (DAT(1)-TOLD)*(190000.-DAT(3))/(DAT(3)-AOLD)+DAT(1)
 C
 C 206 NTCT=NTCT+1
 AOLD=DAT(3)
 TOLD=DAT(1)
 C
 C 208 IF (DAT(IALT)/1000..GT.ALTSTRT) GOTO200
 IF (DAT(IALT)/1000..LT.ALTEND) GOTO820
 C
 IF (IDMP.EQ.1) GOTO209
 C
 BLANK RECORD INDICATOR
 IF (DAT(23).EQ.-9999.) GOTO200
 IF (DAT(24).EQ.-9999.) GOTO200
 C
 MICRO-G RESOLUTION FILTER
 IF (DAT(24).GT.-9.81E-06) GOTO200
 IF (DAT(23).GT.-9.81E-06) GOTO200
 C
 LEAVE GAP IN DATA
209 IF (DAT(3).GT.GAPEND.AND.DAT(3).LT.GAPSTRT) GOTO200
 C
 C FLIGHT DATA ON TAPE
 C
 DAT(4)=DAT(1)-T190
 TIME=DAT(1)
 VEL=DAT(2)
 ALT=DAT(3)
 PLAT=DAT(4)
 PLONG=DAT(5)
 ALPH=DAT(6)
 AELE=DAT(14)
 ABFL=DAT(15)
 WGT=DAT(16)
 C
 RL=DAT(18)
 TIL=DAT(19)
 PIL=DAT(20)
 WML=DAT(21)
 C
 62-STANDARD ATM. ALTITUDE MODEL
 AALT=ALT/3.048
 CALL AT62(AALT,ANS)
 R62=ANS(1)
 P162=ANS(2)
 T162=ANS(3)
 AMACH62=VEL/3.048/ANS(4)
 WM62=8314.34*10.7639*T162*R62/P162
 R62=R62*515.3788
 P162=P162*47.88026
 C
 76-STANDARD ATM. ALTITUDE MODEL
 AALT=ALT/3.048
 CALL AT76(AALT,ANS)
 R76=ANS(1)
 P176=ANS(2)
 T176=ANS(3)
 AMACH76=VEL/3.048/ANS(4)
 WM76=8314.34*10.7639*T176*R76/P176
R76=R76*515.3788
PI76=PI76*47.88026

PREPROCESS ACCELEROMETRY DATA
AX=DAT(23)+XBIAS*9.81/1.0E06
AZ=DAT(24)+ZBIAS*9.81/1.0E06
IF (IDMP.EQ.1) GOTO 2010
IF (ALT.GT.AFIT) AX=-10.**(BAC(1,1)+BAC(2,1)*ALT+BAC(3,1)*ALT**ALT+BAC(4,1)*ALT**ALT)
IF (ALT.GT.AFIT) AZ=-10.**(BAC(1,2)+BAC(2,2)*ALT+BAC(3,2)*ALT**ALT+BAC(4,2)*ALT**ALT)
IF (AZ.GE.0.) GOTO 200
IF (AX.GE.0.) GOTO 200

ACCELEROMETRY DATA IN MICRO-G S
210 IF (AZ.GT.-9.81E-06) AZ=-9.81E-06
IF (AX.GT.-9.81E-06) AX=-9.81E-06
ACCN=-AZ*1.0E06/32.1747/.3048
ACCA=-AX*1.0E06/32.1747/.3048
ACCN=ALOG10 (ACCN)
ACCA=ALOG10 (ACCA)

FORCE RATIOS
FNOA=AZ/AX
ALPHAR=ALPH*PI
SINA=SIN(ALPHAR)
COSA=COS(ALPHAR)
TANA=SINA/COSA
FLOD=(AZ/AX-TANA)/(1. + TANA*AZ/AX)

DATA BOOK PROFILES
XKN62=10.**AKN*(R62/16.01846)**BKN
XKN62=ALOG10 (XKN62)
ZDB=0.
IF (XKN62.LT.-2.99) GOTO 335
ZDB=1.
IF (XKN62.GT..99999) GOTO 335
ZDB=SIN(1.1781+.3927*XKN62)**2
335 CONTINUE
ATLT=AALT/1000.
HORIZ. PATH DIST
RADE=6370.
PH=(90.-PLAT)*PI
TH=PLONG*PI
X2=RADE*SIN(PH)*COS(TH)
Y2=RADE*SIN(PH)*SIN(TH)
Z2=RADE*COS(PH)
IF (L.EQ.0) GOTO 331
HDIST=HDIST+SQRT((X2-X1)**2+(Y2-Y1)**2+(Z2-Z1)**2)
GOTO 332
331 HDIST=0.
332 X1=X2
Y1=Y2
Z1=Z2

TW FROM STS-3,STS-5 DFI (TW)
216  TW=3172.26.2*ALT/1000.
     IF (ALT.LT.75000.) TW=1207.
     IF (ALT.GT.110000.) TW=290.

C  MEAN MOL WGT AND FRESTRM TEMP USED IN SCALING PARA.
C
C  WWML=WM
C  WWML=R62*(8314.34)*TI62/PI62
C  WWML=28.96
C  IF (ALT.LT.88000.) WWML=28.96-(ALT-88000.)/10000.
C     TI=TI62
C     IF (IATM.EQ.1) GOTO218
C     TI=TIL
C  WWML=RL*(8314.34)*TIL/PIL
C     WWML=WM76
C
218  CONTINUE
C
C  DENSITY INDEP. SCALING PARAMETER VARIABLES
C
XLR=32.77
AP=390.83*SINA
AW=1200.
SA=12.058*SQRAT(SINA)
SRAT=VEL/SQRAT(2.0*8314.34*TI/WWML)
AMACH=SRAT*SQRAT(2.0/1.4)
VISC=(TI/273.1)**1.5*6.584E-02*.0672/(TI+110.6)
RENO=(1./.016.01846)*1290.3/12.*VEL/.3048/VISC
TPR=(0.468+.0.532*TW/TI+.0195*AMACH*AMACH/5.)*TI
CPR=SQRT(TPR/TI)*(TI+122.1/10.)**(-5./TI)/
     (TPR+122.1/10.**(-5./TER))
XKNXR=WWML*2.8E-09/12.058*QFAC
OMEG=.63
AK=1.4
AM2=AMACH*AMACH
TO=TI*(1.+(AK-1.)/2.*AM2)
C
C  NORMAL SHOCK RELATIONS
TSOT=(2.*AK*AM2-AK+1.)*((AK-1.)*AM2+2.)
     /((AK+1.)*(AK+1.)*AM2)
RSOR=(AK+1.)*AM2/((AK-1.)*AM2+2.)
AMSM=SQRT(((AK-1.)+2./AM2)/(2.*AK*AM2-AK+1.))
TAS=TSOT*TI
VISC=(TAS/273.1)**1.5*6.584E-02*.0672/(TAS+110.6)
RENORS=RENO*AMSM*VISC/VISC*SQRT(TSOT*RSOR)
XKNXRS=XKNXR*AMSM*RENO/RENORS
C
C  CALCULATE FM AND CONT. COEFS.
TWOTI=TW/TI
 CALL CXC2DB(ALPH,ABFL,AELE,CZC,DELZ,CXC,DELX,CZP,
     .SRAT,TWOTI,KHRO1,CXELFM,CZELFM,CXBFFM,CZBFFM)
 IF (KHRO1.GT.4.OR.KHRO1.LT.1) GOTO 504
C
C     CXC = CX-CONT.  CXC+DELX = CX-FM.
C     CXC = CZ-CONT.  CZC+DELZ = CZ-FM.
C
C   FLORDM=(CZC+DELZ)/(CXC+DELX)
C   FLORDM=-(FLORDM-TANA)/(1.+TANA*FLORDM)
C   FLORDC=(CZC/CXC-TANA)/(1.+TANA*CZC/CXC)
C   FLORDB=(FLORD-FLORDC)/(FLORDM-FLORDC)
C
C  BYPASS AERO CALCULATIONS
IF (IDMP.EQ.1) GOTO257
IF (IX.EQ.3.AND.IY.EQ.26) GOTO257
IF (IX.EQ.3.AND.IY.EQ.27) GOTO257
C
C  BYPASS AERO CALCULATIONS
IF (IX.EQ.3.AND.IY.EQ.28) GOTO257
C
C  BYPASS AERO CALCULATIONS
IF (IX.EQ.3.AND.IY.EQ.29) GOTO257
IF (IX.EQ.1.40.AND.IY.EQ.3) GOTO257

155
IF (IX.EQ.49 .AND. IY.EQ.3) GOT0257
IF (IX.EQ.50 .AND. IY.EQ.3) GOT0257
IF (IX.EQ.51 .AND. IY.EQ.3) GOT0257

C COMPONENT DERIVED DENSITY USING CN-DATA BOOK
CNST=2.*(VEL*VEL*SS6/WGT)
RGUESS=CNST*AZ/(DELZ/2.+CZC)
RHO1=RGUESS
NCON=0

C WRITE (6,*) CNST, RGUESS, RHO1
C CONVERGING NEWTON ROOT SOLVER
412 ZETAE=1.
ZETEP=0.
XKU=XKNR/RHO1
XKP=-XKU/RHO1
XKU=ALOG10(XKU)
XKU1=XKU
C IF (XKU.GT.1.) GOT0413
C ZETAE=0.
C IF (XKU.LT.-3.) GOTO413
C ZETAE=SIN(1.1781+.3927*XKU)**2
C ZETEP=2.*SIN(1.1781+.3927*XKU)*COS(1.1781+.3927*XKU)
C .*.3927*.4343*XKP*10.**(-XKU)
IF (XKU.GE.BIZE) GOT0413
ZETAE=EXP(-AIZE*(BIZE-XKU)**CIZE)
ZETEP=ZETAE*AIZE*CIZE*(BIZE-XKU)**(CIZE-1.)/10.**XKU*XKP*.4343
413 CONTINUE
FUNC=RHO1*(DELZ*ZETAE+CZC)-AZ*CNST
FNCP=DELZ*(ZETAE+RHO1*ZETEP)+CZC
DELT=(-FUNC)/FNCP
RHOI=RHOI+DELT
NCON=NCON+1
C WRITE (6,*) FUNC, FNCP, DELT, RHO1
C WRITE (6,*) CXC, CZC, DELX, DELZ
IF (ABS(DELT).LE.(.00001*RGUESS)) GOTO414
IF (NCON.LE.11) GOTO412
414 CONTINUE

C AERO CALCULATIONS
CAP=(-AX)*CNST/RHO1
CABP=(-CAP-CXC)/DELX
C SCALING PARAMETERS
XKNR=ALOG10(XKNR/RHO1)
VBAR=AMACH*SQRT(CPR/(RHO1*RENOR))
C AERO MODELS
CNBI=1.
IF (XKNR.LT.B1ZE) CNBI=EXP(-A1ZE*(B1ZE-XKNR)**C1ZE)
CAB1=1.
IF (XKNR.LT.B2ZE) CAXB=EXP(-A2ZE*(B2ZE-XKNR)**C2ZE)
CA1=-CXC-DELX*CAB1
CN1=-CZC-DELZ*CNBI
CD1=CN1*SINA+CA1*COSA
CDB1=-CD1-CZC*SINA-CXC*COSA
FLODI=(CN1/CA1-TANA)/(1.+TANA*CN1/CA1)

C COMPONENT DERIVED DENSITY USING CA-FLIGHT DER.
RHO2=RGUESS
NCON=0
CONVERGING NEWTON ROOT SOLVER

417  XKU=XKNNXR/RHO2
    IF (XKU.LE.0) XKU=0.00001
    XKPR=-XKU/RHO2
    ZETAE=1.
    ZETEP=0.
    XKU=ALOGI0(XKU)
    XKU2=XKU
    IF (XKU.GE.B2ZE) GOTO418
    ZETAE=EXP(-A2ZE*(B2ZE-XKU)**C2ZE)
    ZETEP=ZETAE*A2ZE*C2ZE*(B2ZE-XKU)**(C2ZE-1.)/10.**XKU*XKP*.4343

418  CONTINUE
    FUNC=RHO2*(DELX*ZETAE+CXC)-AX*CNST
    FNCP=DELX*(ZETAE+RHO2*ZETEP)+CXC
    DELT=(-FUNC)/FNCP
    RHO2=RHO2+DELT
    WRITE (6,*) RHO2,FUNC,FNCP,DELT
    WRITE (6,*) AMACH, XKNNXR, RENOR,
    NCON=NCON+1
    IF (ABS(DELT).LE.(.00001*RGUESS)) GOTO416
    IF (NCON.LE.11) GOTO417

416  CONTINUE

C AERO CALCULATIONS
C
C WRITE (6,*) XKNNXR, RHO2, AMACH, RENOR, CPR
C
C SCALING PARAMETERS
XKN2=ALOGI0(XKNNXR/RHO2)
C
WRITE (6,*) VKNN2
C
VBAR2=AMACH*SQRT(CPR/(RHO2*RENOR))

C AERO MODELS
C
CAB2=1.
IF (XKNN2.LT.B2ZE) CAB2=EXP(-A2ZE*(B2ZE-XKNN2)**C2ZE)
CNB2=1.
IF (XKNN2.LT.B1ZE) CNB2=EXP(-A1ZE*(B1ZE-XKNN2)**C1ZE)
CA2=CXC-DELX*CAB2
CN2=CXC-DELZ*CNB2
CD2=CN2*SINA+CA2*COSA
CDB2=-CD2-C2C*SINA-CXC*COSA
CDF2=CN2/CA2-TANA/(1.+TANA*CN2/CA2)
IF (ALT.GE.AFIT) FLOD=FLOD2
ACCNP=ALOGI0((CNB2*DELZ+CZC)/(-CNST)*RHO2*I.0E06/9.81)

C COMPOSITE DENSITY
IF (ALT.GT.ARHO) GOTO419
RHO=RHO1
CDB=CDB1
VBAR=VBAR1
XKN=XKN1
PKN=XKNNXR/RHO1
GOTO440

419  RHO=RHO2
CDB=CDB2
VBAR=VBAR2
XKN=XKN2
PKN=XKNNXR/RHO2

440  CONTINUE
C
C REP=RENOX*RHO*SA/XLR
TPP=.2*TO+.5*TW
PN=SQRT(REP*(T1/TPP)**OMEG)
BMMN=XKNNXR/RHO*RSOR*12.058/XLR
VISCR=(TPR/273.1)**1.5*6.584E-02*.0672/(TPR+110.6)
C257

```c
VISCPP = (TPP/273.1)**1.5*6.584E-02*(TI+122.1*10.**(-5./TI))
      .*(TPP+122.1*10.**(-5./TPP))
C12 = SQRT(TPP/TI)*(TI+122.1*10.**(-5./TI))
      .*(TPP+122.1*10.**(-5./TPP))
C13 = SQRT(TAS/TI)*(TI+122.1*10.**(-5./TI))
      .*(TAS+122.1*10.**(-5./TAS))
C21 = VISCPP/VISC*TI/TPR
C22 = VISCPP/VISC*TI/TPP
C23 = VISCPP/VISC*TI/TAS
B11 = AMACH*SQRT(CPR/REP)
B12 = AMACH*SQRT(C12/REP)
B13 = AMACH*SQRT(C13/REP)
B21 = AMACH*SQRT(C21/REP)
B22 = AMACH*SQRT(C22/REP)
B23 = AMACH*SQRT(C23/REP)
B31 = AMACH/SQRT(REP*(TI/TPR)**(OMEG-1.))
B32 = AMACH/SQRT(REP*(TI/TPP)**(OMEG-1.))
B33 = AMACH/SQRT(REP*(TI/TAS)**(OMEG-1.))

C AERO COEFFICIENTS USING DATA BOOK
CNDB = -CZC-DELZ*CNBDB
CADB = -CZDB-(DELX+CXC-CZDB)*CNBDB
CABDB = -(CDB-CX)/DELX
IF(VBAR.LT..08) CABDB = -1.5285*(VBAR-.01)/DELX
IF((XKN.LT.-3.)) CADB = -CXC
IF((XKN.LT.-3.)) CABDB = -1.5285*(VBAR-.01)/DELX
CADD = -CADD-CZC*DELZ-CXC*DELX
CDBD = CNDB* DELX + CADB* COSA
CDBDB = -CDBD-CZC*DELX*COSA
CDBDB = CDBDB/(DELZ*DELX*DELX*COSA)

C LOAD OUTPUT
DAT(30) = CDB
DAT(31) = CXELFM
DAT(32) = OZELFM
DAT(33) = ALOG(0. PN)
DAT(34) = CAB1
DAT(35) = XKN2
DAT(59) = XKN1
DAT(60) = XKN1
DAT(38) = XKN2
DAT(42) = CAP
DAT(46) = CA2
DAT(47) = RHO/R76
DAT(48) = RHO
DAT(56) = CAB2
DAT(57) = CN2
DAT(58) = CN2

257 DAT(1ITIME) = DAT(1ITIME)
DAT(IALT) = DAT(IALT)/1000.
DAT(25) = HDIST
DAT(36) = TW.
DAT(33) = AMACH
DAT(39) = CN2
IF(RL.GT.0.0) DAT(39) = ALOG(LR)
DAT(41) = RHO1/RHO2
DAT(43) = RHO/R76
```

158
DAT(44) = 6.5 + ABFL/10.
DAT(45) = 1.0 + (ALPH - 40.)/5.
DAT(53) = CXBFFM
DAT(54) = CZBFFM
DAT(55) = SQRT(PI*TW/TIL)/(2.*SRAT)
DAT(63) = RHO1/R76
DAT(26) = ACCN
DAT(27) = ACCA
DAT(28) = FLOD
DAT(29) = FNOA
DAT(40) = CNBI
DAT(50) = CN1/CA1
DAT(49) = CN2/CA2
DAT(51) = FLODI
DAT(52) = WM76

L = L + 1

PLOT DAT(IX), DAT(IY)
WRITE (3) DAT(IX), DAT(IY)
IF (DAT(IY) .GT. YMAX) DAT(IY) = YMAX
IF (DAT(IY) .LT. YMIN) DAT(IY) = YMIN
IF (DAT(IX) .GT. XMAX) DAT(IX) = XMAX
IF (DAT(IX) .LT. XMIN) DAT(IX) = XMIN
X = (DAT(IX) - XMIN)/XSF
Y = (DAT(IY) - YMIN)/YSF
IF (ISYMB.EQ. I) GOTO 10
IF (L.EQ. I) CALL CALPLT(X, Y, 3)
CALL CALPLT(X, Y, 2)
GOTO 15
CALL POINT(X, Y)
CONTINUE

WRITE SELECTED DATA TO A FORMATTED OUTPUT TAPE
IF (JSTOP.EQ. I) GOTO 200
WRITE (4, 777) DAT(1), DAT(3), DAT(23), DAT(24), DAT(29)
WRITE (4, 778) DAT(4), DAT(5), AZ, AX, DAT(48), PKN
WRITE (4, 780) DAT(1), DAT(3), DAT(2), DAT(6), DAT(15), DAT(14),
WRITE (6, 669) UNITS(1), UNITS(3), UNITS(2), UNITS(6), UNITS(15),
WRITE (6, 668) ALPH(1), ALPH(3), ALPH(2), ALPH(6), ALPH(15),
ALPH(5), ALPH(47), ALPH(5), ALPH(24), ALPH(23),
ALPHA(14), ALPH(48),
FORMAT (11, 2X, A10)
FORMAT (11, 2X, A10/
FORMAT (5, 1X, E12.5))
FORMAT (1X, 6E12.6, 56X)
FORMAT (11E12.5)
FORMAT (1X, 11E12.6)
IF (L.LT. 2500) GOTO 200
CONTINUE

IF (L.LT. 2500) GOTO 200
CONTINUE
JSTOP = 1

159
WRITE (6, 1012) L
WRITE (3) -8888., -8888.
NTAG = NTAG + 1
L = 0
REWIND 1
IF (ITAG.EQ.1) GOTO 5
NTAG = 0
K = K + 1
CALL NFRAME (XFR, YFR)
CALL CALPLT (5., 3., -3)
GO TO 5

C** END OF PROGRAM, SUCCESSFUL PLOTTING DONE
C

260 CONTINUE
CALL NFRAME (XFR, YFR)
K = K - 1
WRITE (6, 265) K
265 FORMAT (1H1, I3, * FRAMES PLOTTED.*/ *HAVE A NICE DAY.*)
CALL CALPLT (0., 0., 999)
2000 FORMAT (//50X *PLOT NO. *I3//)
1004 FORMAT (//50X *INDEPENDENT VARIABLE SELECTED *A10,* BETWEEN *F10.3,* AND *F10.3)  
1005 FORMAT (//50X *DEPENDENT VARIABLE SELECTED *A10,* BETWEEN *F10.3,* AND *F10.3)  
1012 FORMAT (//50X, I6,* POINTS PLOTTED*)
2017 FORMAT (//50X *SEARCH AEROBET FILE BETWEEN *F10.3,* AND *F10.3,* METERS ALTITUDE*)
STOP

C** SAME VARIABLE ON X- AND Y-AXES
C

650 WRITE (6, 655) XAXIS
655 FORMAT (//IX *PLOTTING *,A10,* AGAINST ITSELF IS NOT ALLOWED.*)
GO TO 820

C

4009 CONTINUE
WRITE (6, 4020)
4020 FORMAT (//IX, *X-AXIS ALPHA UNDEFINED---PLOT TERMINATED*)
GO TO 820

C

4010 CONTINUE
WRITE (6, 4021)
4021 FORMAT (//IX, *Y-AXIS ALPHA UNDEFINED---PLOT TERMINATED*)
GO TO 820

C

683 CONTINUE
WRITE (6, 684) XSF, YSF
684 FORMAT (//IX, *PLOT INHIBITED-SCALE FACTOR INDETERMINATE XSF  
.*F10.3,* YSF *F10.3/)  
GO TO 820

C

386 CONTINUE
WRITE (6, 387)
387 FORMAT (//IX, *PLOT INHIBITED-Y1 SCALE FACTOR 0.*)
GO TO 820

C** EOF WHILE TRYING TO READ HEADER ON TAPE1
C

500 WRITE (6, 505)

160
505 FORMAT (1X*E-O-F ENCOUNTERED WHILE TRYING TO READ HEADER ON TAPE 1
*, PROGRAM HALTED.*)
504 PRINT*,"ERROR"
STOP
END
SUBROUTINE CXCZDB(AL, ABFL, AELE, AZZ, BZZ, CZZ, DZZ, CZF, SR, TWT, KHRO,
* CXELFM, CZELFM, CXBFMM, CZBFMM)
PI=4.*ATAN(1.)
AR=AL*PI/180.
NDER=0
NCS=0
KHRO=4
PDIF=SQR(T(PITWT)/(2.*SR))
PDIF=SQR(T(PI*.25)/(2.*9.)

C IF (KHRO.GT.1) GOTO201
IF (KHRO.GT.2) GOTO301
IF (KHRO.GT.3) GOTO401
CX=-.05925-.011+.004
CZ=-1.205-.004
CXFM=-I.6095
CZFM=-I.3725

C IF(NDER.EQ.1) GOTO11
CXFM=CXFM*(.27568+.02879405*AL+.00015907143*AL*AL
-.7.333333E-06*AL*AL*AL)
CZFM=CZFM*(.104-.0038214286*AL+9.571429E-04*AL*AL
-.7.5E-06*AL*AL*AL)
CX=CMCX(.2959402+.067083253*AL+.023369244*AL*AL
+.3.58036123E-05*AL*AL*AL-.2.047015503E-07*AL*AL*AL)
CZ=CMCX(-.22316013+.0248950704*AL+.000143729041*AL*AL
+.02*(AL-40.)

C IF(NCS.EQ.1) GOTO11
CXEL=.0002-.0008*AELE-.00013*AELE*AELE
+.5.6E-04*AELE
IF(AELE.LE.0.0)CXEL=(-.0026*AELE)-.00016*AELE*AELE
IF(KHRO.EQ.1) GOTO501

201 CX=-.05270
CZ=-1.1538
CXFM=-1.6095
CZFM=-1.3725
CXFM=CXFM*(.27568+.02879405*AL+.00015907143*AL*AL
-.7.333333E-06*AL*AL*AL)
CZFM=CZFM*(.104-.0038214286*AL+9.571429E-04*AL*AL
-.7.5E-06*AL*AL*AL)
CX=CMCX(.2959402+.067083253*AL+.023369244*AL*AL
+.3.58036123E-05*AL*AL*AL-.2.047015503E-07*AL*AL*AL)
CZ=CMCX(-.22316013+.0248950704*AL+.000143729041*AL*AL
+.02*(AL-40.)

C IF(NCS.EQ.1) GOTO11
CXEL=.0002-.0008*AELE-.00013*AELE*AELE
+.5.6E-04*AELE
IF(AELE.LE.0.0)CXEL=(-.0026*AELE)-.00016*AELE*AELE
IF(KHRO.EQ.2) GOTO501

301 CX=2.36E-4*AL*AL-1.8534E-2*AL+.41976
CX=-CX
CZ=-.04345*AL+.6230
CXFM=-1.58
CZFM=-1.5

C CXFM=CXFM*(.27568+.02879405*AL+.00015907143*AL*AL
-.7.333333E-06*AL*AL*AL)
CZFM=CZFM*(.104-.0038214286*AL+9.571429E-04*AL*AL
-.7.5E-06*AL*AL*AL)
CXEL=.0002-.0008*AELE-.00013*AELE*AELE
+.5.6E-04*AELE

161
IF (AELE LE 0.0) CXEL = (-0.0026*AELE) -.00016*AELE*AELE
IF (KHRO.EQ.3) GOTO 501

401 CX = -.056
CZ = -1.115
C CXFM = CXFM*.(2.7568+.02879405*AL-.00015907143*AL*AL)
C -.2.733333E-06*AL*AL*AL)
C CZFM = CZFM*.104-.0038214286*AL+9.571429E-04*AL*AL)
C -.7.5E-06*AL*AL*AL)
C KTLMF
CXFM = -1.6095
CZFM = -1.4567

C****** DATA BOOK FMF AT >600000 FT 0 DEG SIDESLIP, AOA 0-60 DEG***
CZFM = 1.58739E-3 + (9.18422E-3*AL)+(9.66197E-4*AL*AL)+
(-7.16528E-6*AL*AL*AL)
CZFM = -CZFM
CXFM = 7.51105E-1 + (1.64864E-2*AL)+(5.92205E-4*AL*AL)+
(-1.7117E-5*AL*AL*AL)
CXFM = -CXFM

C ****************************** BASIC COEFF. VALUES (@ 40 DEG. )
C CX = 2.36E-4 AL*AL-1.8534E-2 AL+.416761
C CX = 3.5*VBAR-.0515
C CX = .04345*AL+.6230
C CXEL = .0002-.0008*AELE-.00013*AELE*AELE
C +.5*6E-04*AELE
C THIS IS AELE-CXEL FOR NEG AELE BASED UPON N/A STUDY AUG87
IF (AELE LE 0.0) CXEL = -.0009*AELE-.00005*AELE*AELE
IF (KHRO.EQ.4) GOTO 501

501 CONTINUE
C THE FOLLOWING CX IS THE AS-RECEIVED,*APRIL86*VALUE
C CX = .05925-.0011+.004
C THE FOLLOWING IS THE FAD-26 VALUE OF CX
C CX = .05270
C THIS CX IS THE TEN FLIGHT L/D, ALPHA=40 STUDY VALUE.
C CX = -.056
C THE FOLLOWING CZ IS THE AS-RECEIVED,*APRIL86*VALUE
C CZ = .0336E-4 AL+.6230
C THE FOLLOWING CZ IS BASED ON A TEN FLIGHT L/D, ALPHA =40 STUDY
C CZ = 1.1538
C THE FOLLOWING CZ IS BASED ON A TEN FLIGHT L/D, ALPHA =40 STUDY
C CZ = -.056
C THE FOLLOWING CZ IS THE FAD-26 VALUE
C CZ = 1.05
C THE FOLLOWING CZ IS BASED ON A TEN FLIGHT L/D, ALPHA =40 STUDY
C CZ = 1.115
C THIS IS THE PRE-OP DATA BOOK CXFM VALUE (AS-RECEIVED*APRIL86*)
C CXFM = 1.6095
C THIS CXFM IS BASED ON THE TEN FLIGHT L/D, ALPHA =40 STUDY.
C CXFM = 1.57
C THIS IS THE CXFM OF THE SPLIT DIFF
C CXFM = 1.58
C THIS IS THE CXFM BASED UPON THE N/A STUDY, AUG87
C CXFM = 1.50
C CXFM = 1.75
C THIS IS THE PRE-OP DATA BOOK VALUE (AS RECEIVED*APRIL86*)
C CXFM = 1.4567
C THIS IS THE CXFM BASED ON THE TEN FLIGHT L/D, ALPHA =40 STUDY.
C CXFM = 1.55
C THIS IS THE CXFM BASED ON THE SPLIT DIFF
C CXFM = 1.5
C CXFM = -3.934*SIN(AR)*(SIN(AR)+PDIF)
C IF (NDER.EQ.1) GOTO 11
C ALPHA DERIVATIVES
C CXFM = CXFM*.2.7568+.02879405*AL-.00015907143*AL*AL
C -.2.733333E-06*AL*AL*AL)
C CZFM = CZFM*.104-.0038214286*AL+9.571429E-04*AL*AL
C -.7.5E-06*AL*AL*AL)
C CX = CX*(.29594402+.067083253*AL-.0023369244*AL*AL)
THIS IS THE CAC UPDATE BASED ON FAD-26 AND CAO=.056 (APRIL87)
CX=2.36E-4*AL*AL-1.8534E-2*AL+.4176
CX=-CX
CZ=CZ*(-.22316013+.0248950704*AL+.000143729041*AL*AL)
+C+.02*(AL-40.)
THIS IS THE CX ALPHA BASED UPON THE N/A STUDY AUG 87
C67 CX=2.36E-4*AL*AL-1.8534E-2*AL+.416761
CX=-CX
CZ=CZ*(-.22316013+.0248950704*AL+.000143729041*AL*AL)
+C+.02*(AL-40.)
THIS IS THE FAD-L/D CZ VALUE (APRIL87)
CZ=-.04345*AL+.6230

IF (NCS.EQ.1) GOTO 11
CONTROL SURFACE DERIVATIVES
CXEL=-.00623119*AELE-9.7147619E-05*AELE*AELE
CXEL=-.00147435434*AELE-1.32886546E-04*AELE*AELE
+8.698039216E-07*AELE*AELE*AELE
CXEL=.0002-.0008*AELE-.00013*AELE*AELE
+5.6E-04*AELE
IF (AELE.LE.0.) CXEL=(-.0026*AELE)-.00016*AELE*AELE
CZEL=-.0057382024*AELE-1.1632619E-04*AELE*AELE
CXBF=-.0001659109-.0003626653*ABFL
-2.2857795E-05 * ABFL*ABFL
+.1024339905E-07 * ABFL*ABFL*ABFL
AER=AELE*PI/180.
ABR=ABFL*PI/180.
AR=AL*PI/180.
G=AER+AR
CFG=SIN(G)*COS(G)
CFA=SIN(AR)*COS(AR)
CPG=SIN(G)*SIN(G)*(1.+G*2./PI)
CPA=SIN(AR)*SIN(AR)*(1.+AR*2./PI)
CPG=SIN(G)*(SIN(G)+PDIF)
CPA=SIN(AR)*(SIN(AR)+PDIF)
CXELFM=-413.14/2690.* (COS (AER)* (CFG-CFA)+SIN (AER)* (CPG-CPA))
CZELFM=-413.14/2690.* (COS (AER)* (CPG-CPA)-SIN (AER)* (CFG-CFA))
G=ABR+AR
CZBF=.0001086855-.0020996556*ABFL
-3.13694583E-05 * ABFL*ABFL
+.1353946759E-06 * ABFL*ABFL*ABFL
CXEL=0.
CXBF=0.
CZEL=0.
CZBF=0.
CXBFMM=0.
CZELFM=0.
CZBFMM=0.
CXELFM=0.
CX= CX+CXBF+CXEL
CZ=CZ+CZBF+CZEL
CZFM= CZFM+CZBFMM+CZELFM
CZFM=CZFM
11 CONTINUE
TRANSITION BRIDGING FORMULA
AZZ=CZ
SUBROUTINE BETA (UI, RI, WM, TI, TW, BE)
BE = SQRT (8314.34 * TW / WM) / UI + 1.0
BE = 5.0016E-07 / SQRT (BE * TI * RI * RI / (WM * WM * WM))
RETURN
END
SUBROUTINE DATM62 (RS, HH, P, T, A)
DIMENSION ANS(4)
HO = HH
HOO = HH - 1000.
N = 0
DELT = HO - HOO
H = HOO
CALL AT62 (H, ANS)
R = ANS(1)
FX = RS - R
H = HO
1 FOLD = FX
CALL AT62 (H, ANS)
R = ANS(1)
FX = RS - R
DELT = (-FX) / ((FX - FOLD) / DELT)
H = H + DELT
N = N + 1
IF (N .GE. 11) GOTO 2
IF (ABS (DELT) .GT. (.0001 * HH)) GOTO 1
2 P = ANS(2)
T = ANS(3)
A = ANS(4)
RETURN
END
References


Table 1. STS HiRAP Missions

The number used in this report to identify each mission is given with instrument, orbiter name, and entry date.

<table>
<thead>
<tr>
<th>Mission</th>
<th>System data file number</th>
<th>Instrument</th>
<th>Orbiter</th>
<th>Entry date</th>
</tr>
</thead>
<tbody>
<tr>
<td>STS-06</td>
<td>6</td>
<td>S/N 001</td>
<td>Challenger</td>
<td>4/9/83</td>
</tr>
<tr>
<td>STS-07</td>
<td>7</td>
<td>S/N 001</td>
<td>Challenger</td>
<td>6/24/83</td>
</tr>
<tr>
<td>STS-08</td>
<td>8</td>
<td>S/N 001</td>
<td>Challenger</td>
<td>9/5/83</td>
</tr>
<tr>
<td>STS-09</td>
<td>9</td>
<td>S/N 002</td>
<td>Columbia</td>
<td>12/8/83</td>
</tr>
<tr>
<td>STS-41B</td>
<td>11</td>
<td>S/N 001</td>
<td>Challenger</td>
<td>2/11/84</td>
</tr>
<tr>
<td>STS-41C</td>
<td>13</td>
<td>S/N 001</td>
<td>Challenger</td>
<td>4/13/84</td>
</tr>
<tr>
<td>STS-51B</td>
<td>24</td>
<td>S/N 001</td>
<td>Challenger</td>
<td>5/6/85</td>
</tr>
<tr>
<td>STS-51F</td>
<td>26</td>
<td>S/N 001</td>
<td>Challenger</td>
<td>8/6/85</td>
</tr>
<tr>
<td>STS-61A</td>
<td>30</td>
<td>S/N 002</td>
<td>Challenger</td>
<td>11/6/85</td>
</tr>
<tr>
<td>STS-61C</td>
<td>32</td>
<td>S/N 001</td>
<td>Columbia</td>
<td>1/18/86</td>
</tr>
</tbody>
</table>

Table 2. HiRAP Ground Calibration Scale Factors

<table>
<thead>
<tr>
<th>Instrument and flight</th>
<th>Scale factor, $V/\omega$, for acceleration along—</th>
<th>X-axis</th>
<th>Y-axis</th>
<th>Z-axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>S/N 001 before recalibration (STS-06, 07, 08, 41B, 41C, and 51B)</td>
<td>-1.247237 × 10^{-3}</td>
<td>1.253821 × 10^{-3}</td>
<td>1.26810 × 10^{-3}</td>
<td></td>
</tr>
<tr>
<td>S/N 001 after recalibration (STS-61C)</td>
<td>-1.24720 × 10^{-3}</td>
<td>1.269482 × 10^{-3}</td>
<td>-1.256565 × 10^{-3}</td>
<td></td>
</tr>
<tr>
<td>S/N 002 before recalibration (STS-09)</td>
<td>-1.24857 × 10^{-3}</td>
<td>1.269482 × 10^{-3}</td>
<td>-1.256565 × 10^{-3}</td>
<td></td>
</tr>
<tr>
<td>S/N 002 after recalibration (STS-51F, 61A)</td>
<td>-1.250035 × 10^{-3}</td>
<td>1.271533 × 10^{-3}</td>
<td>-1.253671 × 10^{-3}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-(1.283124 × 10^{-6})V_T^2</td>
<td>-(1.344372 × 10^{-6})V_T</td>
<td>-(2.584656 × 10^{-6})V_T</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+(0.1437924 × 10^{-6})V_T^3</td>
<td>+(0.04102193 × 10^{-3})V_T^2</td>
<td>+(1.044413 × 10^{-6})V_T^2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>-(0.1370558 × 10^{-6})V_T^3</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
Figure 1. Arrangement of HiRAP accelerometer triad in the Space Shuttle orbiter vehicle.
Figure 2. Shuttle reentry trajectories overlaid on Earth globe. Mission number and flight date (month/year) shown for each flight.
Figure 3. Acceleration counts and sensor temperature versus time for STS-06.
(c) Y-axis acceleration counts.

(d) X-, Y-, and Z-axis temperatures.

Figure 3. Concluded.
Figure 4. Acceleration counts and sensor temperature versus time for STS-07.
(c) Y-axis acceleration counts.

(d) X-, Y-, and Z-axis temperatures.

Figure 4. Concluded.
Figure 5. Acceleration counts and sensor temperature versus time for STS-08.
(c) Y-axis acceleration counts.

(d) X-, Y-, and Z-axis temperatures.

Figure 5. Concluded.
Figure 6. Acceleration counts, sensor temperature, and sensor temperature counts versus time for STS-09.
(c) $Y$-axis acceleration counts.

(d) $X$-, $Y$-, and $Z$-axis temperatures.

Figure 6. Continued.
(e) $X_-$, $Y_-$, and $Z$-axis temperature.

(f) $X_-$, $Y_-$, and $Z$-axis fine temperature counts.

Figure 6. Continued.
(g) X-, Y-, and Z-axis coarse temperature counts.

Figure 6. Concluded.
Figure 7. Acceleration counts and sensor temperature versus time for STS-41B.
(c) Y-axis acceleration counts.

(d) X-, Y-, and Z-axis temperatures.

Figure 7. Concluded.
Figure 8. Acceleration counts and sensor temperature versus time for STS-41C.

(a) X-axis acceleration counts.

(b) Z-axis acceleration counts.
Figure 8. Concluded.

(c) Y-axis acceleration counts.

(d) X-, Y-, and Z-axis temperatures.
Figure 9. Acceleration counts and sensor temperature versus time for STS-51B.

(a) X-axis acceleration counts.

(b) Z-axis acceleration counts.
(c) $Y$-axis acceleration counts.

(d) $X$, $Y$, and $Z$-axis temperatures.

Figure 9. Concluded.
Figure 10. Acceleration counts and sensor temperature versus time for STS-51F.
(c) $Y$-axis acceleration counts.

(d) $X$, $Y$, and $Z$-axis temperatures.

Figure 10. Concluded.
Figure 11. Acceleration counts and sensor temperature versus time for STS-61A.
(c) Y-axis acceleration counts.

(d) X-, Y-, and Z-axis temperatures.

Figure 11. Concluded.
Figure 12. Acceleration counts and sensor temperature versus time for STS-61C.
(c) Y-axis acceleration counts.

(d) X-, Y-, and Z-axis temperatures.

Figure 12. Concluded.
Figure 13. Time and altitude histories of orbiter state vector subset data for STS-06.

(a) Altitude versus time.
(b) Angle of attack versus altitude.

(c) Angle of attack versus time.

Figure 13. Continued.
(d) Body flap deflection versus altitude.

(e) Body flap deflection versus time.

Figure 13. Continued.
Figure 13. Concluded.
(a) Altitude versus time.

Figure 14. Time and altitude histories of orbiter state vector data subset for STS-07.
(b) Angle of attack versus altitude.

(c) Angle of attack versus time.

Figure 14. Continued.
(d) Body flap deflection versus altitude.

(e) Body flap deflection versus time.

Figure 14. Continued.
(f) Elevon deflection versus altitude.

(g) Elevon deflection versus time.

Figure 14. Concluded.
(a) Altitude versus time.

Figure 15. Time and altitude histories of orbiter state vector data subset for STS-08.
Figure 15. Continued.

(b) Angle of attack versus altitude.

(c) Angle of attack versus time.
(d) Body flap deflection versus altitude.

(c) Body flap deflection versus time.

Figure 15. Continued.
(f) Elevon deflection versus altitude.

(g) Elevon deflection versus time.

Figure 15. Concluded.
Figure 16. Time and altitude histories of orbiter state vector data subset for STS-09.

(a) Altitude versus time.
(b) Angle of attack versus altitude.

(c) Angle of attack versus time.

Figure 16. Continued.
Figure 16. Continued.

(d) Body flap deflection versus altitude.

(e) Body flap deflection versus time.
(f) Elevon deflection versus altitude.

(g) Elevon deflection versus time.

Figure 16. Concluded.
Figure 17. Time and altitude histories of orbiter state vector data subset for STS-41B.

(a) Altitude versus time.
Figure 17. Continued.

(b) Angle of attack versus altitude.

(c) Angle of attack versus time.
(d) Body flap deflection versus altitude.

(e) Body flap deflection versus time.

Figure 17. Continued.
(f) Elevon deflection versus altitude.

(g) Elevon deflection versus time.

Figure 17. Concluded.
Figure 18. Time and altitude histories of orbiter state vector data subset for STS-41C.

(a) Altitude versus time.
(b) Angle of attack versus altitude.

(c) Angle of attack versus time.

Figure 18. Continued.
(d) Body flap deflection versus altitude.

(e) Body flap deflection versus time.

Figure 18. Continued.
(f) Elevon deflection versus altitude.

(g) Elevon deflection versus time.

Figure 18. Concluded.
Figure 19. Time and altitude histories of orbiter state vector data subset for STS-51B.

(a) Altitude versus time.
Figure 19. Continued.

(b) Angle of attack versus altitude.

(c) Angle of attack versus time.
(d) Body flap deflection versus altitude.

(e) Body flap deflection versus time.

Figure 19. Continued.
Figure 19. Concluded.
Figure 20. Time and altitude histories of orbiter state vector data subset for STS-51F.
(b) Angle of attack versus altitude.

(c) Angle of attack versus time.

Figure 20. Continued.
(d) Body flap deflection versus altitude.

(e) Body flap deflection versus time.

Figure 20. Continued.
(f) Elevon deflection versus altitude.

(g) Elevon deflection versus time.

Figure 20. Concluded.
(a) Altitude versus time.

Figure 21. Time and altitude histories of orbiter state vector data subset for STS-61A.
(b) Angle of attack versus altitude.

(c) Angle of attack versus time.

Figure 21. Continued.
(d) Body flap deflection versus altitude.

(e) Body flap deflection versus time.

Figure 21. Concluded.
(f) Elevon deflection versus altitude.

(g) Elevon deflection versus time.

Figure 21. Concluded.
(a) Altitude versus time.

Figure 22. Time and altitude histories of orbiter state vector data subset for STS-61C.
(b) Angle of attack versus altitude.

(c) Angle of attack versus time.

Figure 22. Continued.
(d) Body flap deflection versus altitude.

(c) Body flap deflection versus time.

Figure 22. Continued.
(f) Elevon deflection versus altitude.

(g) Elevon deflection versus time.

Figure 22. Concluded.
Thermal stabilization
Pitch maneuver
1st APU on
Self check
Deorbit burn
2nd, 3rd APU on
Onset of forces
Fuel dump

(a) X-axis acceleration and time-line events.

Figure 23. Section of acceleration data for STS-61C with time-line events labeled.
(b) Y-axis acceleration and time-line events.

Figure 23. Continued.
(c) Z-axis acceleration and time-line events.

Figure 23. Concluded.
Figure 24. Section of acceleration data for STS-61C with RCS thruster activity.
(c) Z-axis acceleration.

Figure 24. Concluded.
(a) X-axis acceleration.

(b) Y-axis acceleration.

(c) Z-axis acceleration.

Figure 25. Section of acceleration data for STS-61C without RCS thruster activity.
(a) X-axis acceleration.

(b) Z-axis acceleration.

Figure 26. Acceleration versus time (no RCS signal) for STS-06.
(c) Y-axis acceleration.

Figure 26. Concluded.
Figure 27. Acceleration versus time (no RCS signal) for STS-07.

(a) X-axis acceleration.

(b) Z-axis acceleration.
(c) Y-axis acceleration.

Figure 27. Concluded.
Figure 28. Acceleration versus time (no RCS signal) for STS-08.

(a) X-axis acceleration.

(b) Z-axis acceleration.
(c) Y-axis acceleration.

Figure 23. Concluded.
Figure 29. Acceleration versus time (no RCS signal) for STS-09.

(a) X-axis acceleration.

(b) Z-axis acceleration.
(c) Y-axis acceleration.

Figure 29. Concluded.
Figure 30. Acceleration versus time (no RCS signal) for STS-41B.

(a) $X$-axis acceleration.

(b) $Z$-axis acceleration.
(c) Y-axis acceleration.

Figure 30. Concluded.
Figure 31. Acceleration versus time (no RCS signal) for STS-41C.
(c) Y-axis acceleration.

Figure 31. Concluded.
Figure 32. Acceleration versus time (no RCS signal) for STS-51B.

(a) X-axis acceleration.

(b) Z-axis acceleration.
(c) Y-axis acceleration.

Figure 32. Concluded.
Figure 33. Acceleration versus time (no RCS signal) for STS-51F.

(a) X-axis acceleration.

(b) Z-axis acceleration.
Figure 33. Concluded.

(c) Y-axis acceleration.
(a) X-axis acceleration.

(b) Z-axis acceleration.

Figure 34. Acceleration versus time (no RCS signal) for STS-61A.
(c) Y-axis acceleration.

Figure 34. Concluded.
Figure 35. Acceleration versus time (no RCS signal) for STS-61C.
(c) Y-axis acceleration.

Figure 35. Concluded.
Figure 36. Acceleration versus time (after data gap filling) for STS-06.
(c) Y-axis acceleration.

Figure 36. Concluded.
Figure 37. Acceleration versus time (after data gap filling) for STS-07.
Figure 37. Concluded.

(c) $Y$-axis acceleration.
Figure 38. Acceleration versus time (after data gap filling) for STS-08.

(a) X-axis acceleration.

(b) Z-axis acceleration.
(c) Y-axis acceleration.

Figure 38. Concluded.
Figure 39. Acceleration versus time (after data gap filling) for STS-09.

(a) X-axis acceleration.

(b) Z-axis acceleration.
(c) Y-axis acceleration.

Figure 39. Concluded.
Figure 40. Acceleration versus time (after data gap filling) for STS-41B.

(a) X-axis acceleration.

(b) Z-axis acceleration.
(c) Y-axis acceleration.

Figure 40. Concluded.
Figure 41. Acceleration versus time (after data gap filling) for STS-41C.

(a) X-axis acceleration.

(b) Z-axis acceleration.
(c) Y-axis acceleration.

Figure 41. Concluded.
Figure 42. Acceleration versus time (after data gap filling) for STS-51B.

(a) X-axis acceleration.

(b) Z-axis acceleration.
(c) $Y$-axis acceleration.

Figure 42. Concluded.
Figure 43. Acceleration versus time (after data gap filling) for STS-51F.

(a) X-axis acceleration.

(b) Z-axis acceleration.
Figure 43. Concluded.
Figure 44. Acceleration versus time (after data gap filling) for STS-61A.

(a) X-axis acceleration.

(b) Z-axis acceleration.
(c) Y-axis acceleration.

Figure 44. Concluded.
Figure 45. Acceleration versus time (after data gap filling) for STS-61C.

(a) $X$-axis acceleration.

(b) $Z$-axis acceleration.
Figure 45. Concluded.

(c) Y-axis acceleration.
Figure 46. One-second averaged aerodynamic acceleration data versus time for STS-06.
(c) Y-axis acceleration.

Figure 46. Concluded.
Figure 47. One-second averaged aerodynamic acceleration data versus time for STS-07.
(c) Y-axis acceleration.

Figure 47. Concluded.
Figure 48. One-second averaged aerodynamic acceleration data versus time for STS-08.

(a) X-axis acceleration.

(b) Z-axis acceleration.
(c) Y-axis acceleration.

Figure 48. Concluded.
Figure 49. One-second averaged aerodynamic acceleration data versus time for STS-09.

(a) X-axis acceleration.

(b) Z-axis acceleration.
(c) Y-axis acceleration.

Figure 49. Concluded.
Figure 50. One-second averaged aerodynamic acceleration data versus time for STS-41B.

(a) X-axis acceleration.

(b) Z-axis acceleration.
(c) Y-axis acceleration.

Figure 50. Concluded.
Figure 51. One-second averaged aerodynamic acceleration data versus time for STS-41C.
(c) Y-axis acceleration.

Figure 51. Concluded.
Figure 52. One-second averaged aerodynamic acceleration data versus time for STS-51B.
(c) Y-axis acceleration.

Figure 52. Concluded.
Figure 53. One-second averaged aerodynamic acceleration data versus time for STS-51F.

(a) X-axis acceleration.

(b) Z-axis acceleration.
(c) $Y$-axis acceleration.

Figure 53. Concluded.
Figure 54. One-second averaged aerodynamic acceleration data versus time for STS-61A.
(c) Y-axis acceleration.

Figure 54. Concluded.
Figure 55. One-second averaged aerodynamic acceleration data versus time for STS-61C.

(a) X-axis acceleration.

(b) Z-axis acceleration.
(c) Y-axis acceleration.

Figure 55. Concluded.
Figure 56. Normalized force coefficients versus Knudsen number.
Figure 57. Axial- and normal-force coefficients versus angle of attack.
Figure 58. Incremental axial- and normal-force coefficients versus body flap deflection.
Figure 59. Incremental axial- and normal-force coefficients versus elevon deflection.
(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

(b) Profile of calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model profile.

Figure 60. Density analysis results for STS-06.
(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

(b) Profile of calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model profile.

Figure 61. Density analysis results for STS-07.
(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

(b) Profile of calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model profile.

Figure 62. Density analysis results for STS-08.
(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

(b) Profile of calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model profile.

Figure 63. Density analysis results for STS-09.
(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

(b) Profile of calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model profile.

Figure 64. Density analysis results for STS-41B.
(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

(b) Profile of calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model profile.

Figure 65. Density analysis results for STS-41C.
(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

(b) Profile of calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model profile.

Figure 66. Density analysis results for STS-51B.
(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

(b) Profile of calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model profile.

Figure 67. Density analysis results for STS-51F.
(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

(b) Profile of calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model profile.

Figure 68. Density analysis results for STS-61A.
Figure 69. Density analysis results for STS-61C.

(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

(b) Profile of calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model profile.
(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

(b) Calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model estimate.

Figure 70. Density analysis results for STS-61A simulating correction of $-1^\circ$ misalignment.
(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

(b) Profile of normal coefficient.

Figure 71. Density analysis results for STS-61A showing correlation with normal coefficient.
(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

(b) 1976 U.S. Standard Atmosphere (ref. 1) profile of molecular weight and alternate profile of molecular weight used to generate results.

Figure 72. Density analysis results for STS-61A with alternate molecular weight profile.
The High Resolution Accelerometer Package (HiRAP) Flight Experiment Summary for the First 10 Flights

Robert C. Blanchard, K. T. Larman, and M. Barrett

NASA Langley Research Center
Hampton, VA 23665-5225

National Aeronautics and Space Administration
Washington, DC 20546-0001

Blanchard: Langley Research Center, Hampton, VA; Larman and Barrett: Lockheed Engineering & Sciences Company, Hampton, VA.

Unclassified

Rarefied-flow aerodynamics; Accelerometer measurements; Upper atmosphere

Unclassified

Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. Z39.18-1992

NASA Langley 1992