IN 1

85181

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

Robert C. Blanchard,
K. T. Larman,
[illegible] M. Barrett

NASA

NASA
Reference
Publication
1267

1992

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

National Aeronautics and
Space Administration
Office of Management
Scientific and Technical
Information Program

Contents

Summary

The High Resolution Accelerometer Package (HiRAP) instrument is a triaxial, orthogonal system of gas-damped accelerometers with a resolution of $1 \times 10^{-6}g$ (1 μg). 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 μg 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 orbiter 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

V	orbiter velocity, m/sec
V_T	voltage
$\overline{X}$, $\overline{Y}$, $\overline{Z}$	distance of HiRAP accelerometers from orbiter center of gravity, m
μg	$= 1 \times 10^{-6} g$
Δ	incremental change
θ	misalignment angle, deg
ρ	atmospheric density, kg/m^3
Subscripts:	
c	corrected measurement
i	index
m	measured
model	model value
Abbreviations:	
ABET	Aerodynamic Best Estimate Trajectory
ACIP	Aerodynamic Coefficient Identification Package
APU	auxiliary power unit
GMT	Greenwich mean time
IMU	inertial measurement unit
OADB	Orbiter Aerodynamic Data Book
OEX	Orbiter Experimental
OV	orbiter vehicle
PCM	pulse code modulation
RCS	reaction control system
STS	Space Transportation System
TIF	time interface file
XBET	Extended Best Estimate Trajectory

Background

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.

The HiRAP instrument uses a set of three orthogonal, pendulous, gas-damped accelerometers, each with a resolution of 1 μg and a measurement range of approximately ± 8000 μg. The instrument weighs 1.13 kg and its size is $8.89 \times 12.70 \times 10.16$ cm. The HiRAP instrument is mounted in the wing box on the cargo bay, such that the orthogonal HiRAP axes are aligned with the OV body axes. A diagram of the HiRAP and its location in the OV are shown in figure 1. In this document, the axes used are oriented as shown in figure 1.

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.

Accelerometer Measurements

The HiRAP flight acceleration measurements are recorded on OEX flight tapes at a rate of 174 Hz for all flights except STS-61C, which is recorded at 112 Hz. The signal of each accelerometer sensor channel is an analog voltage in the range of ± 10 V.

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 Postflight 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 HSKPXXY, convert the coarse and fine temperature counts to degrees Fahrenheit, and write the temperature files TCVXXY. 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 continous 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 biases, 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 μg $(1 \times 10^{-6} g)$ 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 μg is described in a subsequent section.

Each acceleration history is checked for 10 time-line 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 μg. Figure 23 shows the temperature bias after power up for flight STS-61C to be approximately −1850, 760, and −1740 μg 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 μg. 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 μg.

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 ±200 μg. 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 μg on the X-axis and −30 μg 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 μg on the X-axis and of approximately 100 μg 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-μg 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 ±300 μg.

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 (±5000 μg).

9. Instrument saturation. When the accelerations exceed −8000 μg 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 ±8000 μg 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 μg. In the case of the primary thrusters, the maximum signal magnitude is approximately 4000 μg. 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 dampening 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 engineering 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 scale factors are applied in source code JTRATES for the X- and Z-axis accelerations and in source code YCONVXX for the Y-axis acceleration. These converted X- and Z-axis acceleration measurements are contained in files NTCXX and converted Y-axis measurements are contained in files MGWTHXX.

Flights STS-51F and STS-61A were instrumented with the modified version of HiRAP S/N 002. The instrument was modified by the application of a large positive bias to create an offset of the signal of 7000 μg. Also, as part of the instrument modifications, a procedural change was introduced, namely, to evaluate scale factor as a function of temperature monitor voltages. The relationship of scale factor to temperature monitor voltages provided by the ground calibration is used. For these cases, the scale factors are applied in source code ORBPLOTA for the X- and Z-axis accelerations and in source code YCONXX for the Y-axis acceleration. These converted X- and Z-axis acceleration measurements are contained in output files MGXX and the converted Y-axis acceleration measurements are contained in output files MGWTHXX. The scale factors for each axis and instrument are presented in table 2.

The temperature dependency of the scale factors of the modified instrument S/N 002 was evaluated to determine its effect on the acceleration measurements. For a typical change in monitor voltage over the descent period, the scale factor change (and subsequently the acceleration change) is approximately 0.25 percent. For example, at an accelerometer 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. Because of this offset, HiRAP measures linear accelerations 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 acceleration 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 locations and the flight-dependent location of the center 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 factor is used in the conversion process, files MGXX). Program JRATES reads the 1-Hz rotational rates, calculates the resulting induced accelerations, interpolates the induced accelerations to the HiRAP data rate, and subtracts the induced linear accelerations from the HiRAP measurements. The corrected accelerometer data are written to file CGXX.

The complete induced acceleration matrix is as follows:

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

where

ΔA_x induced linear acceleration along X-axis

ΔA_y induced linear acceleration along Y-axis

ΔA_z induced linear acceleration along Z-axis

$\overline{X}$ distance along X-axis of HiRAP to orbiter center of gravity

$\overline{Y}$ distance along Y-axis of HiRAP to orbiter center of gravity

$\overline{Z}$ 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 μg (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

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

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 μg at times prior to the start of the APU's and 225 μg after this time. For the X-axis, the magnitude of the bandwidth is constant at 45 μg. 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 σ 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

θ 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 μg. 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 μg. 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 μg. 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 ± 300 μg 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 μg. 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 μg of the biases calculated with the first method for RCS thruster 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 nonaerodynamic forces. As these data represent only the aerodynamic forces on the OV during unpowered 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:

$$\bar{C}_a = \exp\left[-A\left(B - \log_{10} N_{\mathrm{Kn}}\right)^C\right] \tag{1}$$

$$\bar{C}_n = \exp\left[-D\left(E - \log_{10} N_{\mathrm{Kn}}\right)^F\right] \tag{2}$$

where

$\bar{C}_a$ normalized axial-force coefficient

$\overline{C}_n$ 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 $\overline{C}_a$ and $\overline{C}_n$ 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:

$$C_a = C_{a,c} + (C_{a,f} - C_{a,c})\,\overline{C}_a$$

$$C_n = C_{n,c} + (C_{n,f} - C_{n,c})\,\overline{C}_n$$

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

$$^{N}\mathrm{Kn} = \frac{(\mathrm{MW}_{76}l)}{\rho d} \tag{3}$$

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

ρ 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

$$C_{i,\mathrm{model}} - C_{i,m} = 0 \pm 0.001 \tag{4}$$

where i represents axial or normal coefficient.

The definition of the measured aerodynamic coefficients $C_{i,m}$ is

$$C_{i,m} = A_{i,m}\left(\frac{1}{2}\rho V^2 \frac{S}{M}\right)^{-1} \tag{5}$$

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

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

where ρ_o 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 axial accelerations (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 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

2 FRONT RPT267

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 μg 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 θ 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 30 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 $C_{i,\mathrm{model}}$ 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

****

STS-06

HiRAP S/N 001

OEX FLIGHT TAPES	SDC DATA TAPES SCIENCE	SDC DATA TAPES HOUSEKEEP	SCIENCE TIMES
ST4859	1. NU1174 SCI061	NU1229 HSKP061	DAY 94, ASCENT 66301-67660 SEC. 18:25:01-18:47:40
ST4875	2. NE0520 SCI062	NE0535 HSKP062	DAY 96, ORBIT 71881-74662 SEC. 19:58:01-20:44:22
ST4860	3. NG0279 SCI063	NG0280 HSKP063	DAY 96, ORBIT 72001-74164 SEC. 20:00:01-20:36:04
ST4876	4. NE0609 SCI064	NE0658 HSKP064	DAY 98, ORBIT 76561-78734 SEC. 21:16:01-21:52:14
ST4853	5. NG0633 SCI065	NG1066 HSKP065	DAY 99, DESCENT 64492-68192 SEC. 17:54:52-18:56:32

+++

DIRECT ACCESS FILE	NUMBER OF POINTS	ACCELEROMETER COUNTS MIN.	ACCELEROMETER COUNTS MAX.	DIRECT ACCESS FILE	NUMBER OF POINTS	COARSE TEMPERATURE COUNTS MIN.	COARSE TEMPERATURE COUNTS MAX.
SCI061	236000	X:0	16383	HSKP061	3689	X:75	104
		Y:0	16383			Y:28	249
ASCENT		Z:0	16383			Z:77	105
SCI062	440000	X:0	16017	HSKP062	6870	X:97	242
		Y:38	16322			Y:69	128
ORBIT		Z:0	14919			Z:97	239
SCI063	376000	X:0	15787	HSKP063	5869	X:98	242
		Y:94	16256			Y:60	123
ORBIT		Z:0	14919			Z:97	239

DIRECT ACCESS FILE	NUMBER OF POINTS	ACCELEROMETER COUNTS MIN.	MAX.	DIRECT ACCESS FILE	NUMBER OF POINTS	COARSE TEMPERATURE COUNTS MIN.	MAX.
SCI064	377791	X:5177	9688	HSKP064	5902	X:105	125
		Y:3356	16116			Y:28	249
ORBIT		Z:1990	16383			Z:105	124
SCI065	643453	X:0	16383	HSKP064	10053	X:5	209
		Y:0	16383			Y:6	214
DESCENT		Z:0	16383			Z:6	209

DAY 99
EXTENDED BET SOURCE FILE NC0709
AEROBET SOURCE FILE NK0917

STS-07

HiRAP S/N 001

OEX FLIGHT TAPES	SDC DATA TAPES SCIENCE	HOUSEKEEP	SCIENCE TIMES
ST5075	1. NA0210 SCI071	NA0216 HSKP071	DAY 169, ASCENT 41041-42362 SEC. 11:24:01-11:46:02
ST5076	2. NA0254 SCI072	NA0280 HSKP072	DAY 175, ORBIT 45656-47789 SEC. 12:40:56-13:16:29
ST5077	3. ND0379 SCI073	ND0571 HSKP073	DAY 175, DESCENT 47794-50492 SEC. 13:16:34-14:01:32

++

DIRECT ACCESS FILE	NUMBER OF POINTS	ACCELEROMETER COUNTS MIN.	MAX.	DIRECT ACCESS FILE	NUMBER OF POINTS	COARSE TEMPERATURE COUNTS MIN.	MAX.
SCI071	224000	X:0	16383	HSKP071	3504	X:85	114
		Y:0	16383			Y:84	116
ASCENT		Z:0	16383			Z:87	207
SCI072	370801	X:0	15825	HSKP072	5793	X:61	177
		Y:0	16342			Y:63	230
ORBIT		Z:0	16383			Z:62	207
SCI073	468871	X:5627	16383	HSKP073	7328	X:90	104
		Y:0	16383			Y:96	228
DESCENT		Z:0	16383			Z:90	207

DAY	175
EXTENDED BET SOURCE FILE	NC0709
AEROBET SOURCE FILE	NF1206

STS-08

HiRAP S/N 001

OEX FLIGHT TAPES	SDC DATA TAPES SCIENCE	HOUSEKEEP	SCIENCE TIMES
ST5237	1. NU0229 SCI081	NU0271 HSKP081	DAY 242, ASCENT 22921-24321 SEC. 06:22:01-06:45:21
ST5238	2. NU0279 SCI082	NU0632 HSKP082	DAY 244, ORBIT 25921-28297 SEC. 07:12:01-07:51:37
ST5239	3. NU0705 SCI083	NV0276 HSKP083	DAY 248, DESCENT 23581-27765 SEC. 06:33:01-07:42:45

++

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

DAY	248
EXTENDED BET SOURCE FILE	NC0709
AEROBET SOURCE FILE	NX0484

STS-09

HiRAP S/N 002
(BEFORE RECALIBRATION)

OEX FLIGHT TAPES	SDC DATA TAPES SCIENCE	HOUSEKEEP	SCIENCE TIMES
ST5340	1. NT0838 SCI091	NT1010 HSKP091	DAY 342, ORBIT 9021-10802 SEC. 02:30:21-03:00:02
ST5341	2. S092 SCI092	NU0141 HSKP092	DAY 342, ORBIT 11701-15012 SEC. 03:15:01-04:10:12
ST5342	3. NU0171 SCI093	NU0204 HSKP093	DAY 342, DESCENT 81361-82352 SEC. 22:36:01-22:52:32
ST5343	4. NU0231 SCI094	NU0340 HSKP094	DAY 342, DESCENT 82381-85802 SEC. 22:53:01-23:50:02

++

DIRECT ACCESS FILE	NUMBER OF POINTS	ACCELEROMETER COUNTS MIN.	MAX.	DIRECT ACCESS FILE	NUMBER OF POINTS	COARSE TEMPERATURE COUNTS MIN.	MAX.
SCI091	307374	X:7537	13234	HSKP091	4809	X:42	247
		Y:4594	16383			Y:42	234
ORBIT		Z:263	10804			Z:42	237
SCI092	573000	X:9782	12615	HSKP092	8946	X:99	135
		Y:6348	16383			Y:97	134
ORBIT		Z:2055	16383			Z:98	135
SCI093	171391	X:0	13706	HSKP093	2677	X:91	119
		Y:4413	11841			Y:91	118
DESCENT		Z:0	11742			Z:91	118
SCI094	592449	X:0	16383	HSKP094	9258	X:119	146
		Y:0	16383			Y:118	146
DESCENT		Z:0	16383			Z:118	146

DAY	342
EXTENDED BET SOURCE FILE	NC0709
AEROBET SOURCE FILE	NL0701

STS-41B

HiRAP S/N 001

OEX FLIGHT TAPES	SDC DATA TAPES SCIENCE	HOUSEKEEP	SCIENCE TIMES
ST5370	1. NF0156 SCI111	NF0158 HSKP111	DAY 34, ASCENT 45901-48002 SEC. 12:45:01-13:20:02
ST5371	2. NF0203 SCI112	NF0205 HSKP112	DAY 37, ORBIT 53722-55132 SEC. 14:55:22-15:18:52
ST5372	3. NF0206 SCI113	NF0234 HSKP113	DAY 42, DESCENT 40501-44512 SEC. 11:15:01-12:21:52

++

DIRECT ACCESS FILE	NUMBER OF POINTS	ACCELEROMETER COUNTS MIN.	MAX.	DIRECT ACCESS FILE	NUMBER OF POINTS	COARSE TEMPERATURE COUNTS MIN.	MAX.
SCI111	365000	X:0	16383	HSKP111	5705	X:76	117
		Y:0	16383			Y:75	120
ASCENT		Z:0	16383			Z:78	118
SCI112	244000	X:1036	16376	HSKP112	3823	X:18	55
		Y:12	16383			Y:18	59
ORBIT		Z:4	16383			Z:20	55
SCI113	697104	X:0	16383	HSKP112	10891	X:28	72
		Y:0	16383			Y:30	72
DESCENT		Z:0	16383			Z:29	73

DAY	42
EXTENDED BET SOURCE FILE	NC0709
AEROBET SOURCE FILE	NF0349

STS-41C

HiRAP S/N 001

OEX FLIGHT TAPES	SDC DATA TAPES SCIENCE	HOUSEKEEP	SCIENCE TIMES
ST5630	1. NS0671 SCI131	NS0672 HSKP131	DAY 97, ASCENT 49801-51422 SEC. 13:50:01-14:17:02

ST5631 ----------	2. NS0680 SCI132	NS0813 HSKP132	DAY 99, ORBIT 57011-60102 SEC. 15:51:51-16:41:42
ST5632 ----------	3. NS0816 SCI133	NS0832 HSKP133	DAY 101, ORBIT 48661-48842 SEC. 13:31:01-13:34:02
ST5633 ----------	4. NS0846 SCI134	NS0909 SCI134	DAY 102, ORBIT 31321-31462 SEC. 08:42:01-08:44:22
ST5634 ----------	5. NT1246 SCI135	NU0376 HSKP135	DAY 104, DESCENT 44132-49372 SEC. 12:15:32-13:42:52

+++

DIRECT ACCESS FILE	NUMBER OF POINTS	ACCELEROMETER COUNTS MIN.	MAX.	DIRECT ACCESS FILE	NUMBER OF POINTS	COARSE TEMPERATURE COUNTS MIN.	MAX.
SCI131	282000	X:0	11513	HSKP131	4403	X:75	104
		Y:0	16383			Y:75	107
ASCENT		Z:0	16383			Z:76	209
SCI132	516000	X:65	13970	HSKP132	8070	X:71	108
		Y:3	16328			Y:76	114
ORBIT		Z:0	16383			Z:72	107
SCI133	24447	X:5286	11950	HSKP133	381	X:49	53
		Y:5070	10962			Y:51	56
ORBIT		Z:0	12747			Z:50	54
SCI134	23551	X:0	11282	HSKP134	367	X:91	94
		Y:5252	11046			Y:96	99
ORBIT		Z:2862	11719			Z:91	94
SCI135	911020	X:0	16383	HSKP135	14236	X:59	243
		Y:0	16383			Y:61	205
DESCENT		Z:1	16383			Z:59	207

DAY	104
EXTENDED BET SOURCE FILE	NC0709
AEROBET SOURCE FILE	NC0740

STS-51B

HiRAP S/N 001

OEX FLIGHT TAPES	SDC DATA TAPES SCIENCE	HOUSEKEEP	SCIENCE TIMES
JH51B8	1. NM0236 SCI241	NM0323 HSKP241	DAY 119, ASCENT 57241-58762 SEC. 15:54:01-16:19:22
JH51B9	2. NC0533 SCI242	NC0805 HSKP242	DAY 120, ORBIT 7381-7842 SEC. 02:03:01-02:10:42
JH5B10	3. ND1215 SCI243	ND1218 HSKP243	DAY 120, ORBIT 80461-80941 SEC. 22:21:01-22:29:01
JH5B11	4. ND1219 SCI244	ND1237 SCI244	DAY 126, DESCENT 53450-56917 SEC. 14:50:01-15:48:37
JH5B12	5. NK0659 SCI245	NL0257 HSKP245	DAY 126, DESCENT 56911-57962 SEC. 15:48:01-16:06:02

++

DIRECT ACCESS FILE	NUMBER OF POINTS	ACCELEROMETER COUNTS MIN.	MAX.	DIRECT ACCESS FILE	NUMBER OF POINTS	COARSE TEMPERATURE COUNTS MIN.	MAX.
SCI241	264371	X:0	1040	HSKP241	4130	X:86	117
		Y:0	16383			Y:86	120
ASCENT		Z:0	16383			Z:88	117
SCI242	71422	X:6208	11736	HSKP242	1115	X:127	135
		Y:234	11213			Y:131	139
ORBIT		Z:6570	12244			Z:128	135
SCI243	81599	X:9020	9142	HSKP243	1274	X:137	142
		Y:8265	8559			Y:143	148
ORBIT		Z:9394	9683			Z:136	141
SCI244	593479	X:0	16383	HSKP244	9412	X:28	85
		Y:0	16383			Y:28	90
DESCENT		Z:0	16383			Z:30	204
SCI245	182591	X:16383	16383	HSKP245		X:	
		Y:0	16383			Y:	
DESCENT		Z:16383	16383			Z:	

DAY 126
EXTENDED BET SOURCE FILE NC0709
AEROBET SOURCE FILE NN1264

STS-51F

HiRAP S/N 002
(RECALIBRATED)

OEX FLIGHT TAPES	SDC DATA TAPES SCIENCE	HOUSEKEEP	SCIENCE TIMES
JH5F14	1. NY0338 SCI261	NY0449 HSKP261	DAY 210, ASCENT 75181-76692 SEC. 20:53:01-21:18:12
JH5F15	2. NY0340 SCI262	NY0341 HSKP262	DAY 213, ORBIT 5401-6057 SEC. 01:30:01-01:40:57
JH5F16	3. NY0342 SCI263	NY0343 HSKP263	DAY 218, DESCENT 67021-68702 SEC. 18:37:01-19:05:02
JH5F17	4. NY0344 SCI264	NY0345 HSKP264	DAY 218, DESCENT 68697-69600 SEC. 19:04:57-19:20:00

+++

DIRECT ACCESS FILE	NUMBER OF POINTS	ACCELEROMETER COUNTS MIN.	MAX.	DIRECT ACCESS FILE	NUMBER OF POINTS	COARSE TEMPERATURE COUNTS MIN.	MAX.
SCI261	241000	X:0	16383	HSKP261	3774	X:96	130
		Y:0	16383			Y:51	129
ASCENT		Z:0	16383			Z:104	216
SCI262	110590	X:0	12190	HSKP262	1727	X:108	116
		Y:6845	16064			Y:106	114
ORBIT		Z:0	10216			Z:107	116
SCI263	290532	X:0	16094	HSKP263	4550	X:83	100
		Y:1552	16383			Y:81	98
DESCENT		Z:0	15356			Z:82	215
SCI264	156087	X:0	16383	HSKP264	2454	X:99	107
		Y:0	16383			Y:97	106
DESCENT		Z:0	16383			Z:98	107

DAY 218
EXTENDED BET SOURCE FILE NC0709

AEROBET SOURCE FILE NP1083

STS-61A

HiRAP S/N 002
(RECALIBRATED)

OEX FLIGHT TAPES	SDC DATA TAPES SCIENCE	SDC DATA TAPES HOUSEKEEP	SCIENCE TIMES
JH6A15	1. NM1180 SCI301	NM1267 HSKP301	DAY 303, ASCENT 60661-64137 SEC. 16:51:01-17:47:57
JH6A16	2. NM1271 SCI302	NN0129 HSKP302	DAY 303, ORBIT 85021-492 SEC. 23:37:01-00:08:12
JH6A17	3. NM1065 SCI303	NS0628 HSKP303	DAY 304, ORBIT 23161-25552 SEC. 06:26:01-07:05:52
JH6A18	4. NS0818 SCI304	NS0844 HSKP304	DAY 309, ORBIT 78301-880277 SEC. 21:45:01-22:17:52
JH6A19	5. NA0160 SCI305	NA0260 HSKP305	DAY 310, ORBIT 27601-30022 SEC. 07:40:01-08:20:22
JH6A20	6. NH1067 SCI306	NH1155 HSKP306	DAY 310, DESCENT 59122-62482 SEC. 16:25:22-17:21:22
JH6A21	7. NH1215 SCI307	NM1216 HSKP307	DAY 310, DESCENT 62461-64132 SEC. 17:21:01-17:48:52

Axes X and Z are pegged for this entire channel.

DIRECT ACCESS FILE	NUMBER OF POINTS	ACCELEROMETER COUNTS MIN.	ACCELEROMETER COUNTS MAX.	DIRECT ACCESS FILE	NUMBER OF POINTS	COARSE TEMPERATURE COUNTS MIN.	COARSE TEMPERATURE COUNTS MAX.
SCI301	604000	X:0	13556	HSKP301	9442	X:79	138
		Y:0	16383			Y:79	215
ASCENT		Z:0	16383			Z:79	217
SCI302	318271	X:2184	4253	HSKP302	4972	X:143	178
		Y:7117	12832			Y:142	186
ORBIT		Z:2087	4129			Z:142	217

SCI303	408000	X:3769	9084	HSKP303	6376	X:249	254
		Y:5734	13944			Y:248	254
ORBIT		Z:2915	8174			Z:249	254
SCI304	340991	X:0	3635	HSKP304	5327	X:79	101
		Y:2020	12108			Y:77	99
ORBIT		Z:1635	5347			Z:78	101
SCI305	420799	X:2009	3931	HSKP305	6574	X:102	128
		Y:6742	9997			Y:100	127
ORBIT		Z:1048	4455			Z:101	128
SCI306	584239	X:0	16383	HSKP306	9129	X:52	153
		Y:0	16383			Y:59	153
DESCENT		Z:0	16383			Z:121	215
SCI307	287871	X:0	16383	HSKP307	4497	X:148	154
		Y:0	16383			Y:148	153
DESCENT		Z:16383	16383			Z:147	154

DAY 310
EXTENDED BET SOURCE FILE NC0709
AEROBET SOURCE FILE NY1721

STS-61C

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	321H HSKP321	DAY 15, ORBIT 30311-32851 SEC. 08:25:11-09:07:31
JHDT02	2. 322S SCI322	322H HSKP322	DAY 16, ORBIT 59363-61017 SEC. 16:29:23-16:56:57
JHDT03	3. 323S SCI323	323H HSKP323	DAY 17, ORBIT 55246-56473 SEC. 15:20:46-15:41:13
JH6C13	4. 324S SCI324	324H HSKP324	DAY 18, DESCENT 45818-47737 SEC. 12:43:38-13:15:37

JH6C13 __________	5.	325S SCI325	325H HSKP325	DAY 18, DESCENT 47747-48429 SEC. 13:15:47-13:27:09
JH6C13 __________	6.	326S SCI326	326H HSKP326	DAY 18, DESCENT 48371-49000 SEC. 13:26:11-13:36:40

++

DIRECT ACCESS FILE	NUMBER OF POINTS	ACCELEROMETER COUNTS MIN.	MAX.	DIRECT ACCESS FILE	NUMBER OF POINTS	COARSE TEMPERATURE COUNTS MIN.	MAX.
SCI321	280975	X:6462	16383	HSKP321	4454	X:29	82
		Y:5715	16383			Y:28	88
ORBIT		Z:0	16383			Z:31	83
SCI322	183615	X:0	16383	HSKP322	2910	X:5*	101
		Y:0	16383			Y:5*	108
ORBIT		Z:0	16383			Z:5*	102
SCI323	136095	X:0	16383	HSKP323	2157	X:5*	132
		Y:0	16383			Y:5*	139
ORBIT		Z:0	16383			Z:5*	132
SCI324	213263	X:0	12803	HSKP324	3375	X:144	160
		Y:3334	11835			Y:151	167
DESCENT		Z:0	16383			Z:145	160
SCI325	77000	X:6703	12613	HSKP325	1199	X:159	165
		Y:0	16383			Y:166	172
DESCENT		Z:0	16383			Z:159	166
SCI326	71000	X:8762	16383	HSKP326	1104	X:164	169
		Y:2059	16383			Y:171	175
DESCENT		Z:0	16383			Z:164	170

*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.

DAY	18
EXTENDED BET SOURCE FILE	NC0709
AEROBET SOURCE FILE	NG1083

Appendix B

Orbiter Descent Event Times

Event	Altitude, km	Time, sec
STS-06		
Time blanked out for APU shift	249 to 239	65 390 to 65 470
Deorbit burn	249 to 291	64 [illegible]90 to 64 650
Pitch maneuver	291 to 281	[illegible]25 to 65 000
X-axis saturation	85	66 447
Z-axis saturation	100	66 344
XBET epoch	289	64 510
ABET epoch	123	66 200
STS-07		
Time blanked out for APU shift	253 to 239	47 520 to 47 625
Deorbit burn	295 to 297	46 560 to 46 730
Pitch maneuver	298 to 295	46 800 to 47 060
X-axis saturation	84	48 605
Z-axis saturation	97	48 520
XBET epoch	295	46 560
ABET epoch	208	47 840
STS-08		
Time blanked out for APU shift	210 to 207	25 035 to 25 085
Deorbit burn	219 to 220	24 440 to 24 610
Pitch maneuver	220 to 211	24 700 to 25 020
X-axis saturation	84	26 086
Z-axis saturation	98	25 988
XBET epoch	219	24 450
ABET epoch	187	25 310
STS-09		
Time blanked out for APU shift	217 to 214	82 950 to 82 990
Deorbit burn	240 to 238	82 320 to 82 480
Pitch maneuver	236 to 214	82 600 to 82 990
X-axis saturation	85	84 016
Z-axis saturation	97	83 935
XBET epoch	240	82 320
ABET epoch	161	83 843
STS-41B		
Time blanked out for APU shift	236 to 233	41 534 to 41 559
Deorbit burn	276 to 277	40 560 to 40 750
Pitch maneuver	278 to 271	40 850 to 41 120
X-axis saturation	82	42 574
Z-axis saturation	97	42 474
XBET epoch	276	40 570
ABET epoch	252	41 380

Event	Altitude, km	Time, sec
STS-41C		
Time blanked out for APU shift	319 to 313	46 495 to 46 520
Deorbit burn	500 to 505	44 960 to 45 220
Pitch maneuver	506 to 500	45 300 to 45 500
X-axis saturation	90	47 480
Z-axis saturation	99	47 396
XBET epoch	500	44 600
ABET epoch	213	46 890
STS-51B		
Time blanked out for APU shift	288 to 260	55 575 to 55 725
Deorbit burn	357 to 364	54 280 to 54 570
Pitch maneuver	365 to 362	54 658 to 54 887
X-axis saturation	85	56 627
Z-axis saturation	98	56 540
XBET epoch	357	54 282
ABET epoch	123	56 400
STS-51F		
Time blanked out for APU shift	255 to 250	68 525 to 68 555
Deorbit burn	321 to 324	67 382 to 67 552
Pitch maneuver	325 to 322	67 690 to 67 851
X-axis saturation	79	69 553
Z-axis saturation	92	69 440
XBET epoch	321	67 372
ABET epoch	123	69 260
STS-61A		
Time blanked out for APU shift	267 to 264	61 220 to 61 240
Deorbit burn	334 to 338	60 030 to 60 200
Pitch maneuver	340 to 339	60 269 to 60 504
X-axis saturation	79	62 276
Z-axis saturation	93	62 167
XBET epoch	334	60 022
ABET epoch	121	62 000
STS-61C		
Time blanked out for APU shift	261 to 257	47 700 to 47 725
Deorbit burn	328 to 332	46 472 to 46 704
Pitch maneuver	334 to 329	46 815 to 47 065
X-axis saturation	84	48 711
Z-axis saturation	97	48 625
XBET epoch	328	46 462
ABET epoch	210	48 000

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), $V/°F$
θ_M	temperature for zero volts in fine range M (from table C1 for each sensor), °F
INT	integral part of

Then

$$T_C = \theta_1 + V_C/K_C$$

$$M = \mathrm{INT}(0.5 + 1.6\,V_C)$$

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
θ_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
θ_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	0.03904
K_F, V/°F	0.28254
θ_M, °F:	
$M = 1$	23.86
$M = 2$	39.50
$M = 3$	55.15
$M = 4$	70.84
$M = 5$	86.53
$M = 6$	102.36
$M = 7$	118.20
$M = 8$	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

$$T_C = \theta_1 + V_C/K_C$$

This equation is modified to

$$T_C = \theta_C + V_C/K_C$$

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	0.04096
K_F, V/°F	0.29417
θ_C, °F	30.70
θ_M, °F:	
$M = 1$	30.56
$M = 2$	45.74
$M = 3$ ("halfway" range)	60.75
$M = 4$	75.76
$M = 5$	90.83
$M = 6$	105.56
$M = 7$ ("halfway" range)	120.57
$M = 8$	135.57

(b) Y-axis

K_C, V/°F	0.04089
K_F, V/°F	0.29420
θ_C, °F	29.98

Table C2. Concluded

(b) Concluded

θ_M, °F:	
$M = 1$	29.92
$M = 2$	45.04
$M = 3$ ("halfway" range)	60.03
$M = 4$	75.02
$M = 5$	90.19
$M = 6$	105.11
$M = 7$ ("halfway" range)	120.07
$M = 8$	135.04

(c) Z-axis

K_C, V/°F	0.04073
K_F, V/°F	0.29275
θ_C, °F	29.91
θ_M, °F:	
$M = 1$	29.99
$M = 2$ ("halfway" range)	44.97
$M = 3$	59.95
$M = 4$ ("halfway" range)	75.06
$M = 5$	90.17
$M = 6$ ("halfway" range)	105.32
$M = 7$	120.47
$M = 8$	135.42

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	coarse temperature monitor voltage
V_F	fine temperature monitor voltage
V_e	final (corrected) temperature monitor voltage
M	fine range serial number, 1 to 8
G	slope (gain) of V_F relative to V_C
V_M	value of V_C corresponding to 0 V in fine range M
INT	integral part of

Then

$$M = \mathrm{INT}(0.5 + 1.6V_C)$$

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 μg 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; $G = 7.181 \pm 0.005$]

M	V_M, V
1	−0.0014
2	.6127
3	1.2268
4	1.8407
5	2.4547
6	3.0685
7	3.6823
8	4.2963

(b) Y-axis

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

M	V_M, V
1	0.0000
2	.6142
3	1.2283
4	1.8424
5	2.4564
6	3.0704
7	3.6843
8	4.2981

Table C3. Concluded

(c) Z-axis

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

M	V_M, V
1	−0.0002
2	.6140
3	1.2281
4	1.8424
5	2.4566
6	3.0705
7	3.6845
8	4.2984

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

(a) X-axis

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

M	V_M, V
1	
2	0.6158
3	1.2302
4	1.8463
5	2.4609
6	3.0759
7	3.6912
8	4.3047

(b) Y-axis

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

M	V_M, V
1	
2	0.6145
3	
4	
5	2.4582
6	3.0729
7	3.6875
8	4.3021

Table C4. Concluded

(c) Z-axis

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

M	V_M, V
1	
2	0.6141
3	
4	
5	2.4579
6	3.0726
7	3.6874
8	4.3019

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 thrustor 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 thrustor 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

STS-09 Thrust Removal
RCS Pressure Data Tape: VSN = NC1260

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

STS-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 thrustor 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 thrustor activity removed:

Thrust Start = Firing Time - TLAG

Thrust Stop = Firing Time + TLAG1

Appendix E
Orbiter Weight at Entry and Center-of-Gravity Locations

Flight	Weight at entry interface, lb	Center of gravity, in., at entry interface along—		
		X-axis	Y-axis	Z-axis
STS-06	191 384.0	1101.2	0.3	371.5
STS-07	204 983.0	1091.3	−.6	373.3
STS-08	205 020.0	1091.5	−.1	373.5
STS-09	221 143.4	1087.3	−.1	373.7
STS-41B	202 966.5	1090.7	1.3	372.6
STS-41C	198 152.8	1101.5	−.1	371.6
STS-51B	214 787.4	1085.7	−.3	373.4
STS-51F	218 227.4	1082.3	−.6	373.4
STS-61A	215 255.4	1085.5	−.4	374.2
STS-61C	211 194.4	1085.2	.4	371.4

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	reference temperature serial number, from 1 to 5
T_r	reference temperature at serial r, °F
K_r	scale factor at serial r, V/mg
$K_{T,r}$	scale factor temperature coefficient from r to $r+1$, (V/mg)/°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

$$K = K_r + (T_F - T_r)K_{T,r} \text{ V/m}g$$

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	reference temperature serial number, 1 to 5
T_r	reference temperature at serial r, °F
B_r	bias at serial r relative to nominal 30°F value, μg
$B_{T,r}$	bias temperature coefficient over interval from serial r to $r+1$, $(B_{r+1} - B_r)(T_{r+1} - T_r)$, μg

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

Also define the following:

$T_{F,0}$	datum on-orbit fine temperature, °F
$T_{F,1}$	entry fine temperature, °F
B_0	bias at $T_{F,0}$ relative to nominal 30°F value, μg
B_1	bias at $T_{F,1}$ relative to nominal 30°F value, μg
ΔB_0	bias change from datum, μg

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

Flight	Bias slope, $\mu g/°F$, from—		Bias, μg, from—	
	Post-APU procedure	Ground calibration procedure	Post-APU procedure	Ground calibration procedure
STS-06	−30.6	−26.0	−1457	−1394
STS-07	−22.3	−20.0	−279	−190
STS-08	−26.7	−20.4	−43	51
STS-09	−23.3	−16.4	−3121	−676
STS-41B	−25.5	−21.1	107	217
STS-41C	−22.8	−20.0	−357	−210
STS-51B	−23.8	−20.0	−320	−50
STS-51F	−18.2	−18.8	5448	5426
STS-61A	−21.4	−17.3	4942	4968
STS-61C	−27.9	−25.3	−2054	−2098

Y-axis

Flight	Bias slope, $\mu g/°F$, from—		Bias, μg, from—	
	Post-APU procedure	Ground calibration procedure	Post-APU procedure	Ground calibration procedure
STS-06	10.5	15.0	495	485
STS-07	4.1	10.0	−8	−19
STS-08	8.9	10.6	−91	−90
STS-09	29.7	23.6	808	−493
STS-41B	4.1	10.3	−148	−153
STS-41C	27.8	9.1	−5	−13
STS-51B	6.1	10.9	59	−61
STS-51F	29.0	29.9	889	824
STS-61A	27.7	28.9	1656	1513
STS-61C	15.3	13.5	872	807

Z-axis

Flight	Bias slope, $\mu g/°F$, from—			Bias, μg, from—		
	Post-APU procedure	Ground calibration procedure	Alternate procedure	Post-APU procedure	Ground calibration procedure	Alternate procedure
STS-06	−21.2	−15.8	(a)	−1587	−1666	−1586
STS-07	−19.3	−20.0	↓	−617	−722	−617
STS-08	−28.3	−21.6	↓	−389	−494	−389
STS-09	−13.0	−8.7	↓	−224	−216	−223
STS-41B	−25.3	−19.9	↓	−212	−362	−213
STS-41C	−19.7	−19.9	↓	−659	−742	−660
STS-51B	−22.1	−21.0	↓	−615	−579	−614
STS-51F	−12.4	−12.0	↓	5449	5404	5449
STS-61A	−9.9	−10.7	↓	5195	5126	5198
STS-61C	−18.5	−18.0	↓	−1806	−1977	−1804

[a]Not available.

Appendix H

Source Codes

```
      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(1,1) (INBUF(1),INBUF(615))
      IF(UNIT(1)) 20,22,25
 20   ILEN=LENGTH(1)
      IF(NCYC .GT. 1000) STOP
      CALL BYTE8(ILEN,INBUF)
      GO TO 10
 22   NFILE=NFILE+1
      ENDFILE 2
      PRINT 100, NFILE,BTIME,ETIME,NCYC
 100  FORMAT(///1X,'END OF FILE ',I2/5X,'FILE BEGAN AT ',F9.3/5X,'FILE E
     1NDED AT ',F9.3/5X,'NO. OF DATA CYCLES = ',I4)
      NCYC=0
      BUFFER IN(1,1)(INBUF(1),INBUF(615))
      IF(UNIT(1)) 20,23,25
 23   ENDFILE 2
      STOP
 25   PRINT 250, ETIME
 250  FORMAT(///1X,'PARITY ERROR --- TIME OF LAST GOOD DATA FRAME = ',
     1F9.3)
 30   STOP
      END
      SUBROUTINE BYTE8(ILEN,INBUF)
      DIMENSION INBUF(ILEN)
      COMMON/MASTER/IDATA(4608)
      IF(ILEN .LT. 615) THEN
      IFLAG=208
      K=1
      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
```

```
      IF(L .EQ. 61) K=1
 20   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)
 20   K=K+72
 25   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    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,51,10)
      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,51,10)
      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*10+ITEN)*10+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(1H1,'FORMAT ID        = ',F6.2,/1X,'CPU WORD SIZE    = ',
     1F6.2,/1X,'FRAME SIZE       = ',F6.2,/1X,'NUMBER OF FRAME = ',F6.2,/
     21X,'DATA RANGE       = ',F6.2,/1X,9A10//)
      RETURN
      END
```

```
      SUBROUTINE ASC2DC(IDIS)
      DIMENSION IDIS(9)
      COMMON/MASTER/IDATA(208),IDUM(3888)
      INTEGER ASCII(17),DSPLY(17)
      DATA ASCII/O"53",O"55",O"52",O"57",O"50",O"51",O"44",O"75",O"40",
     1O"54",O"56",O"43",O"133",O"135",O"45",O"73",O"72"/
      DATA DSPLY/O"45",O"46",O"47",O"50",O"51",O"52",O"53",O"54",O"55",
     1O"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
      IDCBIT=1
      IDWD=1
      DO 50 I=1,88
      IF((IDATA(120+I) .LT. O"101") .OR. (IDATA(120+I) .GT.O"132")) GO T
     1O 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),IDCBIT,6)
      IDCBIT=IDCBIT+6
      IF(IDCBIT.LE. 60) GO TO 50
      IDWD=IDWD+1
      IDCBIT=IDCBIT-60
50    CONTINUE
      RETURN
      END
```

```
      PROGRAM SCIREAD(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE9,TAPE1,
     *TAPE2)
C
C     THIS ROUTINE READS AND PRINTS RAW DATA FROM HIRAP TAPES
C
C     WRITTEN BY: J. W. RUSSELL,  DEC 1985
C
      DIMENSION DATAPT(4),NAMES(4),XMAX(4),XMIN(4)
C
        CHARACTER*8 NAMES
       DATA NAMES/'TIME-SEC','X-COUNTS','Z-COUNTS','Y-COUNTS'/
C
      WRITE(6,800)
800   FORMAT(/5X,'JTREAD STARTED',/)
      START=0.0
      KMAX = 225
      IREV = 0
       IGAP = 0
        NCHAN = 4
C
C     READ TAPE HEADER
C
C     READ(9,END=600) ISN,NCHAN,(NAMES(I),I=1,NCHAN),
C    *                (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
C
C
C     WRITE(6,104) ISN,NCHAN,(HDR(I),I=1,8)
C104  FORMAT(1H1,5X,7HFILE   = ,I7,5X,8HNCHAN = ,1I2,5X,15HFOUND HEADER =
C    * ,8A10 )
C     WRITE(6,303)   (NAMES(I),I=1,NCHAN)
C     WRITE(6,303)   (IUNITS(I),I=1,NCHAN)
303   FORMAT(1X,10(A10,2X))
C
C
       READ(9,END=91) (DATAPT(I),I=1,NCHAN)
       WRITE(1) DATAPT(1),DATAPT(2),DATAPT(3),DATAPT(4)
C
C     SET MINIMUM AND MAXIMUM VALUES
C
       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
C
C     MOVE TO DESIRED START AND PRINT RAW DATA UNTIL END OF FILE
C     IS REACHED ELIMINATING BAD POINTS AND TIME REVERSALS
C
90    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
40    FORMAT(5X,'TIME REVERSAL BETWEEN TIME =',1F12.3,' AND TIME =',
     *1F12.3 )
      IREV = 0
C
      IF (DATAPT(1).GT.(TIMENEW+.01)) THEN
          IF (IGAP.EQ.0) THEN
           WRITE(6,113) DATAPT(1),TIMENEW
113       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
C
      END IF
C
C
      TIMENEW = DATAPT(1)
C
        IGAP = 0
C
      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)
3     CONTINUE
C
      WRITE(1) DATAPT(1),DATAPT(2),DATAPT(3),DATAPT(4)
C
```

```
      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)
C
C     THIS ROUTINE READS AND PRINTS RAW DATA FROM HIRAP TAPES
C
C     WRITTEN BY HIRAP PROJECT GROUP - JANUARY 1986
C
      DIMENSION DATAPT(9),NAMES(9),IUNITS(9),HDR(9),XMAX(9),
     *          XMIN(9)
C
         CHARACTER*8 NAMES
       DATA NAMES/'TIME-SEC','X-FINE  ','X-COARSE','Y-FINE  ',
     *           'Y-COARSE','Z-FINE  ','Z-COARSE','POS-V   ',
     *           'NEG-V   '/
C     THE FOLLOWING IS A LIST OF INPUTS TO THIS PROGRAM
C
C     KMAX              COUNT AT WHICH EACH POINT IS WRITTEN TO OUTPUT
C
      KMAX = 4
        NCHAN = 9
        JREV = 0
         IGAP = 0
         STPTIM = 49255.
C
        READ(9) IDAY,IHOUR,IMIN
        WRITE(6,123) IDAY,IHOUR,IMIN
123    FORMAT(2X,'IDAY, IHOUR, IMIN = ',3(2X,I5)/)
C
C
      READ(9,END=1) (DATAPT(I),I=1,NCHAN)
      WRITE(1) (DATAPT(I),I=1,NCHAN)
C
C
C
C
C     DEFINE HEADER IF NECCESSARY
C
C
C
1     CONTINUE
C
      IF (EOF(9) .NE. 0)  THEN
            WRITE(6,10)
            GO TO 7
      ENDIF
C
C
C     SET MINIMUM AND MAXIMUM VALUES
C
      DO 2 I = 2,NCHAN
      XMIN(I) = DATAPT(I)
      XMAX(I) = DATAPT(I)
2     CONTINUE
C
```

```
      ITIME = 0
      START = DATAPT(1)
      TIMENEW = START - 0.1
      KCNT = 0
      NUMPTS = 0
C
C     MOVE TO DESIRED START AND PRINT RAW DATA UNTIL END OF FILE
C     IS REACHED ELIMINATING BAD POINTS AND TIME REVERSALS
C
3     CONTINUE
      READ(9,END=5) (DATAPT(I),I=1,NCHAN)
C
        IF (DATAPT(1).GT.STPTIM) GO TO 108
C
      IF (DATAPT(1) .LT. TIMENEW) THEN
C
           IF (TIMENEW . GT. 86390.0 .AND. DATAPT(1) .LT. 10.0) THEN
                ITIME = 1
           ELSE
               IF (JREV.EQ.0) THEN
                WRITE(6,13) DATAPT(1)
                JREV = 1
               END IF
                GO TO 3
           ENDIF
C
      ELSE IF (DATAPT(1). EQ. TIMENEW) THEN
        GO TO 3
C
      END IF
C
C
        IF (DATAPT(1).GT.(TIMENEW+.7)) THEN
           IF (IGAP.EQ.0) THEN
            WRITE(6,113) DATAPT(1),TIMENEW
113        FORMAT(2X,'TIME GAP BETWEEN TIME ',F12.3,' AND ',F12.3)
            IGAP = 1
           END IF
C
C
        IF (DATAPT(1).GT.(TIMENEW+50.).AND.DATAPT(1).LT.
     *     (TIMENEW+100.))GOTO 3
        END IF
C
C
C
C
      JREV = 0
      TIMENEW = DATAPT(1)
      IGAP = 0
      NUMPTS = NUMPTS + 1
      KCNT = KCNT + 1
      DATAPT(1) = DATAPT(1) + ITIME * 86400.0
C
```

```
      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)
4     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
C
           DO 6 I = 2,NCHAN
           WRITE(6,16) I,NAMES(I),XMIN(I),XMAX(I)
6          CONTINUE
C
      ENDIF
C
      WRITE(6,17) NUMPTS
C
7     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(1H1,4X,'ISN =',I7,5X,'NCHAN =',1I2,5X,
     *'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 TCALIB(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7,TAPE1)
C
      DIMENSION NAMES(10),IUNITS(10),TITLE(8),DATAPT(7),TF(3),TC(3),
     *          VC(3),TEMP(3)
C
      KMAX1 = 4
      KMAX2 = 20
      NP = 0
      REWIND(7)
      READ(5,52,END=1) SN,T1,T2
1     CONTINUE
      IF(EOF(5) .NE. 0) THEN
            WRITE(6,58)
            GO TO 10
      ELSE
            ISN = SN
            WRITE(6,60) ISN,T1,T2
            WRITE(6,53)
      ENDIF
      IF (EOF(7) .NE. 0) THEN
            WRITE(6,57)
            GO TO 10
      ENDIF
C
      KCNT1 = 3
      KCNT2 = 19
4     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)
C
C     CALL TVCCALC TO GET VOLTAGES AND TEMPERATURES
C
            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
C
      GO TO 4
```

```
5       CONTINUE
         WRITE(6,57)
10      CONTINUE
         WRITE(6,59) NP
C
         STOP
C
52      FORMAT(3F8.0)
53      FORMAT(/ 4X,8HTIME,SEC,6X,3HVCX,9X,3HVCY,9X,3HVCZ,7X,6HTEMP-X,
       *        6X,6HTEMP-Y,6X,6HTEMP-Z, /)
54      FORMAT(1F12.3,6F12.5)
57      FORMAT(/ 5X,42HEND OF FILE FOUND ON TAPE 7 - PROGRAM STOP /)
58      FORMAT(/ 5X,42HEND OF FILE FOUND ON TAPE 5 - PROGRAM STOP /)
59      FORMAT(/ 10X,25HTOTAL NUMBER OF POINTS = ,1I6 /)
60      FORMAT(/ 5X,12HHIRAP S/N 00,1I1,/ 5X,7HFILE = ,5HTAPE7,5X,5HT1 = ,
       *         1F10.2,9X,5HT2 = ,1F10.2 /)
         END
         SUBROUTINE TVCCAL1(TF,TC,VC,TEMP)
C
C       THIS ROUTINE COMPUTES TEMPERATURE FROM FINE AND COARSE
C       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)
C
         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 /
C
C        MAX TEMPERATURE RANGE = 5000 MILIVOLTS = 5 VOLTS
C        MAX TEMPERATURE COUNTS RANGE = 250
C        SCALE FACTOR, SF = 5000/(2 ** 8 - 1) = 5000/255
C        IN ACCORDANCE WITH INITIAL HIRAP CALIBRATIONS, BOTH THE
C        FINE AND COARSE TEMPERATURE COUNTS SHOULD BE REDUCED BY 3.
C
         SF = 5.0/250.0
C
C       COMPUTE VOLTAGES AND TEMPERATURE FOR X, Y, AND Z AXES
C
         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.A1) VC(I) = VC2
      TEMP(I) = A(I) + B(I) * VC(I)
1     CONTINUE
      RETURN
      END
      SUBROUTINE TVCCAL2(TF,TC,VC,TEMP)
C
C     THIS ROUTINE COMPUTES TEMPERATURE FROM FINE AND COURSE
C     TEMPERATURE COUNTS OF RECALIBRATED HIRAP S/N 002
C
C
      DIMENSION A(3),B(3),G(3),VM(10,3),TF(3),TC(3),VC(3),TEMP(3)
C
      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 /
C
C     MAX TEMPERATURE RANGE = 5000 MILLIVOLTS = 5 VOLTS
C     MAX TEMPERATURE COUNTS RANGE = 250
C     SCALE FACTOR, SF = 5000/(2 ** 8 - 1) = 5000/255
C     IN ACCORDANCE WITH INITIAL HIRAP CALIBRATIONS , BOTH THE
C     FINE  AND COURSE TEMPERATURE COUNTS SHOULD BE REDUCED BY 3.
C
      SF = 5.0/250.0
C
C     COMPUTE VOLTAGES AND TEMPERATURES FOR X, Y, AND Z AXES
C
      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
C
      RETURN
      END
2.0      5401.0   6057.0
```

```
      PROGRAM JTRATES(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7,
     *                TAPE8,TAPE9,TAPE10,TAPE1,TAPE2)
C
C     THIS IS THE HIRAP PROGRAM THAT CALCULATES ACCELERATIONS DUE TO
C     ANGULAR VELOCITIES AND THEN INTERPOLATES TO MATCH THE HIRAP
C     SCIENCE DATA INTERATION RATE. THE INDUCED ACCELERATIONS ARE
C     THEN ALGEBRAICALLY SUBTRACTED FROM THE RAW MICRO-G FILE.
C     THE CG CORRECTED MICRO-GS ARE PLOTTED IF SO DESIRED.
C
C     WRITTEN BY: JOSEPH S. ROWLEY, JUNE 1983
C     AND MODIFIED BY HIRAP PROJECT GROUP, SEPT 1986
C
C
C
C
      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)
C
C
C
      REAL NEWTIM,NEWX,NEWY,NEWZ
C
C
C     THE FOLLOWING IS A LIST OF INPUTS TO THIS PROGRAM
C
C
C     IFLT              STS FLIGHT NUMBER
C
C     IPLTMG = 1        TO PLOT MICRO-GS ON +/- 1000 SCALE
C            = 0        NO PLOTTING
C
C     T1                START TIME FOR DATA CALCULATION (SEC)
C
C     T2                STOP TIME FOR DATA CALCULATION
C
C     DAY               DAY OF DESCENT
C
C     XEPOCH            START TIME OF RATE DATA (XBET FILE)
C
C     ALTSTRT           START ALT FOR MICRO-G PLOT  (KM)
C
C     ALTSTOP           STOP ALT FOR PLOT  (KM)
C
C     XLENG             LENGTH OF X-AXIS PLOT GRID (IN)
C
C     XCG               VEHICLE CENTER OF GRAVITY LOCATION IN THE ORBITER
C     YCG               STRUCTURAL REFERENCE SYSTEM (IN)
C     ZCG
C
C
C     XMOUNT            HIRAP LOCATION ON ORBITER (IN)
C     YMOUNT
```

```
C     ZMOUNT
C
C     YLENG          LENGTH OF Y-AXIS PLOT GRID
C
C
C
C     TAPES USED IN THIS CODE:
C
C     INPUT:
C
C        TAPE 7      UNCORRECTED MICRO-GS (T,A,X,Z,Y)
C
C        TAPE 9      XBET FILE
C
C     OUTPUT:
C
C        TAPE 8      CORRECTED MICRO-GS (T,A,X,Z)
C
      PRINT 800
800   FORMAT(1H1,4X,'RATES STARTED')
C
C
C
C
      READ(5,810) IFLT,IPLTMG
      READ(5,811) ALTSTRT,ALTSTOP,XLENG,YLENG
      READ(5,811) T1,T2,XEPOCH,DAY
      READ(5,811,END=820) XCG,YCG,ZCG
      DELALT = (ALTSTRT - ALTSTOP)
      FLT = IFLT
      SN= 1.0
      IF(FLT.EQ.9.OR.FLT.EQ.26.OR.FLT.EQ.30) SN = 2.0
C
820   IF(EOF(5).NE.0) THEN
         WRITE(6,821)
         GO TO 999
      ELSE
         WRITE(6,822) IFLT,IPLTMG
         WRITE(6,823) ALTSTRT,ALTSTOP
         WRITE(6,824) T1,T2,XEPOCH,DAY
         WRITE(6,825) XCG,YCG,ZCG
      ENDIF
C
810   FORMAT(2I10)
811   FORMAT(4F10.0)
821   FORMAT(// 5X,34HEOF FOUND ON TAPE 5 - PROGRAM STOP)
822   FORMAT(// 5X,10HFLT = STS ,1I2,5X,9HIPLTMG = ,1I2)
823   FORMAT(/ 5X,10HALTSTRT = ,1F10.2,5X,10HALTSTOP = ,1F10.2)
824   FORMAT(/ 5X,5HT1 = ,1F10.2,/ 5X,5HT2 = ,1F10.2,
     *       / 5X,13HXBET EPOCH = ,1F10.2,/ 5X,6HDAY = ,1F4.0)
825   FORMAT(/ 5X,6HXCG = ,1F10.2,/ 5X,6HYCG = ,1F10.2,
     *       / 5X,6HZCG = ,1F10.2 //)
C
C
```

```
C
C
      DEGRAD = 180./3.1415927
      IF(FLT.EQ.32) THEN
         KMAX = 350
      ELSE
         KMAX = 350
      ENDIF
      KCNT = KMAX - 1
C
C
C
C
C     USE EXACT HIRAP LOCATION AND CG LOCATION AT ENTRY INTERFACE
C     TO OBTAIN THE MOMENT ARMS IN FEET.
C
C     X-AXIS ACCELEROMETER
C
      XMOUNT(1) = 1216.97
      YMOUNT(1) =  - 5.50
      ZMOUNT(1) =  298.55
C
C
C     Y-AXIS ACCELEROMETER
C
      XMOUNT(2) = 1218.27
      YMOUNT(2) =  - 5.50
      ZMOUNT(2) =  298.55
C
C
C     Z-AXIS ACCELEROMETER
C
      XMOUNT(3) = 1218.27
      YMOUNT(3) =  - 5.50
      ZMOUNT(3) =  296.12
C
C
      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),I,YBAR(I),I,ZBAR(I)
111   CONTINUE
C
C
C
826   FORMAT(// 5X,5HXBAR(,1I1,4H) = ,1F10.2,
     *        / 5X,5HYBAR(,1I1,4H) = ,1F10.2,
     *        / 5X,5HZBAR(,1I1,4H) = ,1F10.2 //)
C     READ TAPE HEADERS
C
C
      IFILE = 9
      READ(9,END=600) ISN,NCHAN,(NAMES(I),I=1,NCHAN),
```

```
     .                  (IUNITS(I),I=1,NCHAN),(HDR(I),I=1,8)
      WRITE(6,887) IFILE
      WRITE(6,888) ISN,NCHAN,HDR
C
C
600   IF(EOF(9).NE.0)  THEN
      WRITE(6,802) IFILE
802   FORMAT (// 5X,38HEOF FOUND WHEN READING HEADER ON FILE ,1I2)
      GO TO 999
      ENDIF
C
C
C
C
      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)
C
C
887   FORMAT(/// 5X,5HTAPE ,1I2)
888   FORMAT(//5X,6HISN = ,1I5,5X,8HNCHAN = ,1I5, / 5X,
     *           15HFOUND HEADER : ,8A10 /)
889   FORMAT(4(2X,A10))
890   FORMAT(4(2X,A10) //)
C
C
601   IF(EOF(7).NE.0)  THEN
      WRITE(6,802) IFILE
      GO TO 999
      ENDIF
C
C
C
C
C
      WRITE(8)  ISN7,NCHAN7,(NAMES7(I),I=1,NCHAN7),
     *          (IUNITS7(I),I=1,NCHAN7),(HDR7(I),I=1,8)
C
C
C     READ ANGULAR VELOCITIES AND ACCELERATIONS
C
40    READ(9,END=450) (DATA(J),J=1,NCHAN)
450   IF(EOF(9).NE.0) THEN
      PRINT 465,TIME
465   FORMAT (// 3X,33HEOF FOUND ON TAPE 9 AFTER TIME = ,1F12.3 //)
      GO TO 200
      ENDIF
C
```

```
C
C
        TIME=DATA(1) + XEPOCH
        IF(TIME.LT.T1) GO TO 40
C
        P    =DATA(49)
        Q    =DATA(50)
        R    =DATA(51)
        PDOT=0
        QDOT=0
        RDOT=0
C
C
C
C     CONVERT P,Q,R TO RAD/SEC IF NEEDED
C
      P=P/DEGRAD
      Q=Q/DEGRAD
      R=R/DEGRAD
C
C
C
C
      ANGACLX = XBAR(1)  *  (Q * Q + R * R)  - YBAR(1)  *  (P * Q - RDOT)  -
     .          ZBAR(1)  *  (P * R + QDOT)
C
      ANGACLY =-XBAR(2)  *  (P * Q + RDOT )  + YBAR(2)  *  (P * P + R * R)  -
     .          ZBAR(2)  *  (Q * R - PDOT)
C
      ANGACLZ =-XBAR(3)  *  (P * R - QDOT )  - YBAR(3)  *  (Q * R + PDOT)  +
     .          ZBAR(3)  *  (P * P + Q * Q)
C
C
C     CONVERT FT/SEC2 TO MICRO-GS
C
C
      DX = ANGACLX / 32.174 * 1E06
      DY = ANGACLY / 32.174 * 1E06
      DZ = ANGACLZ / 32.174 * 1E06
C
C
C     WRITE DATA TO TAPE10
C
      WRITE(10) TIME,DX,DZ,DY
C
C
      IF(TIME.GT.T2) GO TO 200
      GO TO 40
C
200   CONTINUE
      REWIND 10
      TIMSCI = 0

      NEWTIM=NEWX=NEWY=NEWZ=0
```

```
C
C
C
C
65     READ(10,END=66) TIME,DX,DZ,DY
66     IF(EOF(10).NE.0) THEN
        PRINT  67,TIME
67     FORMAT (// 5X,34HEOF FOUND ON TAPE 10 AFTER TIME = ,1F12.3 //)
        GO TO 900
        ENDIF
C
C
          OLDX   = NEWX
          OLDY   = NEWY
          OLDZ   = NEWZ
          OLDTIM = NEWTIM
          NEWX   = DX
          NEWY   = DY
          NEWZ   = DZ
          NEWTIM = TIME
C
        IF((TIMSCI - OLDTIM).GT.0.995) GO TO 65
C
C
50     READ(7,END=51) (DATA7(J),J=1,NCHAN7)
C
C
51     IF(EOF(7).NE.0) THEN
        PRINT 466,DATA7(1)
466    FORMAT (// 5X,33HEOF FOUND ON TAPE 7 AFTER TIME = ,1F12.3 //)
        GO TO 900
        ENDIF
C
C
        TIMSCI = DATA7(1)
        ALT = DATA7(2)
        XMG = DATA7(3)
        ZMG = DATA7(5)
C
C
        IF(TIMSCI.LT.OLDTIM) GO TO 50
        IF((TIMSCI - OLDTIM).GT.0.995) GO TO 65
C
C
C      STRAIGHT LINE INTERPOLATION TO MATCH HIRAP DATA RATE
C
C
        XANGA = ((TIMSCI - OLDTIM) / (NEWTIM - OLDTIM))
     .          * (NEWX - OLDX) + OLDX
C
C      YANGA = ((TIMSCI - OLDTIM) / (NEWTIM - OLDTIM))
C    .          * (NEWY - OLDY) + OLDY
C
        ZANGA = ((TIMSCI - OLDTIM) / (NEWTIM - OLDTIM))
```

```
      .          * (NEWZ - OLDZ) + OLDZ
C
C
C
C
C     CORRECT MICRO-GS TO VEHICLE CG
C
       XMG = XMG + XANGA
       ZMG = ZMG + ZANGA
C
C
       DX = -XANGA
       DZ = -ZANGA
C
C     WRITE CALCULATED DATA TO TAPE8
C
       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)
C
C
       IF(TIMSCI.GT.T2) GO TO 900
       GO TO 50
C
C
  900 CONTINUE
C
       IF(IPLTMG.EQ.1) THEN
          CALL PSEUDO
C
C
          REWIND(1)
C
C        PLOT X-AXIS INDUCED ACCELERATIONS
C
          IY = 1
          CALL CALPLT(2.0,2.0,-3)
          CALL DRAW(ALTSTRT,ALTSTOP,IY,FLT,SN,DAY,XLENG,YLENG)
128      READ(1,END=129) TIME,ALT,DX,DZ
          ALT = ALT/1000.
          IF(ALT.GT.ALTSTRT) GO TO 128
          TPLOT = (ALTSTRT - ALT) * (XLENG/DELALT)
          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
```

```
C
130       CONTINUE
           GO TO 133
           REWIND(1)
C
C         PLOT Y-AXIS INDUCED ACCELERATIONS
C
           IY = 2
           CALL NFRAME
           CALL CALPLT(2.0,2.0,-3)
           CALL DRAW(ALTSTRT,ALTSTOP,IY,FLT,SN,DAY,XLENG,YLENG)
131       READ(1,END=132) TIME,ALT,DX,DZ
           ALT = ALT/1000.
           IF(ALT.GT.ALTSTRT) GO TO 131
           TPLOT = (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
132       IF(EOF(1).NE.0.OR.TIME.GT.T2) GO TO 133
           GO TO 131
C
133       CONTINUE
           REWIND(1)
C
C         PLOT Z-AXIS INDUCED ACCELERATIONS
C
           IY = 3
           CALL NFRAME
           CALL CALPLT(2.0,2.0,-3)
           CALL DRAW(ALTSTRT,ALTSTOP,IY,FLT,SN,DAY,XLENG,YLENG)
134       READ(1,END=135) TIME,ALT,DX,DZ
           ALT = ALT/1000.
           IF(ALT.GT.ALTSTRT) GO TO 134
           TPLOT = (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
135       IF(EOF(1).NE.0.OR.TIME.GT.T2) GO TO 136
           GO TO 134
C
136       CONTINUE
           GO TO 36
           REWIND(1)
           READ(1)
C
C         PLOT X-AXIS MICROGS
C
           IY = 1
           CALL NFRAME
           CALL CALPLT(2.0,2.0,-3)
           CALL DRAW1(ALTSTRT,ALTSTOP,IY,FLT,SN,DAY,XLENG,YLENG)
28        READ(1,END=29) TIME,XMG,ZMG,YMG
```

```
         IF(ALT.GT.ALTSTRT) GO TO 28
         TPLOT = (TIME - ALTSTRT) * (XLENG/DELALT)
         YPLOT = (XMG + 1000.0) * (YLENG/2000.0)
         IF(YPLOT.GT.0.AND.YPLOT.LT.YLENG) THEN
            CALL PNTPLT(TPLOT,YPLOT,21,1)
         ENDIF
29       IF(EOF(1).NE.0.OR.TIME.GT.T2) GO TO 30
         GO TO 28
C
30       CONTINUE
         REWIND(1)
         READ(1)
C
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)
31       READ(1,END=32) TIME,XMG,ZMG,YMG
         IF(ALT.GT.ALTSTRT) GO TO 31
         TPLOT = (TIME - ALTSTRT) * (XLENG/DELALT)
         YPLOT = (YMG + 1000.0) * (YLENG/2000.0)
         IF(YPLOT.GT.0.AND.YPLOT.LT.YLENG) THEN
            CALL PNTPLT(TPLOT,YPLOT,21,1)
         ENDIF
32       IF(EOF(1).NE.0.OR.TIME.GT.T2) GO TO 33
         GO TO 31
C
33       CONTINUE
         REWIND(1)
         READ(1)
C
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)
34       READ(1,END=35) TIME,XMG,ZMG
         IF(ALT.GT.ALTSTRT) GO TO 34
         TPLOT = (TIME - ALTSTRT) * (XLENG/DELALT)
         YPLOT = (ZMG + 1000.0) * (YLENG/2000.0)
         IF(YPLOT.GT.0.AND.YPLOT.LT.YLENG) THEN
            CALL PNTPLT(TPLOT,YPLOT,21,1)
         ENDIF
35       IF(EOF(1).NE.0.OR.TIME.GT.T2) GO TO 36
         GO TO 34
C
36       CONTINUE
         CALL CALPLT(0.0,0.0,999)
      ENDIF
999   CONTINUE
      PRINT 801
```

```
801     FORMAT(// 5X,'RATES COMPLETED')
        STOP
        END
        SUBROUTINE DRAW(ALTSTRT,ALTSTOP,IY,FLT,SN,DAY,XLENG,YLENG)
C
C       THIS ROUTINE DRAWS GRID FOR HIRAP ACCELEROMETER VS TIME PLOTS
C
C       IY = 1 FOR X AXIS
C          = 2 FOR Y AXIS
C          = 3 FOR Z AXIS
C
        DIMENSION TITLE2(4),TITLE3(3),YCHAR(3),LABEL1(6),
       *          LABEL2(2)
C
        DATA TITLE2 / 10HHIRAP S/N ,10H00  ACCELE,10HROMETER DA,
       *              10HTA - STS   /
        DATA TITLE3 / 10HINDUCED AC,10HCELERATION,10HS                /
        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 : CG1,10H3II        /
C
        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)
C
C       DRAW TICK MARKS ON X AXIS
C
        NX = DELALT/2
        YBAR = 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,YBAR,3)
        CALL CALPLT(XBAR,DELY,2)
1       CONTINUE
C
C       DRAW TICK MARKS ON Y AXIS
C
        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)
3     CONTINUE
C
C     LABEL X AXIS
C
      DELX = DELX * 10
      YBAR = -0.25
      XVAL = ALTSTRT + 20.0
      IEND = NX/10 + 1
      DO 5 I = 1,IEND
      XVAL = XVAL - 20.0
      XST = - 0.20
      XBAR = XST + DELX * (I-1)
      CALL NUMBER(XBAR,YBAR,0.10,XVAL,0.0,-1)
5     CONTINUE
C
C     LABEL Y AXIS
C
      YST = -0.05
      DELY = DELY * 2
      IEND = NY/2 + 1
      DO 6 I = 1,IEND
      YBAR = YST + DELY * (I-1)
      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)
6     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)
C
C     THIS ROUTINE DRAWS GRID FOR HIRAP ACCELEROMETER VS TIME PLOTS
C
C     IY = 1 FOR X AXIS
C        = 2 FOR Y AXIS
C        = 3 FOR Z AXIS
C
      DIMENSION TITLE2(4),TITLE3(3),YCHAR(3),LABEL1(6),
     *          LABEL2(2)
C
      DATA TITLE2 / 10HHIRAP S/N ,10H00  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 : CG1,10H3II        /
C
      DELALT = (ALTSTRT - ALTSTOP)
      CALL CALPLT(0.0,YLENG,2)
      CALL CALPLT(XLENG,0.0,3)
      CALL CALPLT(0.0,0.0,2)
C
C     DRAW TICK MARKS ON X AXIS
C
      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
C
```

```
C     DRAW TICK MARKS ON Y AXIS
C
       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)
3     CONTINUE
C
C     LABEL X AXIS
C
       DELX = DELX * 5
       YBAR = -0.25
       XVAL = ALTSTRT - 500.0
       IEND = NX/5 + 1
       DO 5 I = 1,IEND
       XVAL = XVAL + 500.0
       XST = - 0.20
       XBAR = XST + DELX * (I-1)
       CALL NUMBER(XBAR,YBAR,0.10,XVAL,0.0,-1)
5     CONTINUE
C
C     LABEL Y AXIS
C
       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.1000) XST = -0.49
       IF(YVAL.GE.10000) XST = -0.59
       IF(YVAL.LE.-10000) XST = -0.65
       CALL NUMBER(XST,YBAR,0.10,YVAL,0.0,-1)
6     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)
C
C
C
C     THIS PROGRAM FINDS THE ZERO REFERENCE VALUES FOR THE 44 CHANNELS
C     CONTAINED ON THE REACTION CONTROL SYSTEM THRUST CHAMBERS DATA TAPE.
C     THESE VALUES ARE USED AS INPUTS TO THE PROGRAM THRUST.
C
C     WRITTEN BY: ROBIN WADDELL   FEBRUARY 1984
C     MODIFIED BY: JOSEPH S. ROWLEY  MARCH 1985
C
C
C
      DIMENSION DATAPT(100),DIFVAL(100,100),SAMVAL(100,100)
C
      INTEGER I,L,NAMES(100),IUNITS(100),HDR(8),NDIFPTS(100),FLAG,
     .FAKE(100)
C
C
C
C     THE FOLLOWING ARE INPUTS TO THIS PROGRAM
C
C     STOP          X-AXIS SATURATION TIME
C
C
C
C
C
C     INPUTS GO HERE
C
      STOP=69553.000
C
C
C
      FLAG=0
      DATAPT(1) = 0.0
C
      DO 10 I=1,50
      NDIFPTS(I)=0
   10 CONTINUE
C
C
C
      PRINT 105
  105 FORMAT(1H1,"ZPRESS PROCEDURE STARTED")
      PRINT 2,STOP
    2 FORMAT(1X,"STOP=",F6.0)
C
C
C
C
C     READ HEADER AND PRINT NAMES AND IUNITS
C
```

```
C
 110   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
C
       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      READ PRESSURES FOR ALL CHANNELS
C
C
  40   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
C
C
C
C
C
C      IF FIRST PASS INCLUDE ALL VAULES
C
       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
```

```
      GO TO 40
      ENDIF
C
C
C
C
C     ERROR CODES (1E+37 OR GREATER) AND PRESSURES GREATER
C     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
C
      IF(J .EQ. NDIFPTS(I)) THEN
      NDIFPTS(I)=NDIFPTS(I)+1
      DIFVAL(I,NDIFPTS(I))=DATAPT(I)
      SAMVAL(I,NDIFPTS(I))=1
      ENDIF
C
      ENDIF
 100  CONTINUE
 102  CONTINUE
C
C
      GO TO 40
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)
C
C     THIS ROUTINE FINDS FIRING TIMES FOR ORBITER RCS THRUSTERS
C
C        NE1020 WITH TCOD924,THRUS924
C     WRITTEN BY: JOSEPH S. ROWLEY, JUNE 1983
C
C
C
C
      REAL  STOP,PRESS(100),ZERO(50),START,LAG,TCLAG
C
C
      INTEGER NAMES1(100),IUNITS1(100),HDR(8),NCHAN1,
               NAMES(100),IUNITS(100),HDR1(8),NCHAN
C
C
C
C
C
C      THE FOLLOWING IS A LIST OF INPUTS TO THIS PROGRAM
C
C
C      START          START TIME FOR DATA CALCULATION
C
C      STOP           STOP TIME FOR DATA CALCULATION
C
C      LAG            LAG TIME REQUIRED TO ENLARGE FIRING INTERVAL
C
C     TCLAG          TIME THAT LAG VALUE IS CHANGED
C
C      ZERO           ZERO FOR EACH CHAMBER
C
C
C
      PRINT 800
  800 FORMAT(/,/,1X,'THRUST STARTED')
C
C
C
      DO 22 I=1,50
      ZERO(I) = 0
   22 CONTINUE
C
C
C     INPUTS GO HERE
C
      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
C
C
C
C
C
C      WRITE TAPE HEADER ON TAPE8 AND TAPE10
C
      ISN1=1
      NCHAN1=2
      NAMES1(1)=4HTIME
      NAMES1(2)=10HRCS FIRING
      IUNITS1(1)= 7HSECONDS
      IUNITS1(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
C
C
      WRITE(8)   ISN1,NCHAN1,(NAMES1(I),I=1,NCHAN1),
     .                 (IUNITS1(I),I=1,NCHAN1),(HDR1(I),I=1,8),
     .          (HDR1(I),I=1,8)
C**   PRINT*,ISN1,NCHAN1,(NAMES1(I),I=1,NCHAN1),
C**  .        (IUNITS1(I),I=1,NCHAN1)
C**99  FORMAT(4X,5HISN1=,I5,2X,7HNCHAN1=,I3,2X,2A7,5X,8A10)
C
C
      NAMES1(2)=8HRCS CODE
      WRITE(10)  ISN1,NCHAN1,(NAMES1(I),I=1,NCHAN1),
     .                 (IUNITS1(I),I=1,NCHAN1),(HDR1(I),I=1,8)
C
C
C     READ TAPE HEADER
C
C
 101  READ(9,END=600) ISN,NCHAN,(NAMES(I),I=1,NCHAN),
     .                 (IUNITS(I),I=1,NCHAN),(HDR(I),I=1,8)
C
C
  600 IF (EOF(9) .NE. 0)  THEN
      PRINT 802
  802 FORMAT (1X,"EOF FOUND WHEN ATTEMPTING TO READ HEADER")
      GO TO 900
```

```
      END IF
C
C
C
C
   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,38HSTOP  TIME LESS THAN LAST TIME FOUND =,F16.9)
      GO TO 900
      END IF
C
      IF (START .GT. TIME) GO TO 40
C
C
      DO 57 I=3,46
      IF (PRESS(I) .GT. 1E+18 .OR. PRESS(I) .LT. ZERO(I)) GO TO 57
C
      IF (I .LE. 18) THEN
                                         CODE=1
      ELSE IF (I .LE. 32) THEN
                                         CODE=2
      ELSE
                                         CODE=3
      END IF
C
      WRITE(10) TIME,CODE
C
   57 CONTINUE
C
C
C
C
C
      IF (STOP .LE. TIME) GO TO 300
      GO TO 40
C
  300 PRINT 245,TIME
  245 FORMAT (1X,4X,46HSTOP TIME DESIRED IS LESS THAN LAST TIME READ=,
     .    F16.9)
C
      LTIME=TIME
C
      REWIND(10)
C
      PTIME=0
      LFSTOP=0
C
```

```
   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
C
C
C
   30 IF (TIME - PTIME .GT. 0.041) THEN
C
         IF (LFSTOP .EQ. 0) THEN
               STIME=TIME - 0.04
               LFSTOP=1
C
         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
C
         END IF
C
      END IF
C
C
      PTIME=TIME
C
      IF (LTIME .GT. TIME) GO TO 60
C
C
  900 CONTINUE
      PRINT 801
  801 FORMAT(1X,'THRUST COMPLETED')
      STOP
      END
```

```
      PROGRAM THFIT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     . TAPE9,TAPE10,TAPE11,TAPE12,TAPE13)
C
C     THIS ROUTINE REMOVES ODD DATA SPIKES THAT ARE LEFT AFTER OTHER
C     THRUST REMOVING EFFORTS.
C
C     WRITTEN BY: JOSEPH S. ROWLEY,  JAN 1985
C
C
C
C
      DIMENSION X(10000),Y(10000),
     .          NAMES(10),HDR(8),IUNITS(10),DATAPT(10),BUF(65)
C
      REAL LOWER
C
C
C
C
C      THE FOLLOWING IS A LIST OF INPUTS TO THIS PROGRAM
C
C
C
C      CHANIV        CHANNEL TO BE FIT AS INDEPENDENT VARIABLE
C
C      CHANDV        CHANNEL TO BE FIT AS DEPENDENT VARIABLE
C
C      START         START TIME
C
C      STOP          STOP TIME
C
C      SIZE          NUMBER OF POINTS FOR LEAST SQUARES FIT
C
C      GAPSTR        TIME FOR START OF APU TRANSITION - Z AXIS ONLY
C
C      GAPSTP        TIME FOR STOP OF APU TRANSITION - Z AXIS ONLY
C
C      SATTIME       SATURATION TIME FOR DEPENDENT CHANNEL
C
C      SERSKIP       NUMBER OF SERIALS TO SKIP TO LOCATE DATA
C
C      BURNSTR       START TIME OF DEORBIT BURN
C
C      BURNSTP       STOP TIME OF DEORBIT BURN
C
C
C
C
      WRITE(6,777)
777   FORMAT(1H1,13HTHFIT STARTED //)
C
C
C     INPUTS GO HERE
C
```

```
      CHANIV=1
      CHANDV=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
C
C
C
C
      PRINT*,'CHANIV= ',CHANIV
      PRINT*,'CHANDV= ',CHANDV
      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
C
C
C
C
C     SKIP TO DESIRED SERIAL IF NEEDED
C
      IF(SERSKIP .NE. 0) THEN
C
C     GENERATE FILE ENVIRONMENT TABLE FOR SPECIFIED DIRECT ACCESS FILE
C
      NAME=L"TAPE9"
      CALL GENFET(FET,NAME,BUF,65)
C
      CALL SKIPR (FET,SERSKIP)
      END IF
C
C
C
C
C
C
C
```

```
C
C     READ,WRITE,AND PRINT TAPE HEADER
C
C
      READ(9,END=600)   ISN,NCHAN,(NAMES(I),I=1,NCHAN),(IUNITS(I),I=1,
     .         NCHAN),(HDR(I),I=1,8)
C
  600 IF (EOF(9) .NE. 0)  THEN
      PRINT 802
  802 FORMAT (1X,"EOF FOUND WHEN ATTEMPTING TO READ HEADER")
      GO TO 400
      END IF
C
C
      WRITE(12)  ISN,NCHAN,(NAMES(I),I=1,NCHAN),(IUNITS(I),I=1,
     .            NCHAN),(HDR(I),I=1,8)
C
      PRINT 104, ISN,HDR
  104 FORMAT(/,1X,4HFILE,I7,14H FOUND HEADER=,8A10)
      PRINT 303,  (NAMES(I),I=1,NCHAN)
      PRINT 303,  (IUNITS(I),I=1,NCHAN)
      PRINT 302
  303 FORMAT(1X,8(A10))
  302 FORMAT(/)
C
C
C
C
C
C     DETERMINE UPPER,LOWER,UPPER2,LOWER2 FOR DESIRED CHANNEL
C
C
      IF (CHANDV .EQ. 3) THEN
                          UPPER=45
                          LOWER=45
                          UPPER2=45
                          LOWER2=45
C
      ELSE IF (CHANDV .EQ. 4) THEN
C
                          UPPER=75
                          LOWER=75
                          UPPER2=225
                          LOWER2=225
      ELSE
C
                        PRINT*,'PROBLEM WITH CHANDV DEFINITION'
C
      END IF
C
C
C
C
C      CURVE FITTING STARTS HERE
```

```
C
C
C
C
      DATAPT(1)=0
      FAKE=0
      SIZE=SIZE1
   34 P=0
      SUMX=0
      SUMY=0
      SUMXSQ=0
      SUMXY=0
      REWIND (11)
C
C
   40 READ(9,END=500)  (DATAPT(J),J=1,NCHAN)
  500 IF(EOF(9) .NE. 0) THEN
      PRINT 165,DATAPT(1)
  165 FORMAT (1X,'EOF FOUND WHEN READING DATA ON TAPE9
     .        AFTER TIME= ',F16.9)
      GO TO 400
      END IF
C
C
      IF (DATAPT(1) .LT. START)  GO TO 40
      IF (DATAPT(1) .GE. STOP) THEN
      PRINT*,'STOP TIME FOUND AFTER READ AT LABLE 40=',DATAPT(1)
      PRINT*,'  '
      PRINT*,'  '
      GO TO 100
      END IF
C
      IF (DATAPT(1) .GT. CURVBK .AND. FAKE .NE. 1)  THEN
         SIZE=SIZE2
         FAKE=1
         GO TO 34
      END IF
C
C
C
      IF (DATAPT(1) .GE. BURNSTR .AND. DATAPT(1) .LE. BURNSTP) GO TO 40
C
C
      IF (DATAPT(1) .GE. GAPSTR .AND. DATAPT(1) .LE. GAPSTP
     .    .AND. CHANDV .EQ. 4) GO TO 40
C
C
C
      IF (P .EQ. 0) THEN
                  FIRSTX=DATAPT(CHANIV)
      END IF
      P=P+1
      IF (P .GT. SIZE) PRINT*,'PROBLEM WITH SIZE OF P'
C
```

```
C
      WRITE(11)  (DATAPT(J),J=1,NCHAN)
      WRITE(13)  (DATAPT(J),J=1,NCHAN)
C
C
      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)
C
C
C     PRINT*,'CHECK 3 ',SIZE,P
      IF (P .NE. SIZE) GO TO 40
C
      IF (DATAPT(1) .LE. SATTIME)  TIMLAS=DATAPT(1)
C     PRINT*,'CHECK 1'
C
C
      AVGX=SUMX/P
      AVGY=SUMY/P
      SLOPE=(SUMXY-P*AVGX*AVGY)/(SUMXSQ-P*AVGX*AVGX)
      YINT = AVGY - SLOPE * AVGX
C
C     PRINT*,'SUMX=',SUMX
C     PRINT*,'SUMY=',SUMY
C     PRINT*,'SLOPE=',SLOPE
C     PRINT*,'YINT=',YINT
C     PRINT*,'AVGX=',AVGX
C     PRINT*,'AVGY=',AVGY
C     PRINT*,'SUMXY=',SUMXY
C     PRINT*,'SUMXSQ=',SUMXSQ
C     PRINT*,'P=',P
C
C
C
C
C     CALCULATE DATA FROM ABOVE EQUATION
C
C
      REWIND (11)
C
      PASS =0
      PASS1=0
C
C
   30 READ(11,END=31)   TIME
   31 IF(EOF(11) .NE. 0) THEN
C     PRINT 32,TIME
C  32 FORMAT (1X,'EOF FOUND WHEN READING DATA ON TAPE11
C    .         AFTER 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
C
C
C
C
      YVALUE = SLOPE * ( TIME - FTIME ) + YINT
      DELT = TIME - FTIME
C
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
C
C
  100 CONTINUE
C
      PRINT*,'  '
      PRINT*,'  '
C
      PRINT*,'DATA SPIKE REMOVING STARTED'
C
C
      TIME=0
C
C
      REWIND (13)
      REWIND (10)
C
      TIME=0
C
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
C
```

```
C
C
      IF (DATAPT(1) .LT. START) GO TO 55
      IF (DATAPT(1) .GE. STOP) GO TO 400
C
C
      IF (DATAPT(1) .LT. TIME)  GO TO 55
      IF (DATAPT(1) .GT. TIME)  GO TO 56
      GO TO 57
C
C
C
C
C
   56 READ(10,END=556)  TIME,YVALUE
  556 IF(EOF(10) .NE. 0) THEN
      PRINT 557,TIME
  557 FORMAT (1X,'EOF FOUND WHEN READING DATA ON TAPE10 IN 2ND LOCATION
     .        AFTER TIME= ',F16.9)
      GO TO 400
      END IF
C
C
      IF (DATAPT(1) .LT. TIME)  GO TO 55
      IF (DATAPT(1) .GT. TIME)  GO TO 56
C
C
C
C
   57 IF ((DATAPT(1) .GE. TIMLAS) .AND. (DATAPT(1) .LE.
     .     SATTIME)) GO TO 55
C
C
C
      IF (TIME .NE. DATAPT(1)) THEN
            PRINT*,'PROBLEM WITH TIME MATCH AT TIME=', TIME,DATAPT(1)
            GO TO 400
      END IF
C
C
C
      IF (CHANDV .EQ. 4 .AND. DATAPT(1) .GE. GAPSTP) THEN
                                                 UPPER=UPPER2
                                                 LOWER=LOWER2
      END IF
C
C
C
C
C
C
      AUPPER= YVALUE + UPPER
      ALOWER= YVALUE - LOWER
C
```

```
C
C
C     ELIMINATE SPIKE IF NECCESSARY
C
C     IF(DATAPT(2).LT.87000.AND.DATAPT(3).GT.-4000.) GO TO 55
C     IF(DATAPT(2).LT.86000.AND.DATAPT(3).GT.-4700.) GO TO 55
C
C
C
C
      IF (DATAPT(CHANDV) .GT. AUPPER .OR. DATAPT(CHANDV) .LT.
     .   ALOWER) GO TO 55

      WRITE(12) (DATAPT(I),I=1,NCHAN)

      GO TO 55
C
C
C
  400 PRINT*,' THFIT COMPLETED'
C
      STOP
      END
```

```
      PROGRAM RCOMBIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7,
     .               TAPE8,TAPE9,TAPE10,TAPE1)
C
C     THIS ROUTINE WAS WRITTEN TO COMPLEMENT JSRSMTH (THE HIRAP
C     DATA SMOOTHING ROUTINE) WHICH IS LIMITED TO SMOOTHING ONLY
C     ONE CHANNEL AT A TIME. THE PURPOSE OF THIS ROUTINE IS TO
C     COMBINE THREE FILES (X,Y,Z-AXIS) INTO ONE FOR ALL THREE.
C
C     WRITTEN BY: JOSEPH S. ROWLEY,  SEPT 1983
C     MODIFIED BY: ROBIN WADDELL, JAN 1984
C
C
C
C
      DIMENSION XDATA(10),YDATA(10),ZDATA(10)
C
C
      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
C
C
C     THE FOLLOWING IS A LIST OF INPUTS TO THIS PROGRAM
C
C
C     COUNT            NUMBER OF CHANNELS TO BE COMBINED
C
C     START            START TIME FOR COMBINING FILES
C
C     STOP             STOP TIME FOR COMBINING FILES
C
C
C
C
      PRINT 800
  800 FORMAT(/,/,1X,'COMBINE STARTED')
C
C
C
C
C
C
C     INPUTS GO HERE
C
      COUNT=2
      START=0
      STOP=100000.
C
C
C
C
C
C
C
```

```
C
C
C
C     READ TAPE HEADERS
C
C
C
      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)
C
  102 IF (EOF(7) .NE. 0)  THEN
      PRINT 103
  103 FORMAT (1X,"EOF FOUND WHEN READING TAPE7 HEADER")
      GO TO 900
      END IF
C
      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
  105 FORMAT(1X,8(A10))
  106 FORMAT(/)
      END IF
C
      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)
C
  802 IF (EOF(8) .NE. 0)  THEN
      PRINT 803
  803 FORMAT (8X,"EOF FOUND WHEN READING TAPE8 HEADER")
      GO TO 900
      END IF
C
      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
  805 FORMAT(1X,8(A10))
  806 FORMAT(/)
C
      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)
C
  902 IF (EOF(9) .NE. 0)  THEN
      PRINT 903
  903 FORMAT (1X,"EOF FOUND WHEN READING TAPE9 HEADER")
      GO TO 900
```

```
      END IF
C
      PRINT 904,ISN9,HDR9
  904 FORMAT(/,1X,'FILE ',I7,' FOUND HEADER=',8A10)
      PRINT 905,  (NAMES9(I),I=1,NCHAN9)
      PRINT 905,  (IUNITS9(I),I=1,NCHAN9)
      PRINT 906
  905 FORMAT(1X,8(A10))
  906 FORMAT(/)
      END IF
C
       NCHAN7=COUNT+1
      WRITE(10)  ISN7,NCHAN7,(NAMES7(I),I=1,NCHAN7),
     .           (IUNITS7(I),I=1,NCHAN7),(HDR7(I),I=1,8)
C
      NCHAN7=COUNT+1
C      MOVE TO DESIRED START AND COMBINE FILES UNITL STOP
C      OR END OF FILE
C
      XDATA(1)=0
      YDATA(1)=0
      ZDATA(1)=0

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

C
C
      IF(XDATA(1).LT.YDATA(1))GO TO 40
      IF(XDATA(1).GT.YDATA(1))GO TO 60
      GO TO 999
```

```
 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
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 PRSENT 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,
     *   TAPE10=2006,TAPE9=2006,TAPE13=2006)
      INTEGER OUTTAPE
C
      DIMENSION TIM(15000),TEMPER(15000,6),TEMP(6)
C
C        INPUT BLOCK: CHANGE FOR EACH FLIGHT
C
C
      T1 = 65500.
      T2 = 68100.
C
      I = 0
C
C         END INPUT BLOCK
C
C
C
C
      INTAPE = 8
      OUTTAPE = 10
      NFILE = 9
      READ (NFILE)
   10 I = I+1
      READ (NFILE,END=20) TIM(I),(TEMPER(I,J),J=1,6)
      GO TO 10
   20 CONTINUE
      NT = I-1
      FINTIM = TIM(NT)
C
C
      REWIND (INTAPE)
      REWIND (OUTTAPE)
   30 READ (INTAPE,END=70) TYME,XCOUNT,ZCOUNT,YCOUNT
C
C
      IF (TYME.LT.T1) GO TO 30
      IF (TYME.GT.T2) GO TO 70
C
      IF (TYME.GT.FINTIM) THEN
         DO 40 NN = 1, 6
            TEMP(NN) = 9999.
   40    CONTINUE
C
      ELSE
         CALL IUNI (15000,NT,TIM,6,TEMPER,1,TYME,TEMP,-1,IE)
C
      ENDIF
C
C     CONVERT FROM COUNTS TO UNITS OF MICROGS
C
      IF (IE.NE.0) THEN
         WRITE (6,50) IE,TYME
   50    FORMAT (2X,'INTERPOLATION ERROR ',I3,'AT TIME = ',1X,F12.4)
```

```
      ENDIF
C
C    SN 001 SCALE FACTOR VALUES: PRE-MOD
C
      SFX = -1.247237E-3
      SFY = 1.253821E-3
      SFZ = -1.246810E-3
C
      VX = -10.+(XCOUNT/16383.)*20.
      VY = -10.+(YCOUNT/16383.)*20.
      VZ = -10.+(ZCOUNT/16383.)*20.
C
C   CONVERSION TO MICROG'S WITH SCALE FACTOR
C
      XMG = VX/SFX
      YMG = VY/SFY
      ZMG = VZ/SFZ
C
C  WRITE NON-DETRENDED MICROG TO OUTPUT TAPE10
C
C
      WRITE (OUTTAPE) TYME,XMG,YMG,ZMG,(TEMP(K),K=1,6)
C
C
      IF (TYME.LT.(T1+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))
C
C
      ENDIF
      GO TO 30
 70 CONTINUE
C
 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)
C
C
      WRITE (6,90) T1,T2
 90 FORMAT (2X,'REQUESTED TIME PROCESSED T1 - T2 =',2(2X,F11.4))
      STOP
      END
```

```
      PROGRAM FILLSQR(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     *   TAPE1,TAPE2)
C
      DIMENSION X(2000),Y(2000),Z(2000),NAMES(10),IUNITS(10),
     *HDR(8),
     *A(3),B(3),C(3),SDEV(3),T(2000),ACC(2000,3),GAPSTRT(100),
     *GAPSTOP(100),NFIL(100)
C
      EQUIVALENCE (ACC(1,1),X(1)), (ACC(1,2),Z(1))
C
C     READ WINDOW SIZE
C
      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)
C
      DO 5 I = 1,100
           GAPSTRT(I) = 1.0E+06
           GAPSTOP(I) = 1.0E+06
           NFIL(I) = 0.0
5     CONTINUE
C
1     CONTINUE
C
      IF (EOF(1) .NE. 0) THEN
           WRITE(6,50)
           IFILE = 1
           GO TO 14
      ENDIF
C
      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
C     WRITE(6,56) ISN,NCHAN,(HDR(I),I=1,8)
C     WRITE(6,51) NAMES(1),NAMES(3)
C     WRITE(6,57) IUNITS(1),IUNITS(3)
      READ(5,52,END=2) TSTART,TLIMIT
      GO TO 3
2     CONTINUE
      TLIMIT = 0.0
      TSTART = 0.0
3     CONTINUE
      WRITE(6,55) TSTART,TLIMIT
      DELTIM = 1.0/FREQ * 5.0
      WRITE(6,58) DELTIM
      WRITE(6,59)
      NMED = NPTS/2
      IFILE = 0
      IGAP = 0
      IPTS = NPTS
      READ(1,END=1) T(1),(ACC(1,J),J=1,2)
```

```
      WRITE(2) T(1),(ACC(1,J),J=1,2)
      TFIRST = T(1)
C
4     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
C
      IF (TSTART .EQ. 0.0) THEN
            TSTART = T(1)
            WRITE(6,60) TSTART
      ENDIF
C
      I = 1
C
6     CONTINUE
      I = I + 1
      READ(1,END=1) T(I),(ACC(I,J),J=1,2)
      IF (I .EQ. 1) GO TO 6
C
      IF ((T(I) - T(I-1)) .LT. DELTIM) THEN
            IF (I .LT. IPTS) GO TO 6
            GO TO 7
      ELSE
            IGAP = IGAP + 1
            GAPSTRT(IGAP) = T(I-1)
            GAPSTOP(IGAP) = T(I)
            GAP = T(I) - T(I-1)
            NFIL(IGAP) = GAP * FREQ - 0.5
            IF (IGAP .EQ. 100) GO TO 7
            GO TO 6
      ENDIF
C
7     CONTINUE
      IPTS = I
      WRITE(6,62) IPTS,IGAP,T(1),IPTS,T(IPTS)
      DO 20 L = 1,IGAP
            WRITE(6,100) L,GAPSTRT(L),L,GAPSTOP(L),L,NFIL(L)
100   FORMAT(10X,8HGAPSTRT(,1I2,4H) = ,1F10.3,5X,8HGAPSTOP(,1I2,4H) = ,
     *1F10.3,5X,5HNFIL(,1I2,4H) = ,1I4)
20    CONTINUE
C
      IF (IGAP .GT. 0) THEN
            CALL POINTS(IPTS,T,ACC,A,B,C,SDEV,TSTART)
      ENDIF
C
      ISTOP = NMED
      IF (IPTS .LT. NPTS) ISTOP = IPTS/2
      IF (IFILE .EQ. 1) ISTOP = IPTS
      JCNT = 1
8     CONTINUE
C
      DO 9 I = 1,ISTOP
C
```

```
            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)
9      CONTINUE
C
       IF (IFILE .EQ. 1) GO TO 15
       ICNT = 0
       JCNT = JCNT - 1
C
       IF (IGAP .GT. JCNT) THEN
            NCNT = IGAP - JCNT
C
            DO 10 I = 1,NCNT
                 GAPSTRT(I) = GAPSTRT(I+JCNT)
                 GAPSTOP(I) = GAPSTOP(I+JCNT)
                 NFIL(I) = NFIL(I+JCNT)
                 ICNT = ICNT + 1
10          CONTINUE
C
       ENDIF
C
       IGAP = ICNT
C
       IF (JCNT .GT. 0) THEN
C
            DO 11 I = ICNT+1,JCNT
                 GAPSTRT(I) = 1.0E+06
                 GAPSTOP(I) = 1.0E+06
                 NFIL(I) = 0.0
11          CONTINUE
C
       ENDIF
C
       ICNT = 0
C
       DO 13 I = ISTOP+1,IPTS
            ICNT = ICNT + 1
            T(ICNT) = T(I)
C
            DO 12 J = 1,2
                 ACC(ICNT,J) = ACC(I,J)
12          CONTINUE
C
13     CONTINUE
C
       I = ICNT
       WRITE(6,101) T(1),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,
      *        15HFOUND 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,9HTSTART = ,1F10.3)
 61    FORMAT(// 5X,9HTFIRST = ,1F10.3,5X,8HTLAST = ,1F10.3)
 62    FORMAT(/10X,7HIPTS = ,1I5,5X,7HIGAP = ,1I4,5X,7HT(1) = ,1F10.3,
      *5X,2HT(,1I4,4H) = ,1F10.3)
       END
       SUBROUTINE BANDSQR(X,Y,NX,SDEV,XSQSUM,XSUM,XYSUM,YSUM,NTOT,
      *XCUBSUM,XFORSUM,XSQYSUM,A,B,C)
C
       DIMENSION X(2000,3),Y(2000,3),SDEV(3),XSQSUM(3),XSUM(3),NTOT(3),
      * 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)
```

```
                  XSQSUM(J) = XSQSUM(J) + X(I,J)**2
                  XSQYSUM(J) = XSQYSUM(J) + X(I,J)**2 * Y(I,J)
                  XYSUM(J) = XYSUM(J) + X(I,J) * Y(I,J)
                  YSUM(J) = YSUM(J) + Y(I,J)
               ENDIF
C
1          CONTINUE
C
       NTOT(J) = L
2      CONTINUE
C
C      WRITE(6,10)
C
C      DO 3 I = 1,2
C          WRITE(6,11) I,NX(I),I,XSUM(I),I,XSQSUM(I),I,XYSUM(I),I,
C     *                YSUM(I),I,XCUBSUM(I),I,XFORSUM(I),I,XSQYSUM(I),
C     *                I,NTOT(I)
C3     CONTINUE
C
10     FORMAT(/ 15X,14HBANDSQR OUTPUT )
11     FORMAT(10X,3HNX(,1I1,4H) = ,1I4,5X,5HXSUM(,1I1,4H) = ,1E12.5,5X,
      *7HXSQSUM(,1I1,4H) = ,1E12.5, / 10X,6HXYSUM(,1I1,4H) = ,1E12.5,5X,
      *5HYSUM(,1I1,4H) = ,1E12.5,5X,8HXCUBSUM(,1I1,4H) = ,1E12.5 / 10X,
      *8HXFORSUM(,1I1,4H) = ,1E12.5,5X,8HXSQYSUM(,1I1,4H) = ,1E12.5,5X,
      *5HNTOT(,1I1,4H) = ,1I6 / )
       RETURN
       END
       SUBROUTINE GAPFILL(NFILL,TIM1,TIM2,A,B,C,SDEV,FREQ,IDEV,TSTART)
C
       DIMENSION Y(3),A(3),B(3),C(3),SDEV(3),DIST(300),DNORM(300)
C
       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, .79389, .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,
```

```
     * .92647, .92785, .92922, .93056, .93189, .93319, .93448, .93574,
     * .93699, .93822, .93943, .94062, .94179, .94295, .94408, .94520,
     * .94630, .94738, .94845, .94950, .95053, .95154, .95254, .95352,
     * .95449, .95543, .95637, .95728, .95818, .95907, .95994, .96080,
     * .96164, .96246, .96327, .96407, .96485, .96562, .96638, .96712,
     * .96784, .96856, .96926, .96995, .97062, .97128, .97193, .97257,
     * .97320, .97381, .97441, .97500, .97558, .97615, .97670 /
      DATA (DNORM(I),I=201,300) / .97725,.97778,.97831,.97882,.97932,
     * .97982, .98030, .98077, .98124, .98169, .98214, .98257, .98300,
     * .98341, .98382, .98422, .98461, .98500, .98537, .98574, .98610,
     * .98645, .98679, .98713, .98745, .98778, .98809, .98840, .98870,
     * .98899, .98928, .98956, .98983, .99010, .99036, .99061, .99086,
     * .99111, .99134, .99158, .99180, .99202, .99224, .99245, .99266,
     * .99286, .99305, .99324, .99343, .99361, .99379, .99396, .99413,
     * .99430, .99446, .99461, .99477, .99492, .99506, .99520, .99534,
     * .99547, .99560, .99573, .99585, .99598, .99609, .99621, .99632,
     * .99643, .99653, .99664, .99674, .99683, .99693, .99702, .99711,
     * .99720, .99728, .99736, .99744, .99752, .99760, .99767, .99774,
     * .99781, .99788, .99795, .99801, .99807, .99813, .99819, .99825,
     * .99831, .99836, .99841, .99846, .99851, .99856, .99861 /
      DATA (DIST(I),I=1,150) /  .00, .01, .02, .03, .04, .05, .06,
     * .07, .08, .09, .10, .11, .12, .13, .14, .15, .16, .17, .18,
     * .19, .20, .21, .22, .23, .24, .25, .26, .27, .28, .29, .30,
     * .31, .32, .33, .34, .35, .36, .37, .38, .39, .40, .41, .42,
     * .43, .44, .45, .46, .47, .48, .49, .50, .51, .52, .53, .54,
     * .55, .56, .57, .58, .59, .60, .61, .62, .63, .64, .65, .66,
     * .67, .68, .69, .70, .71, .72, .73, .74, .75, .76, .77, .78,
     * .79, .80, .81, .82, .83, .84, .85, .86, .87, .88, .89, .90,
     * .91, .92, .93, .94, .95, .96, .97, .98, .99,1.00,1.01,1.02,
     *1.03,1.04,1.05,1.06,1.07,1.08,1.09,1.10,1.11,1.12,1.13,1.14,
     *1.15,1.16,1.17,1.18,1.19,1.20,1.21,1.22,1.23,1.24,1.25,1.26,
     *1.27,1.28,1.29,1.30,1.31,1.32,1.33,1.34,1.35,1.36,1.37,1.38,
     *1.39,1.40,1.41,1.42,1.43,1.44,1.45,1.46,1.47,1.48,1.49 /
      DATA (DIST(I),I=151,300) / 1.5,1.5,1.5,1.5,1.5,1.6,1.6,
     *1.57,1.58,1.59,1.60,1.61,1.62,1.63,1.64,1.65,1.66,1.67,1.68,
     *1.69,1.70,1.71,1.72,1.73,1.74,1.75,1.76,1.77,1.78,1.79,1.80,
     *1.81,1.82,1.83,1.84,1.85,1.86,1.87,1.88,1.89,1.90,1.91,1.92,
     *1.93,1.94,1.95,1.96,1.97,1.98,1.99,2.00,2.01,2.02,2.03,2.04,
     *2.05,2.06,2.07,2.08,2.09,2.10,2.11,2.12,2.13,2.14,2.15,2.16,
     *2.17,2.18,2.19,2.20,2.21,2.22,2.23,2.24,2.25,2.26,2.27,2.28,
     *2.29,2.30,2.31,2.32,2.33,2.34,2.35,2.36,2.37,2.38,2.39,2.40,
     *2.41,2.42,2.43,2.44,2.45,2.46,2.47,2.48,2.49,2.50,2.51,2.52,
     *2.53,2.54,2.55,2.56,2.57,2.58,2.59,2.60,2.61,2.62,2.63,2.64,
     *2.65,2.66,2.67,2.68,2.69,2.70,2.71,2.72,2.73,2.74,2.75,2.76,
     *2.77,2.78,2.79,2.80,2.81,2.82,2.83,2.84,2.85,2.86,2.87,2.88,
     *2.89,2.90,2.91,2.92,2.93,2.94,2.95,2.96,2.97,2.98,2.99 /
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
C
          DO 1 J = 1,2
                IF (IDEV .EQ. 1) Z = RANF()
C
                IF (IDEV .EQ. 1 .AND. J .NE. 1) THEN
C
                      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
C
                      DELACC = ANS * SDEV(J)
                ELSE
                      DELACC = 0.0
                ENDIF
C
                Y(J) = A(J) + B(J)*TIMEX + C(J)*TIMEX**2 + DELACC
1         CONTINUE
C
          WRITE(2) TIME,(Y(J),J=1,2)
          KCNT = KCNT + 1
C
          IF (KCNT .EQ. KMAX) THEN
                KCNT = 0
                WRITE(6,10) TIME,(Y(J),J=1,2)
          ENDIF
C
2     CONTINUE
C
3     CONTINUE
      IF (KCNT .NE. 0) WRITE(6,10) TIME,(Y(J),J=1,2)
      RETURN
C
10    FORMAT(5X,1F10.3,3(5X,1E12.5))
11    FORMAT( // 10X,19HGAPFILL DATA POINTS )
      END
      SUBROUTINE POINTS(IPTS,T,ACC,A,B,C,SDEV,TSTART)
C
      DIMENSION T(2000),ACC(2000,3),XBAR(3),VAR(3),SDEV(3),A(3),B(3),
     * C(3),XSUM(3),XSQSUM(3),NTOTAL(3),YSUM(3),XYSUM(3),YBAR(3),NX(3),
     * XSQYSUM(3),XCUBSUM(3),XFORSUM(3),TZ(2000,3),AZ(2000,3),SUM(3)
C
      ICNT = 0
      ICNTMAX = 3
C
```

```
          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)
7               CONTINUE
C
1         CONTINUE
C
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
C               WRITE(6,22) I,XSQSUM(I),I,XSUM(I),I,XYSUM(I),I,YSUM(I),I,
C        *                     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)/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)
C
C         COMPUTE VARIANCE
C
                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
3               CONTINUE
C
                VAR(I) = SUM(I)/(N - 3)
                SDEV(I) = SQRT(VAR(I))
4         CONTINUE
C
```

```
C     IF (ICNT .EQ. 1) WRITE(6,20)
C     IF (ICNT .EQ. 2) WRITE(6,21)
C     IF (ICNT .EQ. 3) WRITE(6,25)
C     IF (ICNT .EQ. 4) WRITE(6,26)
C     WRITE(6,23)
C
       IF (ICNT .EQ. ICNTMAX) THEN
            WRITE(6,25)
            WRITE(6,23)
C
            DO 5 I = 1,2
                 WRITE(6,24) XBAR(I),YBAR(I),SDEV(I),A(I),B(I),C(I),
     *                           NTOTAL(I)
5          CONTINUE
C
       ENDIF
C
       IF (ICNT .LT. ICNTMAX) THEN
C
            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
6          CONTINUE
C
            GO TO 2
       ENDIF
C
       RETURN
C
20     FORMAT(/ 5X,24HFIRST STANDARD DEVIATION )
21     FORMAT(/ 5X,25HSECOND STANDARD DEVIATION )
22     FORMAT(/ 10X,7HXSQSUM(,1I1,4H) = ,1E12.5,5X,5HXSUM(,1I1,4H) = ,
     *1E12.5,5X,6HXYSUM(,1I1,4H) = ,1E12.5,5X,5HYSUM(,1I1,4H) = ,1E12.5
     * / 10X,8HXCUBSUM(,1I1,4H) = ,1E12.5,5X,8HXFORSUM(,1I1,4H) = ,
     *1E12.5,5X,8HXSQYSUM(,1I1,4H) = ,1E12.5,5X,7HNTOTAL(,1I1,4H) = ,
     *1I6 / )
23     FORMAT(3X,4HXBAR,8X,4HYBAR,10X,4HSDEV,10X,1HA,10X,1HB,11X,1HC,11X,
     *4HNPTS / )
24     FORMAT(6E12.5,3X,1I6)
25     FORMAT(/ 5X,24HTHIRD STANDARD DEVIATION )
26     FORMAT(/ 5X,25HFOURTH STANDARD DEVIATION )
       END
```

```
      PROGRAM GPREM06 (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE8,TAPE3
     *    ,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
C
C        UPDATE THE INPUTS WITH RESULTS FROM NCULL
C
      TIMSCI = 0.
C
C       INPUT BLOCK: CHANGE FOR EACH FLIGHT
C
      T1 = 65500.
      T2 = 68000.
C
C       WRITE(5,2005)
C
C
C        END INPUT BLOCK
C
C
C    T1... THE INITIAL TIME OF THE DATA SECTION
C    T2... THE FINAL TIME OF THE DATA SECTION.
C
      JJ = 1
   10 CONTINUE
      READ (3,20,END=30) TEND(JJ),TBEG(JJ)
   20 FORMAT (2X,F10.2,2X,F10.2)
C
C       WRITE(6,602) JJ,TBEG(JJ),TEND(JJ)
C
      JJ = JJ+1
      GO TO 10
   30 IGAPS = JJ-1
C
      WRITE (6,40) IGAPS
   40 FORMAT (2X,'NUMBER OF GAPS = ',I7)
C
C     CALCULATE THE MIDPOINT OF TIME
C
      TMID = (T2-T1)/2.+T1
C
C       SWITCH INPUT AND OUTPUT DEVICES AFTER RE-WRITING
```

```
C
      NGAPS = 0
      INTAPE = 8
      REWIND (INTAPE)
C
      DO 70 LL = 1, IGAPS
   50    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
C
C       WRITE(6,952) LL,TBEG(LL),TEND(LL),TYME,AX,AY,AZ
C
         WRITE (9) TYME,AX,AY,AZ,(TEMP(I),I=1,6)
         GO TO 50
   60    WRITE (9) TYME,AX,AY,AZ,(TEMP(I),I=1,6)
   70 CONTINUE
C
C    TRANSFER DATA TO OUTPUT AFTER GAPS ARE EXHAUSTED
C
   80 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
   90 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
C
      WRITE (6,120) NGAPS
  120 FORMAT (2X,'NO. OF DATUM REPLACED WITH 9999= ',1X,I7)
C
C
      INTAPE = 9
      REWIND (INTAPE)
C
C       TAPE 9 SHOULD NOW HOLD ALL 9999 IN THRUST PLACES
C
      T1 = 65500.
      T2 = 65900.
C
C      CALL STATS PROGRAM TO CHECK STATS FOR 400 SECOND PERIOD
C
C     CALL FINDSLP TO GET SLOPE FOR USE IN FINDSIG
C
      CALL FINDSLP (INTAPE)
C
      WRITE (6,130) AXSLP,AXINT,AYSLP,AYINT,AZSLP,AZINT
C
```

```
  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
C         CALCULATE STANDARD ERROR TO TEST VALUES AGAINST
C
      CALL FINDSIG (INTAPE)
C
      WRITE (6,140) AXSE,AYSE,AZSE
  140 FORMAT (2X,'FIRST STANDARD ERROR  AX,AY,AZ NO CULLING=:',/,3(2X,
     *    E15.7)/)
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)
C
      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
C
      CALZ = AZSLP*(XX)+AZINT
      CALY = AYSLP*(XX)+AYINT
      CALX = AXSLP*(XX)+AXINT
C
      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
C
      AZSE = SQRT(SUMAZSE/(NCOUNT-2))
      AXSE = SQRT(SUMAXSE/(NCOUNT-2))
      AYSE = SQRT(SUMAYSE/(NCOUNT-2))
      REWIND (INTAPE)
C
C
C
      RETURN

      END
C
      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
      REWIND (INTAPE)
      SUMX = 0.
      SUMXSQ = 0.
      SUMZSQ = 0.
      SUMYSQ = 0.
      SUMZ = 0.
      SUMY = 0.
      NCOUNT = 0
      RES = 0.
C
      SUMAZ = 0.
      SUMAX = 0.
      SUMAY = 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.
```

```
      AYSE = 0.
      AXSE = 0.
      SUMAZSE = 0.
      SUMAYSE = 0.
      SUMAXSE = 0.
C
C      CALCULATE STANDARD ERROR ON AZ AND AX
C
C
   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
C
      SUMAZ = SUMAZ+AZ
      SUMAX = SUMAX+AX
      SUMAY = SUMAY+AY
      SUMAZT = SUMAZT+(XX*AZ)
      SUMAXT = SUMAXT+(XX*AX)
      SUMAYT = SUMAYT+(XX*AY)
C
      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 = ',4F14.4,/,2X,
     *       '6 TEMP POINTS = ',6(1X,F10.3))
   30    FORMAT (2X,'LAST DATA POINT READ = ',4F14.4,/,2X,
     *       '6 TEMP POINTS = ',6(1X,2F12.2))
C
      ENDIF
C
C
   40 GO TO 10
C
   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)
C
C
      AZINT = SUMAZ/NCOUNT-AZSLP*(SUMX/NCOUNT)
      AXINT = SUMAX/NCOUNT-AXSLP*(SUMX/NCOUNT)
      AYINT = SUMAY/NCOUNT-AYSLP*(SUMX/NCOUNT)
C
      REWIND (INTAPE)
      RETURN
      END
```

```
      PROGRAM FILLCDE(INPUT,OUTPUT,TAPE5=INPUT
     *,TAPE6=OUTPUT,TAPE8,TAPE3,TAPE9,TAPE14)
       DIMENSION TEND(5000), TBEG(5000), TEMP(6)
       DIMENSION ORT(1600)
       DIMENSION XT(1600),YT(1600),ZT(1600)
C
       DIMENSION ARAN(10000),XRAN(1600)
       DIMENSION AAY(1600), AAX(1600), AAZ(1600),AT(1600)
C
       DIMENSION TX(1600) ,TY(1600), TZ(1600)
       DIMENSION AX2T(1600),AY2T(1600),AZ2T(1600)
       DIMENSION AX2(1600),AY2(1600),AZ2(1600)
       DIMENSION  W(1600),WY(1600,6),WX(1600,6),WZ(1600,6)
       DIMENSION WAX(6,6),WAY(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)
C
       REAL FINTIM
       INTEGER INTAPE,OUTTAPE
C
      LOGICAL FLGX, FLGZ
C
C
      NORMLY = 0
C
      NCULL = 0
       TIMSCI=0.
      FLGX = .TRUE.
      FLGZ = .TRUE.
C
      NPC = 0
      NSAT = 0
      FINTIM = 68050.
C      INPUT BLOCK: CHANGE FOR EACH FLIGHT
         TI400 = 65500.
C       END INPUT BLOCK
C
C
C
       INTAPE = 9
       REWIND(INTAPE)
C
C
C    READ IN CHUNKS OF DATA, 9 SECONDS IN LENGTH
C
C
      DO 625 L =1,1600
625   W(L) = 1.0
C
       NPC=NPC+1
1250   IF(NPC.EQ.1)THEN
         TBEG(NPC) = TI400
         TEND(NPC) = TBEG(NPC) + 9.
       ELSE
```

```
          TBEG(NPC) = TEND(NPC-1)
          TEND(NPC) = TBEG(NPC) + 9.
        END IF
C
C     BUILD ORIGINAL DATA SET
C
         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.
624       WZ(I,J) = 0.
626     CONTINUE
C
         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.
629       WAZ(K,M) = 0.
628     CONTINUE
C
C
         ND = 0
         NX = 0
         NY = 0
         NZ = 0
         NAD = 0
C
1299    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
C
C
         NAD = NAD + 1
         AT(NAD) = TYME - TBEG(1)
        ORT(NAD) = TYME
         TX(NAD) = TEMP(4)
         TY(NAD) = TEMP(5)
         TZ(NAD) = TEMP(6)
C
C    FILTER DATA FOR ERRONEOUS POINTS
C
C
         AAX(NAD) = AX
         AAY(NAD) = AY
         AAZ(NAD) = AZ
C
C   BUILD INTERIM DATA SET
```

```
C
          IF (AX.EQ.9999.) THEN
          GO TO 1291
         ELSE
          NX = NX + 1
          XT(NX) = TYME - TBEG(1)
          AX2(NX) = AX
          END IF
C
1291     IF (AY.EQ.9999.) THEN
            GO TO 1292
          ELSE
           NY = NY + 1
           YT(NY) = TYME  - TBEG(1)
           AY2(NY) = AY
          END IF
C
1292     IF (AZ.EQ.9999.) THEN
           GO TO 1299
          ELSE
           NZ = NZ + 1
           ZT(NZ) = TYME - TBEG(1)
           AZ2(NZ) = AZ
          END IF
C
         GO TO 1299
1300     CONTINUE
C
C
C
           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,WAX,BX,WX,IERR)
C        WRITE(6,681) IERR,NX
681      FORMAT(2X,'IERR = ',I5,' NUMBER OF DATA = ',I6)
C
          XSE = (SX(1)/NX)**0.5
C
C
C
C     IF XSE LT 1.0, ASSUME X IS SATURATED AND DO NOT CULL FURTHER
C     ONLY WRITE OUT ORIGINAL DATA OUT TO THE TAPE
C
C
          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
17      FORMAT(2X,'ARRAY AX2T AX2=',2(1X,F12.4),/,1600(2(1X,F12.4)))
7021   FORMAT(2X,'X 0TH,1ST,2ND = ',3(1X,E12.4),' XSE= ,IERR= ,NX= ',
      *E12.4,2(1X,I5))
```

```
C
7036    CONTINUE
C
         CALL LSQPOL(1600,NY,YT,1,AY2,W,3,3,RY,SY,WAY,BY,WY,IERR)
C
C       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
C
         CALL LSQPOL(1600,NZ,ZT,1,AZ2,W,3,3,RZ,SZ,WAZ,BZ,WZ,IERR)
C
C       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
C
C
C    SUM SHOULD GIVE SUM OF RESIDUALS FOR EACH POINT
C
C
7009    FORMAT(2X,'TBEG = ',F14.4,' TEND = ',F14.4,)
C
7010   FORMAT(2X,'Y 0TH,1ST,2ND = ',3(1X,E12.4),' YSE= ,IERR= ,NY= ',
     *E12.4,2(1X,I5))
7020    FORMAT(2X,'Z 0TH,1ST,2ND = ',3(1X,E12.4),' ZSE= ,IERR= ,NZ= ',
     *  E12.4,2(1X,I5))
C
7028    CONTINUE
        NCULL = 1
C
C    CULL THE ORIGINAL DATA ON THE BASIS OF 1 STANDARD ERROR
C
7025     CONTINUE
         NDBY = 0
         NDBZ = 0
         NDBX = 0
        DO 7030  J = 1, NAD
C
C   NAD = NUMBER OF POINTS IN ORIGINAL DATA SET
C
        TIM = AT(J)
C      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
8002    CONTINUE
C002    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
```

```
          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
          AZ2(NDBZ) = AAZ(J)
          AZ2T(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
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,1ST,2ND = ',3(1X,E12.4),' XSE = ',E12.4,
     * ' NCULL,IE,NDBX= ',3(1X,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
C       WRITE(6,681) IERR,NDBY
        YSE = (SY(1)/NDBY)**0.5
C
        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,AZ2T,1,AZ2,W,3,3,RZ,SZ,WAZ,BZ,WZ,IERR)
C
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,1ST,2ND =',3(1X,E12.4),' YSE = ',
     * E12.4,' NCULL,IE,NDBY= ',3(1X,I5))
7052  FORMAT(2X,'Z 0TH,1ST,2ND = ',3(1X,E12.4),' ZSE = ',E12.4,
     * ' NCULL,IE,NDBZ= ',3(1X,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 CULLS
 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)
 1400    X = 0.0
 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.
 524      WZ(I,J) = 0.
 526     CONTINUE
 C
         DO 528 K = 1,6
              BX(K,1) = 0.
              BY(K,1) = 0.
              BZ(K,1) = 0.
         DO 529  M = 1,6
           WAX(K,M) = 0.
```

```
        WAY(K,M) = 0.
529     WAZ(K,M) = 0.
528     CONTINUE
C
C
C    CALCULATE STATS ON DATA AFTER IT HAS BEEN FILLED
C
C
      CALL LSQPOL(1600,NAD,AT,1,AAX,W,3,3,RX,SX,WAX,BX,WX,IERR)
C
C        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))
C
C
      CALL LSQPOL(1600,NAD,AT,1,AAY,W,3,3,RY,SY,WAY,BY,WY,IERR)
C
C        WRITE(6,681) IERR,NAD
         YSE = (SY(1)/NAD)**0.5
C
         WRITE(6,9151) BY(1,1),BY(2,1),BY(3,1),YSE,NCULL,IERR,NAD
C
      CALL LSQPOL(1600,NAD,AT,1,AAZ,W,3,3,RZ,SZ,WAZ,BZ,WZ,IERR)
C
C        WRITE(6,681) IERR,NAD
         ZSE = (SZ(1)/NAD)**0.5
C
         WRITE(6,9152) BZ(1,1),BZ(2,1),BZ(3,1),ZSE,NCULL,IERR,NAD
C
         NPC = NPC + 1
C
C   CONTINUE CYCLES
C
         GO TO 1250
C
1307        WRITE(6,943)
943        FORMAT(2X,'ONE ARRAY HAS NO DATA IN IT')
C
      GO TO 1311
C
1310    CONTINUE
         NSAT = NSAT + 1
         IF (NSAT.EQ.1) THEN
         WRITE(6,949)
949     FORMAT(2X,'X SATURATION REACHED ,SIMPLY TRANSFER DATA')
C
        END IF
C
```

```
C
C      WRITE REMAINING DATA IN BUFFERTO OUTPUT TAPE
C
C
         DO 1450 J = 1, NAD
         WRITE(14) ORT(J),AAX(J),AAY(J),AAZ(J),TX(J),TY(J),TZ(J)
1450    CONTINUE
C
C       CONTINUE TO TRANSFER DATA IWTH NO CULLLING
C
1311    CONTINUE
C
C
7000    WRITE(6,6079) TYME,AX,AY,AZ
6079    FORMAT(2X,'LAST TYME,AX,AY,AZ = ',4(2X,F15.4))
C
C
       WRITE(14) EOF
C
        END
```

```
      PROGRAM INTTIM (INPUT,OUTPUT,TAPE5=INPUT.TAPE6=OUTPUT,TAPE8,TAPE9)
C
C
      INTAPE = 8
      REWIND (INTAPE)
      TBEG = 61260
      TLAST = 62696
C
C     ESTABLISH TIME PERIOD OVER WHICH TO AVERAGE
C
      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
      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
C
C
   70 CONTINUE
C
      T2 = TYME
C
C     CALCULATE AVERAGE AND WRITE OUT
C
C     FOR THE CASE WHEN SATURATION HAS OCCURED YET THRUSTING
C     STILL ACCOUNTS FOR MOST OF THE DATA. REPLACE THE AVERAGE
C     WHICH IS UNDEFINED WITH DATA REPRESENTING SATURATION.
      IF (NRDX.EQ.0) THEN
      AVGX =-8000.
      WRITE(6,7890) TYME
7890  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
7891  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
7892  FORMAT(2X,'FALSE Z SATURATION VALUE AT TYME = ',F12.4)
```

```
        ELSE
      AVGZ = AVGZ/NRDZ
      END IF
      TXAVG = TXAVG/NT
      TYAVG = TYAVG/NT
      TZAVG = TZAVG/NT
      ITIME = INT( (Tl+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, Tl, T2, NRD, AVGX, AVGY. AVGZ, TXAVG, TYAVG, TZAVG
3200 CONTINUE
80 FORMAT (2X, 'RTIME Tl T2 NRD AVGX AVGY AVGZ = ', /, 2X, lX, F14 5, 2 (1X,
      *F14 5), I5,3(1X,F14 5), /, lX, '3 TEMPS = ',3(1X,F14 5))
C
      NRD = 0
      NRDX = 0
      NRDY = 0
      NRDZ = 0
      NT = 0
      TXAVG = 0
      TYAVG = 0
      TZAVG = 0
      AVGX = 0
      AVGY = 0
      AVGZ = 0
C
C     CALCULATE NEW TIME PERIOD
C
      NSEC = NSEC+ 1
      TST = TEND
      TEND = TST+ 1.0
C
      GO TO 30
C
C
C
     90 WRITE (6, 100)
        T2 = TYME
    100 FORMAT (2X, 'END OF TAPE 8 ENCOUNTERED ' )
C
```

```
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= ',1X,F14.5,I5,3(1X,F14.5)
     */,1X,'3 TEMPS = ',3(1X,F14.5))
C
      END
```

```
      PROGRAM ORBPLTA(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7,
     *                TAPE8,TAPE1,TAPE3)
C
C      WRITTEN BY HIRAP PROJECT GROUP - JANUARY 1986
C
C      THIS PROGRAM PLOTS HIRAP ACCELERATION DATA IN MICRO-GS.
C      ALTITUDE IS READ FROM A MERGED HCDAT FILE (IF AVAILABLE)
C      AND IS LINEARLY INTERPOLATED TO MATCH THE HIRAP DATA RATE.
C      ALTITUDE (METERS) IS THEN WRITTEN TO THE MICRO-G FILE
C      AND WILL BE RETAINED FOR THE DATA ANALYSIS PROCESS SO THAT
C      ACCELERATION DATA MAY BE ANALYSED WITH RESPECT TO TIME AND/OR
C      ALTITUDE.
C
C
C
C      IMG = 1  INPUT DATA IN MICRO-GS (FILE 8)
C
C      ICG = 0  MICRO-GS NOT YET CORRECTED TO THE VEHICLE CG
C      ICG = 1  MICRO-GS CORRECTED TO THE VEHICLE CG
C
C      IFLT = STS FLIGHT NUMBER
C
C      IPLTMG = 0 ACCELEROMETER MICRO-G DATA NOT PLOTTED
C             = 1 PLOTS ACCELEROMETER DATA IN MICRO-GS
C
C      IRECAL = 0  OLD (BEFORE MOD OR RECAL) HIRAP CALIBRATIONS
C      IRECAL = 1  RECALIBRATED HIRAP CALIBRATIONS, USING THE
C                  CORRECTED TEMPERATURE MONITOR VOLTAGE, VC
C
C      INORBIT = 0  GROUND CALIBRATION TEMPERATURE BIAS
C              = 1  IN-ORBIT BIAS CALIBRATION I
C              = 2  IN-ORBIT BIAS CALIBRATION II
C              = 3  IN-ORBIT BIAS CALIBRATION III
C
C     **************************************************************
C     *                                                            *
C     *   NOTE : CURRENTLY THE IN-ORBIT BIAS CALIBRATIONS ARE      *
C     *          FOR THE BEFORE-RECAL S/N 001 HIRAP ONLY           *
C     *         (STS-6,7,8,11,13,AND 24)                           *
C     *                                                            *
C     **************************************************************
C
C
C     PLOTTING ROUTINE INPUTS:
C
C           STRTALT = X AXIS ORIGIN IN KM
C           STOPALT = X AXIS FINAL ALT IN KM
C
C           STRTY = Y AXIS ORIGIN IN MICRO-GS (OR COUNTS)
C           STOPY = Y AXIS MAXIMUM MICRO-GS (OR COUNTS)
C
C           T1 = START TIME FOR FILE, SECONDS
C
C           T2 = STOP TIME FOR FILE, SECONDS
```

```
C
C           DAY =  DAY OF DATA FILE TO BE PLOTTED
C
C           ALTTIC = INCREMENT FOR TICK MARKS ON X AXIS, KM
C
C           YTIC = INCREMENT FOR TICK MARKS ON Y AXIS, MICRO-GS
C
C           ALTLABL = INCREMENT OF LABELED ALT, KM
C
C           YLABL = INCREMENT OF LABELED MICRO-GS
C
C
C
C
      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)
C
      ISTOP = 0
      IPNT = 0
      IFILE = 6
      NFILE = 0
      NP = 0
      JPLOT = 0
C
C
      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
C
      IF(INORBIT.GT.4) THEN
         WRITE(3,64) INORBIT
         ISTOP = 1
      ENDIF
C
      IF(SN.EQ.2.AND.INORBIT.NE.0) THEN
         WRITE(3,65) ISN,INORBIT
         ISTOP = 1
      ENDIF
C
      IF(DELALT.LT.ALTTIC) THEN
         WRITE(3,66) DELALT,ALTTIC
         ISTOP = 1
      ENDIF
C
      IF(DELMG.LT.YTIC) THEN
```

```
            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
C
         IF(ISTOP.EQ.1) STOP
C
C
         IF(IMEAN.EQ.1) THEN
            KMAX = 1
         ELSE
            KMAX = 350
            IF(FLT.EQ.32) KMAX = 225
         ENDIF
         KCNT = KMAX - 1
C
```

```
C
       IF (NFILE.EQ.4) GO TO 15
       IFILE = IFILE + 1
       NFILE = NFILE + 1
       REWIND(IFILE)
       REWIND(NFILE)
7     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,STRTY,STOPY
            WRITE(6,53) IFILE,NFILE,T1,T2,DAY
       ENDIF
C
       READ(IFILE,END=10) ISN,NCHAN,(NAMES(I),I=1,NCHAN),
     *                (IUNITS(I),I=1,NCHAN),(HDR(I),I=1,8)
10    IF (EOF(IFILE).NE.0) THEN
            WRITE(6,51) IFILE
            STOP
       ENDIF
C
       ISN = 1
       NCHAN = 5
       NAMES(2) = 3HALT
       NAMES(3) = 6HX-AXIS
       NAMES(4) = 6HZ-AXIS
       NAMES(5) = 6HY-AXIS
       IUNITS(2) = 6HMETERS
       IUNITS(3) = 8HMICRO-GS
       IUNITS(4) = 8HMICRO-GS
       IUNITS(5) = 8HMICRO-GS
C
       WRITE(8) ISN,NCHAN,(NAMES(I),I=1,NCHAN),(IUNITS(I),I=1,NCHAN),
     *           (HDR(I),I=1,8)
       WRITE(6,50) ISN,NCHAN,HDR
C
C          READ HDAT FILE
C
            READ(NFILE)
            I = 0
16         CONTINUE
            I = I + 1
            READ(NFILE,END=17) (HDAT(J),J=1,3)
            HTIME(I) = HDAT(1)
            HALT(I) = HDAT(3)
17         IF (EOF(NFILE).NE.0) THEN
                 NT = I - 1
                 WRITE(6,62) NFILE,NT
                 GO TO 1
            ENDIF
            GO TO 16
C
C
```

```
C     READ DATA FILE
C
1     CONTINUE
      READ(IFILE,END=2) TIME,XMG,ZMG,YMG
C
      IF(TIME.LT.T1) GO TO 1
      IF(TIME.GT.T2) GO TO 15
C
      CALL IUNI(5000,NT,HTIME,1,HALT,1,TIME,ALT,-1,IE)
      WRITE(8) TIME,ALT,XMG,ZMG,YMG
2     IF (EOF(IFILE).NE.0) THEN
             WRITE(6,58) TIME
             GO TO 15
      ENDIF
      GO TO 1
C
C
C
C
15    CONTINUE
      WRITE(6,63) (NAMES(I),I=1,NCHAN)
      WRITE(6,36) (IUNITS(I),I=1,NCHAN)
C
C
      IF(IPLTMG.EQ.1)THEN
         REWIND(8)
         READ(8)
C
C        PLOT X-AXIS MICROGS AS A FUNCTION OF ALTITUDE
C
         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)
20       READ(8,END=21) TIME,ALT,XMG,ZMG,YMG
         ALT = ALT/1000.
         IF(ALT.GT.STRTALT) GO TO 20
         TPLOT = (STRTALT - ALT) * (XLENG/DELALT)
         YPLOT = (XMG - STRTY) * (YLENG/DELMG)
         IF(YPLOT.GE.0.AND.YPLOT.LT.YLENG) THEN
            CALL PNTPLT(TPLOT,YPLOT,21,1)
         ENDIF
21       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
C
22       CONTINUE
```

```
      GO TO 32
      REWIND(8)
      READ(8)
C
C     PLOT Y-AXIS MICRO-GS AS A FUNCTION OF ALTITUDE
C
      IY = 2
      CALL NFRAME
      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)
 30   READ(8,END=31) TIME,ALT,XMG,ZMG
      ALT = ALT/1000.
      IF(ALT.GT.STRTALT) GO TO 30
      TPLOT = (STRTALT - ALT) * (XLENG/DELALT)
      YPLOT = (YMG - STRTY) * (YLENG/DELMG)
      IF(YPLOT.GE.0.AND.YPLOT.LT.YLENG) THEN
         CALL PNTPLT(TPLOT,YPLOT,21,1)
      ENDIF
 31   IF(EOF(8).NE.0.OR.TIME.GT.T2) GO TO 32
      GO TO 30
C
 32   CONTINUE
      REWIND(8)
      READ(8)
C
C     PLOT Z-AXIS MICRO-GS AS A FUNCTION OF ALTITUDE
C
      IY = 3
      JPLOT = 0
      CALL NFRAME
      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)
 40   READ(8,END=41) TIME,ALT,XMG,ZMG
      ALT = ALT/1000.
      IF(ALT.GT.STRTALT) GO TO 40
      TPLOT = (STRTALT - ALT) * (XLENG/DELALT)
      YPLOT = (ZMG - STRTY) * (YLENG/DELMG)
      IF(YPLOT.GE.0.AND.YPLOT.LT.YLENG) THEN
         CALL PNTPLT(TPLOT,YPLOT,21,1)
      ENDIF
 41   IF(EOF(8).NE.0.OR.TIME.GT.T2) GO TO 42
      GO TO 40
C
C
 42     CALL CALPLT(0.0,0.0,999)
      ENDIF
C
      STOP
C
 50   FORMAT(/ 5X,7HFILE = ,1I5,5X,8HNCHAN = ,1I2,5X,15HFOUND HEADER =
     *        ,8A10 / )
 51   FORMAT(// 5X,49HEOF FOUND WHEN ATTEMPTING TO READ HEADER ON FILE
```

```
     *        ,1I2 )
52    FORMAT(5F8.0)
53    FORMAT(5X,8HIFILE = ,1I2,/ 5X,8HNFILE = ,1I1,/ 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 ,1I2,5X,6HIMG = ,1I1,5X,6HICG = ,1I1,
     *        5X,9HIRECAL = ,1I1,5X,8HIMEAN = ,1I1,5X,9HIPLTMG = ,1I1,
     *        5X,10HINORBIT = ,1I1 /)
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 = ,1I6 / )
62    FORMAT(/ 5X,32HNUMBER OF MERGED POINTS ON FILE ,1I1,3H = ,
     *         1I4 /)
63    FORMAT(/1X,5(2X,A10))
36    FORMAT( 1X,5(2X,A10) //)
64    FORMAT(// 5X,18HWRONG! (INORBIT = ,1I3,1H),
     *        / 5X,46HINPUT VALUE OF INORBIT MUST BE = 0,1,2,3, OR,4,
     *        / 5X,9HTRY AGAIN )
65    FORMAT(// 5X,13HWRONG! (SN = ,1I3,12H, INORBIT = ,1I3,1H),
     *        / 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! (DELALT = ,1F8.0,11H, ALTTIC = ,1F8.0,1H),
     *        / 5X,44HDELALT 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
C     THIS ROUTINE DRAWS GRID FOR HIRAP ACCELEROMETER VS TIME PLOTS
```

```
C
C     IY = 1 FOR X AXIS
C        = 2 FOR Y AXIS
C        = 3 FOR Z AXIS
C
      DIMENSION TITLE1(5),TITLE2(4),TITLE3(8),YCHAR(3),LABEL1(6),
     *          LABEL2(4),LABEL3(2),LABEL4(2)
C
      DATA TITLE1 / 10HHIRAP S/N ,10H00  UNCORR,10HECTED ACCE,
     *              10HLEROMETER ,10HDATA, STS- /
      DATA TITLE2 / 10HHIRAP S/N ,10H00  ACCELE,10HROMETER DA,
     *              10HTA - STS   /
      DATA TITLE3 / 10HCORRECTED ,10HTO VEHICLE,10H CG                /
C     DATA TITLE3 / 10HCALCULATED,10H INDUCED A,10HCCELERATIO,
C    *              10HN DUE TO H,10HIRAP OFFSE,10HT FROM VEH,
C    *              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      /
C
      DELALT = (STRTALT - 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)
C
C     DRAW TICK MARKS ON X AXIS
C
      NX = DELALT/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
```

```
C
C     DRAW TICK MARKS ON Y AXIS
C
      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)
3     CONTINUE
C
C     LABEL X AXIS
C
      DELX = DELX * BIGTICX
      YBAR = -0.25
      XVAL = STRTALT + ALTLABL
      IEND = NX/BIGTICX + 1
      DO 5 I = 1,IEND
      XVAL = XVAL - ALTLABL
      XST = - 0.20
      XBAR = XST + DELX * (I-1)
      CALL NUMBER(XBAR,YBAR,0.10,XVAL,0.0,-1)
5     CONTINUE
C
C     LABEL Y AXIS
C
      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)
6     CONTINUE
      XBAR = 1.0
      YBAR = YLENG + 1.0
      IF(ICG.EQ.1) THEN
         CALL CHARACT(XBAR,YBAR,0.10,TITLE2,0.0,40)
      ELSE
```

```
            CALL CHARACT(XBAR,YBAR,0.10,TITLE1,0.0,50)
         ENDIF
         XBAR = XBAR + 1.12
         CALL NUMBER(XBAR,YBAR,0.10,SN,0.0,-1)
         IF(ICG.EQ.1) THEN
            XBAR = XBAR + 2.48
         ELSE
            XBAR = XBAR + 3.60
         ENDIF
         CALL NUMBER(XBAR,YBAR,0.10,FLT,0.0,-1)
         XBAR = 1.0
         YBAR = YLENG + 0.75
         IF(IMEAN.EQ.1) THEN
            CALL CHARACT(XBAR,YBAR,0.10,LABEL4,0.0,20)
            YBAR = YLENG + 0.50
         ELSEIF(ICG.EQ.1) THEN
            CALL CHARACT(XBAR,YBAR,0.10,TITLE3,0.0,80)
            YBAR = YLENG + 0.50
         ENDIF
         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 = YLENG + 0.50
         IF(IMEAN.EQ.1.OR.ICG.EQ.1) THEN
            YBAR = YLENG + 0.25
         ENDIF
         IF(ICG.EQ.1) THEN
            CALL CHARACT(XBAR,YBAR,0.10,LABEL3,0.0,20)
         ELSE
            CALL CHARACT(XBAR,YBAR,0.10,LABEL2,0.0,40)
         ENDIF
         RETURN
         END
         SUBROUTINE ORBCAL1(XCOUNT,TEMP,XMICROG)
C
C       THIS ROUTINE COMPUTES THE SCALE FACTOR AND THE BIAS IN THE X, Y,
C       AND Z ACCELEROMETERS FOR HIRAP S/N 001 USING THE CALCULATED TEMPERATURE
C       FROM THE OLD (BEFORE MOD) CALIBRATIONS. THE INPUT IS XCOUNT,
C       AND THE OUTPUT ACCELEROMETER DATA IN MICRO-GS IS XMICROG.
C       THE MICRO-G CALIBRATION ROUTINE USED HERE IS THE IN-ORBIT
C       FLIGHT DERIVED SET OF EQUATIONS FOR THE TEMPERATURE DEPENDENT
C       BIAS.  THIS ROUTINE IS A FIRST ATTEMPT AT A MORE ACCURATE BIAS
C       DETERMINATION THAN CURRENT GROUND CALIBRATION TECHNIQUES.
C
        DIMENSION B(3), BCOEF(4,3), SF(3), TEMP(3), XCOUNT(3), XMICROG(3),
       *          VX(3)
C
        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 /
C
        SF(1) = -1.247237E-3
```

```
      SF(2) =  1.253821E-3
      SF(3) = -1.246810E-3
C
C
C     COMPUTE ACCELEROMETER VOLTAGES FROM COUNTS DATA
C
      DO 1 I = 1,3
      VX(I) = -10.0 + XCOUNT(I)/16383.0 * 20.0
1     CONTINUE
C
C     COMPUTE BIAS FOR ALL THREE AXES
C
      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)
2     CONTINUE
C
C
C     COMPUTE ACCELERATION IN MICRO-GS
C
      DO 3 I = 1,3
      XMICROG(I) = VX(I)/SF(I) - B(I)
3     CONTINUE
C
      RETURN
      END
      SUBROUTINE ORBCAL2(XCOUNT,TEMP,XMICROG)
C
C     THIS ROUTINE COMPUTES THE SCALE FACTOR AND THE BIAS IN THE X, Y,
C     AND Z ACCELEROMETERS FOR HIRAP S/N 001 USING THE CALCULATED TEMPERATURE
C     FROM THE OLD (BEFORE MOD) CALIBRATIONS. THE INPUT IS XCOUNT,
C     AND THE OUTPUT ACCELEROMETER DATA IN MICRO-GS IS XMICROG.
C     THE MICRO-G CALIBRATION ROUTINE USED HERE IS THE IN-ORBIT
C     FLIGHT DERIVED SET OF EQUATIONS FOR THE TEMPERATURE DEPENDENT
C     BIAS.  THIS ROUTINE IS AN ATTEMPT AT A MORE ACCURATE BIAS
C     DETERMINATION THAN CURRENT GROUND CALIBRATION TECHNIQUES.
C
      DIMENSION B(3), BCOEF(4,3), SF(3), TEMP(3), XCOUNT(3), XMICROG(3),
     *          VX(3)
C
      DATA BCOEF  / 1241., 3.50654,-0.7643, 0.0057957,
     *              217., -39.143, 0.9526, -0.0060287,
     *              953., -2.17931,-0.6482, 0.0050837 /
C
      SF(1) = -1.247237E-3
      SF(2) =  1.253821E-3
      SF(3) = -1.246810E-3
C
C
C     COMPUTE ACCELEROMETER VOLTAGES FROM COUNTS DATA
C
      DO 1 I = 1,3
      VX(I) = -10.0 + XCOUNT(I)/16383.0 * 20.0
1     CONTINUE
```

```
C
C     COMPUTE BIAS FOR ALL THREE AXES
C
      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)
2     CONTINUE
C
C
C     COMPUTE ACCELERATION IN MICRO-GS
C
      DO 3 I = 1,3
      XMICROG(I) = VX(I)/SF(I) - B(I)
3     CONTINUE
C
      RETURN
      END
      SUBROUTINE ORBCAL3(XCOUNT,TEMP,XMICROG)
C
C     THIS ROUTINE COMPUTES THE SCALE FACTOR AND THE BIAS IN THE X, Y,
C     AND Z ACCELEROMETERS FOR HIRAP S/N 001 USING THE CALCULATED TEMPERATURE
C     FROM THE OLD (BEFORE MOD) CALIBRATIONS. THE INPUT IS XCOUNT,
C     AND THE OUTPUT ACCELEROMETER DATA IN MICRO-GS IS XMICROG.
C     THE MICRO-G CALIBRATION ROUTINE USED HERE IS THE IN-ORBIT
C     FLIGHT DERIVED SET OF EQUATIONS FOR THE TEMPERATURE DEPENDENT
C     BIAS.  THIS ROUTINE IS AN ATTEMPT AT A MORE ACCURATE BIAS
C     DETERMINATION THAN CURRENT GROUND CALIBRATION TECHNIQUES.
C
      DIMENSION B(3), BCOEF(4,3), SF(3), TEMP(3), XCOUNT(3), XMICROG(3),
     *          VX(3)
C
      DATA BCOEF  / 2465., -56.2229, 0.25003, 0.0,
     *              -916.,  22.503, -0.12344, 0.0,
     *              1737., -49.3751, 0.23295, 0.0 /
C
      SF(1) = -1.247237E-3
      SF(2) =  1.253821E-3
      SF(3) = -1.246810E-3
C
C
C     COMPUTE ACCELEROMETER VOLTAGES FROM COUNTS DATA
C
      DO 1 I = 1,3
      VX(I) = -10.0 + XCOUNT(I)/16383.0 * 20.0
1     CONTINUE
C
C     COMPUTE BIAS FOR ALL THREE AXES
C
      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)
2     CONTINUE
C
C
```

```
C     COMPUTE ACCELERATION IN MICRO-GS
C
      DO 3 I = 1,3
      XMICROG(I) = VX(I)/SF(I) - B(I)
3     CONTINUE
C
      RETURN
      END
      SUBROUTINE XCALIB1(XCOUNT,TEMP,XMICROG)
C
C     THIS ROUTINE COMPUTES THE SCALE FACTOR AND THE BIAS IN THE X, Y,
C     AND Z ACCELEROMETERS FOR HIRAP S/N 001 USING THE CALCULATED TEMPERATURE
C     FROM THE OLD (BEFORE MOD) CALIBRATIONS. THE INPUT IS XCOUNT,
C     AND THE OUTPUT ACCELEROMETER DATA IN MICRO-GS IS XMICROG.
C
      DIMENSION B(3), BCOEF(4,3), SF(3), TEMP(3), XCOUNT(3), XMICROG(3),
     *          VX(3)
C
      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 /
C
      SF(1) = -1.247237E-3
      SF(2) = 1.2538210E-3
      SF(3) = -1.246810E-3
C
C
C     COMPUTE ACCELEROMETER VOLTAGES FROM COUNTS DATA
C
      DO 1 I = 1,3
      VX(I) = -10.0 + XCOUNT(I)/16383.0 * 20.0
1     CONTINUE
C
C     COMPUTE BIAS FOR ALL THREE AXES
C
      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)
2     CONTINUE
C
C
C     COMPUTE ACCELERATION IN MICRO-GS
C
      DO 3 I = 1,3
      XMICROG(I) = VX(I)/SF(I) - B(I)
3     CONTINUE
C
      RETURN
      END
      SUBROUTINE XCALIB2(XCOUNT,TEMP,XMICROG)
C
C     THIS ROUTINE COMPUTES THE SCALE FACTOR AND THE BIAS IN THE X, Y,
C     AND Z ACCELEROMETERS FOR HIRAP S/N 002 USING THE CALCULATED TEMPERATURE
C     FROM THE OLD (BEFORE RECAL) CALIBRATIONS. THE INPUT IS XCOUNT,
```

```
C      AND THE OUTPUT ACCELEROMETER DATA IN MICRO-GS IS XMICROG.
C
       DIMENSION B(3), BCOEF(4,3), SF(3), TEMP(3), XCOUNT(3), XMICROG(3),
     *           VX(3)
C
       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 /
C
       SF(1) = -1.248570E-3
       SF(2) =  1.269482E-3
       SF(3) = -1.256565E-3
C
C      COMPUTE ACCELEROMETER VOLTAGES FROM COUNTS DATA
C
       DO 1 I = 1,3
       VX(I) = -10.0 + XCOUNT(I)/16383.0 * 20.0
1      CONTINUE
C
C      COMPUTE BIAS FOR ALL THREE AXES
C
       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)
2      CONTINUE
C
C
C      COMPUTE ACCELERATION IN MICRO-GS
C
       DO 3 I = 1,3
       XMICROG(I) = VX(I)/SF(I) - B(I)
3      CONTINUE
C
       RETURN
       END
       SUBROUTINE RECAL1(XCOUNT,VTEMP,XMICROG)
C
C      THIS ROUTINE COMPUTES THE SCALE FACTOR AND THE BIAS IN THE X, Y,
C      AND Z ACCELEROMETERS FOR THE RECALIBRATED HIRAP S/N 001 USING
C      THE CORRECTED TEMPERATURE VOLTAGE, VTEMP.
C      THE INPUT ACCELEROMETER DATA IN COUNTS IS XCOUNT,
C      AND THE OUTPUT ACCELEROMETER DATA IN MICRO-GS IS XMICROG.
C
       DIMENSION B(3), BCOEF(4,3), SF(3), SFCOEF(4,3), VTEMP(3),
     *           XCOUNT(3), XMICROG(3), VX(3)
C
       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 /
C
C      COMPUTE ACCELEROMETER VOLTAGES FROM COUNTS DATA
C
```

```
      DO 1 I = 1,3
      VX(I) = -10.0 + XCOUNT(I)/16383.0 * 20.0
1     CONTINUE
C
C     COMPUTE BIAS FOR ALL THREE AXES
C
      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)
2     CONTINUE
C
C     COMPUTE SCALE FACTOR FOR ACCELEROMETER
C
      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
3     CONTINUE
C
C     COMPUTE ACCELERATION IN MICRO-GS
C
      DO 4 I = 1,3
      XMICROG(I) = VX(I)/SF(I) - B(I)
4     CONTINUE
C
      RETURN
      END
      SUBROUTINE RECAL2(XCOUNT,VTEMP,XMICROG)
C
C     THIS ROUTINE COMPUTES THE SCALE FACTOR AND THE BIAS IN THE X, Y,
C     AND Z ACCELEROMETERS FOR THE RECALIBRATED HIRAP S/N 002 USING
C     THE CORRECTED TEMPERATURE VOLTAGE, VTEMP.
C     THE INPUT ACCELEROMETER DATA IN COUNTS IS XCOUNT,
C     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)
C
      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 /
C
C     COMPUTE ACCELEROMETER VOLTAGES FROM COUNTS DATA
C
      DO 1 I = 1,3
      VX(I) = -10.0 + XCOUNT(I)/16383.0 * 20.0
1     CONTINUE
C
C     COMPUTE BIAS FOR ALL THREE AXES
C
      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)
```

```
2     CONTINUE
C
C     COMPUTE SCALE FACTOR FOR ACCELEROMETER
C
      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
3     CONTINUE
C
C     COMPUTE ACCELERATION IN MICRO-GS
C
      DO 4 I = 1,3
      XMICROG(I) = VX(I)/SF(I) - B(I)
4     CONTINUE
C
      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
 5    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
 7    NCHAN=4
C
 10   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 ALT=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
      AVZMG=SUMZMG/KNTZ
     AVYMG=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)
C
C        TAPE (1) = ATRW          (EPOCH)     201 WORD FILE
C              2  = XBET          (EPOCH)   66 WORD FILE
C              3  = HIRPB           (DECIMAL TIME, NO HEADER)
C              4  = XLAR          (DECIMAL TIME, TWO HEADER RECORDS)
C
C -------------------------
C        TAPE (5) = MERGED FILES USING DECIMAL TIME
C              6  = PRINT OUT AT TEN SECOND INTERVALS
C
C ------------------------------------------------------------
C
      DIMENSION DAT(201),UNITS(24),ALPHA(24),TITLES(8),DLAR(29),DXT(66)
C
      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
C
      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.
C
C     HIRAP HEADER READ FOR TEST RUN OF STS-24
       READ(3,1666)
       KVNT = 0
C
C  ABRIDGED  24-WORD  DATA  ARRAY
C
      ALPHA(1)=10HTIME
      UNITS(1)=10HSEC
C
      ALPHA(2)=10HVEL
      UNITS(2)=10HM/SEC
C
      ALPHA(3)=10HALT
```

```
      UNITS(3)=10HMETERS
C
      ALPHA(4)=10HPLAT
      UNITS(4)=10HDEG
C
      ALPHA(5)=10HPLONG
      UNITS(5)=10HDEG
C
      ALPHA(6)=10HALPHA
      UNITS(6)=10HDEG
C
      ALPHA(7)=10HBETA
      UNITS(7)=10HDEG
C
      ALPHA(8)=10HP
      UNITS(8)=10HDEG/S
C
      ALPHA(9)=10HQ
      UNITS(9)=10HDEG/S
C
      ALPHA(10)=10HR
      UNITS(10)=10HDEG/S
C
      ALPHA(11)=10HPDOT
      UNITS(11)=10HDEG/S2
C
      ALPHA(12)=10HQDOT
      UNITS(12)=10HDEG/S2
C
      ALPHA(13)=10HRDOT
      UNITS(13)=10HDEG/S2
C
      ALPHA(14)=10HAELE
      UNITS(14)=10HDEG
C
      ALPHA(15)=10HABFL
      UNITS(15)=10HDEG
C
      ALPHA(16)=10HWGT
      UNITS(16)=10HKG
C
      ALPHA(17)=10HDUMY1
      UNITS(17)=10H
C
      ALPHA(18)=10HRL
      UNITS(18)=10HKG/M3
C
      ALPHA(19)=10HTIL
      UNITS(19)=10HKELV
C
      ALPHA(20)=10HPIL
      UNITS(20)=10HN/M2
C
      ALPHA(21)=10HWML
```

```
      UNITS(21)=10HKG/KMOL
C
      ALPHA(22)=10HDUMY2
      UNITS(22)=10H
C
      ALPHA(23)=10HAX
      UNITS(23)=10HM/S2
C
      ALPHA(24)=10HAZ
      UNITS(24)=10HM/S2
C
C -------------------------------
C
C                    WRITE  HEADERS :   TAPE(5)  AND  PRINT - OUT
C
      ISEQ=1
      NWDS=24
      WRITE(5)ISEQ,NWDS,(ALPHA(I),I=1,NWDS),(UNITS(I),I=1,NWDS),
     .TITLES
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,NWDS)
 1003 FORMAT(2X,*.....LABELS AND UNITS FOR DATA WORDS .....*/
     .(3(3X,*(*,I3,*)*,2X,A10,3X,A10)))
C
      WRITE(6,1004)
 1004 FORMAT(///4X,*TIME     ALT       WML       AELE      ABFL       AX
     *   AZ        VEL       TIL       PIL       RL       ALPHA*/)
C
C
C ------------------
C
C INITIALIZE COUNTERS
      TIME=-1.
      N6=10
      NZP=1
C
C
C OPTIONAL HIRAP AX , AZ SMOOTHING
      IF(IXSM.EQ.1)GOTO81
      IF(IZSM.EQ.1)GOTO81
      GOTO8
C
 81   K=0
      J=0
 1    J=J+1
      READ(3,1666)(DAXZ(I,J),I=1,3)
      IF(EOF(3))2,1
 2    K=K+1
C
      DAXZ(1,K)=DAXZ(1,K+3)
```

```
C
      IF(IXSM.EQ.1)GOTO82
      DAXZ(2,K)=DAXZ(2,K+3)
      GOTO91
 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
      DO 85 I=2,NS
      IF(SMOO(I).LE.DM)GOTO85
      DM=SMOO(I)
      IM=I
 85   CONTINUE
      IF(NS.LE.4)GOTO88
      ISMO=0
      DO 86 I=1,NS
      IF(I.EQ.IM)GOTO86
      ISMO=ISMO+1
      SMO(ISMO)=SMOO(I)
 86   CONTINUE
      NS=NS-1
      DO 87 I=1,NS
 87   SMOO(I)=SMO(I)
      GOTO84
 88   DAXZ(2,K)=DM
C
 91   IF(IZSM.EQ.1)GOTO92
      DAXZ(1,K)=DAXZ(1,K+3)
      GOTO93
 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
      DO 5 I=2,NS
      IF(SMOO(I).LE.DM)GOTO5
      DM=SMOO(I)
      IM=I
 5    CONTINUE
      IF(NS.LE.4)GOTO11
      ISMO=0
      DO 6 I=1,NS
      IF(I.EQ.IM)GOTO6
      ISMO=ISMO+1
      SMO(ISMO)=SMOO(I)
 6    CONTINUE
      NS=NS-1
      DO 7 I=1,NS
 7    SMOO(I)=SMO(I)
      GOTO4
 11   DAXZ(3,K)=DM
```

```
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)DHP1,DHP2,DHP3
         DHP1=DHP1*1.
C       WRITE(6,1666)DHP1,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.
      PIL=DXT(43)*4.448222/.3048/.3048
      WML=DXT(41)*DXT(41)*1.4*8314.34*TIL/VEL/VEL
C
C
 30   IF(ALT.LT.XTRAN)GOTO50
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)GOTO70
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
```

```
C       WRITE(6,2323)DHP1,DHP2,DHP3
C      IF((DHP1.LE.TIME+.5).AND.(DHP3.GT.-9000.))AZ=DHP3
C
 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
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
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)
 2999   FORMAT(5(1X,E12.6))
C
C
       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
      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)
C
      NAMELIST/OPT/XL,YL,ALTSTRT,ALTEND,GAPSTRT,GAPEND,ISYMB,IGRID,ITPLT
     .,ITAG,AFIT,NFIT,XBIAS,ZBIAS,ARHO,IDMP,IATM,QFAC
C
      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=
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....  XAXIS   (A10)     ALPHA FOR INDEPENDENT VARIABLE
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
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)=10H
 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
      ALPHA(25)=10HHDIST
      UNITS(25)=10HKILOMETERS
C
      UNITS(3)=10HKILOMETERS
      UNITS(1)=10HSECX100
C
      ALPHA(26)=10HACCN
      UNITS(26)=10HLOG10MIC-G
C
      ALPHA(27)=10HACCA
      UNITS(27)=10HLOG10MIC-G
C
      ALPHA(28)=10HFLOD
C
      ALPHA(29)=10HFNOA
C
      ALPHA(30)=10HCDB
C
      ALPHA(31)=10HCXELFM
      UNITS(31)=10H
C
      ALPHA(32)=10HCZELFM
      UNITS(32)=10H
C
      ALPHA(33)=10HPN
      UNITS(33)=10HLOG10
C
      ALPHA(34)=10HCAB1
C
      ALPHA(35)=10HXKN2
      UNITS(35)=10HLOG10
C
      ALPHA(36)=10HTW
      UNITS(36)=10HK
```

```
C
      ALPHA(37)=10HAMACH
C
      ALPHA(38)=10HACCNP
      UNITS(38)=10HLOG10MIC-G
C
      ALPHA(39)=10HCN2
C
      ALPHA(40)=10HCNB1
C
      ALPHA(41)=10HRHON/RHOA
C
      ALPHA(42)=10HCAP
C
      ALPHA(43)=10HGAELE
      UNITS(43)=10HGRAPHUNITS
C
      ALPHA(44)=10HGABFL
      UNITS(44)=10HGRAPHUNITS
C
      ALPHA(45)=10HGALPH
      UNITS(45)=10HGRAPHUNITS
C
      ALPHA(46)=10HCA2
      ALPHA(47)=10HRHO/R76
C
C
      ALPHA(48)=10HRHO
      UNITS(48)=10HKG/M3
C
      ALPHA(49)=10HCA1
C
      ALPHA(50)=10HCN1
C
      ALPHA(51)=10HFLOD1
C
       ALPHA(52)=10HMW76
       UNITS(52)=10HKG/KMOLE
C
      ALPHA(53)=10HCXBFFM
C
      ALPHA(54)=10HCZBFFM
C
      ALPHA(55)=10HPDIFF
C
      ALPHA(56)=10HCAB2
      ALPHA(57)=10HCNB2
      ALPHA(58)=10HCDB2
C
C
C
      ALPHA(59)=10HXKN
      UNITS(59)=10H(LOG10)
C
      ALPHA(60)=10HFLOD2
C
      ALPHA(63)=10HAELE
      UNITS(63)=10HDEG
C
C     SET  CONSTANTS
C
      SS6=2690.*.3048*.3048
      PI=4.*ATAN(1.)
      PIS=PI/180.
C
C
```

```
C KN VS RHO   62ATM  KN=10^A*RHO^B    (RHO LBM/FT3)

      AKN=-7.7176662-1.5973
      BKN=-.99057554
C
C
C              EXP(-A2ZE*(B2ZE-LOG10(KN))^C2ZE)      KN < B2ZE
      A2ZE=.2262
      B2ZE=1.2042
      C2ZE=1.8410
      A1ZE=.2998
      B1ZE=1.3849
      C1ZE=1.7128
C
C
C
C************************************
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 .....*/
     .(3(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
C
C
C
C READ / WRITE  NAMELIST OPT
    5 CONTINUE
C
      XL=4.
      YL=4.
      ALTSTRT=180000.
      ALTEND=60000.
      GAPSTRT=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-------------------------------------------------------------------
      IF(EOF(5)) 260,666
  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
C IDENTIFY DATA WITH ALPHA INFO.
 7    DO 9 I=1,NWDS
         IF(XAXIS.EQ.ALPHA(I)) IX=I
 9       IF(YAXIS.EQ.ALPHA(I)) IY=I
      IF(IX.EQ.0)GO TO 4009
      IF(IY.EQ.0)GO TO 4010
  904 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)GOTO650
C
C  TIME, ALTITUDE SCALING FOR PLOTS
C
      ITIME=1
      IALT=3
      ALTEND=ALTEND/1000.
      ALTSTRT=ALTSTRT/1000.
C
      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
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)GOTO901
      NOX=XL
      NOY=YL
      CALL GRID(0.,0.,1.,1.,NOX,NOY)
      GOTO902
  901 CONTINUE
  902 CONTINUE
      IF(NTAG.NE.0)GOTO903
      IF(ITPLT.EQ.1)GOTO931
      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)
      Y00=YL/2.
      CALL CALPLT(0.,Y00,3)
      CALL CALPLT(XL,Y00,2)
 932  CALL CHARWH(W1,ZA,ZB,.15,TITLES,9)
      Y1=-1.5
      X1=(XL-W1)/2.0
      CALL CHARACT(X1,Y1,.15,TITLES,0.,9)
 903  CALL CHARWH(W1,ZA,ZB,.15,BCDY,20)
      X1=-(.5+W1)
      Y1=YL+.5-.25*NTAG
      CALL CHARACT(X1,Y1,.15,BCDY,0.,20)
C
      WRITE(3)BCDX,BCDY,TITLES(1)
      NTCT=0
C
C
C
C......... READ DATA FROM TAPE1
C
C
      IF(IDMP.EQ.1)GOTO200
C
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 REDORD 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
C
 200   READ(1)(DAT(I),I=1,24)
```

```
      IF(EOF(1))820,205
C
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)
 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
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
C  LEAVE GAP IN DATA
 209  IF(DAT(3).GT.GAPEND.AND.DAT(3).LT.GAPSTRT)GOTO200
C
C
C  FLIGHT DATA ON TAPE1
C
C     DAT(1)=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/.3048
      CALL AT62(AALT,ANS)
      R62=ANS(1)
      PI62=ANS(2)
      TI62=ANS(3)
      AMACH62=VEL/.3048/ANS(4)
      WM62=8314.34*10.7639*TI62*R62/PI62
      R62=R62*515.3788
      PI62=PI62*47.88026
C  76-STANDARD ATM. ALTITUDE MODEL
      AALT=ALT/.3048
      CALL AT76(AALT,ANS)
      R76=ANS(1)
      PI76=ANS(2)
      TI76=ANS(3)
      AMACH76=VEL/.3048/ANS(4)
      WM76=8314.34*10.7639*TI76*R76/PI76
```

```
      R76=R76*515.3788
      PI76=PI76*47.88026
C
C
C PREPROCESS ACCELEROMETRY DATA
      AX=DAT(23)+XBIAS*9.81/1.0E06
      AZ=DAT(24)+ZBIAS*9.81/1.0E06
      IF(IDMP.EQ.1)GOTO210
      IF(ALT.GT.AFIT)AX=-10.**(BAC(1,1)+BAC(2,1)*ALT+BAC(3,1)*ALT*ALT
     .+BAC(4,1)*ALT*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*ALT)
      IF(AX.GE.0.)GOTO200
      IF(AZ.GE.0.)GOTO200
C
C 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)
C
C FORCE RATIOS
      FNOA=AZ/AX
      ALPHAR=ALPH*PIS
      SINA=SIN(ALPHAR)
      COSA=COS(ALPHAR)
      TANA=SINA/COSA
      FLOD=(AZ/AX-TANA)/(1.+TANA*AZ/AX)
C
C
C DATA BOOK PROFILES
      XKN62=10.**AKN*(R62/16.01846)**BKN
      XKN62=ALOG10(XKN62)
      ZDB=0.
      IF(XKN62.LT.-2.99)GOTO335
      ZDB=1.
      IF(XKN62.GT..99999)GOTO335
      ZDB=SIN(1.1781+.3927*XKN62)**2
 335  CONTINUE
      ATLT=AALT/1000.
      VBARDB=(-12.62977)+ATLT*.1951838
     .-ATLT*ATLT*.00156941
     .+ATLT*ATLT*ATLT*6.586803E-06
     .-ATLT*ATLT*ATLT*ATLT*1.4461354E-08
     .+ATLT*ATLT*ATLT*ATLT*ATLT*1.586619E-11
     .-ATLT*ATLT*ATLT*ATLT*ATLT*ATLT*6.8824603E-15
C
C  HORIZ. PATH DIST
      RADE=6370.
      PH=(90.-PLAT)*PIS
      TH=PLONG*PIS
      X2=RADE*SIN(PH)*COS(TH)
      Y2=RADE*SIN(PH)*SIN(TH)
      Z2=RADE*COS(PH)
      IF(L.EQ.0)GOTO331
      HDIST=HDIST+SQRT((X2-X1)**2+(Y2-Y1)**2+(Z2-Z1)**2)
      GOTO332
 331  HDIST=0.
 332  X1=X2
      Y1=Y2
      Z1=Z2
C
C
C 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
C MEAN MOL WGT AND FRESTRM TEMP USED IN SCALING PARA.
C     WWML=WML
C     WWML=R62*(8314.34)*TI62/PI62
      WWML=28.96
      IF(ALT.GT.88000.)WWML=28.96-(ALT-88000.)/10000.
      TI=TI62
      IF(IATM.EQ.1)GOTO218
      TI=TIL
      WWML=RL*(8314.34)*TIL/PIL
      WWML=WM76
 218  CONTINUE
C
C DENSITY INDEP. SCALING PARAMETER VARIABLES
      XLR=32.77
      AP=390.83*SINA
      AW=1200.
      SA=12.058*SQRT(SINA)
      SRAT=VEL/SQRT(2.0*8314.34*TI/WWML)
      AMACH=SRAT*SQRT(2.0/1.4)
      VISC=(TI/273.1)**1.5*6.584E-02*.0672/(TI+110.6)
      RENOR=(1./16.01846)*1290.3/12.*(VEL/.3048)/VISC
      TPR=(0.468+0.532*TW/TI+0.195*AMACH*AMACH/5.)*TI
      CPR=SQRT(TPR/TI)*(TI+122.1*10.**(-5./TI))/
     .(TPR+122.1*10.**(-5./TPR))
      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.)
      AMSOM=SQRT(((AK-1.)+2./AM2)/(2.*AK*AM2-AK+1.))
      TAS=TSOT*TI
      VISCS=(TAS/273.1)**1.5*6.584E-02*.0672/(TAS+110.6)
      RENORS=RENOR*AMSOM*VISC/VISCS*SQRT(TSOT*RSOR)
      XKNXRS=XKNXR*AMSOM*RENOR/RENORS
C
C CALCULATE FM AND CONT. COEFFS.
      TWOTI=TW/TI
      CALL CXCZDB(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                 CZC = CZ-CONT.       CZC+DELZ = CZ-FM.
C
      FLODFM=(CZC+DELZ)/(CXC+DELX)
      FLODFM=(FLODFM-TANA)/(1.+TANA*FLODFM)
      FLODC=(CZC/CXC-TANA)/(1.+TANA*CZC/CXC)
      FLODB=(FLOD-FLODC)/(FLODFM-FLODC)
C
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     IF(IX.EQ.3.AND.IY.EQ.28)GOTO257
      IF(IX.EQ.3.AND.IY.EQ.29)GOTO257
      IF(IX.EQ.40.AND.IY.EQ.3)GOTO257
```

```
      IF(IX.EQ.49.AND.IY.EQ.3)GOTO257
      IF(IX.EQ.50.AND.IY.EQ.3)GOTO257
      IF(IX.EQ.51.AND.IY.EQ.3)GOTO257
C
C
C
C
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=XKNXR/RHO1
      XKP=-XKU/RHO1
      XKU=ALOG10(XKU)
       XKU1=XKU
C     IF(XKU.GT.1.)GOTO413
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.B1ZE)GOTO413
      ZETAE=EXP(-A1ZE*(B1ZE-XKU)**C1ZE)
      ZETEP=ZETAE*A1ZE*C1ZE*(B1ZE-XKU)**(C1ZE-1.)/10.**XKU*XKP*.4343
 413  CONTINUE
      FUNC=RHO1*(DELZ*ZETAE+CZC)-AZ*CNST
      FNCP=DELZ*(ZETAE+RHO1*ZETEP)+CZC
      DELT=(-FUNC)/FNCP
      RHO1=RHO1+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
C                  AERO CALCULATIONS
      CAP=(-AX)*CNST/RHO1
      CABP=(-CAP-CXC)/DELX
C                  SCALING PARAMETERS
      XKN1=ALOG10(XKNXR/RHO1)
      VBAR1=AMACH*SQRT(CPR/(RHO1*RENOR))
C                  AERO MODELS
      CNB1=1.
      IF(XKN1.LT.B1ZE)CNB1=EXP(-A1ZE*(B1ZE-XKN1)**C1ZE)
      CAB1=1.
      IF(XKN1.LT.B2ZE)CAB1=EXP(-A2ZE*(B2ZE-XKN1)**C2ZE)
      CA1=-CXC-DELX*CAB1
      CN1=-CZC-DELZ*CNB1
      CD1=CN1*SINA+CA1*COSA
      CDB1=-CD1-CZC*SINA-CXC*COSA
      CDB1=CDB1/(DELZ*SINA+DELX*COSA)
      FLOD1=(CN1/CA1-TANA)/(1.+TANA*CN1/CA1)
C
C
C
C
C
C COMPONENT DERIVED DENSITY USING CA-FLIGHT DER.
      RHO2=RGUESS
      NCON=0
```

```
C                         CONVERGING NEWTON ROOT SOLVER
 417  XKU=XKNXR/RHO2
      IF(XKU.LE.0)XKU=.00001
      XKP=-XKU/RHO2
      ZETAE=1.
      ZETEP=0.
      XKU=ALOG10(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
C     WRITE(6,*) RHO2,FUNC,FNCP,DELT
C     WRITE(6,*) AMACH,XKNXR,RENOR
      NCON=NCON+1
      IF(ABS(DELT).LE.(.00001*RGUESS))GOTO416
      IF(NCON.LE.11)GOTO417
 416  CONTINUE
C
C                AERO CALCULATIONS
C     WRITE(6,*) XKNXR,RHO2,AMACH,RENOR,CPR
      CNP=(-AZ)*CNST/RHO2
      CNBP=(-CNP-CZC)/DELZ
C                  SCALING PARAMETERS
      XKN2=ALOG10(XKNXR/RHO2)
C     WRITE(6,*) XKN2
      VBAR2=AMACH*SQRT(CPR/(RHO2*RENOR))
C                AERO MODELS
      CAB2=1.
      IF(XKN2.LT.B2ZE)CAB2=EXP(-A2ZE*(B2ZE-XKN2)**C2ZE)
      CNB2=1.
      IF(XKN2.LT.B1ZE)CNB2=EXP(-A1ZE*(B1ZE-XKN2)**C1ZE)
      CA2=-CXC-DELX*CAB2
      CN2=-CZC-DELZ*CNB2
      CD2=CN2*SINA+CA2*COSA
      CDB2=-CD2-CZC*SINA-CXC*COSA
      CDB2=CDB2/(DELZ*SINA+DELX*COSA)
      FLOD2=(CN2/CA2-TANA)/(1.+TANA*CN2/CA2)
      IF(ALT.GE.AFIT)FLOD=FLOD2
      ACCNP=ALOG10((CNB2*DELZ+CZC)/(-CNST)*RHO2*1.0E06/9.81)
C
C               COMPOSITE DENSITY
      IF(ALT.GT.ARHO)GOTO419
      RHO=RHO1
      CDB=CDB1
      VBAR=VBAR1
      XKN=XKN1
       PKN=XKNXR/RHO1
      GOTO440
 419  RHO=RHO2
      CDB=CDB2
      VBAR=VBAR2
      XKN=XKN2
       PKN=XKNXR/RHO2
 440  CONTINUE
C
C
      REP=RENOR*RHO*SA/XLR
      TPP=.2*TO+.5*TW
      PN=SQRT(REP*(TI/TPP)**OMEG)
      BMMN=XKNXRS/RHO*RSOR*12.058/XLR
      VISCPR=(TPR/273.1)**1.5*6.584E-02*.0672/(TPR+110.6)
```

```
      VISCPP=(TPP/273.1)**1.5*6.584E-02*.0672/(TPP+110.6)
      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=VISCPR/VISC*TI/TPR
      C22=VISCPP/VISC*TI/TPP
      C23=VISCS/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
C
C
C
C
C AERO COEFFICIENTS USING DATA BOOK
C     CNDB=-CZC-DELZ*CNBDB
C     CADB=-CZDB-(DELX+CXC-CZDB)*CNBDB
C     CABDB=(-CADB-CXC)/DELX
C
C     IF(VBAR.LT..08)CABDB=-1.5285*(VBAR-.01)/DELX
C     IF(VBAR.LT..08)CADB=-CXC+1.5285*(VBAR-.01)
C     IF(XKN.LT.-3.)CABDB=-.2
C     IF(XKN.LT.-3.)CADB=-CXC
C     CABDB=-1.5285*(VBAR-.01)/DELX
C     CADB=-CXC+1.5285*(VBAR-.01)
C     CDDB=CNDB*SINA+CADB*COSA
C     CDBDB=-CDDB-CZC*SINA-CXC*COSA
C     CDBDB=CDBDB/(DELZ*SINA+DELX*COSA)
C
C
C
C LOAD OUTPUT
      DAT(30)=CDB
      DAT(31)=CXELFM
      DAT(32)=CZELFM
      DAT(33)=ALOG10(PN)
      DAT(34)=CAB1
      DAT(35)=XKN2
      DAT(59)=XKN1
      DAT(60)=XKU1
      DAT(38)=XKU2
      DAT(42)=CAP
      DAT(46)=CA2
      DAT(47)=RHO/R76
      DAT(48)=RHO
      DAT(56)=CAB2
      DAT(57)=CNB2
      DAT(58)=CDB2
C
 257  DAT(ITIME)=DAT(ITIME)
      DAT(IALT)=DAT(IALT)/1000.
      DAT(25)=HDIST
      DAT(36)=TW
      DAT(37)=AMACH
      DAT(39)=CN2
C     IF(RL.GT.0.0)DAT(39)=ALOG10(RL)
      DAT(41)=RHO1/RHO2
      DAT(43)=RHO/R76
```

```
      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)=CNB1
      DAT(50)=CN1/CA1
      DAT(49)=CN2/CA2
      DAT(51)=FLOD1
       DAT(52)=WM76
C
C
 253  L=L+1
C
C PLOT  DAT(IX) , DAT(IY)
      WRITE(3)DAT(IX),DAT(IY)
C      IF(DAT(IY).GT.YMAX)DAT(IY)=YMAX
C      IF(DAT(IY).LT.YMIN)DAT(IY)=YMIN
C      IF(DAT(IX).GT.XMAX)DAT(IX)=XMAX
C      IF(DAT(IX).LT.XMIN)DAT(IX)=XMIN
      X=(DAT(IX)-XMIN)/XSF
      Y=(DAT(IY)-YMIN)/YSF
      IF(ISYMB.EQ.1)GOTO10
      IF(L.EQ.1)CALL CALPLT(X,Y,3)
      CALL CALPLT(X,Y,2)
      GOTO15
 10   CALL POINT(X,Y)
 15   CONTINUE
C
C   WRITE SELECTED DATA TO A FORMATTED OUTPUT TAPE
C
         IF(JSTOP.EQ.1) GOTO 200
            WRITE(4,777)DAT(1),DAT(3),DAT(23),DAT(24),DAT(29)
C     C   DAT(50),DAT(49),DAT(59),DAT(35)
C           WRITE(4,778)DAT(4),DAT(5),AZ,AX,DAT(48),PKN
C4       WRITE(4,780)DAT(1),DAT(3),DAT(2),DAT(6),DAT(15),DAT(14),
C4     * DAT(47),AZ,AX,DAT(48),PKN
            IF(KCNT.GT.1)GOTO 776
            WRITE(6,668)ALPHA(1),ALPHA(3),ALPHA(2),ALPHA(6),ALPHA(15),
     *                  ALPHA(14),ALPHA(47),ALPHA(5),ALPHA(24),ALPHA(23),
     *                  ALPHA(48)
            WRITE(6,669)UNITS(1),UNITS(3),UNITS(2),UNITS(6),UNITS(15),
     *                  UNITS(14),UNITS(47),UNITS(5),UNITS(24),UNITS(23),
     *                  UNITS(48)
          KCNT=KCNT+1
 776      CONTINUE
C         WRITE(6,779)DAT(1),DAT(3),PKN,DAT(6),DAT(15),DAT(14),
C     *                  DAT(47),DAT(5),AZ,AX,DAT(48)
 668      FORMAT(11(2X,A10))
 669      FORMAT(11(2X,A10)/)
 777      FORMAT(5(1X,E12.5))
 778      FORMAT(1X,6E12.6,56X)
 779      FORMAT(11E12.5)
C4 780     FORMAT(1X,11E12.6)
C
C
      IF(L.LT.2500)GOTO200
C
C
 820  CONTINUE
       JSTOP=1
```

```
      WRITE(6,1012)L
      WRITE(3)-8888.,-8888.
      NTAG=NTAG+1
      L=0
      REWIND 1
      IF(ITAG.EQ.1)GOTO5
      NTAG=0
      K=K+1
       CALL NFRAME(XFR,YFR)
      CALL CALPLT(5.,3.,-3)
      GO TO 5
C
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.*/*0HAVE A NICE DAY.*)
      CALL CALPLT(0.,0.,999)
 2000 FORMAT(//50X*PLOT NO. *I3//)
 1004 FORMAT(/1X,*INDEPENDENT VARIABLE SELECTED *A10,* BETWEEN *F10.3,
     .* AND *F10.3)
 1005 FORMAT(/1X,*DEPENDENT VARIABLE SELECTED *A10,* BETWEEN *F10.3,
     .* AND *F10.3)
 1012 FORMAT(//50X,I6,* POINTS PLOTTED*/)
 1011 FORMAT(/1X,*X-AXIS LENGTH *F10.3,*  Y-AXIS LENGTH *F10.3)
 1017 FORMAT(/1X,*NO PLOT GRID OPTION SELECTED*)
 1018 FORMAT(/1X,*PLOT GRID OPTION SELECTED,GRID SPACING 1"*)
 1021 FORMAT(/1X*SEARCH AEROBET FILE BETWEEN *F10.3,* AND *F10.3,
     .* METERS ALTITUDE*)
      STOP
C
C
C**   SAME VARIABLE ON X- AND Y-AXES
C
  650 WRITE(6,655) XAXIS
  655 FORMAT(//1X*PLOTTING *,A10,* AGAINST ITSELF IS NOT ALLOWED.*/)
      GO TO 820
C
C
 4009 CONTINUE
      WRITE(6,4020)
 4020 FORMAT(/1X,*X-AXIS ALPHA UNDEFINED---PLOT TERMINATED*/)
      GO TO 820
C
 4010 CONTINUE
      WRITE(6,4021)
 4021 FORMAT(/1X,*Y-AXIS ALPHA UNDEFINED---PLOT TERMINATED*/)
      GO TO 820
C
C
  683 CONTINUE
       WRITE(6,684)XSF,YSF
  684 FORMAT(//1X,*PLOT INHIBITED-SCALE FACTOR INDETERMINATE  XSF
     .*F10.3,*  YSF  *F10.3/)
      GO TO 820
C
  386 CONTINUE
      WRITE(6,387)
  387 FORMAT(/1X,*PLOT INHIBITED-Y1 SCALE FACTOR 0.*/)
      GO TO 820
C
C**   EOF WHILE TRYING TO READ HEADER ON TAPE1
C
  500 WRITE(6,505)
```

```
  505 FORMAT(1X*E-O-F ENCOUNTERED WHILE TRYING TO READ HEADER ON TAPE1
     +, PROGRAM HALTED.*)
  504 PRINT*,"ERROR"
      STOP
      END
      SUBROUTINE CXCZDB(AL,ABFL,AELE,AZZ,BZZ,CZZ,DZZ,CZP,SR,TWT,KHRO,
     *   CXELFM,CZELFM,CXBFFM,CZBFFM)
      PI=4.*ATAN(1.)
      AR=AL*PI/180.
      NDER=0
       NCS=0
       KHRO=4
C     PDIF=SQRT(PI*TWT)/(2.*SR)
      PDIF=SQRT(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=-1.6095
       CZFM=-1.3725
C        IF(NDER.EQ.1)GOTO11
       CXFM=CXFM*(.27568+.02879405*AL-.00015907143*AL*AL
     .-2.733333E-06*AL*AL*AL)
      CZFM=CZFM*(.104-.0038214286*AL+9.571429E-04*AL*AL
     .-7.5E-06*AL*AL*AL)
      CX=CX*(.29594402+.067083253*AL-.0023369244*AL*AL
     .+3.58036123E-05*AL*AL*AL-2.047015503E-07*AL*AL*AL*AL)
     . + .002*(AL-40.)
      CZ=CZ*(-.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
     .-2.733333E-06*AL*AL*AL)
      CZFM=CZFM*(.104-.0038214286*AL+9.571429E-04*AL*AL
     .-7.5E-06*AL*AL*AL)
      CX=CX*(.29594402+.067083253*AL-.0023369244*AL*AL
     .+3.58036123E-05*AL*AL*AL-2.047015503E-07*AL*AL*AL*AL)
     . + .002*(AL-40.)
      CZ=CZ*(-.22316013+.0248950704*AL+.000143729041*AL*AL)
     . + .02*(AL-40.)
C     IF(NCS.EQ.1)GOTO 11
      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
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)
      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.3) GOTO501
  401  CX=-.056
       CZ=-1.115
C     CXFM=CXFM*(.27568+.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    KTLFM
       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)+
     .(-1.17117E-5*AL*AL*AL)
      CXFM = -CXFM
C *******************************************************************
      CX=2.36E-4*AL*AL-1.8534E-2*AL+.416761
      CX=-CX
      CZ=-.04345*AL+.6230
      CXEL=.0002-.0008*AELE-.00013*AELE*AELE
     .+5.6E-04*AELE
C THIS IS AELE-CXEL FOR NEG AELE BASED UPON N/A STUDY AUG87
       IF(AELE.LE.0.)CXEL=-.0009*AELE-.00005*AELE*AELE
      IF (KHRO.EQ.4) GOTO501
  501 CONTINUE
C BASIC COEFF. VALUES (@ 40 DEG. )
C     CX=-1.55*VBAR-.0515
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=-1.205-.004
C THE FOLLOWING CZ IS THE FAD-26 VALUE
C     CZ=-1.1538
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.
C87   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
C67    CXFM=-1.50
C     CZFM=-1.75
C THIS CZFM IS THE PRE-OP DATA BOOK VALUE (AS RECEIVED*APRIL86*)
C     CZFM=-1.4567
C THIS IS THE CZFM BASED ON THE TEN FLIGHT L/D, ALPHA =40 STUDY.
C     CZFM=-1.55
C THIS IS THE CZFM BASED ON THE SPLIT DIFF
C     CZFM=-1.5
C     CZFM=-3.934*SIN(AR)*(SIN(AR)+PDIF)
C
C     IF(NDER.EQ.1)GOTO11
C ALPHA DERIVATIVES
C     CXFM=CXFM*(.27568+.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
```

```
C     .+3.58036123E-05*AL*AL*AL-2.047015503E-07*AL*AL*AL*AL)
C     . + .002*(AL-40.)
C THIS IS THE CAC UPDATE BASED ON FAD-26 AND CAO=.056 (APRIL87)
C        CX=2.36E-4*AL*AL-1.8534E-2*AL+.41976
C 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
C        CX=-CX
C      CZ=CZ*(-.22316013+.0248950704*AL+.000143729041*AL*AL)
C     . + .02*(AL-40.)
C THIS IS THE FAD-L/D CZ VALUE (APRIL87)
C        CZ=-.04345*AL+.6230
C
C
C        IF(NCS.EQ.1)GOTO 11
C CONTROL SURFACE DERIVATIVES
C      CXEL=-.00623119*AELE-9.7147619E-05*AELE*AELE
C      CXEL=-.00147435434*AELE-1.328865546E-04*AELE*AELE
C     .+8.698039216E-07*AELE*AELE*AELE
C      CXEL=.0002-.0008*AELE-.00013*AELE*AELE
C     .+5.6E-04*AELE
C      IF(AELE.LE.0.0)CXEL=(-.0026*AELE)-.00016*AELE*AELE
C THIS IS AELE-CXEL FOR NEG AELE BASED UPON N/A STUDY AUG87
C67     IF(AELE.LE.0.)CXEL=-.0009*AELE-.00005*AELE*AELE
       CZEL=-.0057382024*AELE-1.1632619E-04*AELE*AELE
       CXBF=-.0001659109-.0003626653*ABFL
      .-2.2857795E-05 * ABFL*ABFL
      .+1.024339905E-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)
C      CPG=SIN(G)*SIN(G)*(1.+G*2./PI)
C      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
       CFG=SIN(G)*COS(G)
C      CPG=SIN(G)*SIN(G)*(1.+G*2./PI)
       CPG=SIN(G)*(SIN(G)+PDIF)
       CXBFFM=-135.75/2690.*(COS(ABR)*(CFG-CFA)+SIN(ABR)*(CPG-CPA))
       CZBFFM=-135.75/2690.*(COS(ABR)*(CPG-CPA)-SIN(ABR)*(CFG-CFA))
       CZBF=.0001086855-.0020996556*ABFL
      .-3.13694583E-05 * ABFL*ABFL
      .+1.353946759E-06 * ABFL*ABFL*ABFL
C        CXEL=0.
C        CXBF=0.
C        CZEL=0.
C        CZBF=0.
C        CXBFFM=0.
C        CXELFM=0.
C        CZBFFM=0.
C        CZELFM=0.
       CX=CX+CXBF+CXEL
       CZ=CZ+CZBF+CZEL
C      CXFM=CXFM+CXBFFM+CXELFM
        CXFM=CXFM
C      CZFM=CZFM+CZBFFM+CZELFM
        CZFM=CZFM
C
 11    CONTINUE
C TRANSITION BRIDGING FORMULA
       AZZ=CZ
```

```
      BZZ=CZFM-CZ
      CZZ=CX
      DZZ=CXFM-CX
C     CZP=CZBFFM+CZFM
      RETURN
      END
      SUBROUTINE BETA(UI,RI,WM,TI,TW,BE)
      BE=SQRT(8314.34*TW/WM)/UI+1.
      BE=5.0016E-07*1./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)GOTO2
      IF(ABS(DELT).GT.(.0001*HH))GOTO1
    2 P=ANS(2)
      T=ANS(3)
      A=ANS(4)
      RETURN
      END
```

References

1. *U.S. Standard Atmosphere, 1976.* NOAA, NASA, and U.S. Air Force, Oct. 1976.

2. Blanchard, Robert C.: Rarefied Flow Lift-to-Drag Measurements of the Shuttle Orbiter. *ICAS Proceedings 1986—15th Congress of the International Council of the Aeronautical Sciences, Volume 2,* P. Santini and R. Staufenbiel, eds., 1986, pp. 1421–1430. (Available as ICAS-86-2.10.2.)

3. Blanchard, Robert C.; and Buck, Gregory M.: Rarefied-Flow Aerodynamics and Thermosphere Structure From Shuttle Flight Measurements. *J. Spacecr. & Rockets,* vol. 23, no. 1, Jan.–Feb. 1986, pp. 18–24.

4. Blanchard, R. C.; Hinson, E. W.; and Nicholson, J. Y.: Shuttle High Resolution Accelerometer Package Experiment Results: Atmospheric Density Measurements Between 60 and 160 km. *J. Spacecr. & Rockets,* vol. 26, no. 3, May–June 1989, pp. 173–180.

5. Blanchard, R. C.; and Larman, K. T.: Rarefied Aerodynamics and Upper Atmospheric Flight Results From the Orbiter High Resolution Accelerometer Package Experiment. *Technical Papers—AIAA Atmospheric Flight Mechanics Conference,* Aug. 1987, pp. 139–150. (Available as AIAA-87-2366.)

6. Blanchard, R. C.; and Rutherford, J. F.: Shuttle Orbiter High Resolution Accelerometer Package Experiment: Preliminary Flight Results. *J. Spacecr. & Rockets,* vol. 22, no. 4, July–Aug. 1985, pp. 474–480.

7. Buck, Gregory M.; and Blanchard, Robert C.: Rarefied Aerodynamics and Upper Atmosphere Density From Multiple Orbiter Flight Measurements. AIAA-86-0269, Jan. 1986.

8. Thompson, J. M.; Russell, J. W.; and Blanchard, R. C.: Methods for Extracting Aerodynamic Accelerations From Orbiter High Resolution Accelerometer Package Flight Data. *A Collection of Technical Papers—AIAA Atmospheric Flight Mechanics Conference,* American Inst. of Aeronautics and Astronautics, Aug. 1987, pp. 130–138. (Available as AIAA-87-2365.)

9. Thomas, D. J.: *HIRAP S/N 001 Calibration Analysis Report.* KTR-105 (Contract NAS9-16791), KMS Fusion, Inc., Mar. 1983.

10. Findlay, John; Kelly, Mel; and McConnell, Judy: *An AEROdynamic Best Estimate Trajectory File (AEROBET) for NASA Langley Research Center Shuttle Investigations.* AMA Rep. No. 82-9 (Contract NAS1-16087), Analytical Mechanics Associates, Inc., Mar. 1982.

11. *Aerodynamic Design Data Book. Volume 1—Orbiter Vehicle, 1978: Part 1,* NASA CR-171859, 1978 (revised Apr. 1982). *Part 2,* NASA CR-171854, 1978 (revised Apr. 1982).

12. *HIRAP S/N 001 Recalibration Analysis Report.* KTR-133 (Contract NAS9-16791), KMS Fusion, Inc., Apr. 1986.

13. *High Resolution Accelerometer Package (HiRAP), PN 6489-300001-1, SN 001—Dynamic and Static Calibration Data.* Rep. No. 6489-928005 (Contract NAS9-16791), Bell Aerospace Textron, Aug. 1988.

14. Thomas, D. J.: *Proposed Algorithms for Computing HIRAP Temperatures and Temperature Effect Corrections.* KTR-102 (Contract NAS9-15588), KMS Fusion, Inc., Aug. 31, 1982.

15. *High Resolution Accelerometer Package (HiRAP), Bell Aerospace Textron P/N 6489-300001-1, Serial No. 001—Calibration Data.* Rep. No. 6489-928004 (Contract NAS9-16791), Bell Aerospace Textron, Oct. 15, 1985.

16. *HIRAP S/N 002 Recalibration Analysis Report.* KTR-109 (Contract NAS9-16791), KMS Fusion, Inc., Mar. 1985.

17. Thomas, D. J.: *HIRAP S/N 002 Calibration Analysis Report.* KTR-106 (Contract NAS9-16791), KMS Fusion, Inc., Apr. 1983.

Table 1. STS HiRAP Missions

[The number used in this report to identify each mission is given with instrument, orbiter name, and entry date]

Mission	System data file number	Instrument	Orbiter	Entry date
STS-06	6	S/N 001	*Challenger*	4/9/83
STS-07	7	S/N 001	*Challenger*	6/24/83
STS-08	8	S/N 001	*Challenger*	9/5/83
STS-09	9	S/N 002	*Columbia*	12/8/83
STS-41B	11	S/N 001	*Challenger*	2/11/84
STS-41C	13	S/N 001	*Challenger*	4/13/84
STS-51B	24	S/N 001	*Challenger*	5/6/85
STS-51F	26	S/N 002	*Challenger*	8/6/85
STS-61A	30	S/N 002	*Challenger*	11/6/85
STS-61C	32	S/N 001	*Columbia*	1/18/86

Table 2. HiRAP Ground Calibration Scale Factors

Instrument and flight	Scale factor, V/μg, for acceleration along—		
	X-axis	Y-axis	Z-axis
S/N 001 before recalibration (STS-06, 07, 08, 41B, 41C, and 51B)	-1.247237×10^{-3}	1.253821×10^{-3}	1.26810×10^{-3}
S/N 001 after recalibration (STS-61C)	-1.24720×10^{-3}	1.269482×10^{-3}	-1.256565×10^{-3}
S/N 002 before recalibration (STS-09)	-1.24857×10^{-3}	1.269482×10^{-3}	-1.256565×10^{-3}
S/N 002 after recalibration (STS-51F, 61A)	-1.250035×10^{-3} $-(1.283124 \times 10^{-6})V_T^2$ $+(0.1437924 \times 10^{-6})V_T^3$	1.271533×10^{-3} $-(1.344372 \times 10^{-6})V_T$ $+(0.04102193 \times 10^{-3})V_T^2$	-1.253671×10^{-3} $-(2.584656 \times 10^{-6})V_T$ $+(1.044413 \times 10^{-6})V_T^2$ $-(0.1370558 \times 10^{-6})V_T^3$

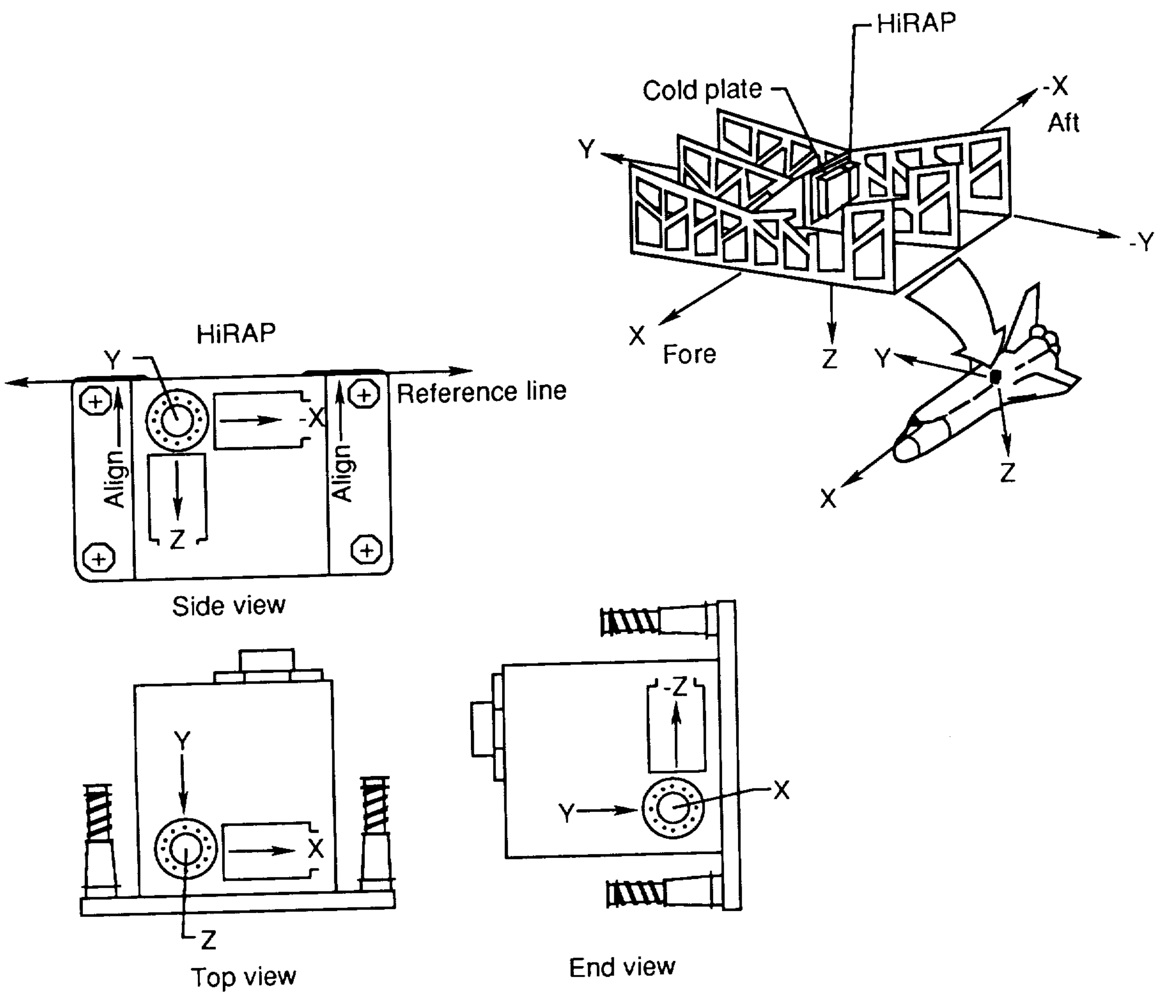


Figure 1. Arrangement of HiRAP accelerometer triad in the Space Shuttle orbiter vehicle.

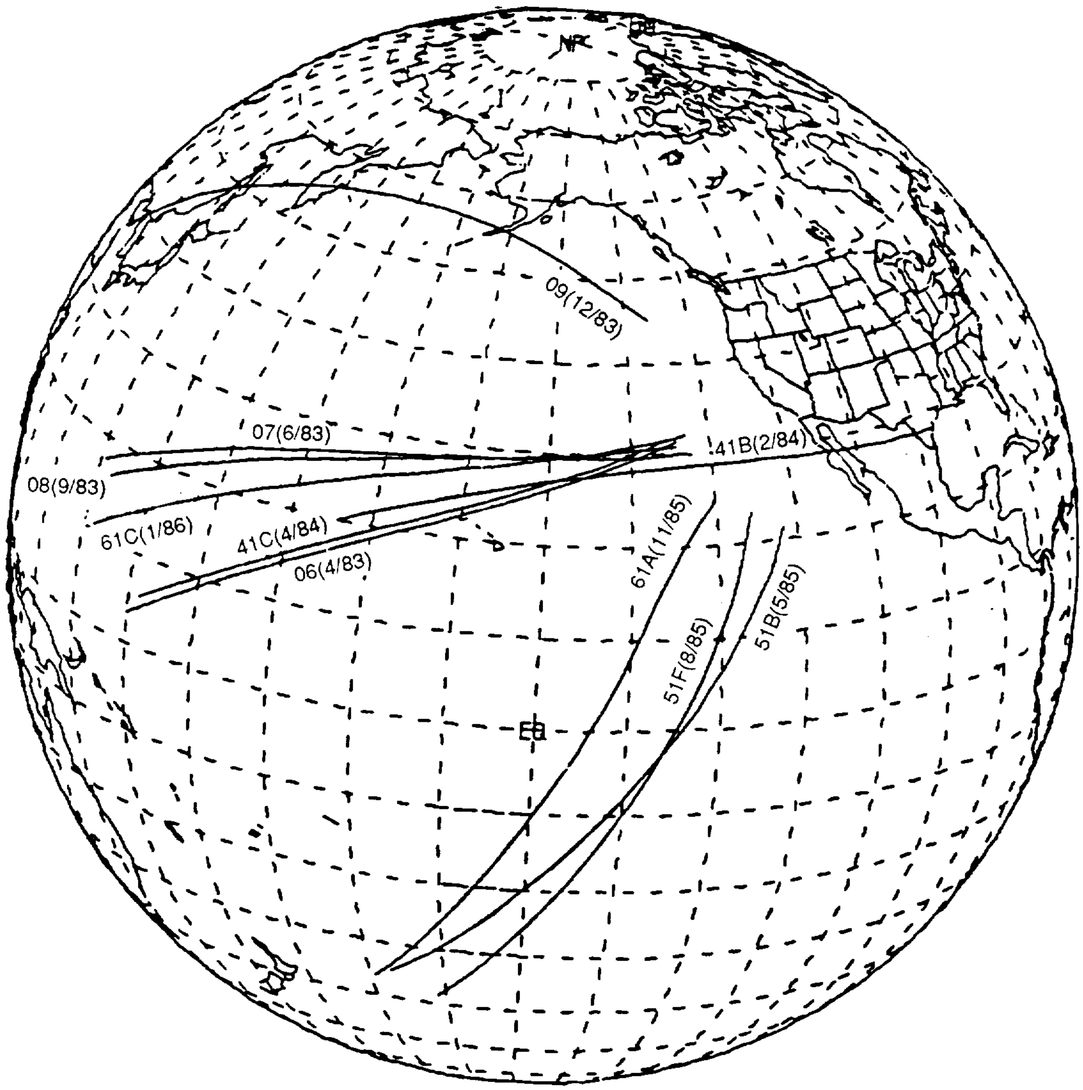
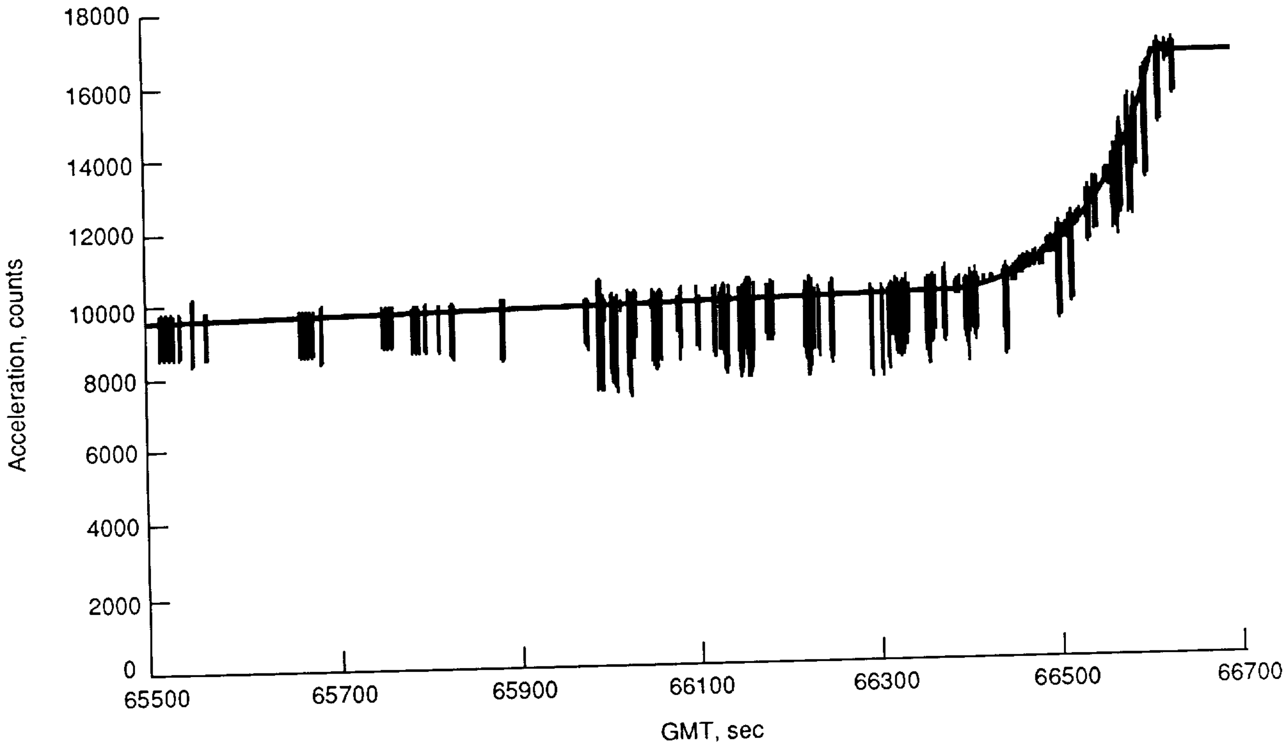
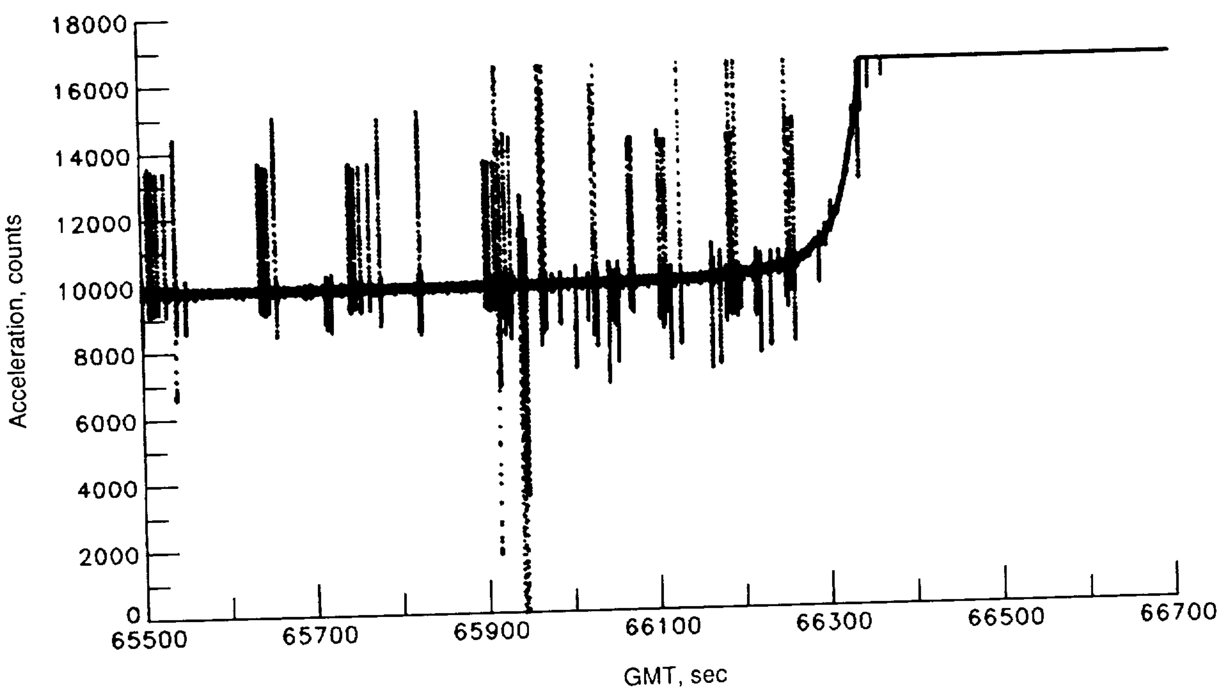


Figure 2. Shuttle reentry trajectories overlaid on Earth globe. Mission number and flight date (month/year) shown for each flight.

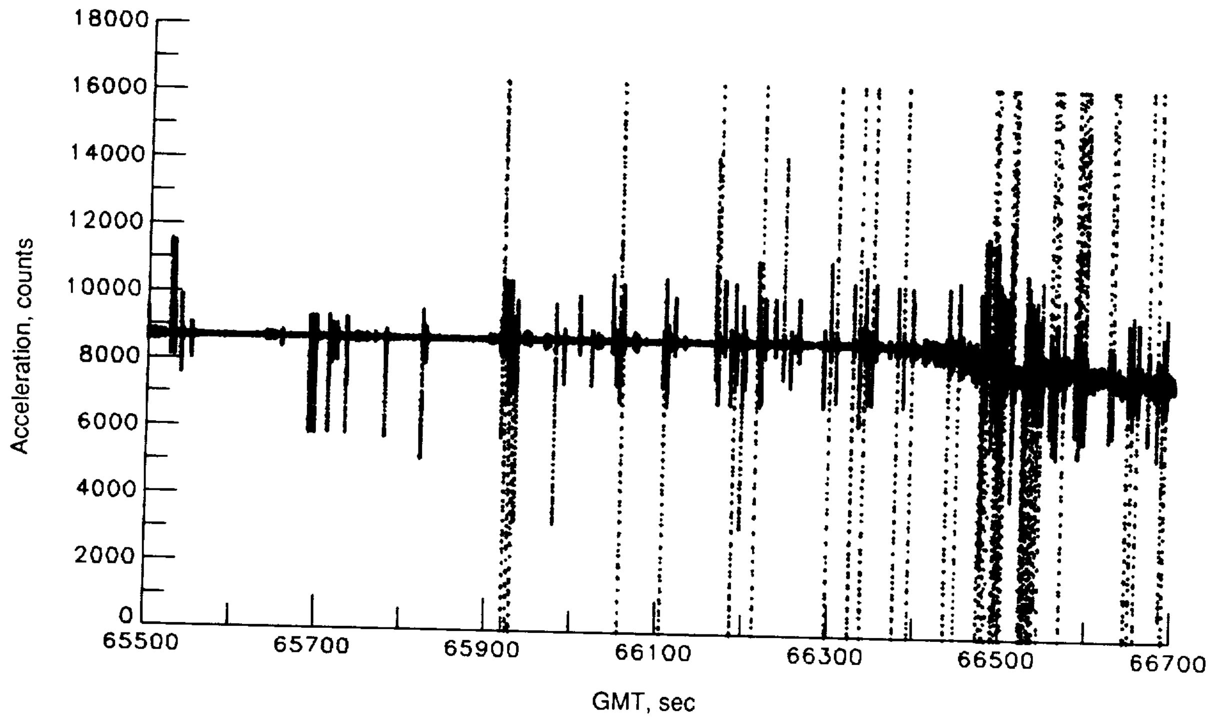


(a) X-axis acceleration counts.

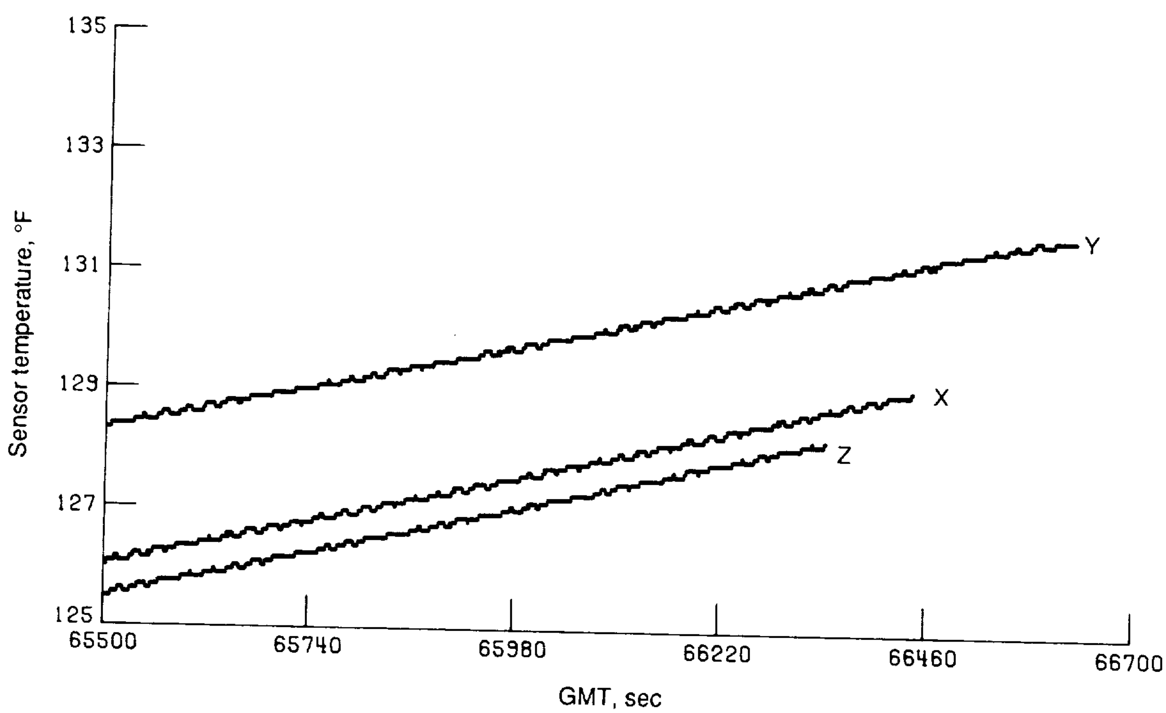


(b) Z-axis acceleration counts.

Figure 3. Acceleration counts and sensor temperature versus time for STS-06.

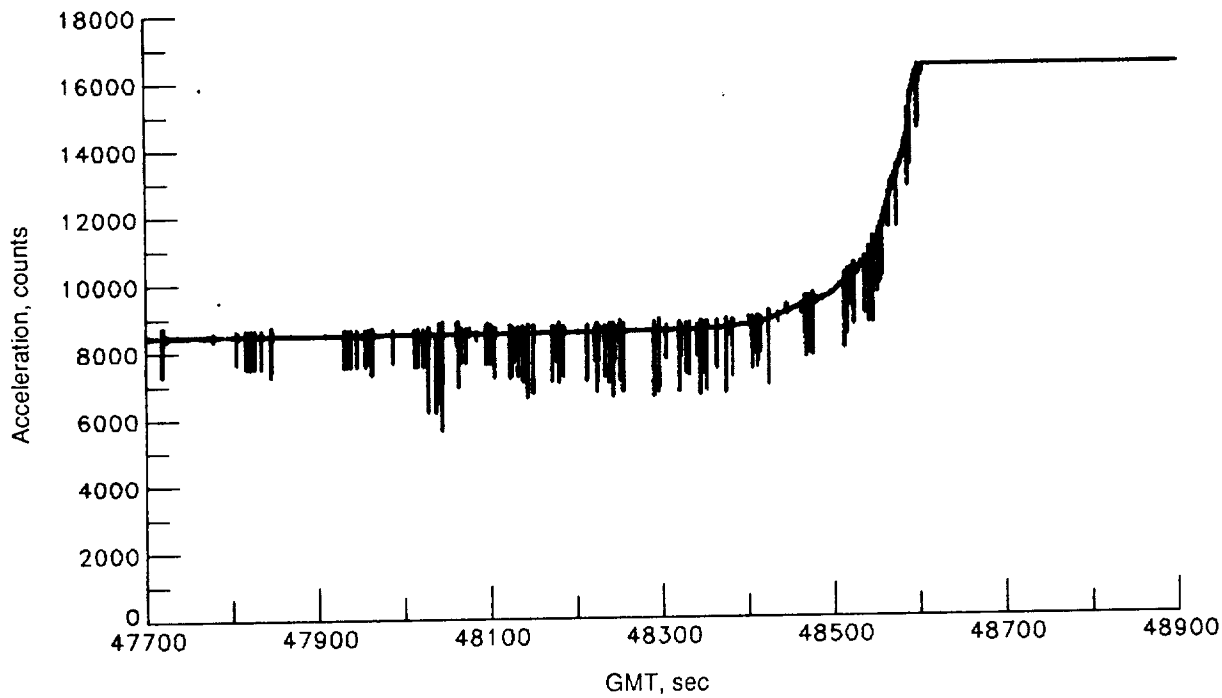


(c) Y-axis acceleration counts.

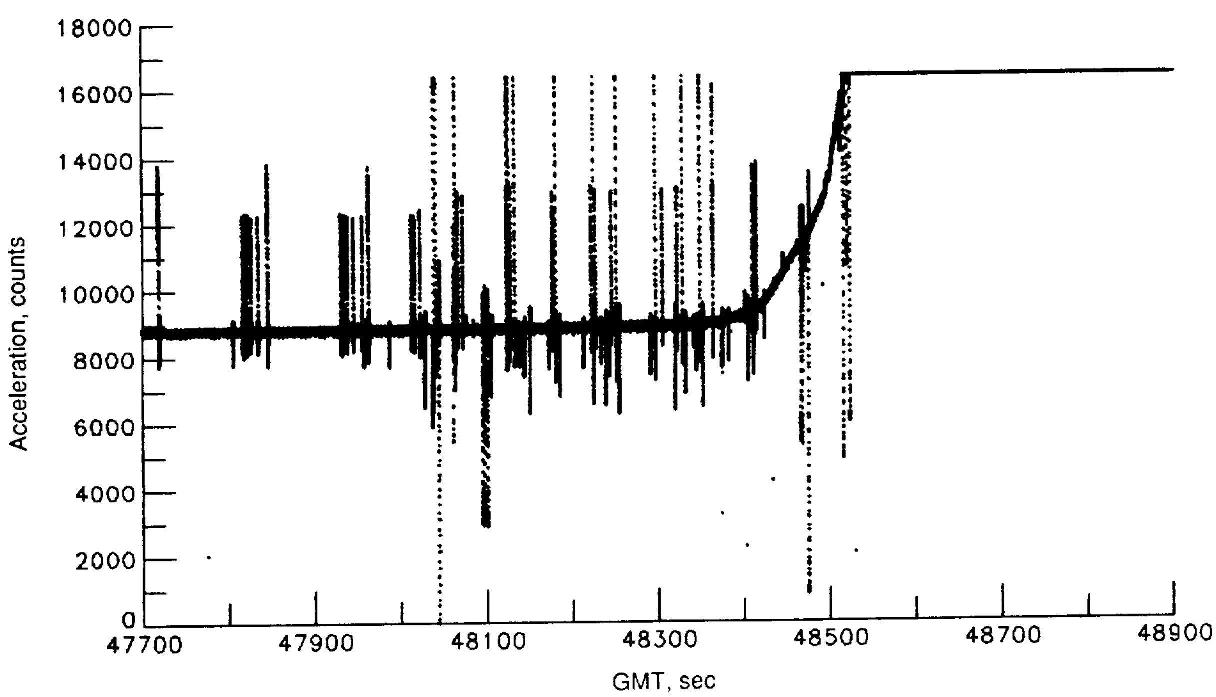


(d) X-, Y-, and Z-axis temperatures.

Figure 3. Concluded.

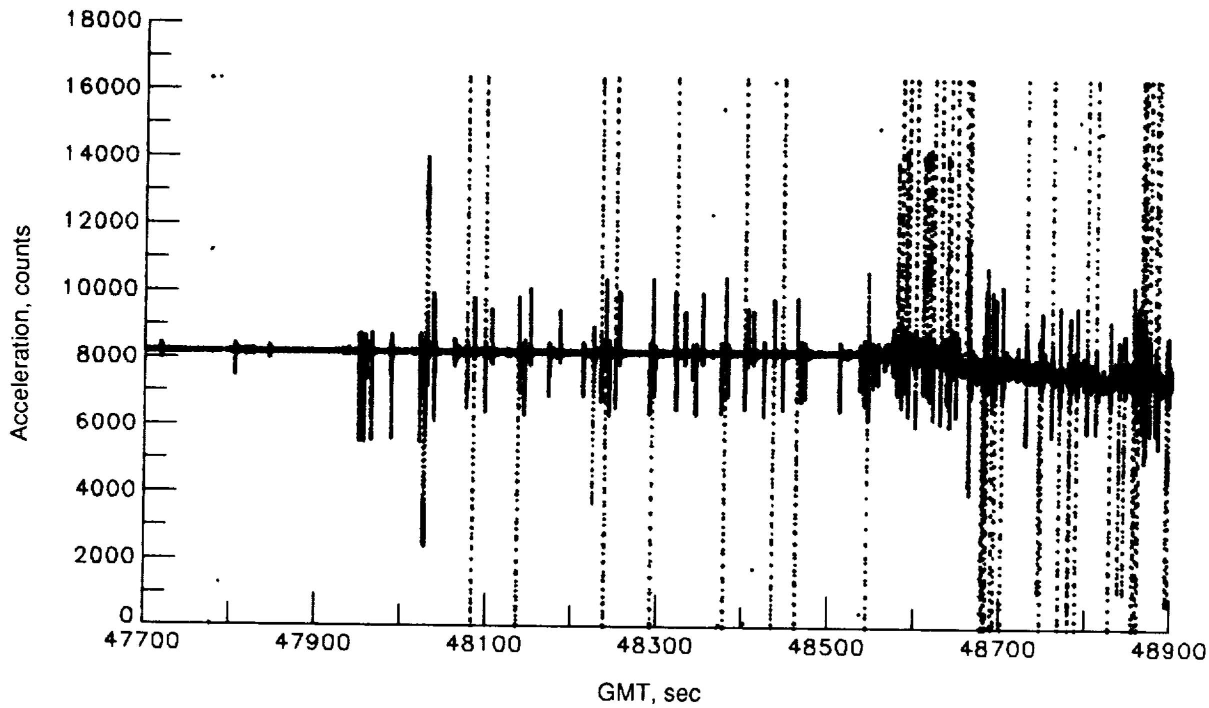


(a) X-axis acceleration counts.

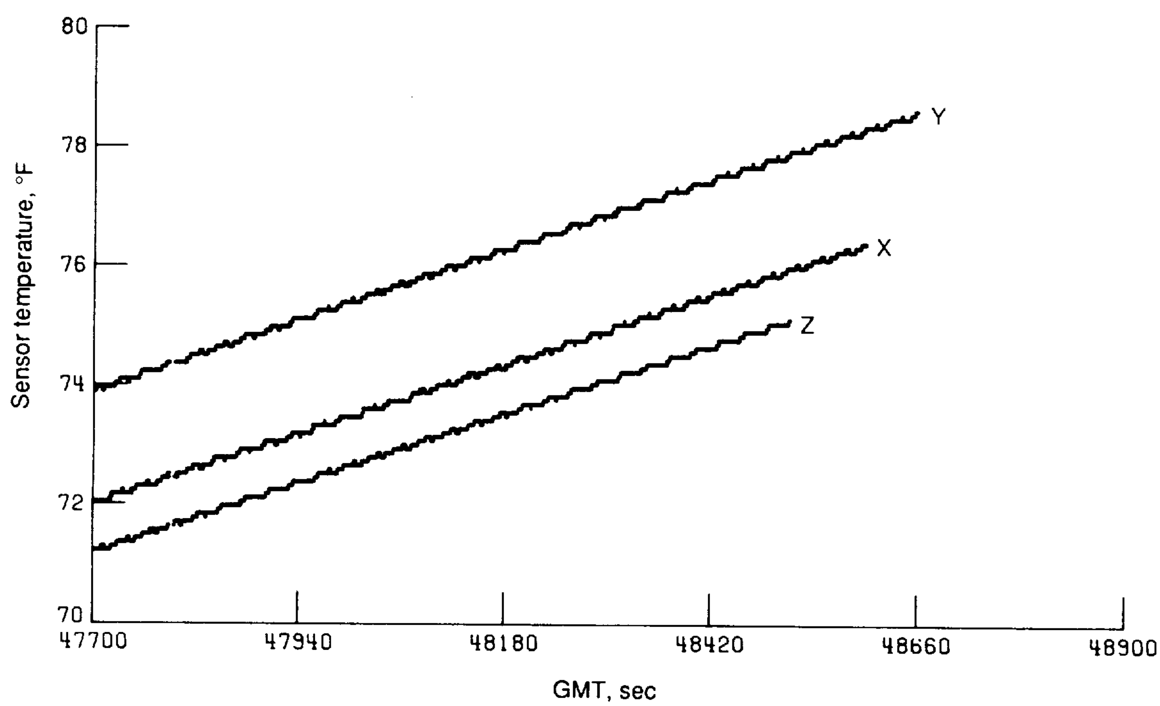


(b) Z-axis acceleration counts.

Figure 4. Acceleration counts and sensor temperature versus time for STS-07.

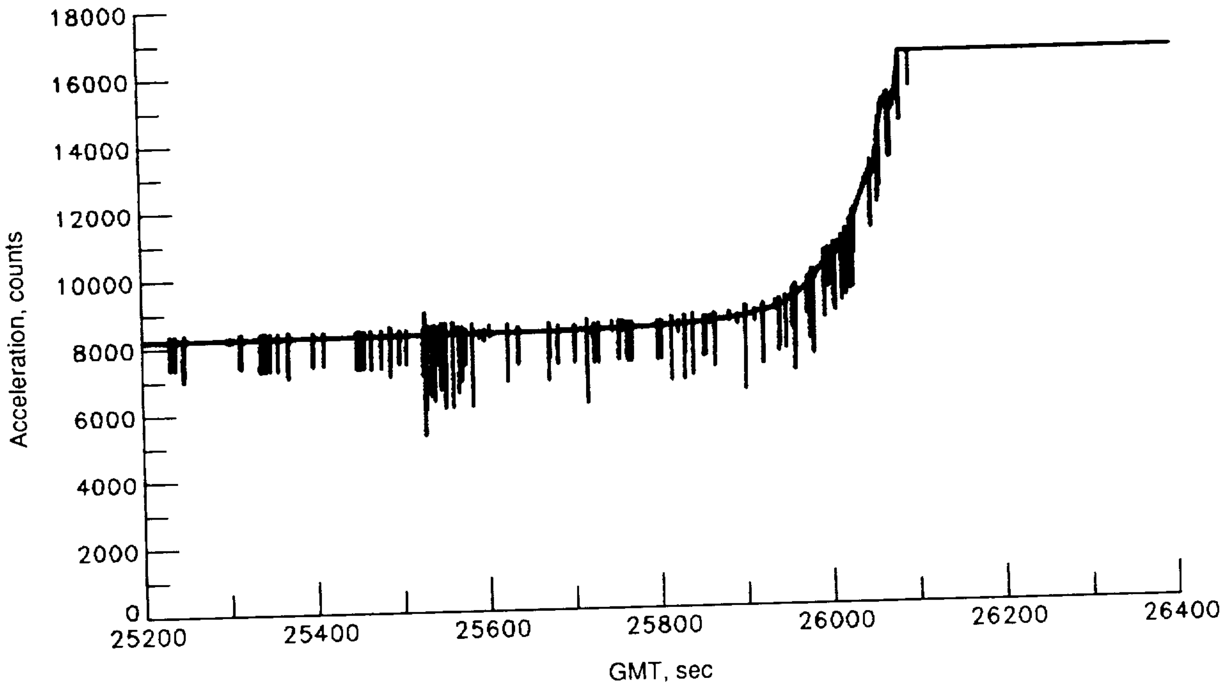


(c) Y-axis acceleration counts.

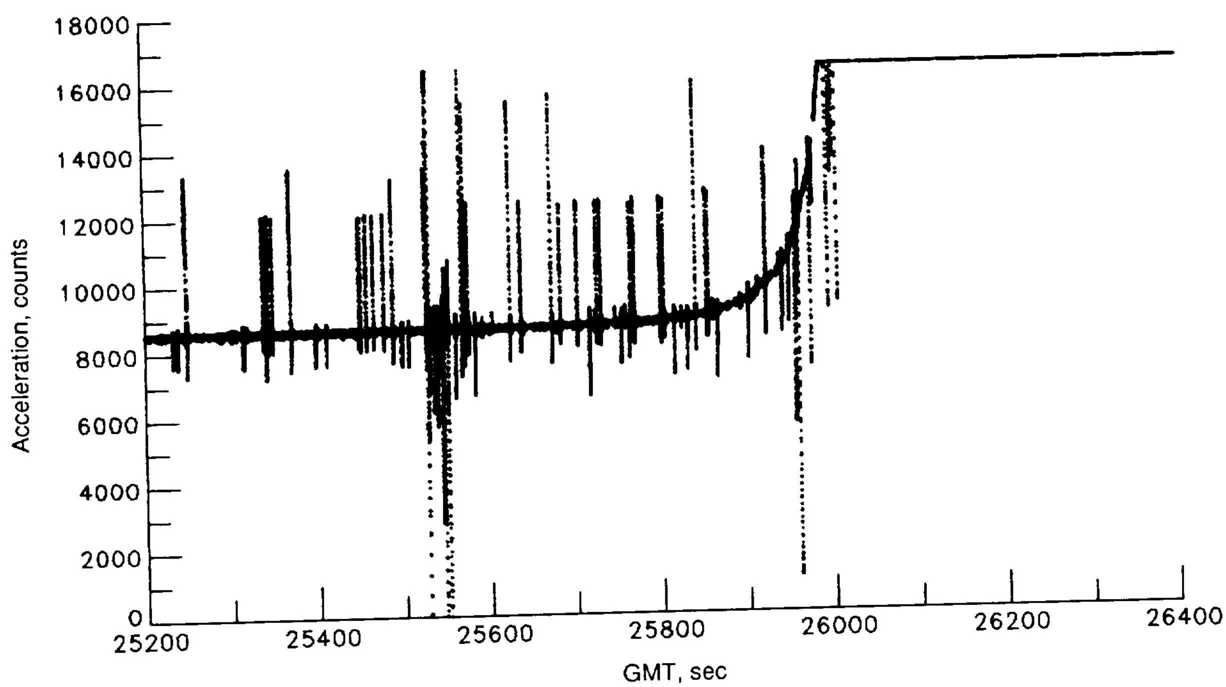


(d) X-, Y-, and Z-axis temperatures.

Figure 4. Concluded.

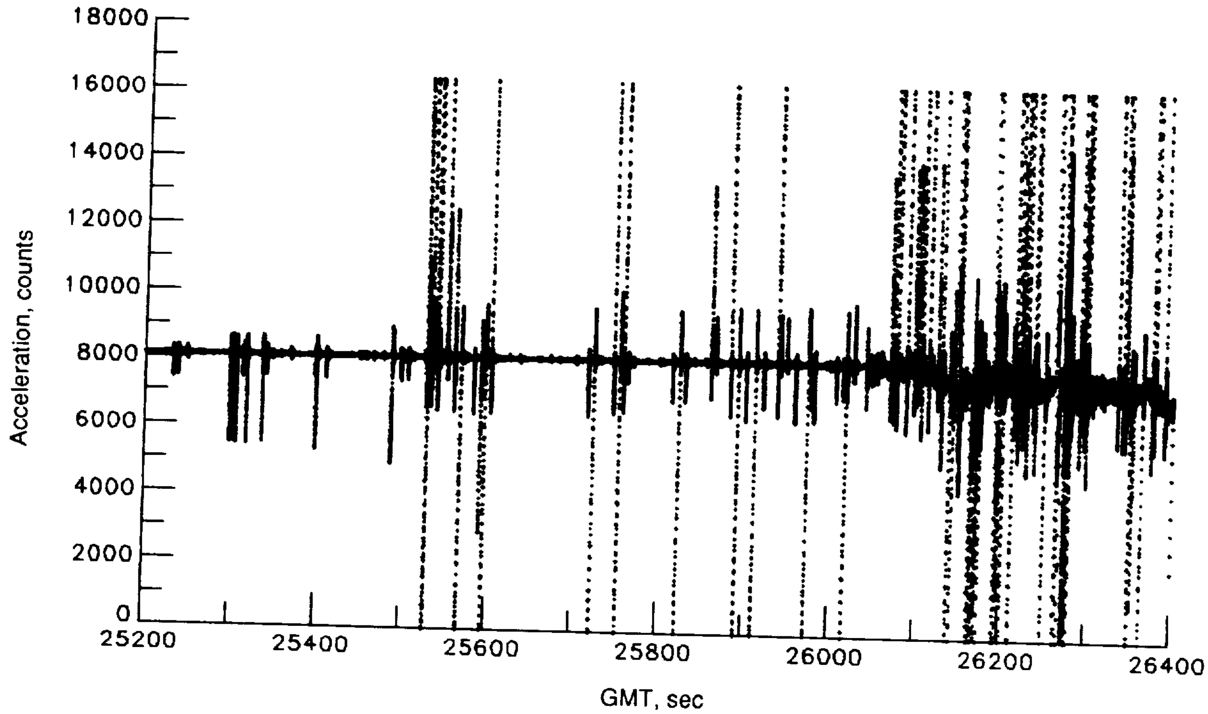


(a) X-axis acceleration counts.

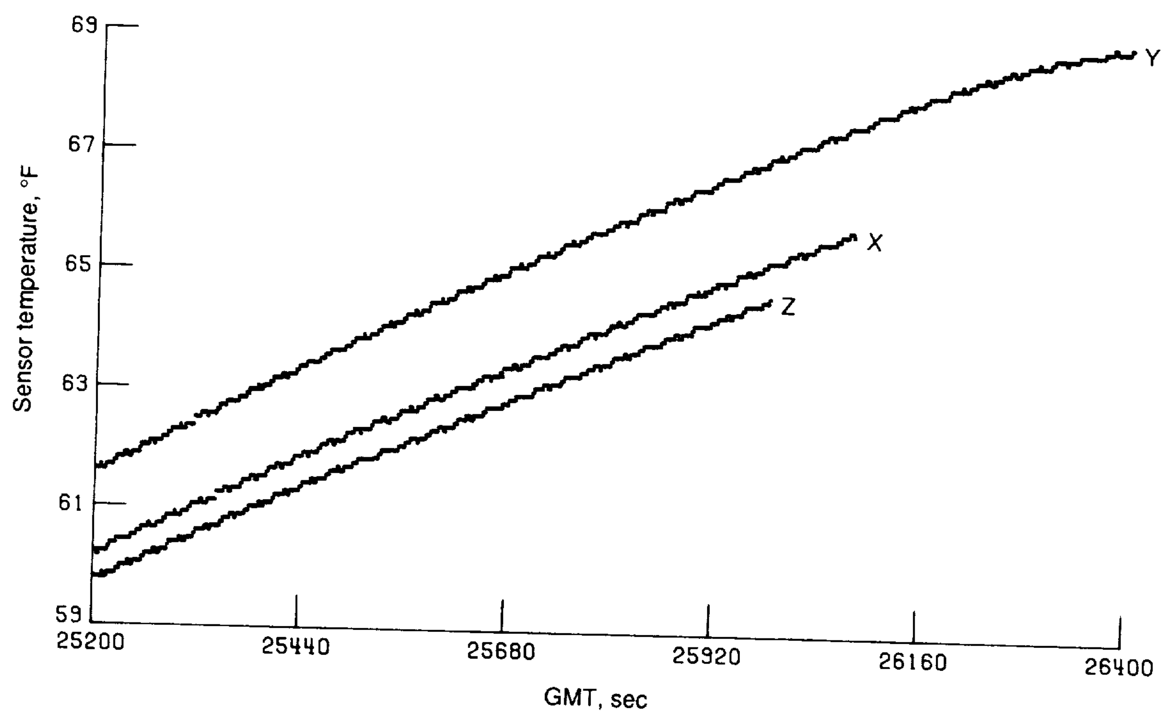


(b) Z-axis acceleration counts.

Figure 5. Acceleration counts and sensor temperature versus time for STS-08.

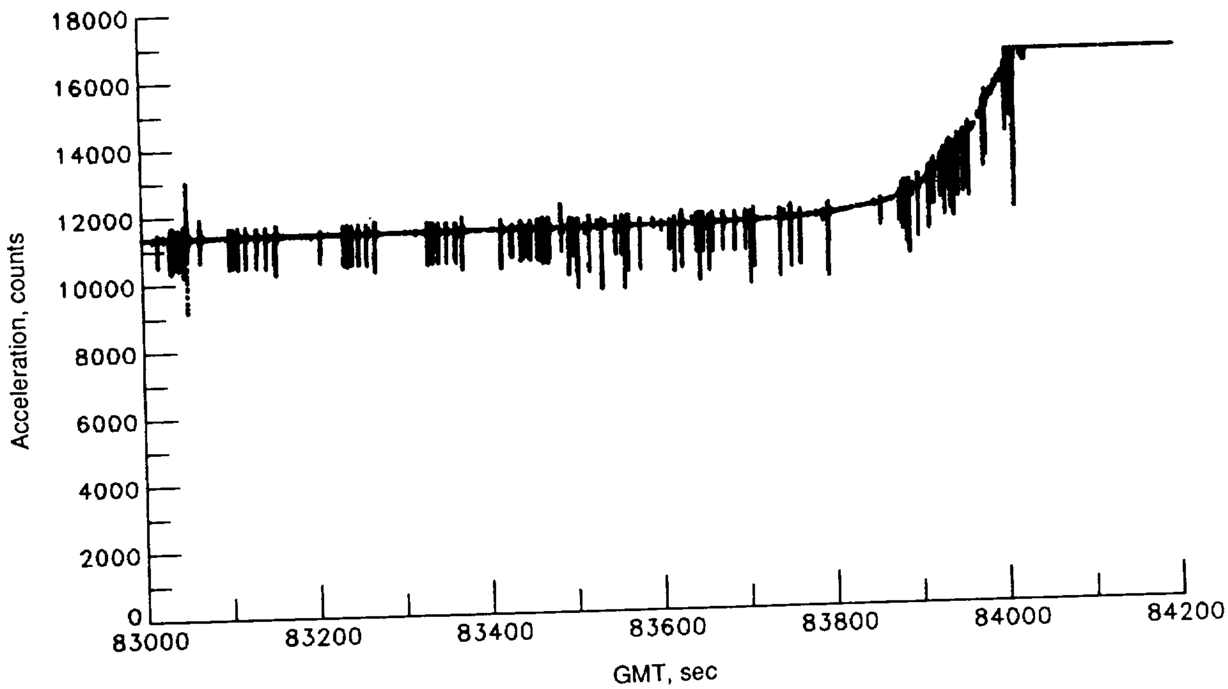


(c) Y-axis acceleration counts.

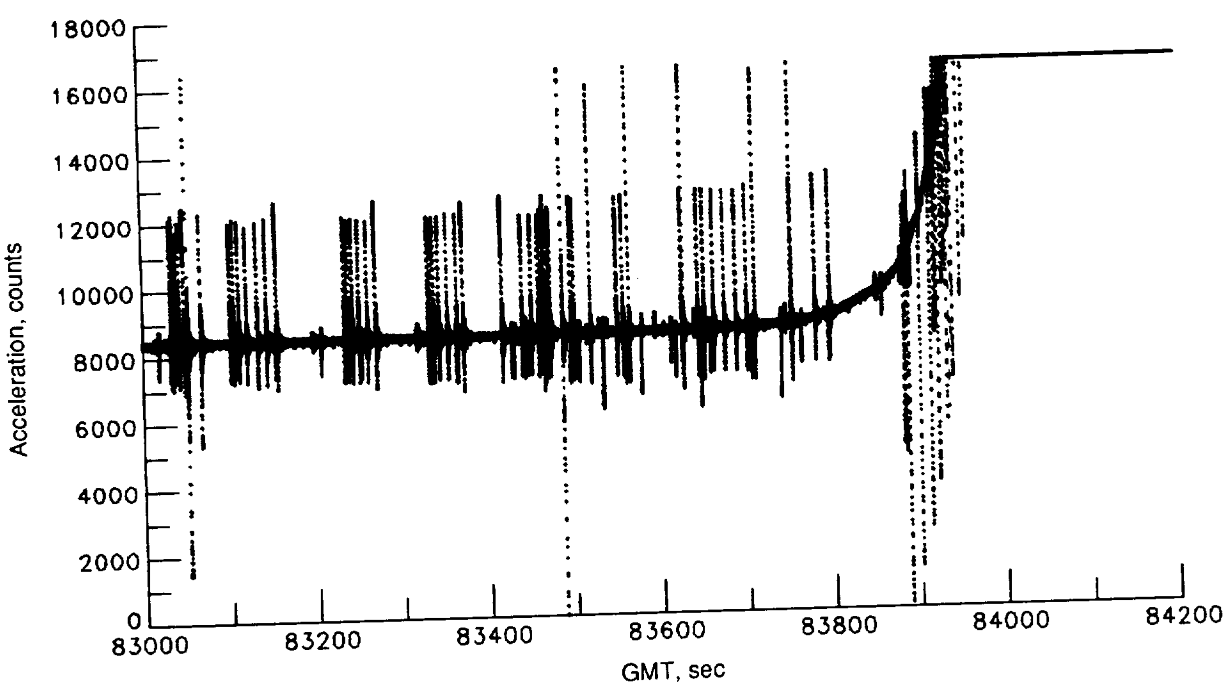


(d) X-, Y-, and Z-axis temperatures.

Figure 5. Concluded.

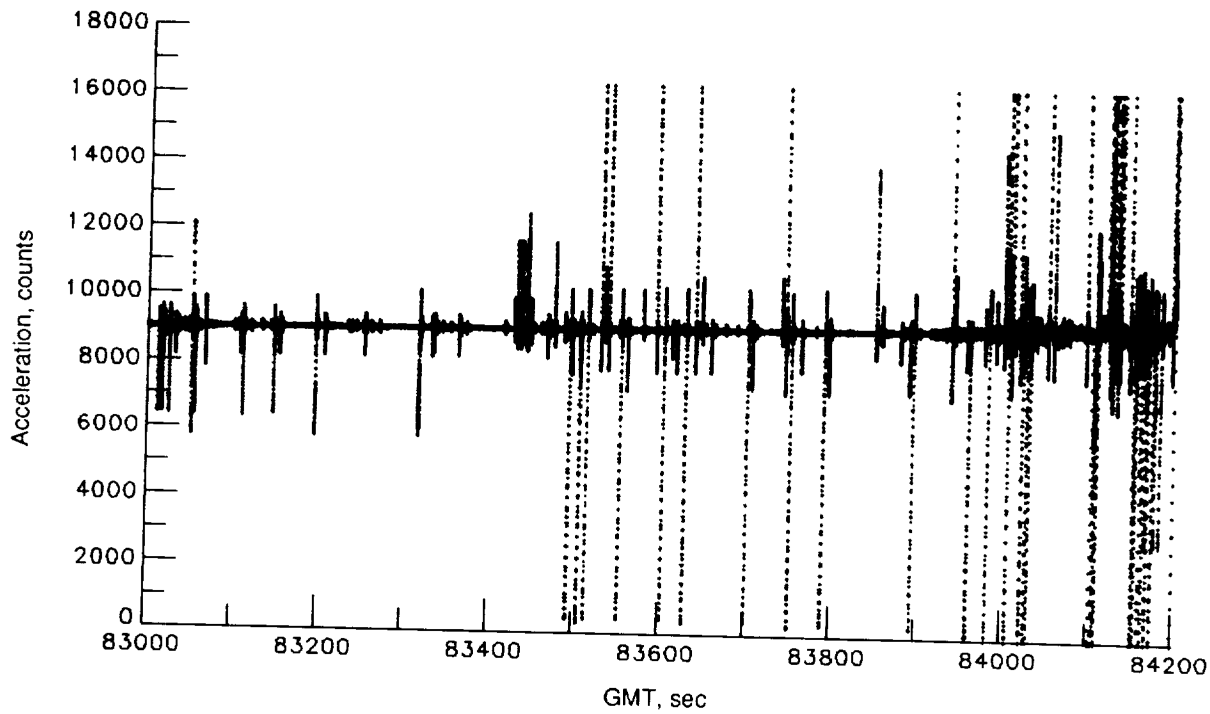


(a) X-axis acceleration counts.

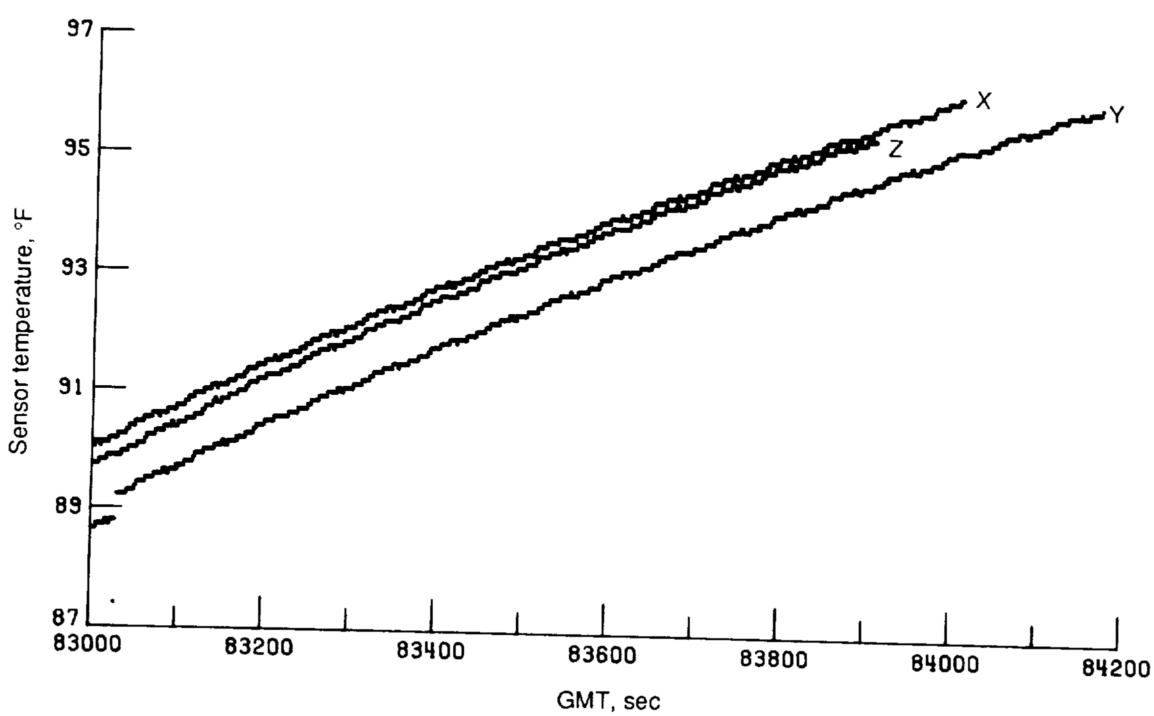


(b) Z-axis acceleration counts.

Figure 6. Acceleration counts, sensor temperature, and sensor temperature counts versus time for STS-09.

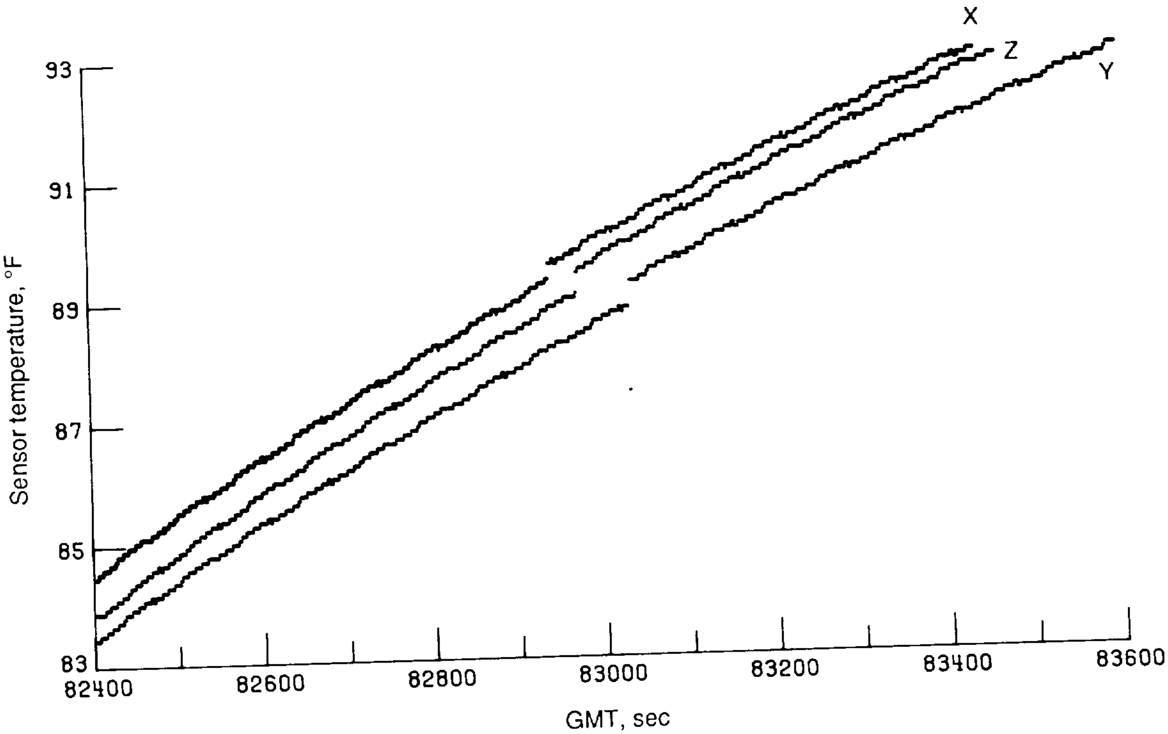


(c) Y-axis acceleration counts.

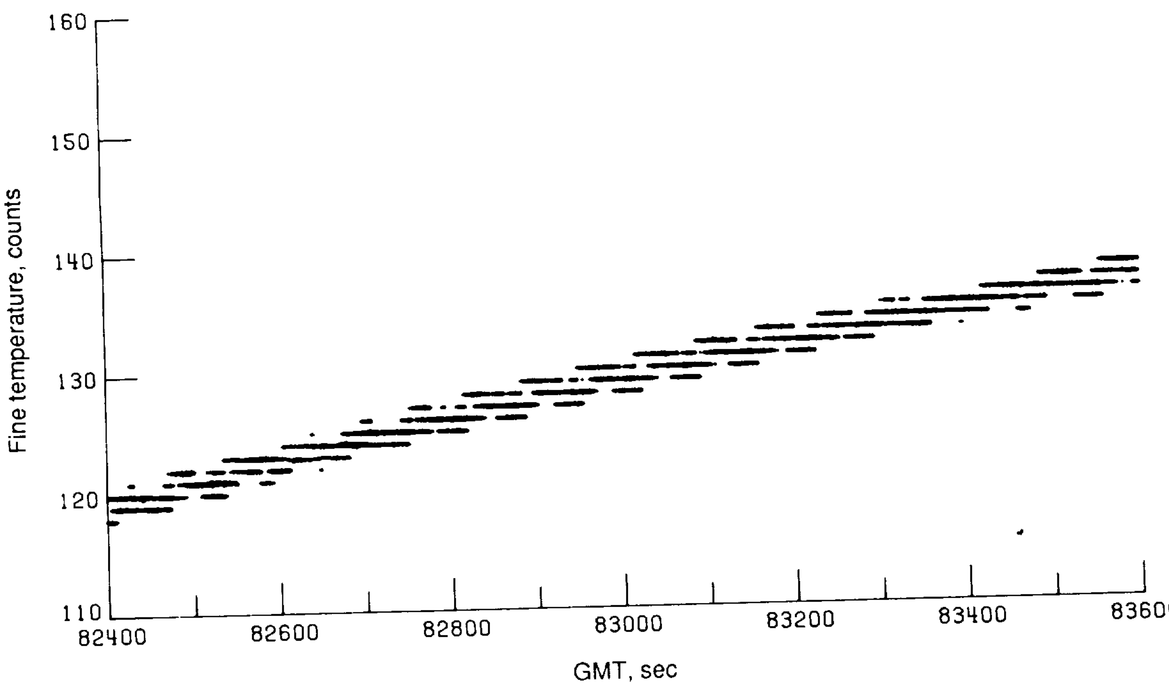


(d) X-, Y-, and Z-axis temperatures.

Figure 6. Continued.

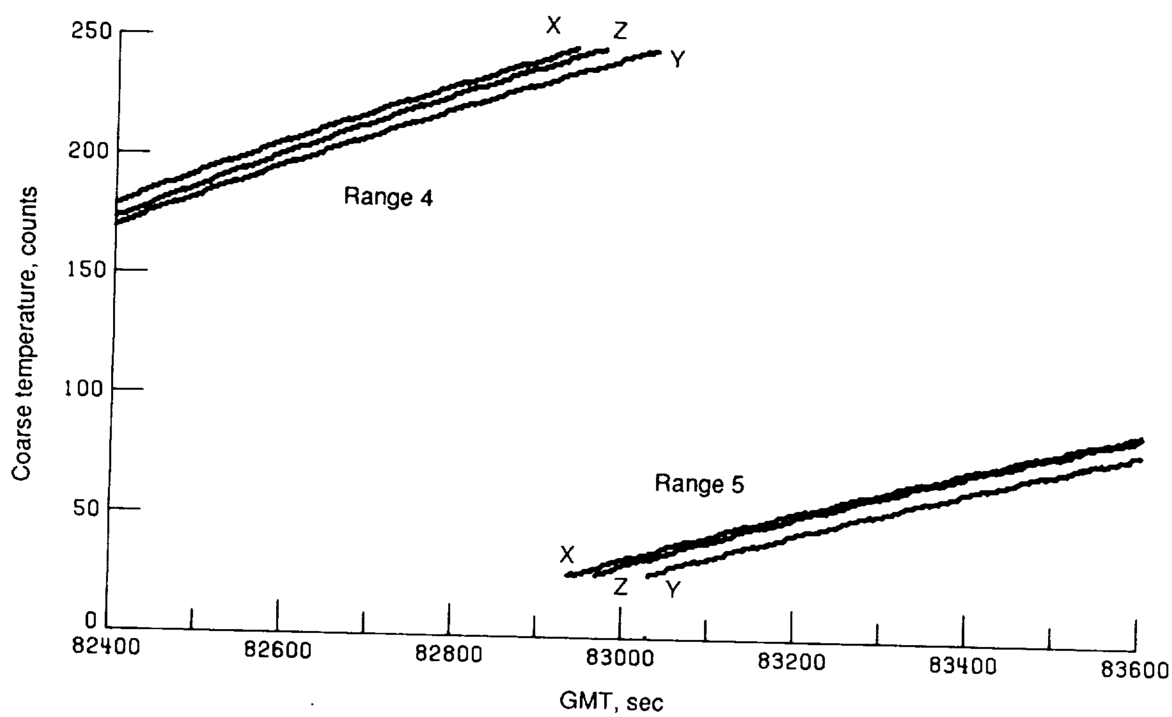


(e) X-, Y-, and Z-axis temperature.



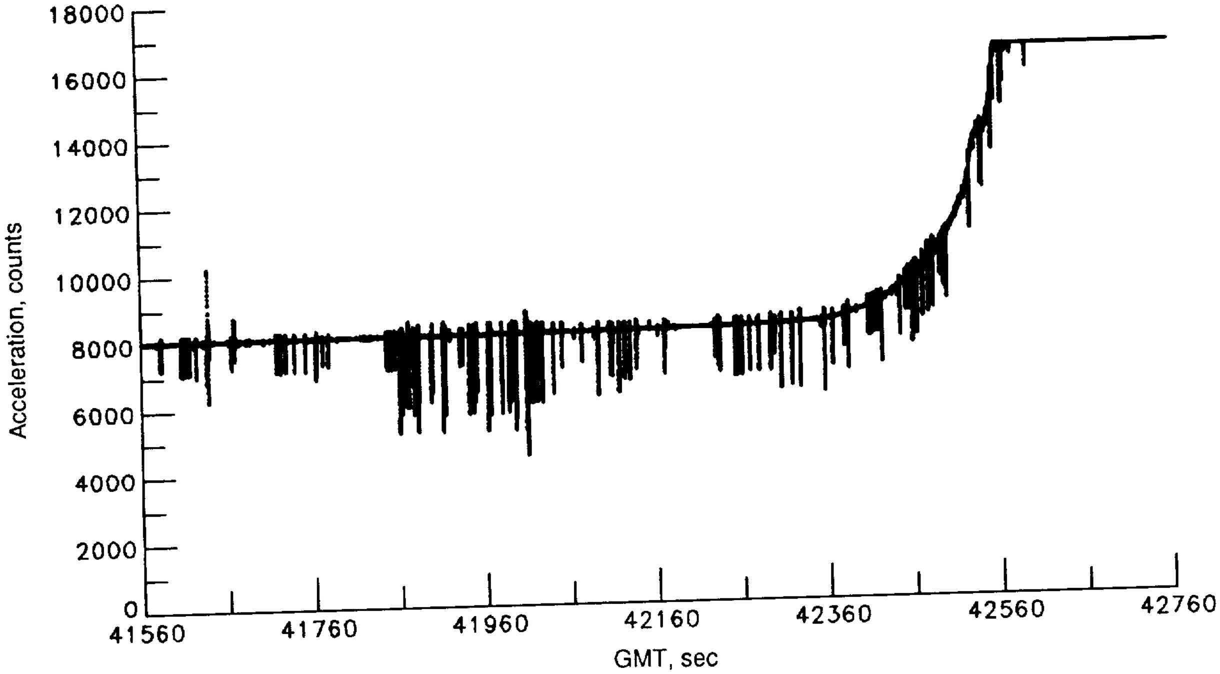
(f) X-, Y-, and Z-axis fine temperature counts.

Figure 6. Continued.

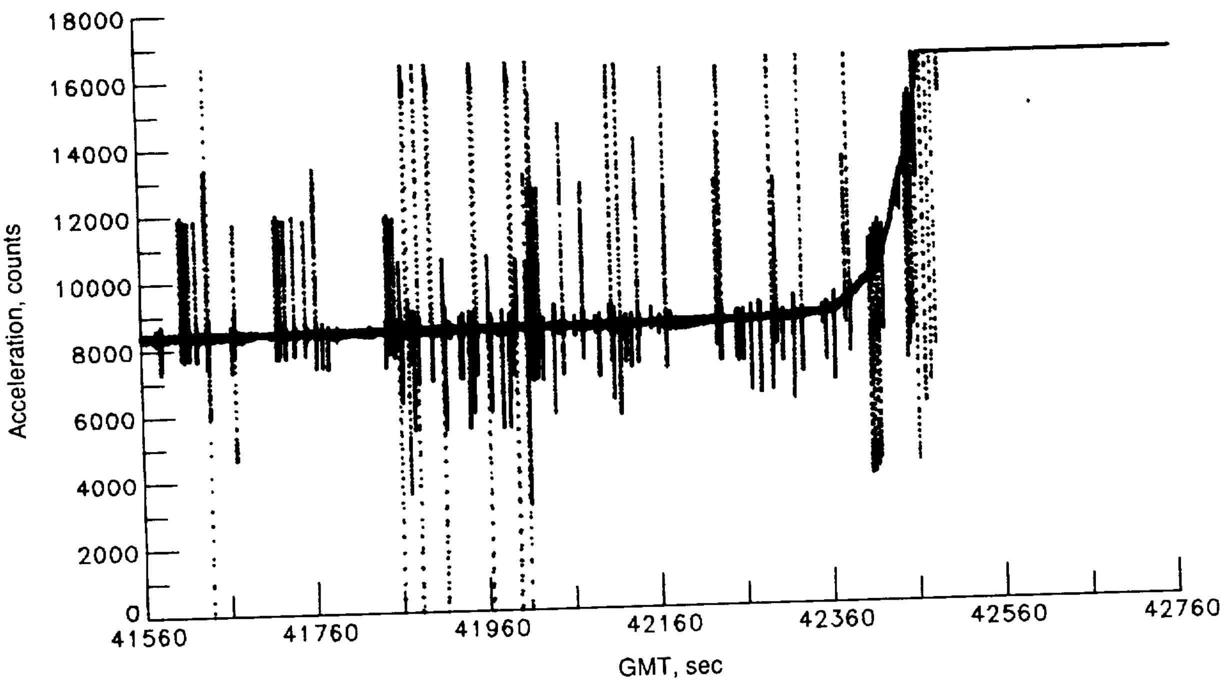


(g) X-, Y-, and Z-axis coarse temperature counts.

Figure 6. Concluded.

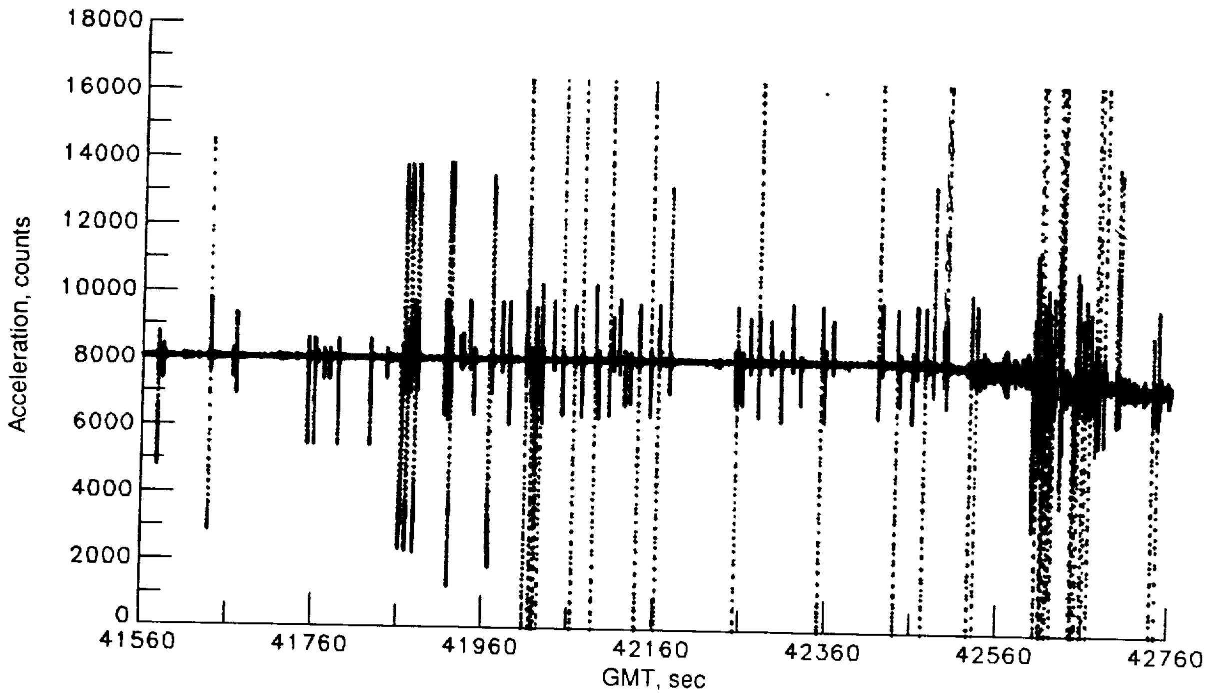


(a) X-axis acceleration counts.

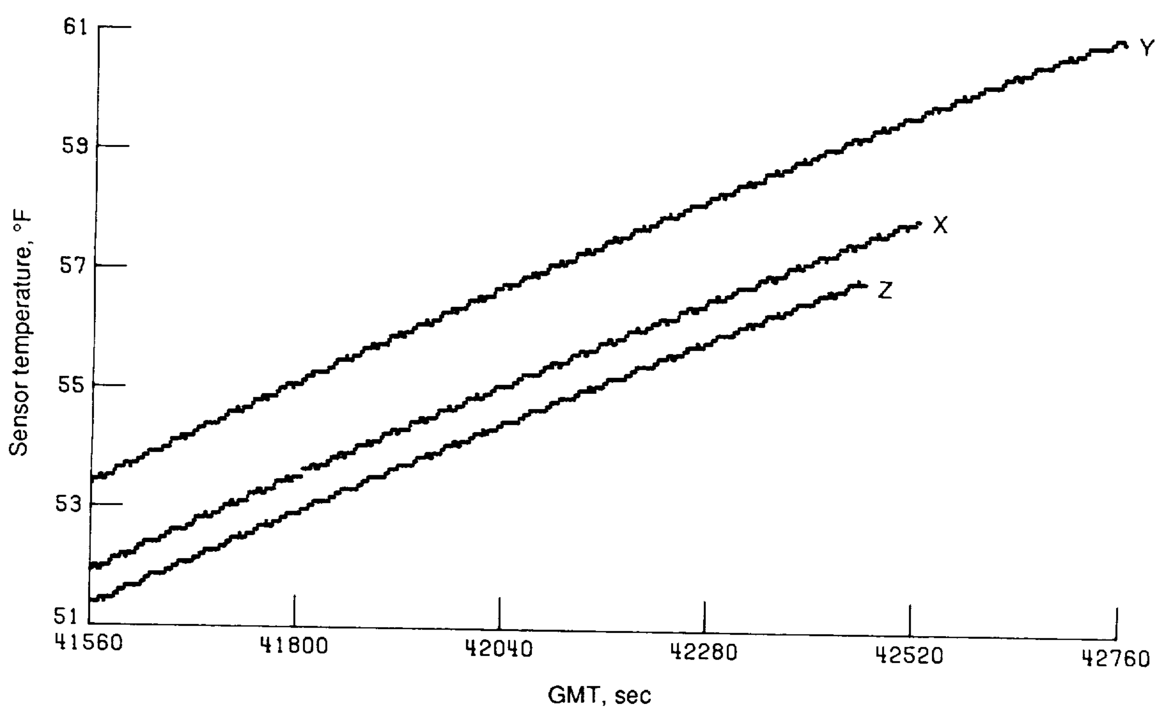


(b) Z-axis acceleration counts.

Figure 7. Acceleration counts and sensor temperature versus time for STS-41B.

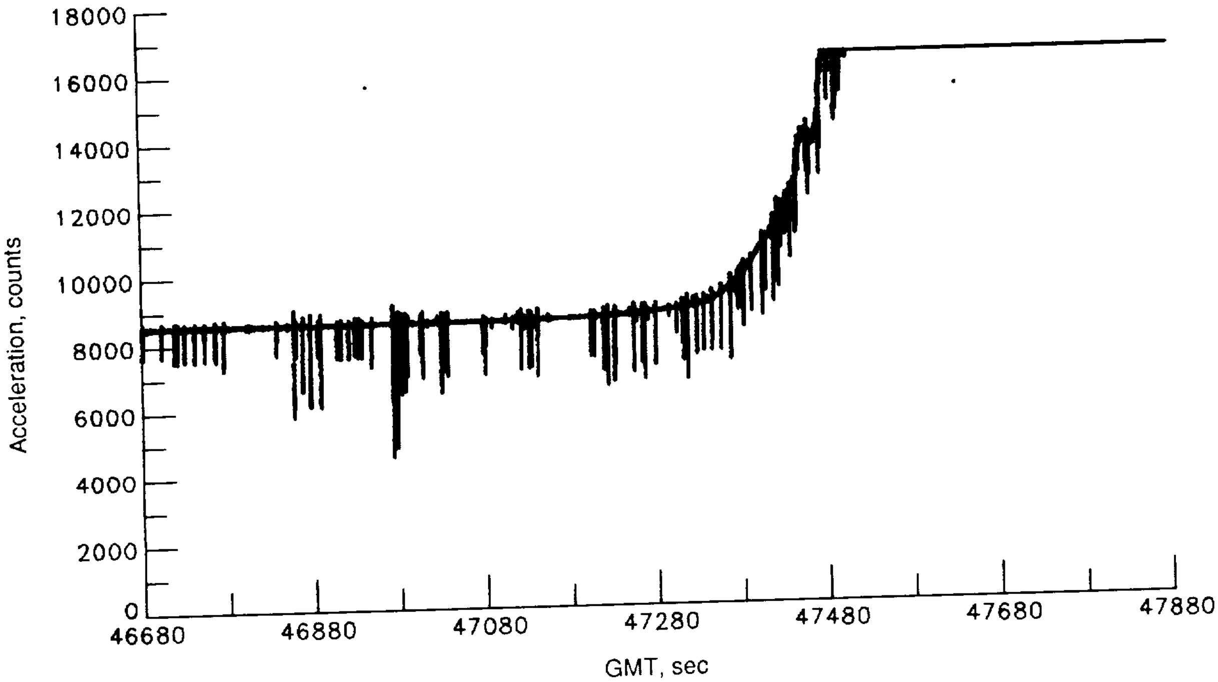


(c) Y-axis acceleration counts.

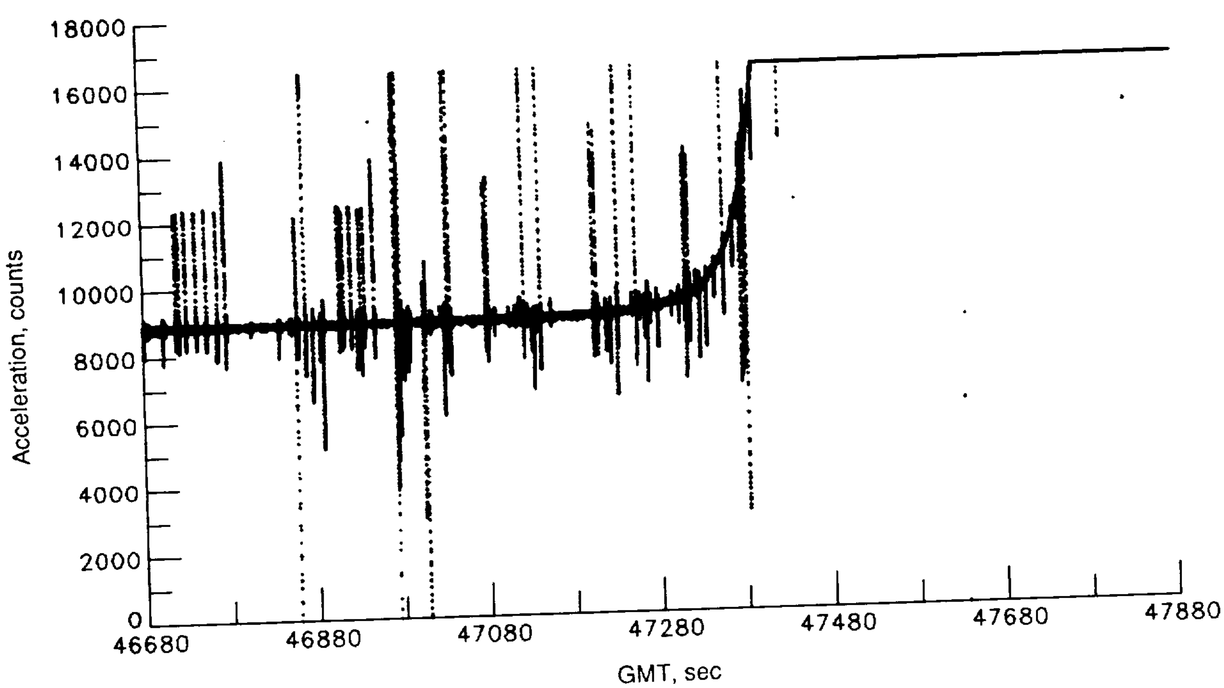


(d) X-, Y-, and Z-axis temperatures.

Figure 7. Concluded.

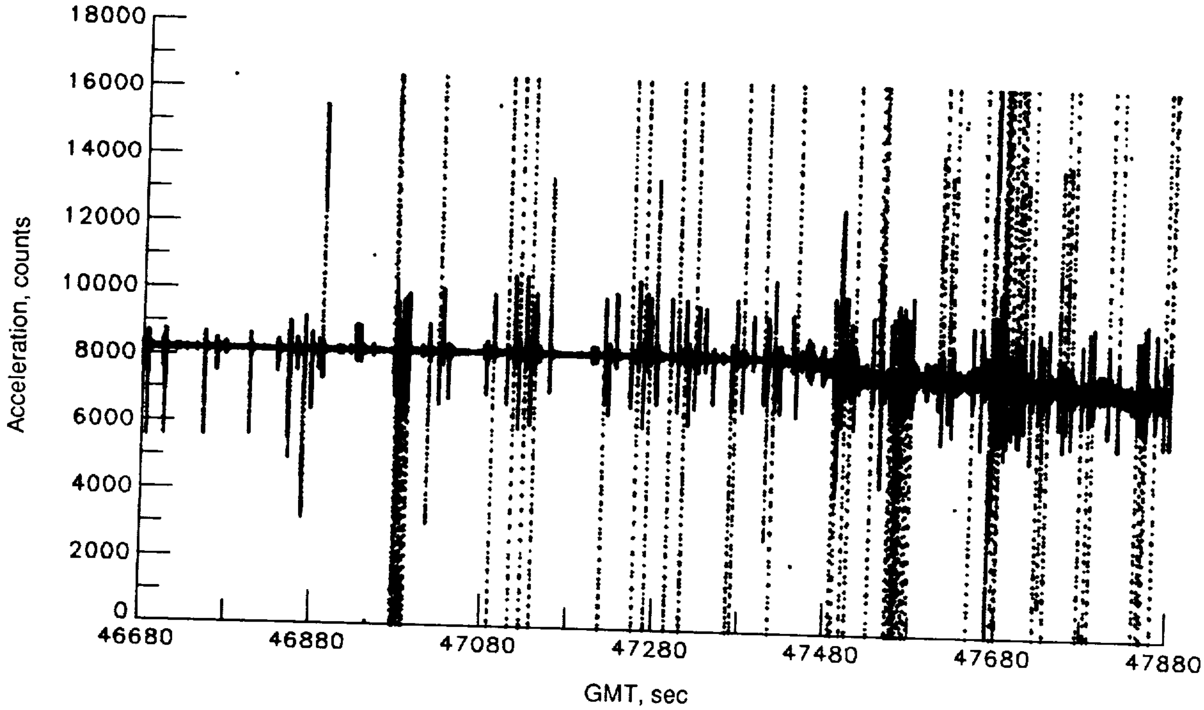


(a) X-axis acceleration counts.

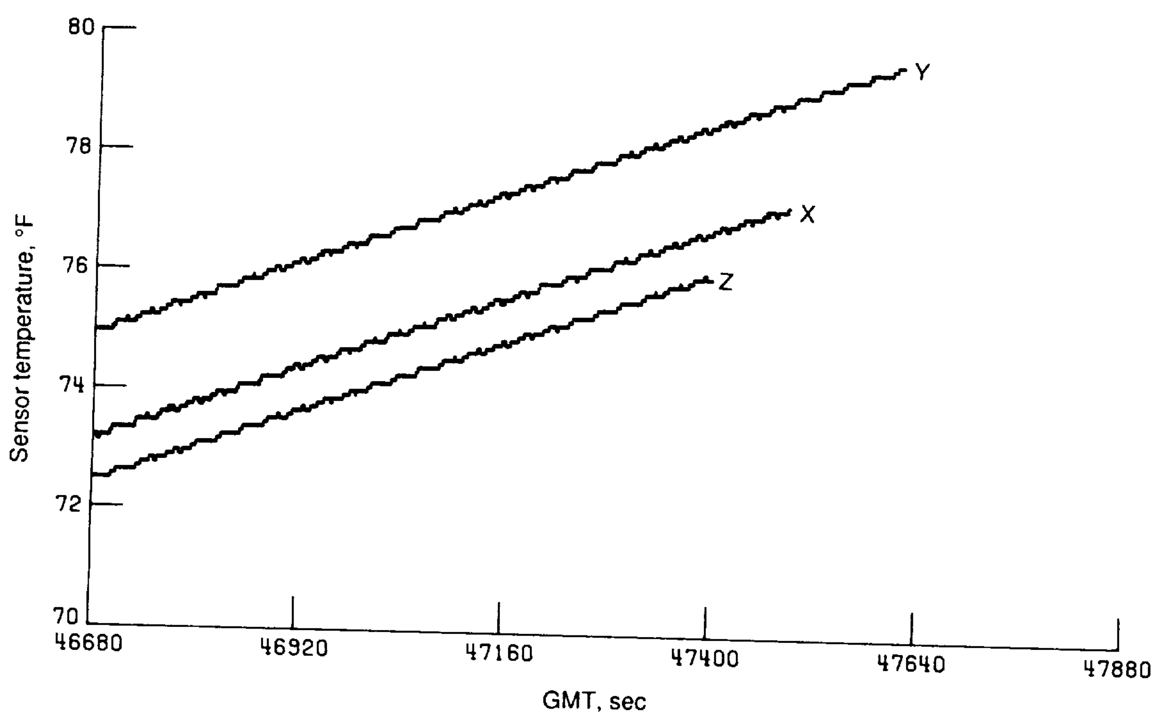


(b) Z-axis acceleration counts.

Figure 8. Acceleration counts and sensor temperature versus time for STS-41C.

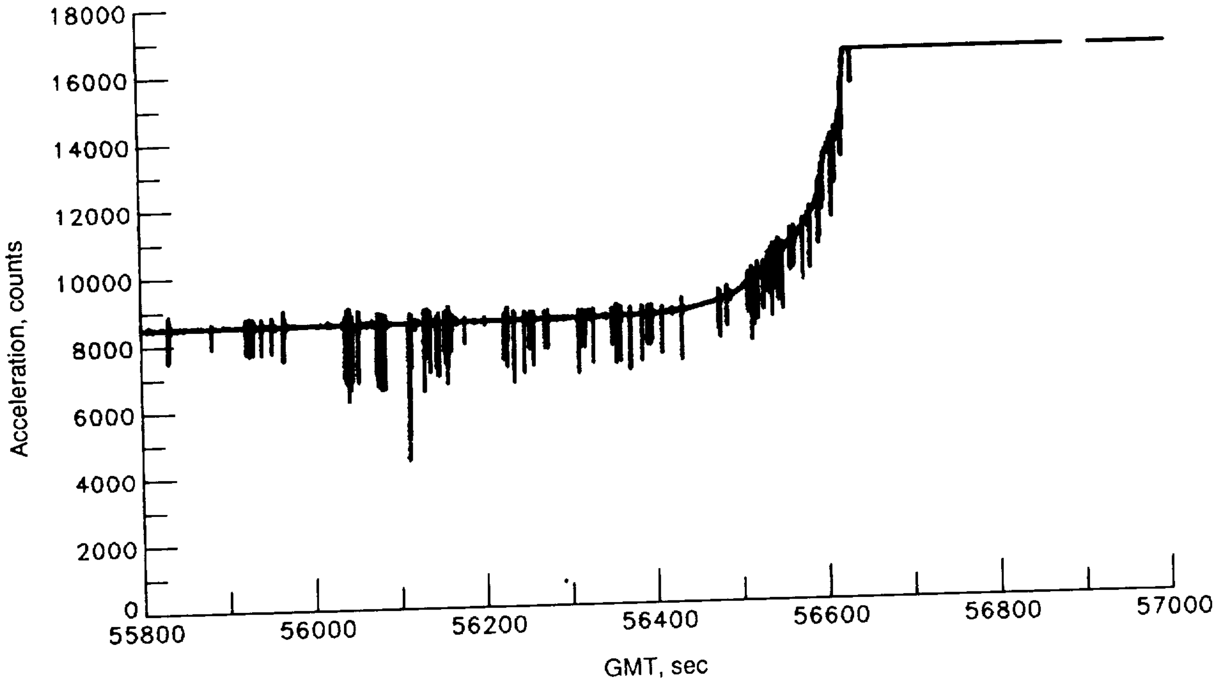


(c) Y-axis acceleration counts.

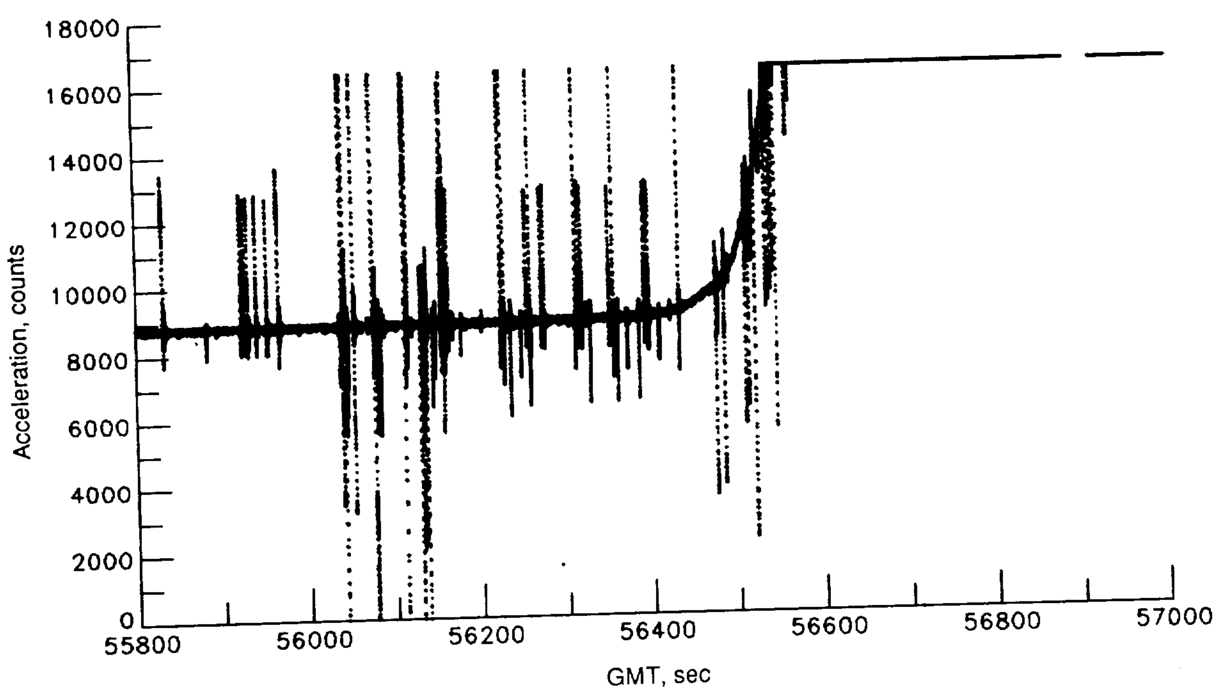


(d) X-, Y-, and Z-axis temperatures.

Figure 8. Concluded.

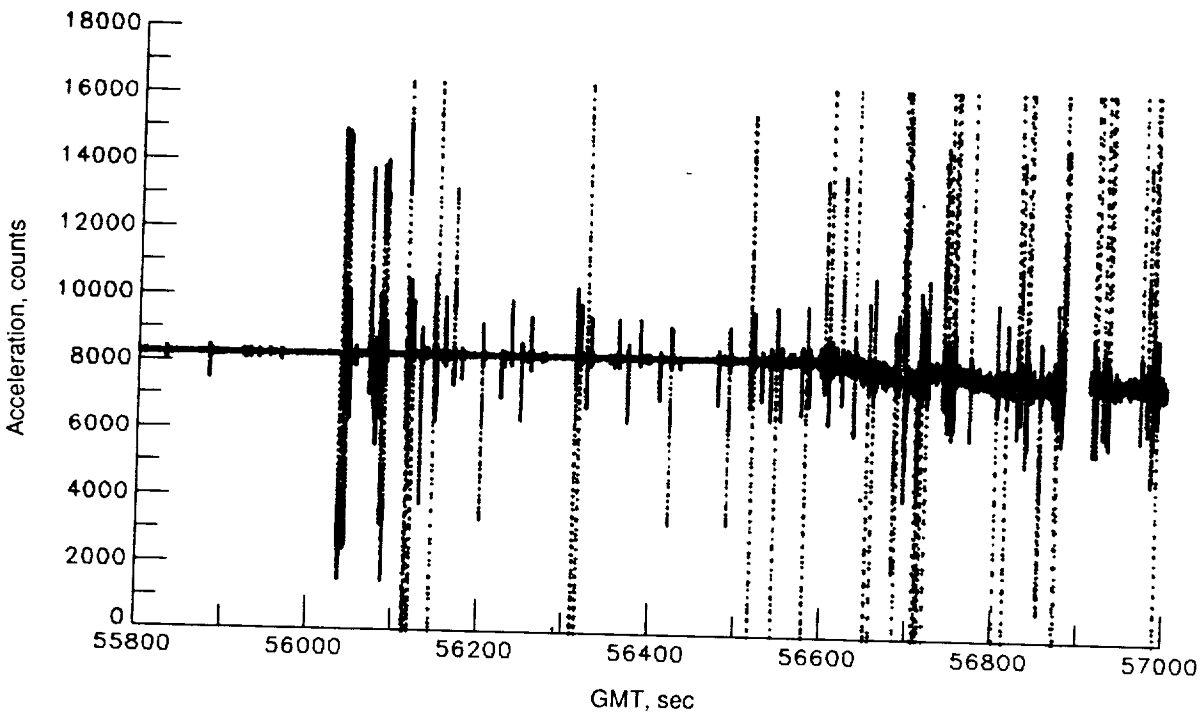


(a) X-axis acceleration counts.

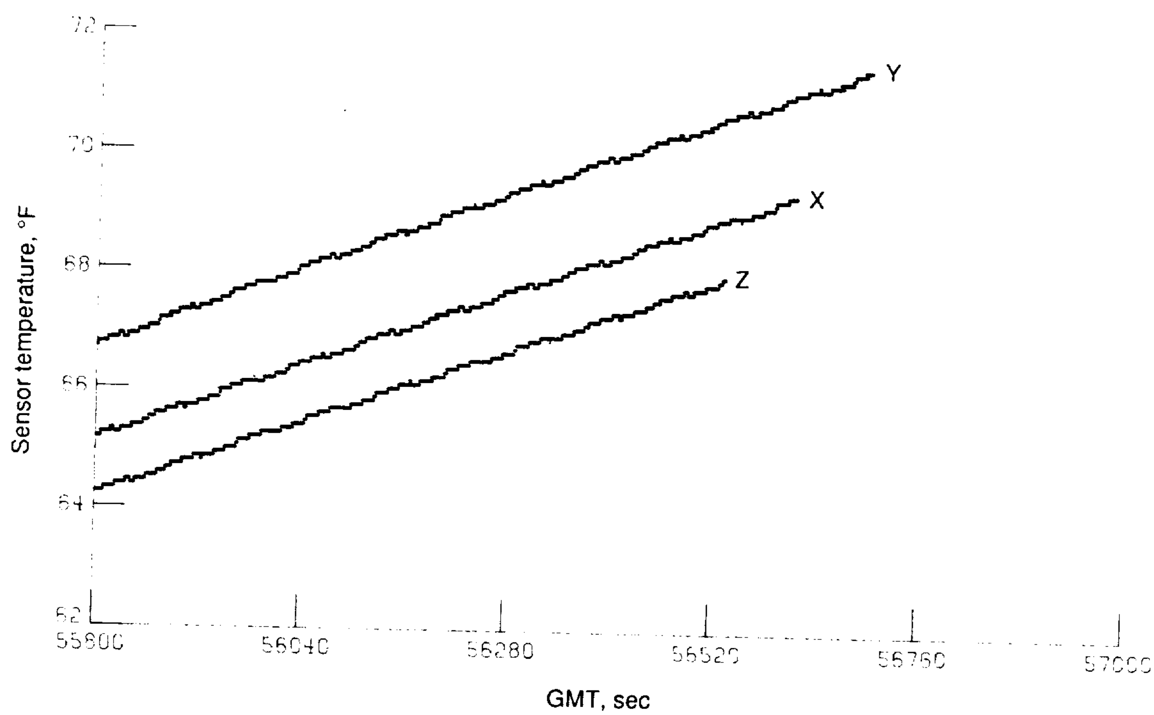


(b) Z-axis acceleration counts.

Figure 9. Acceleration counts and sensor temperature versus time for STS-51B.

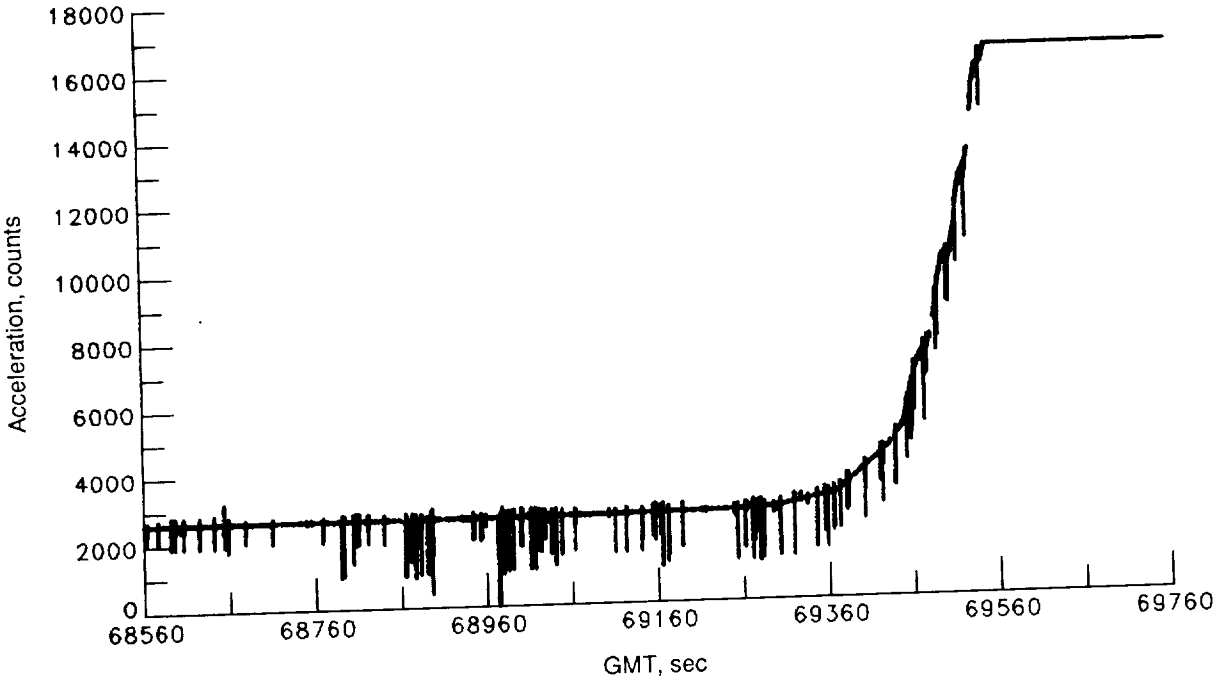


(c) Y-axis acceleration counts.

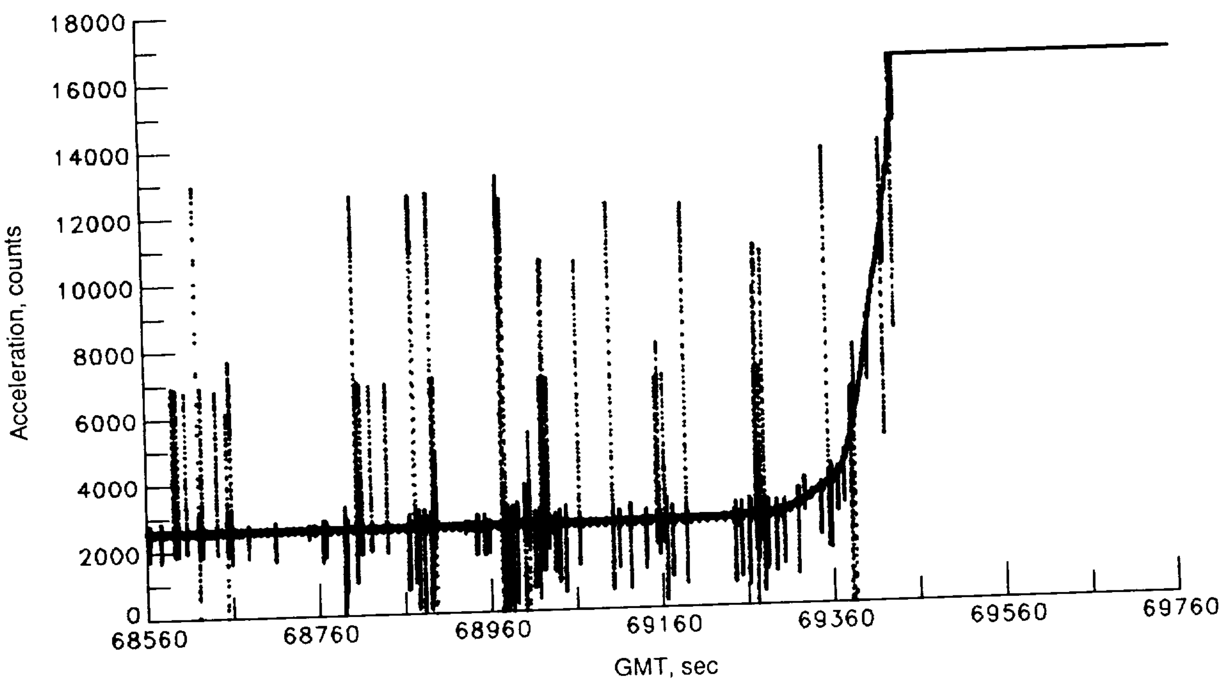


(d) X-, Y-, and Z-axis temperatures.

Figure 9. Concluded.

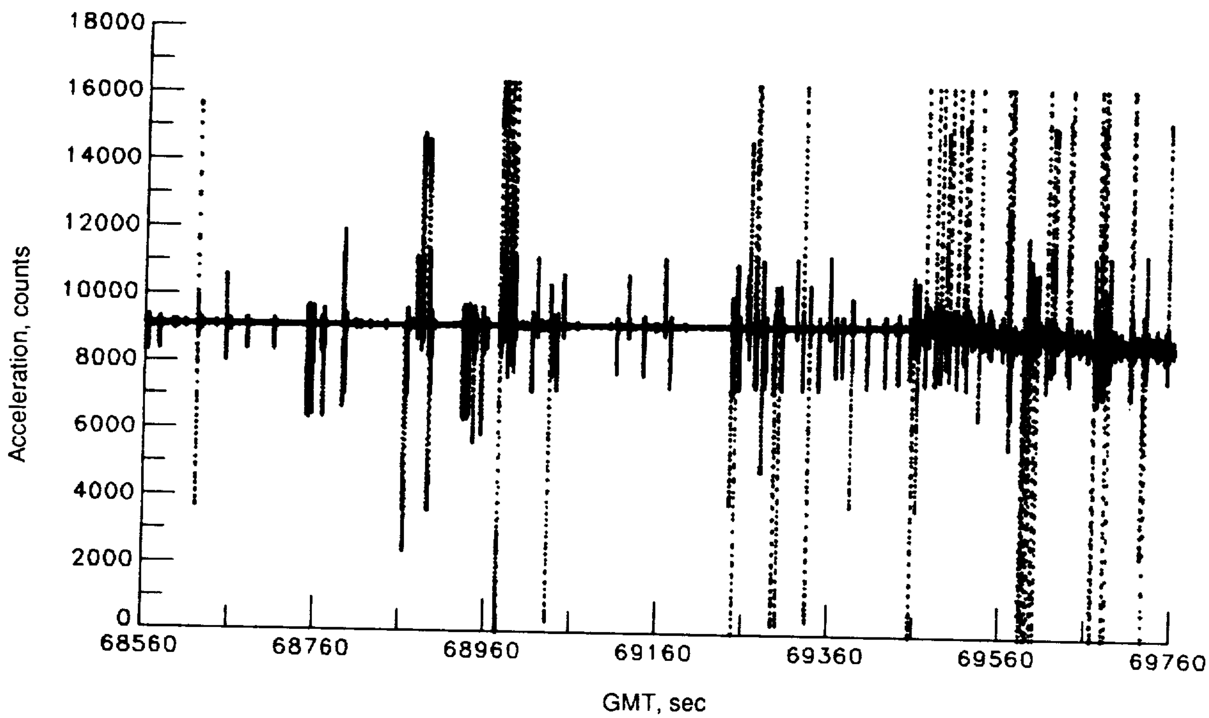


(a) X-axis acceleration counts.

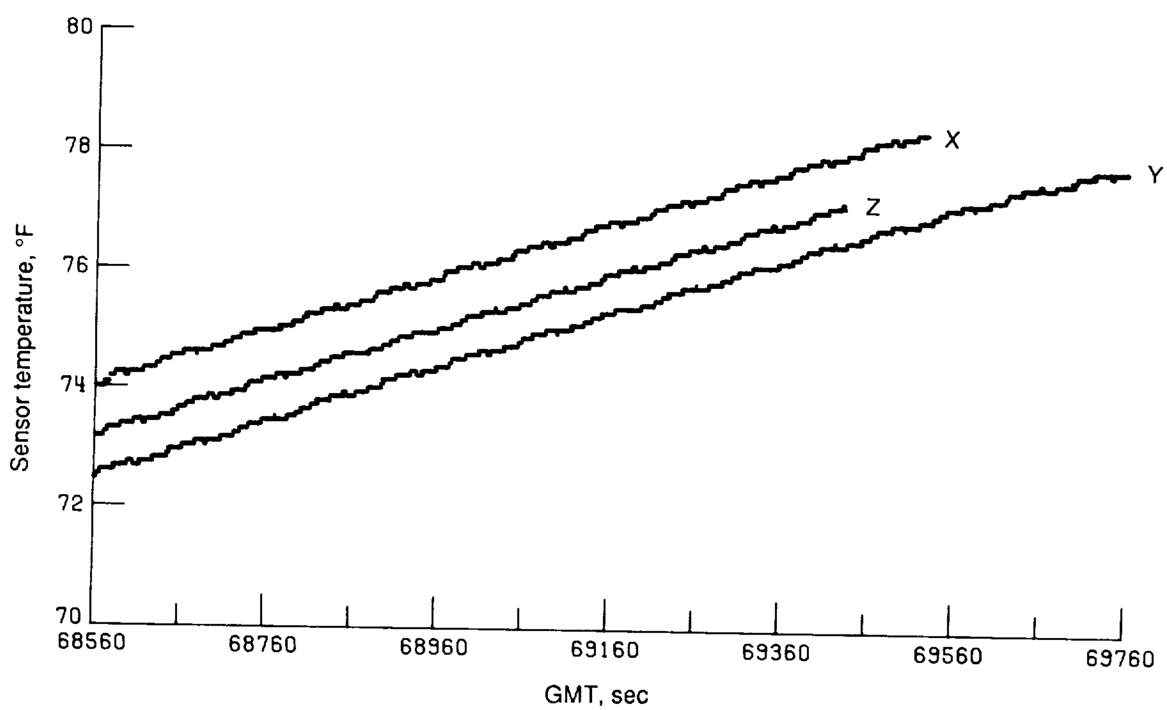


(b) Z-axis acceleration counts.

Figure 10. Acceleration counts and sensor temperature versus time for STS-51F.

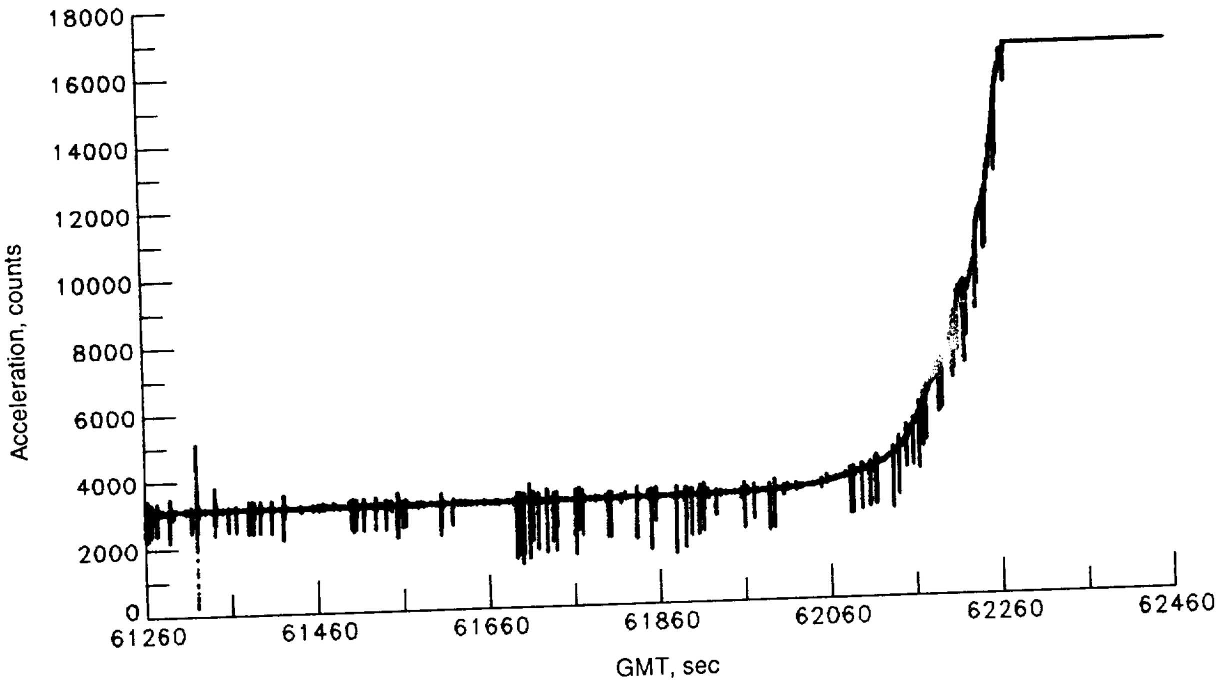


(c) Y-axis acceleration counts.

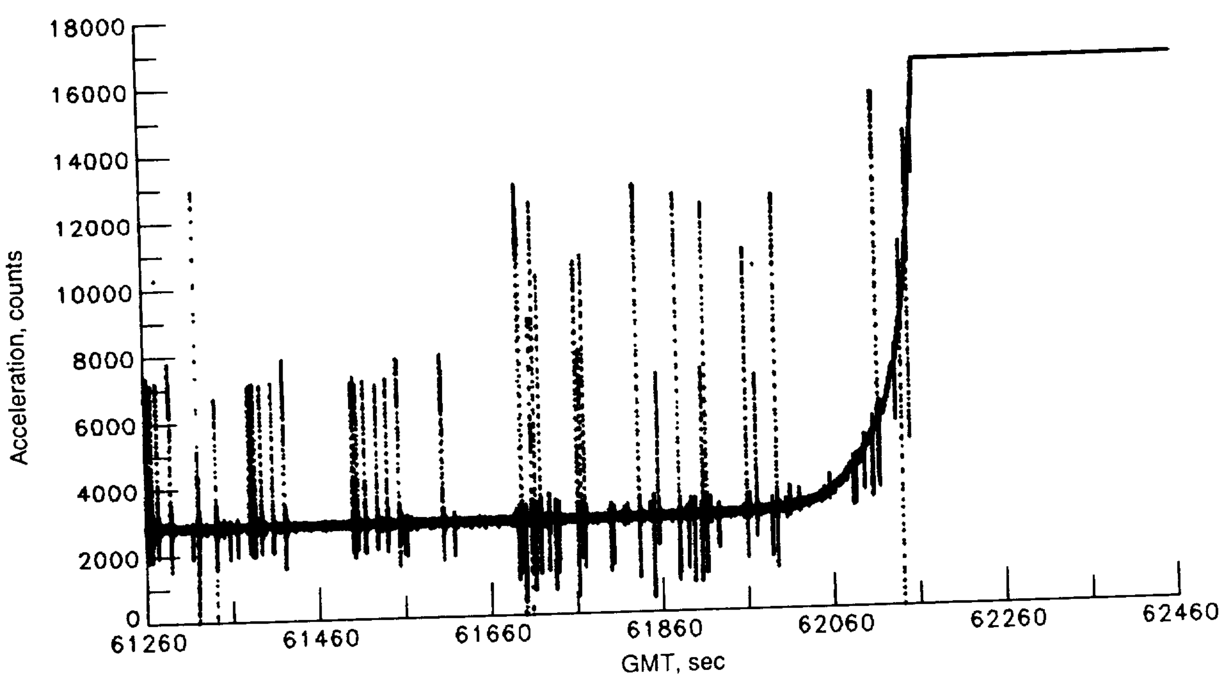


(d) X-, Y-, and Z-axis temperatures.

Figure 10. Concluded.

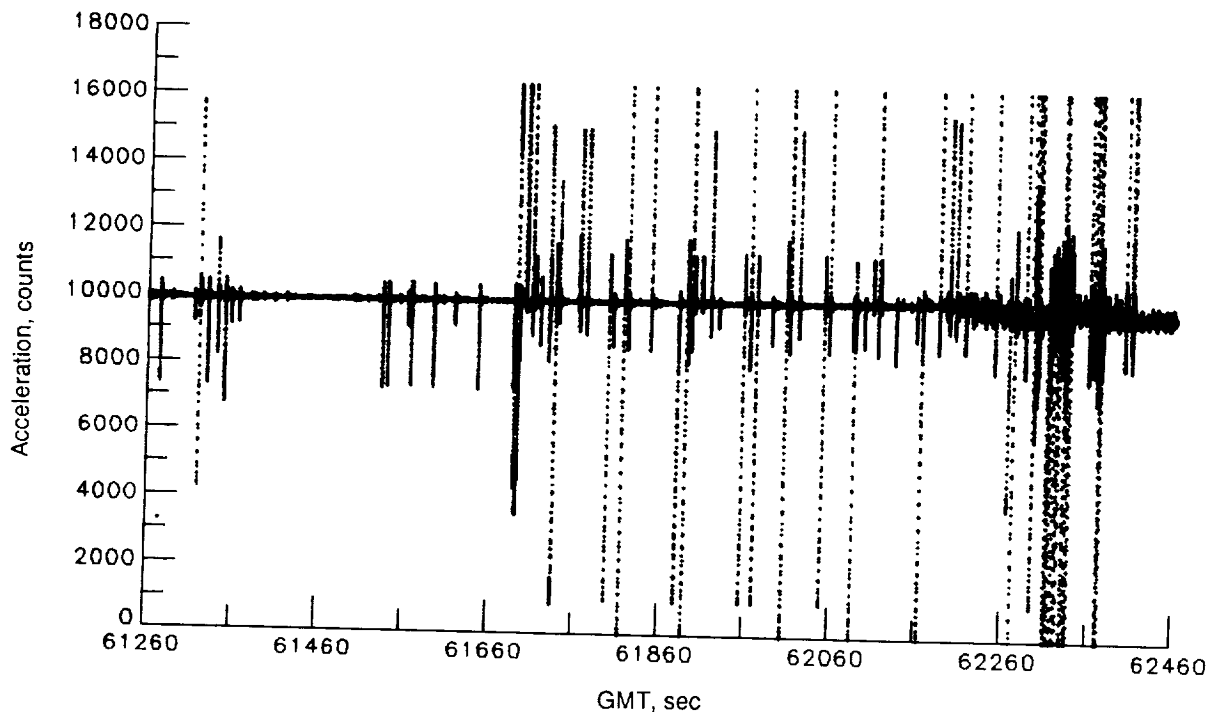


(a) X-axis acceleration counts.

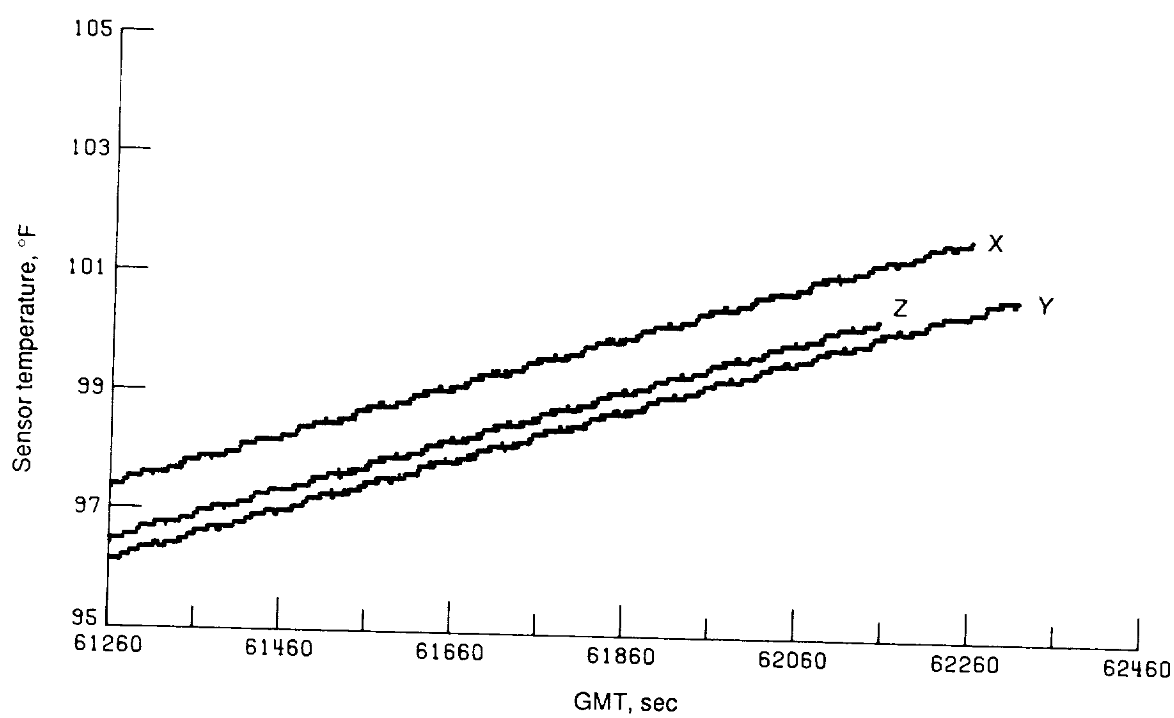


(b) Z-axis acceleration counts.

Figure 11. Acceleration counts and sensor temperature versus time for STS-61A.

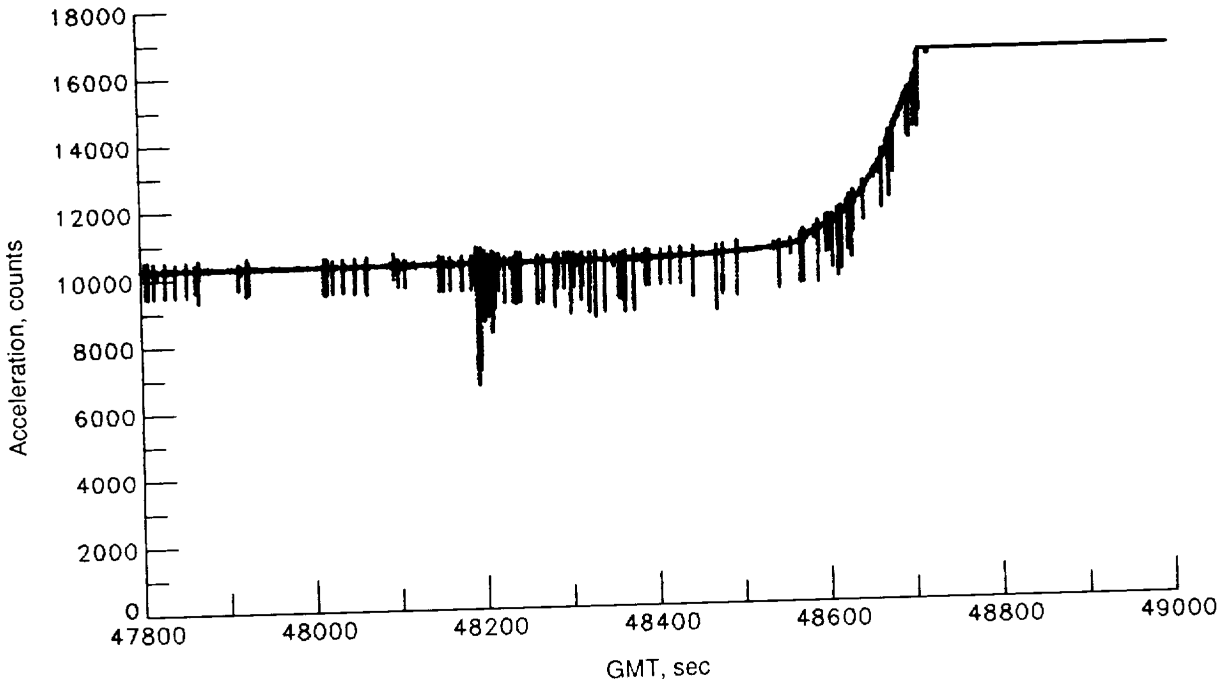


(c) Y-axis acceleration counts.

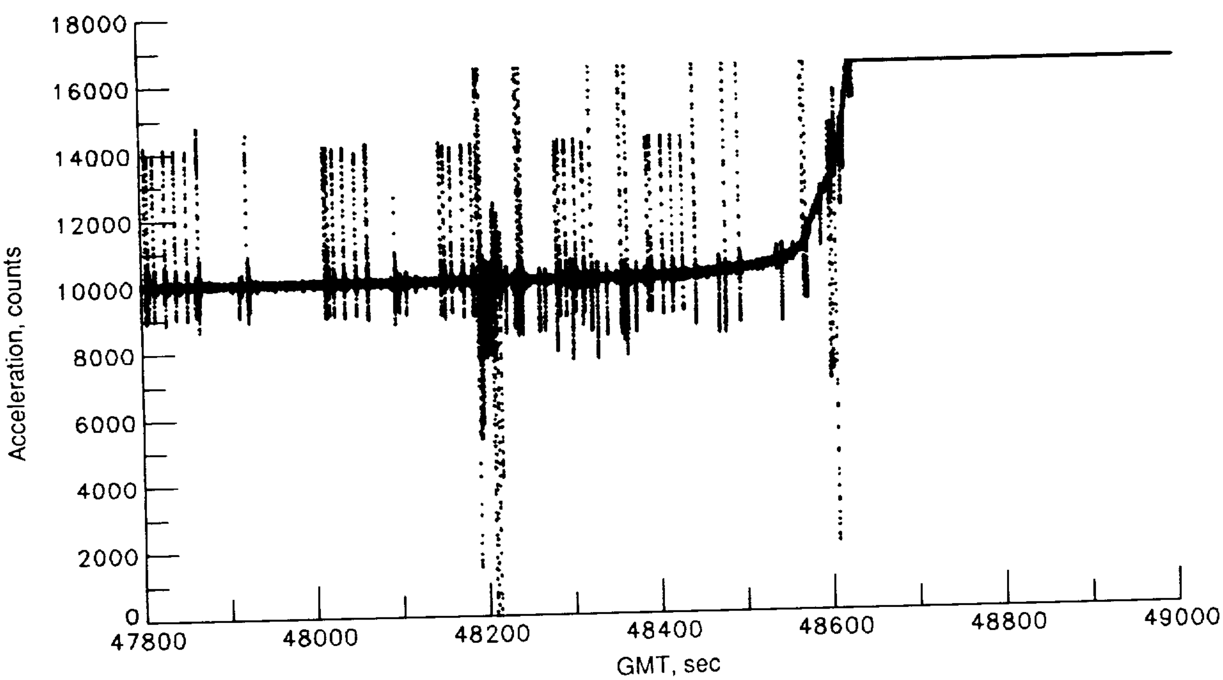


(d) X-, Y-, and Z-axis temperatures.

Figure 11. Concluded.

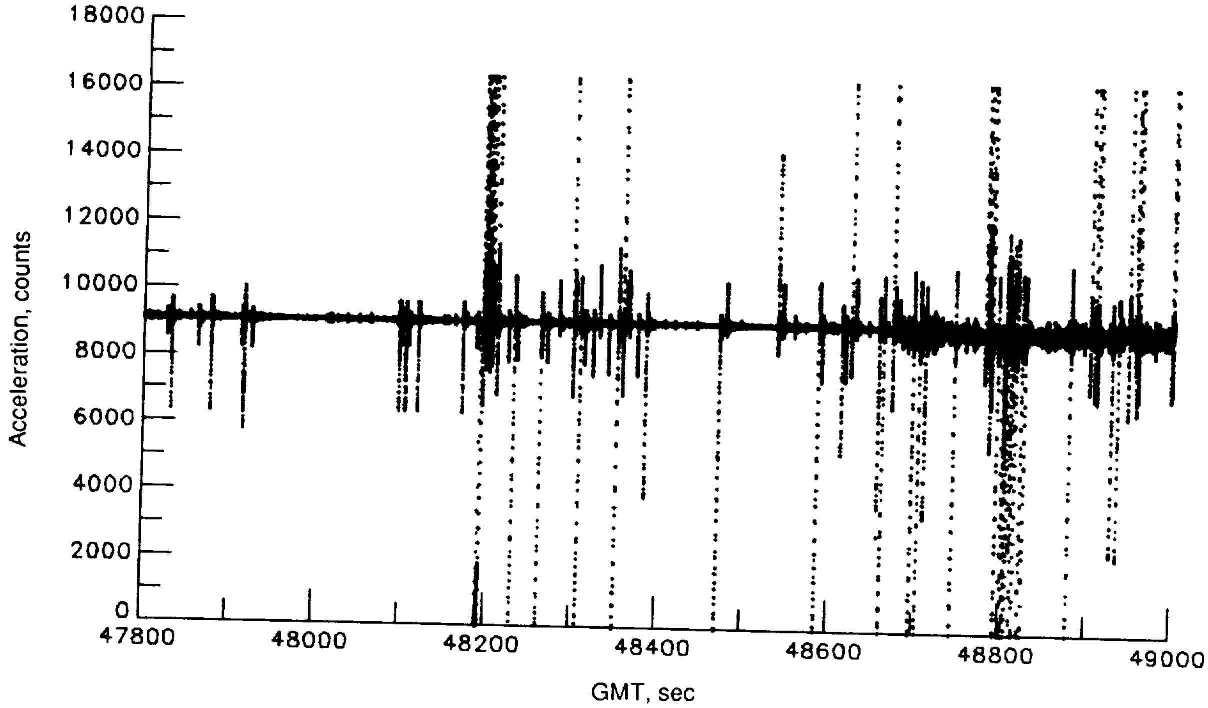


(a) X-axis acceleration counts.

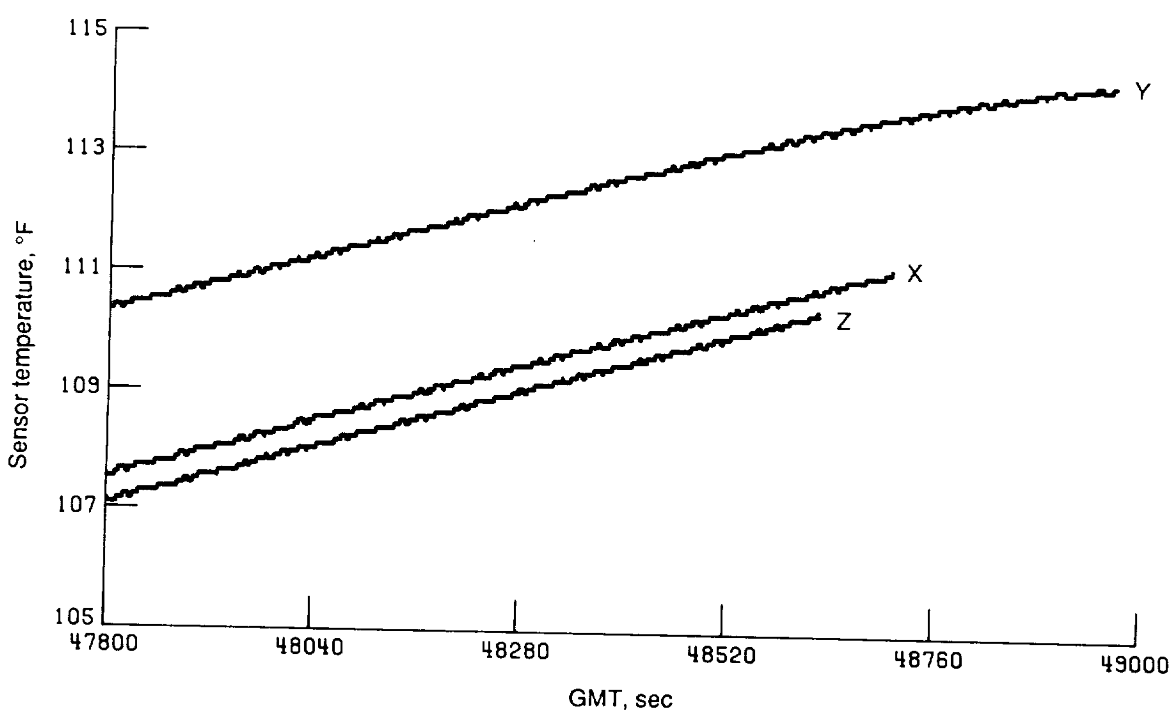


(b) Z-axis acceleration counts.

Figure 12. Acceleration counts and sensor temperature versus time for STS-61C.



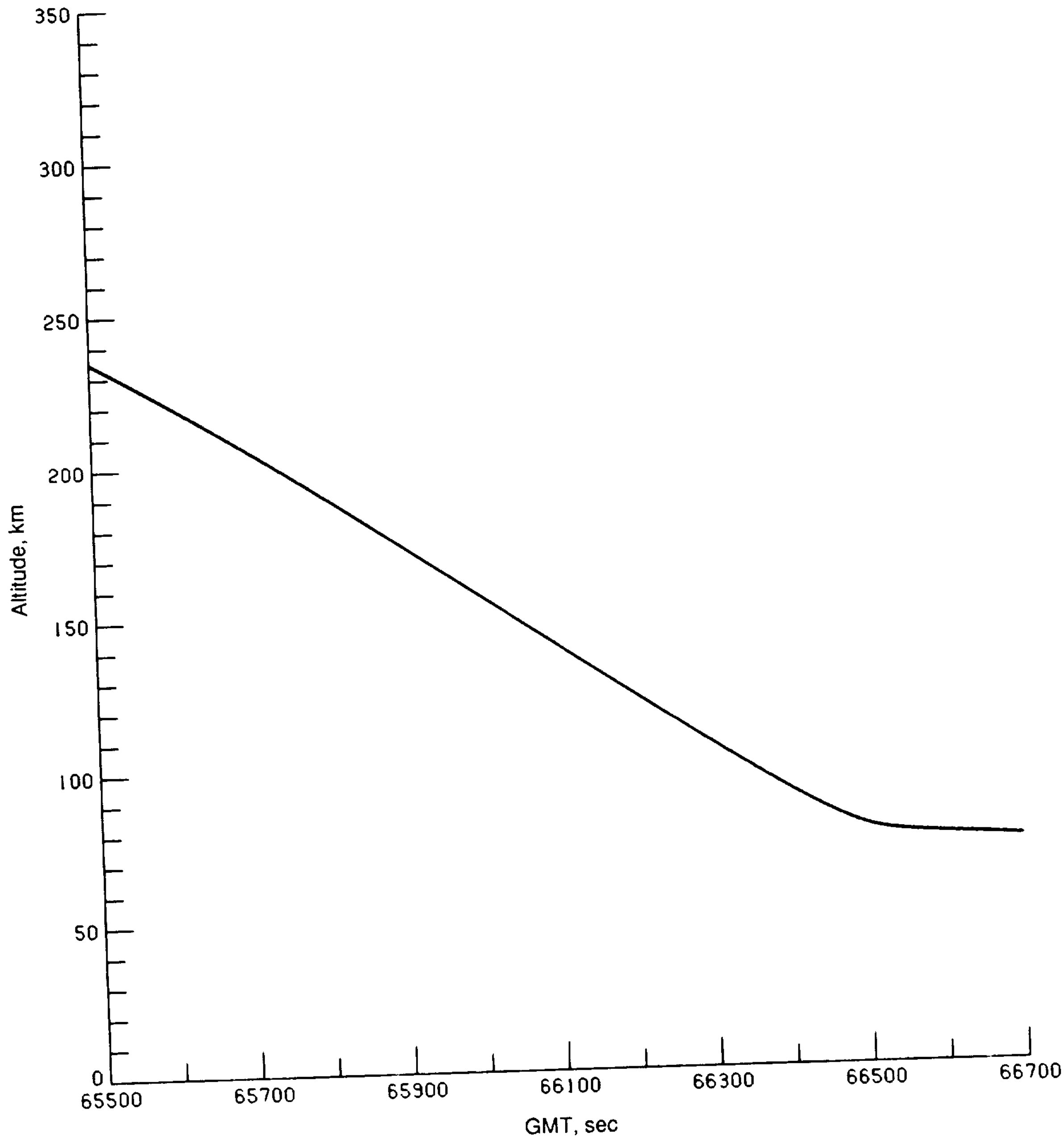
(c) Y-axis acceleration counts.



(d) X-, Y-, and Z-axis temperatures.

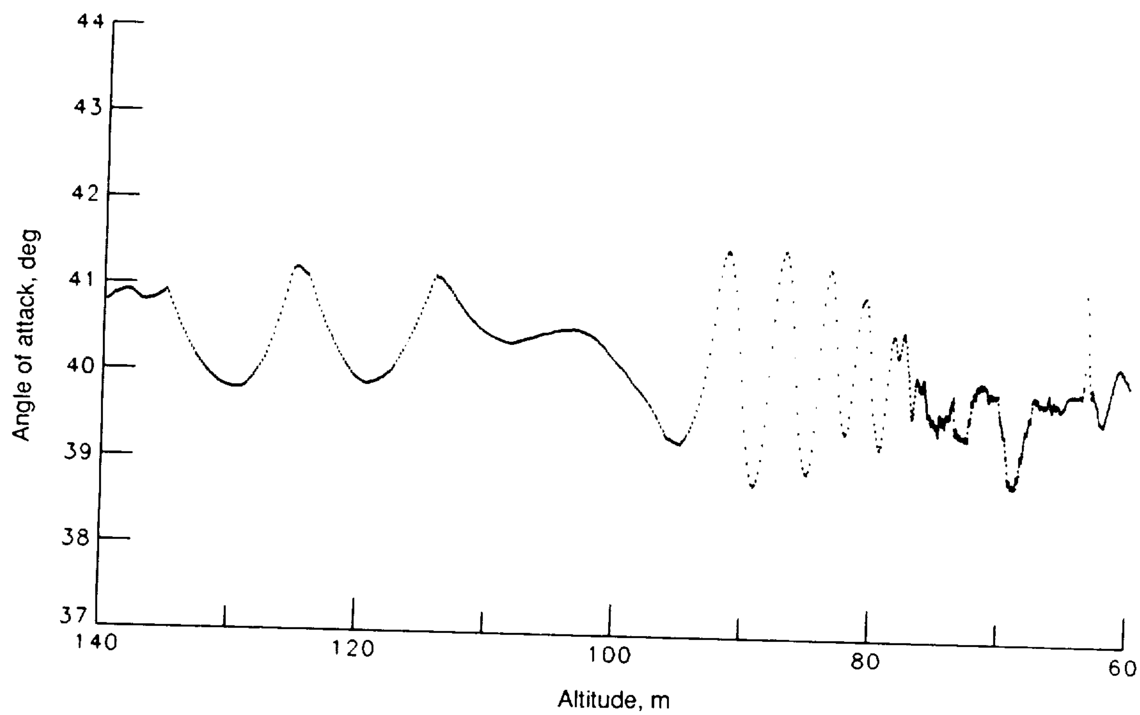
Figure 12. Concluded.

C-3

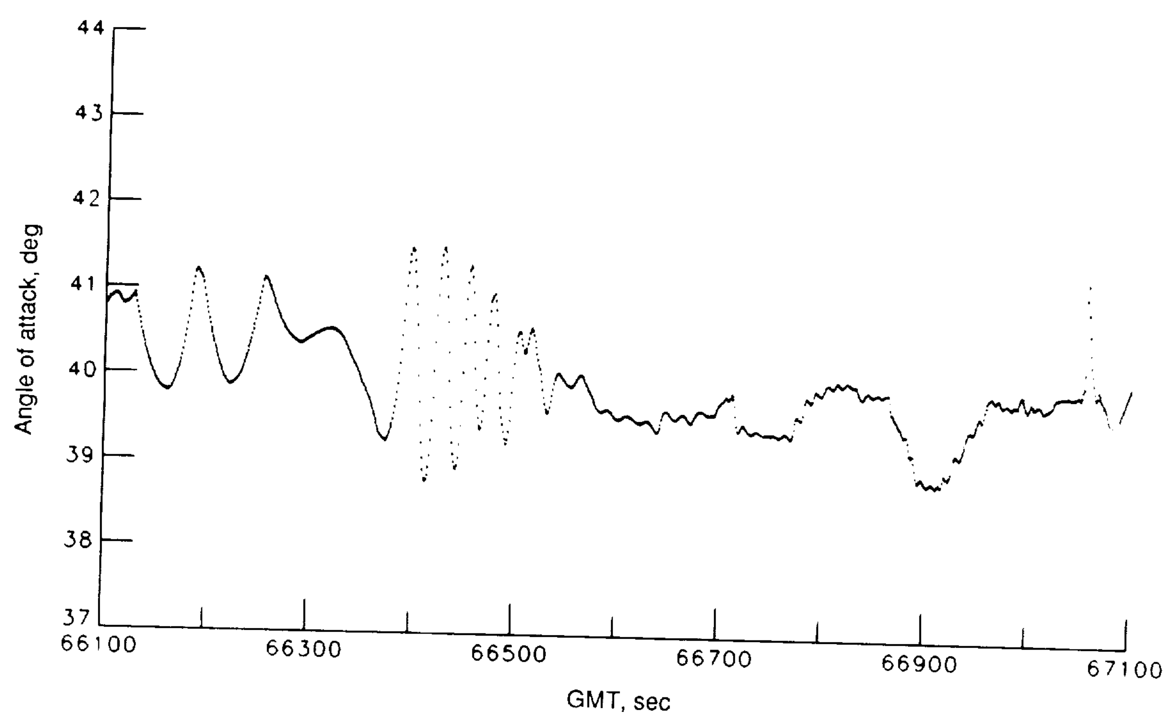


(a) Altitude versus time.

Figure 13. Time and altitude histories of orbiter state vector subset data for STS-06.

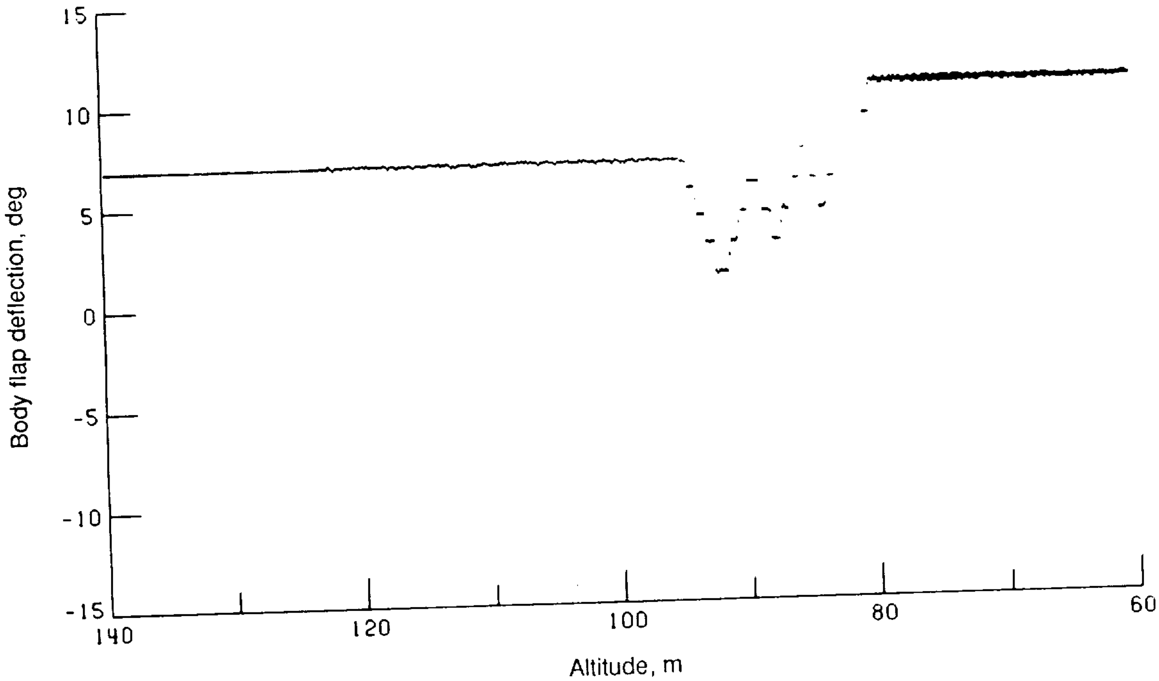


(b) Angle of attack versus altitude.

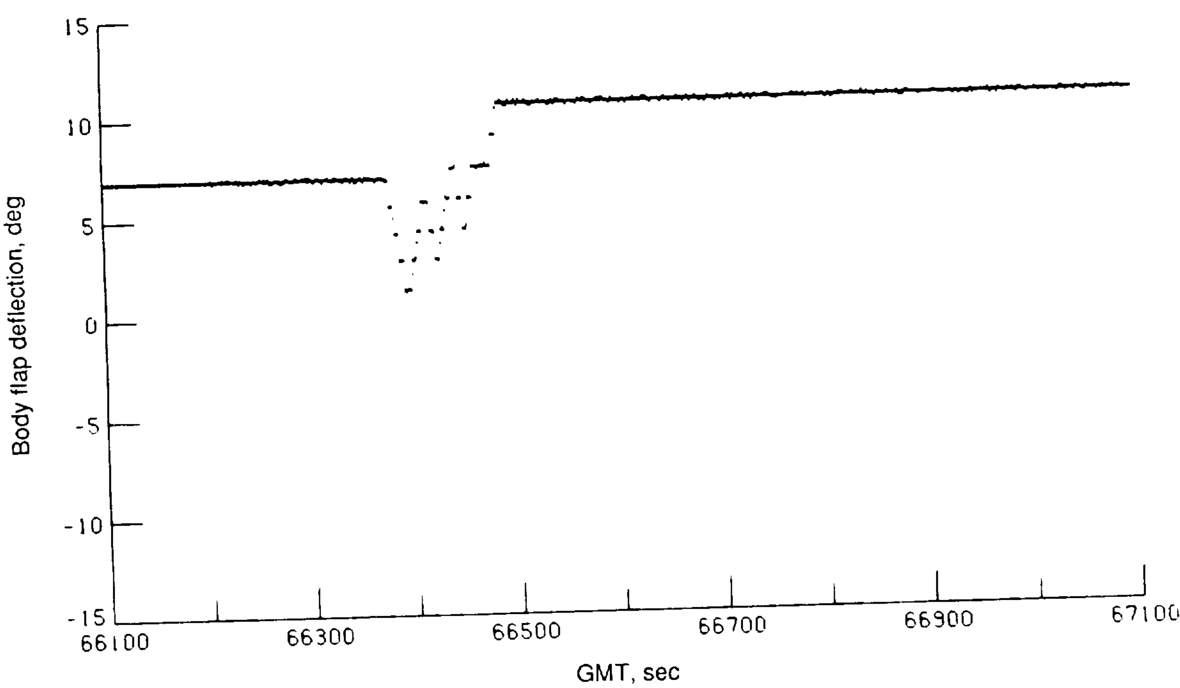


(c) Angle of attack versus time.

Figure 13. Continued.

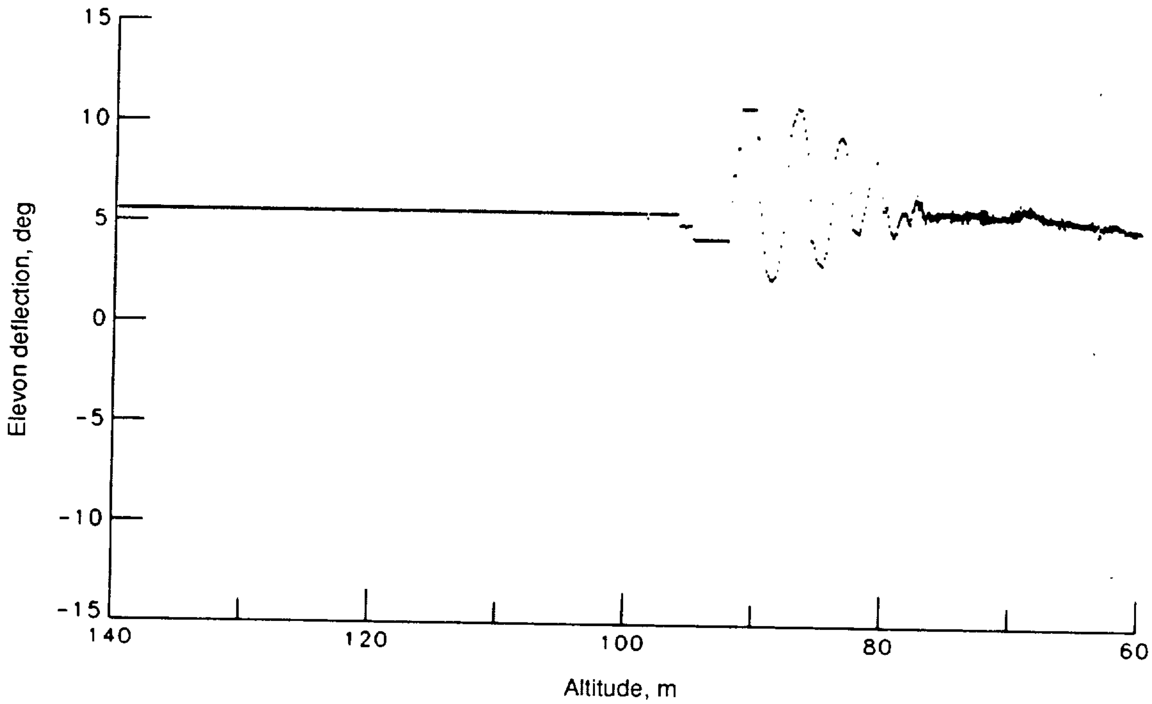


(d) Body flap deflection versus altitude.

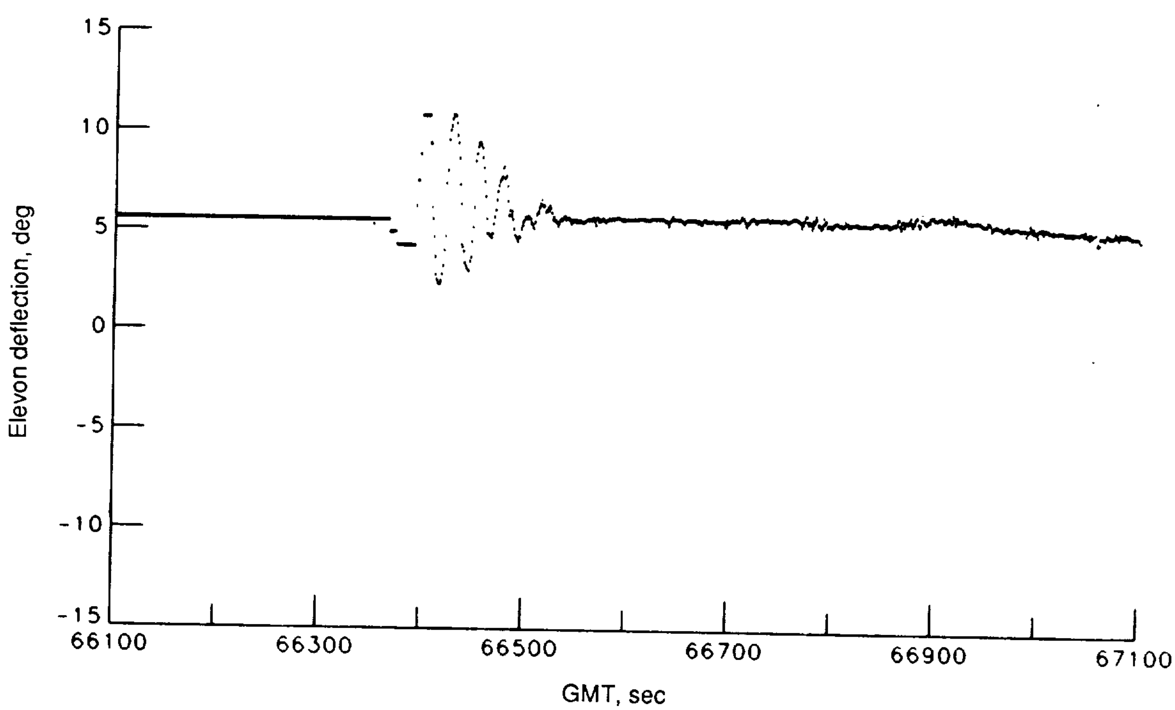


(e) Body flap deflection versus time.

Figure 13. Continued.

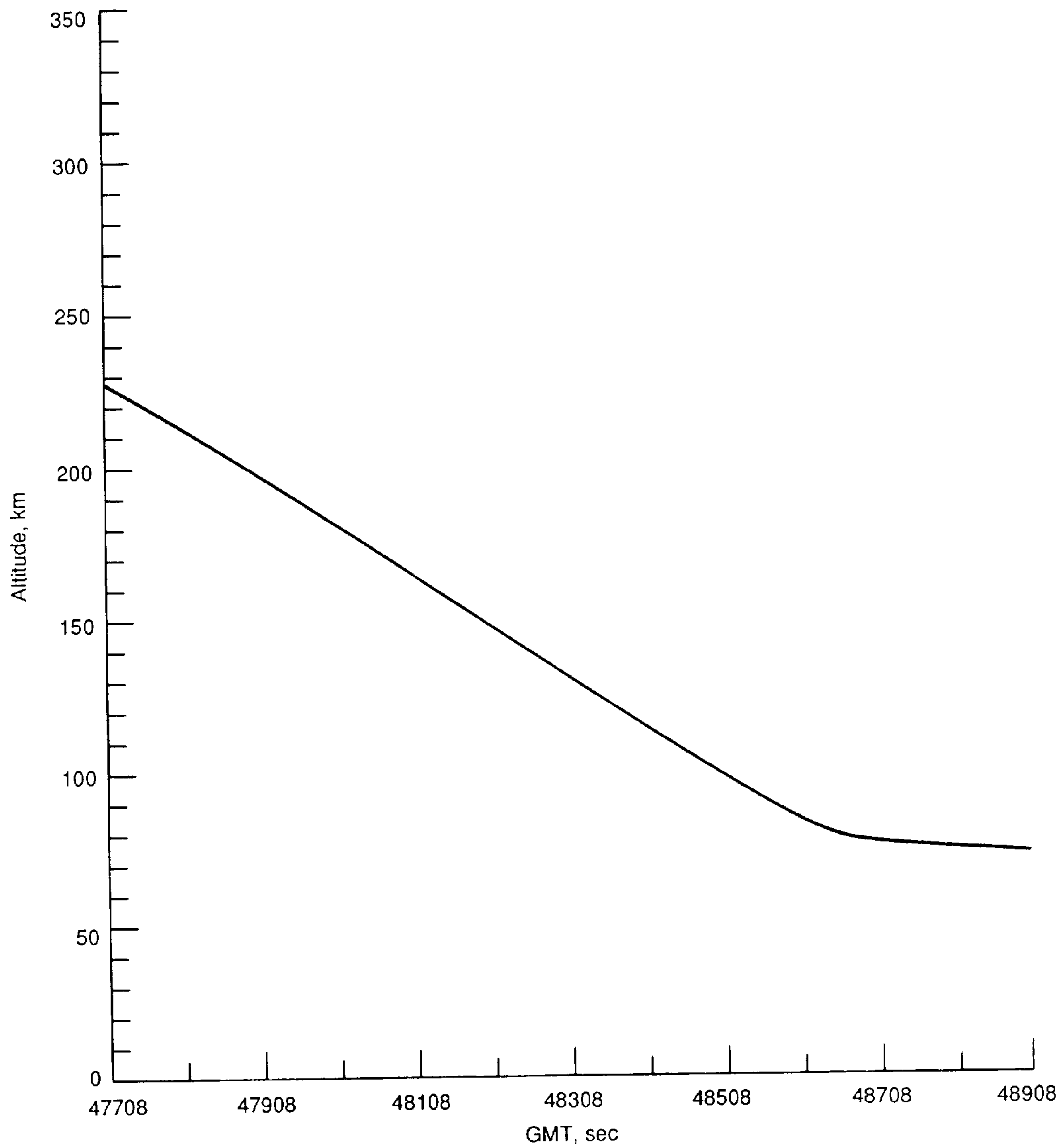


(f) Elevon deflection versus altitude.



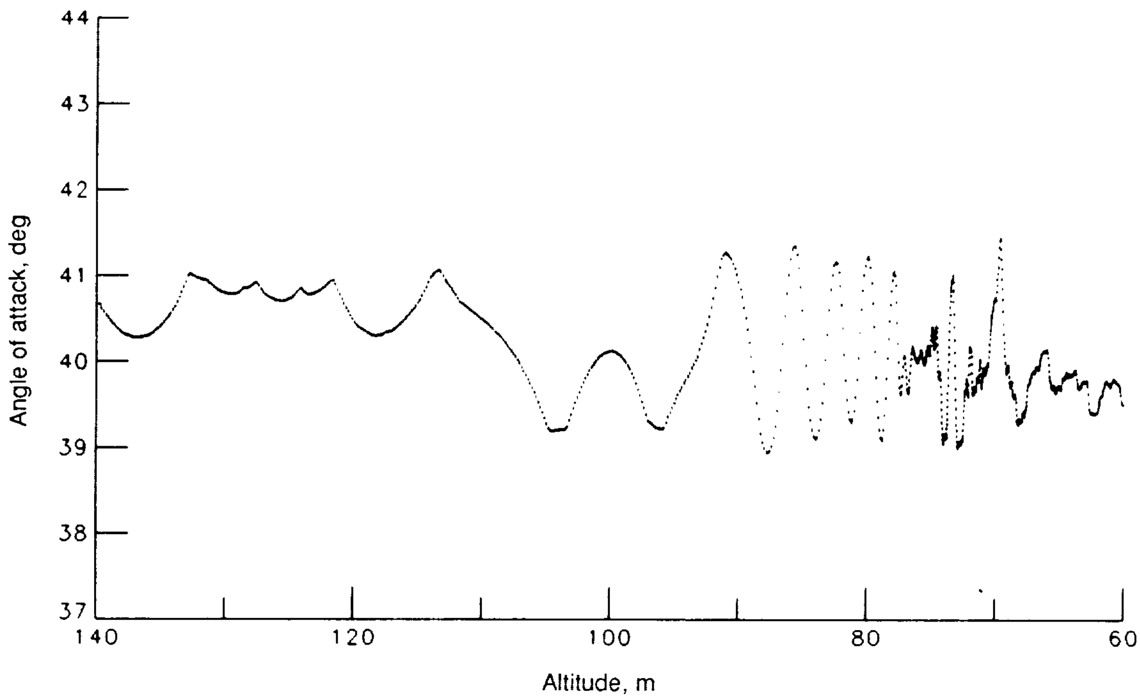
(g) Elevon deflection versus time.

Figure 13. Concluded.

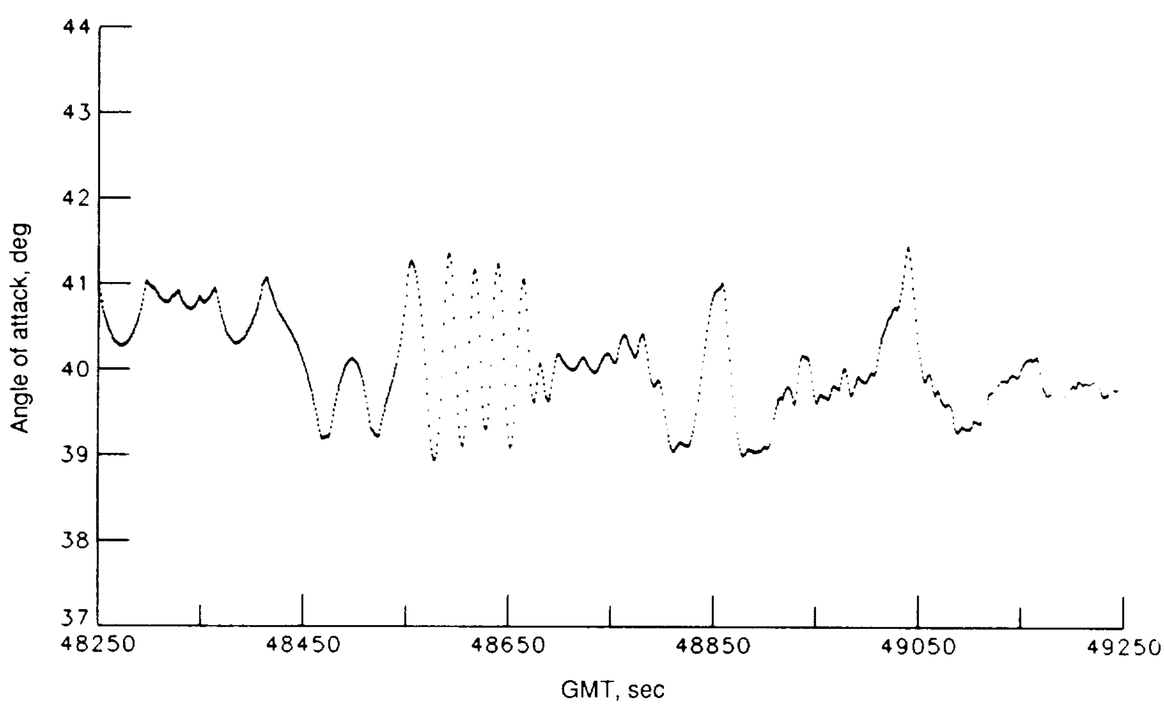


(a) Altitude versus time.

Figure 14. Time and altitude histories of orbiter state vector data subset for STS-07.

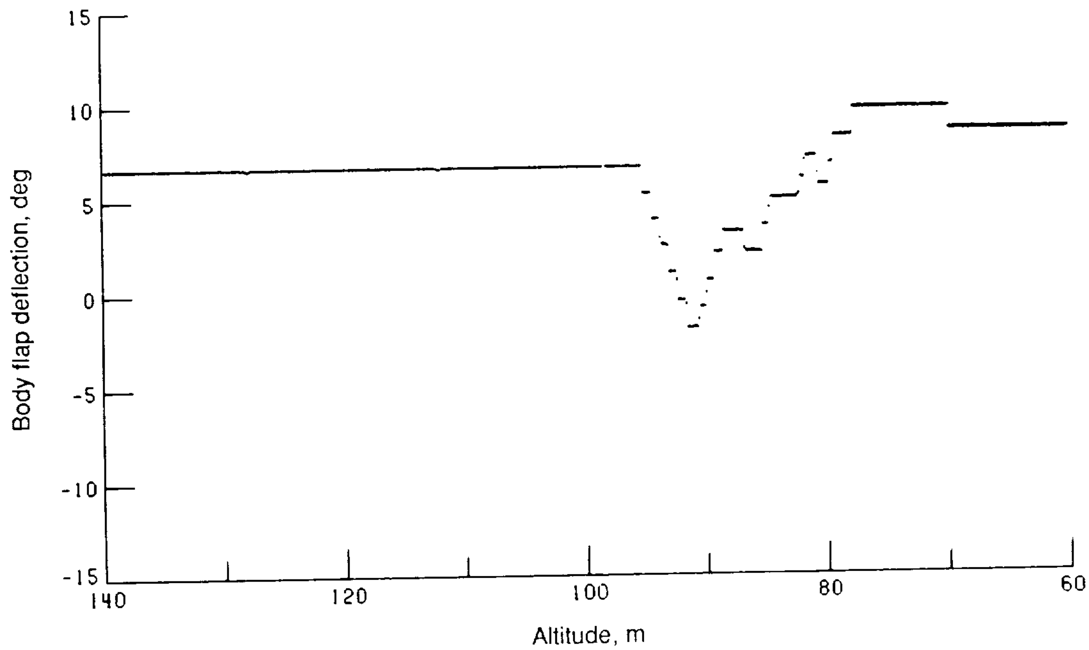


(b) Angle of attack versus altitude.

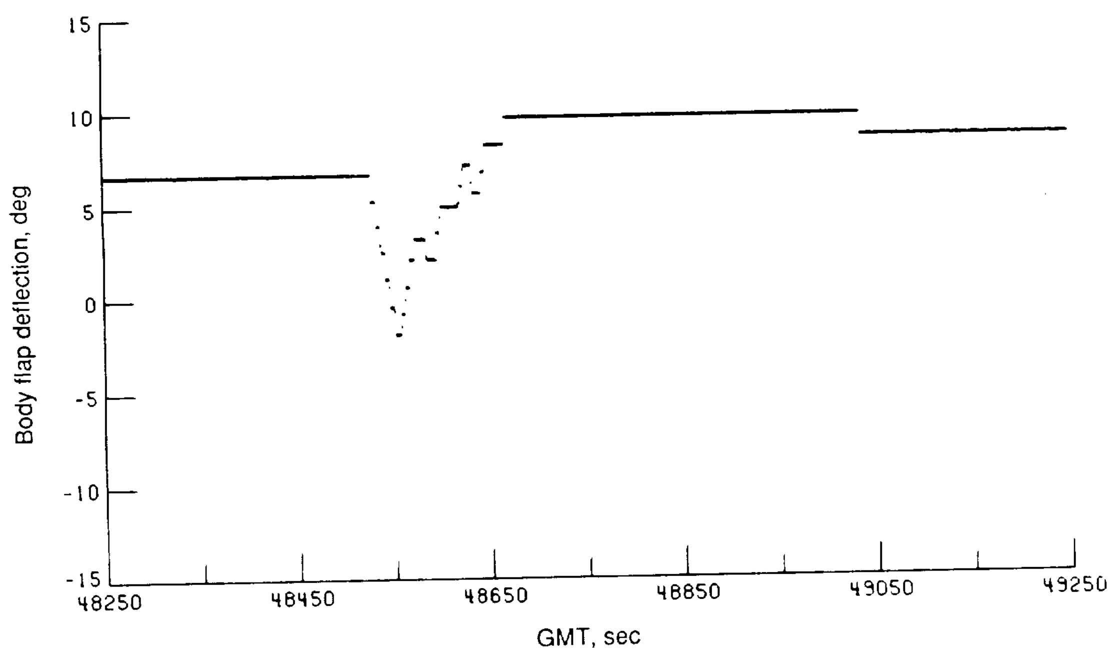


(c) Angle of attack versus time.

Figure 14. Continued.

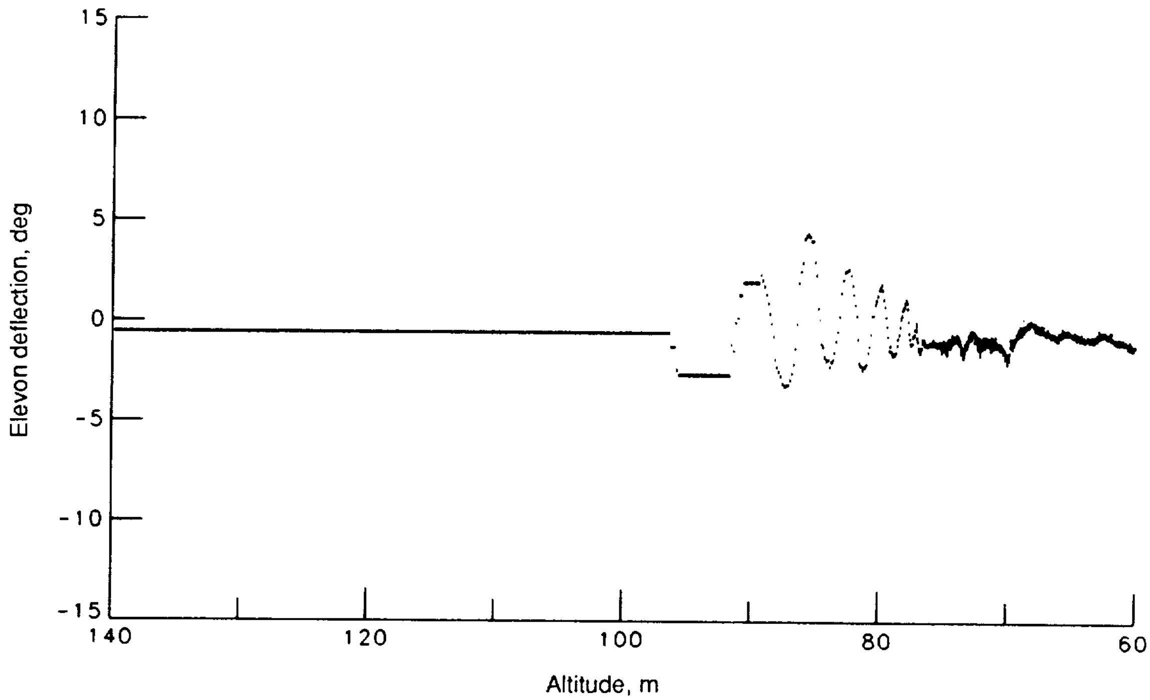


(d) Body flap deflection versus altitude.

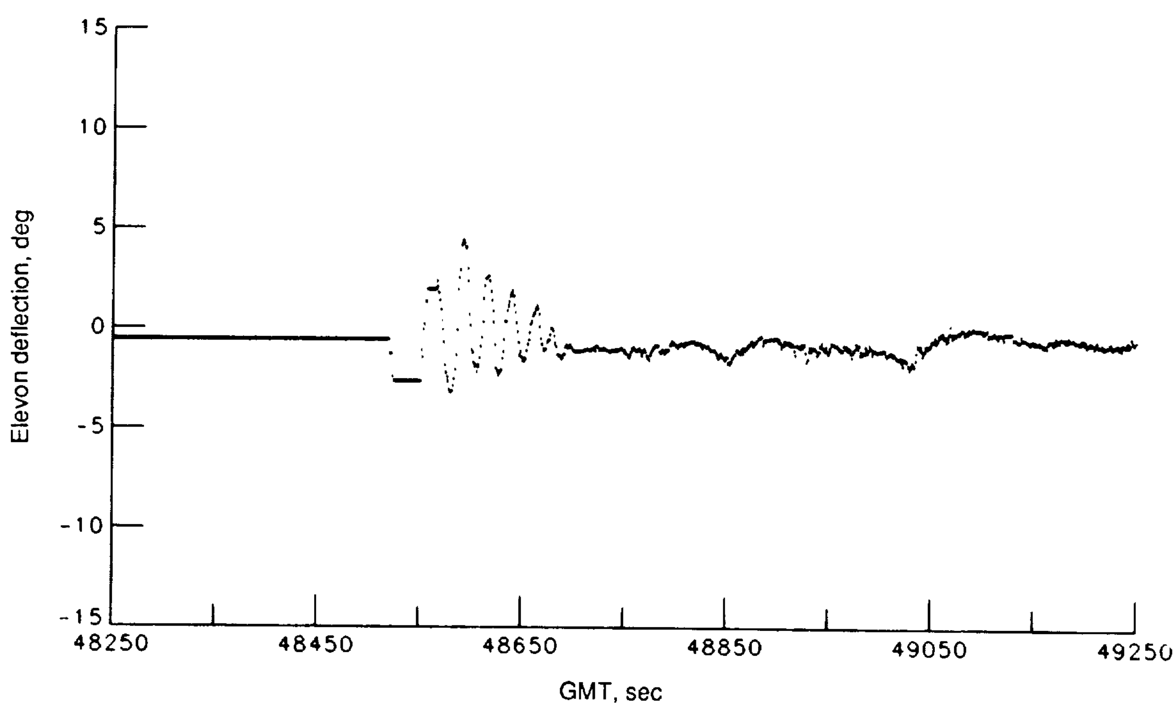


(e) Body flap deflection versus time.

Figure 14. Continued.

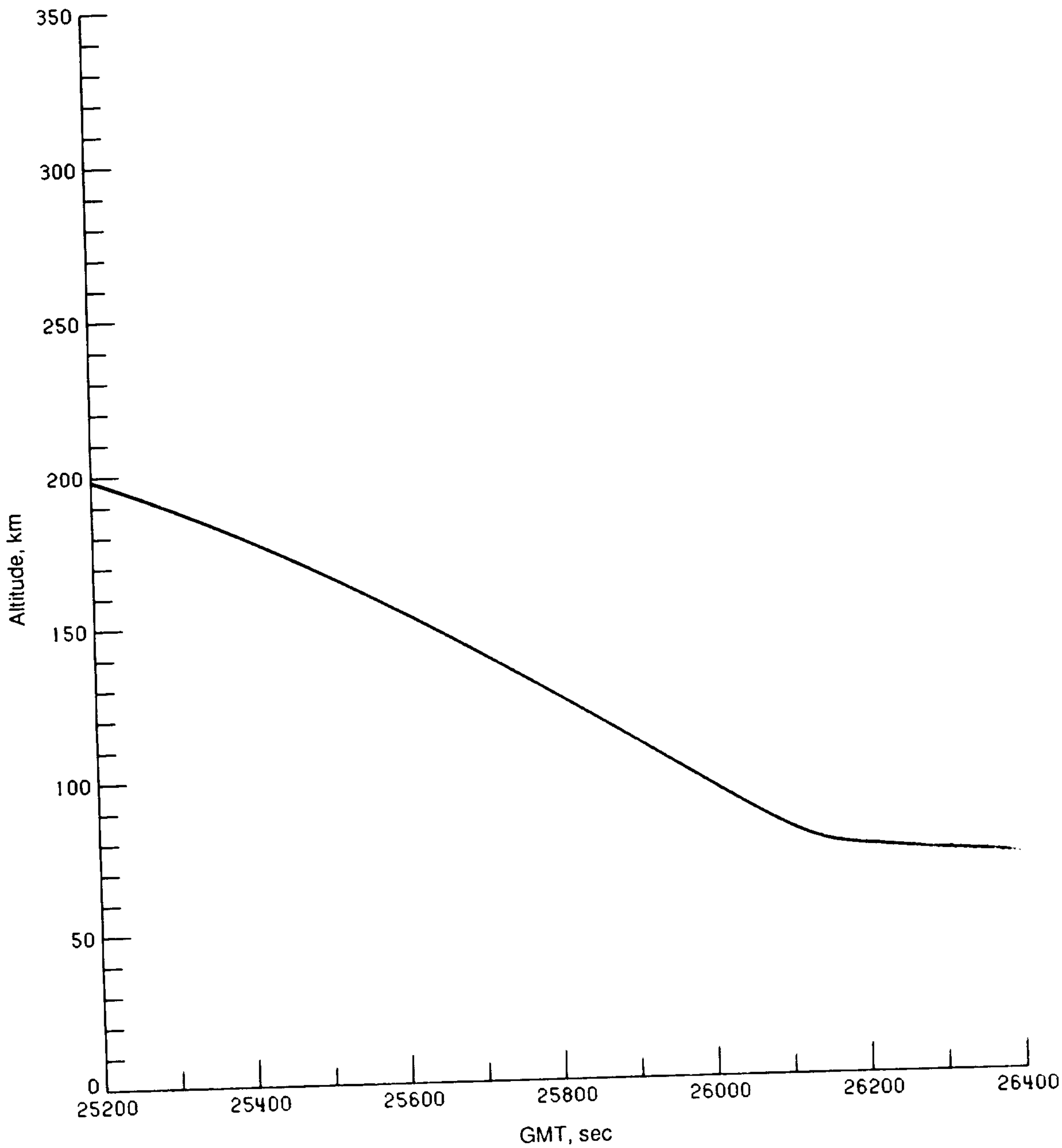


(f) Elevon deflection versus altitude.



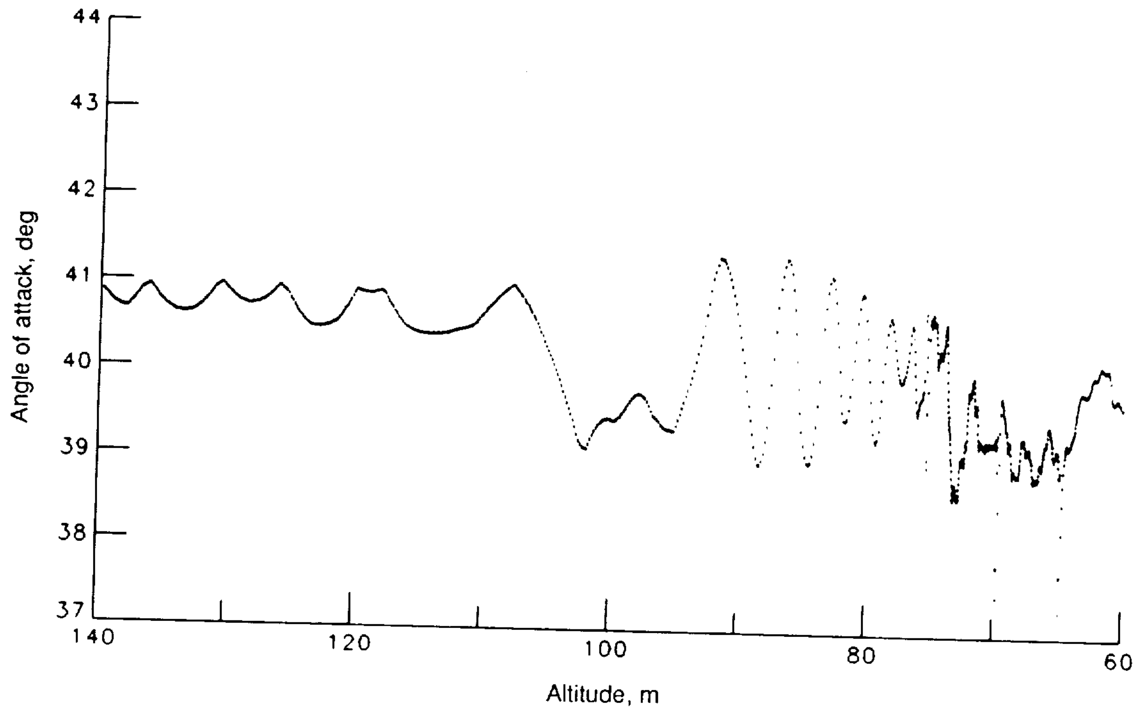
(g) Elevon deflection versus time.

Figure 14. Concluded.

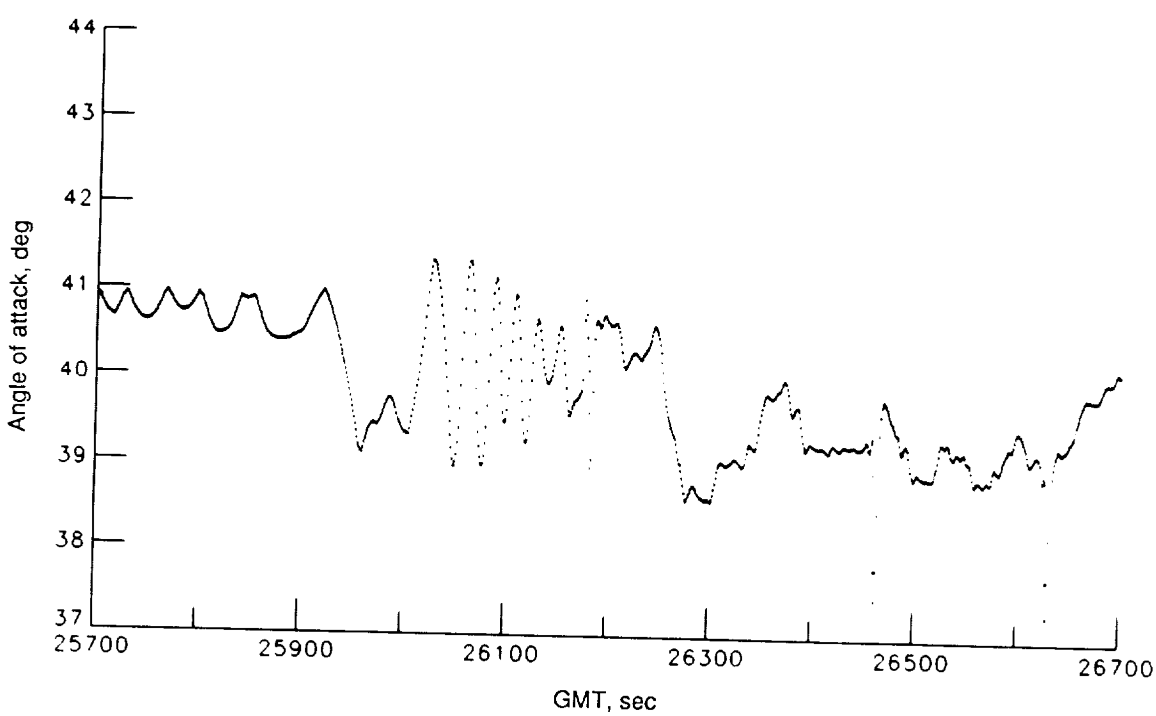


(a) Altitude versus time.

Figure 15. Time and altitude histories of orbiter state vector data subset for STS-08.

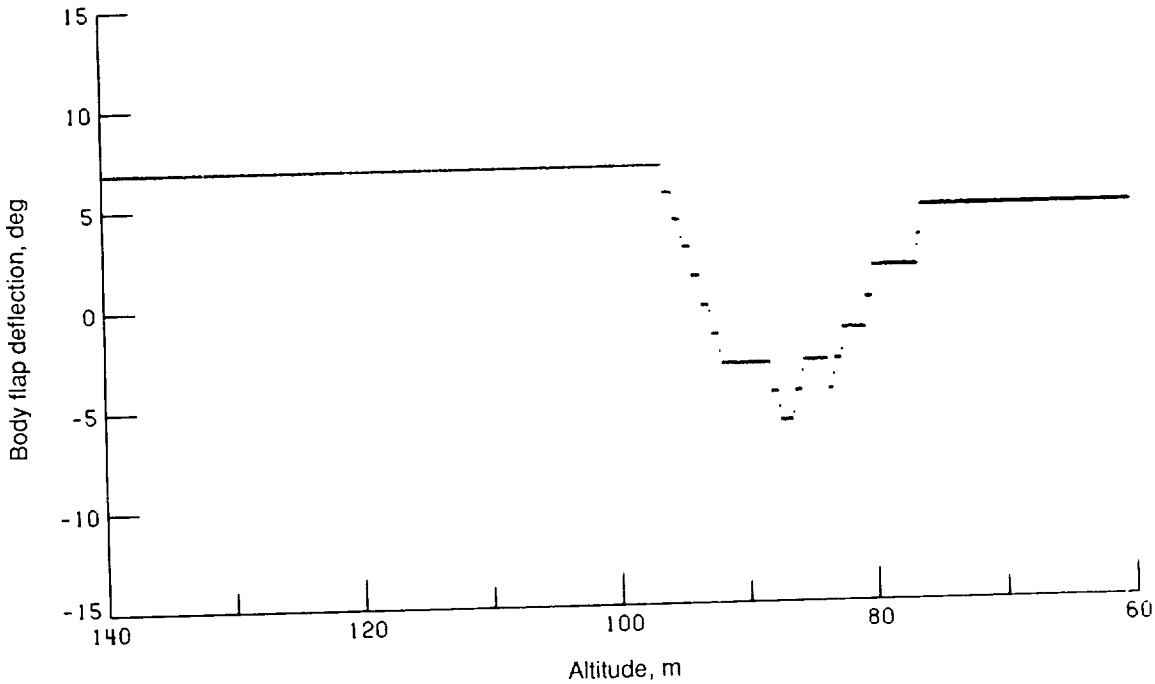


(b) Angle of attack versus altitude.

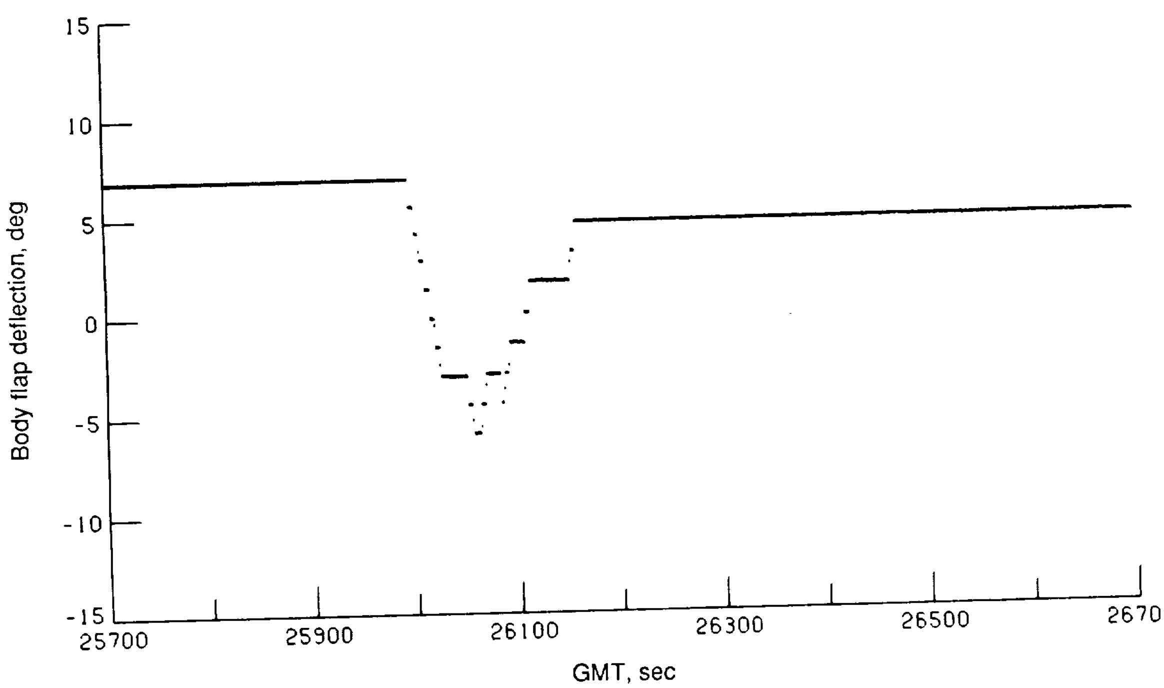


(c) Angle of attack versus time.

Figure 15. Continued.

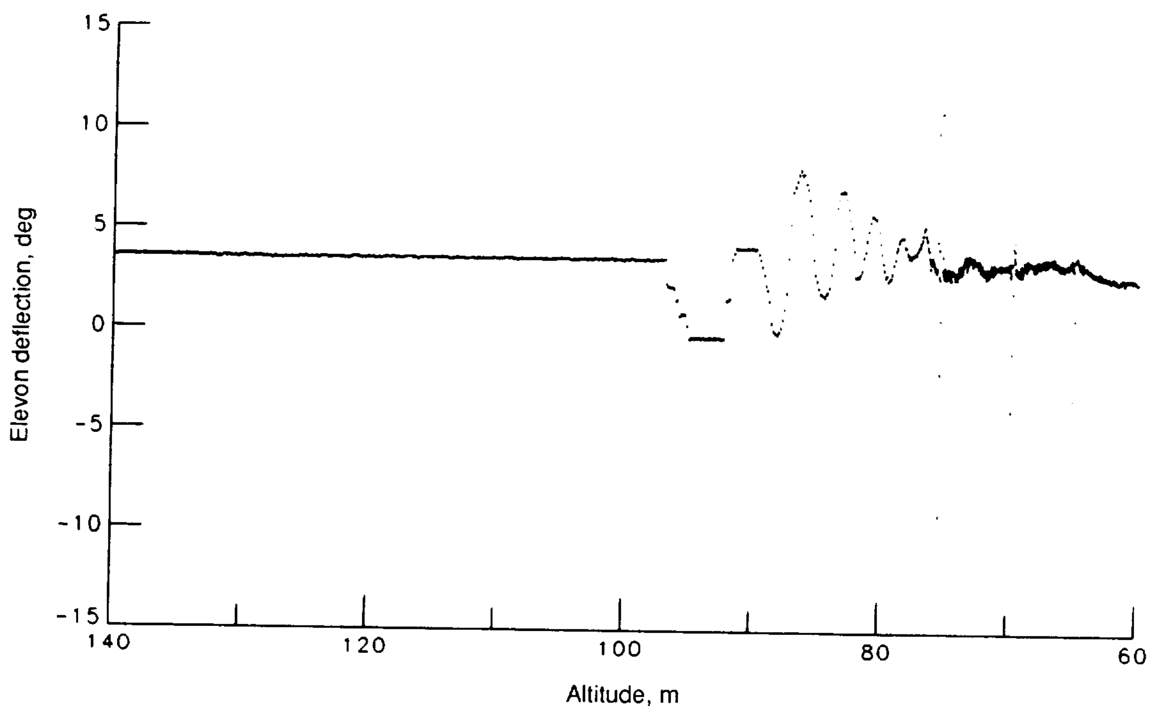


(d) Body flap deflection versus altitude.

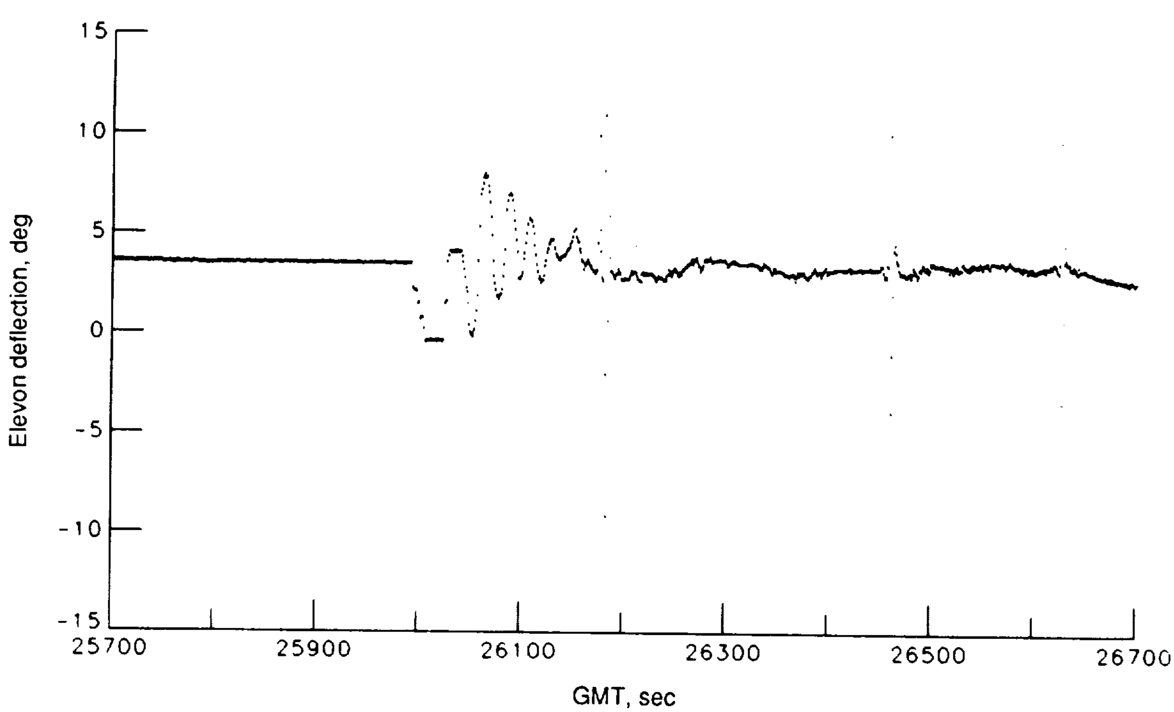


(e) Body flap deflection versus time.

Figure 15. Continued.

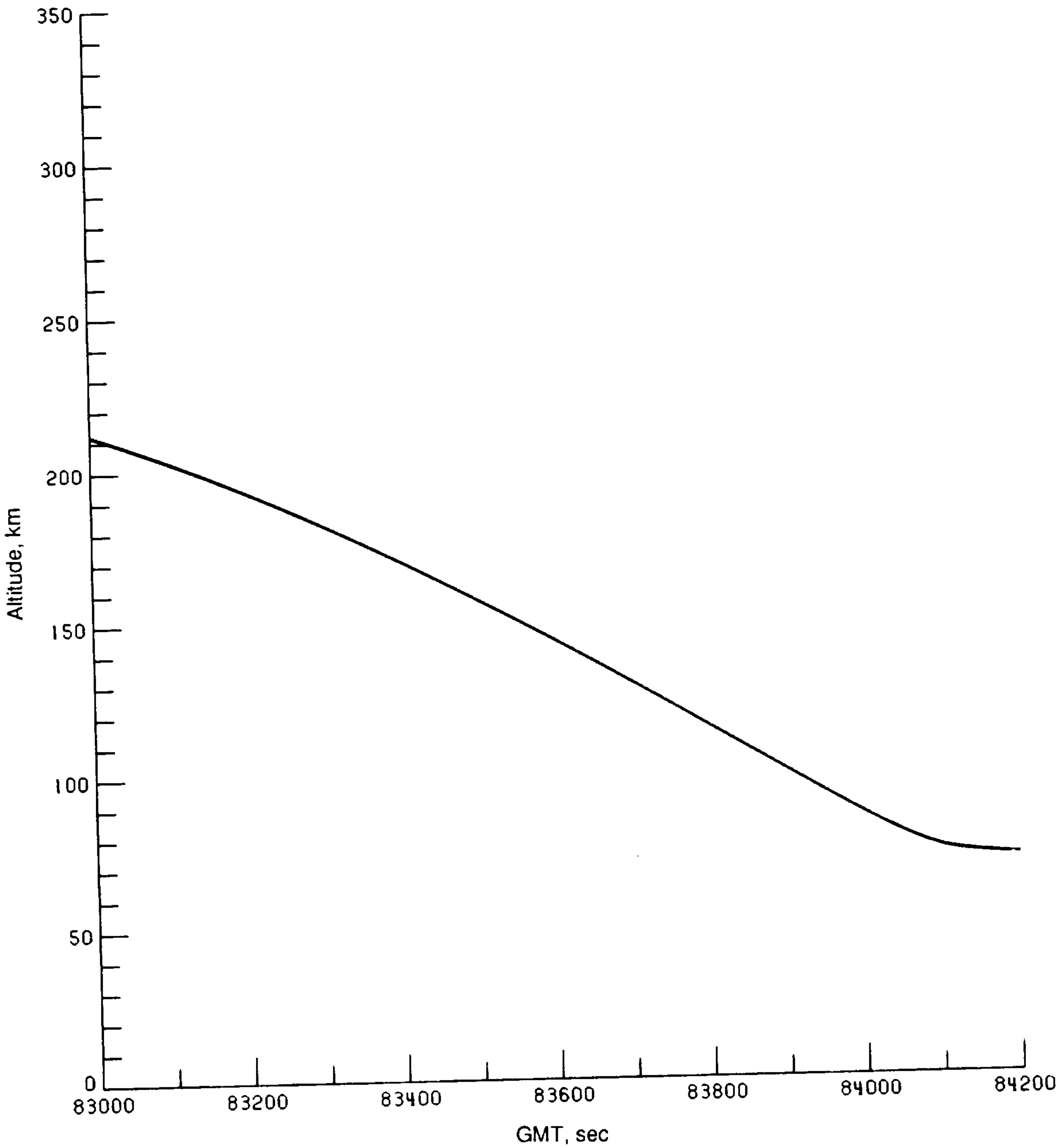


(f) Elevon deflection versus altitude.



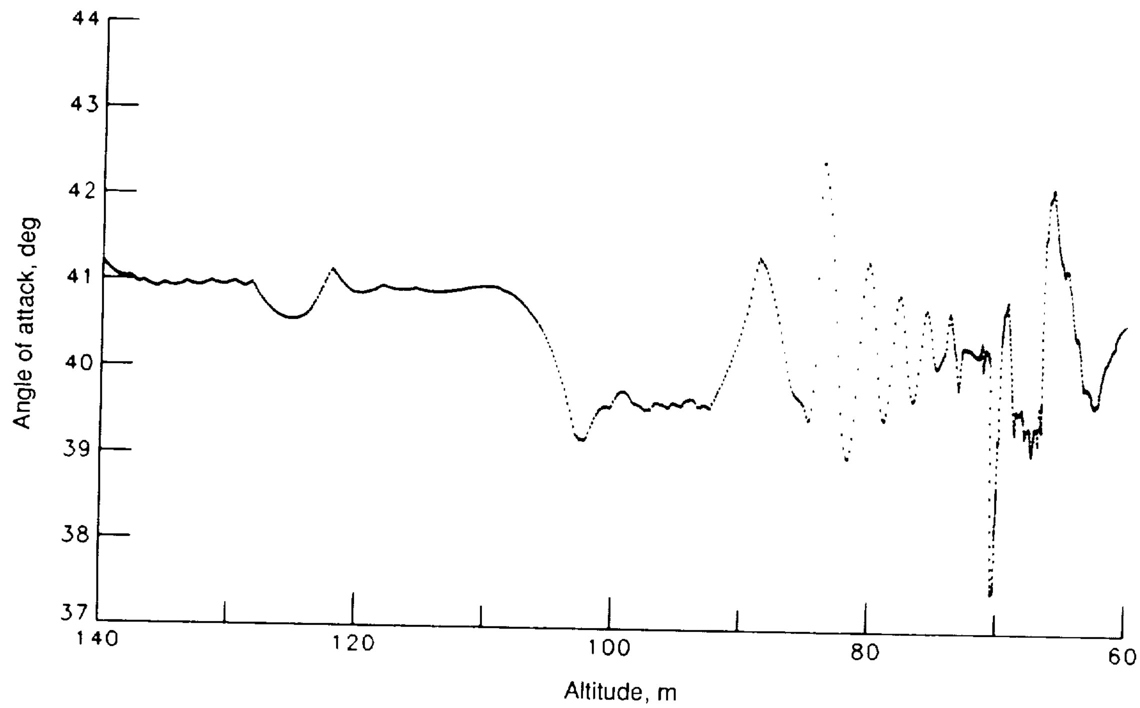
(g) Elevon deflection versus time.

Figure 15. Concluded.

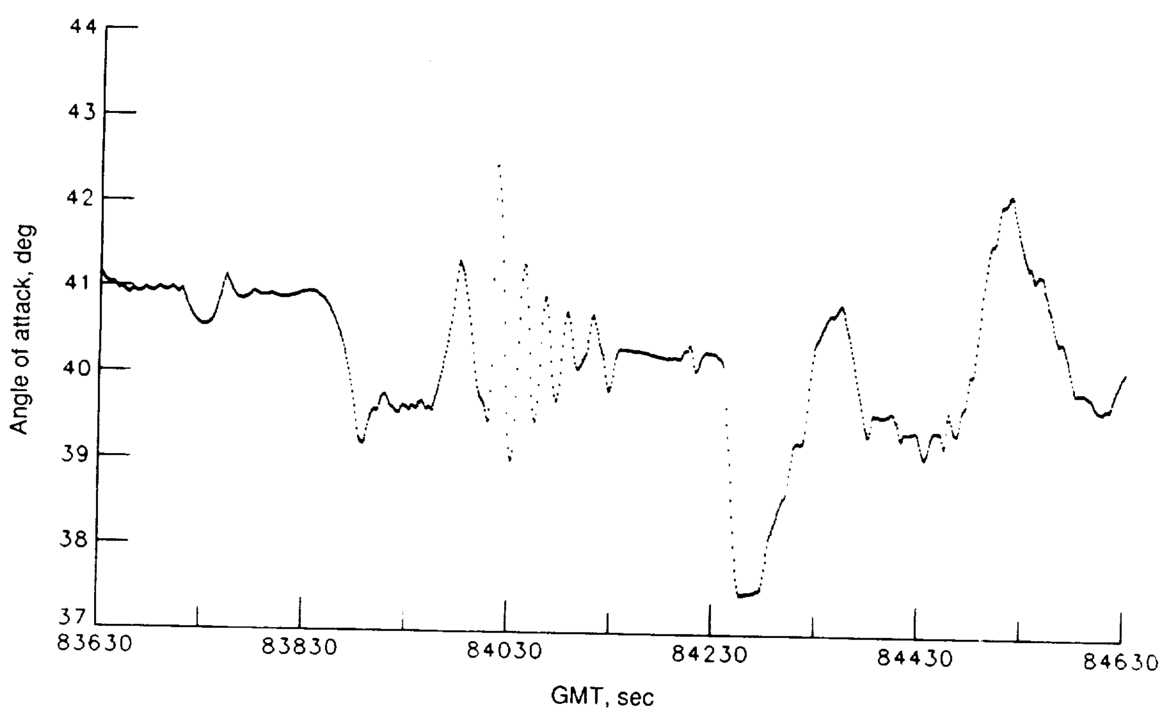


(a) Altitude versus time.

Figure 16. Time and altitude histories of orbiter state vector data subset for STS-09.

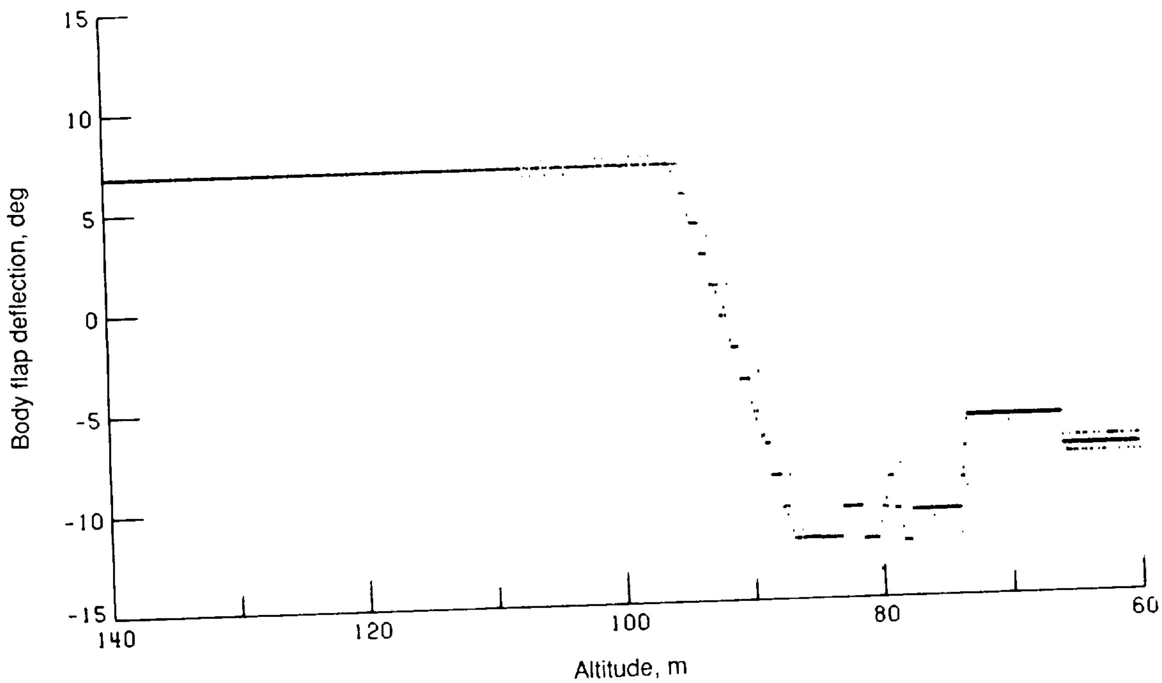


(b) Angle of attack versus altitude.

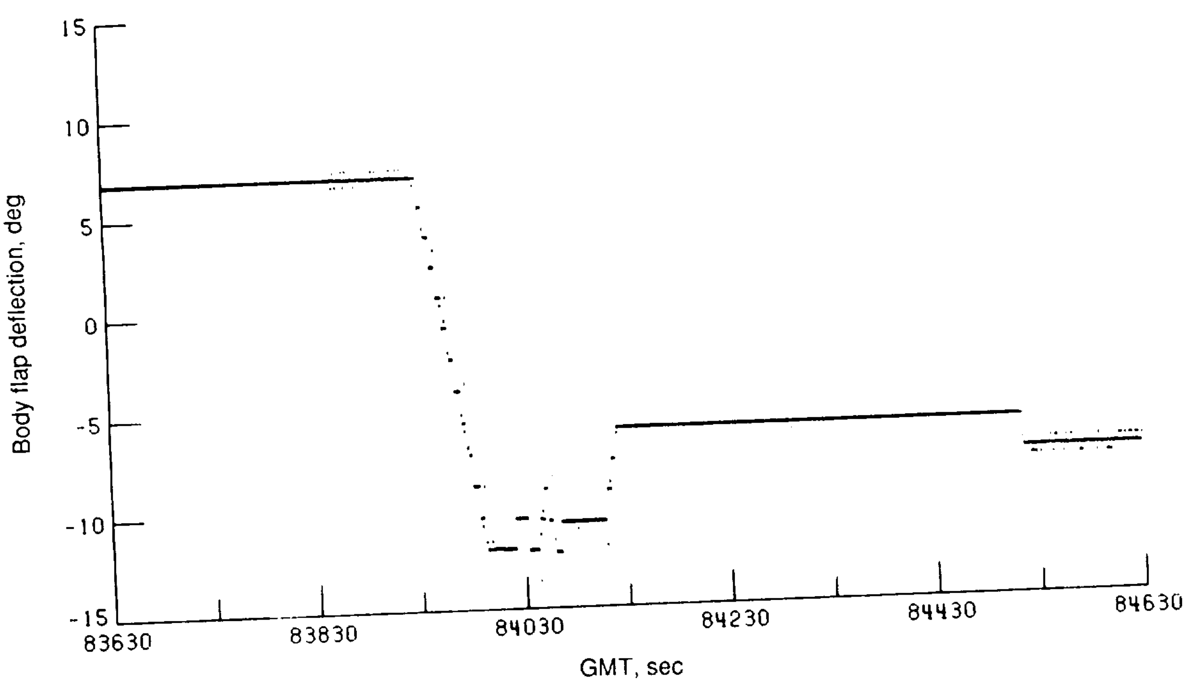


(c) Angle of attack versus time.

Figure 16. Continued.

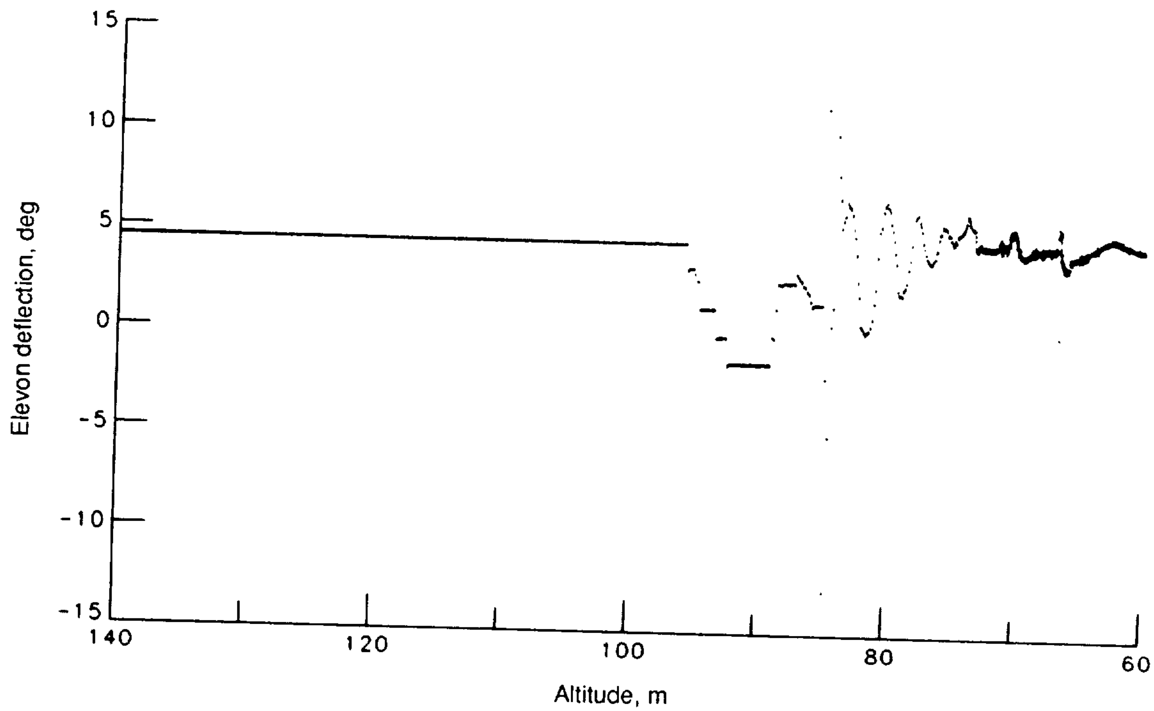


(d) Body flap deflection versus altitude.

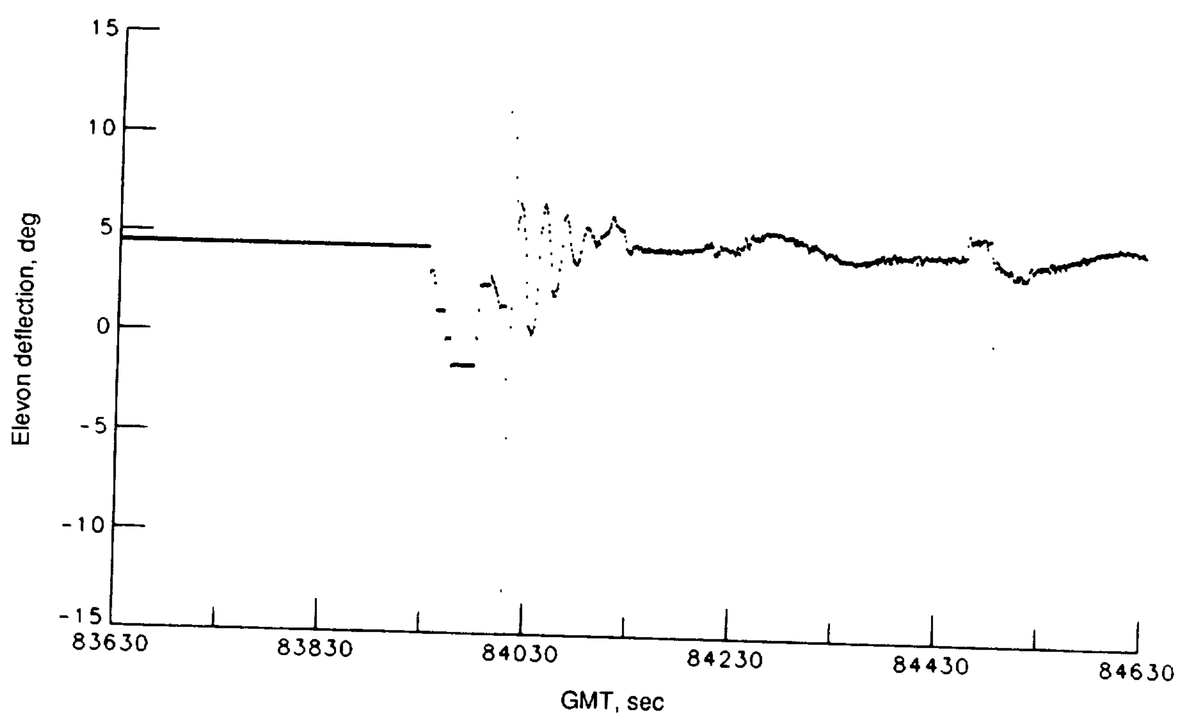


(e) Body flap deflection versus time.

Figure 16. Continued.

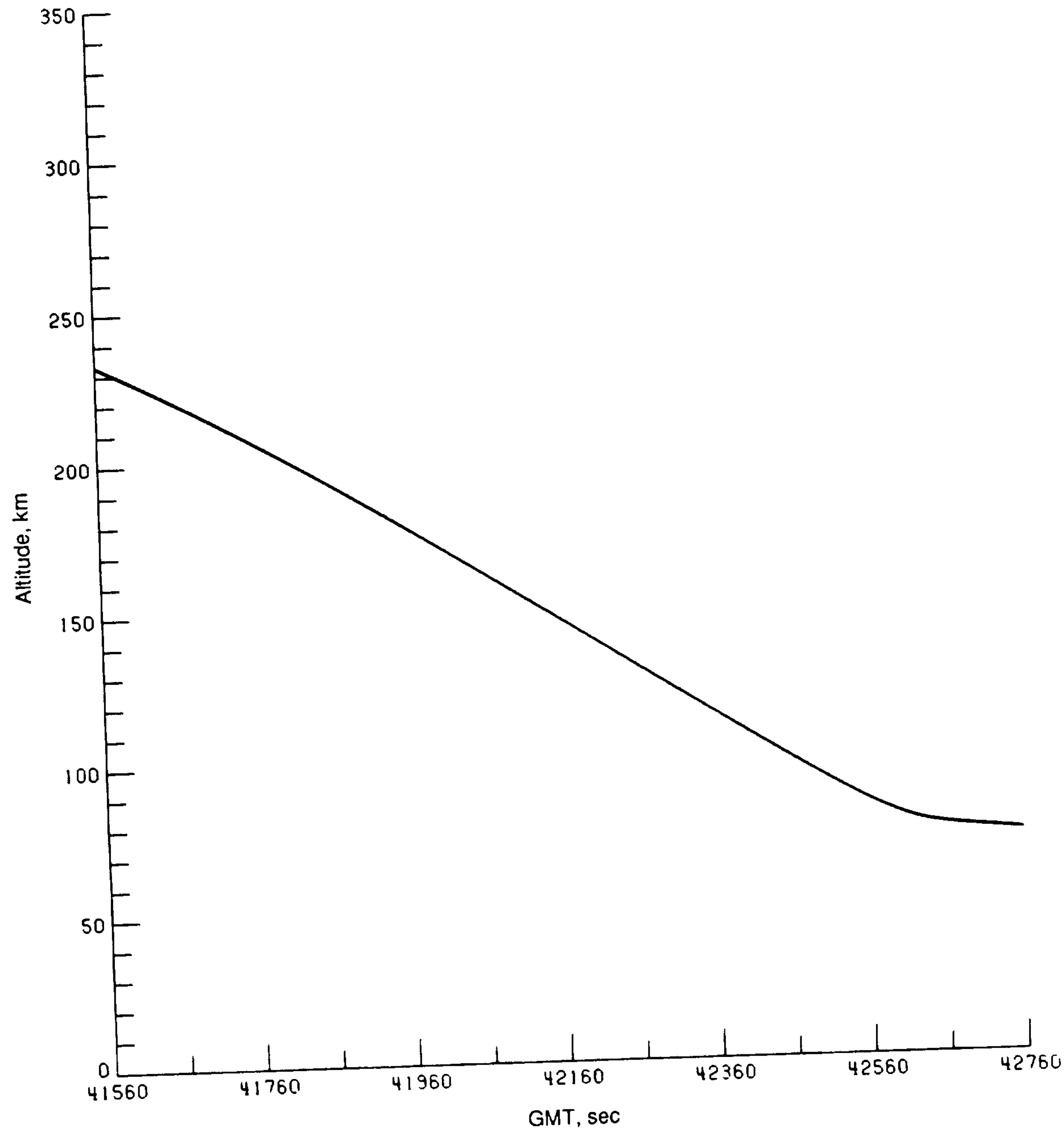


(f) Elevon deflection versus altitude.



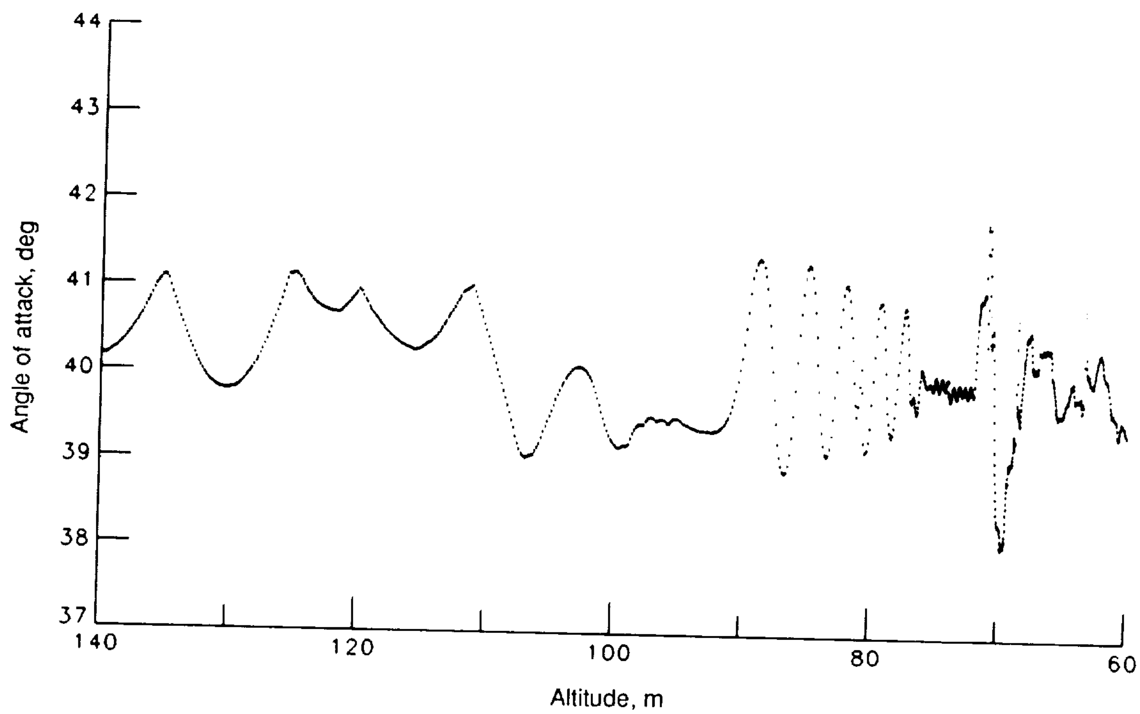
(g) Elevon deflection versus time.

Figure 16. Concluded.

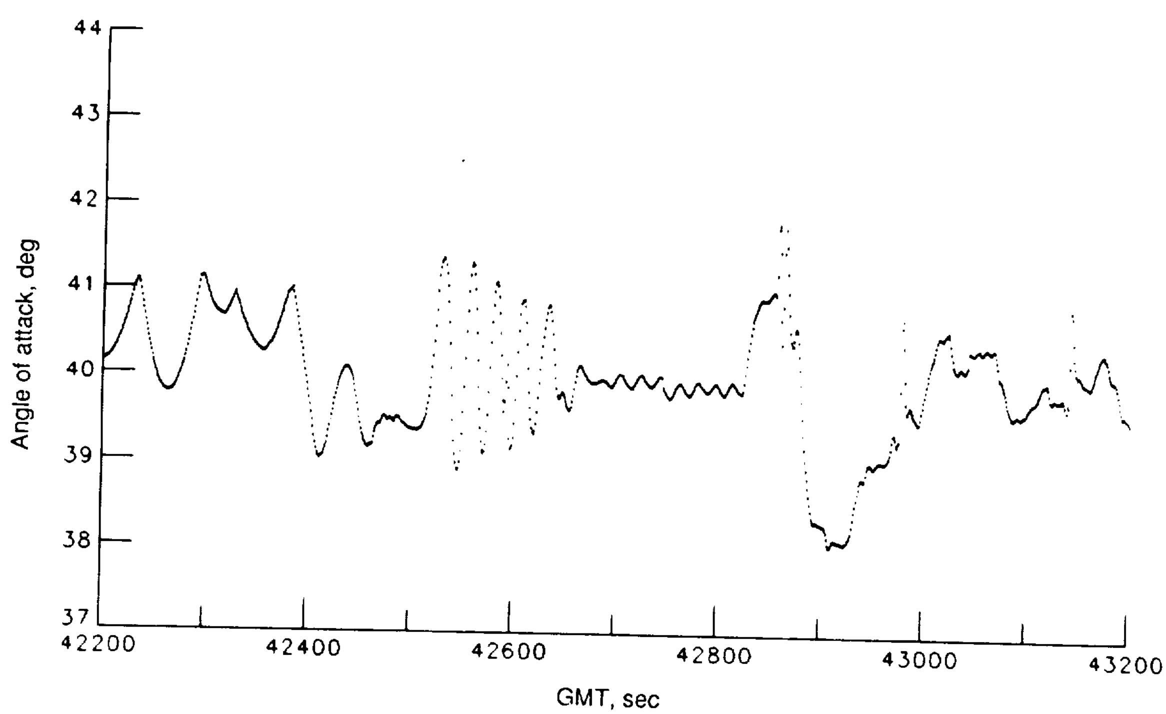


(a) Altitude versus time.

Figure 17. Time and altitude histories of orbiter state vector data subset for STS-41B.

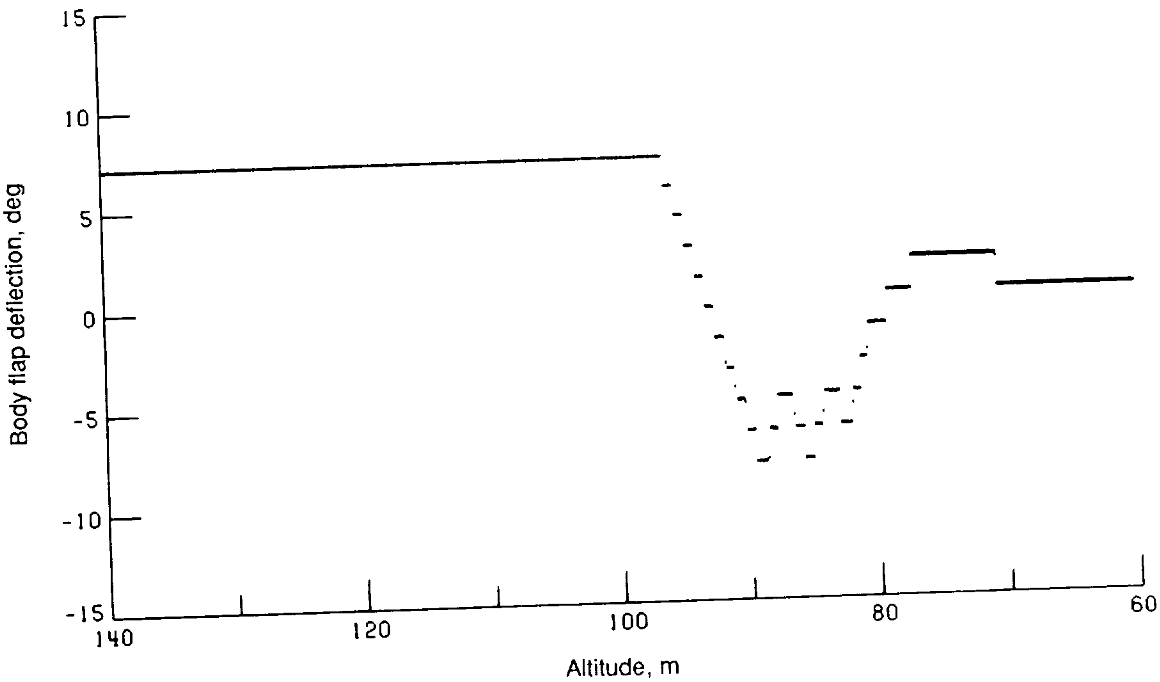


(b) Angle of attack versus altitude.

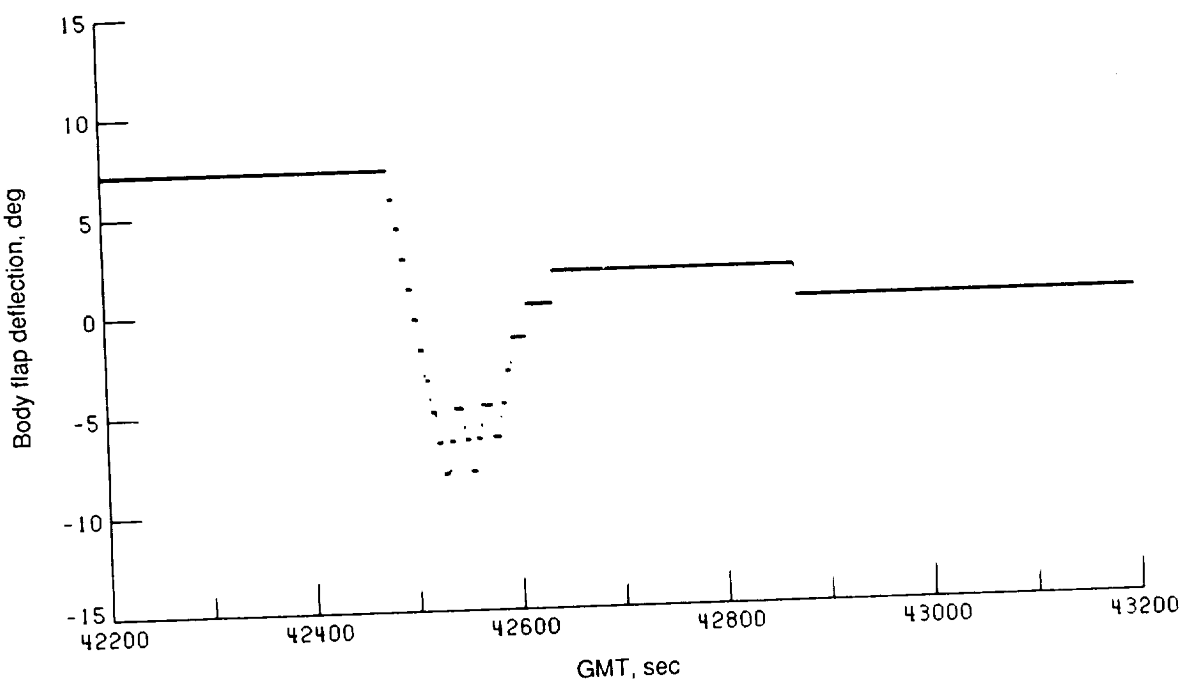


(c) Angle of attack versus time.

Figure 17. Continued.

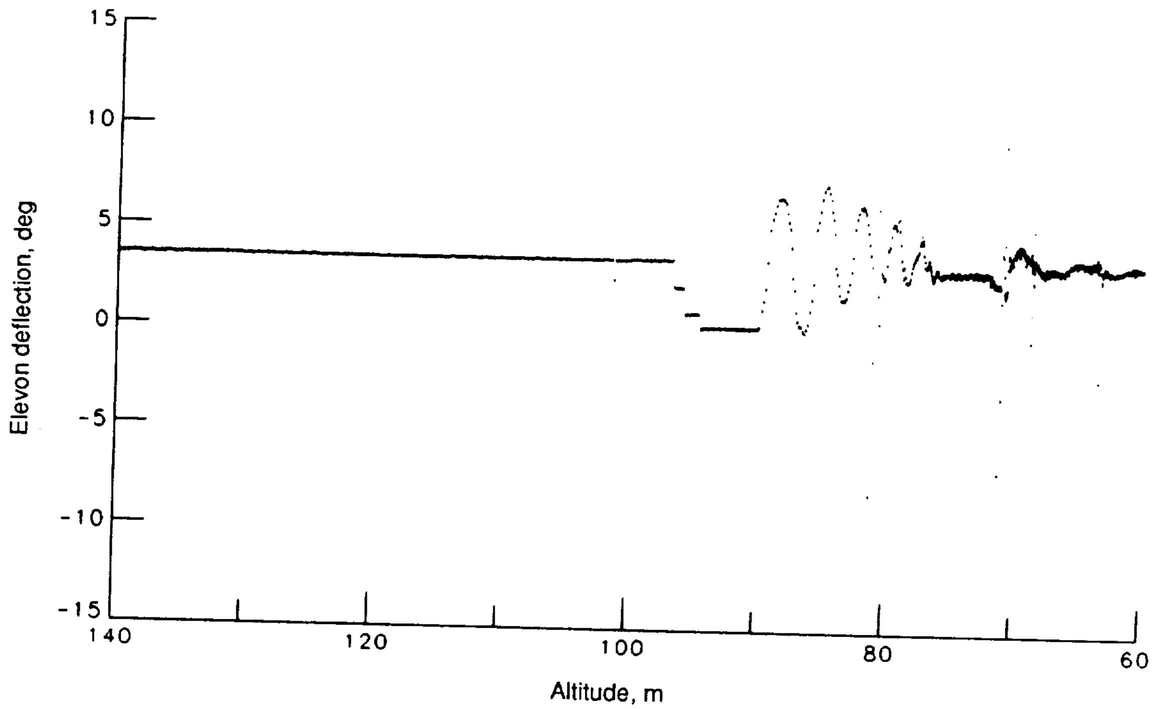


(d) Body flap deflection versus altitude.

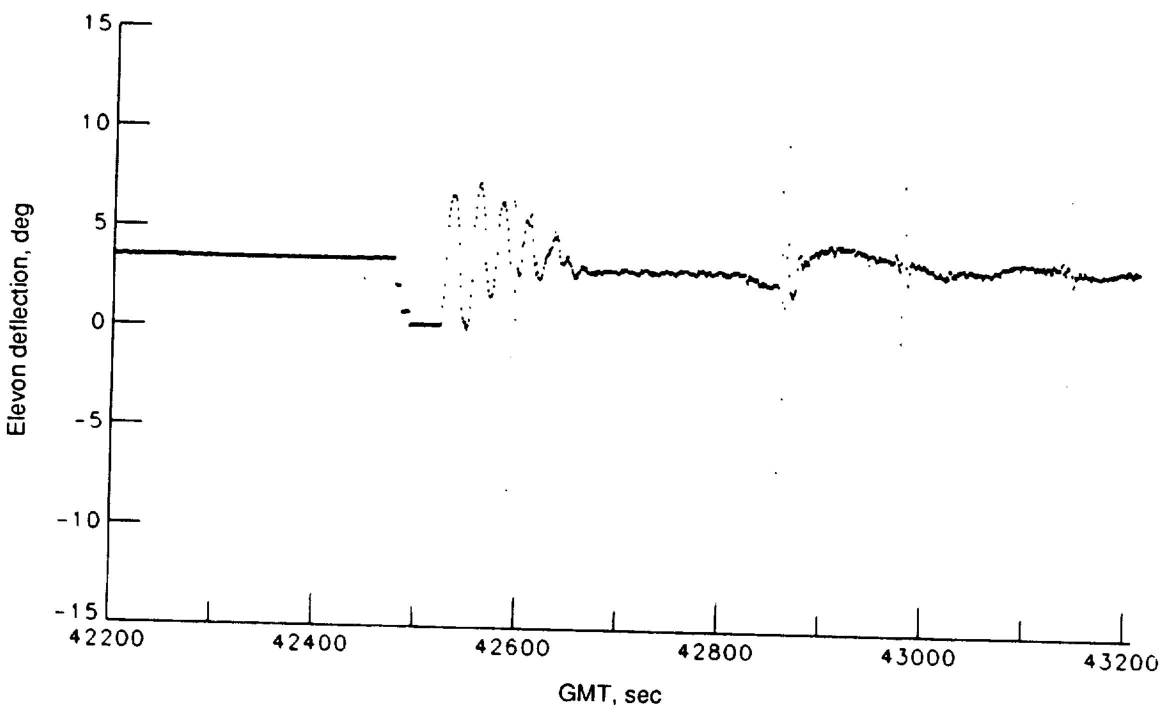


(e) Body flap deflection versus time.

Figure 17. Continued.

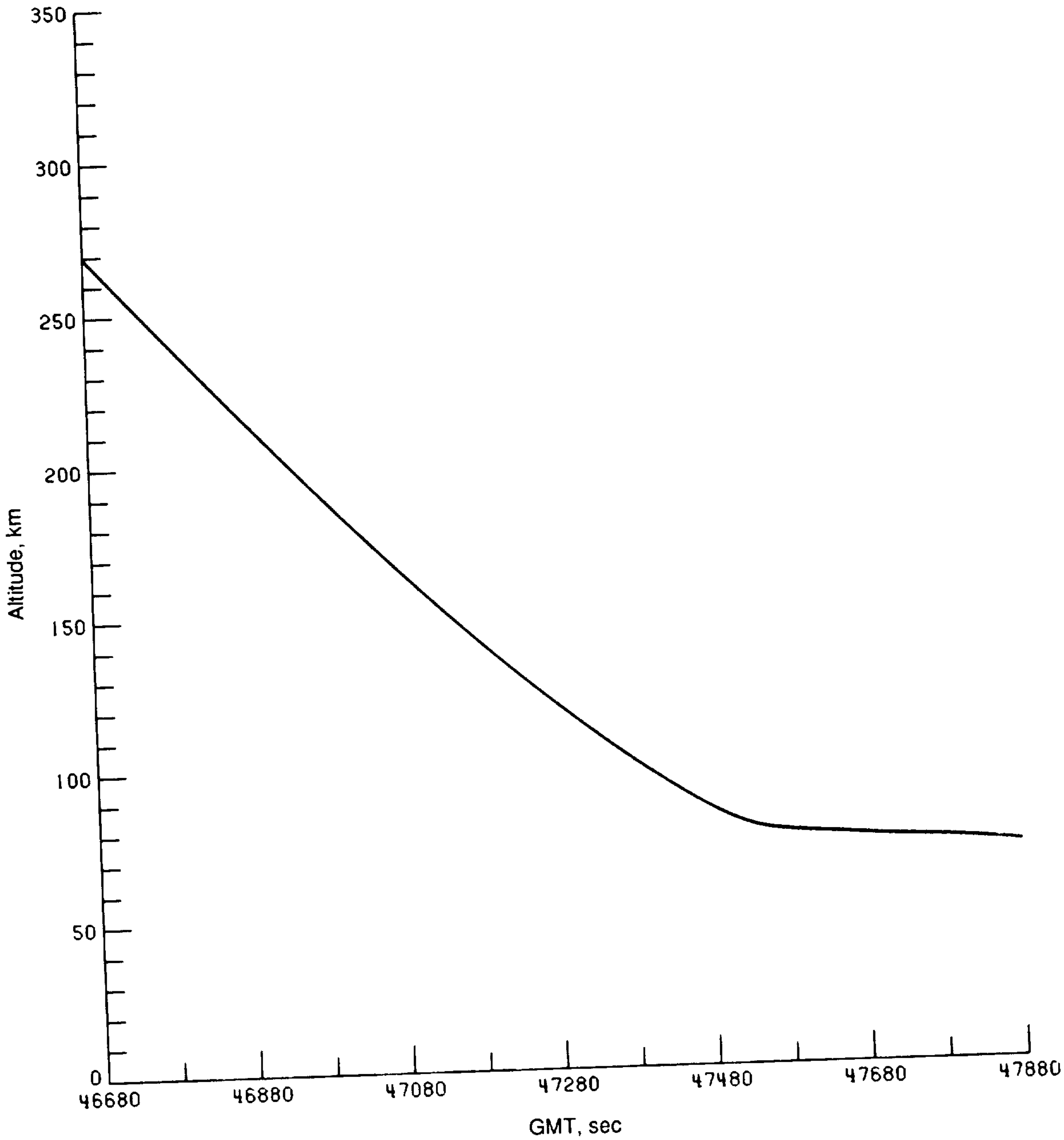


(f) Elevon deflection versus altitude.



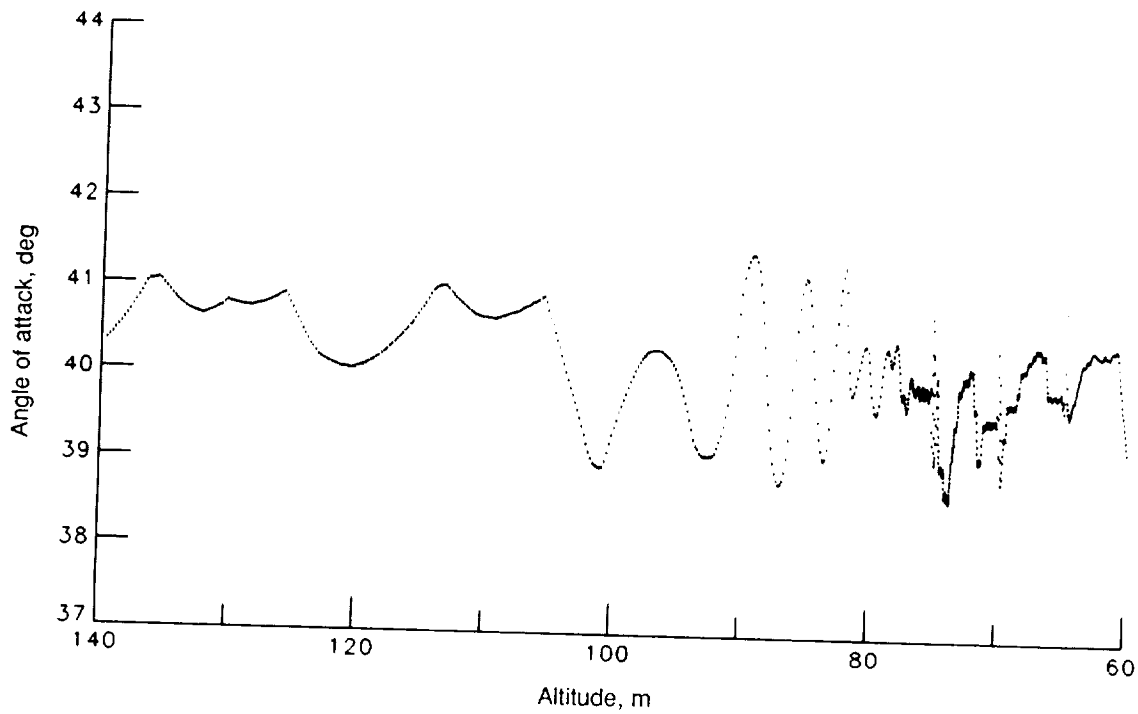
(g) Elevon deflection versus time.

Figure 17. Concluded.

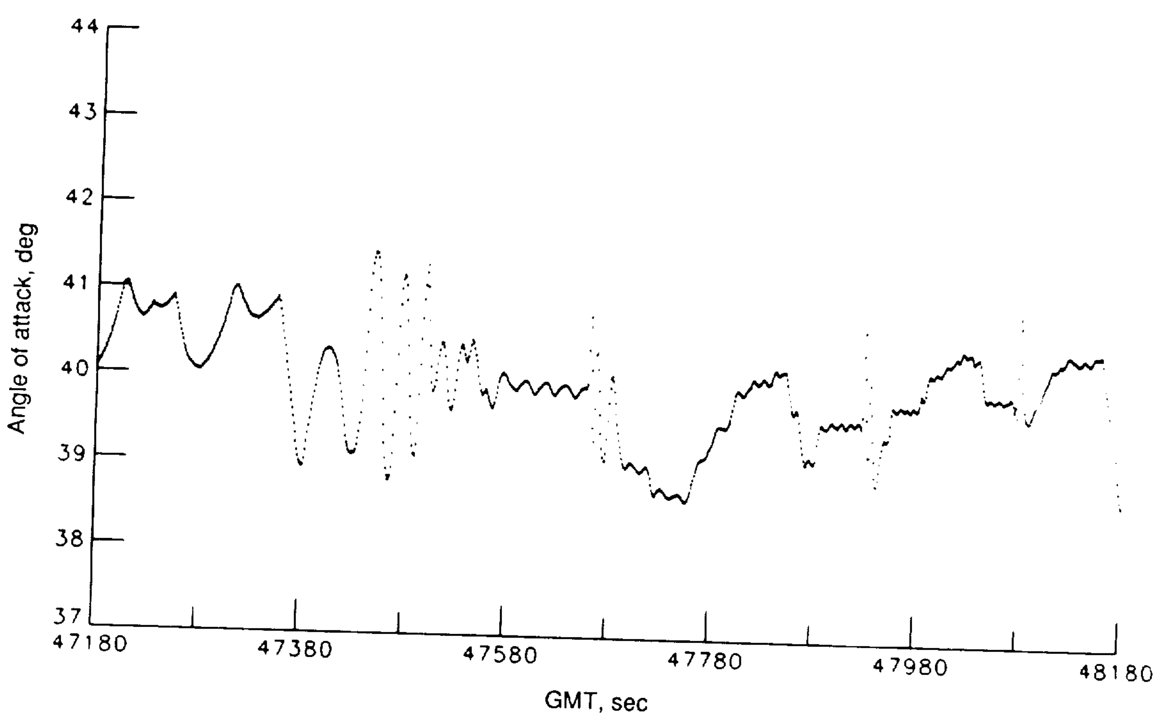


(a) Altitude versus time.

Figure 18. Time and altitude histories of orbiter state vector data subset for STS-41C.

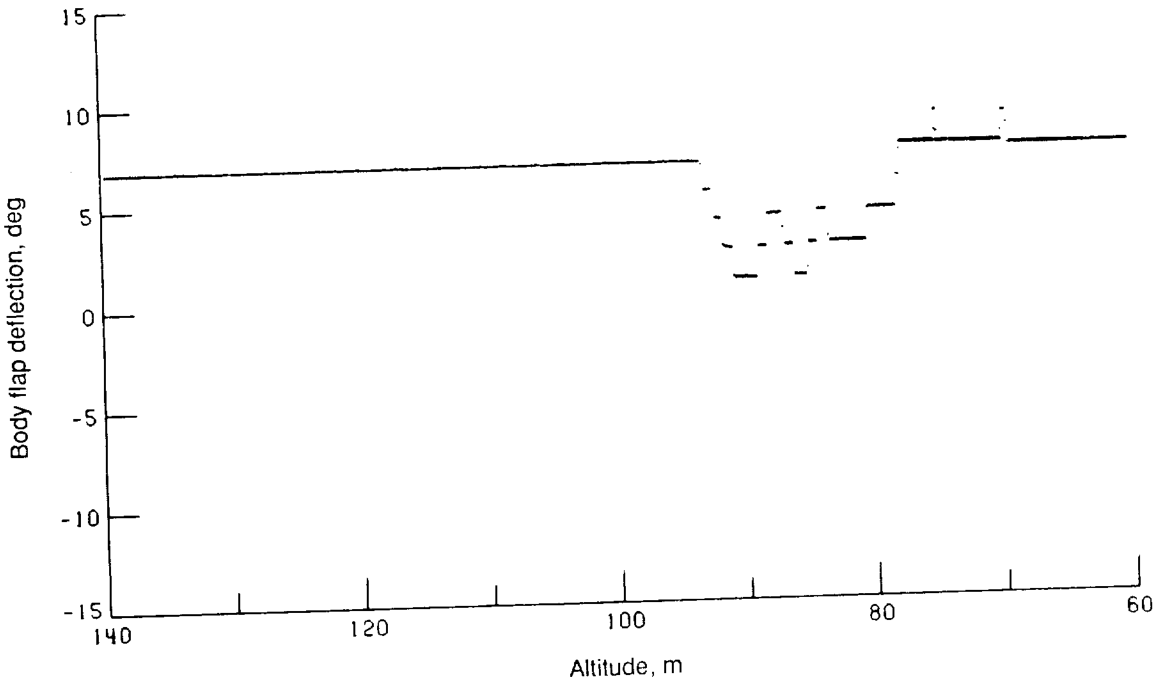


(b) Angle of attack versus altitude.

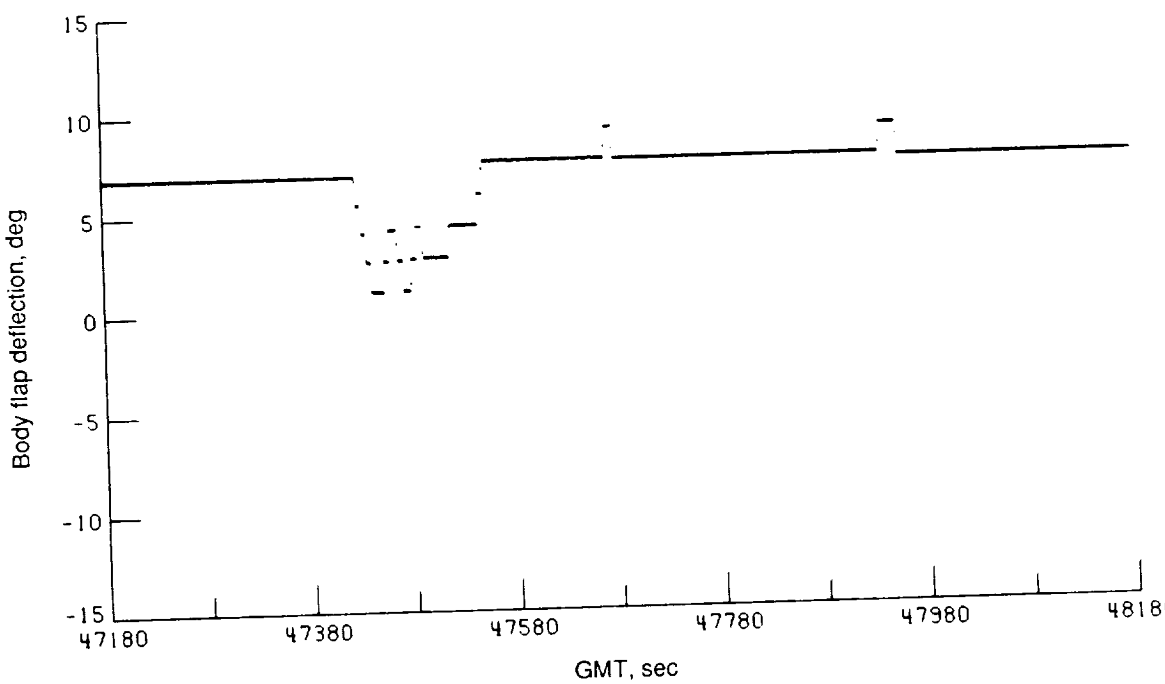


(c) Angle of attack versus time.

Figure 18. Continued.

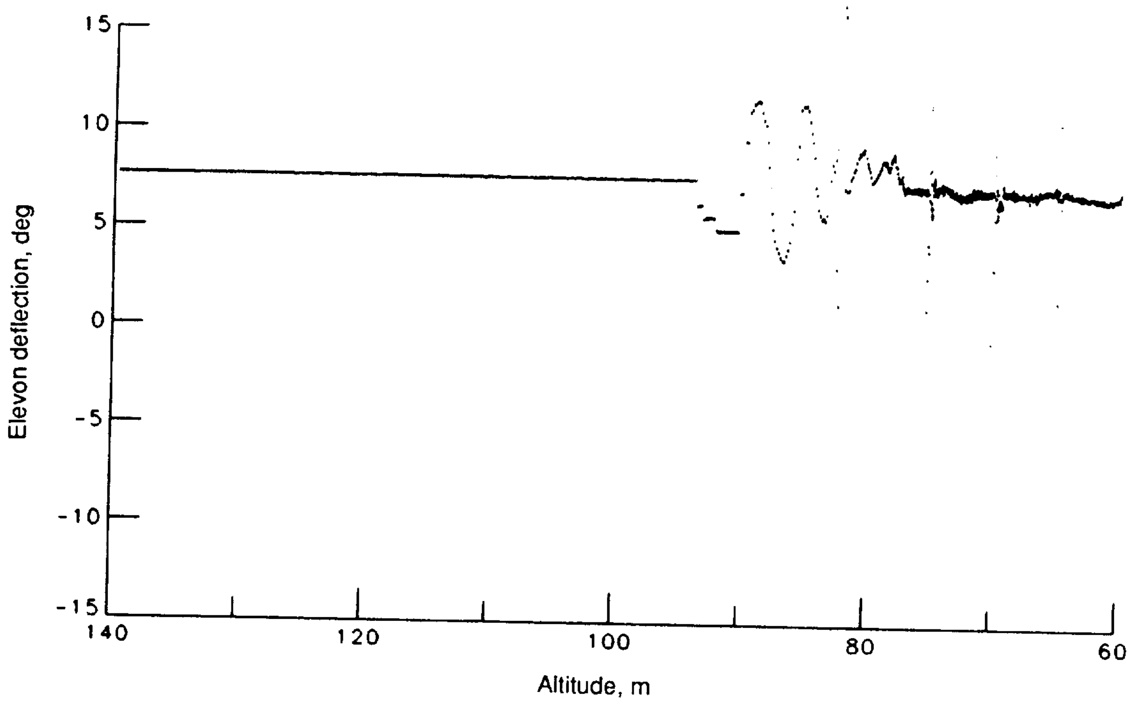


(d) Body flap deflection versus altitude.

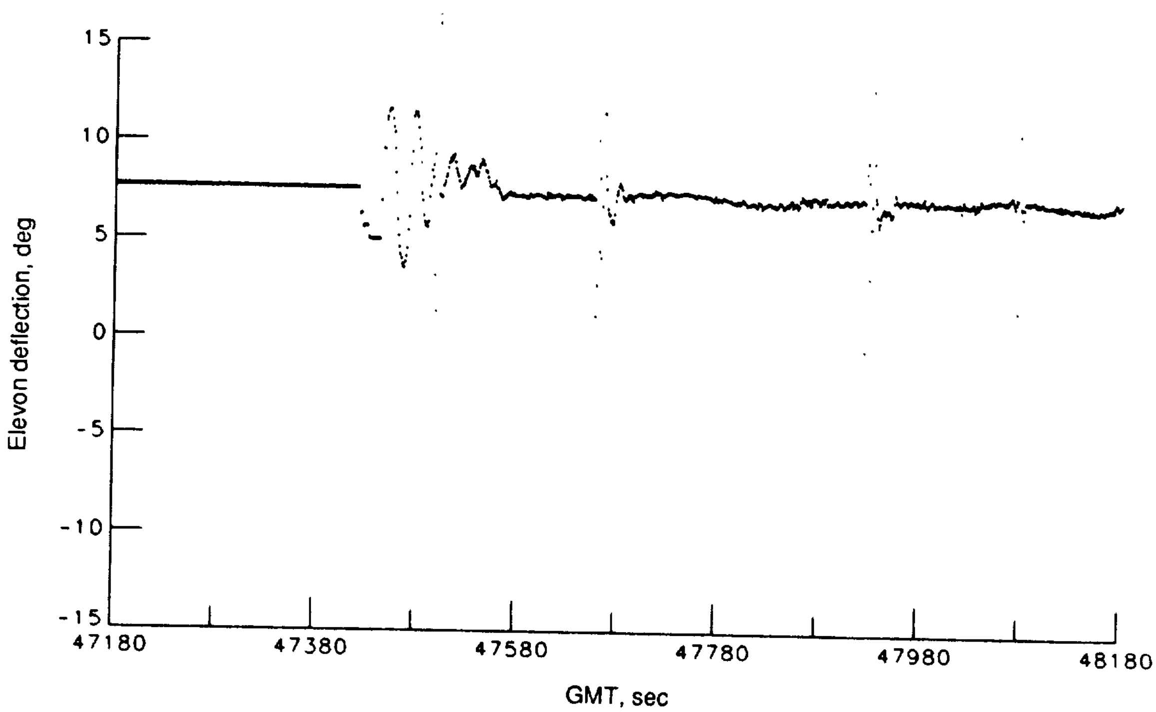


(e) Body flap deflection versus time.

Figure 18. Continued.

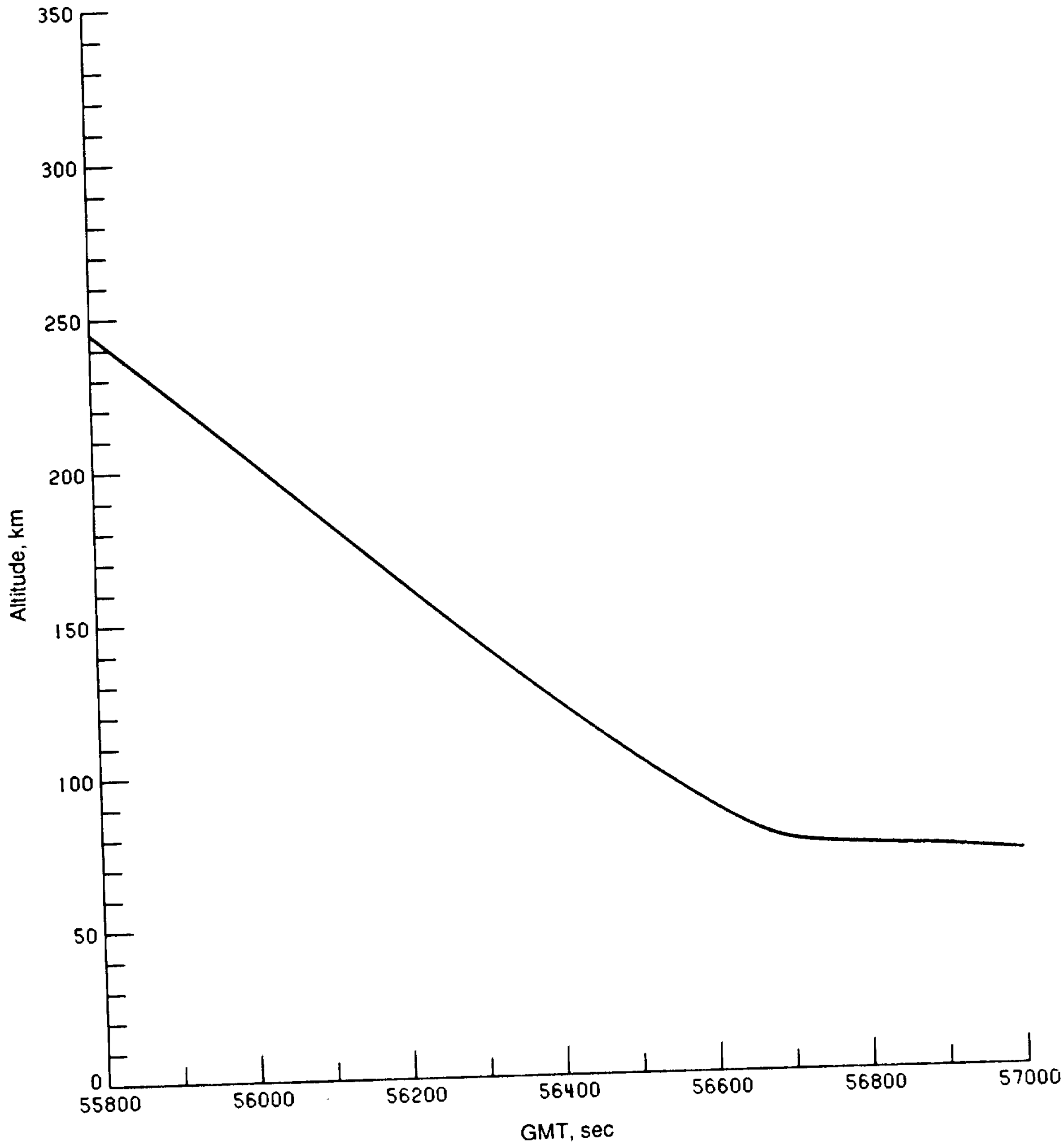


(f) Elevon deflection versus altitude.



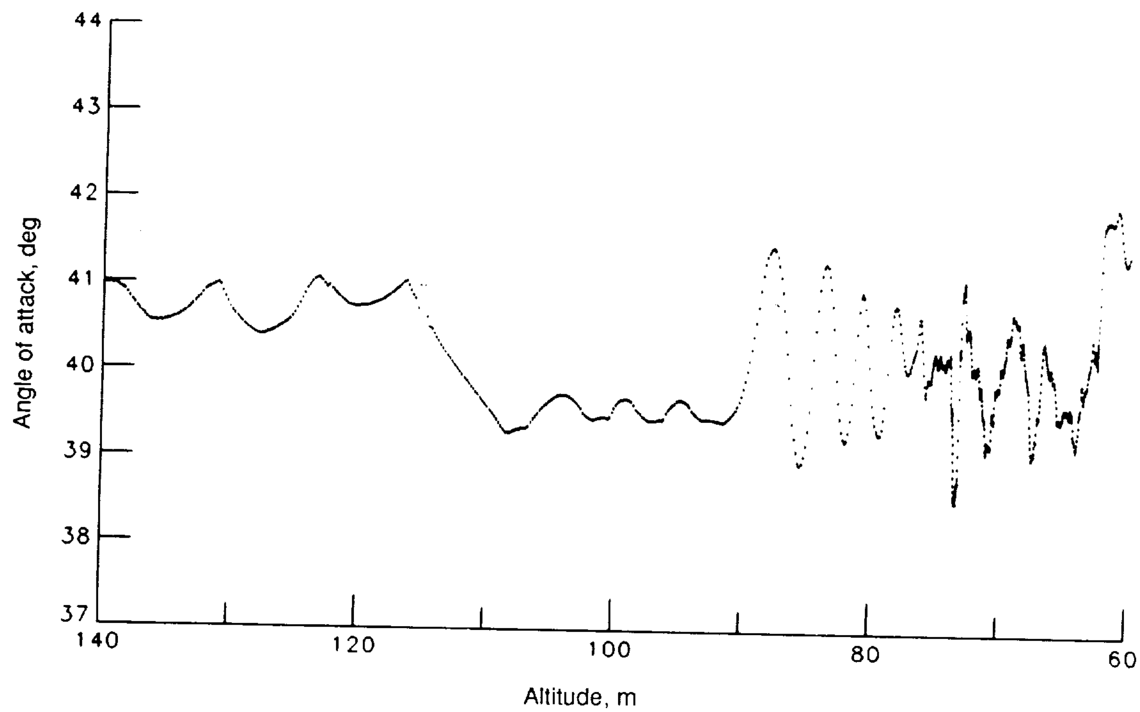
(g) Elevon deflection versus time.

Figure 18. Concluded.

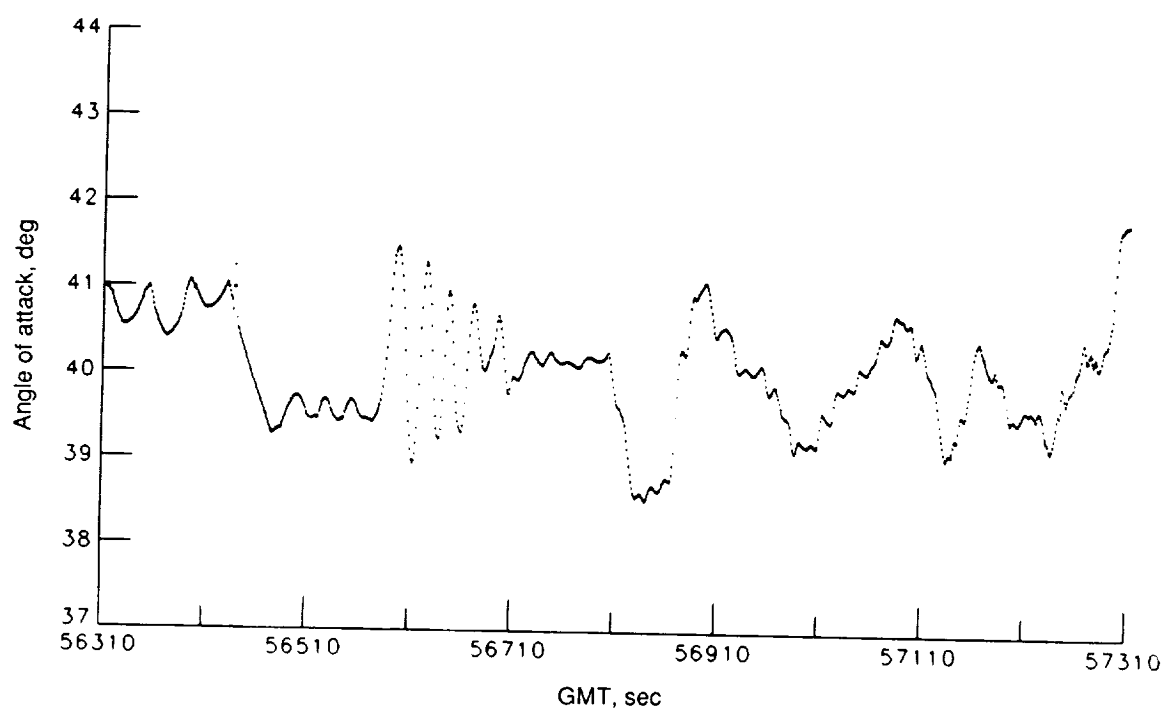


(a) Altitude versus time.

Figure 19. Time and altitude histories of orbiter state vector data subset for STS-51B.

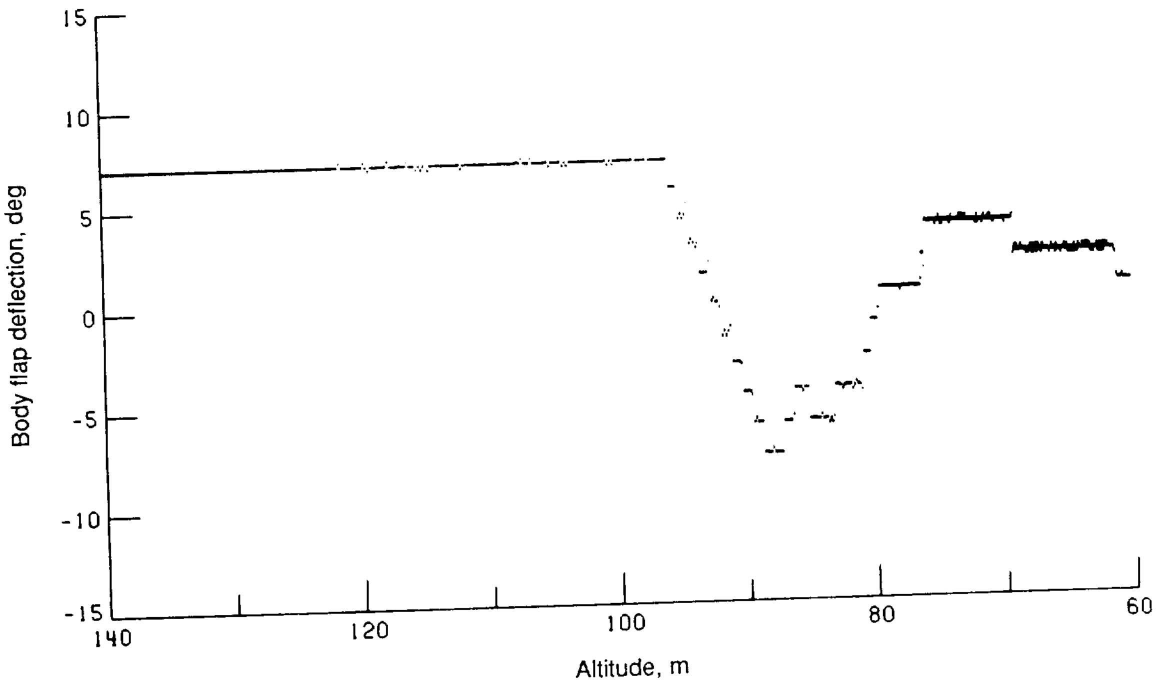


(b) Angle of attack versus altitude.

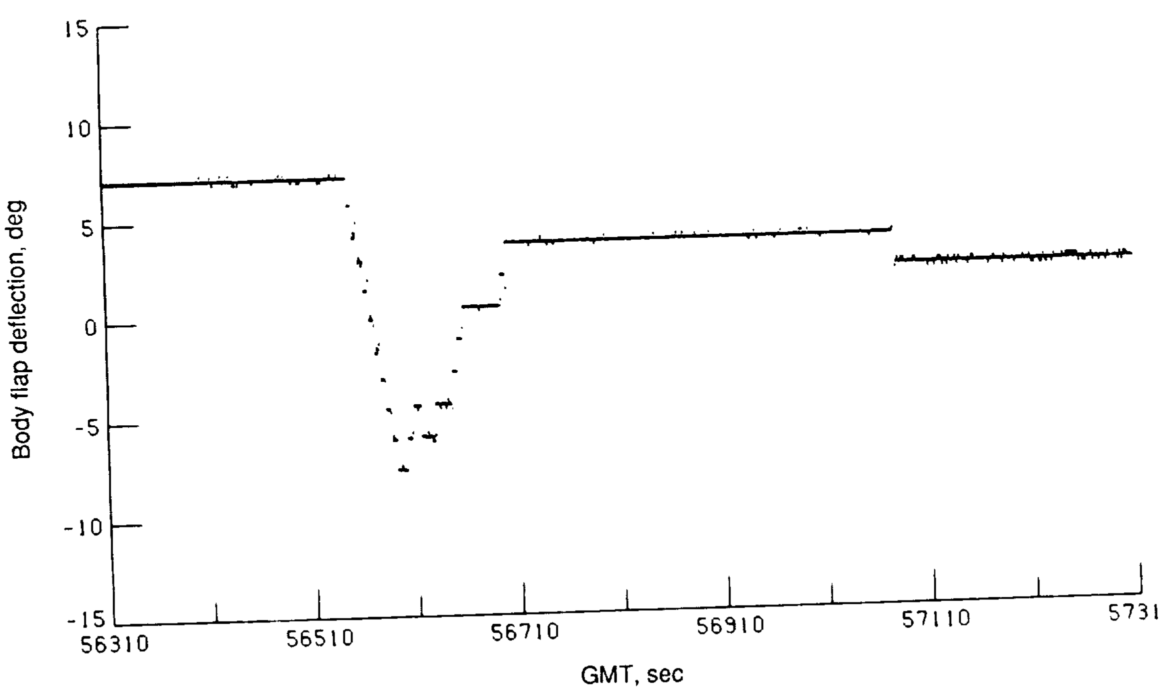


(c) Angle of attack versus time.

Figure 19. Continued.

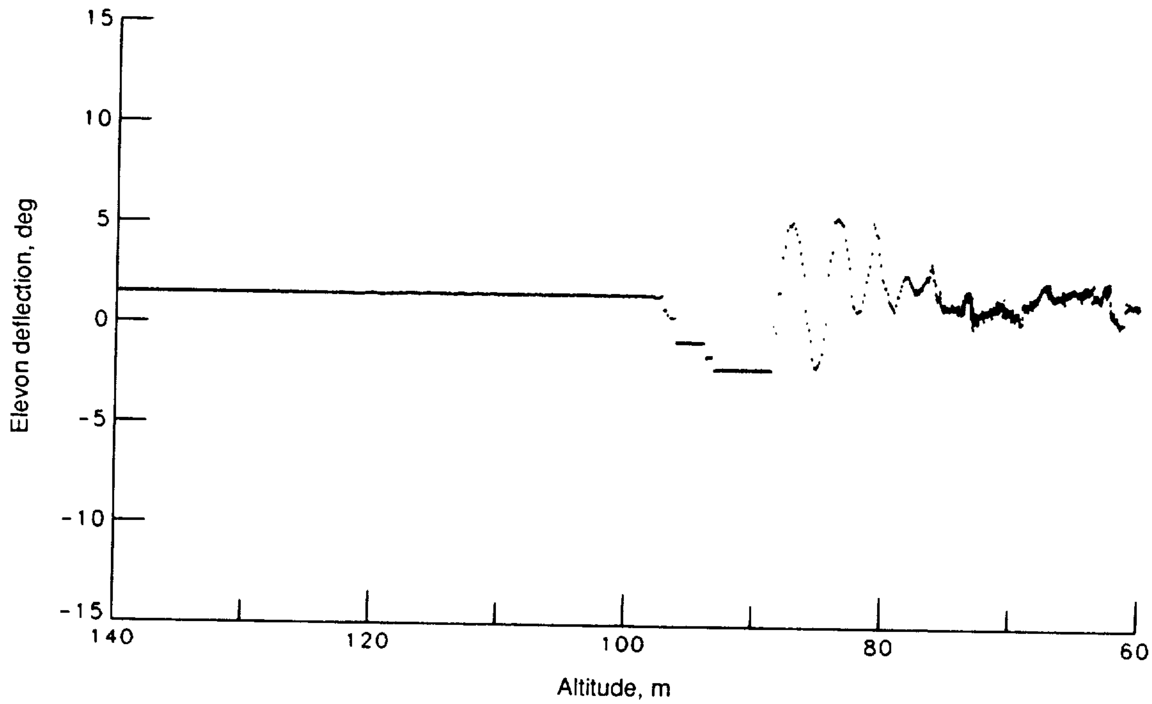


(d) Body flap deflection versus altitude.

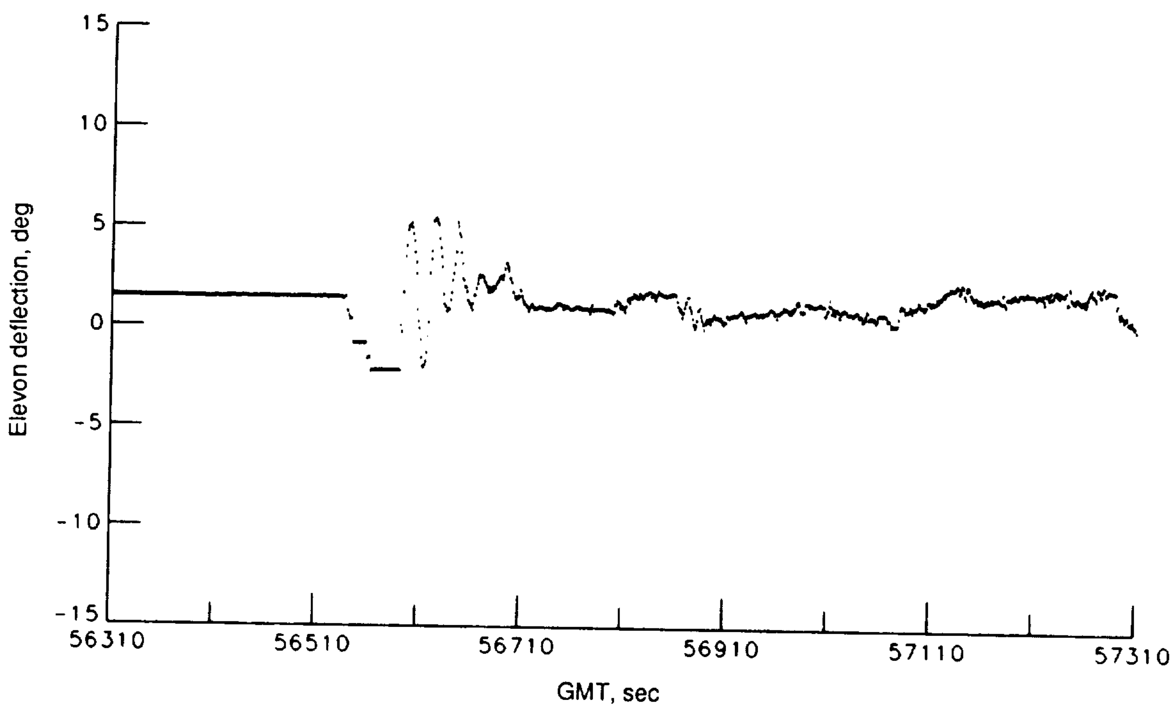


(e) Body flap deflection versus time.

Figure 19. Continued.

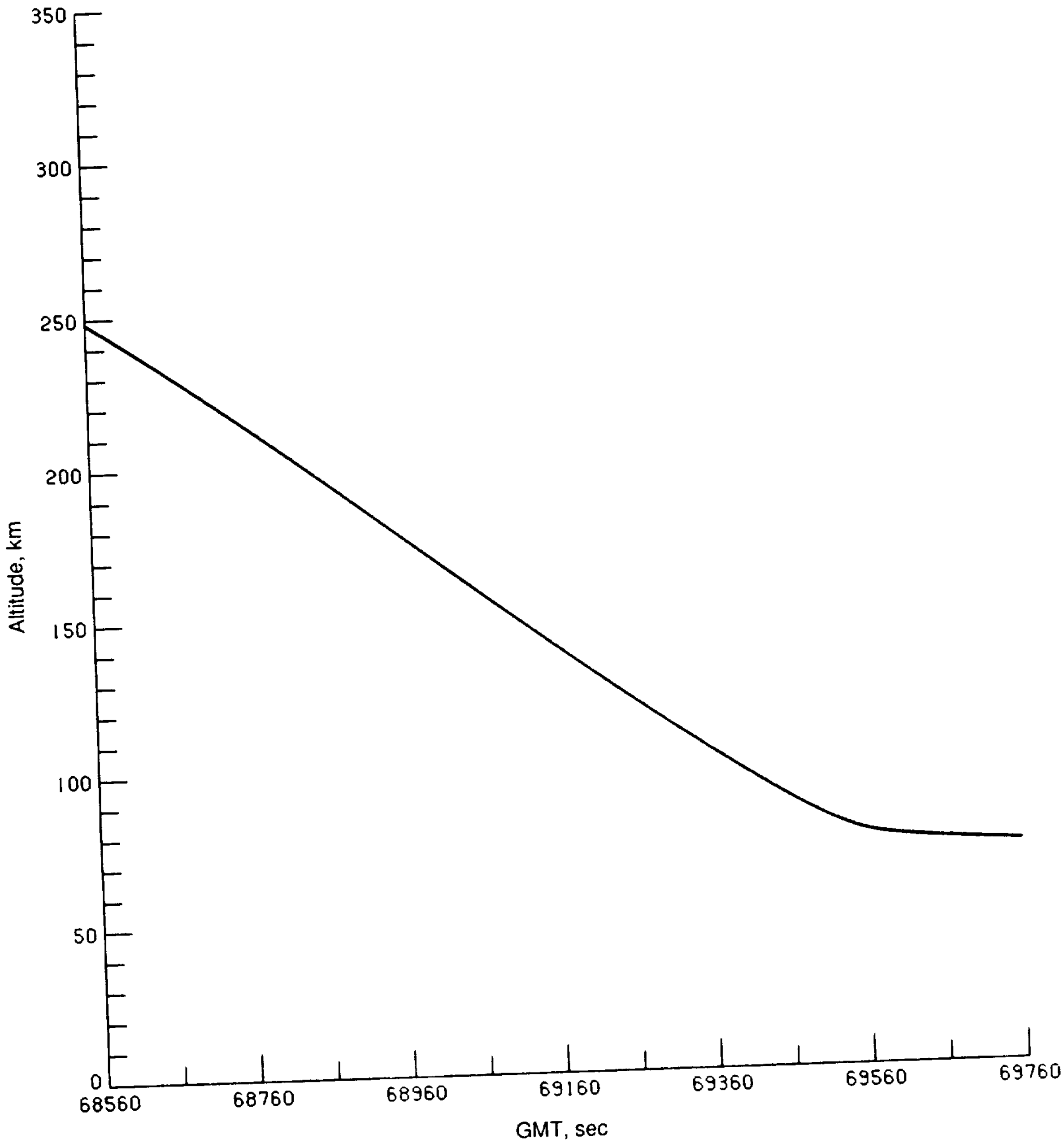


(f) Elevon deflection versus altitude.



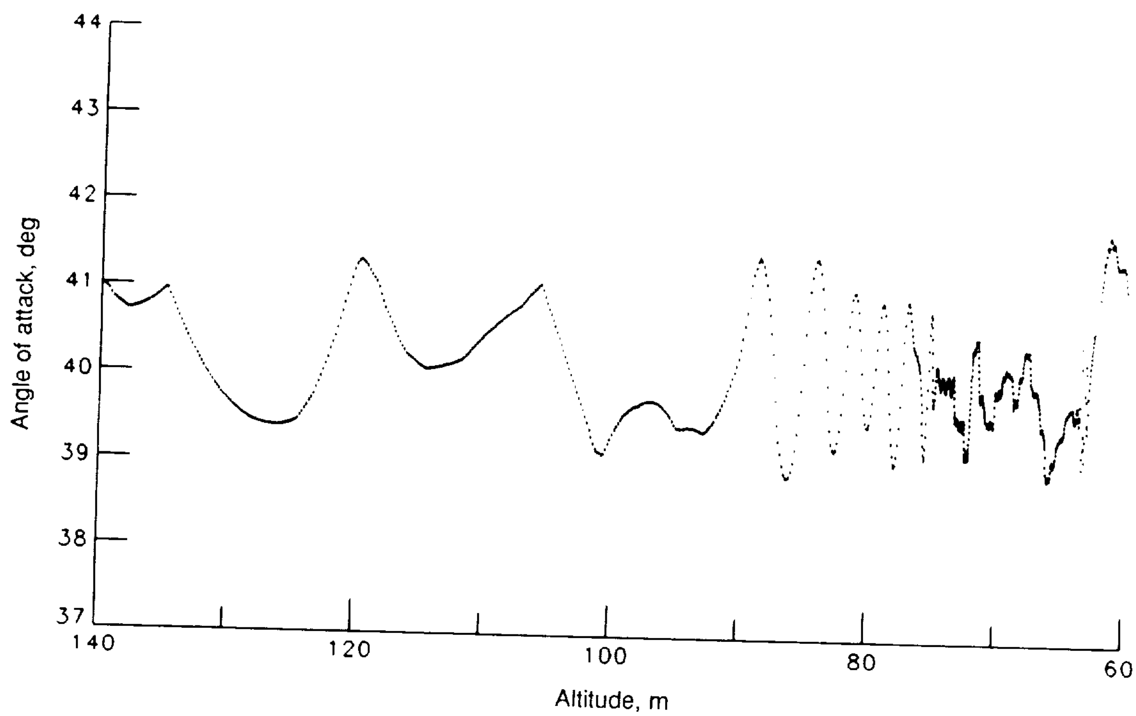
(g) Elevon deflection versus time.

Figure 19. Concluded.

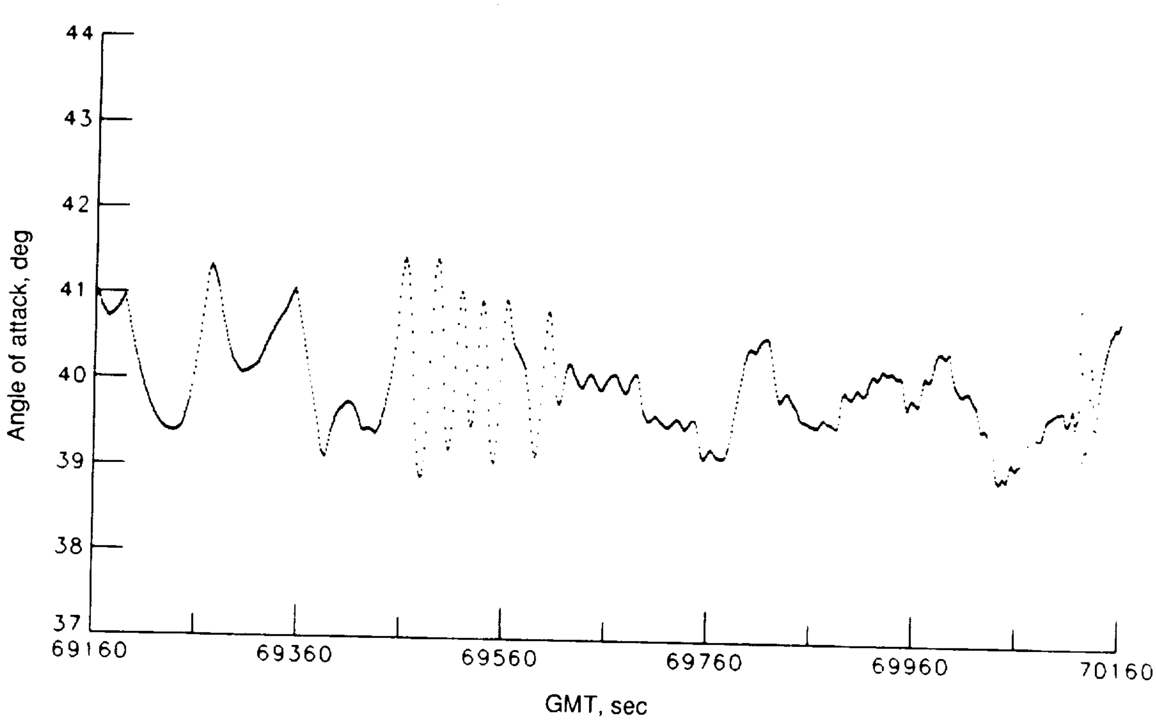


(a) Altitude versus time.

Figure 20. Time and altitude histories of orbiter state vector data subset for STS-51F.

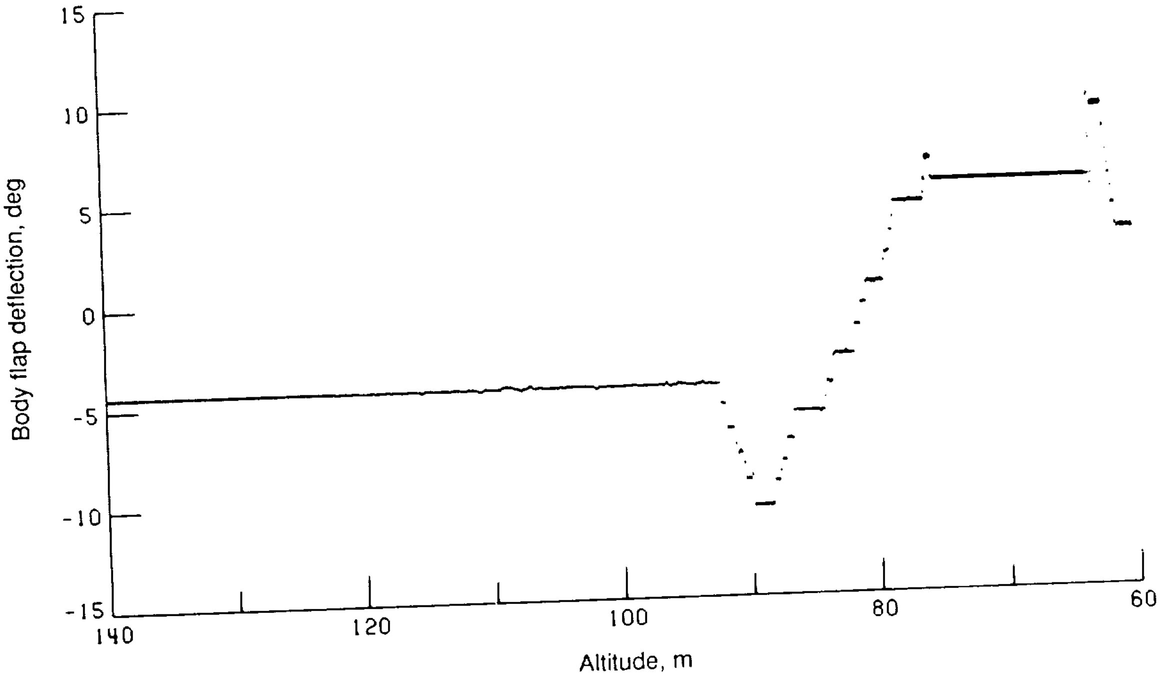


(b) Angle of attack versus altitude.

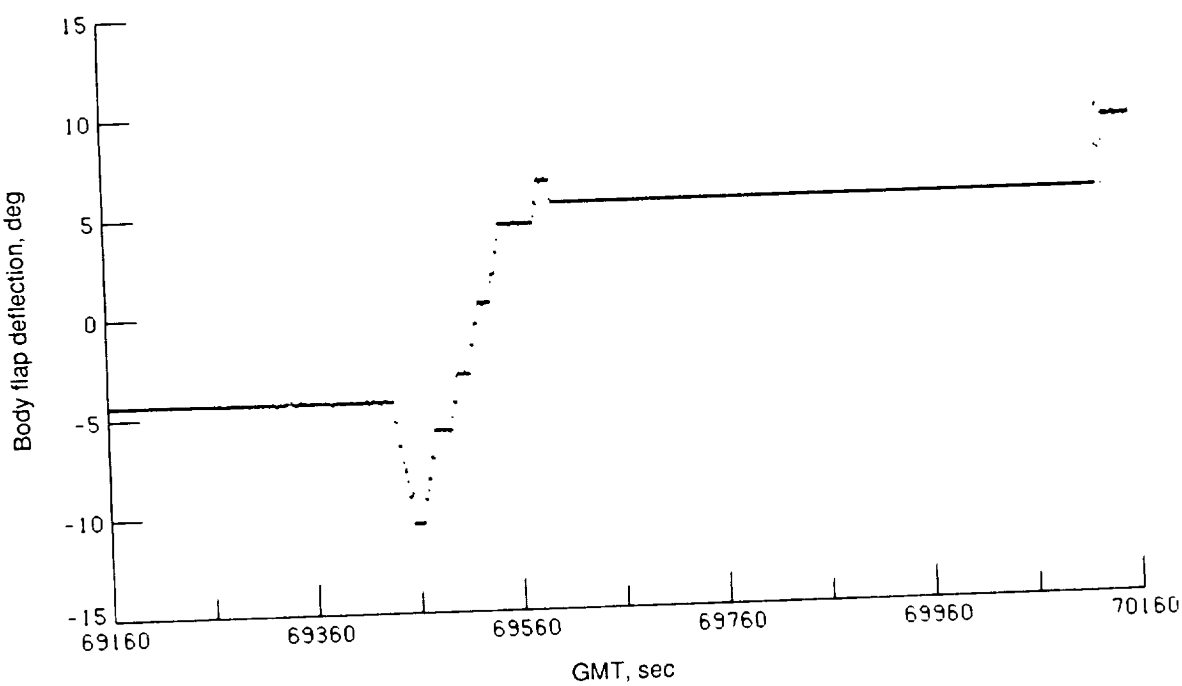


(c) Angle of attack versus time.

Figure 20. Continued.

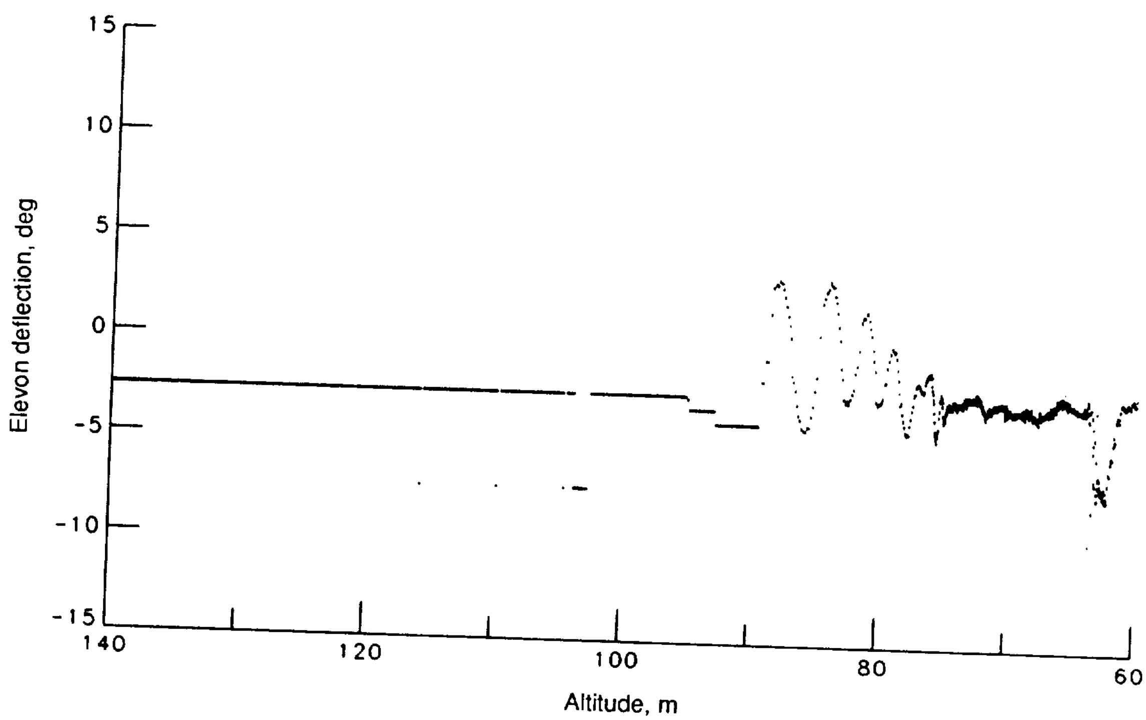


(d) Body flap deflection versus altitude.

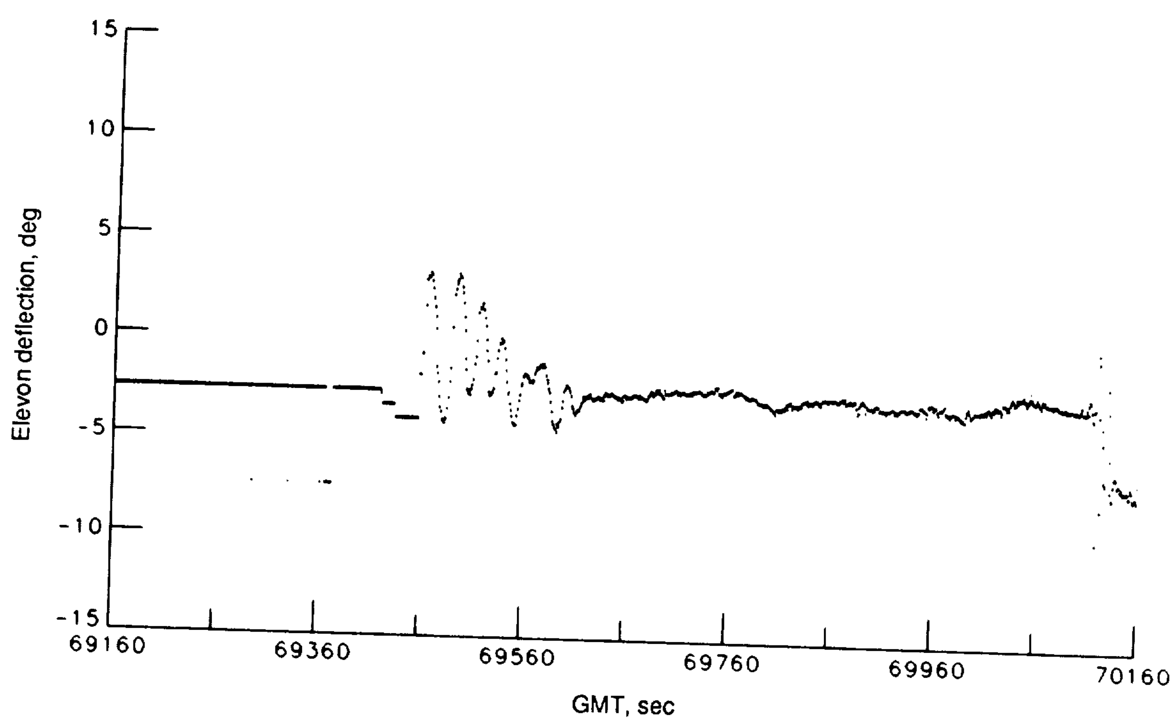


(e) Body flap deflection versus time.

Figure 20. Continued.

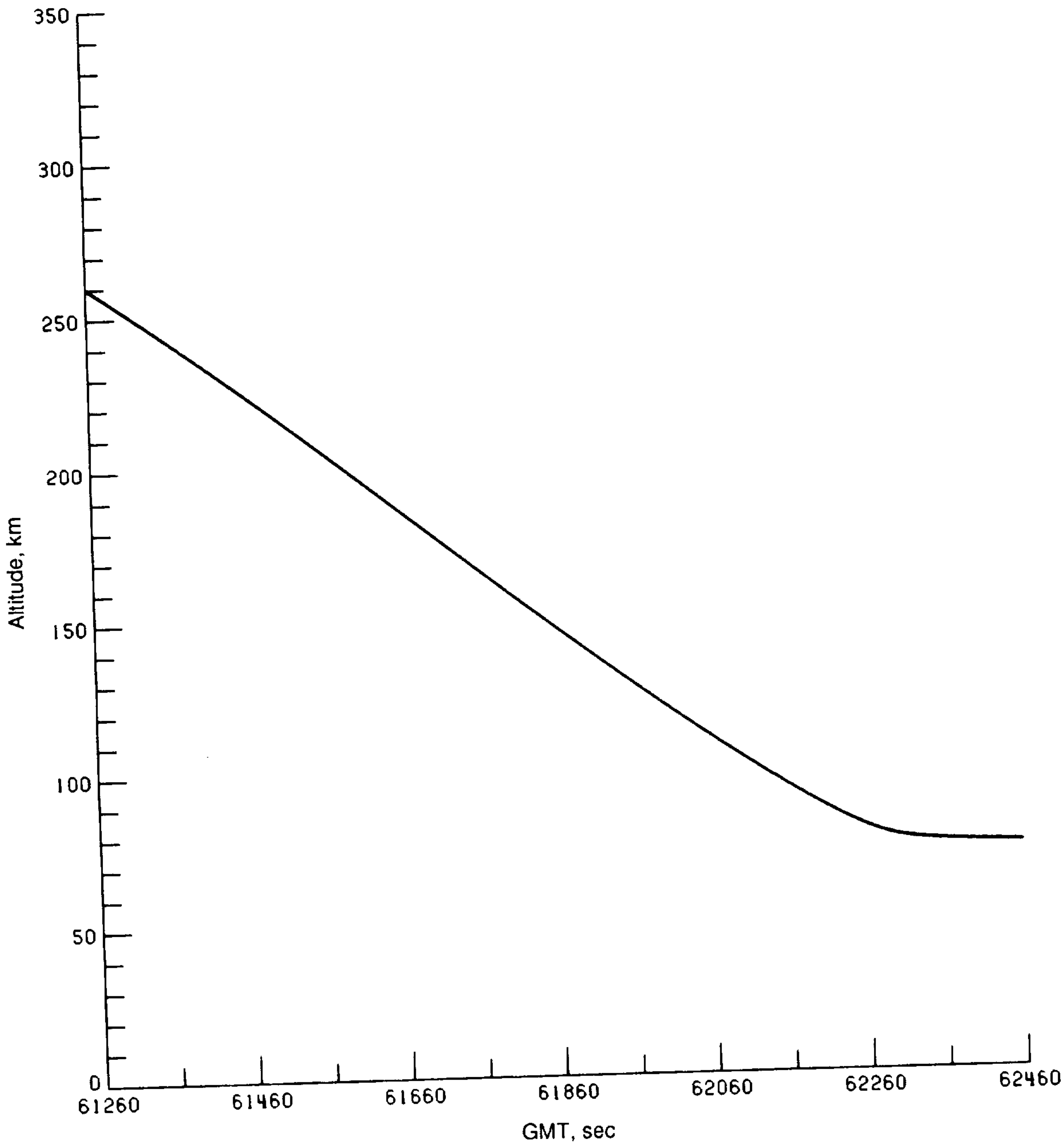


(f) Elevon deflection versus altitude.



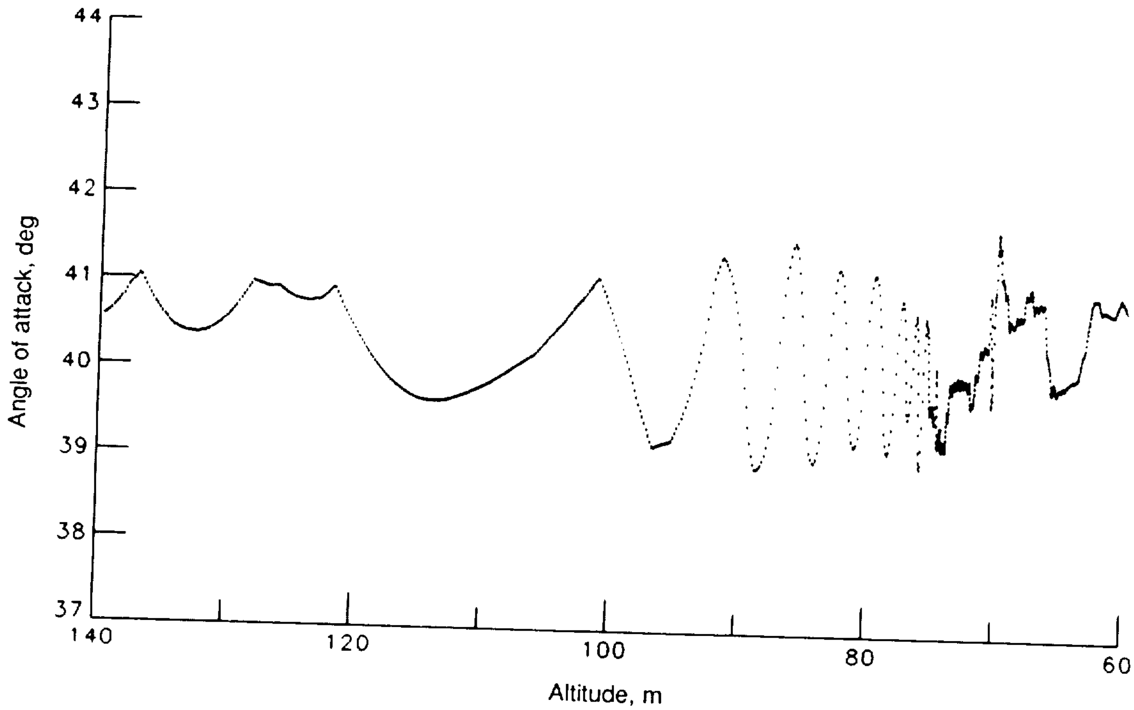
(g) Elevon deflection versus time.

Figure 20. Concluded.

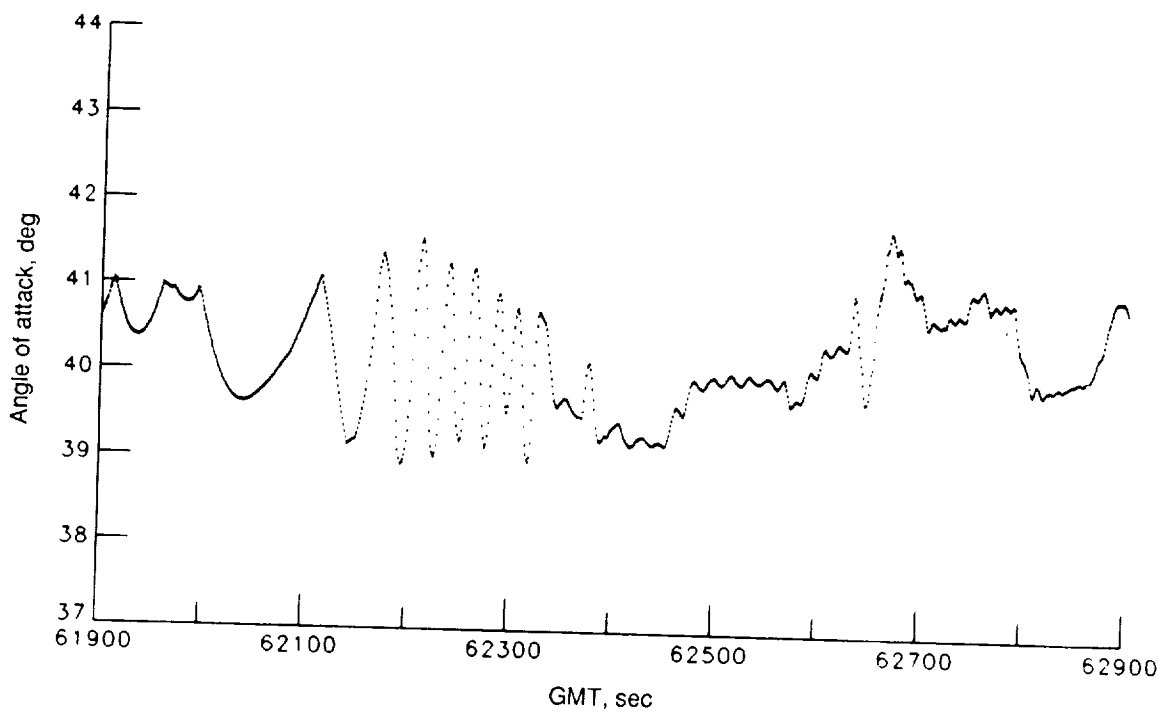


(a) Altitude versus time.

Figure 21. Time and altitude histories of orbiter state vector data subset for STS-61A.

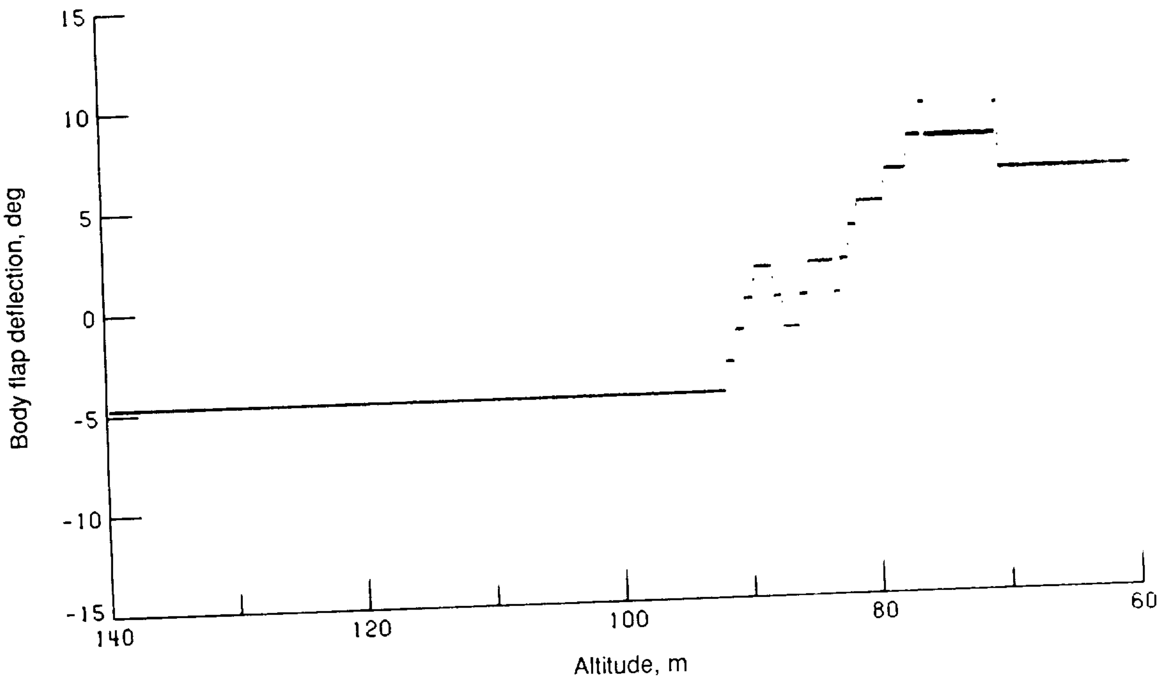


(b) Angle of attack versus altitude.

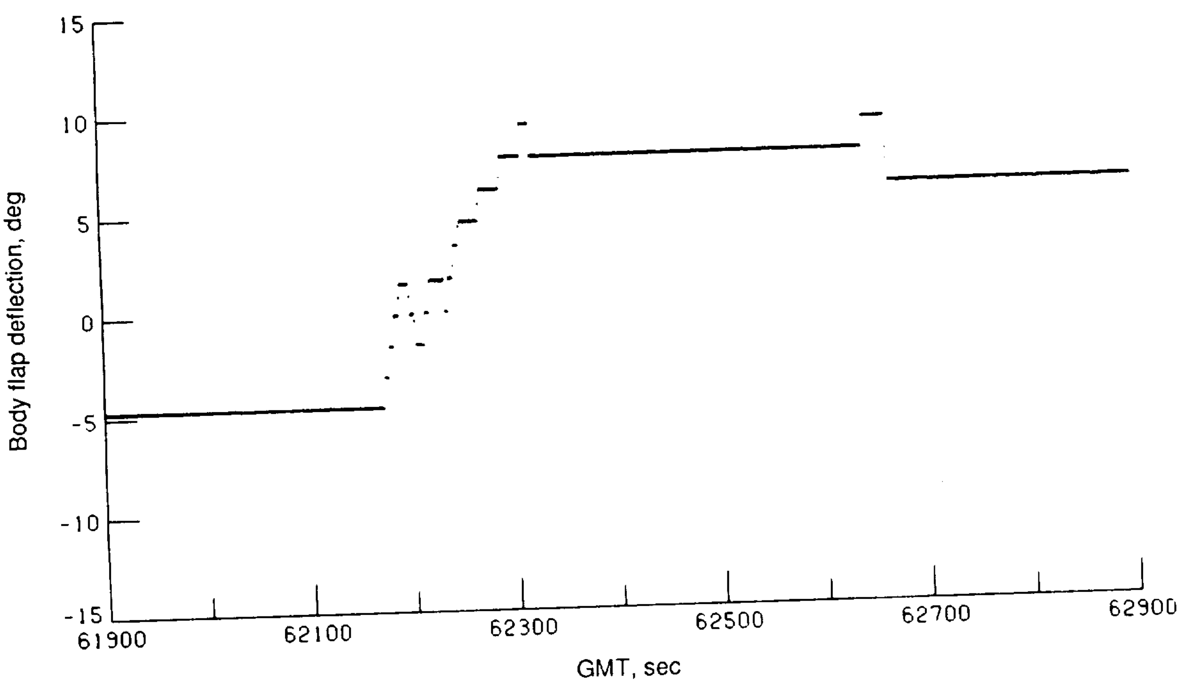


(c) Angle of attack versus time.

Figure 21. Continued.

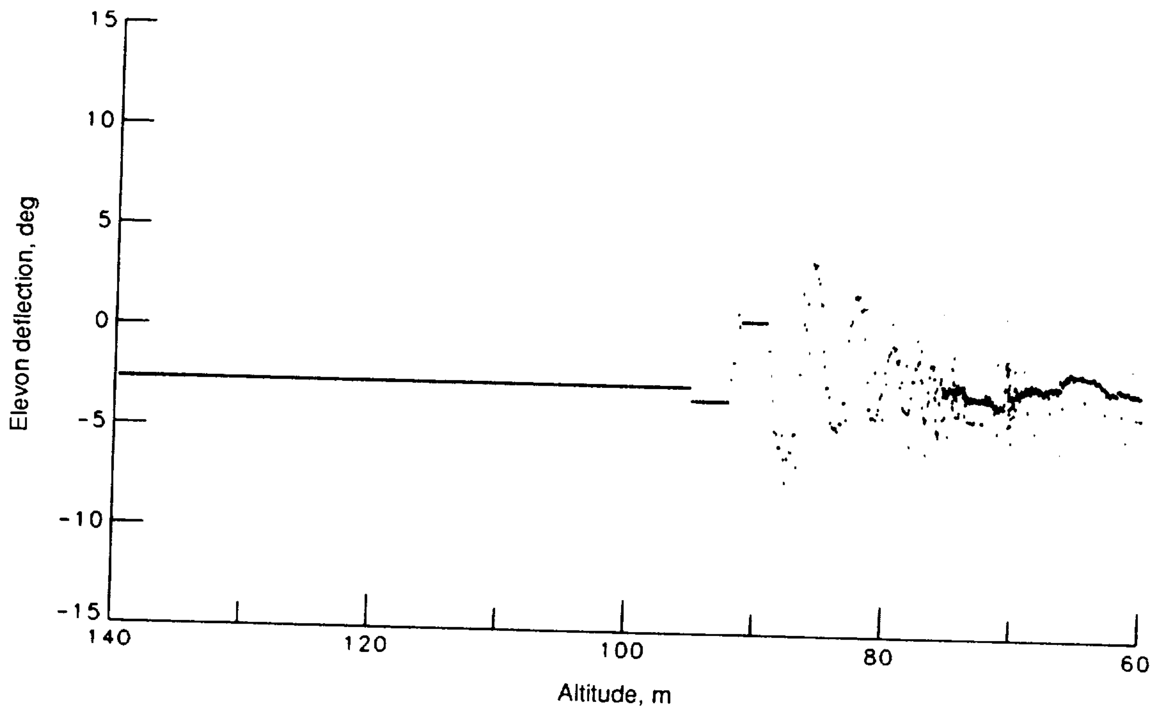


(d) Body flap deflection versus altitude.

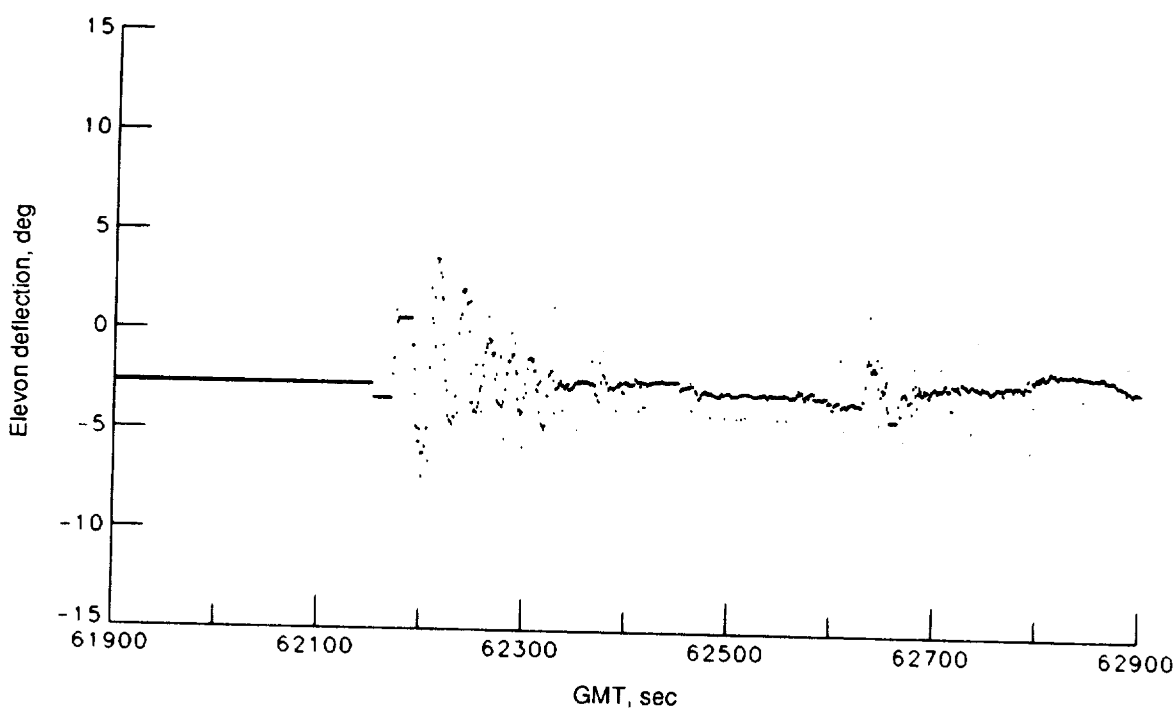


(e) Body flap deflection versus time.

Figure 21. Concluded.

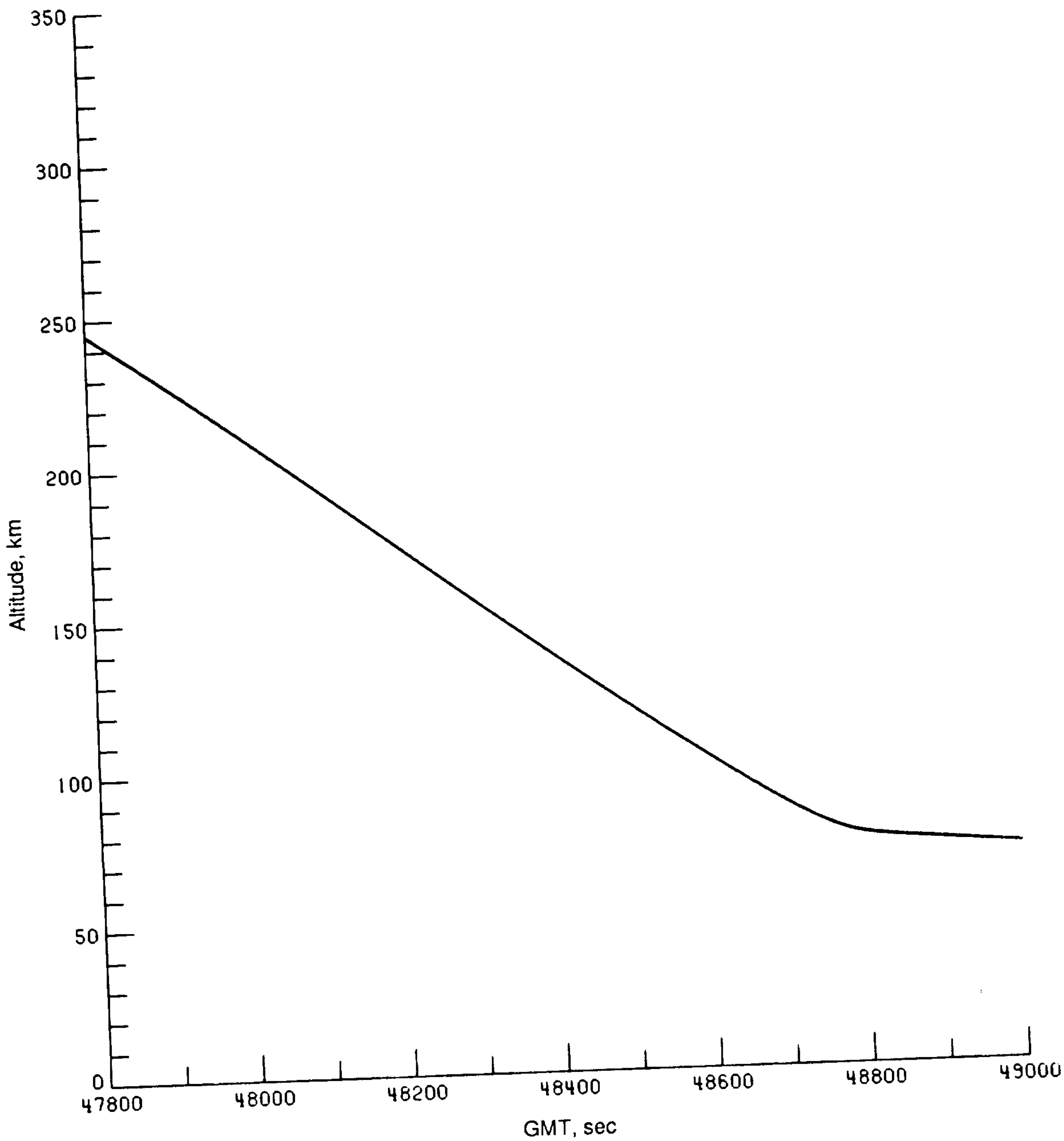


(f) Elevon deflection versus altitude.



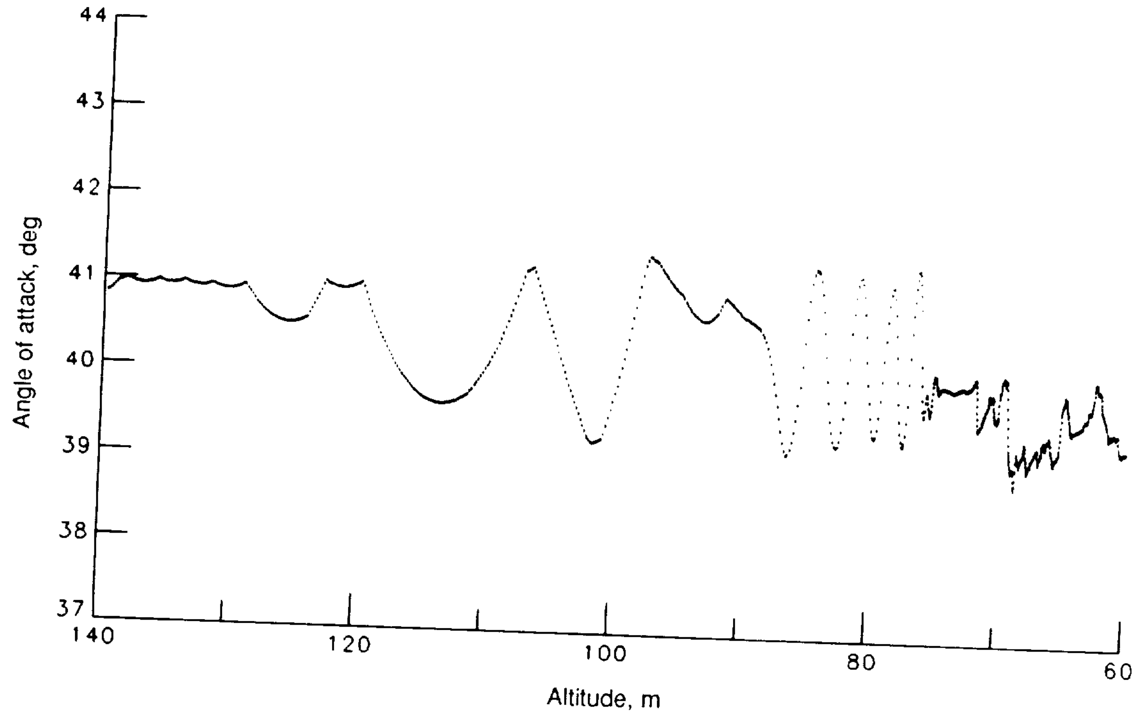
(g) Elevon deflection versus time.

Figure 21. Concluded.

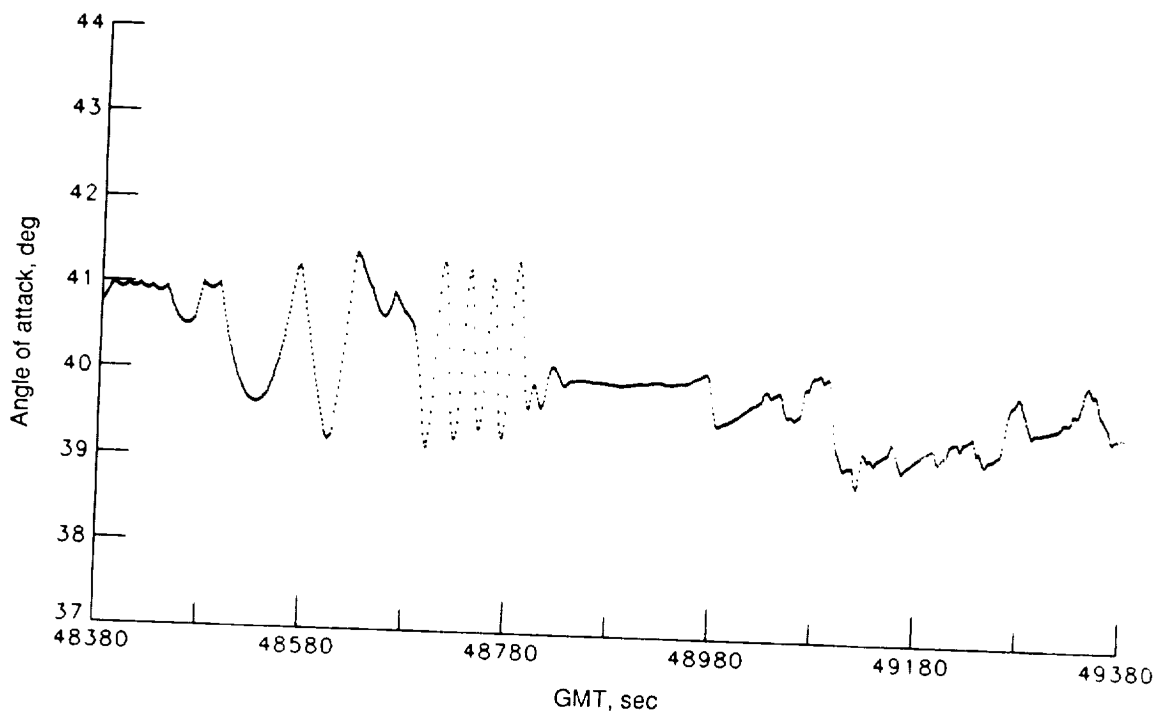


(a) Altitude versus time.

Figure 22. Time and altitude histories of orbiter state vector data subset for STS-61C.

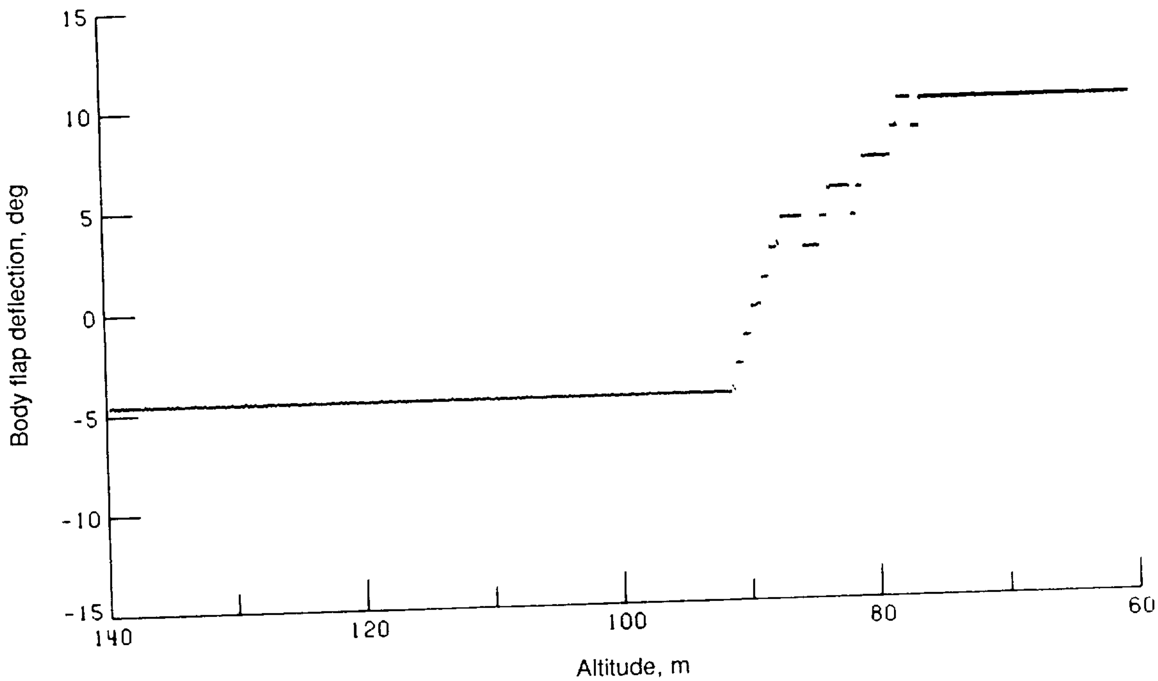


(b) Angle of attack versus altitude.

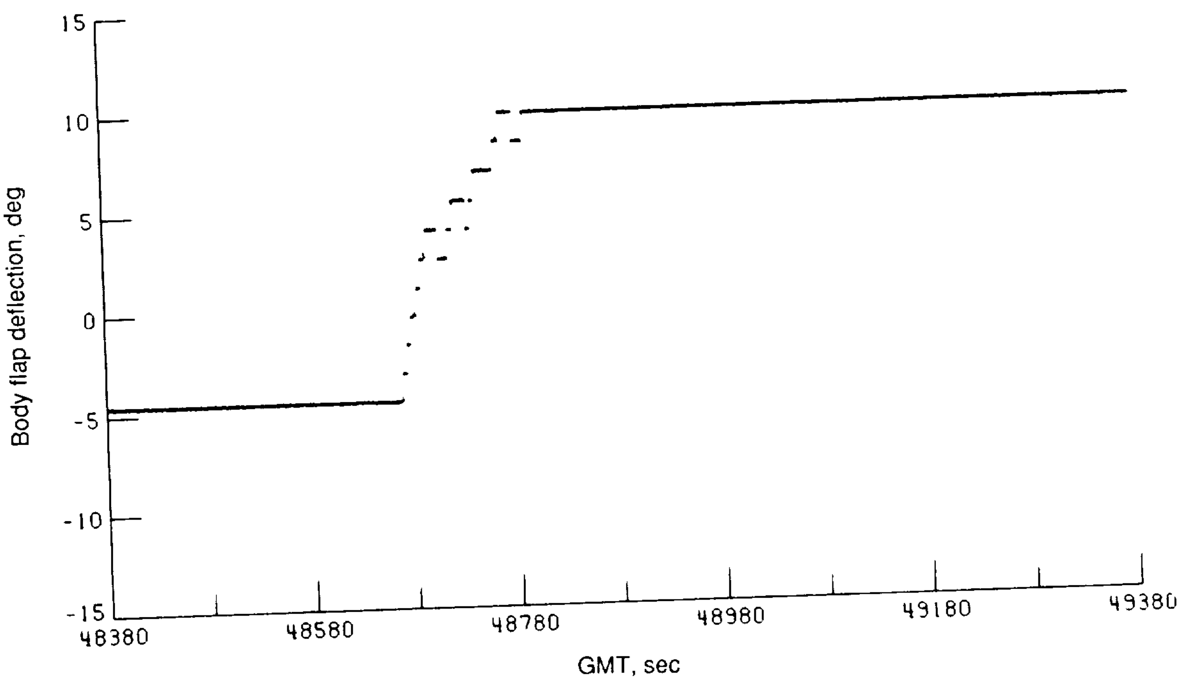


(c) Angle of attack versus time.

Figure 22. Continued.

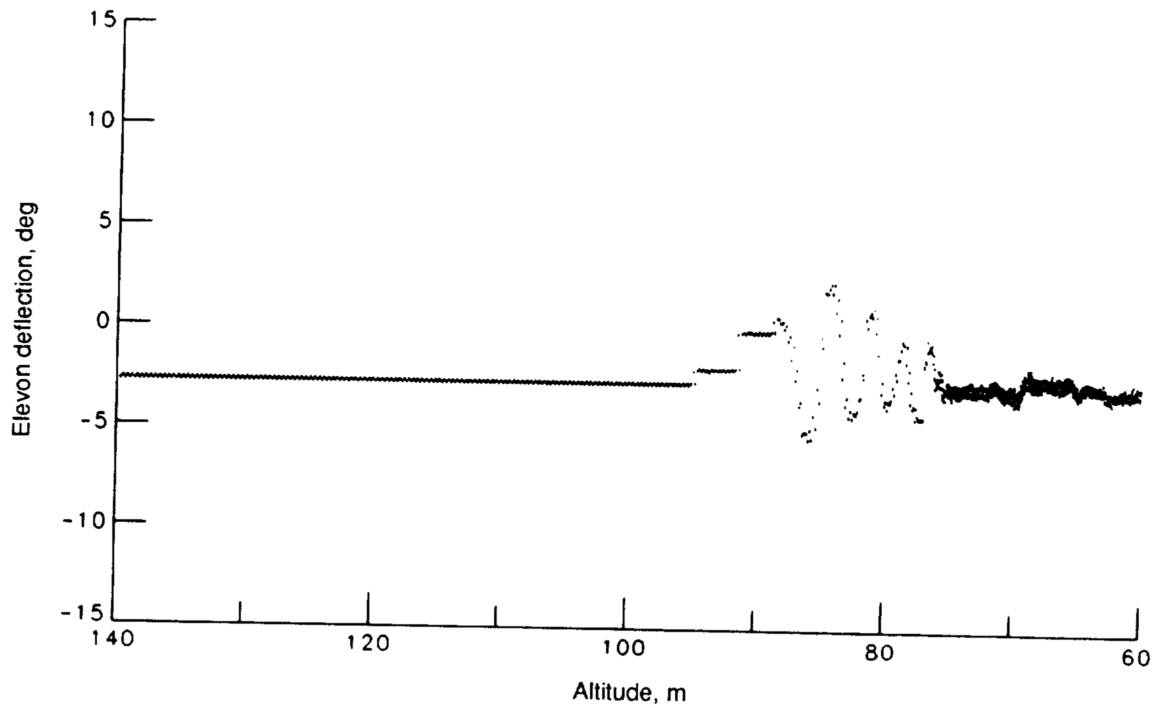


(d) Body flap deflection versus altitude.

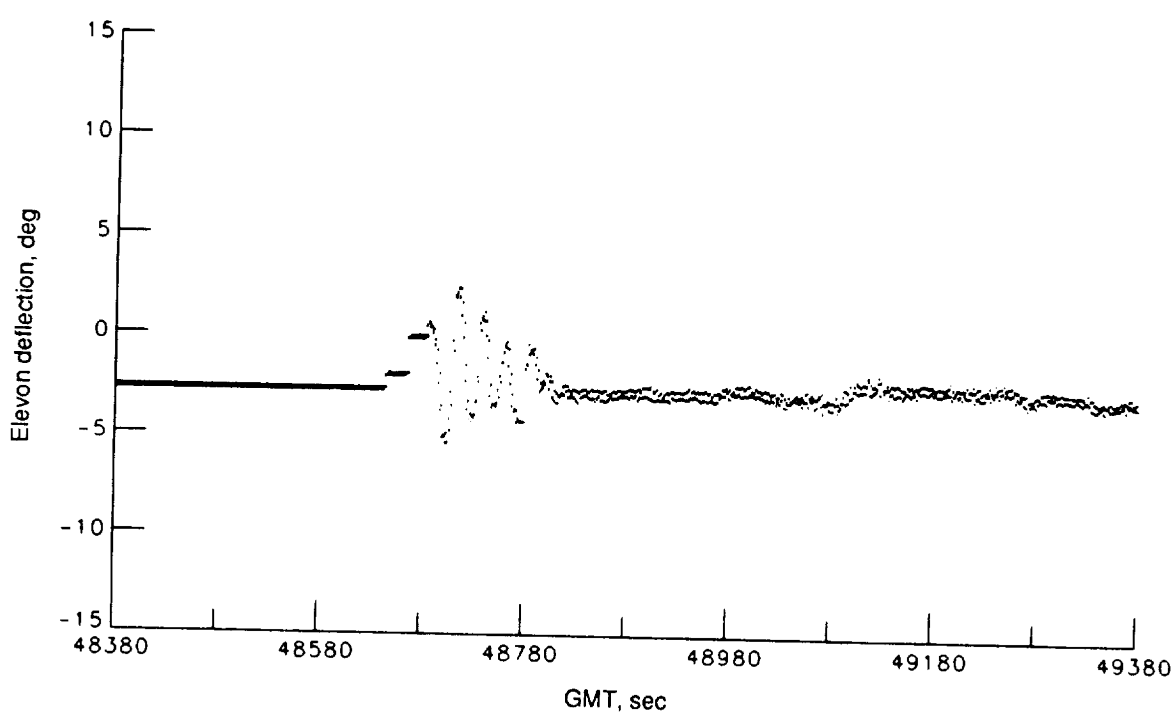


(e) Body flap deflection versus time.

Figure 22. Continued.

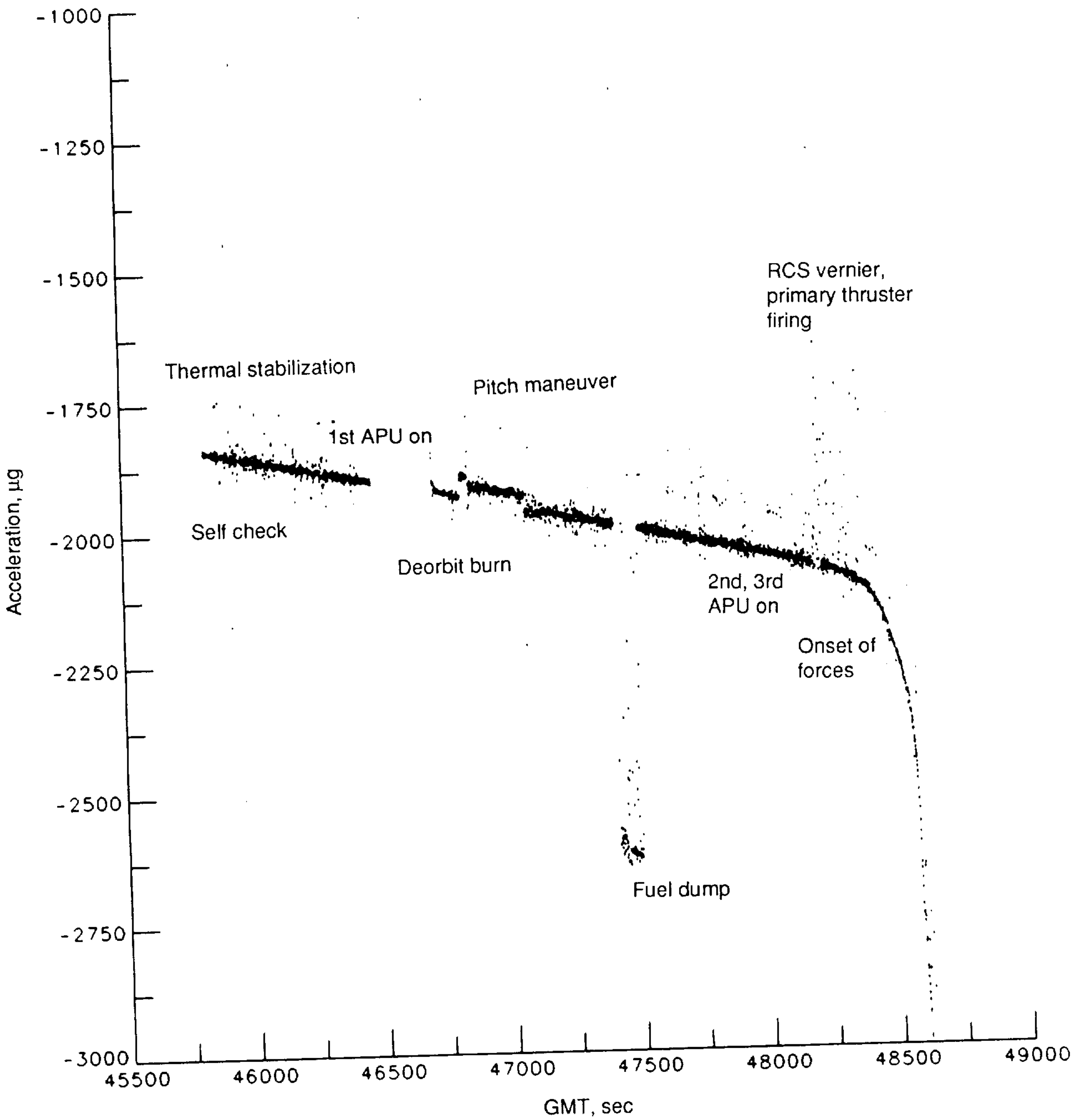


(f) Elevon deflection versus altitude.



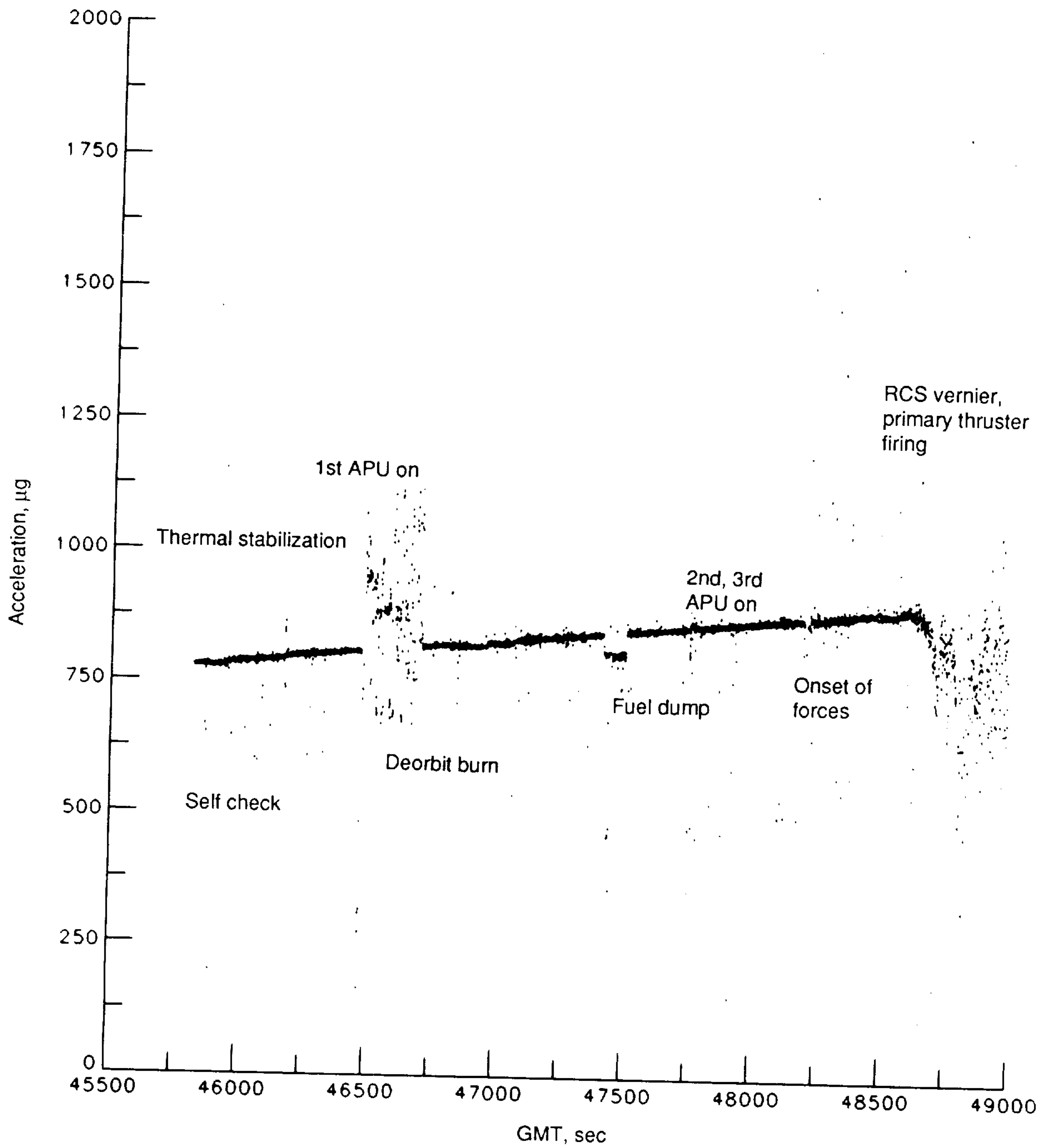
(g) Elevon deflection versus time.

Figure 22. Concluded.



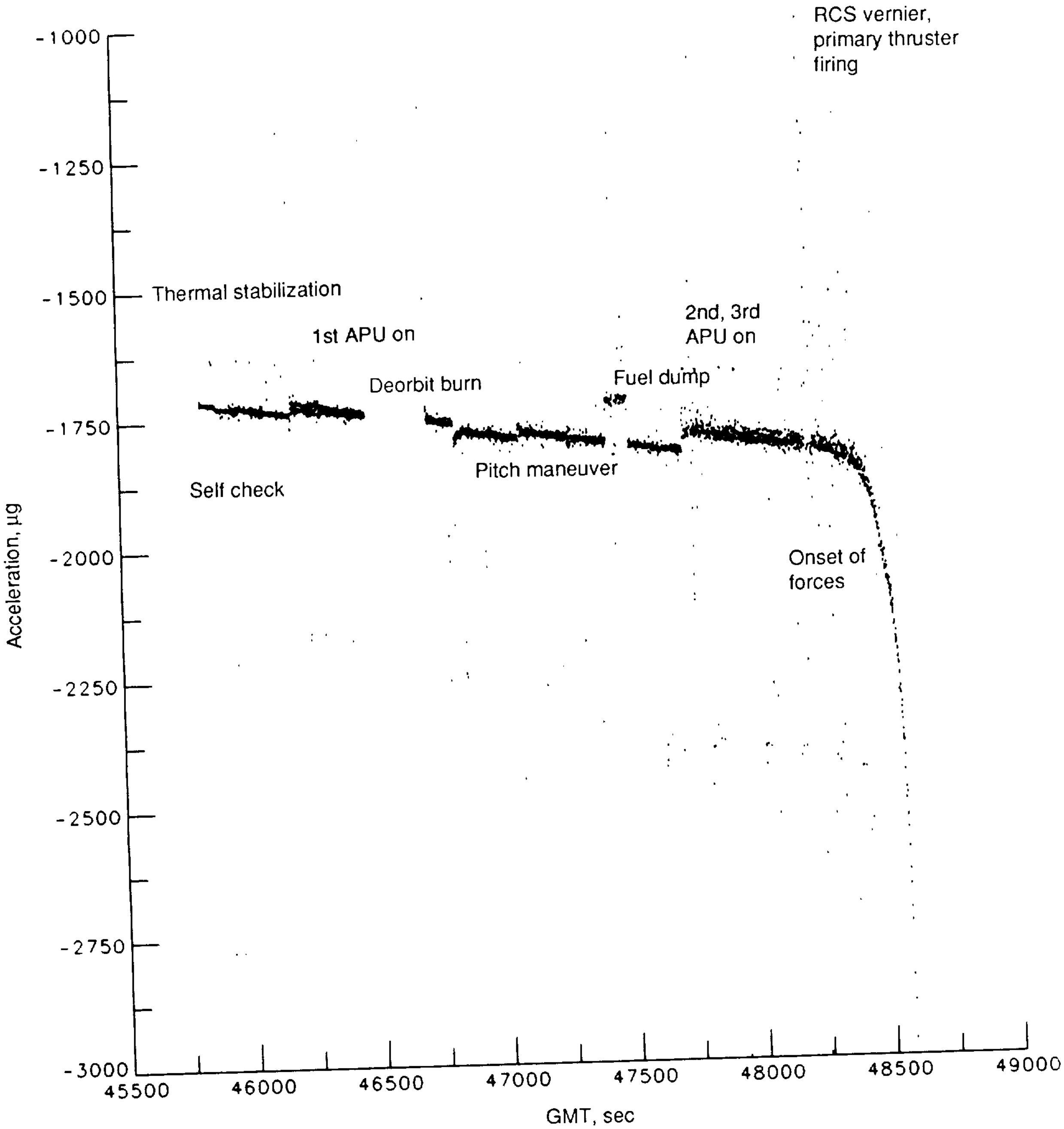
(a) X-axis acceleration and time-line events.

Figure 23. Section of acceleration data for STS-61C with time-line events labeled.



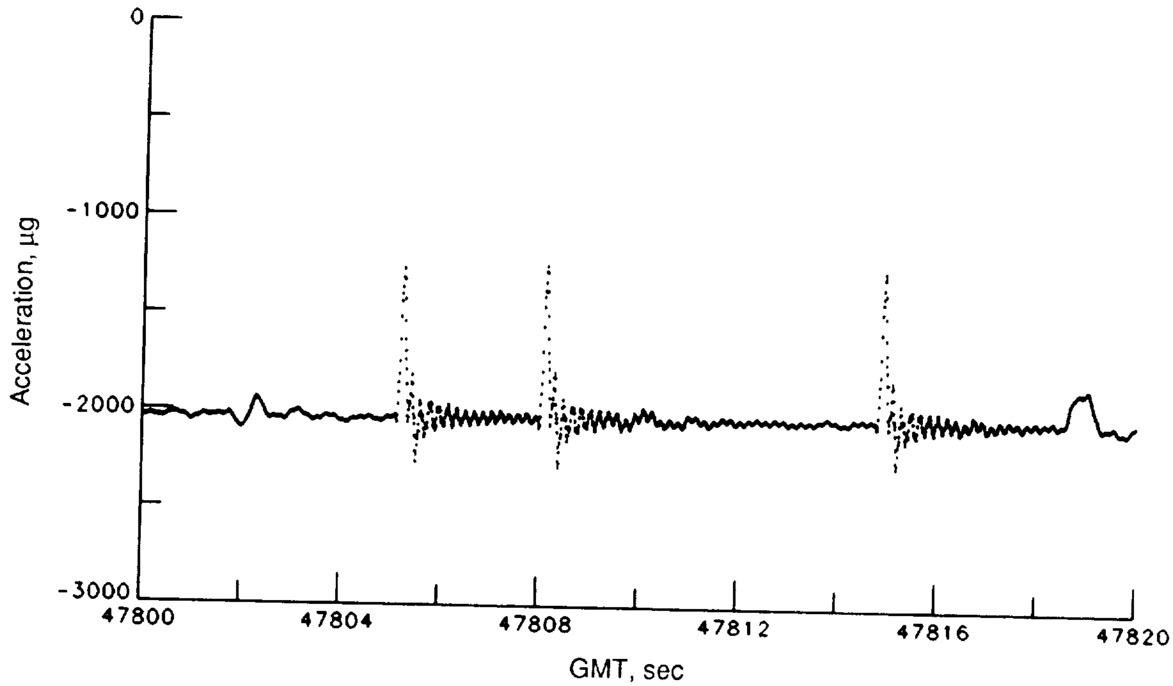
(b) Y-axis acceleration and time-line events.

Figure 23. Continued.

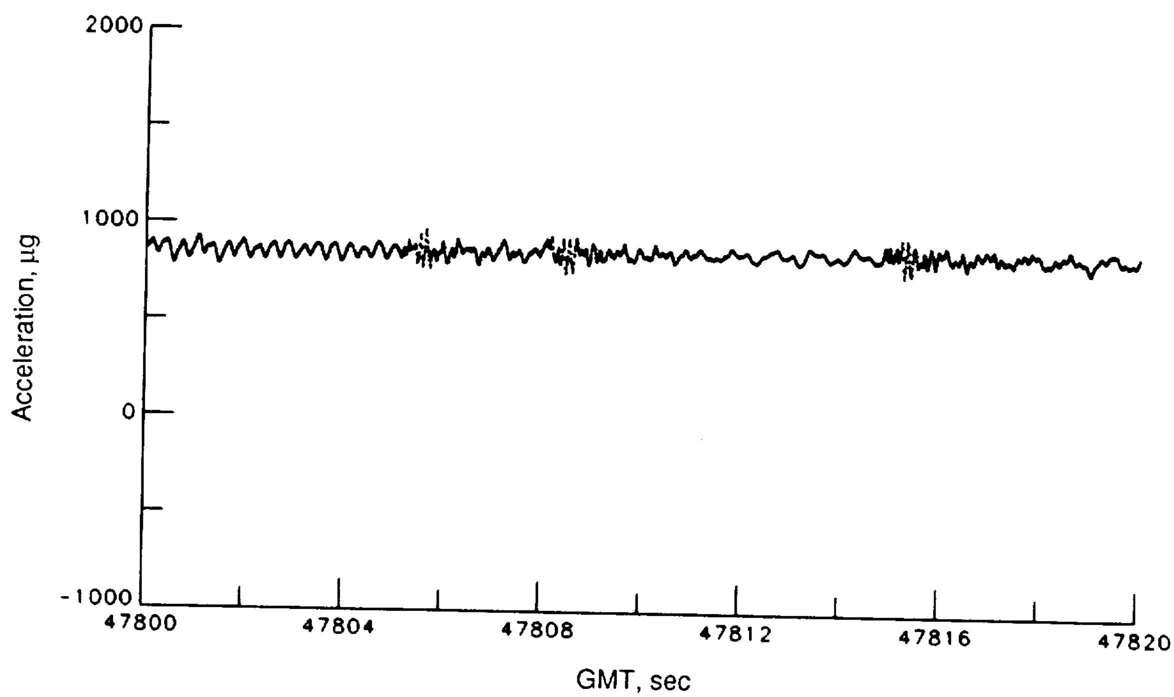


(c) Z-axis acceleration and time-line events.

Figure 23. Concluded.

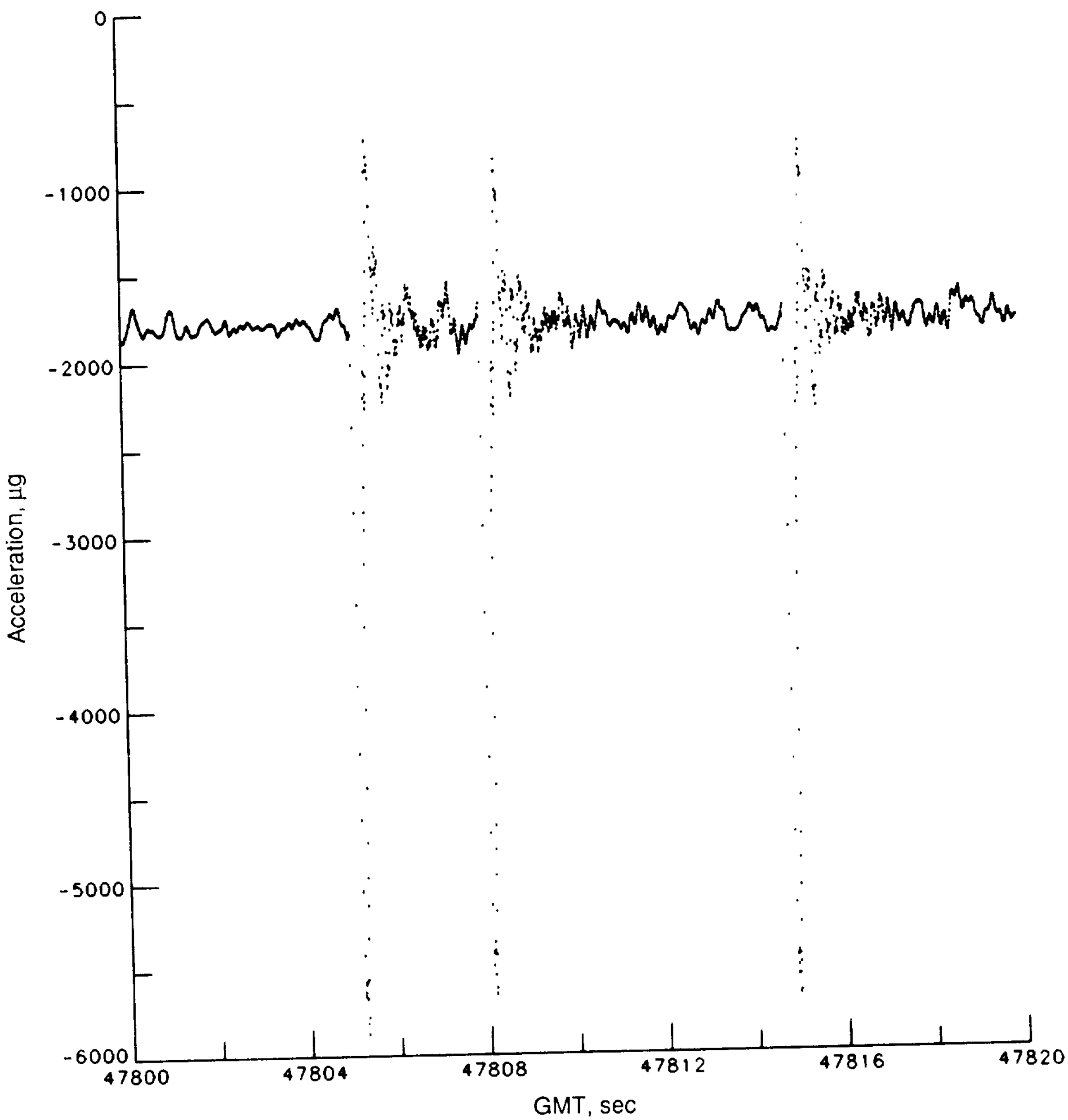


(a) X-axis acceleration.



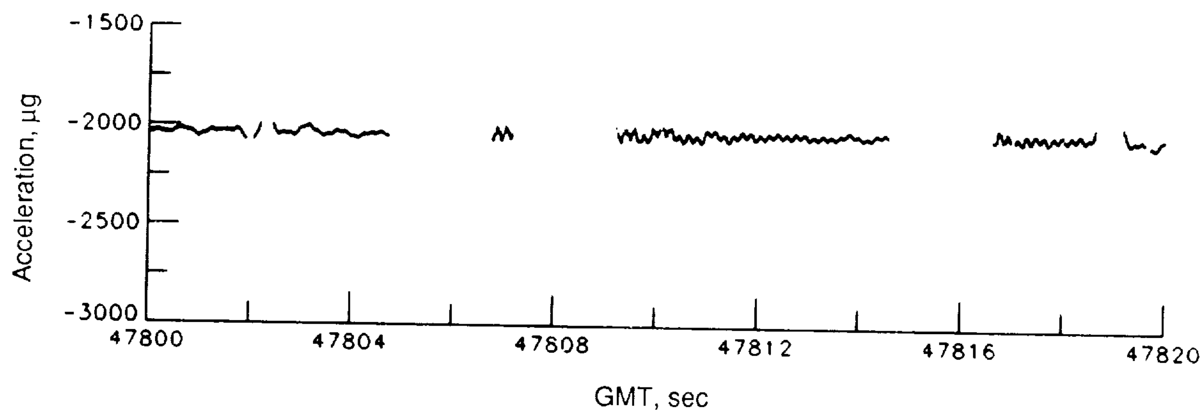
(b) Y-axis acceleration.

Figure 24. Section of acceleration data for STS-61C with RCS thruster activity.

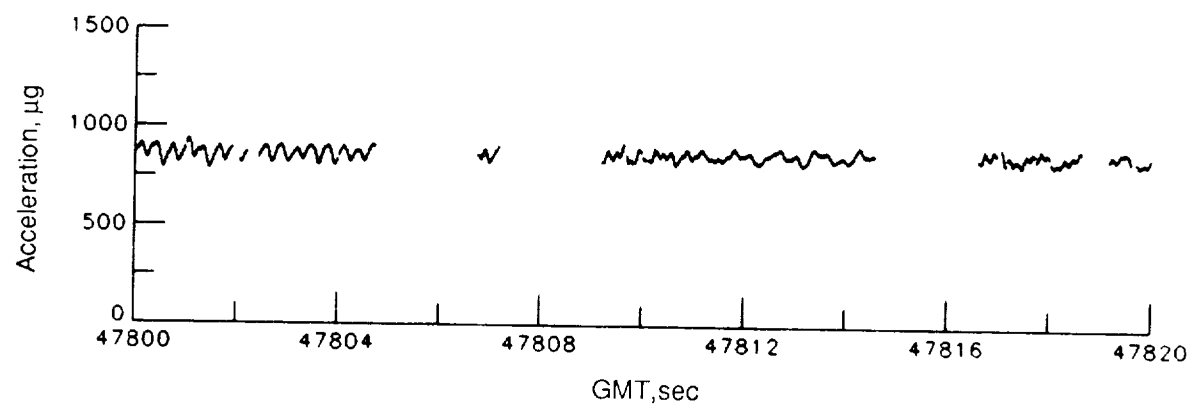


(c) Z-axis acceleration.

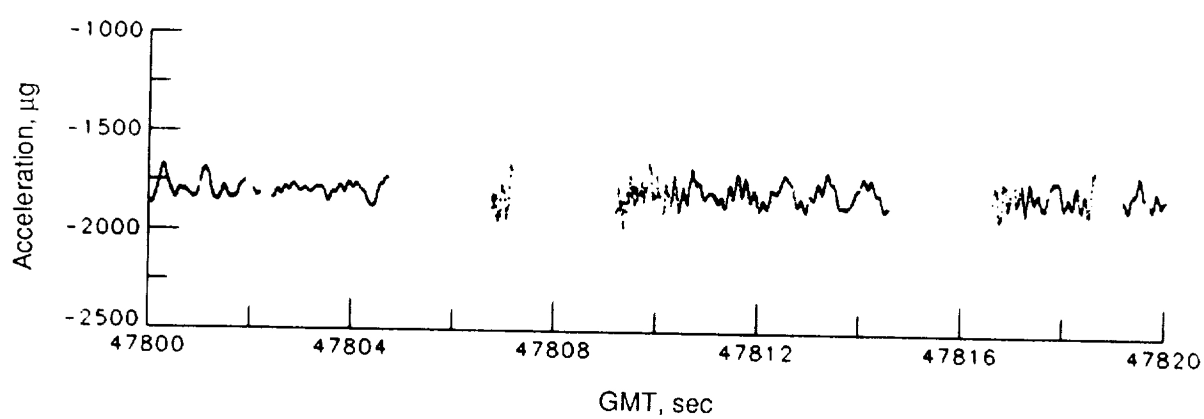
Figure 24. Concluded.



(a) X-axis acceleration.

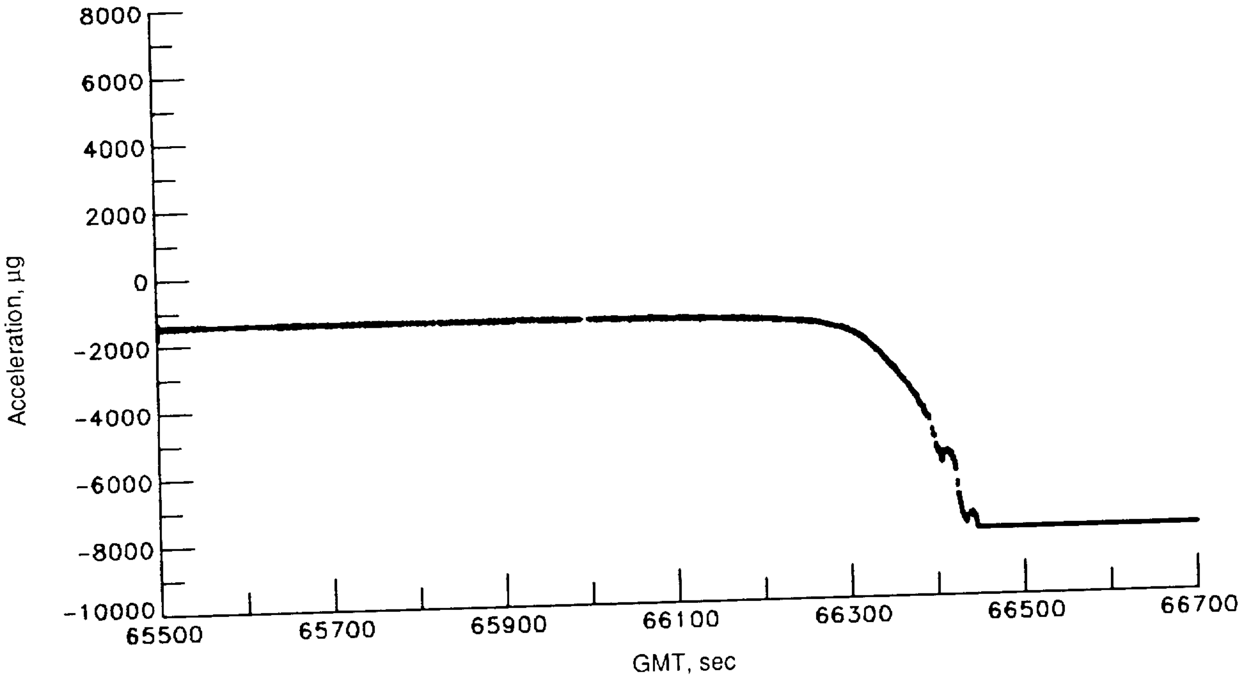


(b) Y-axis acceleration.

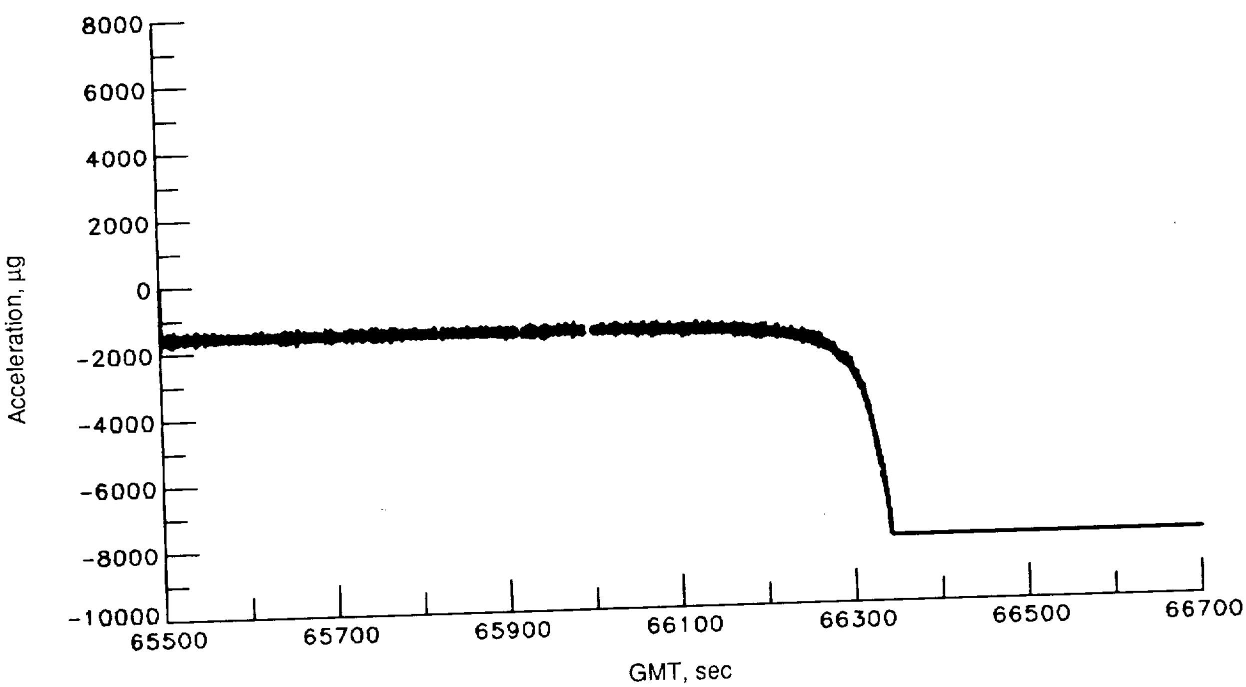


(c) Z-axis acceleration.

Figure 25. Section of acceleration data for STS-61C without RCS thruster activity.

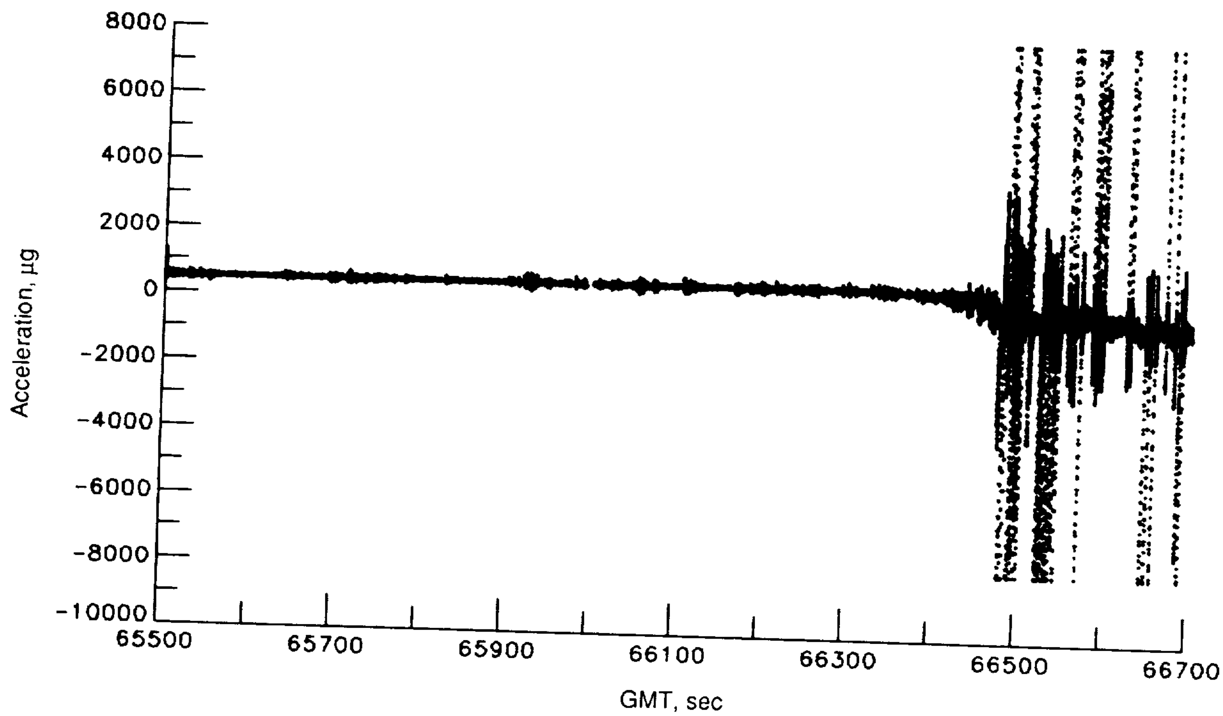


(a) X-axis acceleration.



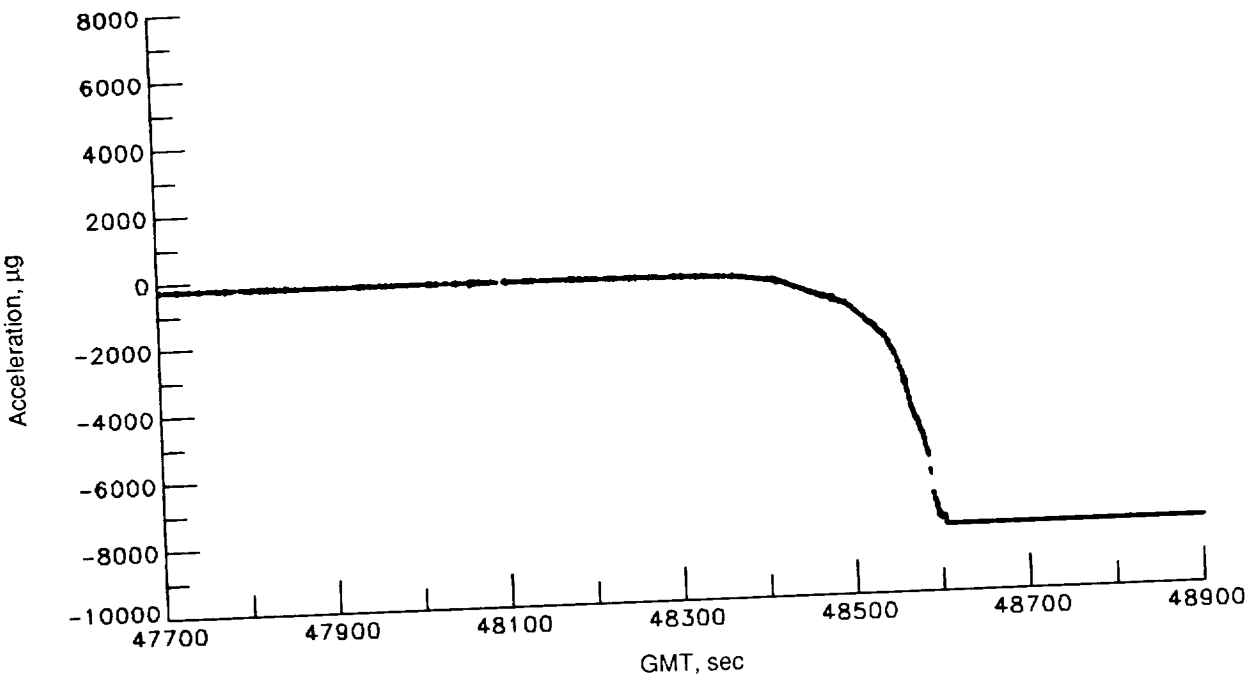
(b) Z-axis acceleration.

Figure 26. Acceleration versus time (no RCS signal) for STS-06.

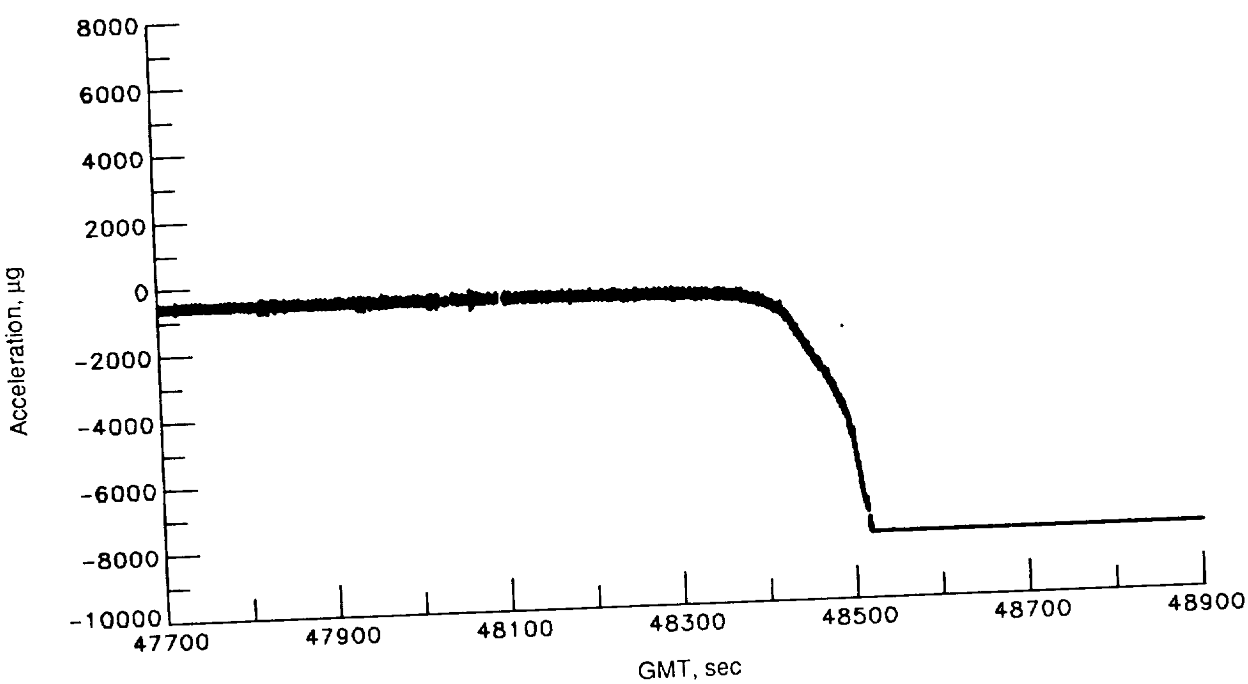


(c) Y-axis acceleration.

Figure 26. Concluded.

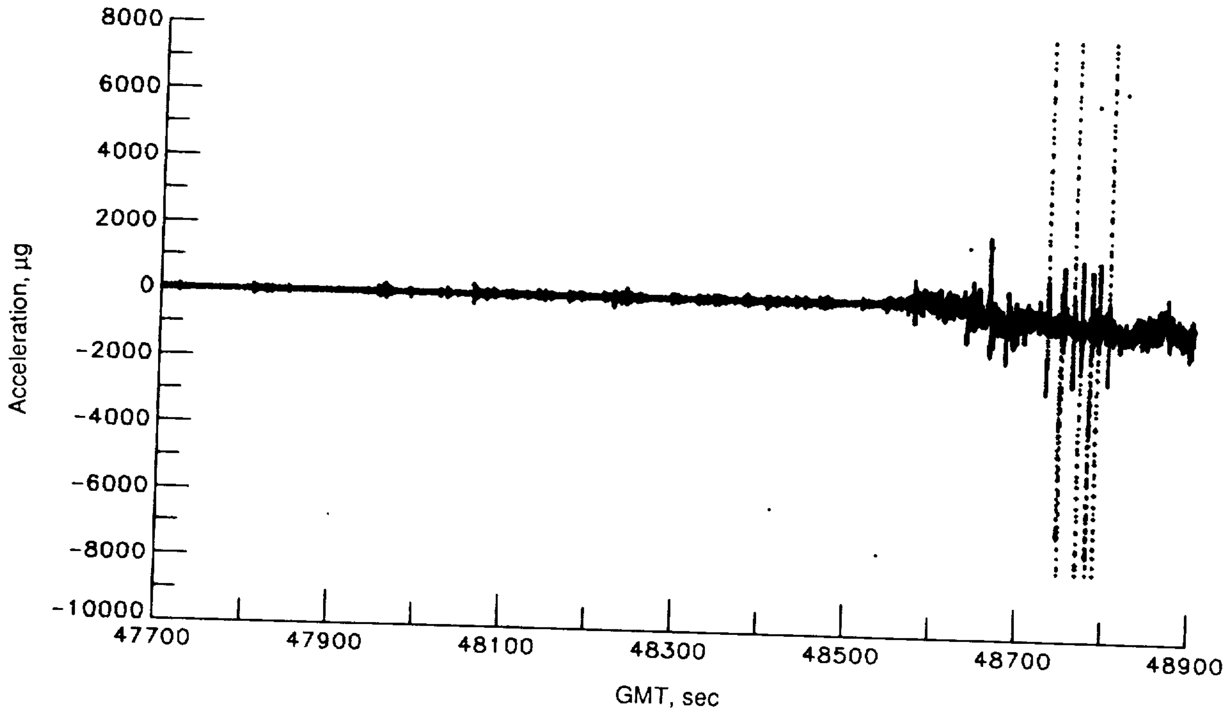


(a) X-axis acceleration.



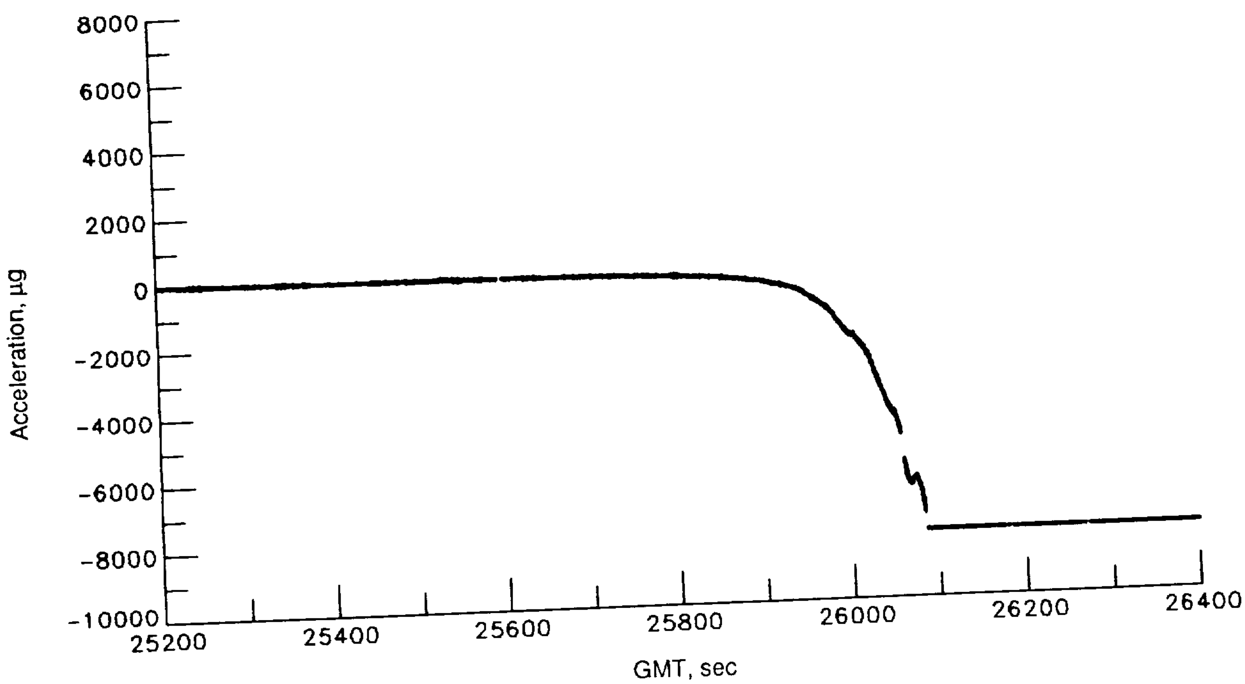
(b) Z-axis acceleration.

Figure 27. Acceleration versus time (no RCS signal) for STS-07.

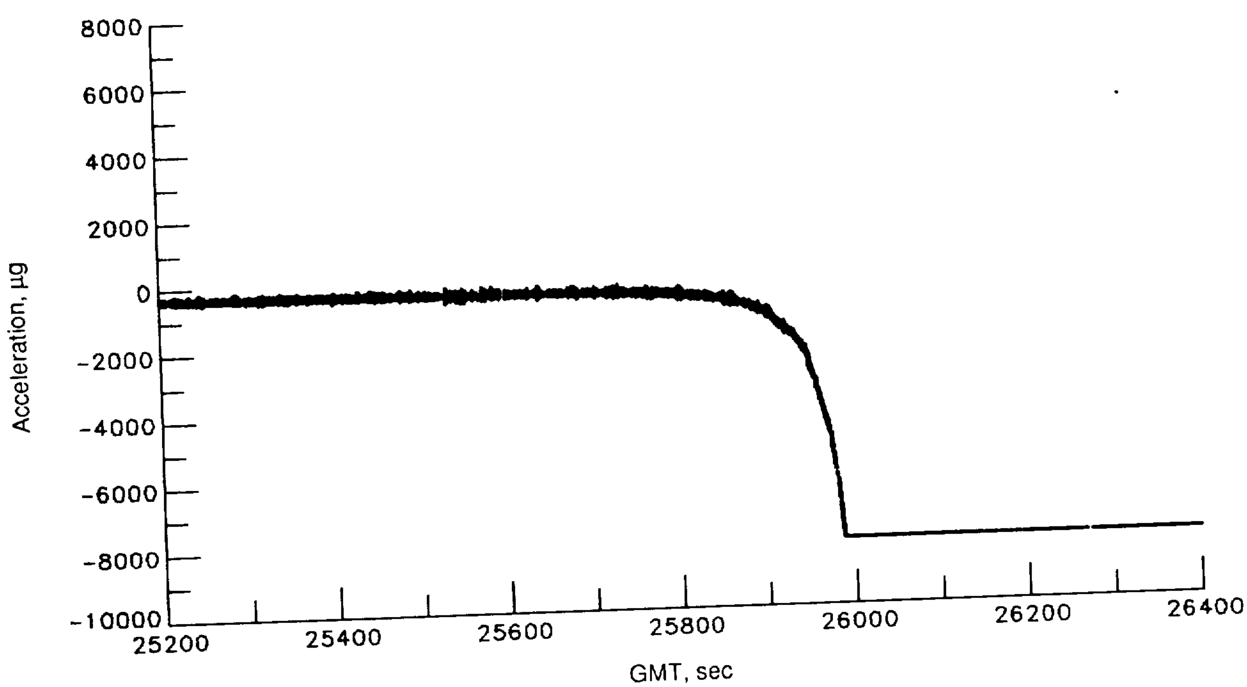


(c) Y-axis acceleration.

Figure 27. Concluded.

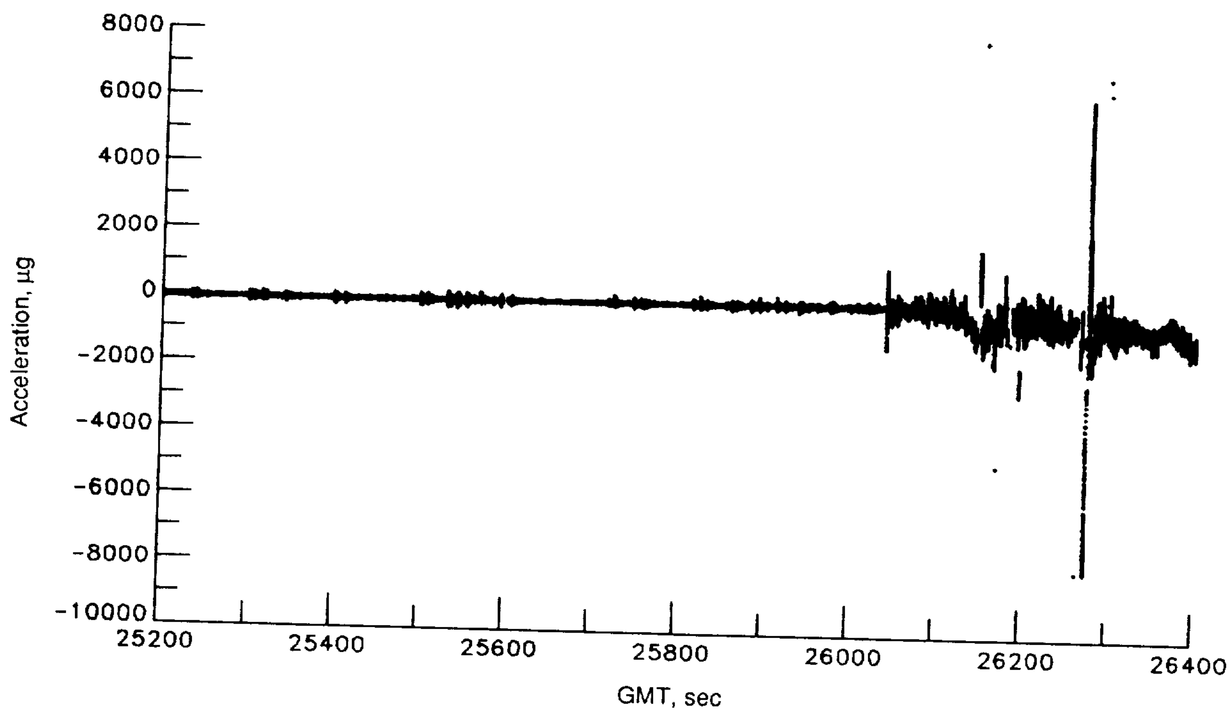


(a) X-axis acceleration.



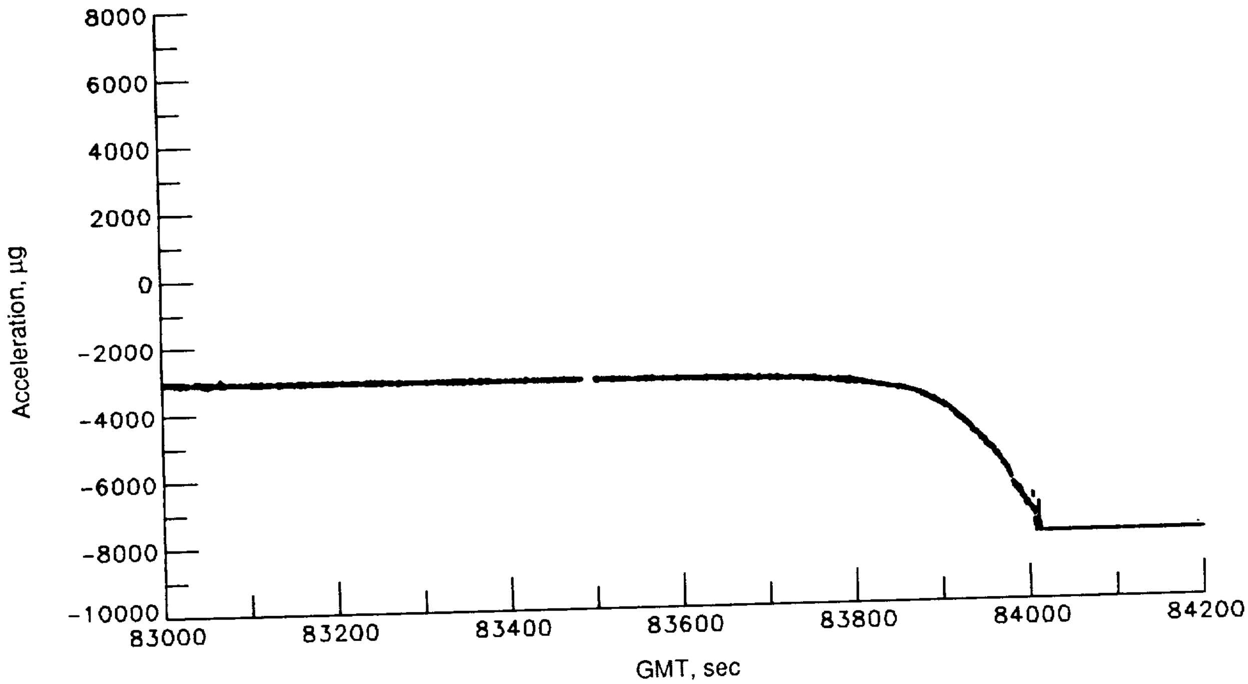
(b) Z-axis acceleration.

Figure 28. Acceleration versus time (no RCS signal) for STS-08.

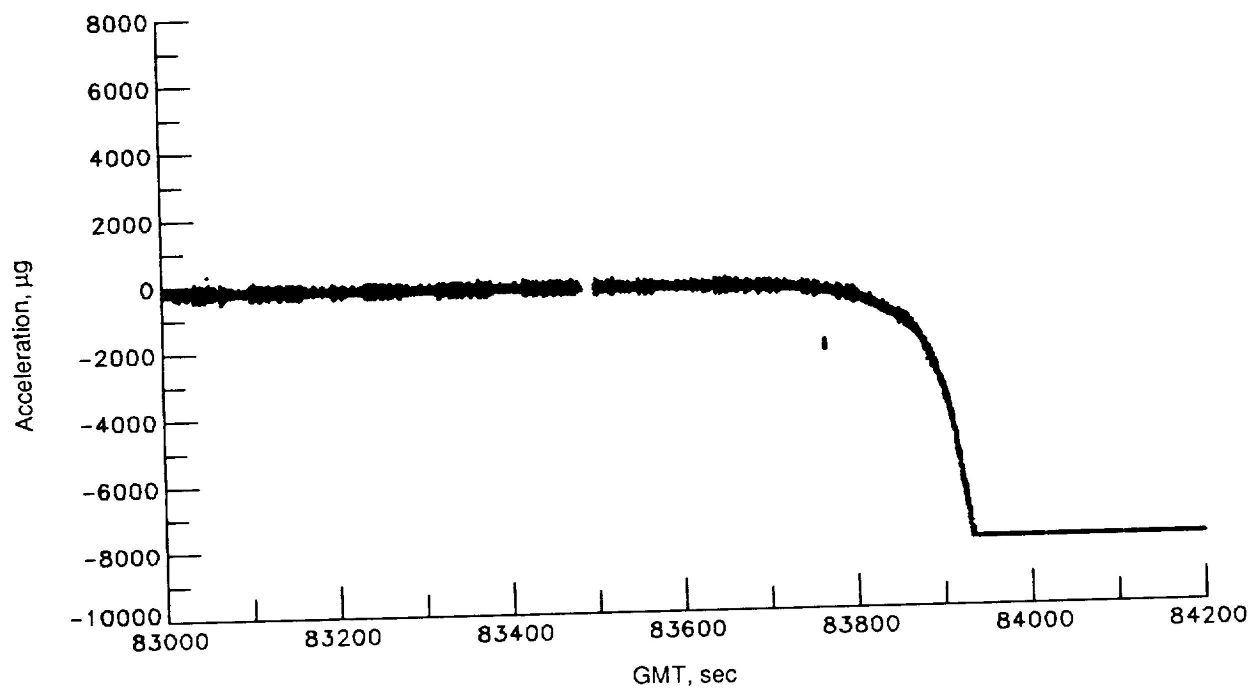


(c) Y-axis acceleration.

Figure ?3. Concluded.

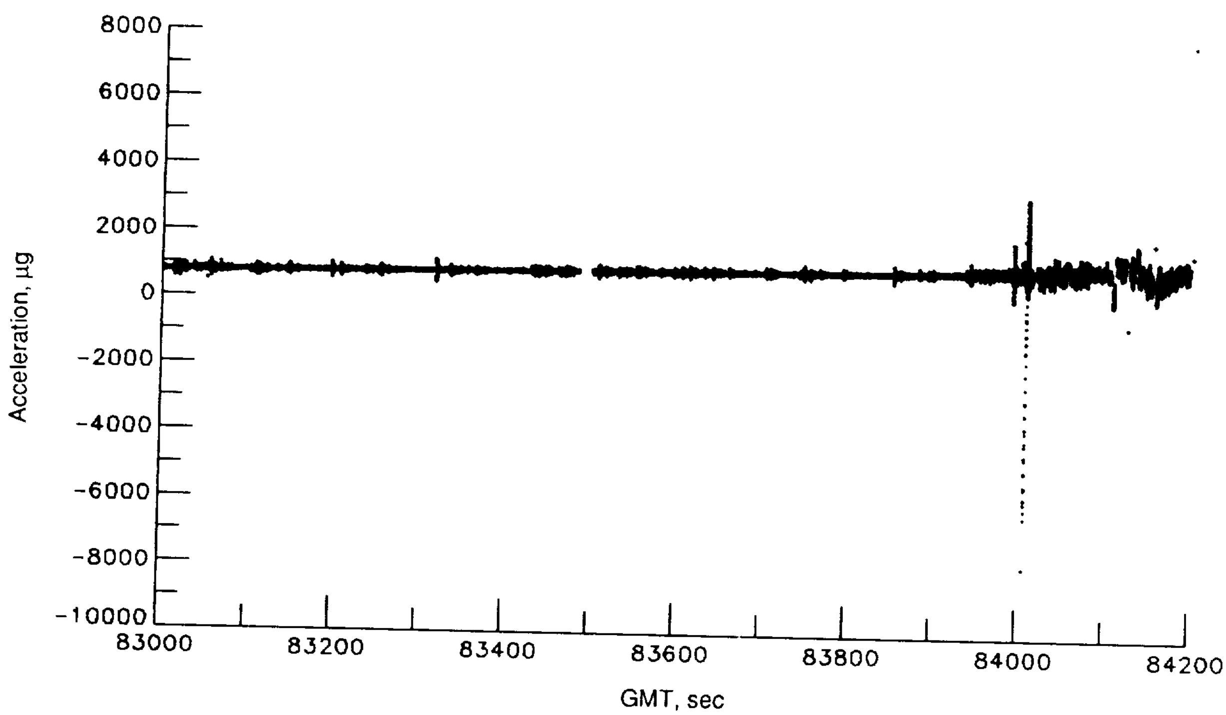


(a) X-axis acceleration.



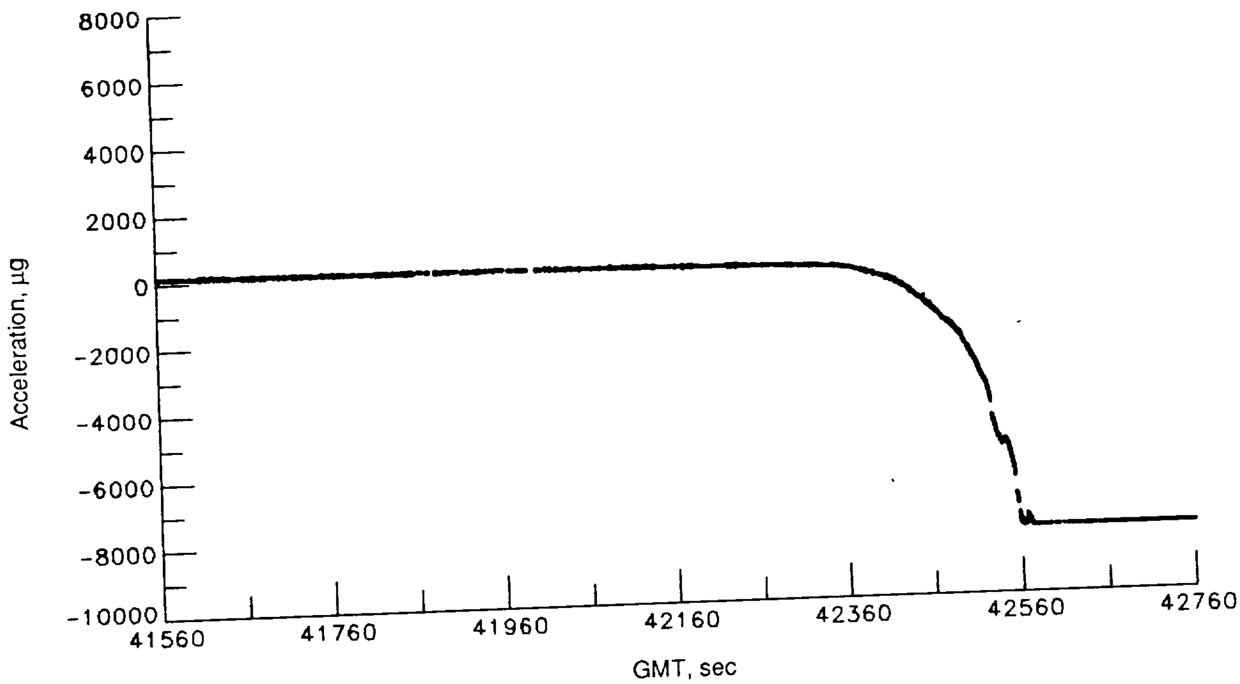
(b) Z-axis acceleration.

Figure 29. Acceleration versus time (no RCS signal) for STS-09.

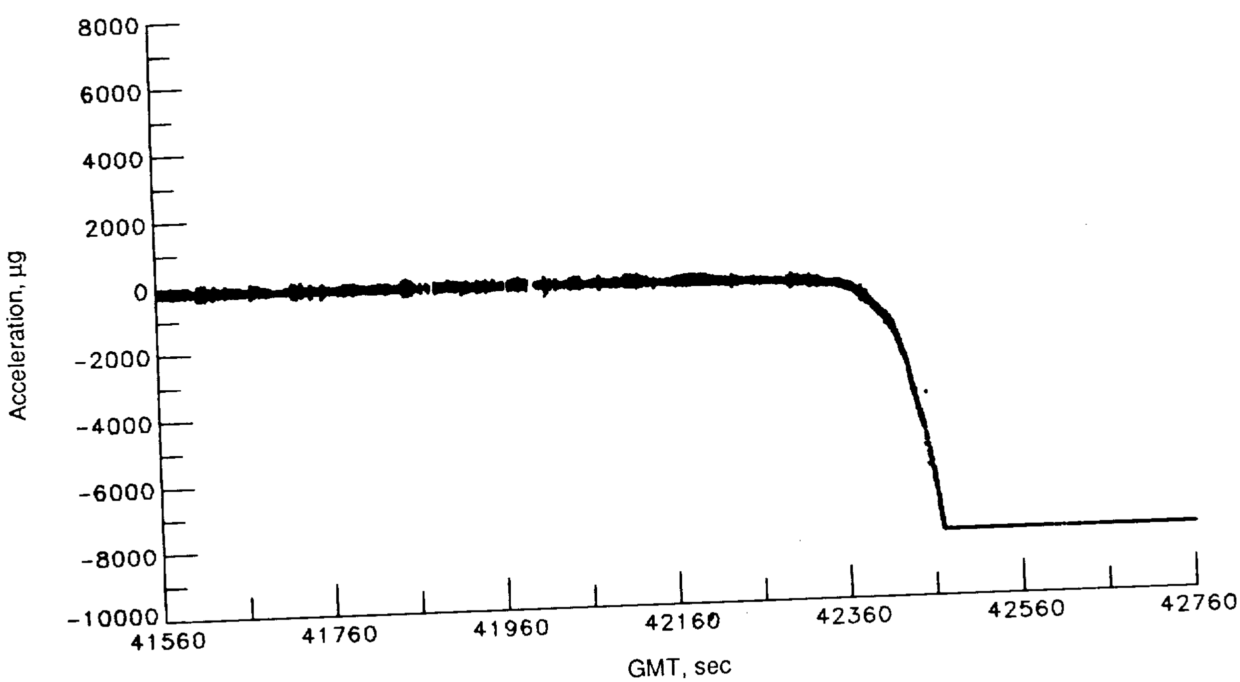


(c) Y-axis acceleration.

Figure 29. Concluded.

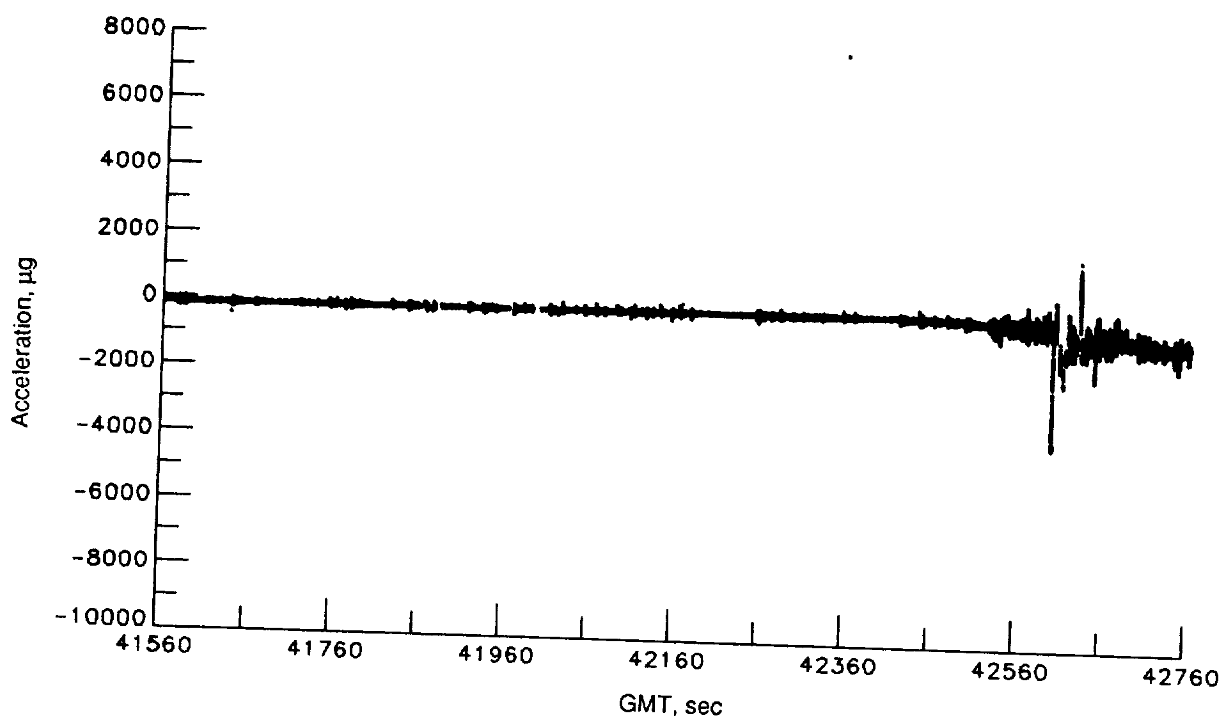


(a) X-axis acceleration.



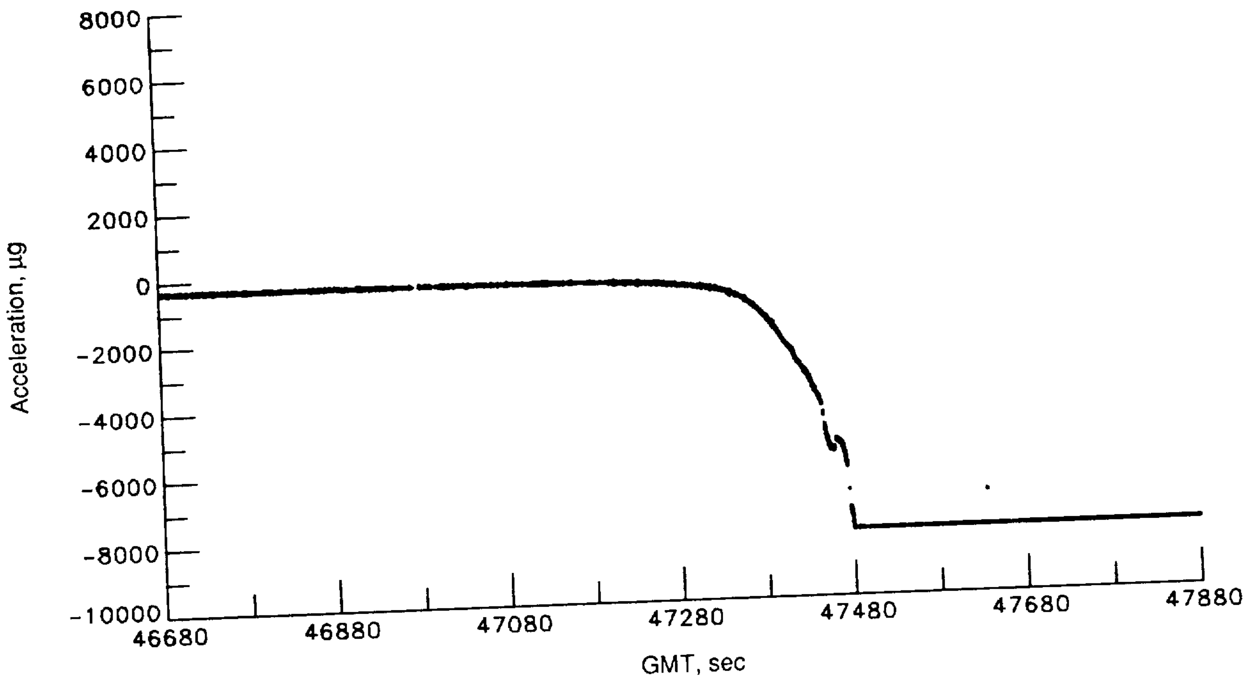
(b) Z-axis acceleration.

Figure 30. Acceleration versus time (no RCS signal) for STS-41B.

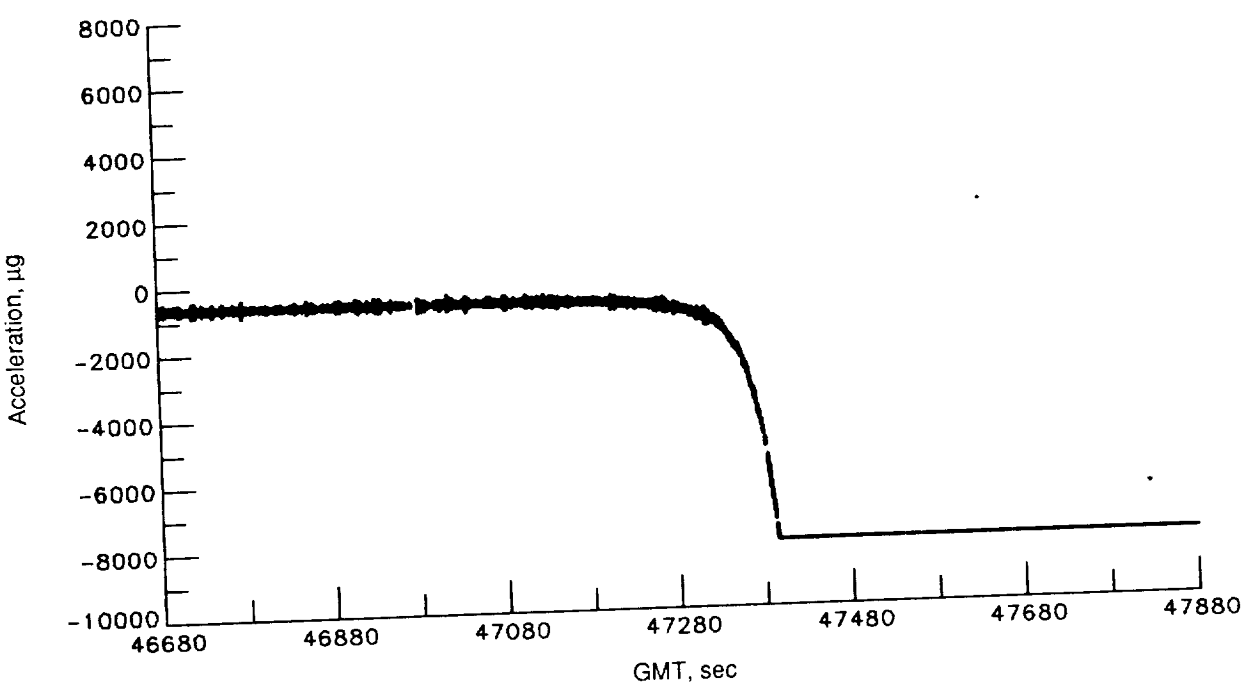


(c) Y-axis acceleration.

Figure 30. Concluded.

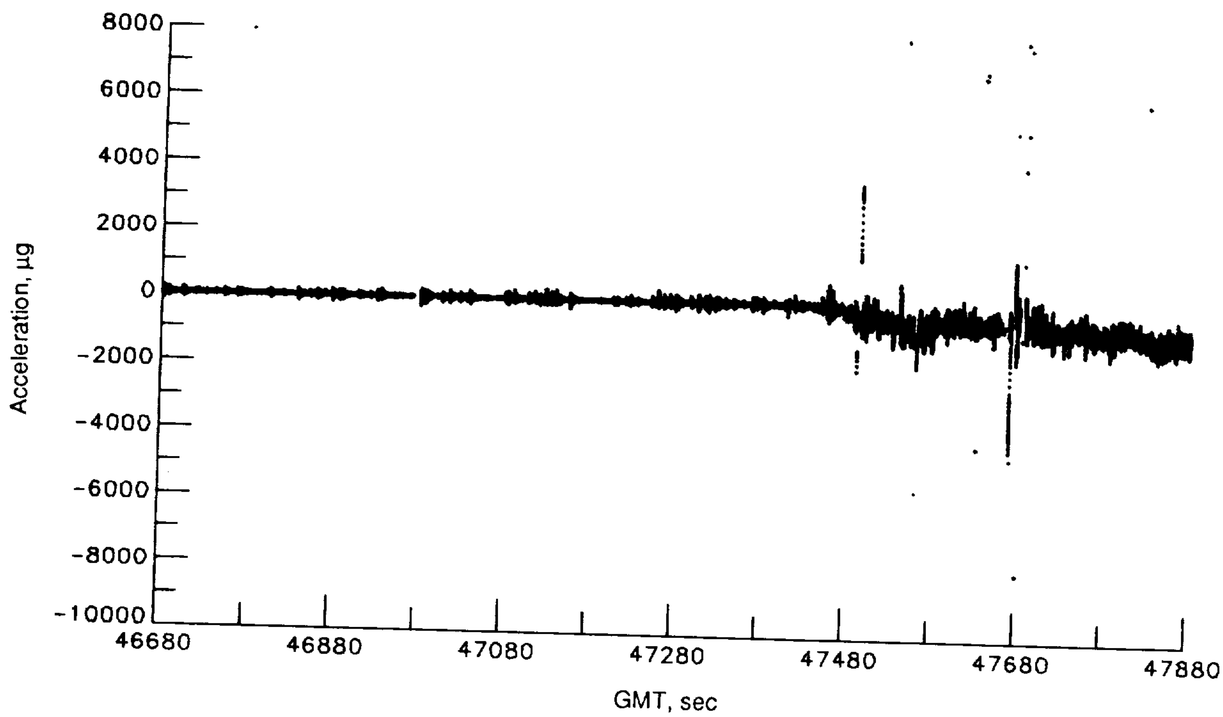


(a) X-axis acceleration.



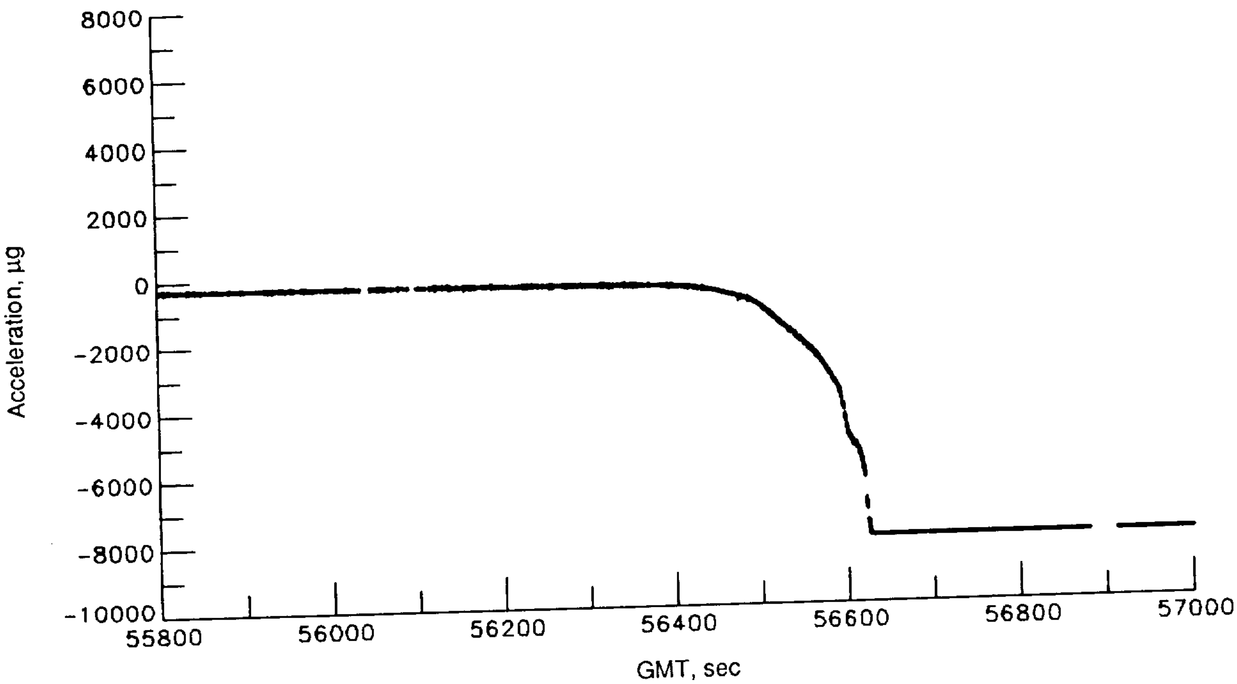
(b) Z-axis acceleration.

Figure 31. Acceleration versus time (no RCS signal) for STS-41C.

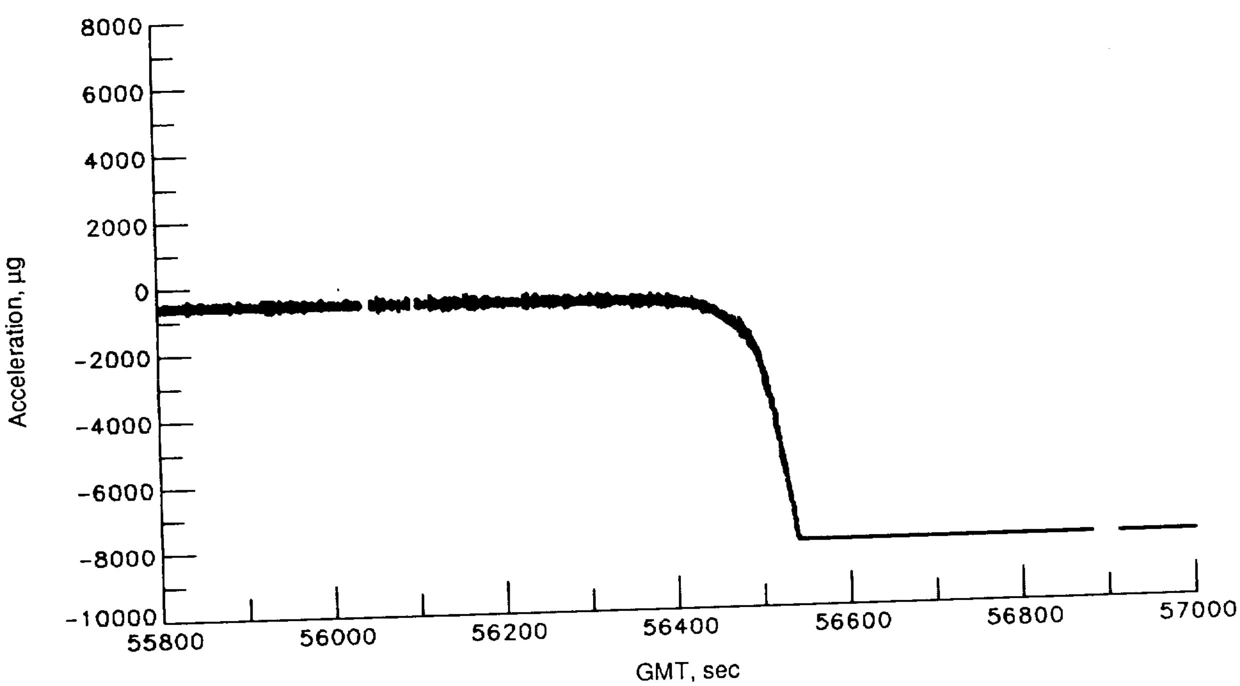


(c) Y-axis acceleration.

Figure 31. Concluded.

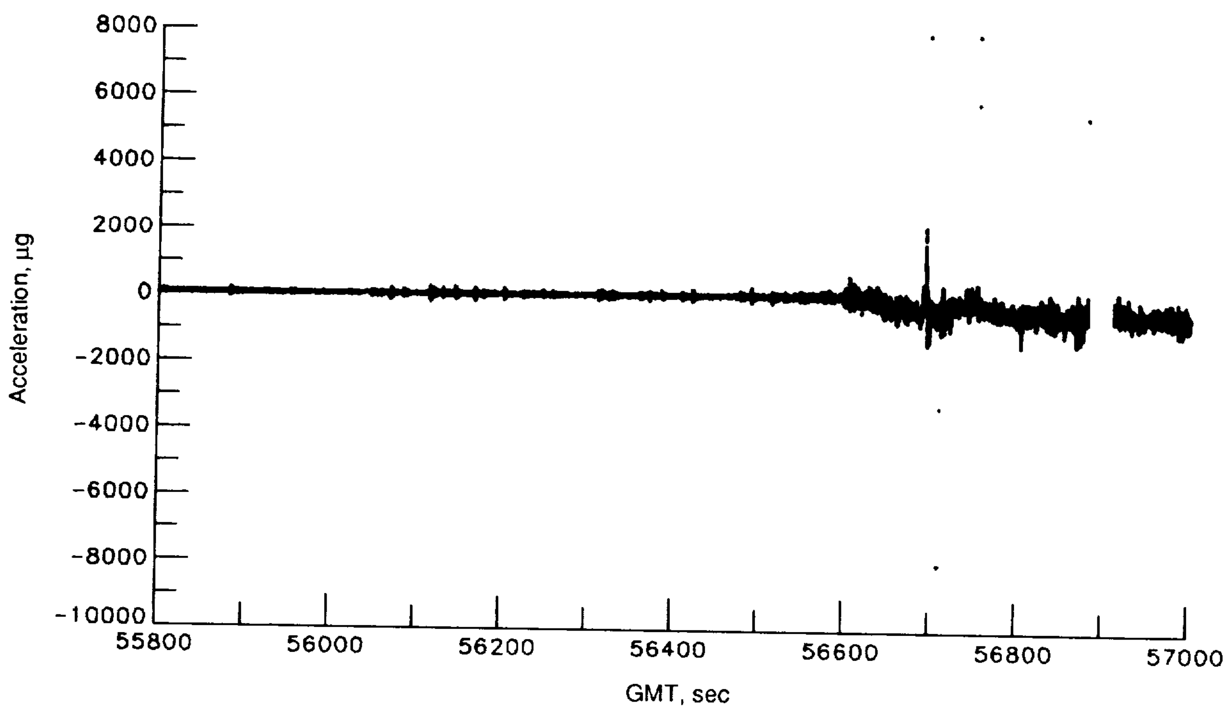


(a) X-axis acceleration.



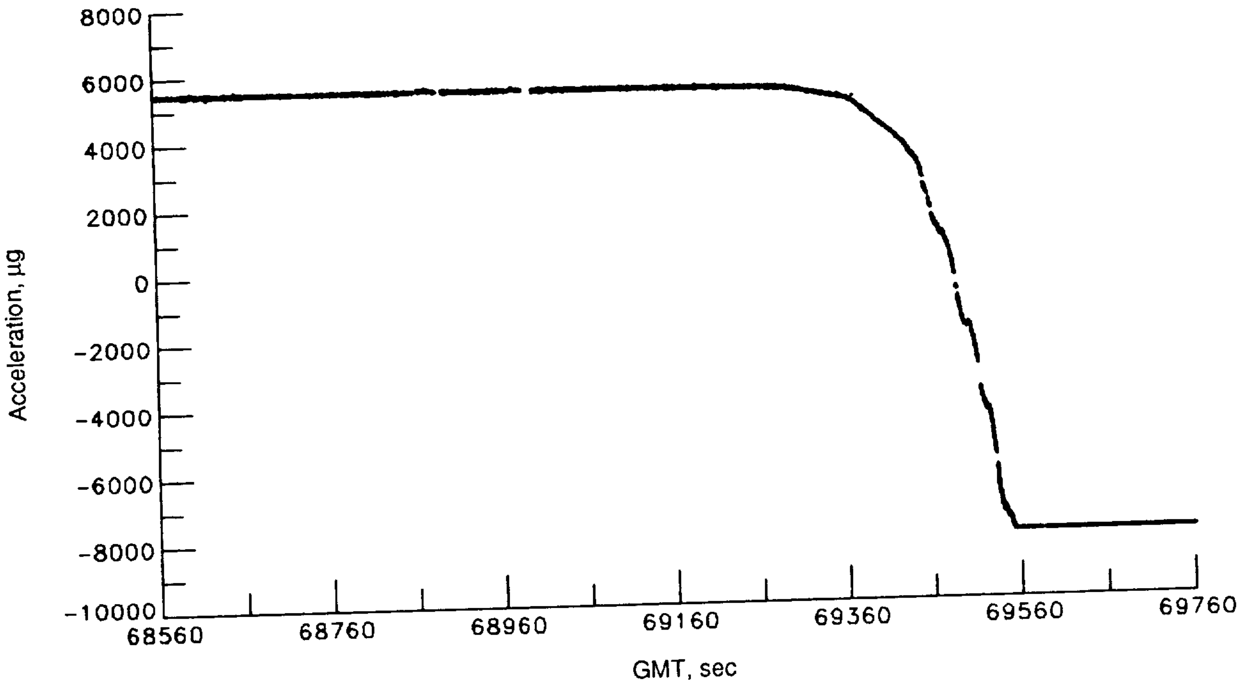
(b) Z-axis acceleration.

Figure 32. Acceleration versus time (no RCS signal) for STS-51B.

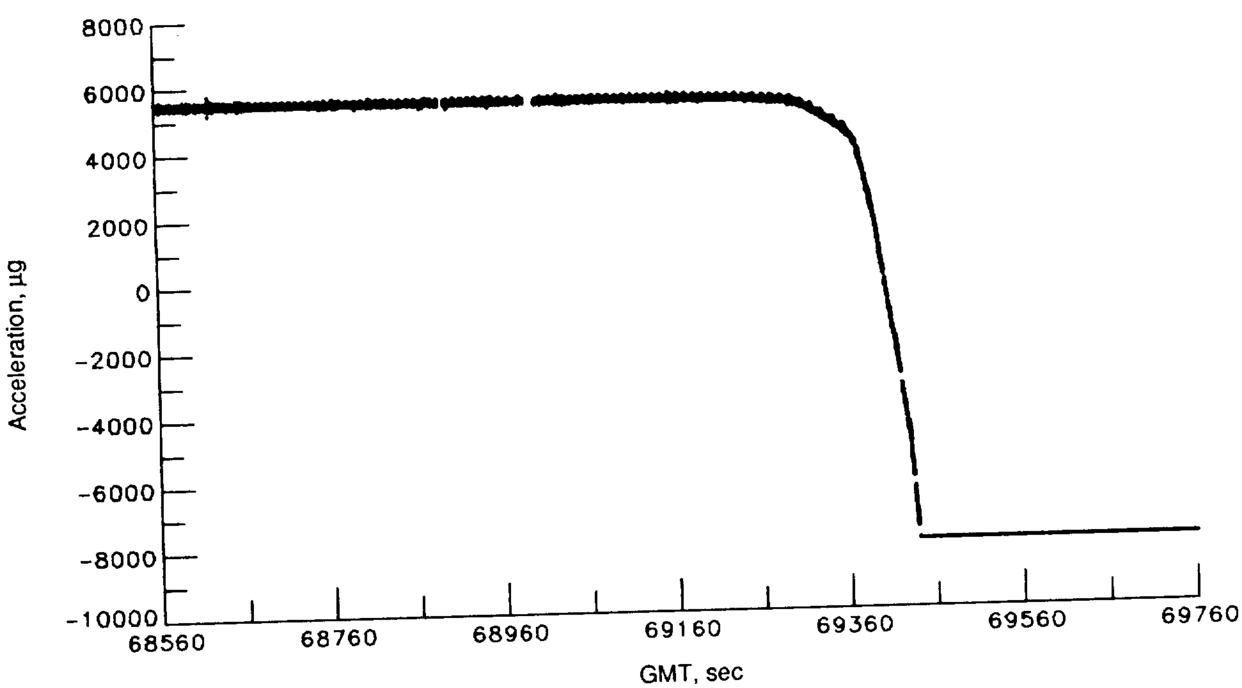


(c) Y-axis acceleration.

Figure 32. Concluded.

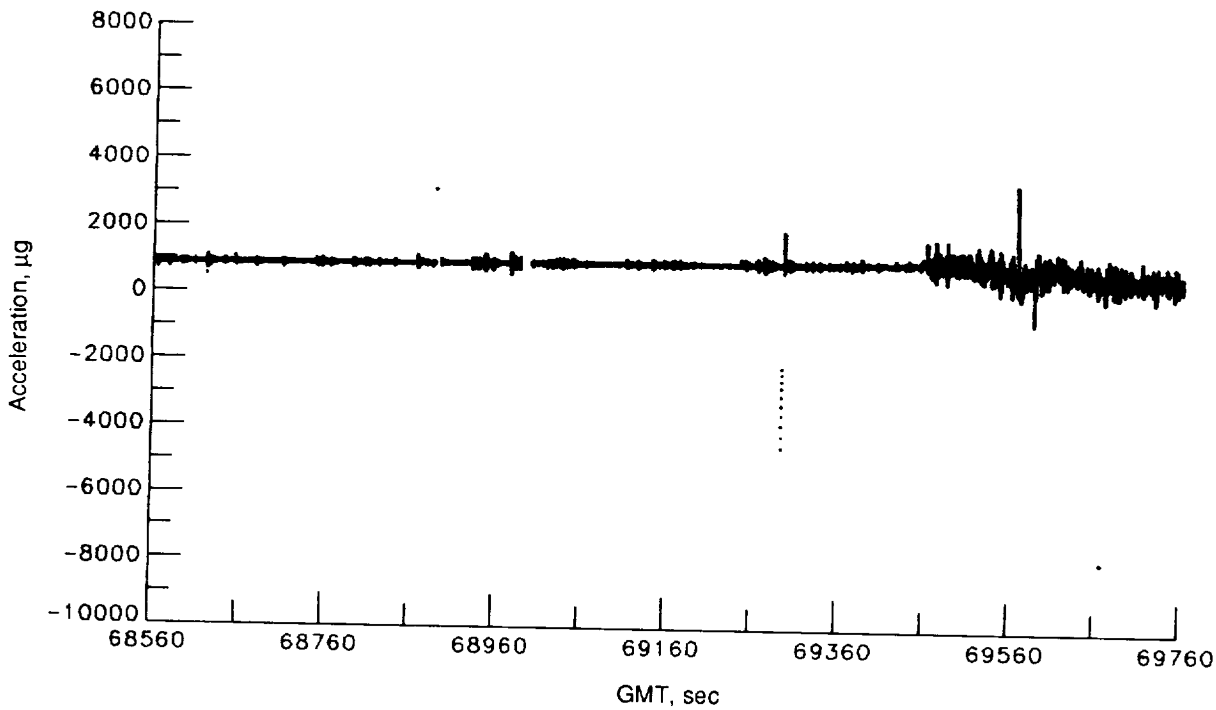


(a) X-axis acceleration.



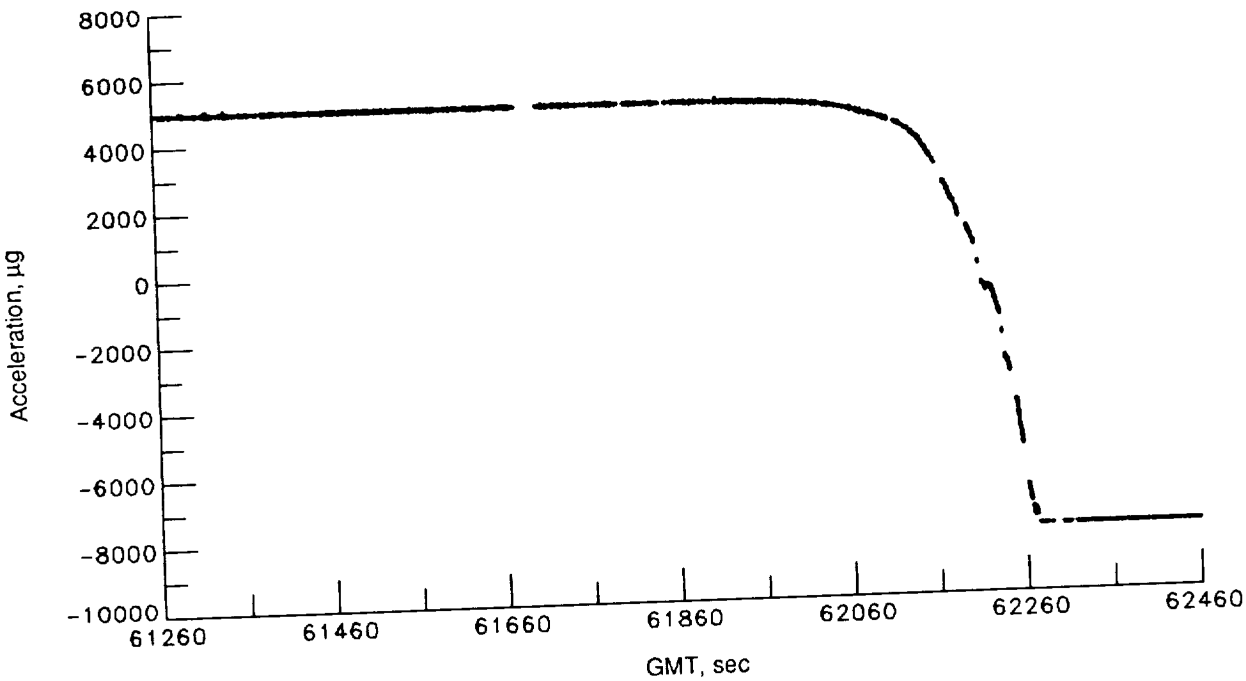
(b) Z-axis acceleration.

Figure 33. Acceleration versus time (no RCS signal) for STS-51F.

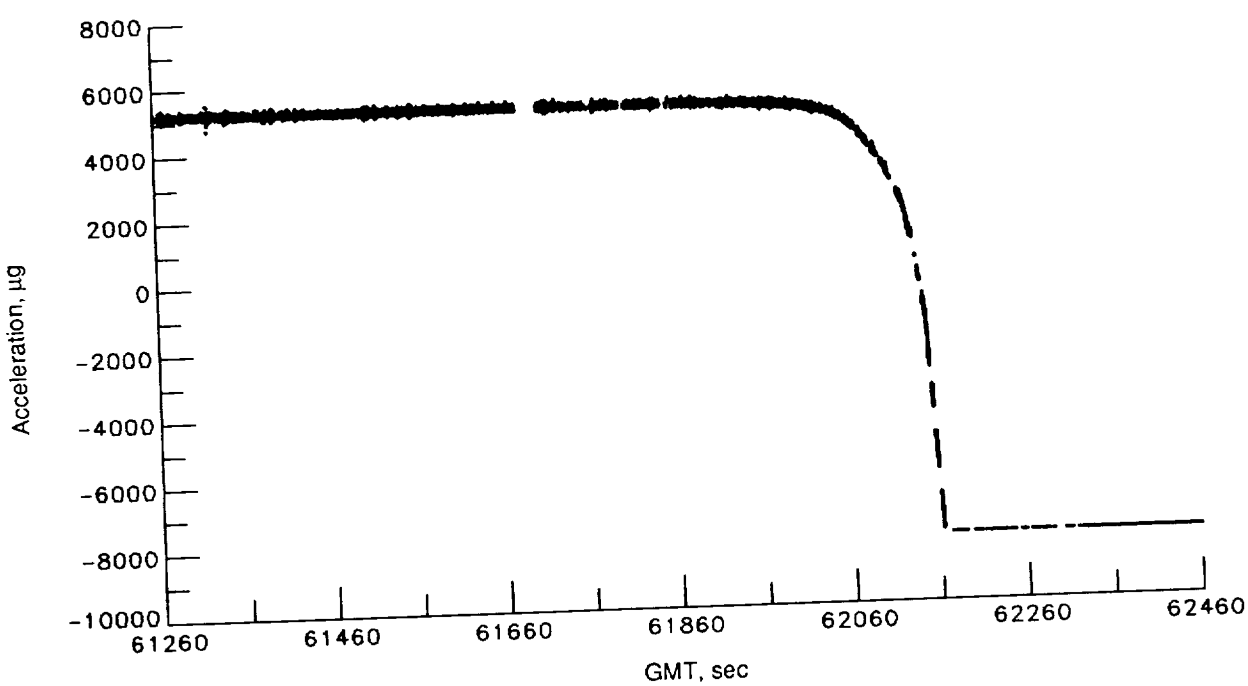


(c) Y-axis acceleration.

Figure 33. Concluded.

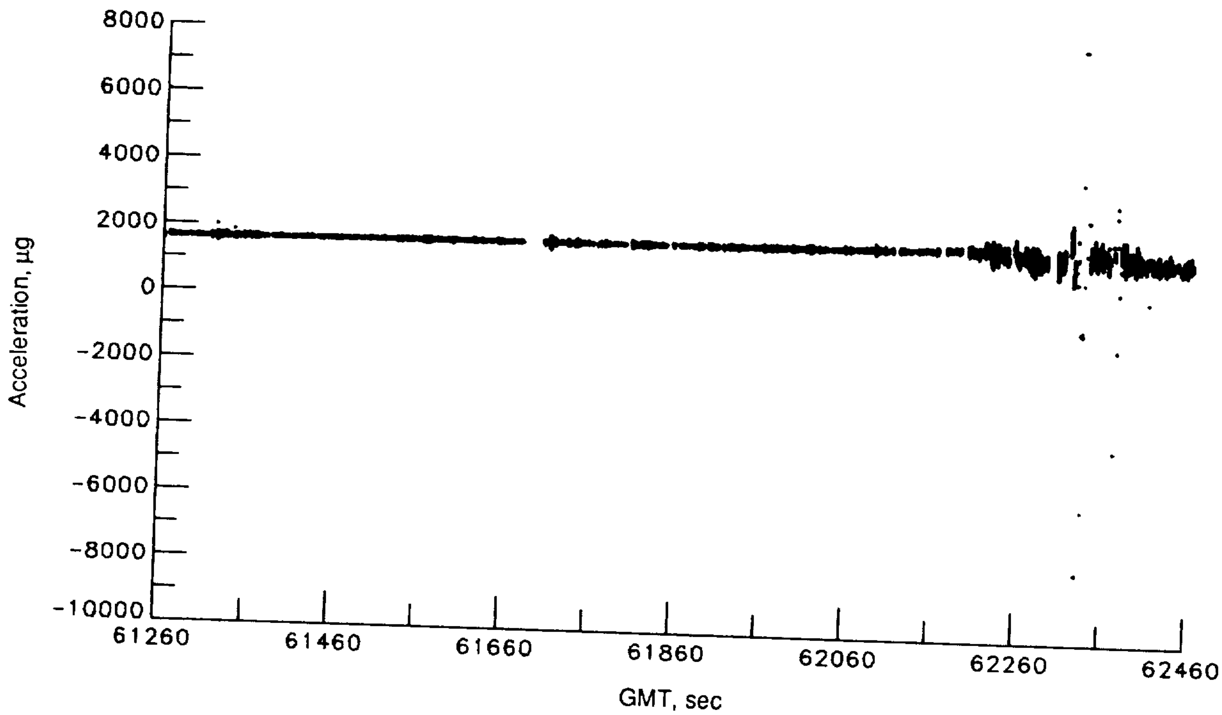


(a) X-axis acceleration.



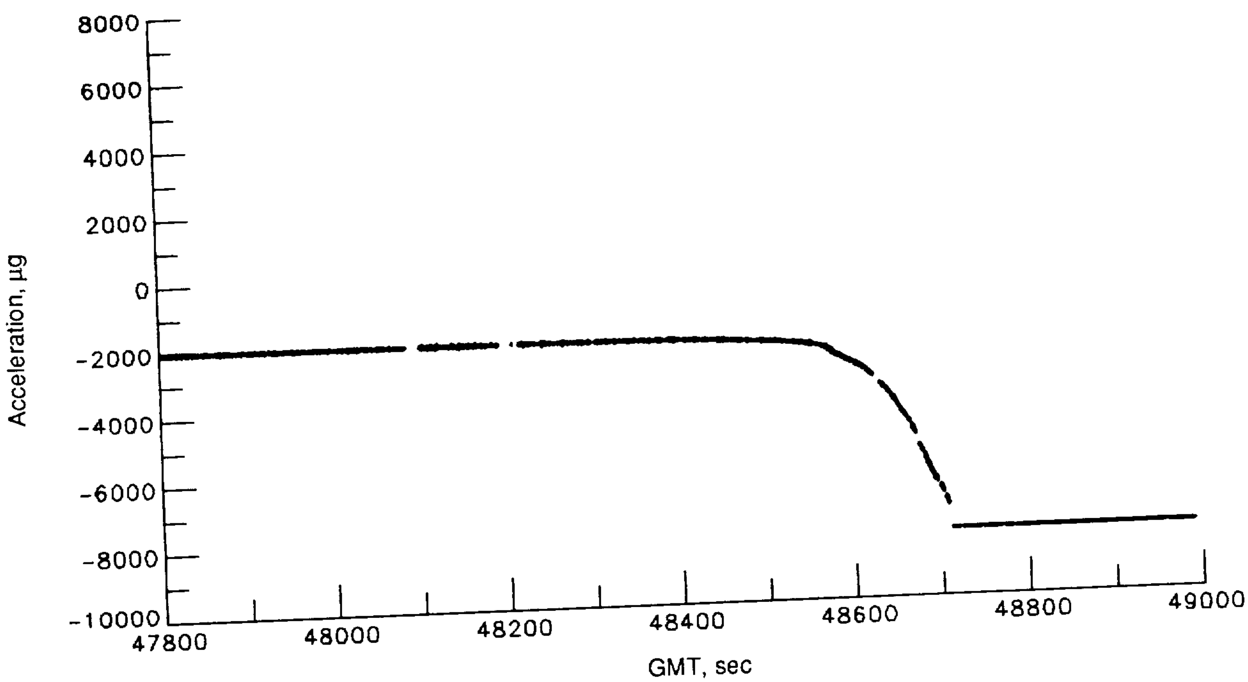
(b) Z-axis acceleration.

Figure 34. Acceleration versus time (no RCS signal) for STS-61A.

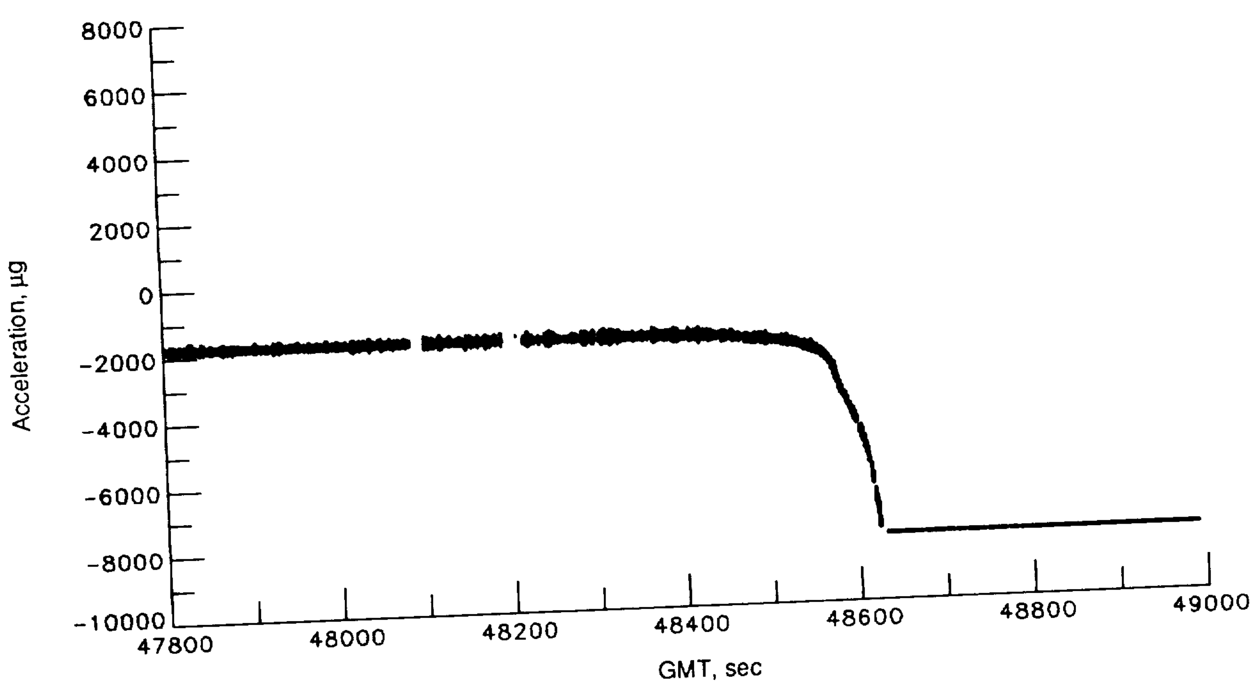


(c) Y-axis acceleration.

Figure 34. Concluded.

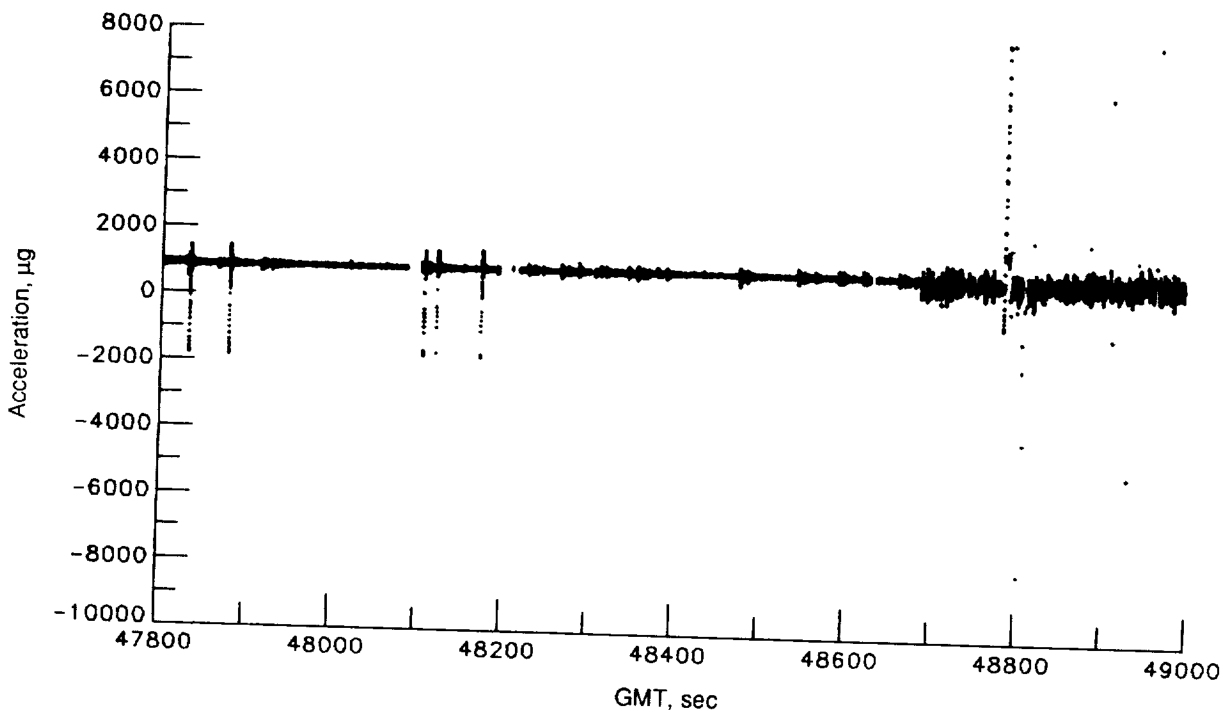


(a) X-axis acceleration.



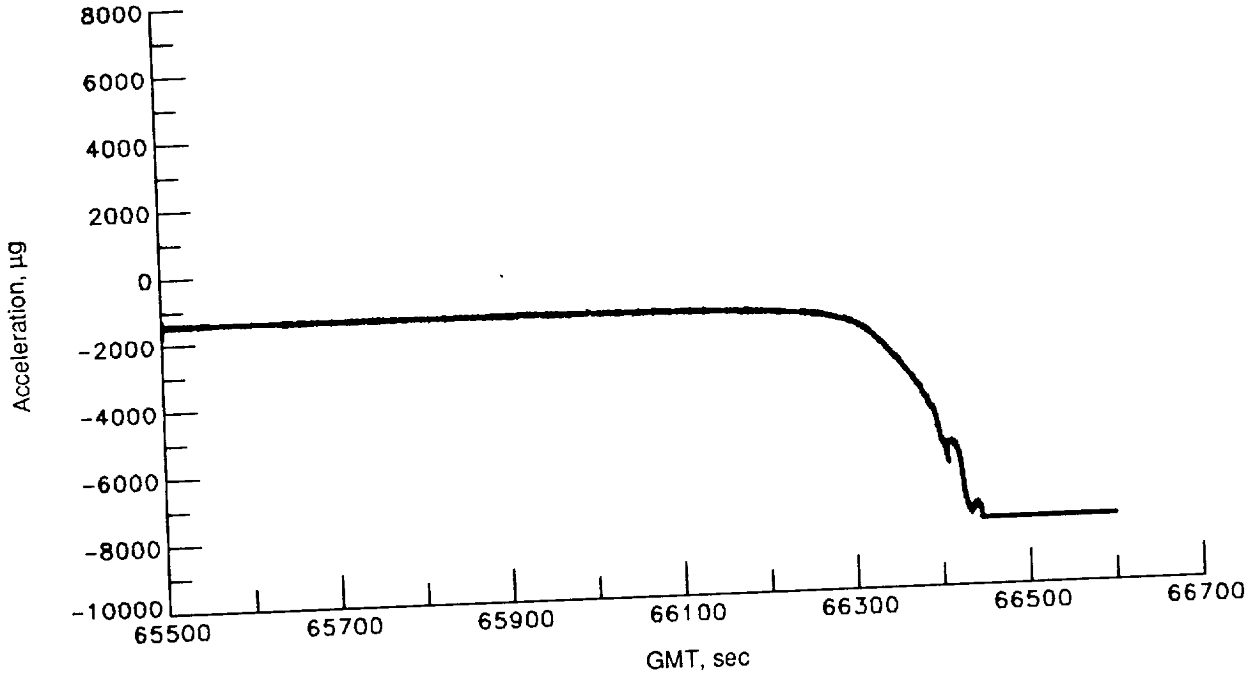
(b) Z-axis acceleration.

Figure 35. Acceleration versus time (no RCS signal) for STS-61C.

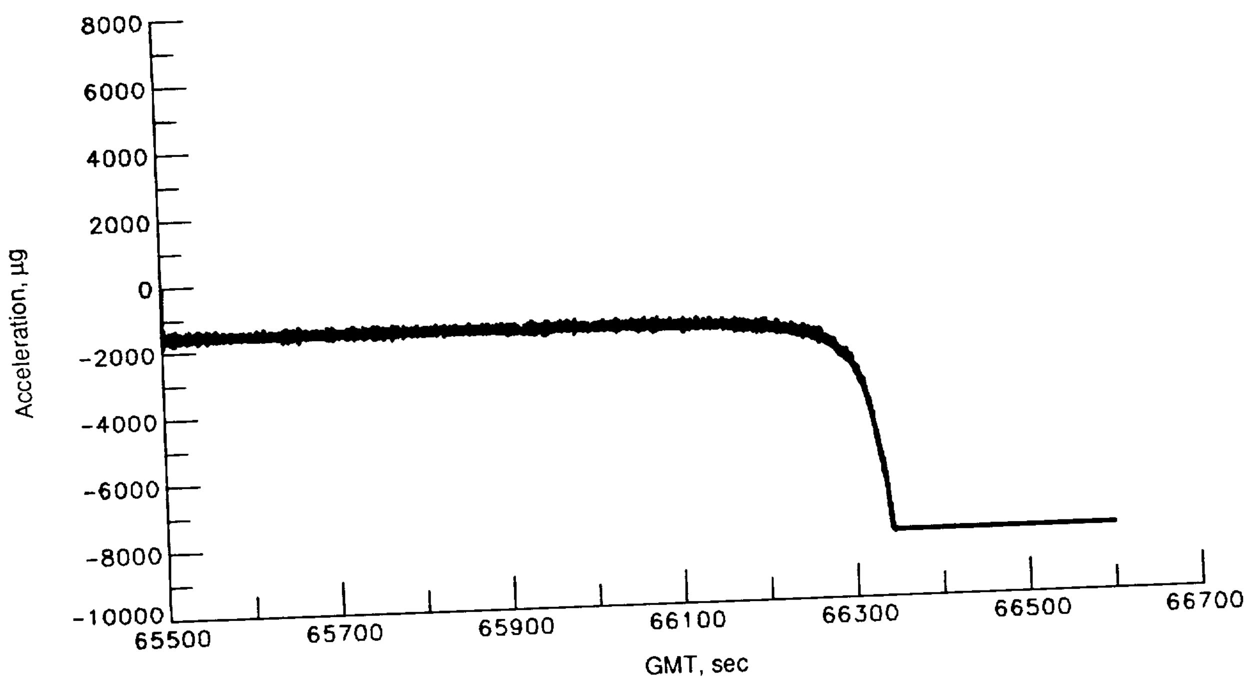


(c) Y-axis acceleration.

Figure 35. Concluded.

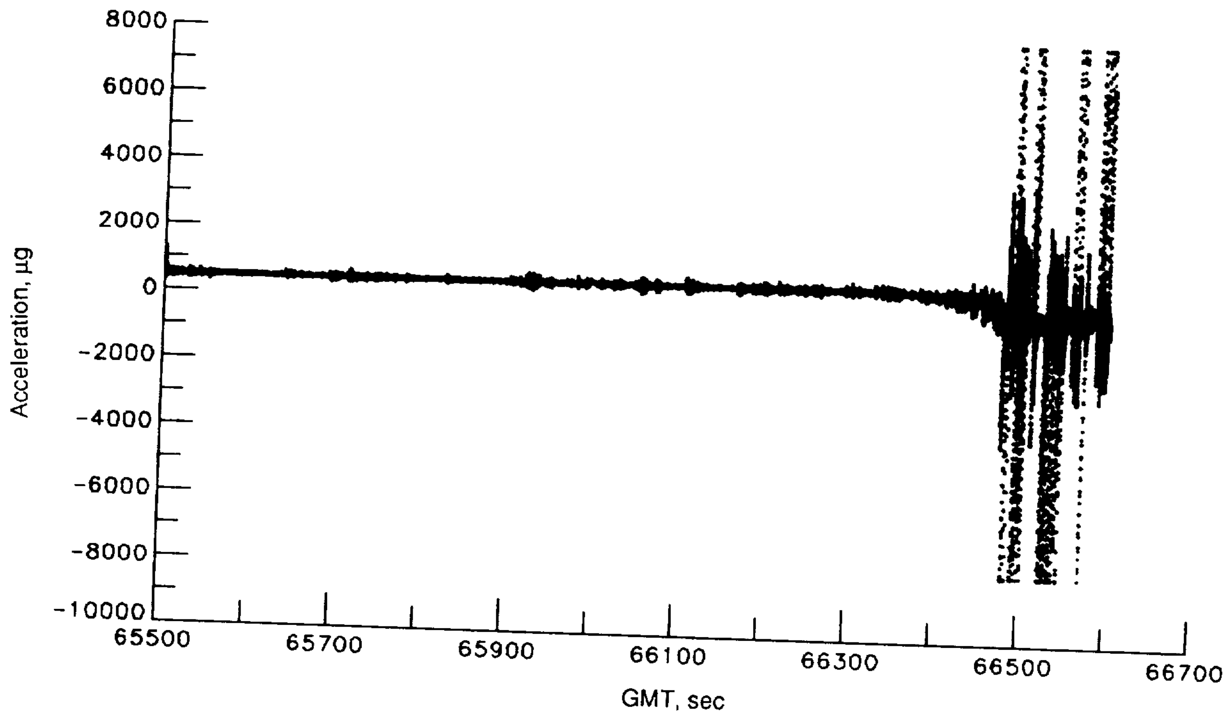


(a) X-axis acceleration.



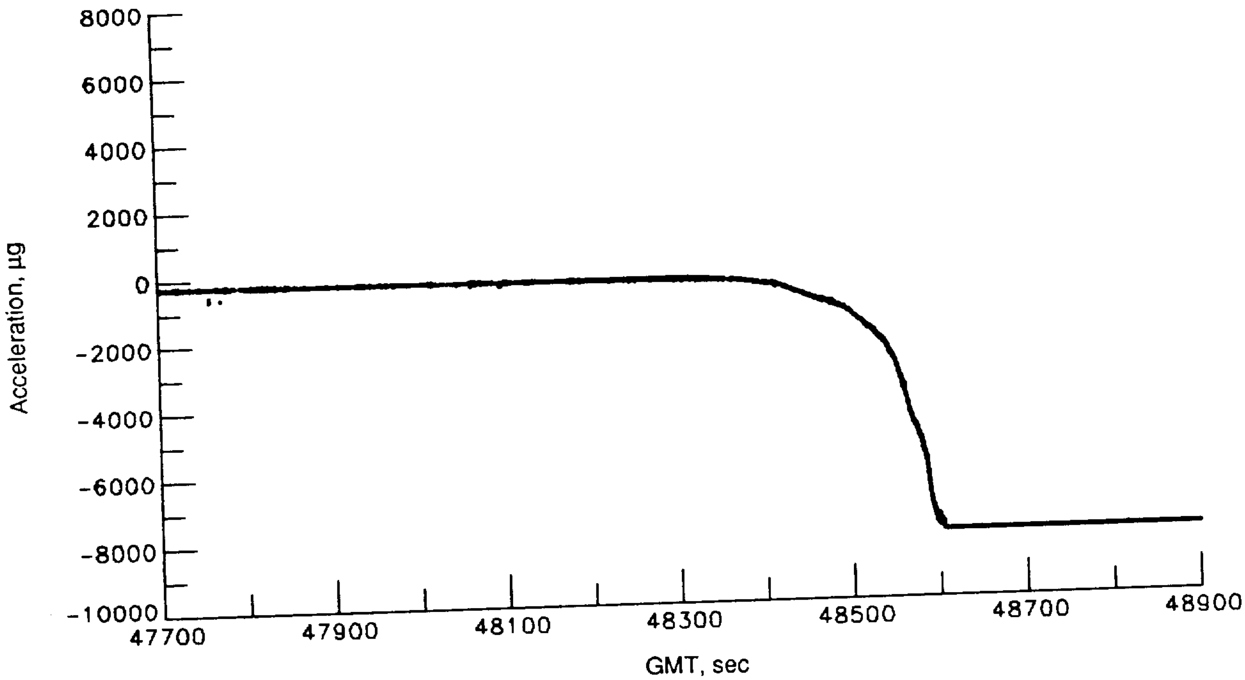
(b) Z-axis acceleration.

Figure 36. Acceleration versus time (after data gap filling) for STS-06.

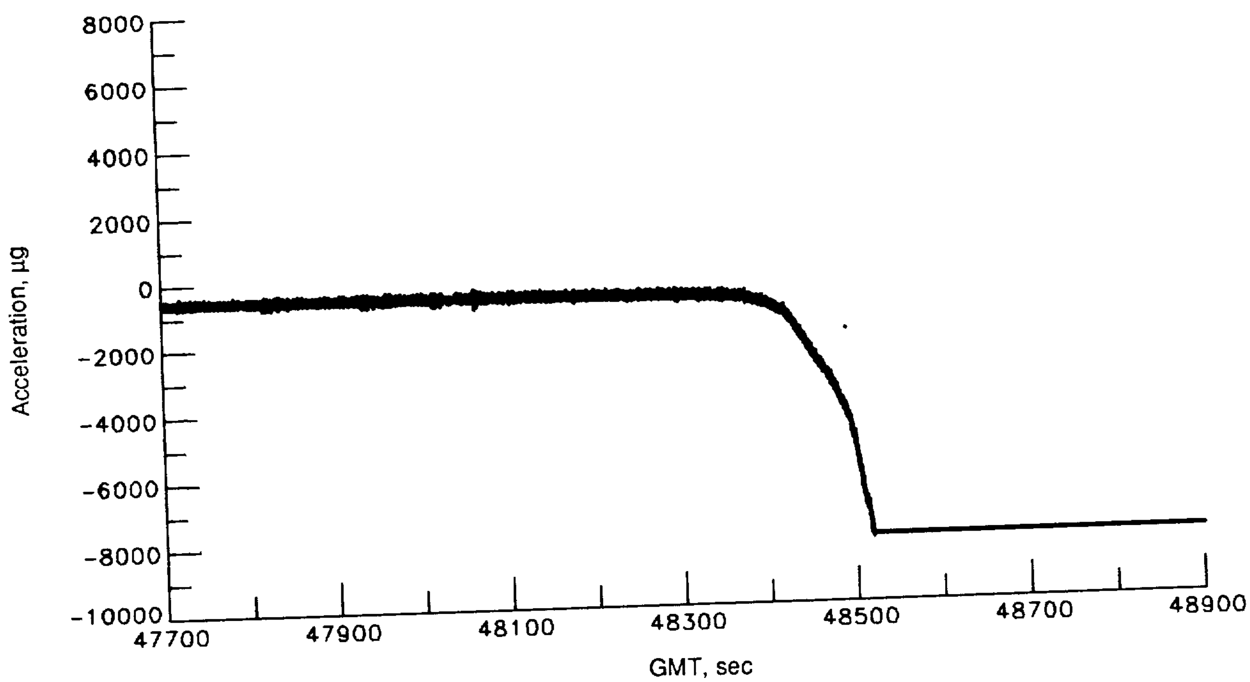


(c) Y-axis acceleration.

Figure 36. Concluded.

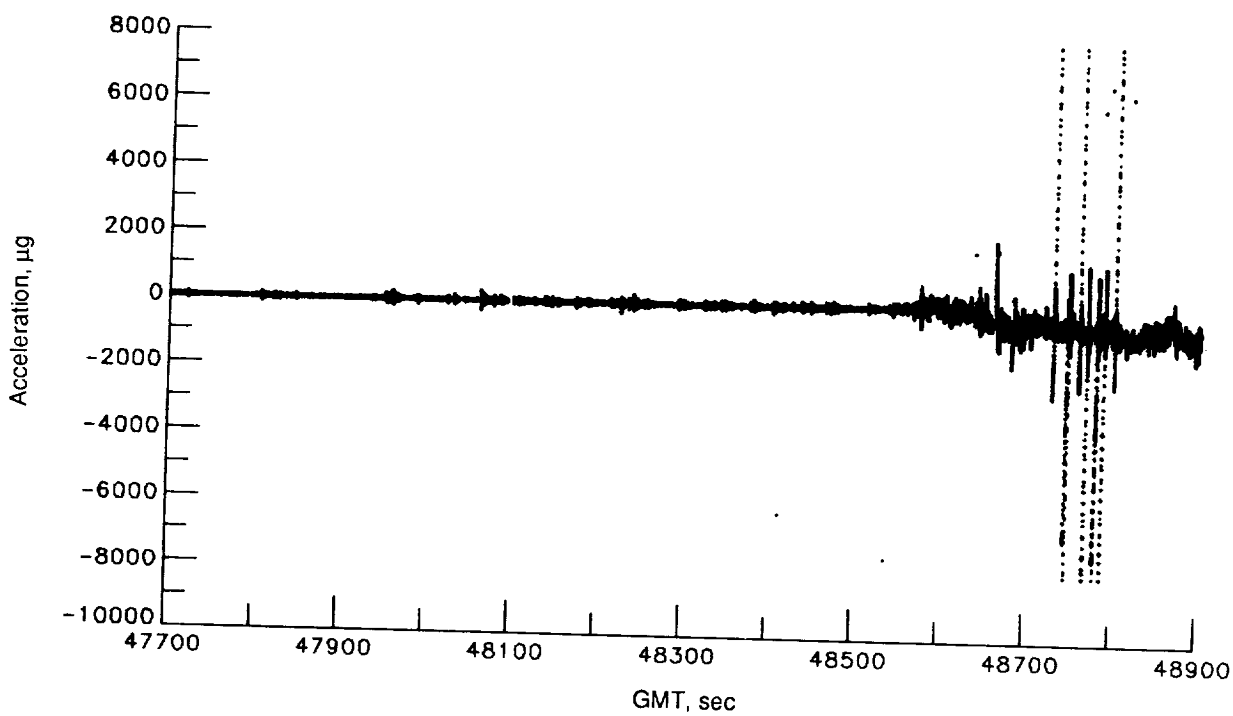


(a) X-axis acceleration.



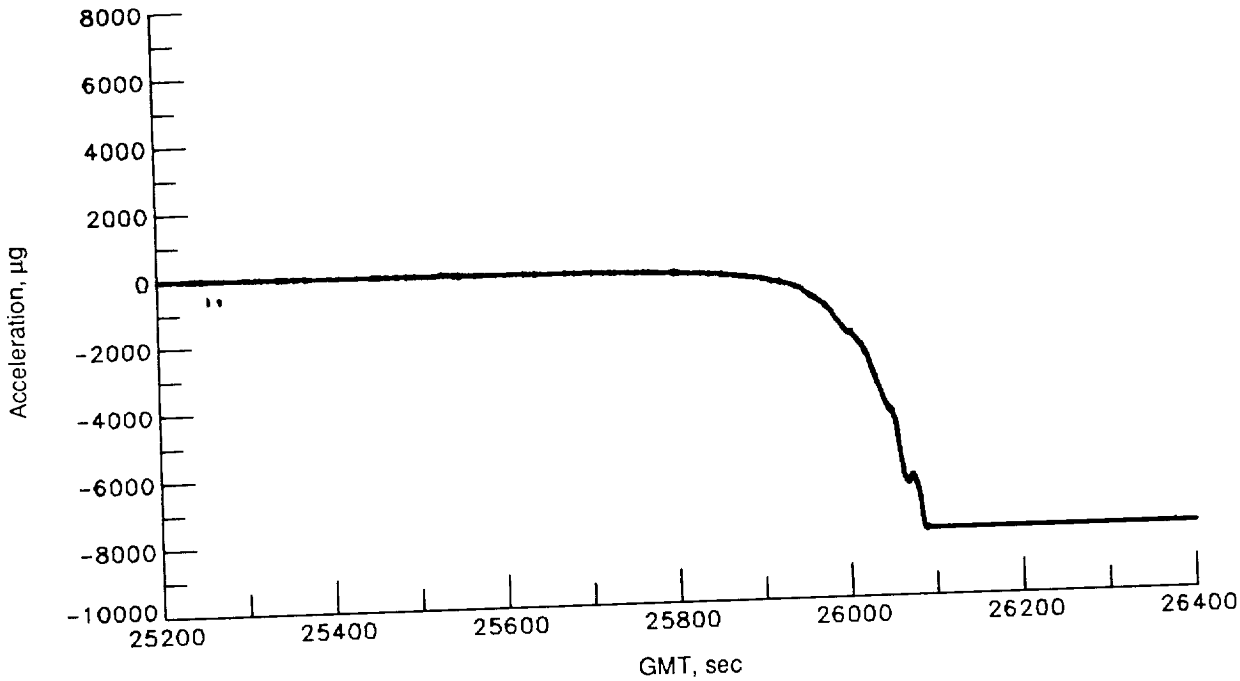
(b) Z-axis acceleration.

Figure 37. Acceleration versus time (after data gap filling) for STS-07.

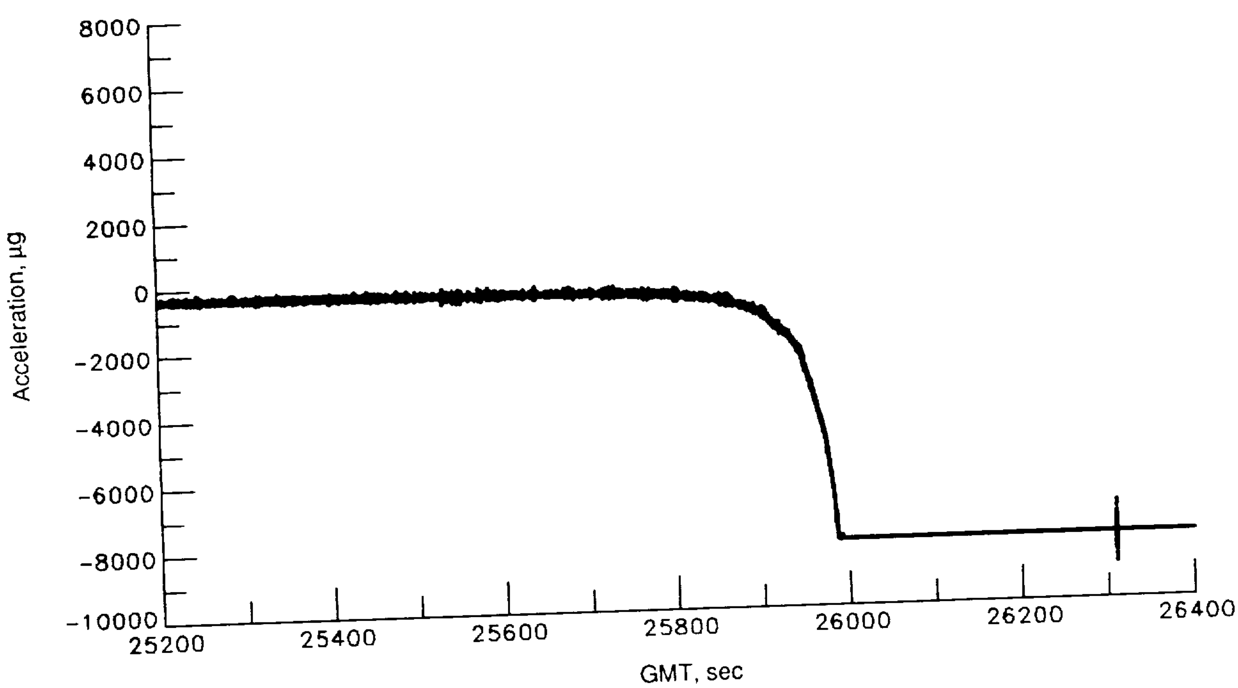


(c) Y-axis acceleration.

Figure 37. Concluded.

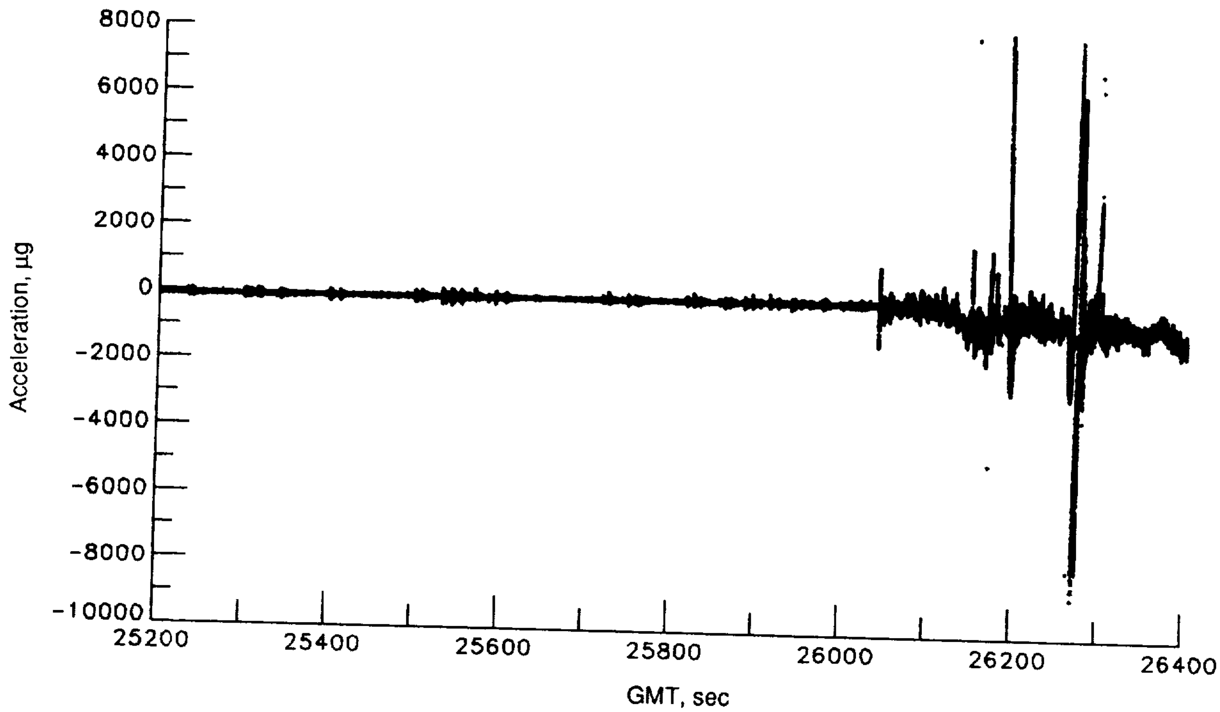


(a) X-axis acceleration.



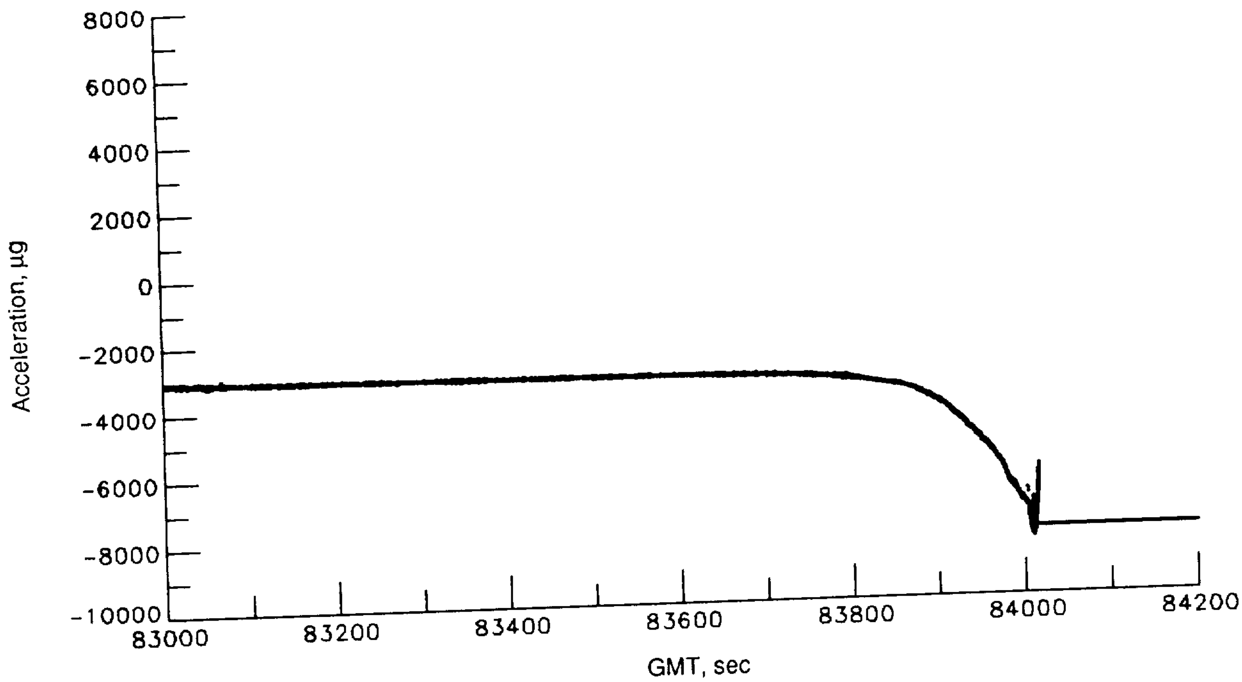
(b) Z-axis acceleration.

Figure 38. Acceleration versus time (after data gap filling) for STS-08.

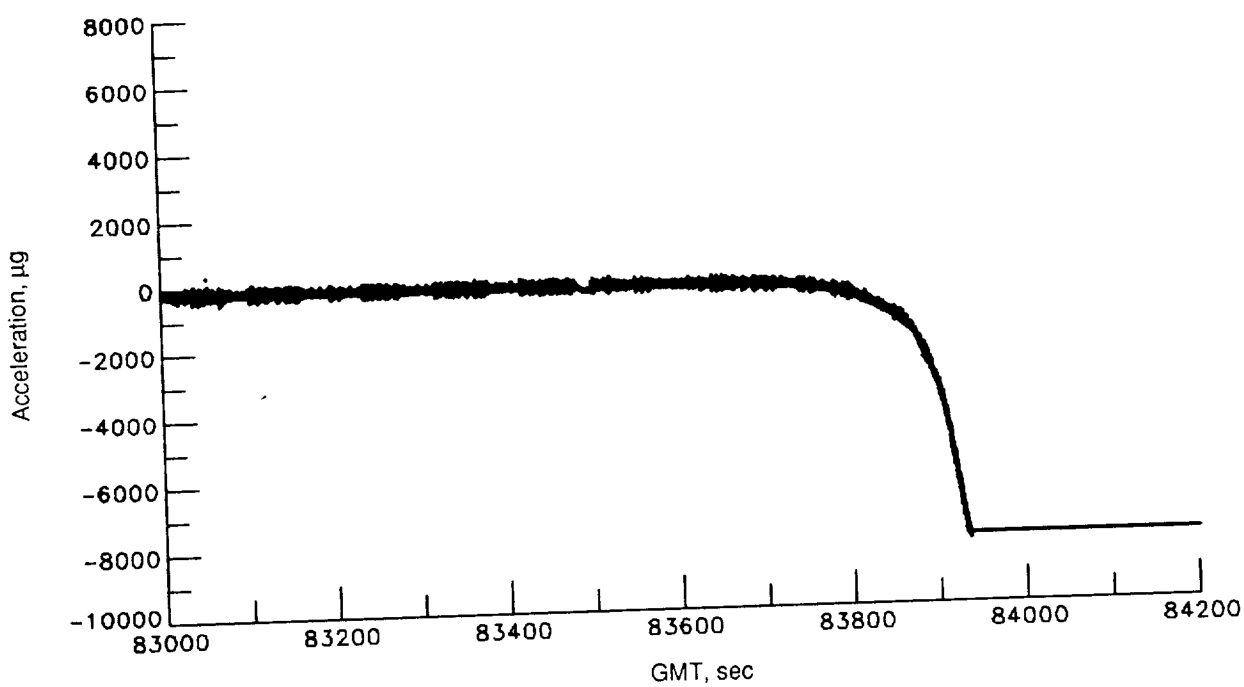


(c) Y-axis acceleration.

Figure 38. Concluded.

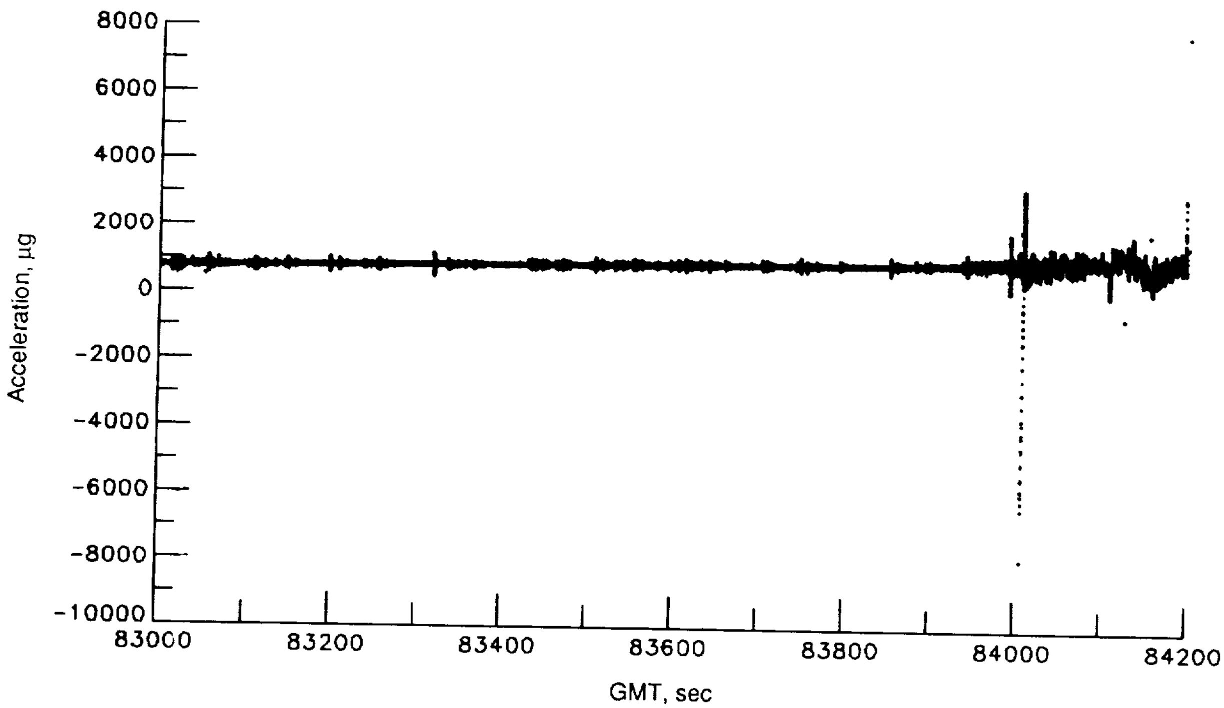


(a) X-axis acceleration.



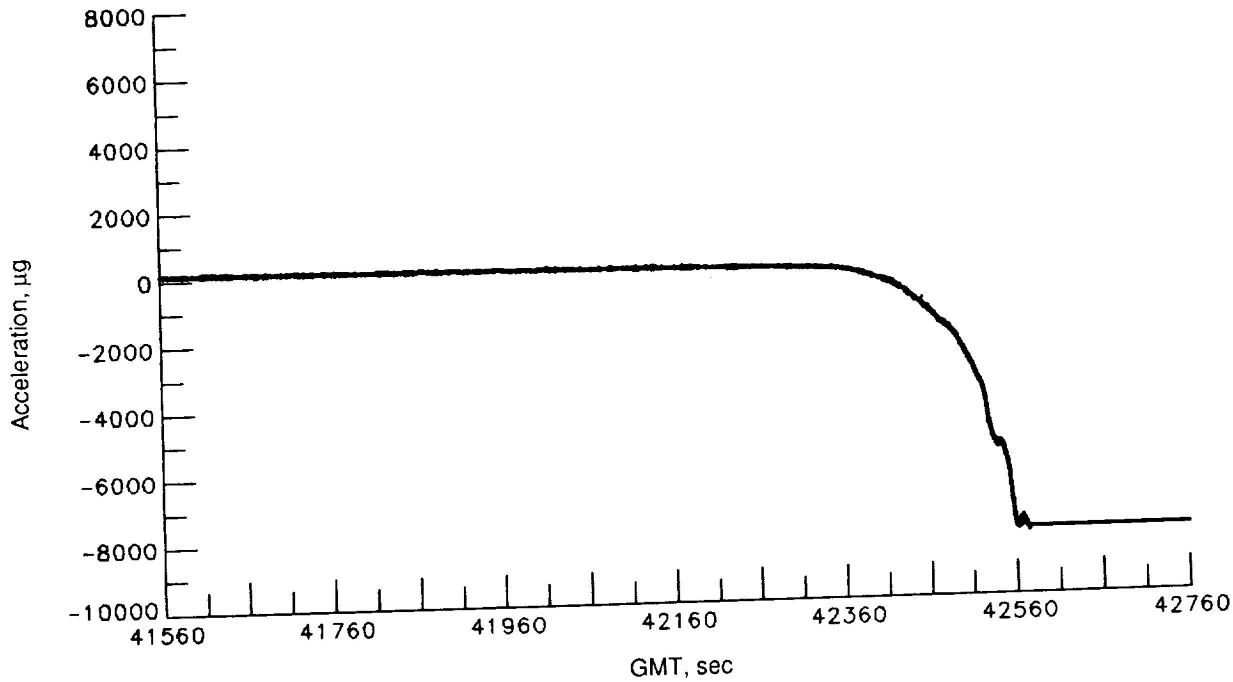
(b) Z-axis acceleration.

Figure 39. Acceleration versus time (after data gap filling) for STS-09.

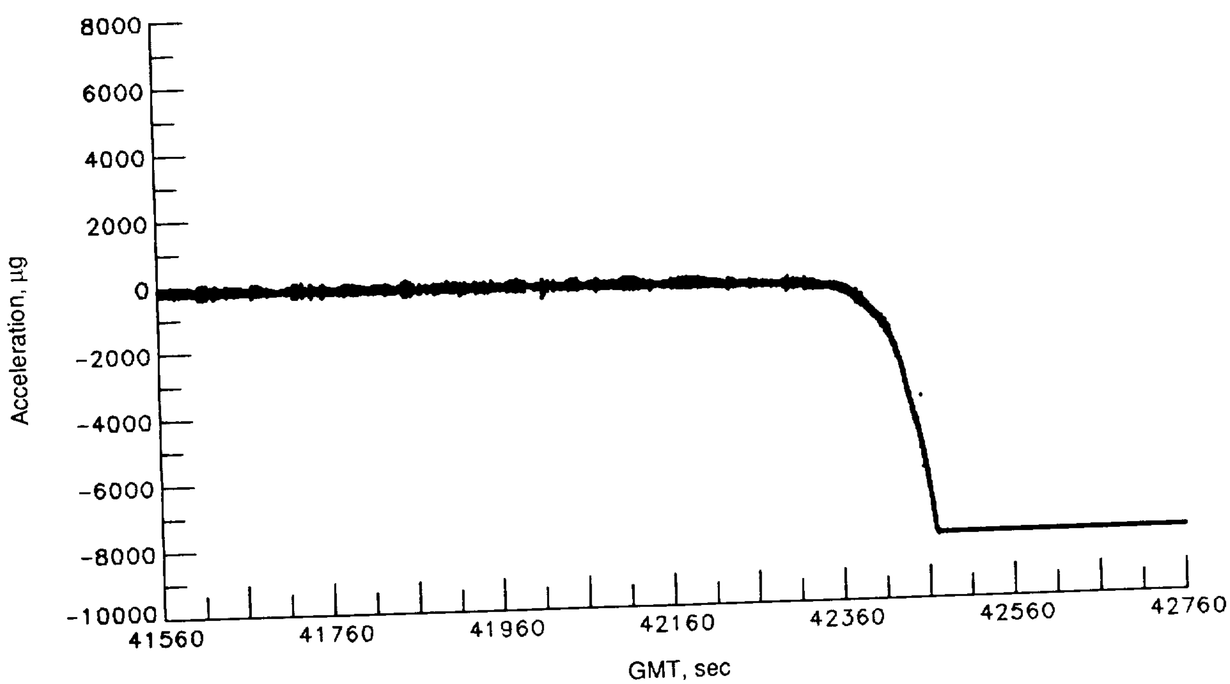


(c) Y-axis acceleration.

Figure 39. Concluded.

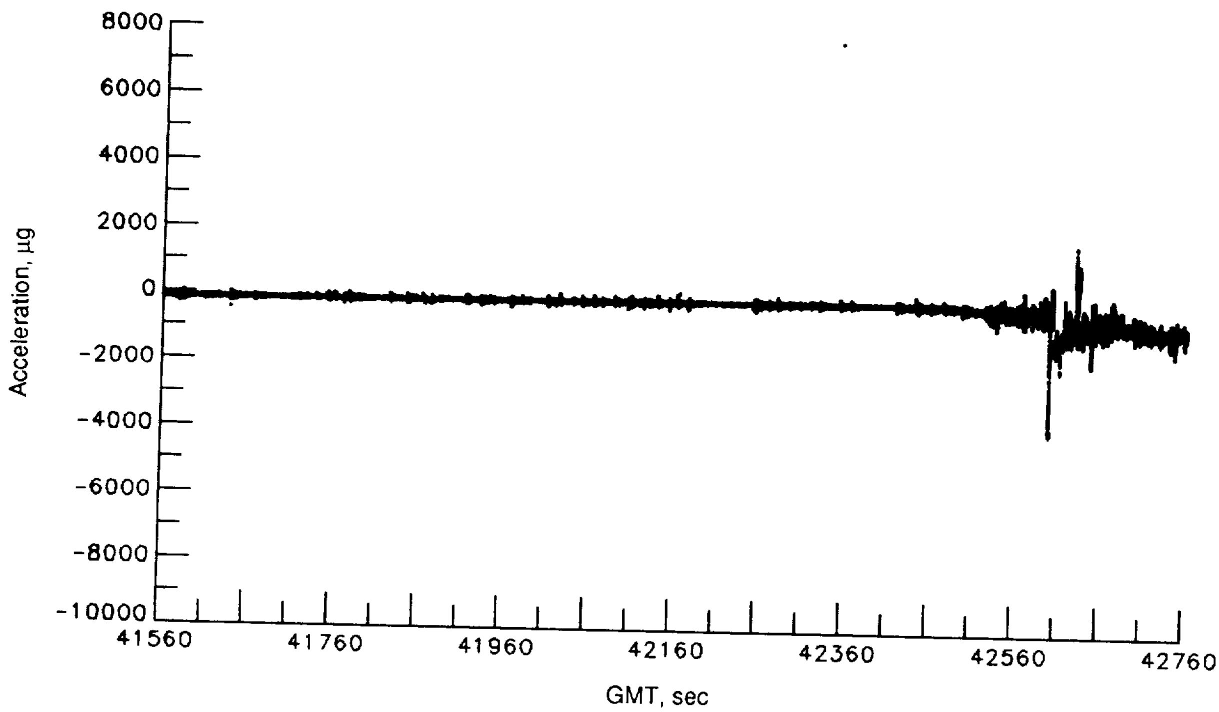


(a) X-axis acceleration.



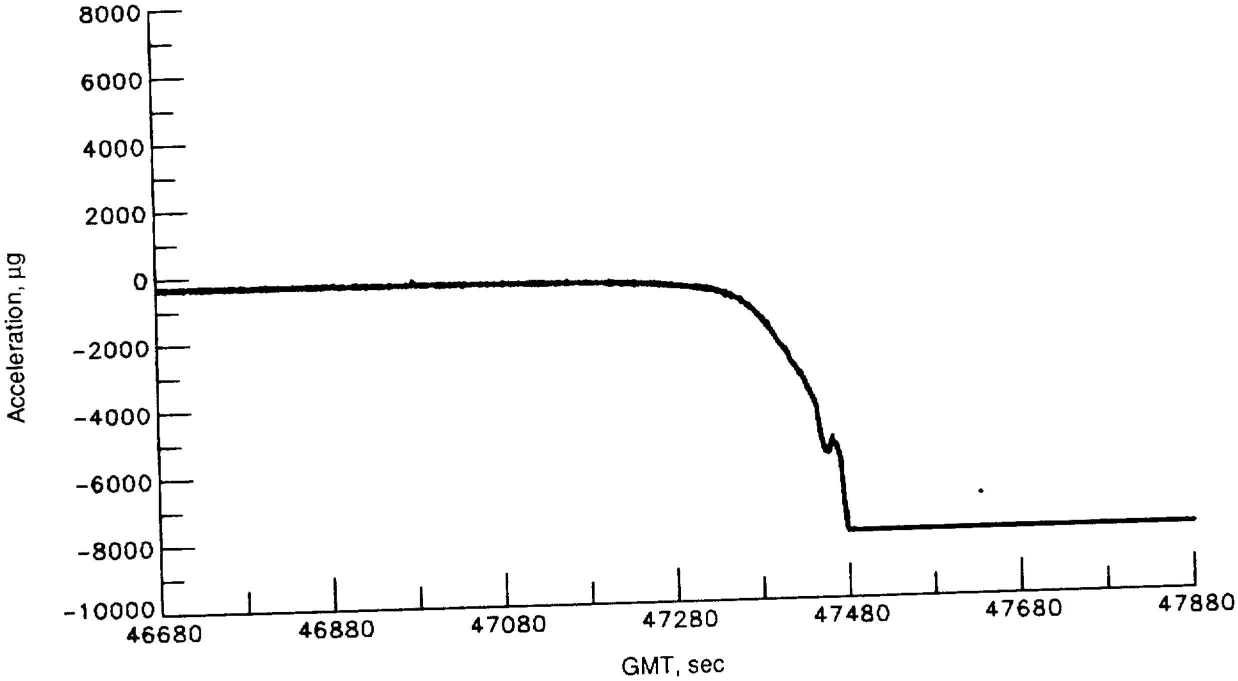
(b) Z-axis acceleration.

Figure 40. Acceleration versus time (after data gap filling) for STS-41B.

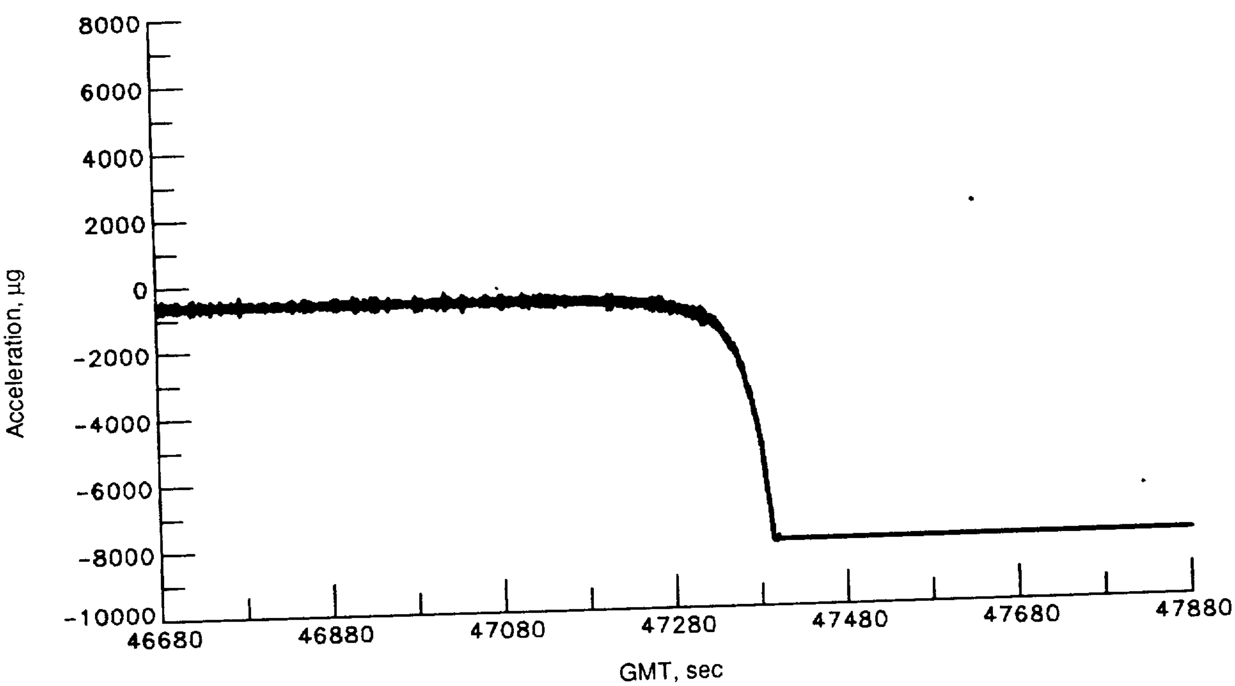


(c) Y-axis acceleration.

Figure 40. Concluded.

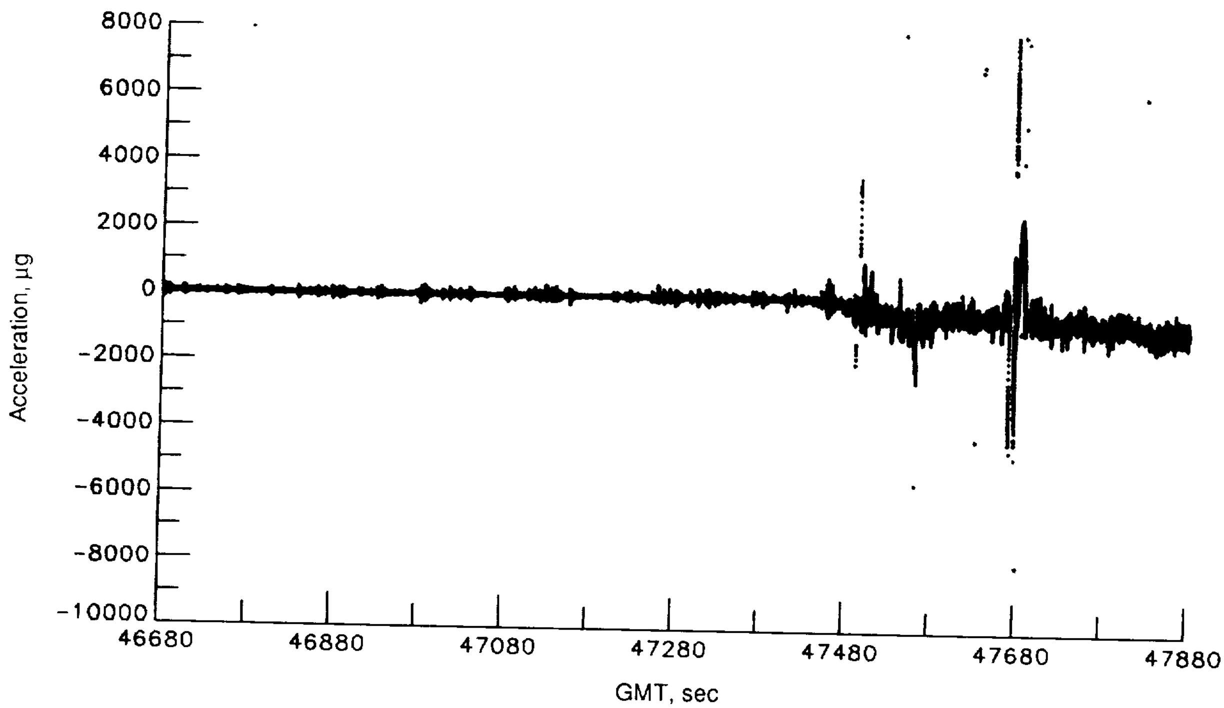


(a) X-axis acceleration.



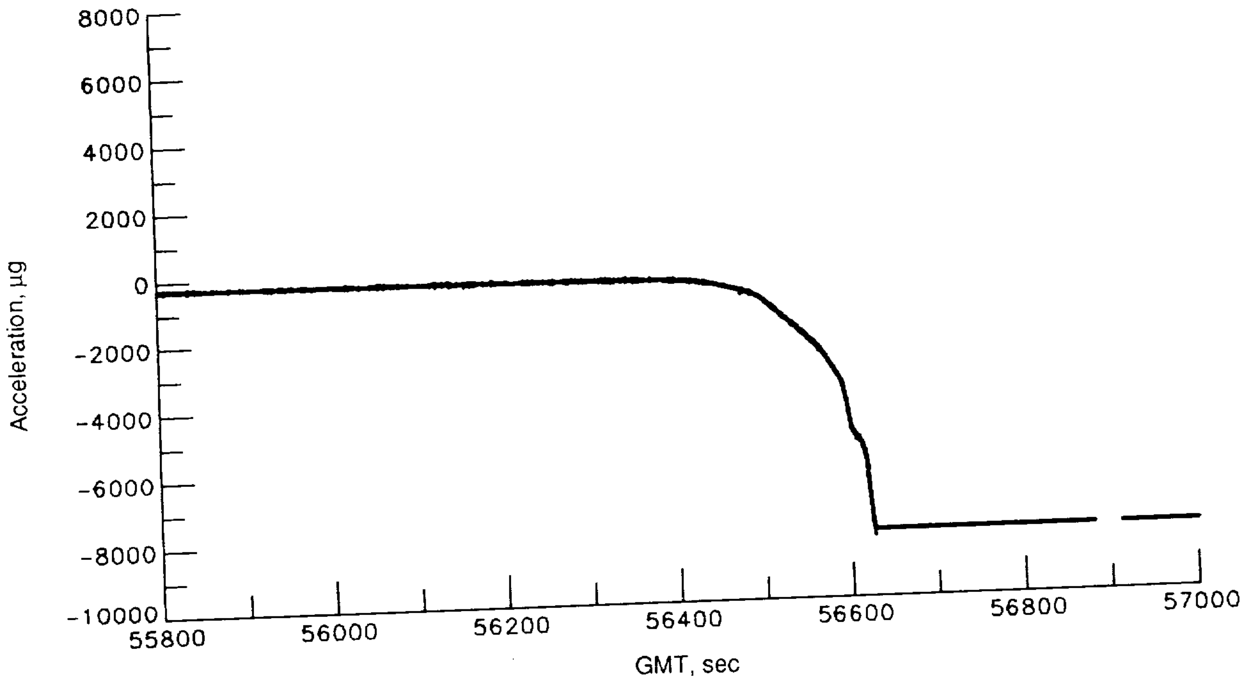
(b) Z-axis acceleration.

Figure 41. Acceleration versus time (after data gap filling) for STS-41C.

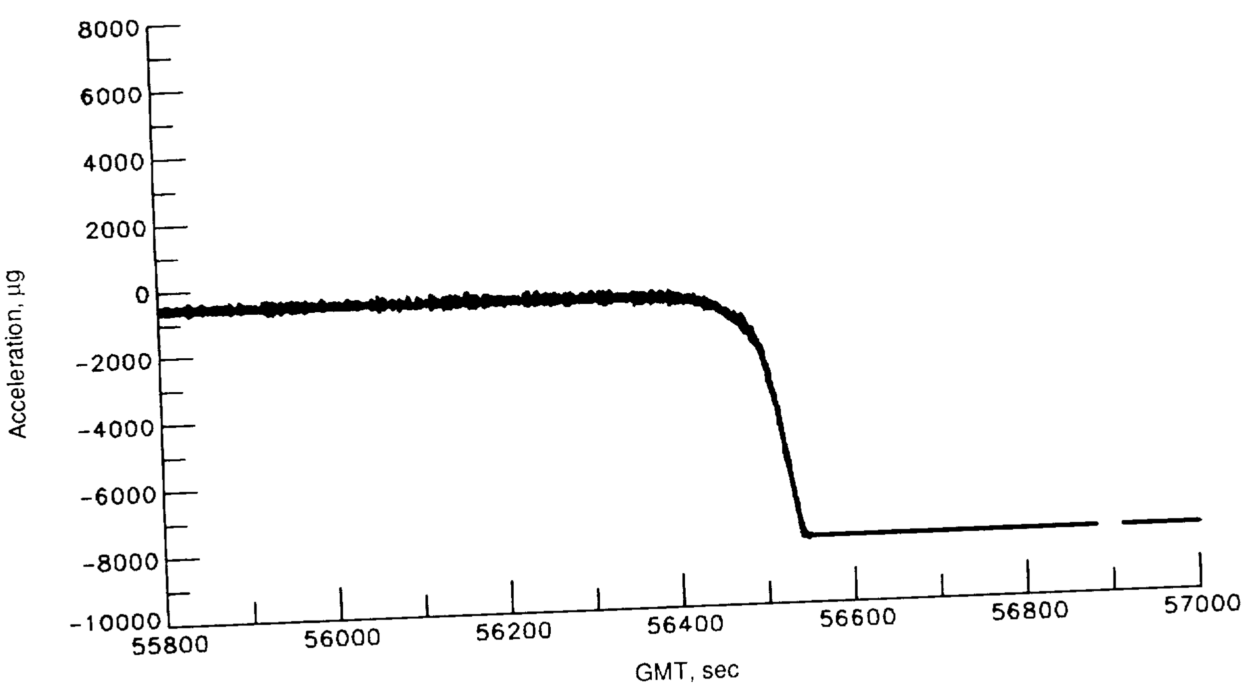


(c) Y-axis acceleration.

Figure 41. Concluded.

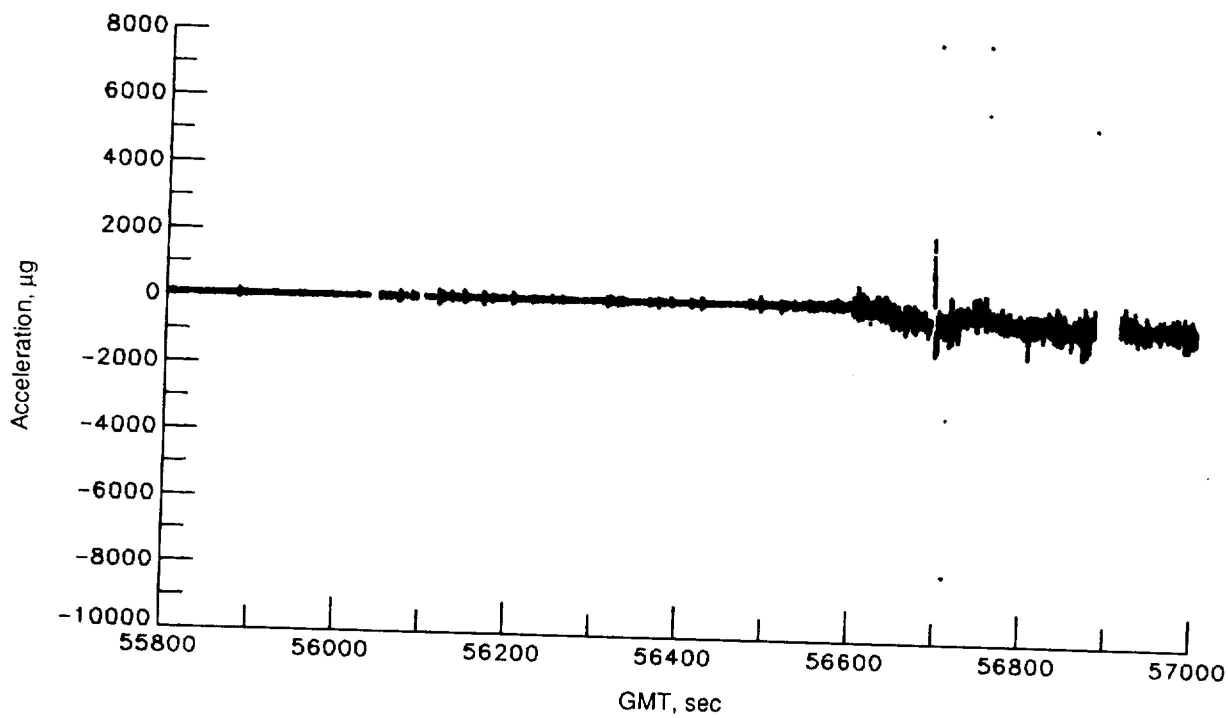


(a) X-axis acceleration.



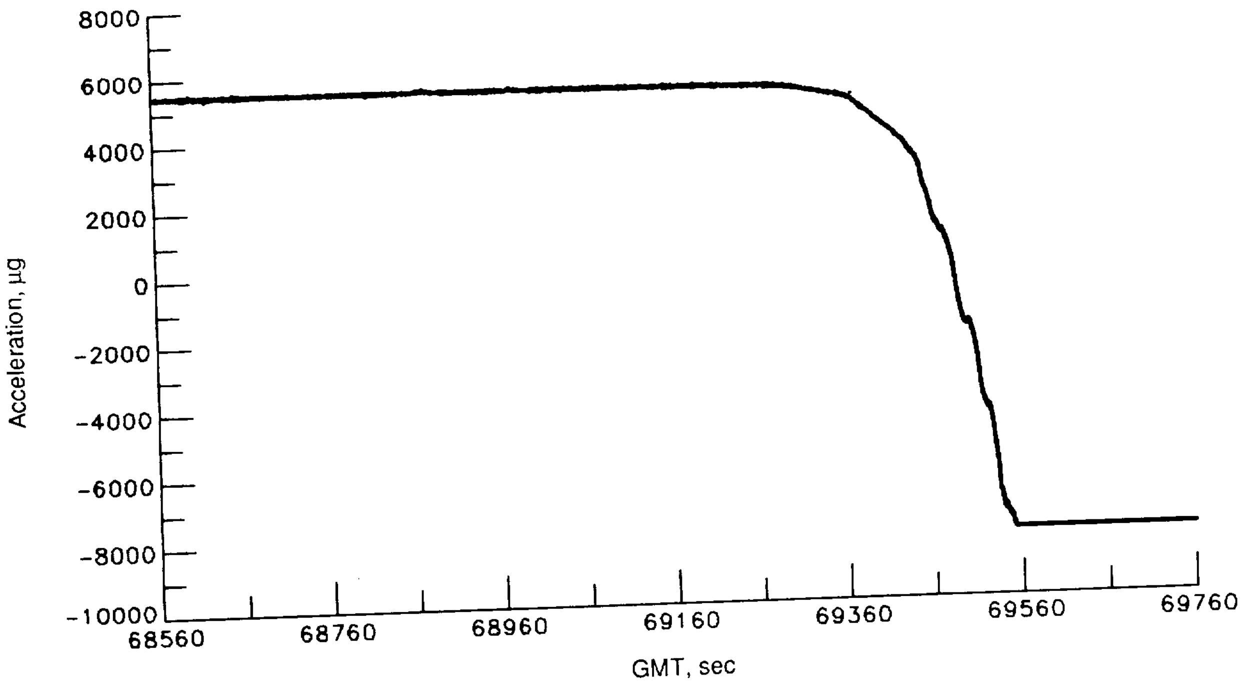
(b) Z-axis acceleration.

Figure 42. Acceleration versus time (after data gap filling) for STS-51B.

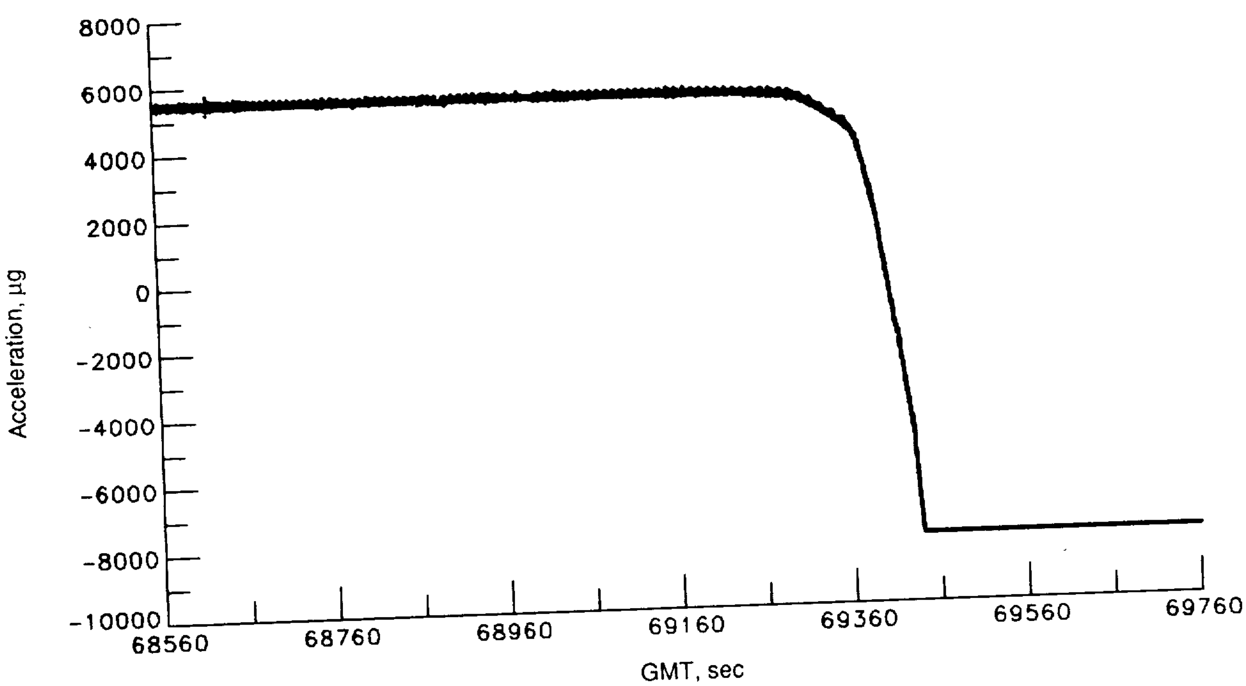


(c) Y-axis acceleration.

Figure 42. Concluded.

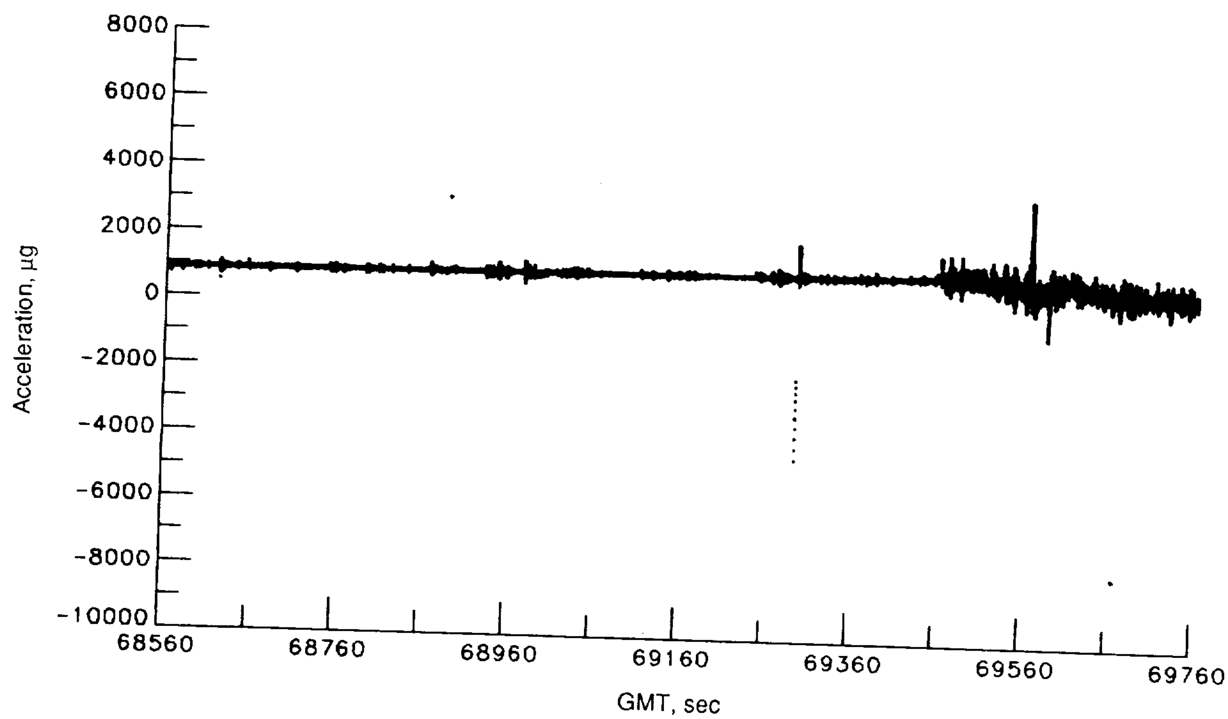


(a) X-axis acceleration.



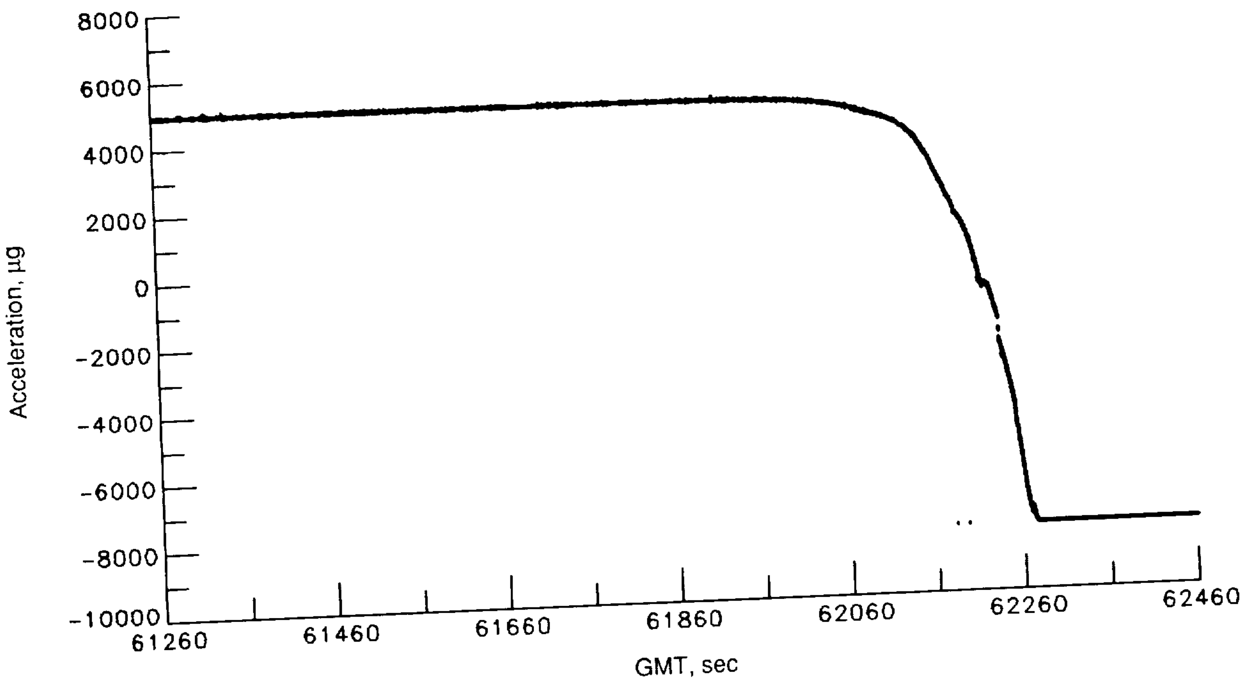
(b) Z-axis acceleration.

Figure 43. Acceleration versus time (after data gap filling) for STS-51F.

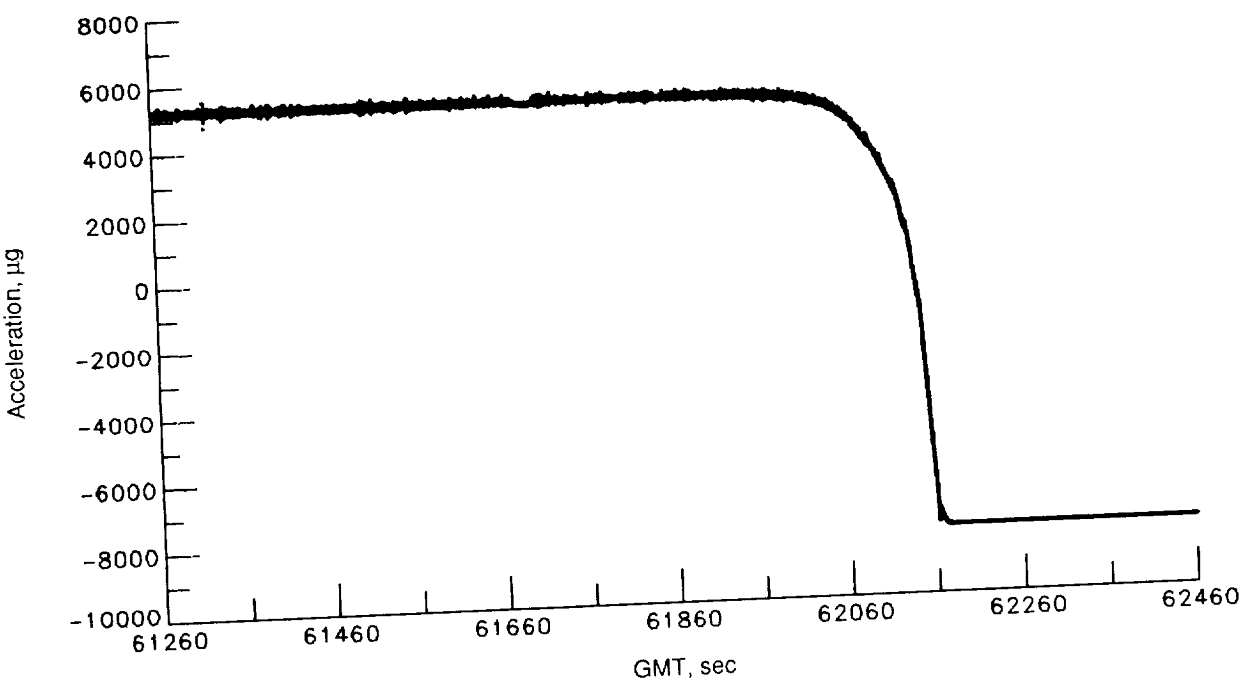


(c) Y-axis acceleration.

Figure 43. Concluded.

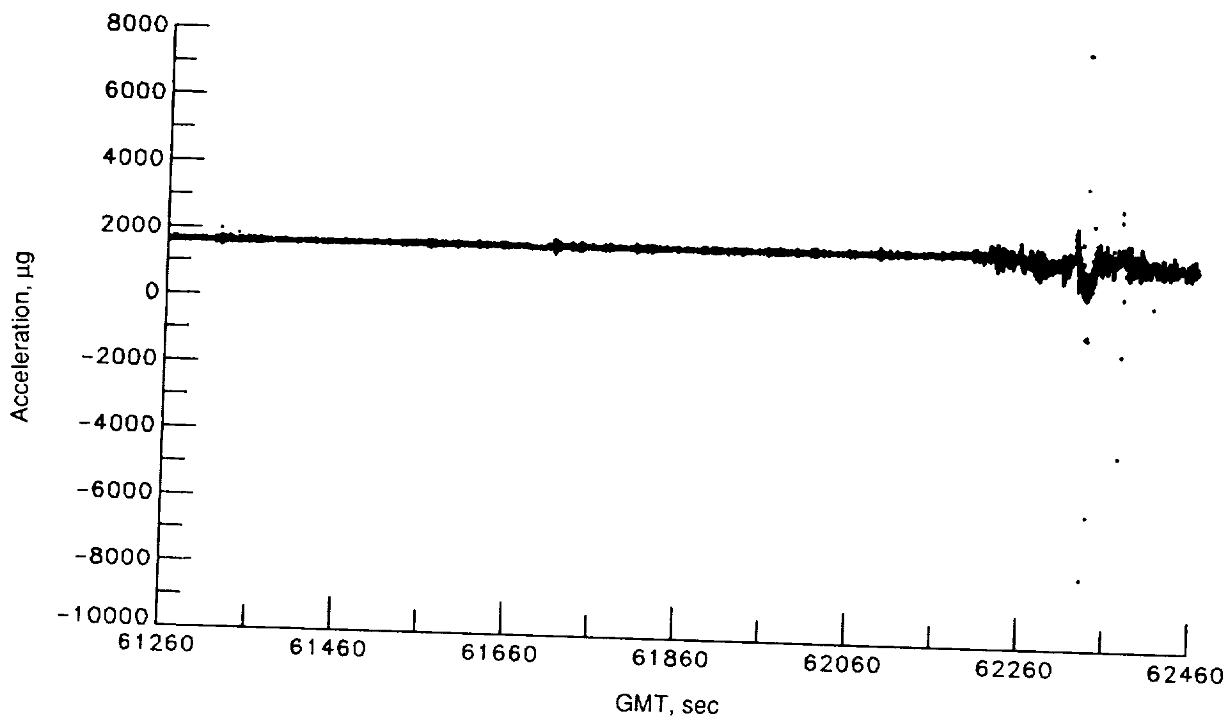


(a) X-axis acceleration.



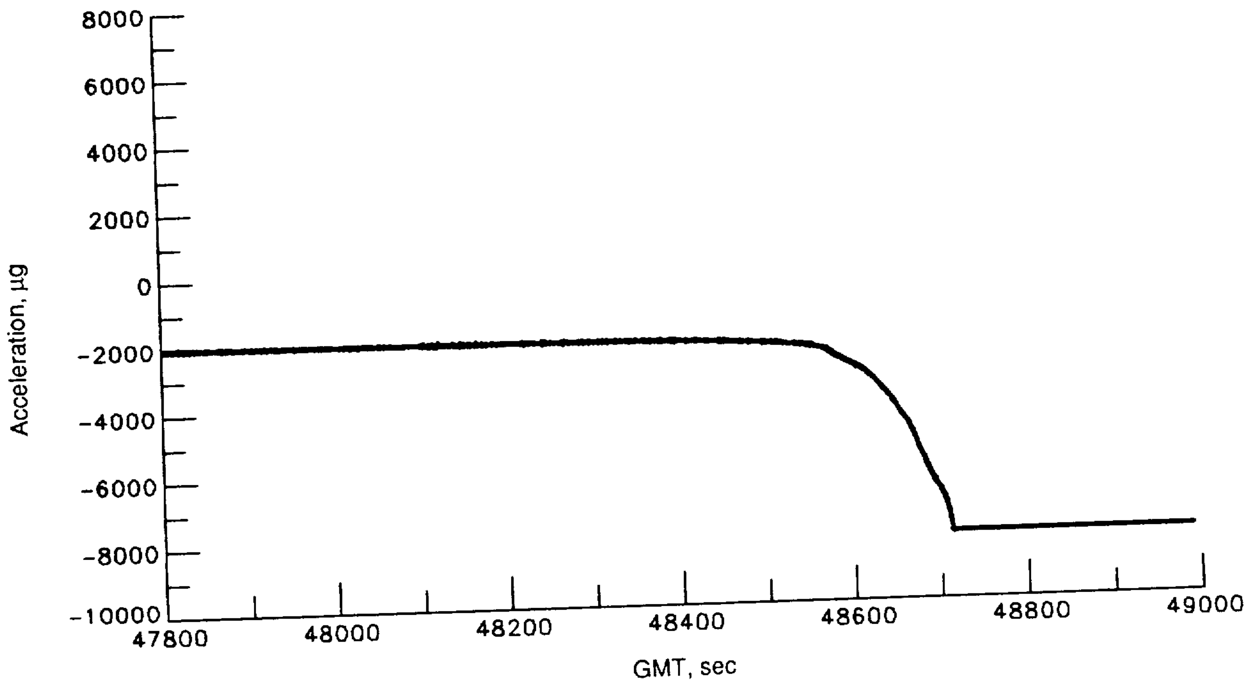
(b) Z-axis acceleration.

Figure 44. Acceleration versus time (after data gap filling) for STS-61A.

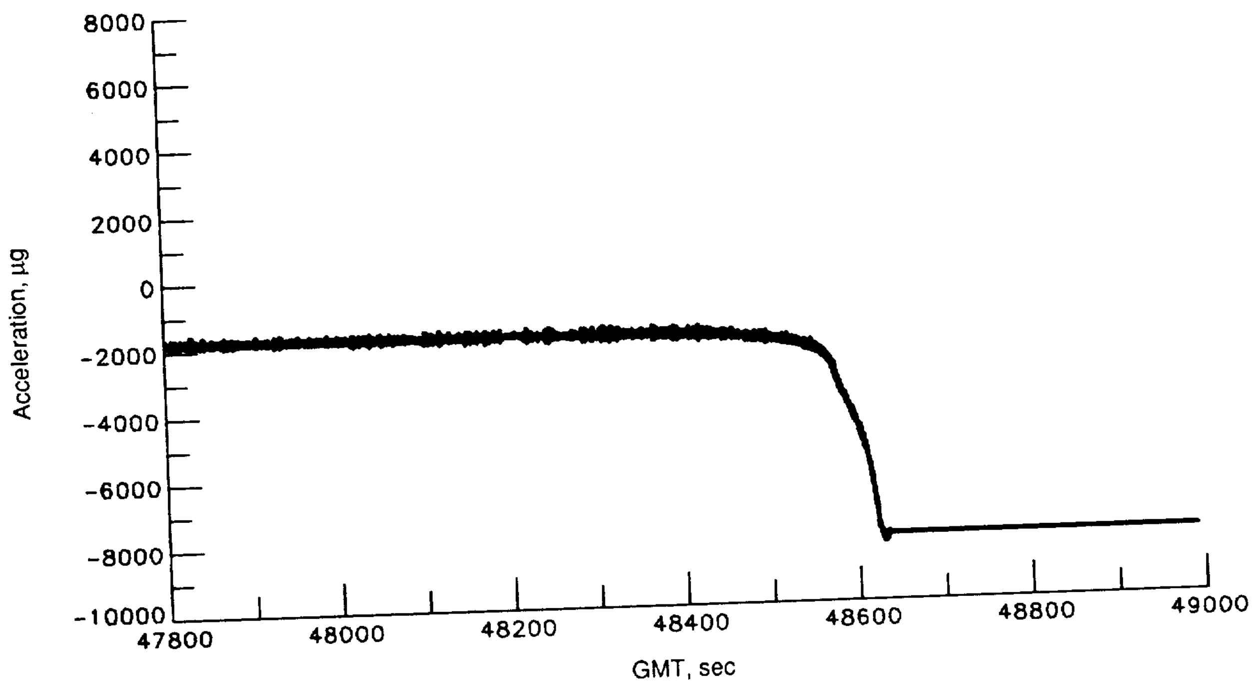


(c) Y-axis acceleration.

Figure 44. Concluded.

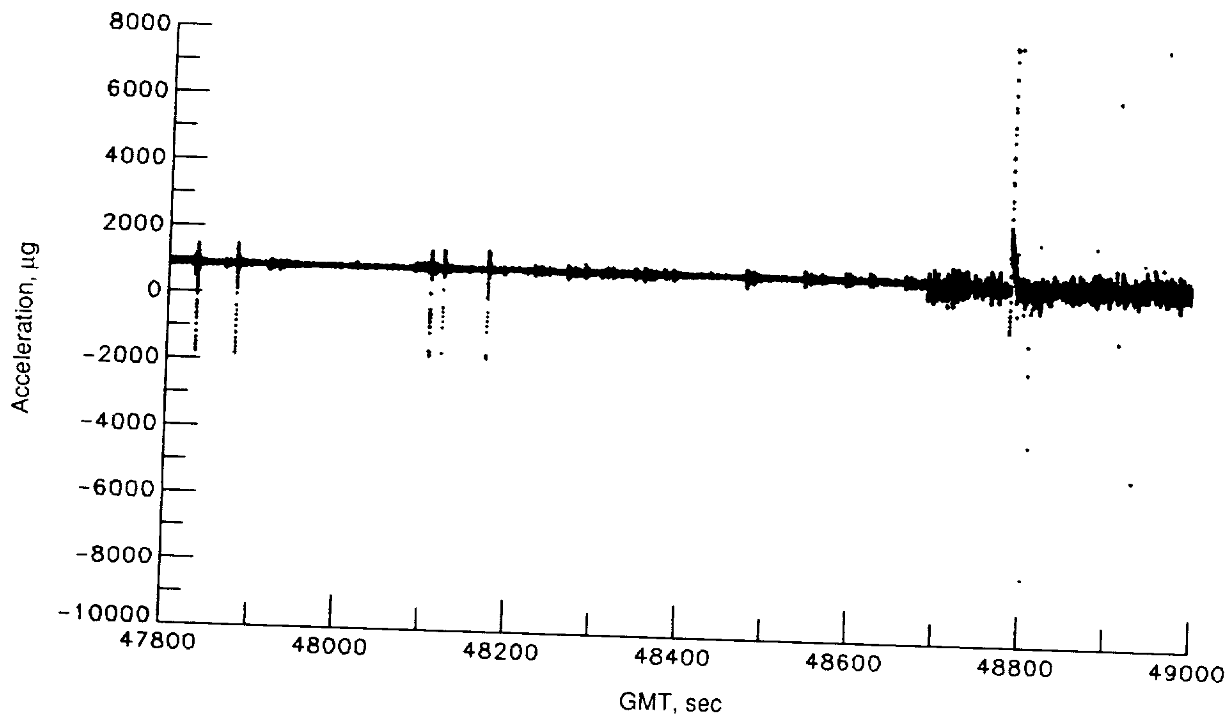


(a) X-axis acceleration.



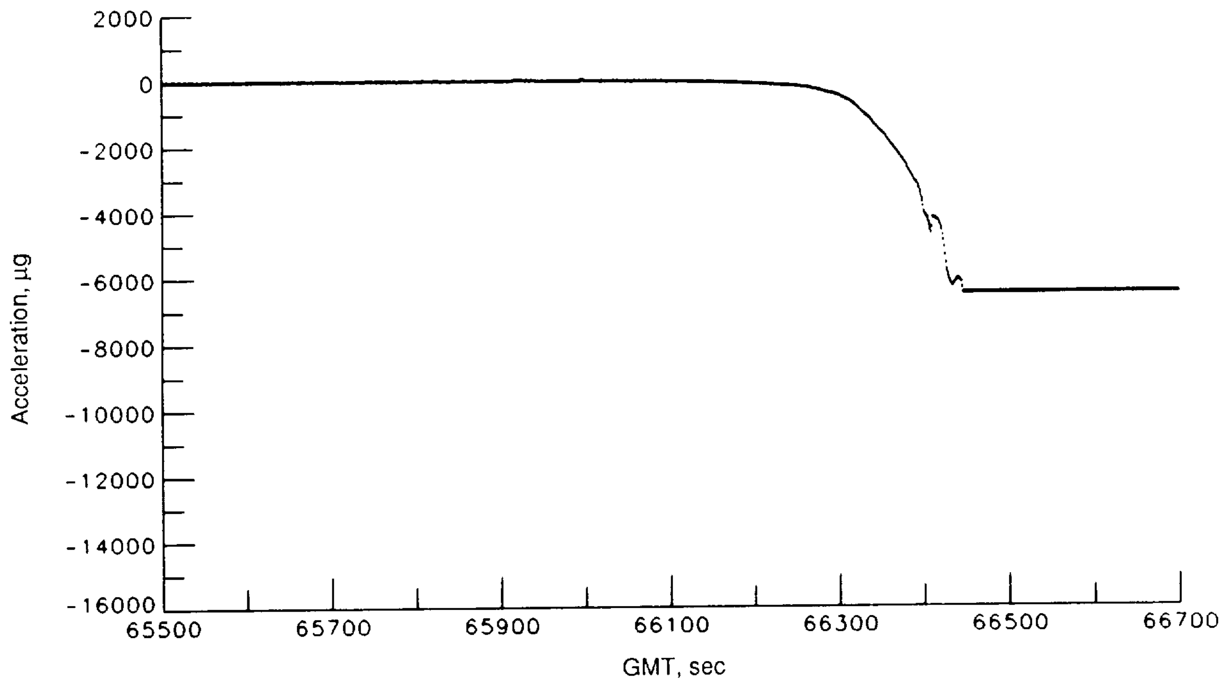
(b) Z-axis acceleration.

Figure 45. Acceleration versus time (after data gap filling) for STS-61C.

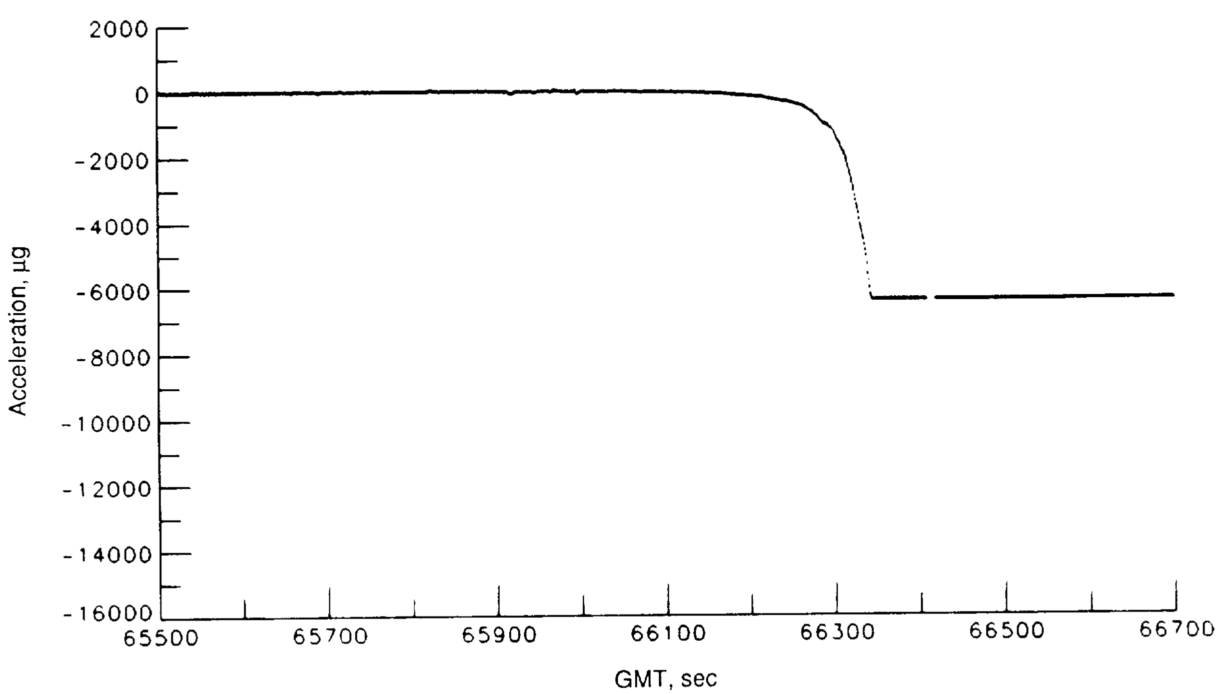


(c) Y-axis acceleration.

Figure 45. Concluded.

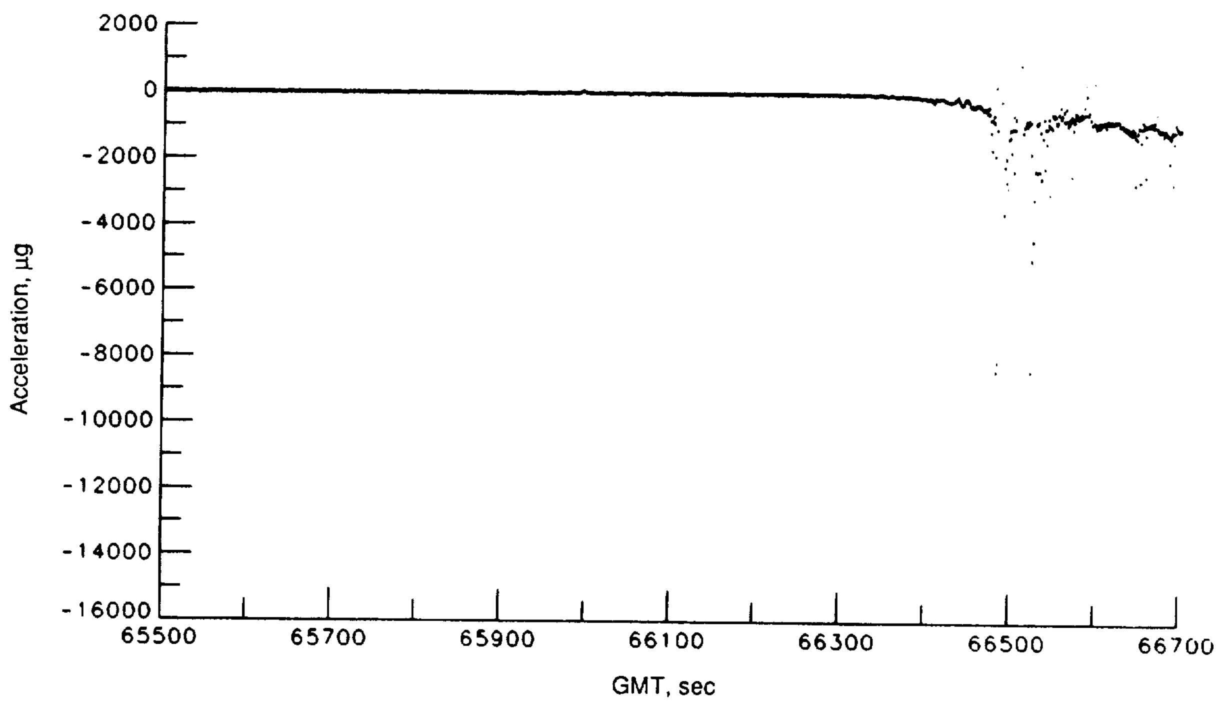


(a) X-axis acceleration.



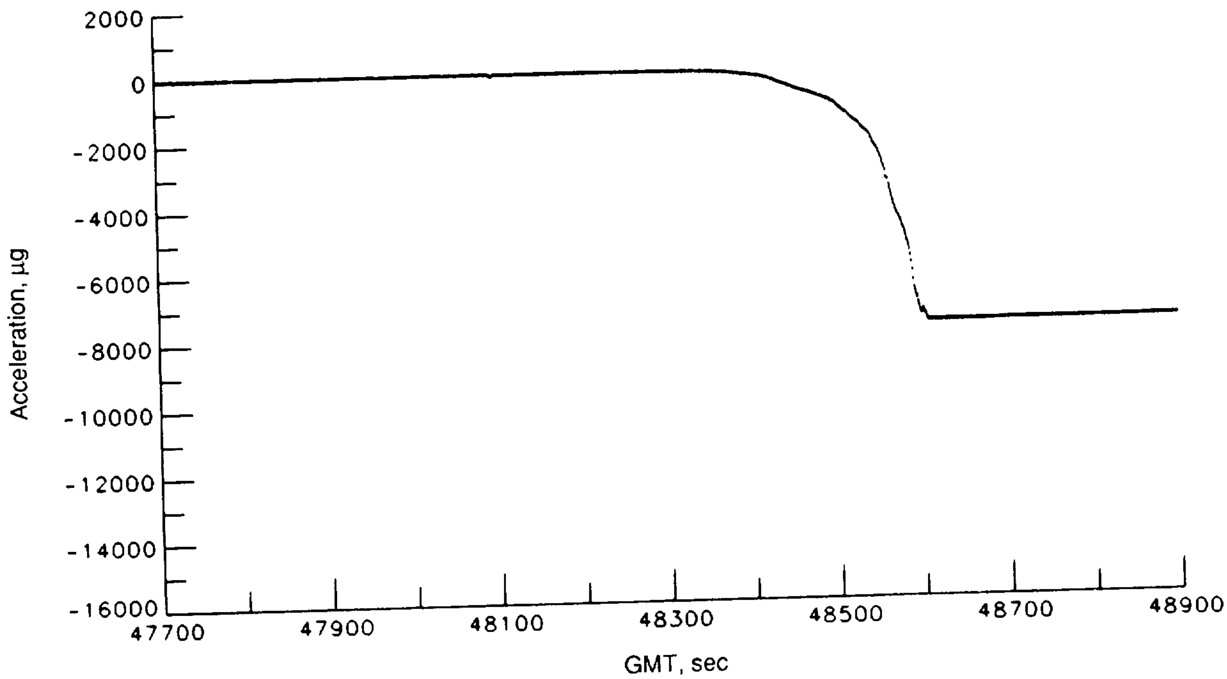
(b) Z-axis acceleration.

Figure 46. One-second averaged aerodynamic acceleration data versus time for STS-06.

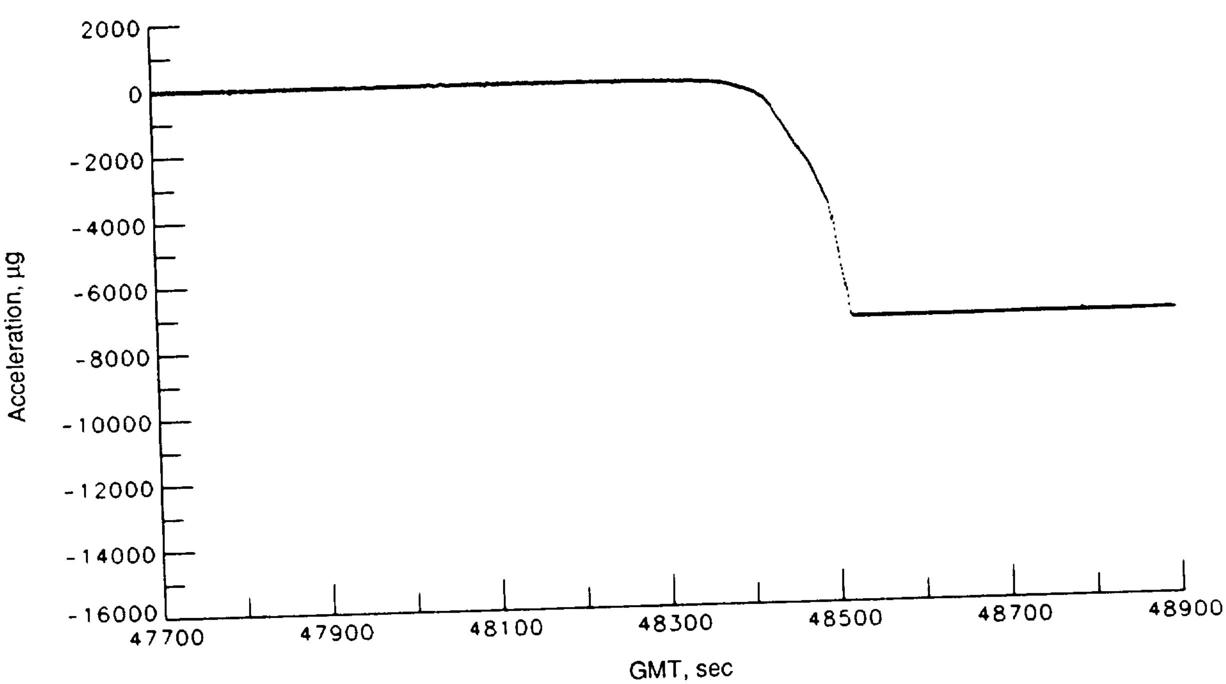


(c) Y-axis acceleration.

Figure 46. Concluded.

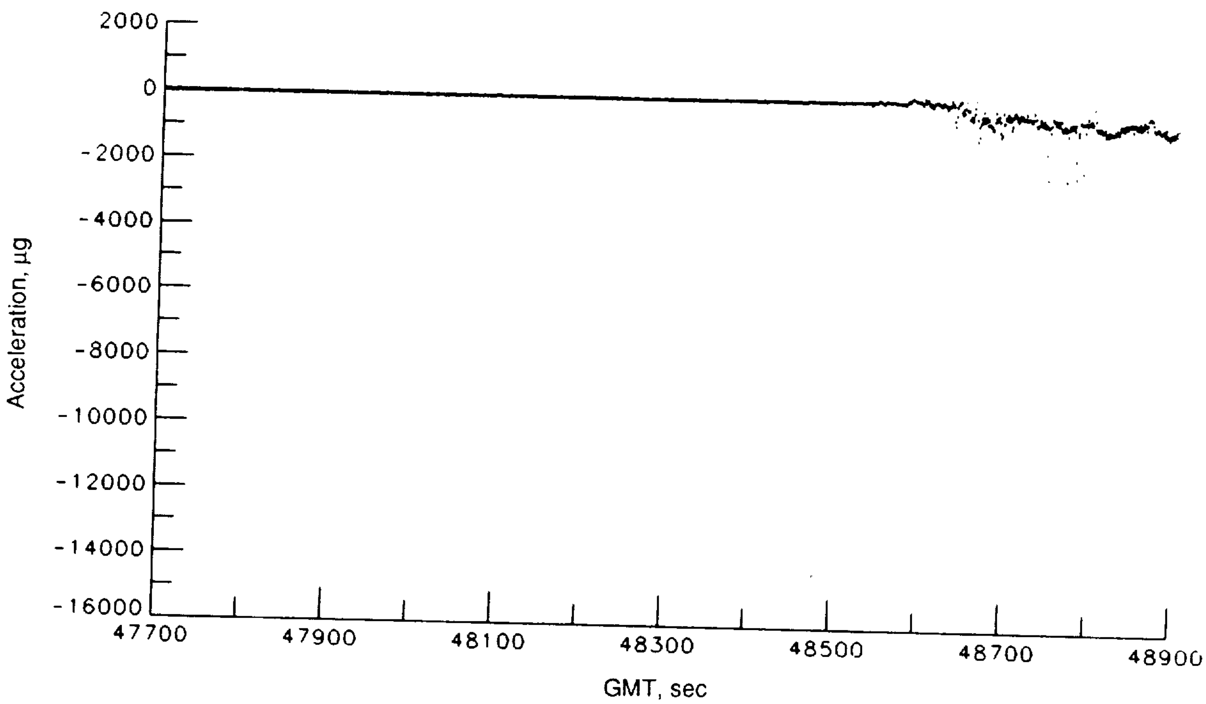


(a) X-axis acceleration.



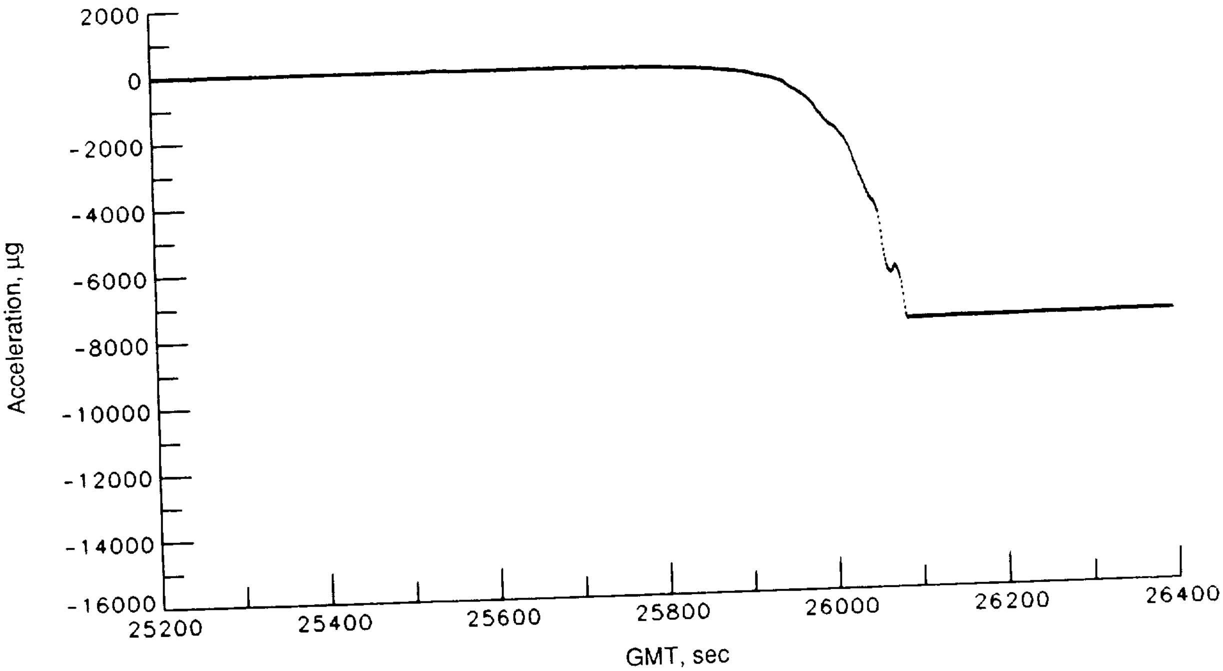
(b) Z-axis acceleration.

Figure 47. One-second averaged aerodynamic acceleration data versus time for STS-07.

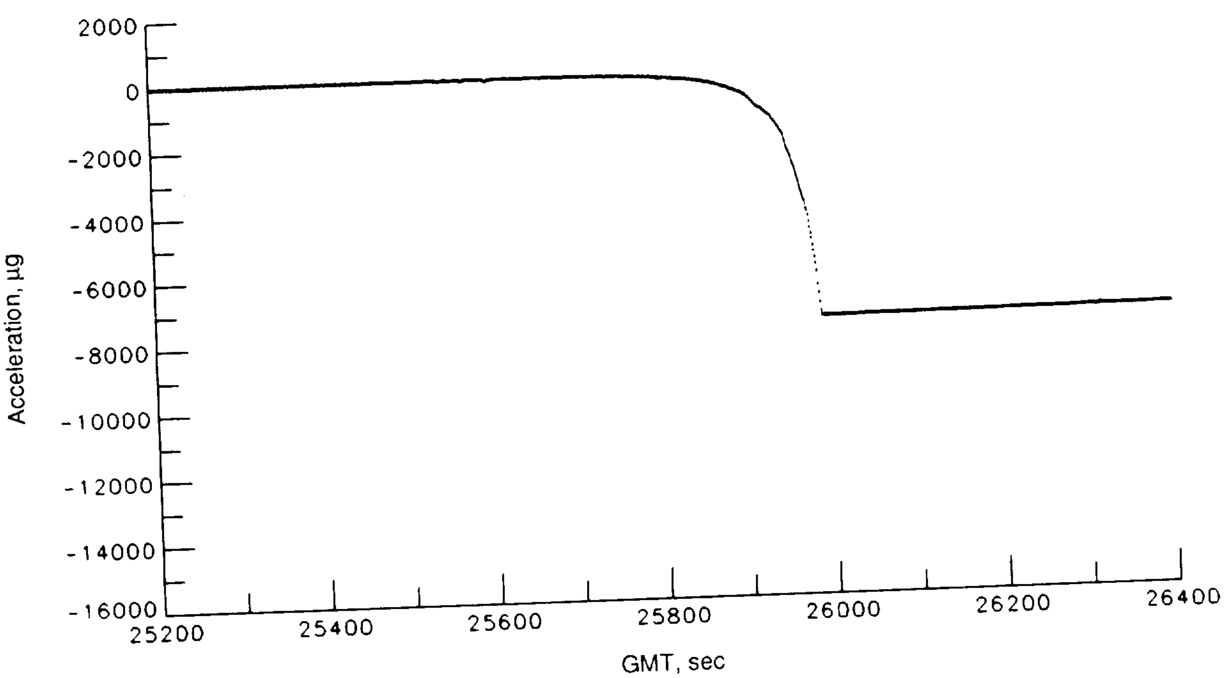


(c) Y-axis acceleration.

Figure 47. Concluded.

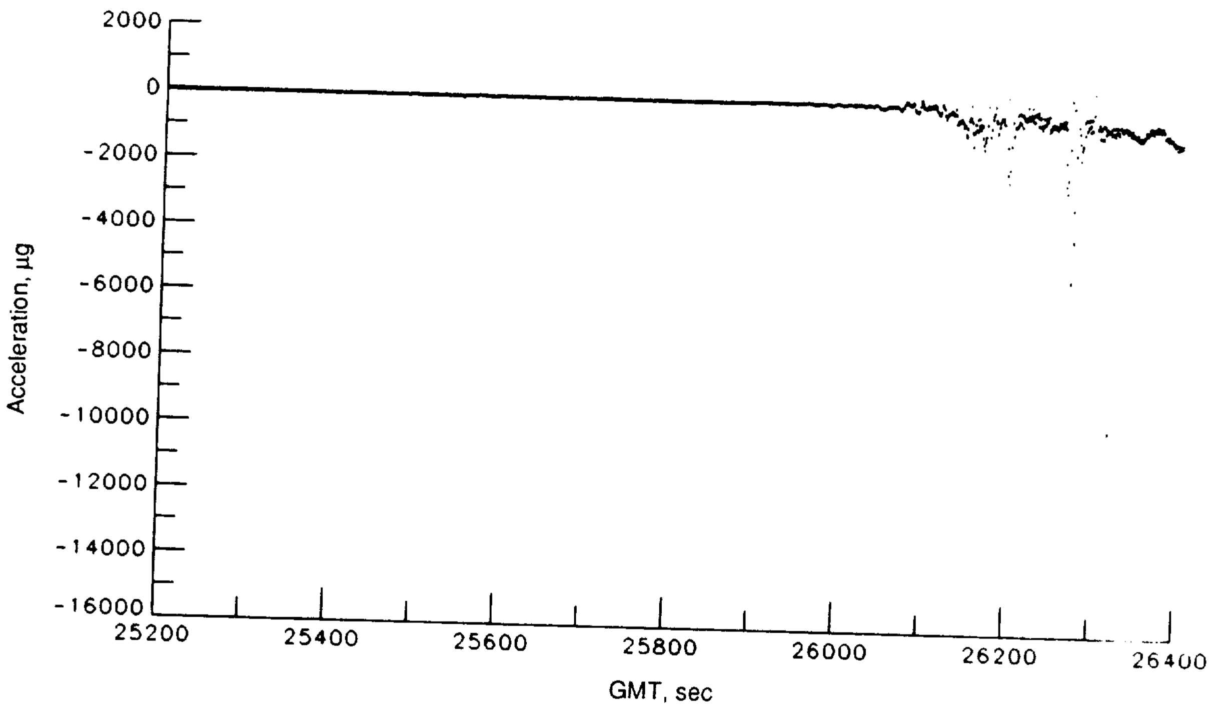


(a) X-axis acceleration.



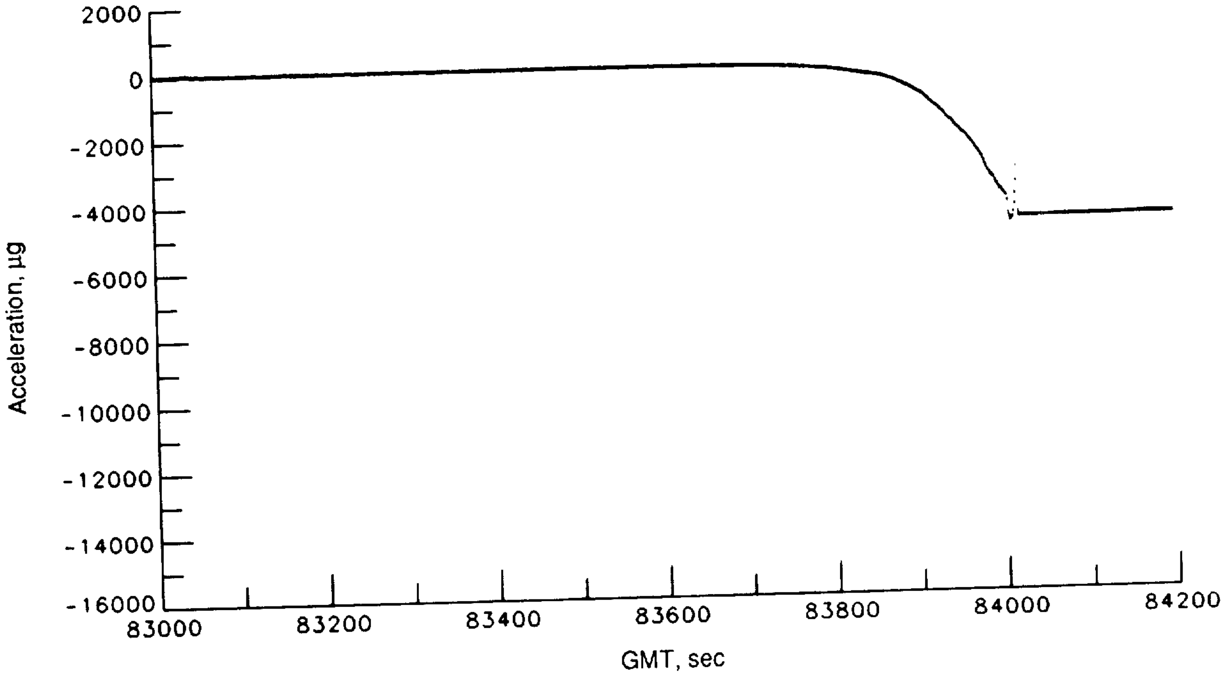
(b) Z-axis acceleration.

Figure 48. One-second averaged aerodynamic acceleration data versus time for STS-08.

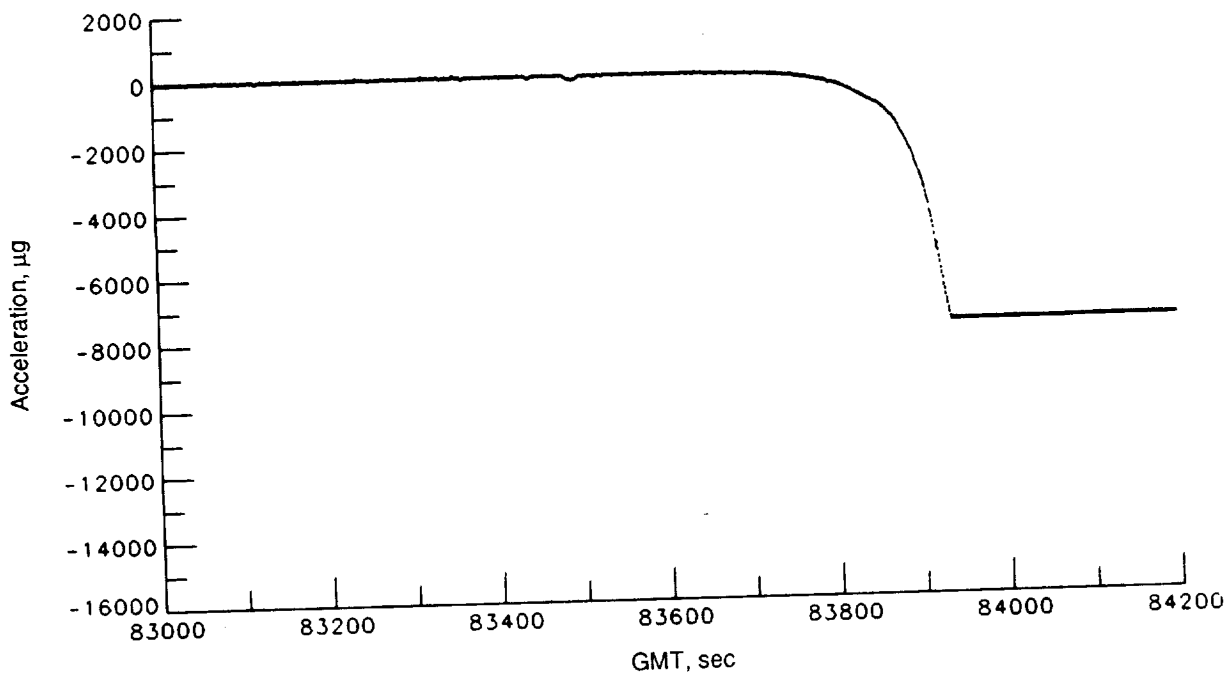


(c) Y-axis acceleration.

Figure 48. Concluded.

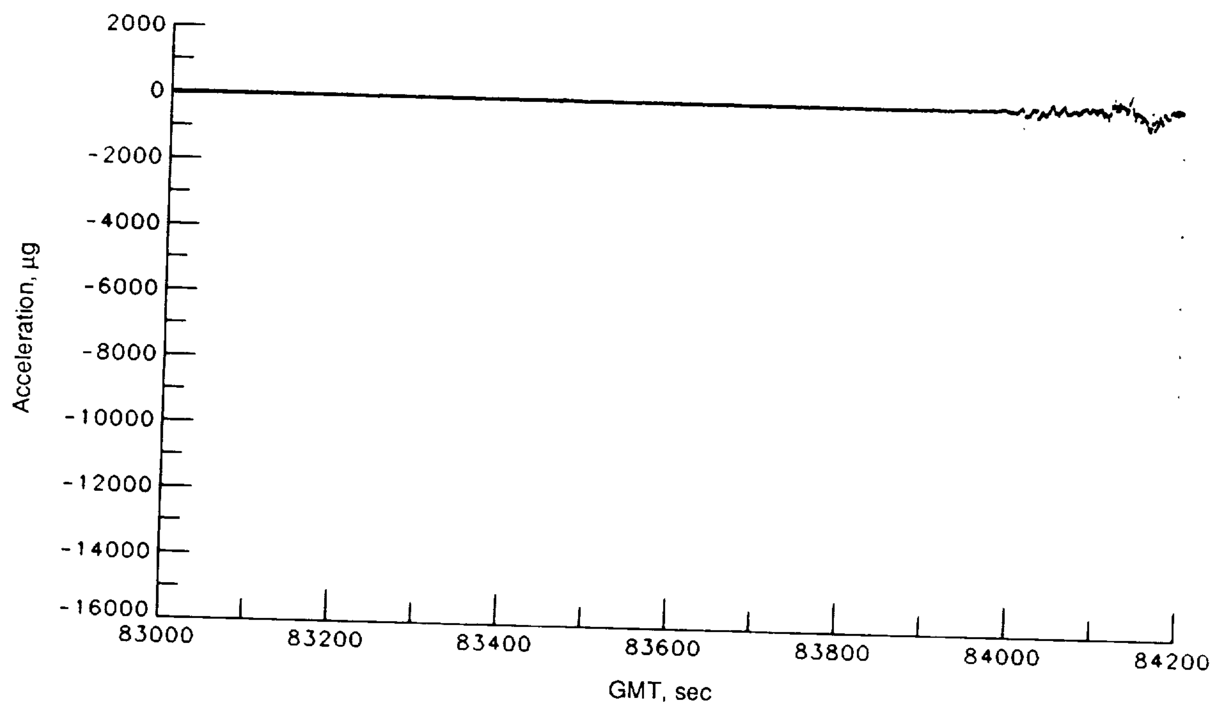


(a) X-axis acceleration.



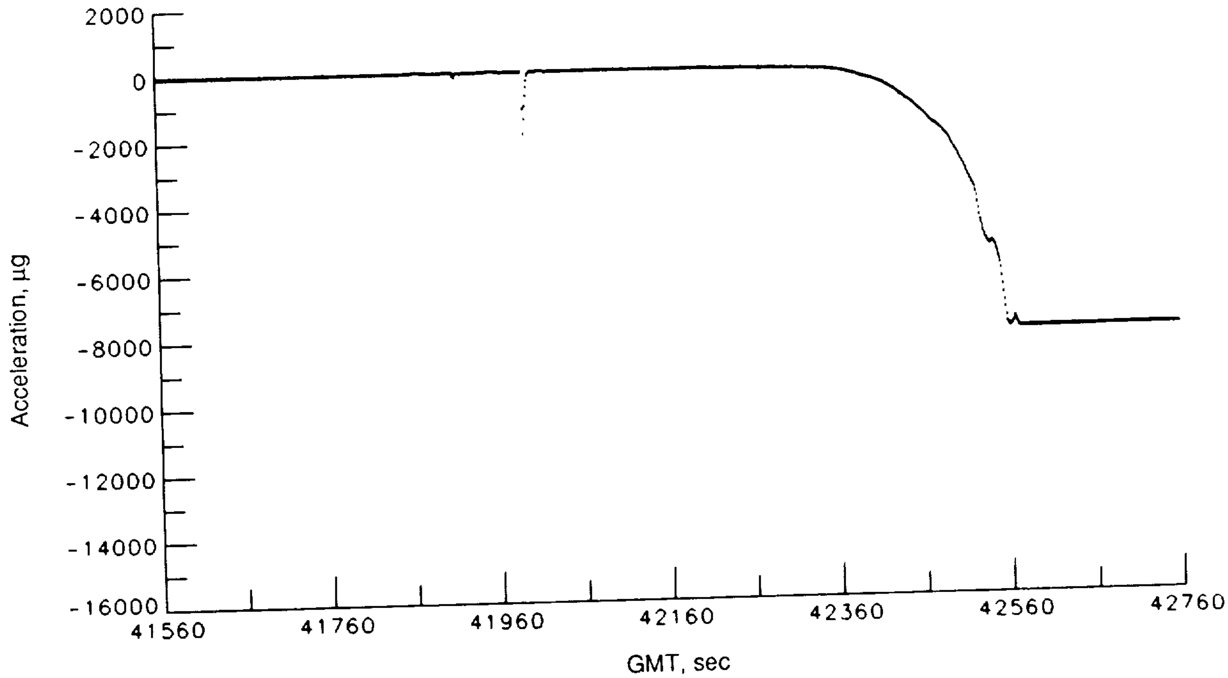
(b) Z-axis acceleration.

Figure 49. One-second averaged aerodynamic acceleration data versus time for STS-09.

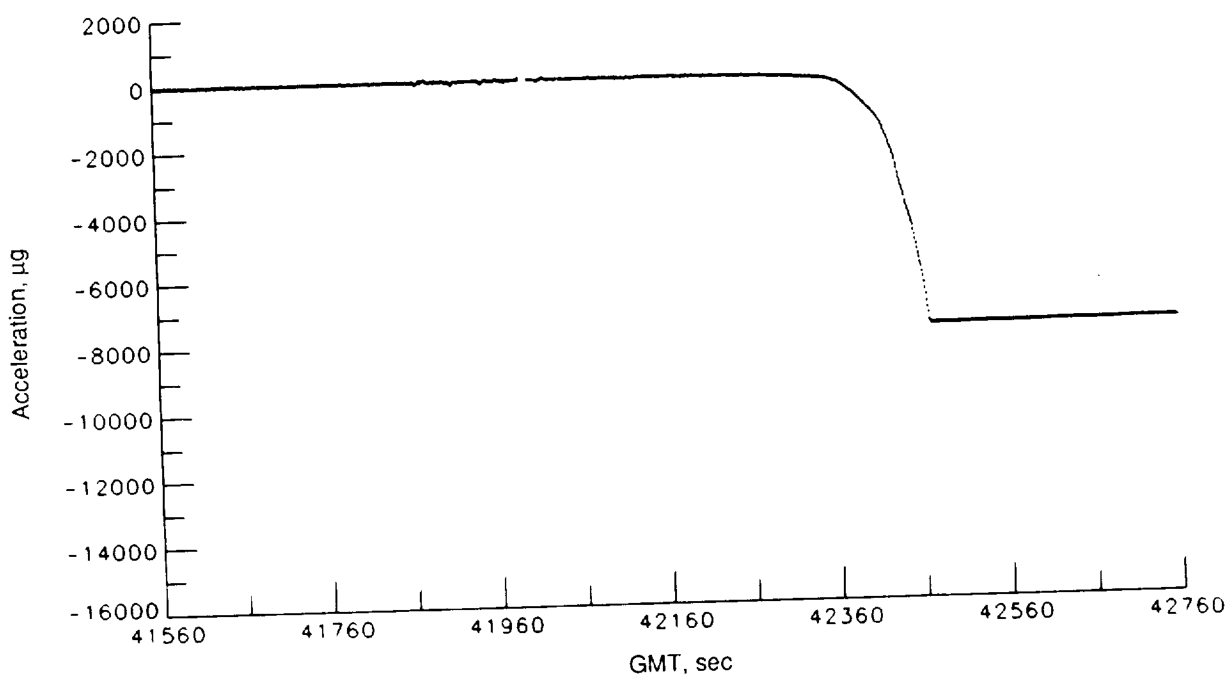


(c) Y-axis acceleration.

Figure 49. Concluded.

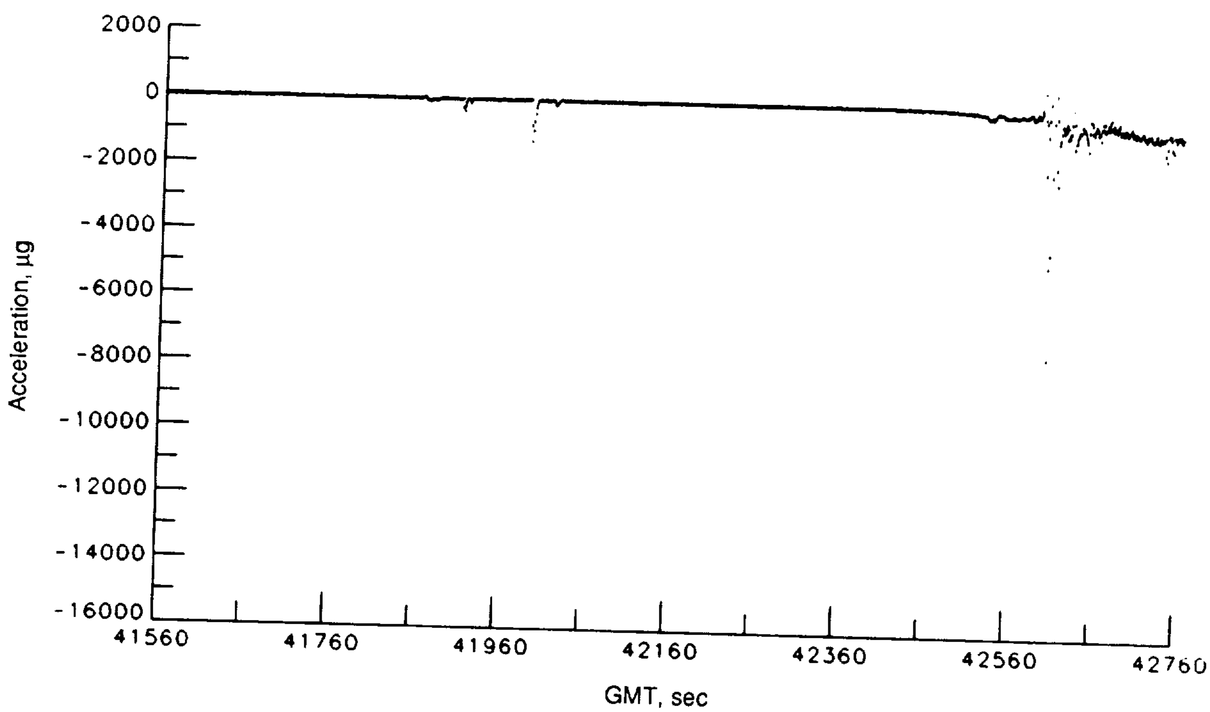


(a) X-axis acceleration.



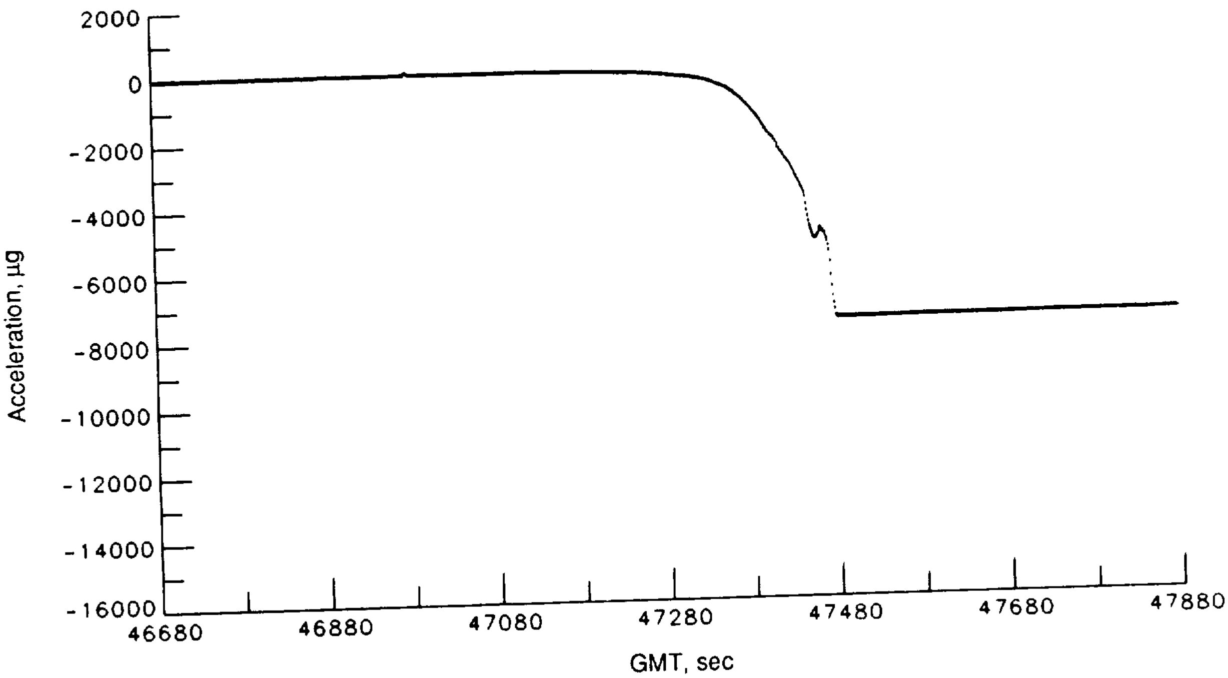
(b) Z-axis acceleration.

Figure 50. One-second averaged aerodynamic acceleration data versus time for STS-41B.

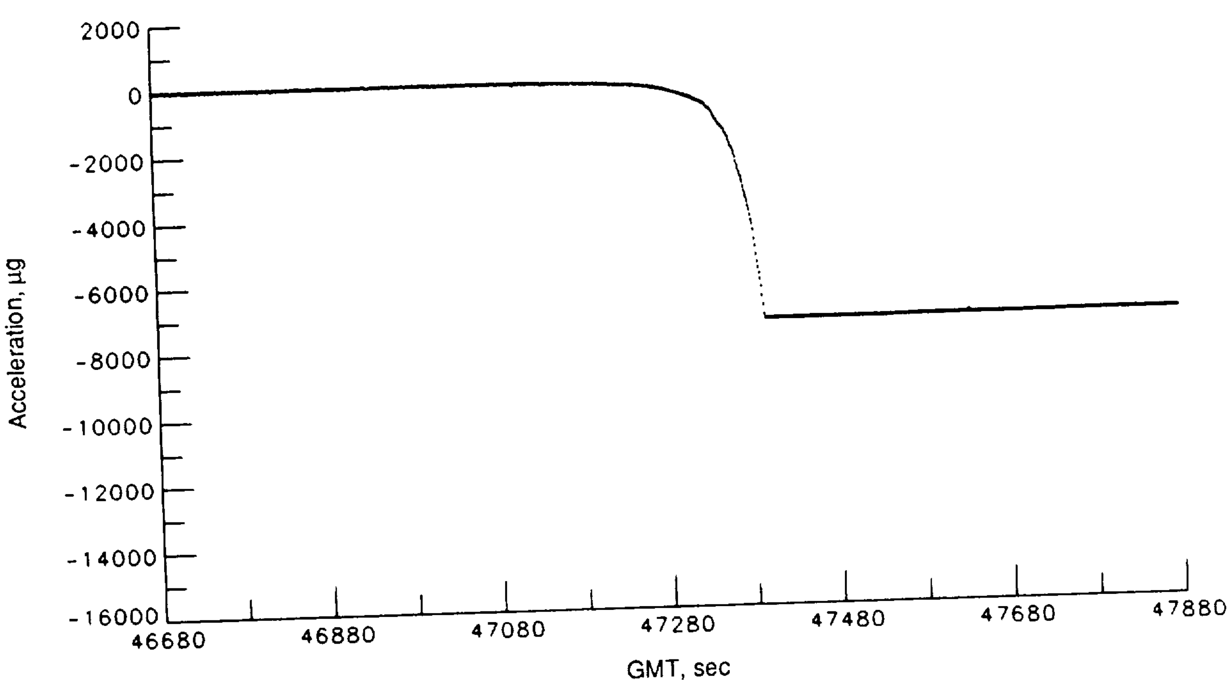


(c) Y-axis acceleration.

Figure 50. Concluded.



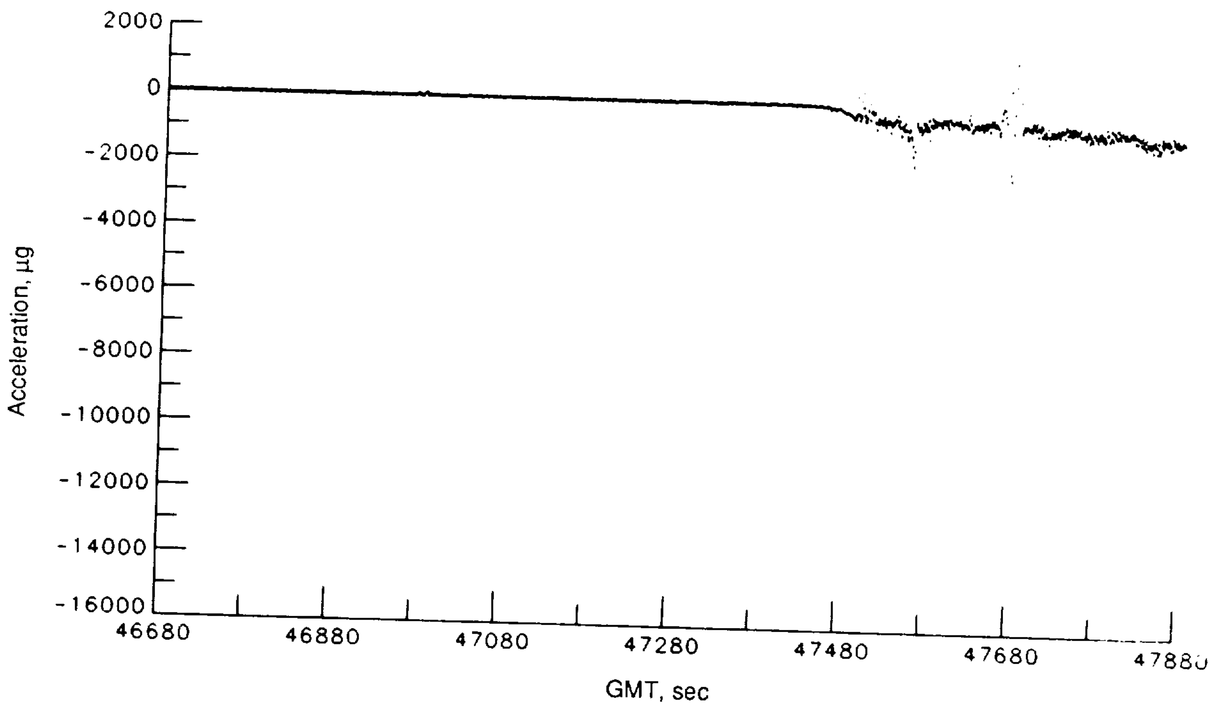
(a) X-axis acceleration.



(b) Z-axis acceleration.

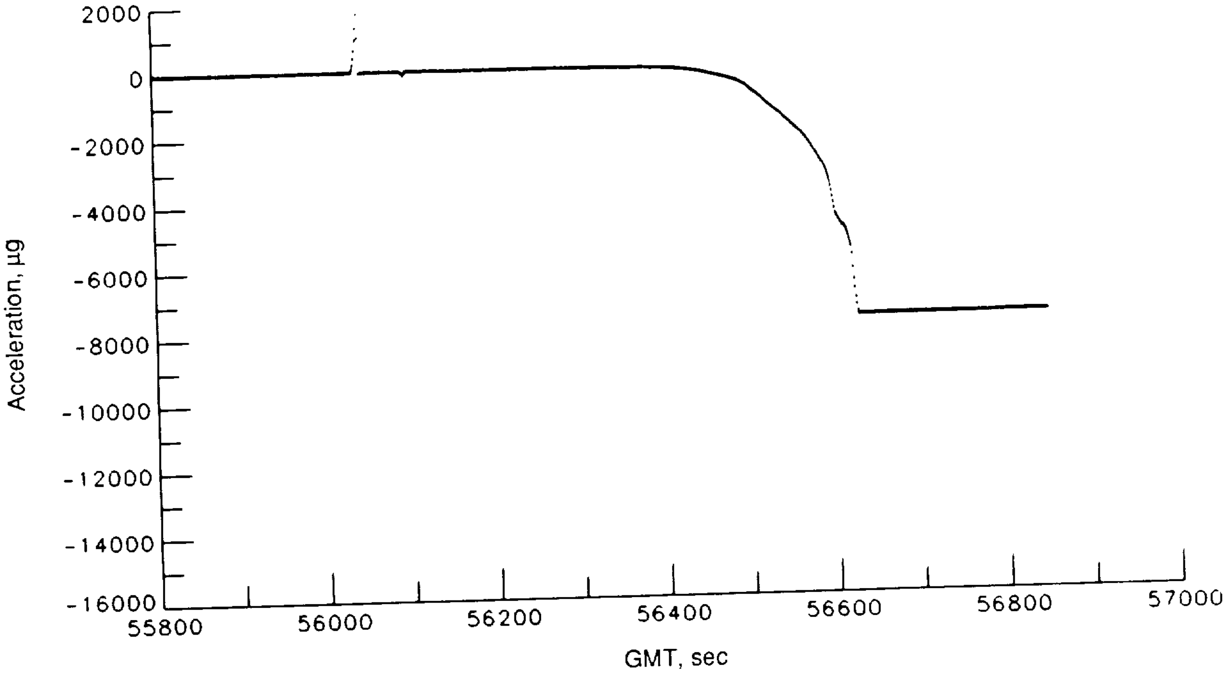
Figure 51. One-second averaged aerodynamic acceleration data versus time for STS-41C.

C-4

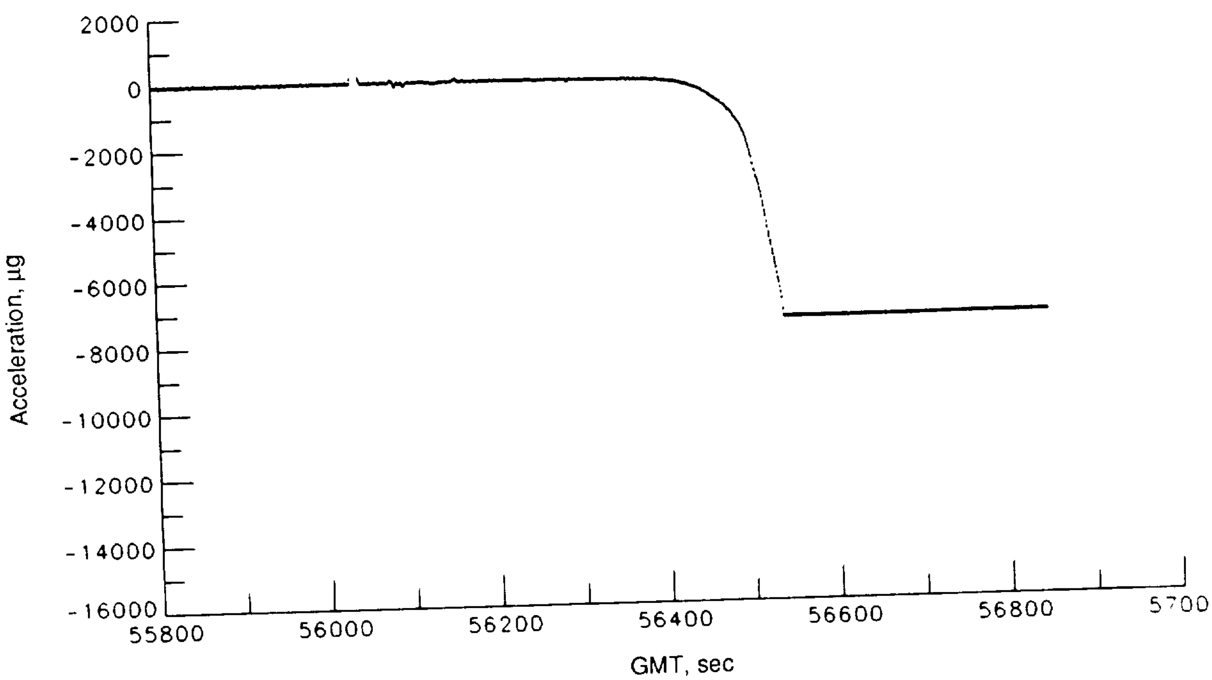


(c) Y-axis acceleration.

Figure 51. Concluded.

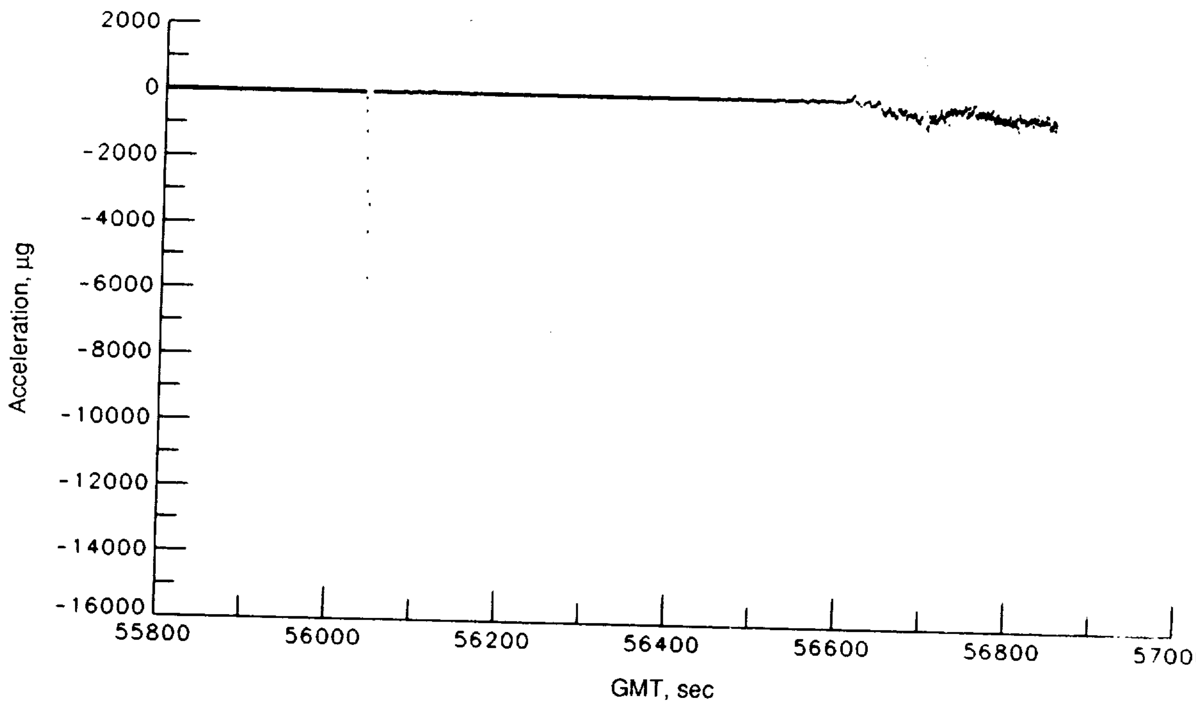


(a) X-axis acceleration.



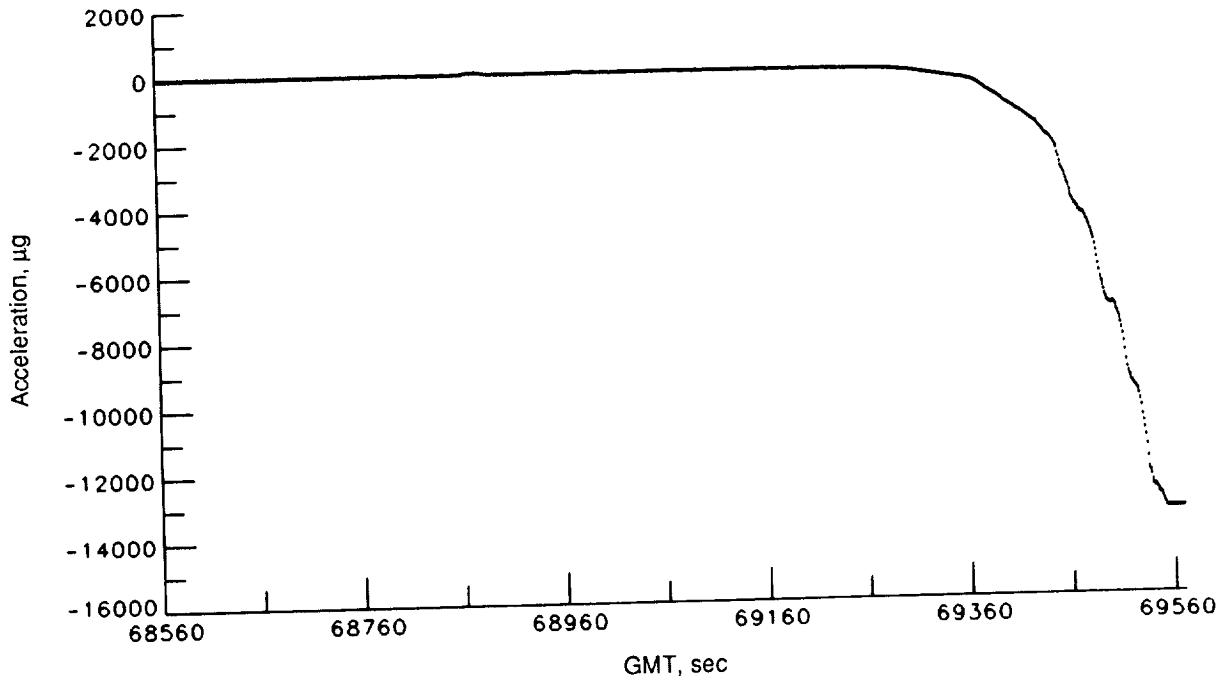
(b) Z-axis acceleration.

Figure 52. One-second averaged aerodynamic acceleration data versus time for STS-51B.

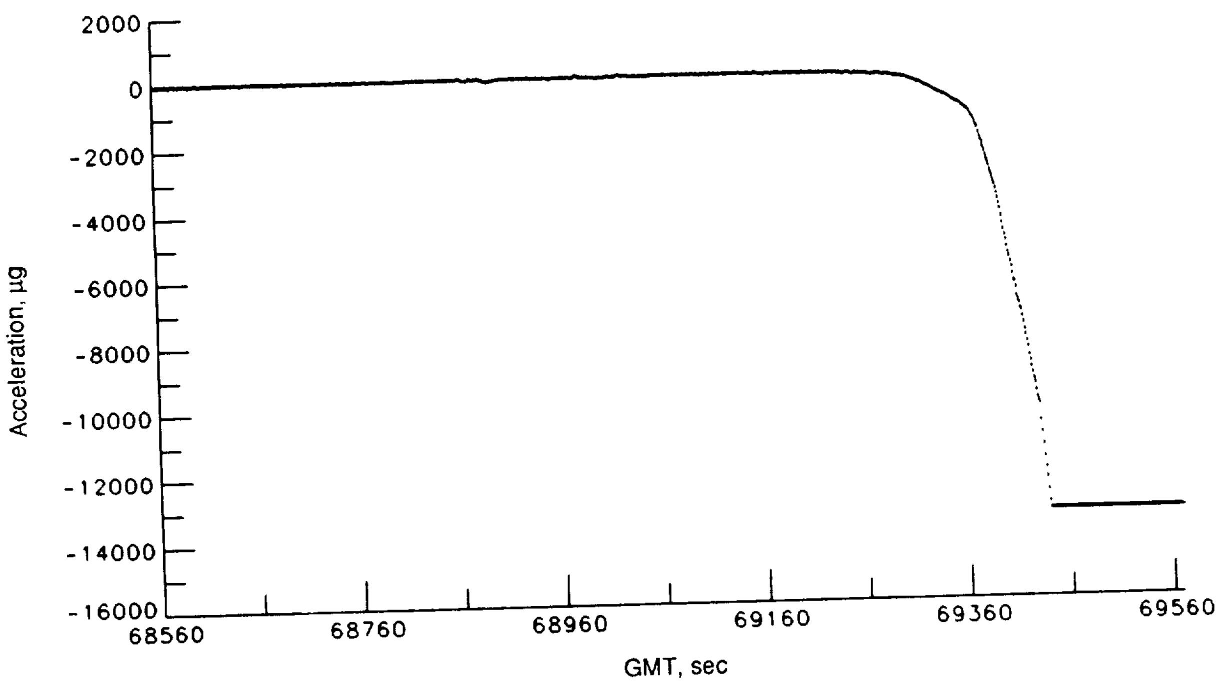


(c) Y-axis acceleration.

Figure 52. Concluded.

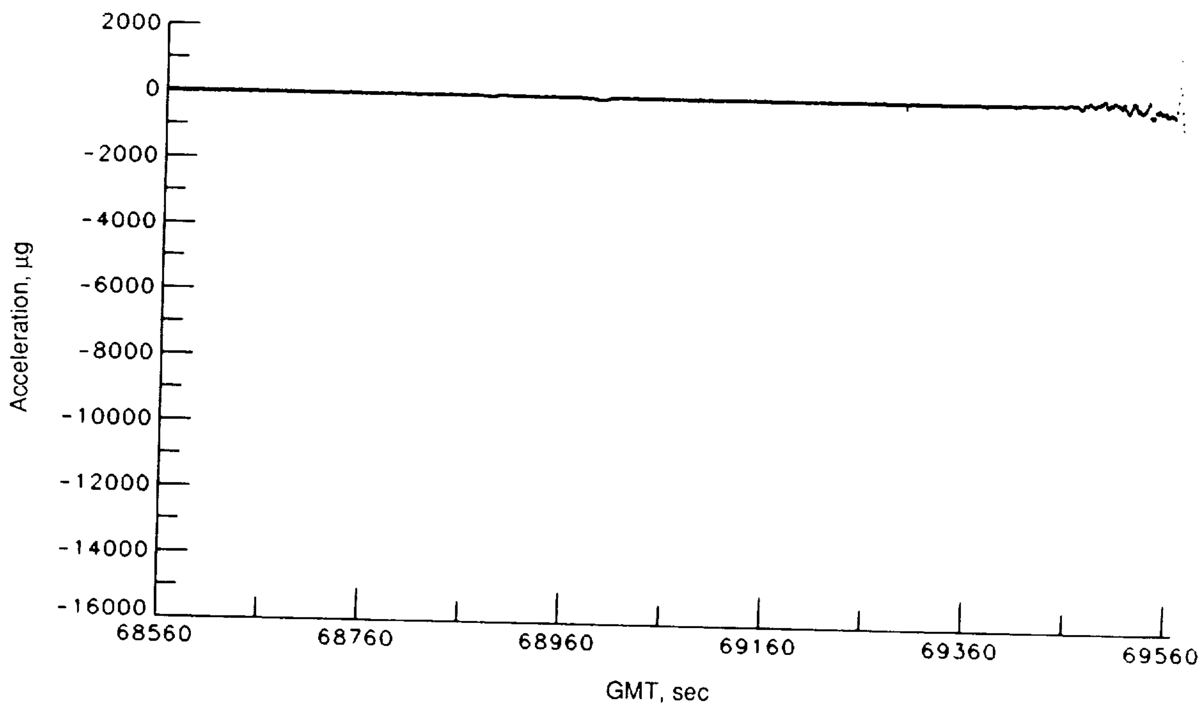


(a) X-axis acceleration.



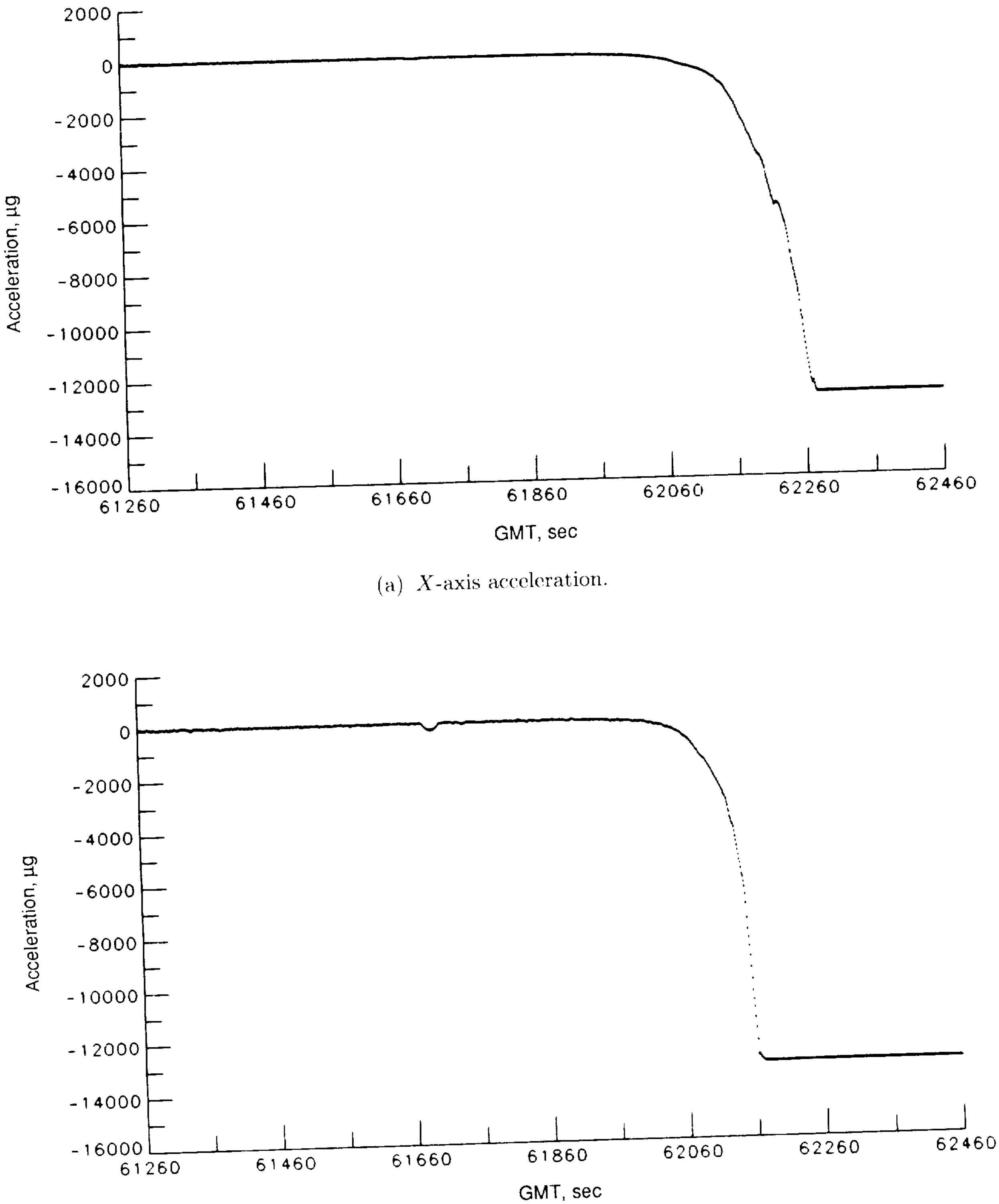
(b) Z-axis acceleration.

Figure 53. One-second averaged aerodynamic acceleration data versus time for STS-51F.



(c) Y-axis acceleration.

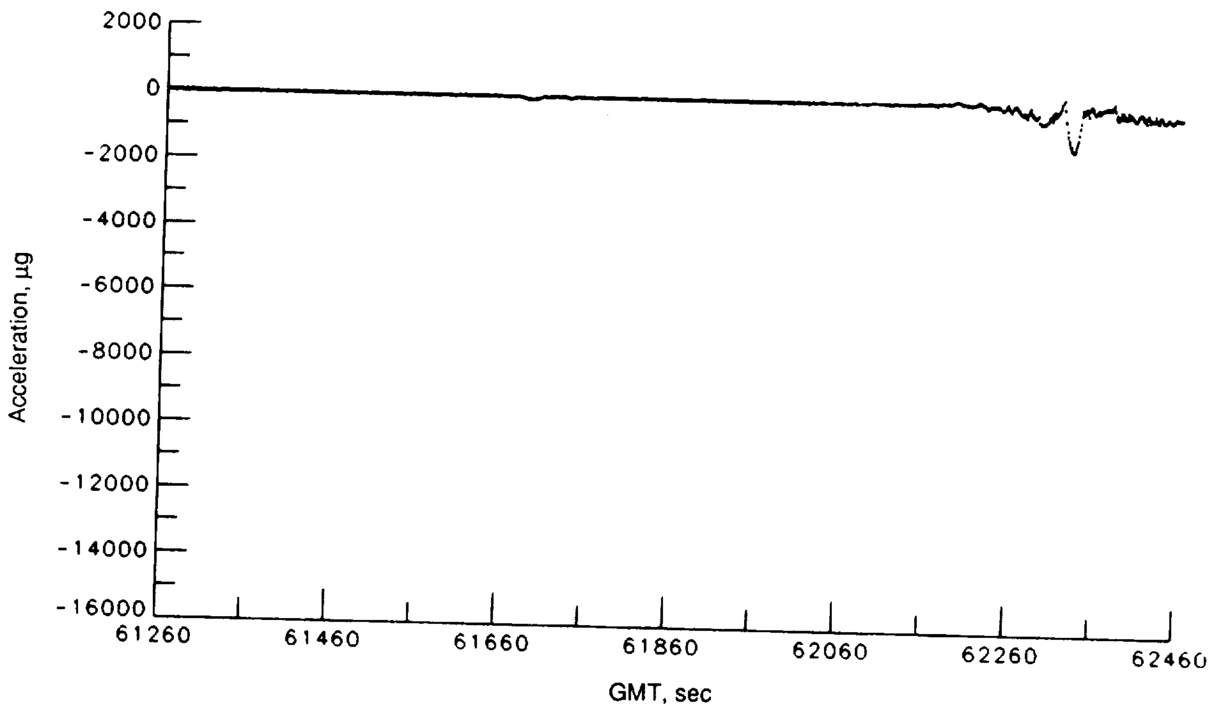
Figure 53. Concluded.



(a) X-axis acceleration.

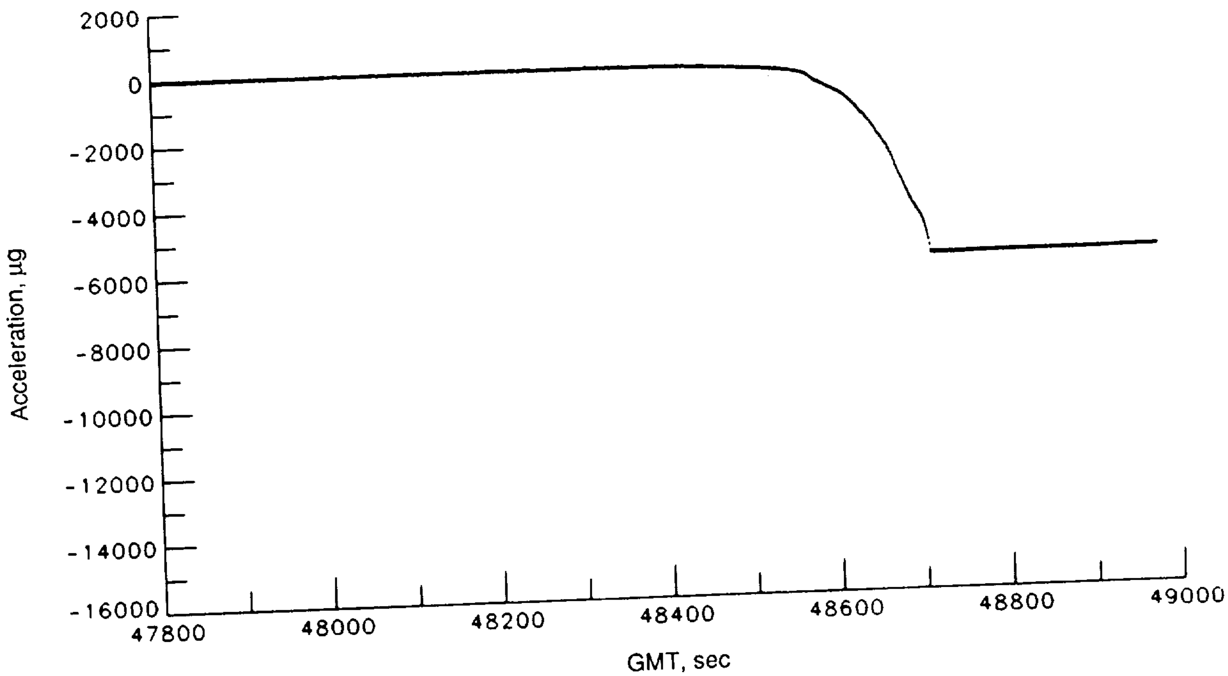
(b) Z-axis acceleration.

Figure 54. One-second averaged aerodynamic acceleration data versus time for STS-61A.

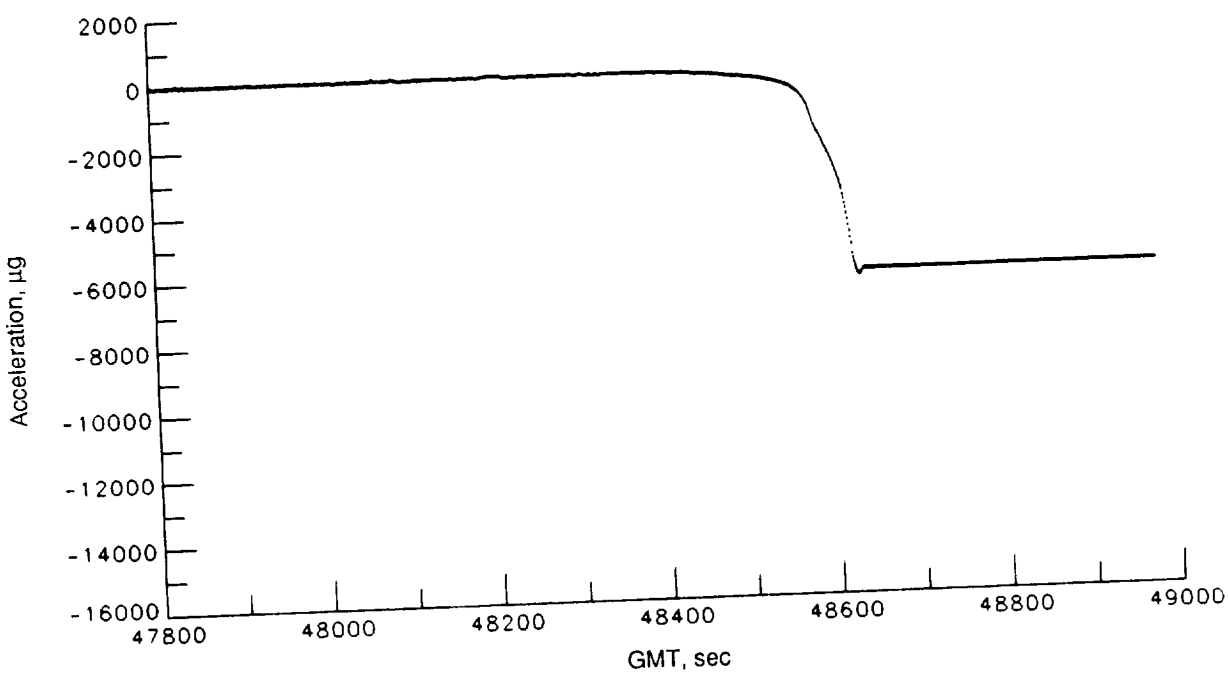


(c) Y-axis acceleration.

Figure 54. Concluded.

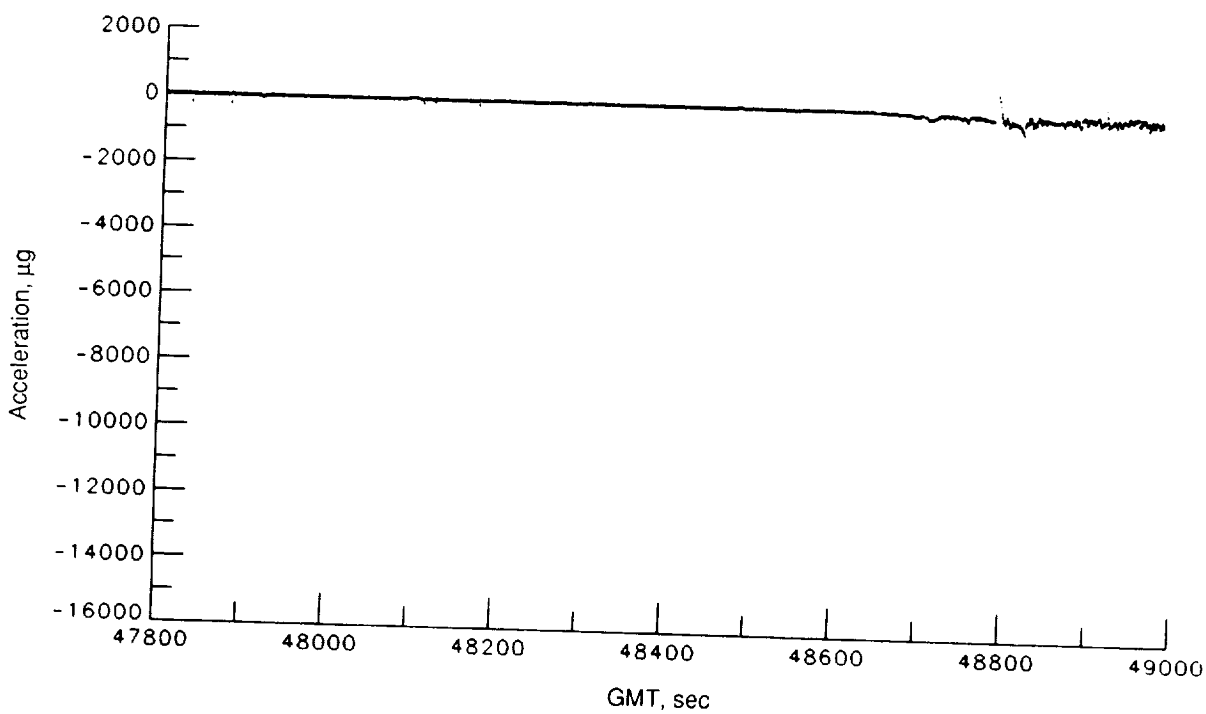


(a) X-axis acceleration.



(b) Z-axis acceleration.

Figure 55. One-second averaged aerodynamic acceleration data versus time for STS-61C.



(c) Y-axis acceleration.

Figure 55. Concluded.

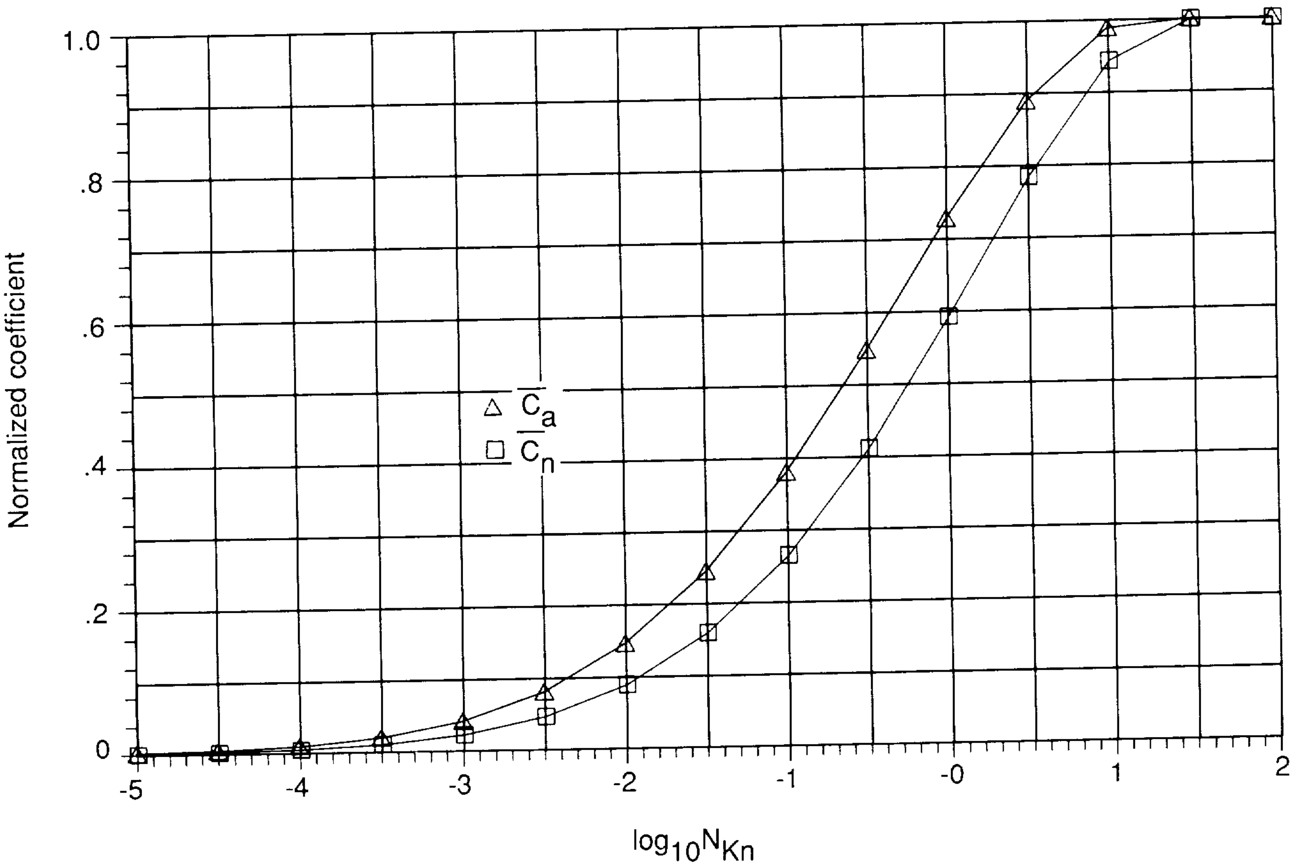
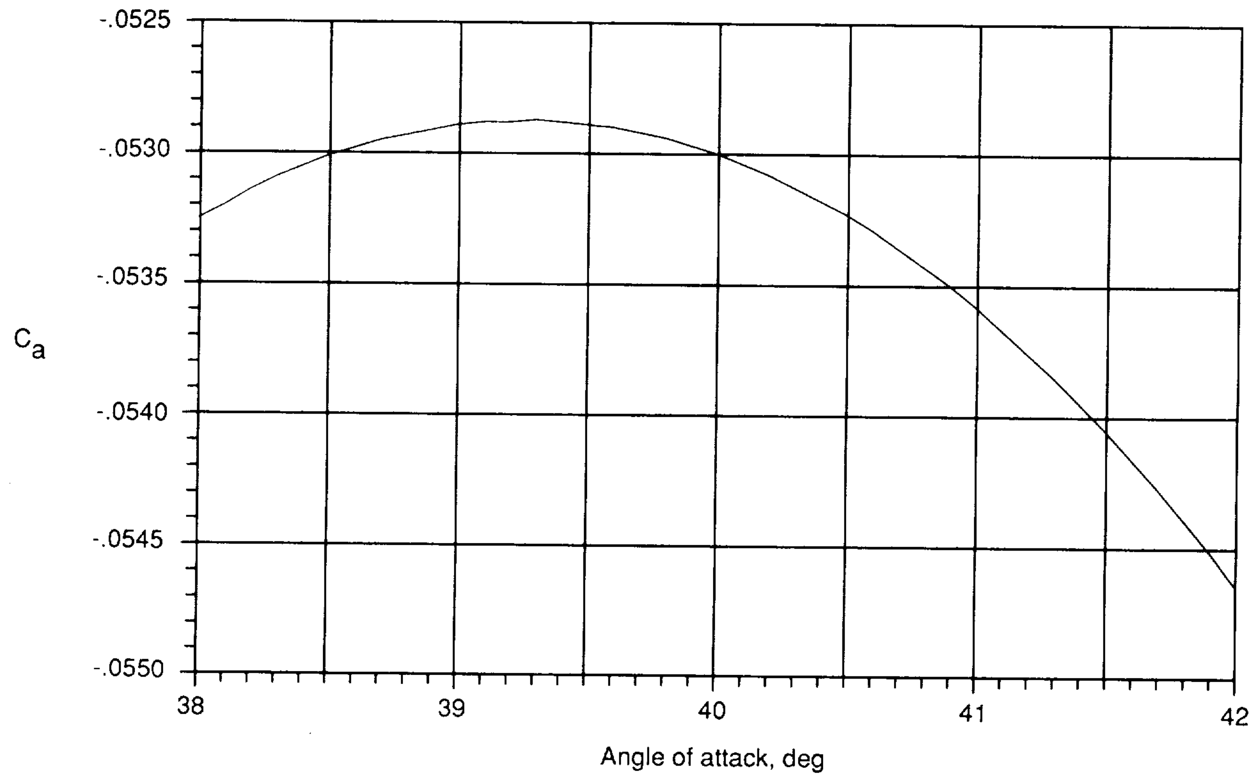


Figure 56. Normalized force coefficients versus Knudsen number.



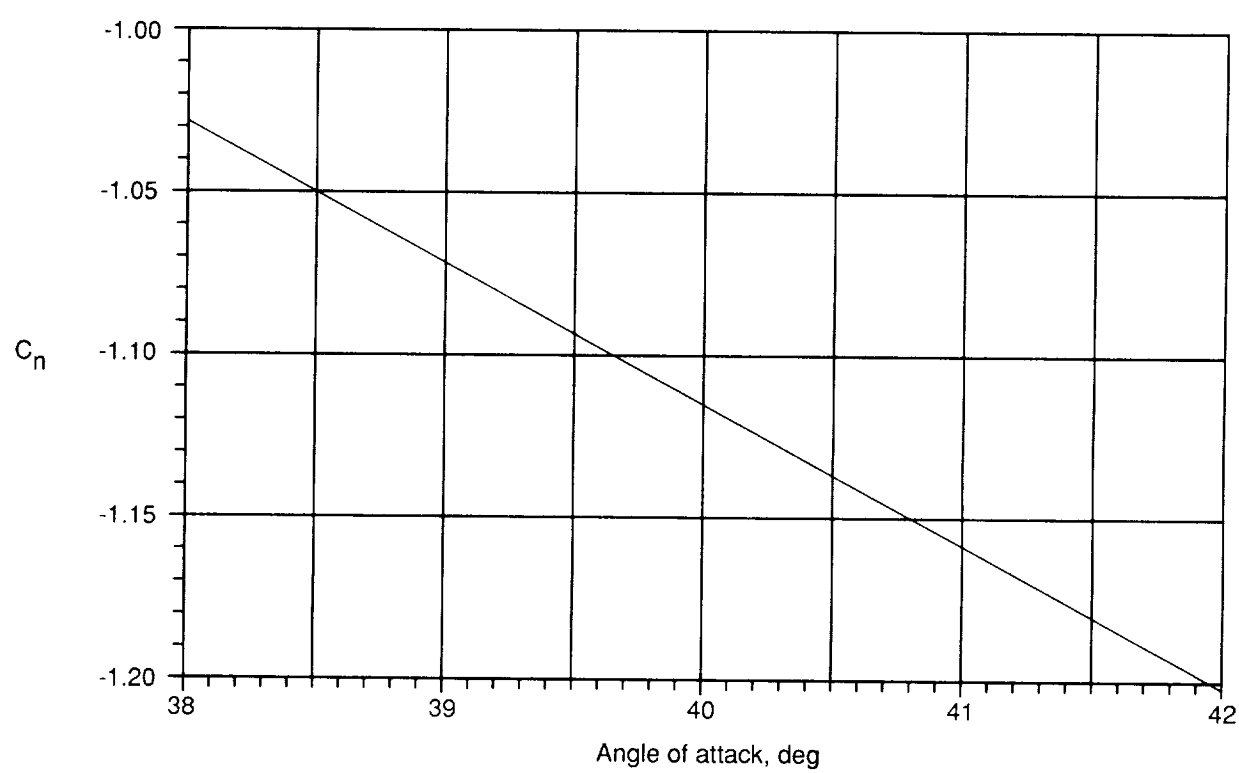
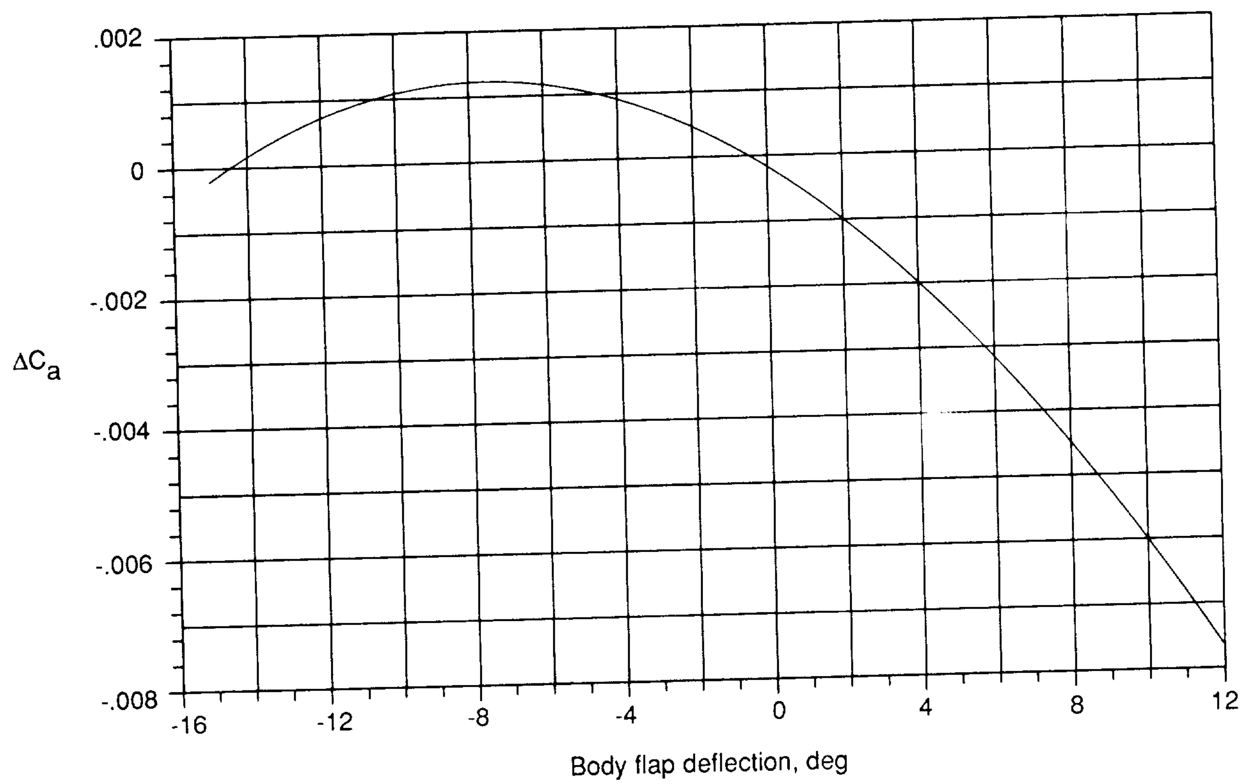


Figure 57. Axial- and normal-force coefficients versus angle of attack.



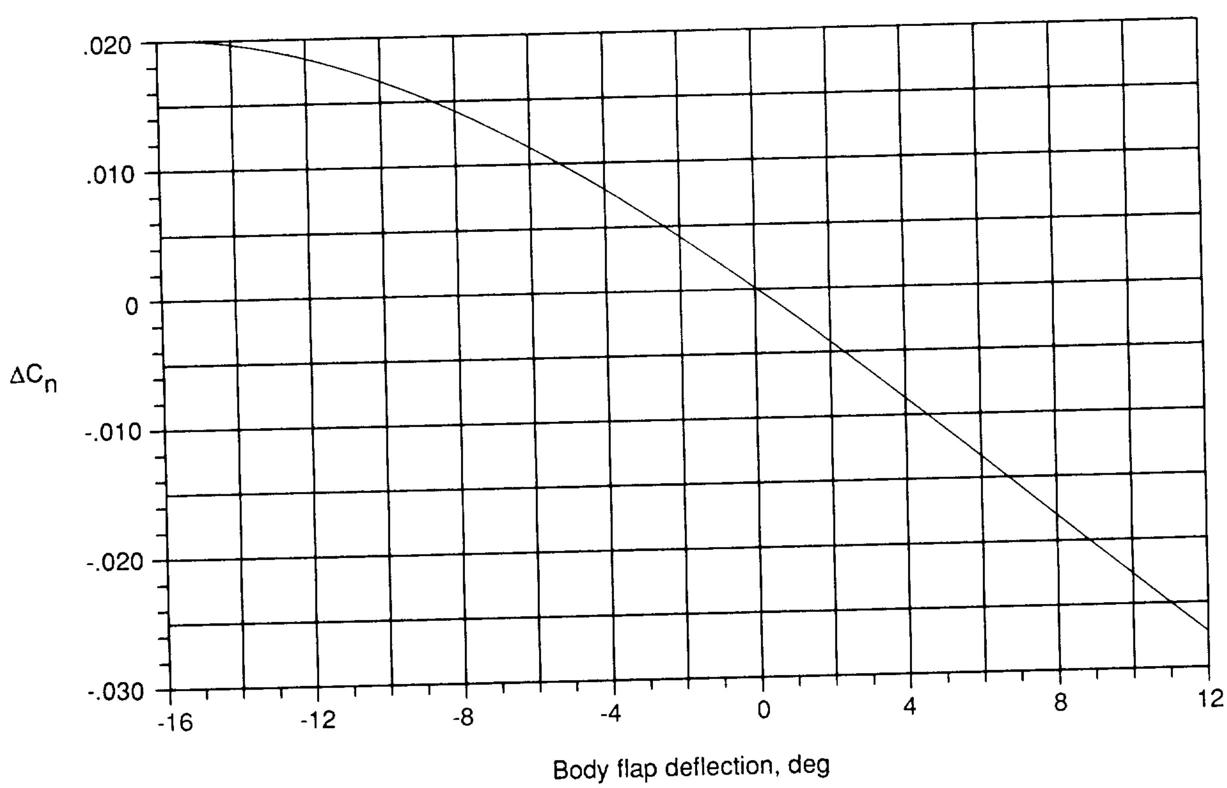


Figure 58. Incremental axial- and normal-force coefficients versus body flap deflection.

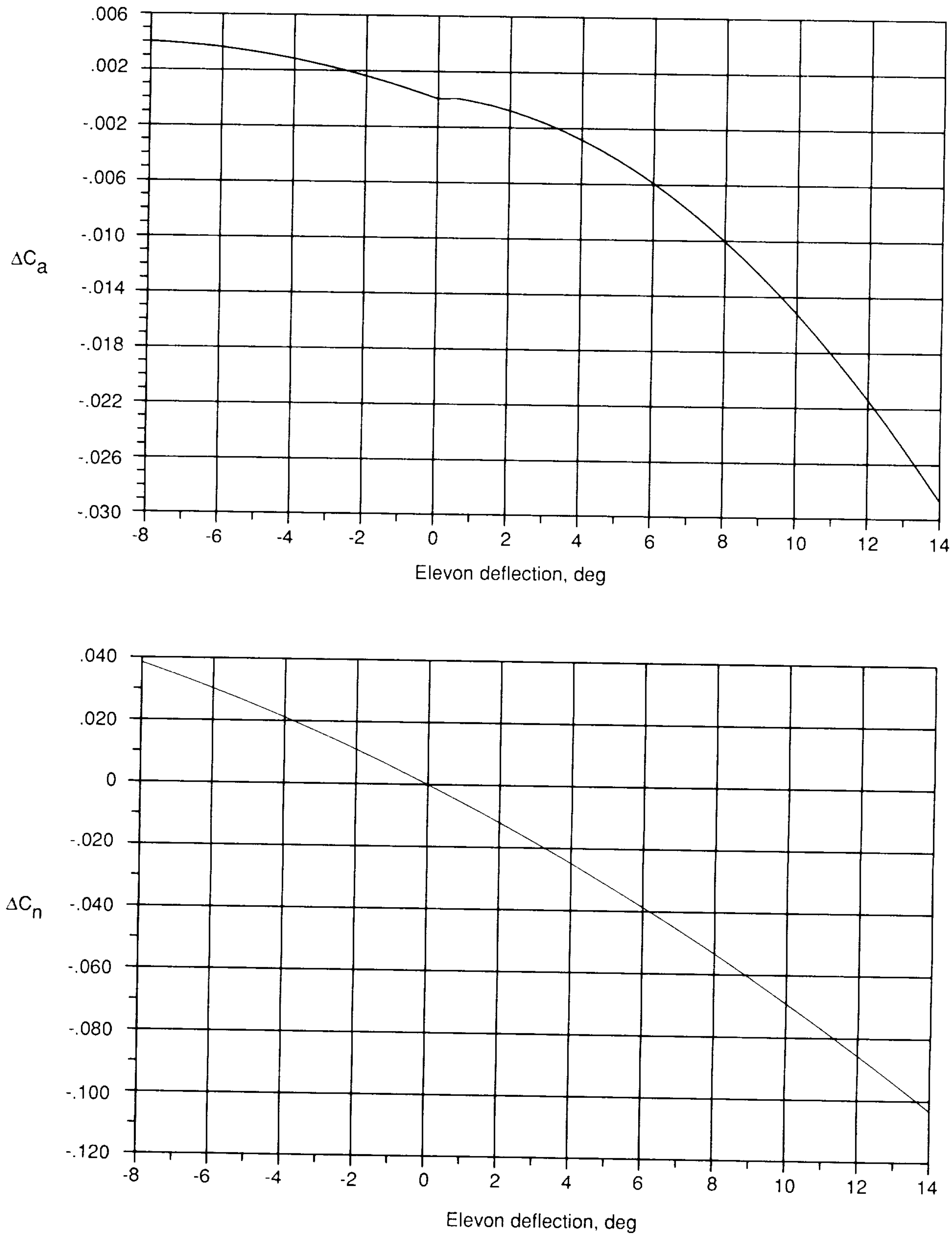
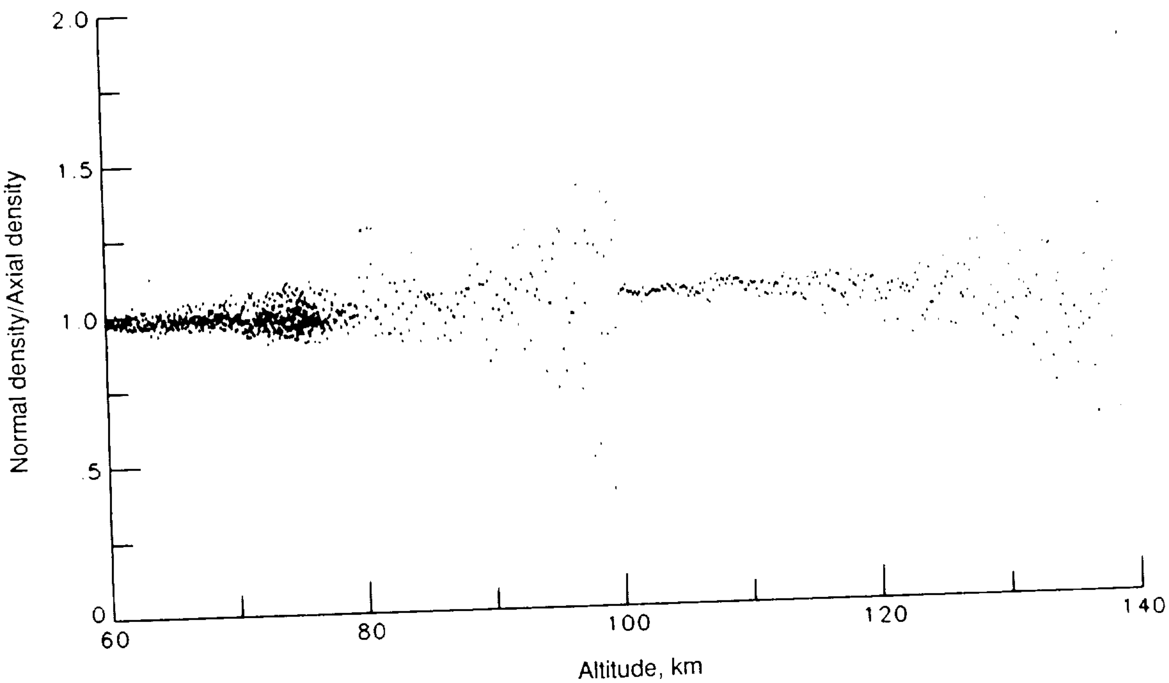
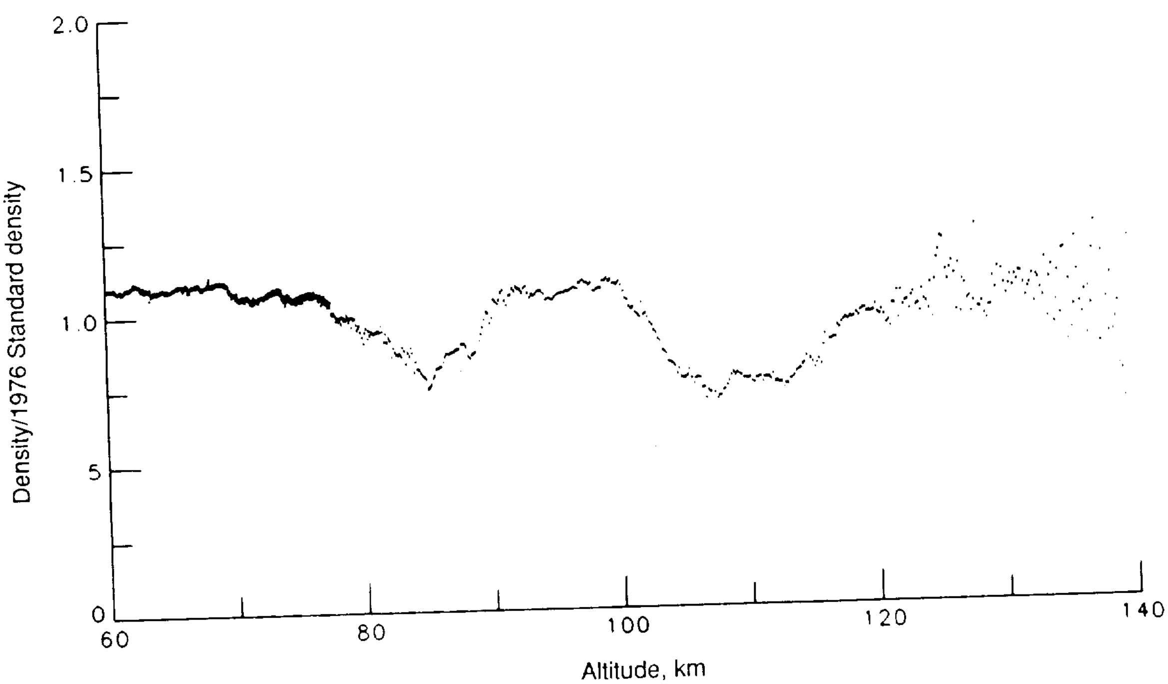


Figure 59. Incremental axial- and normal-force coefficients versus elevon deflection.

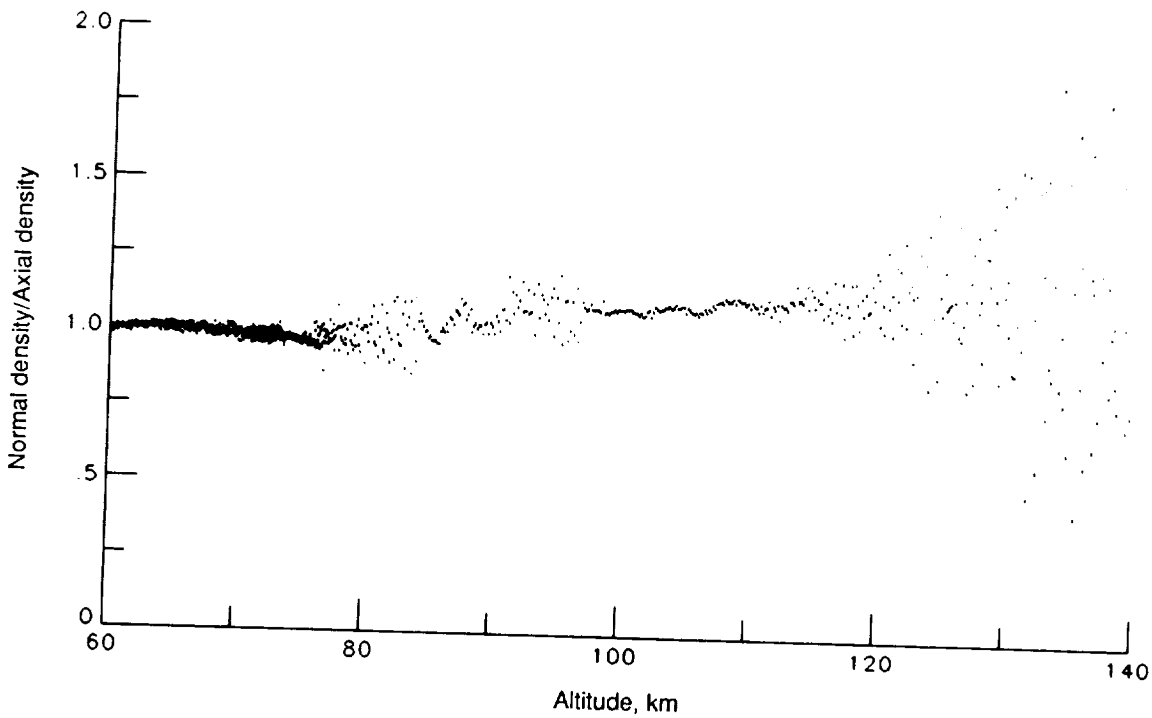


(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

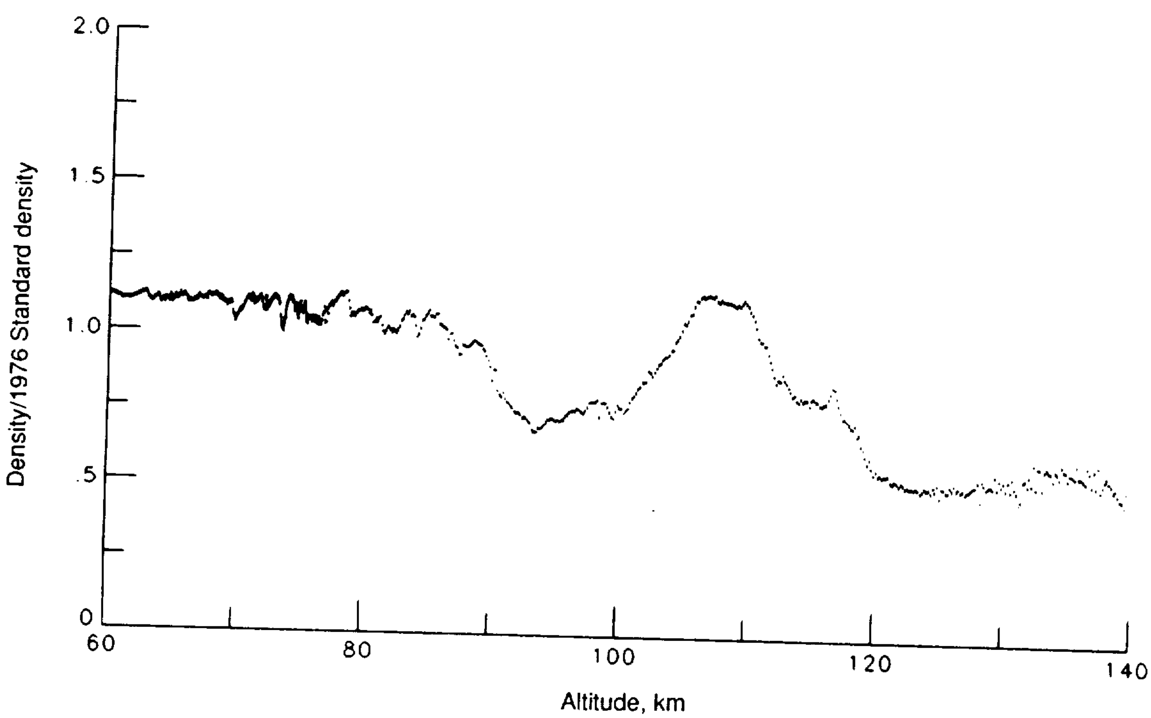


(b) Profile of calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model profile.

Figure 60. Density analysis results for STS-06.

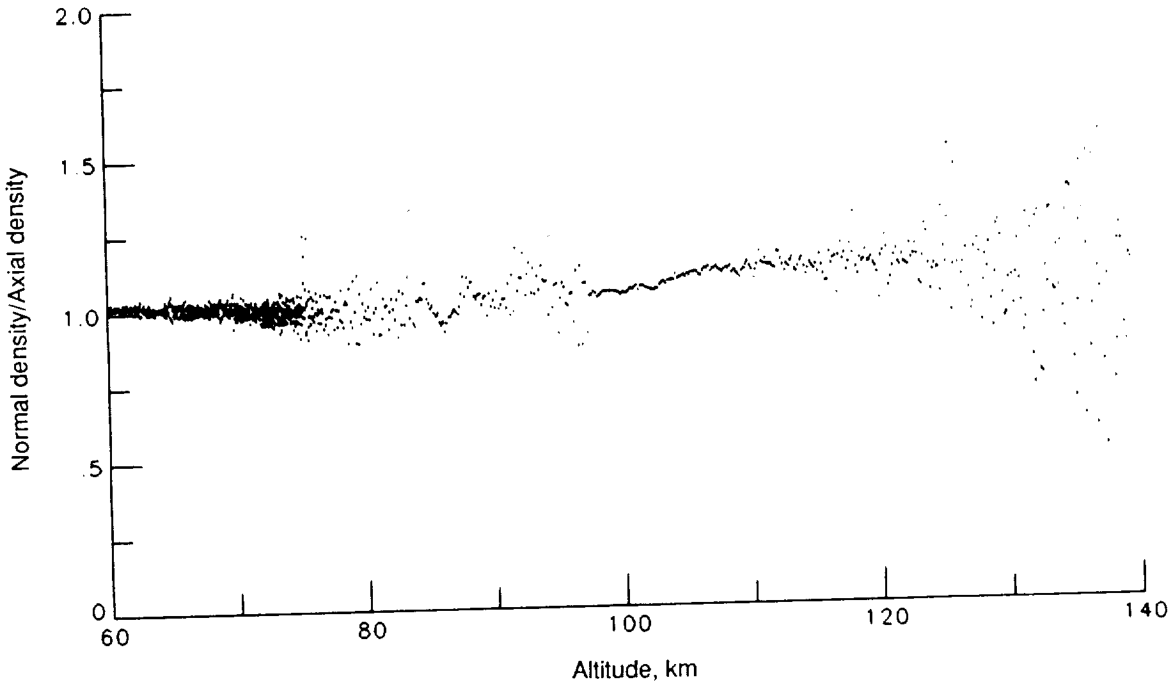


(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

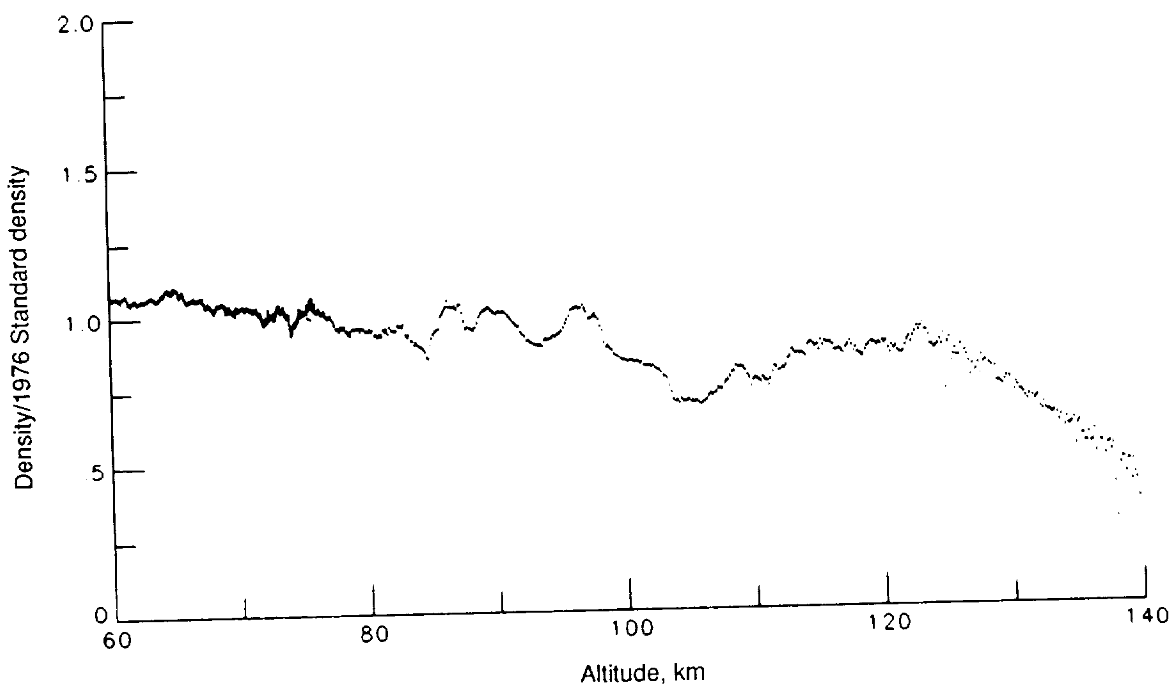


(b) Profile of calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model profile.

Figure 61. Density analysis results for STS-07.

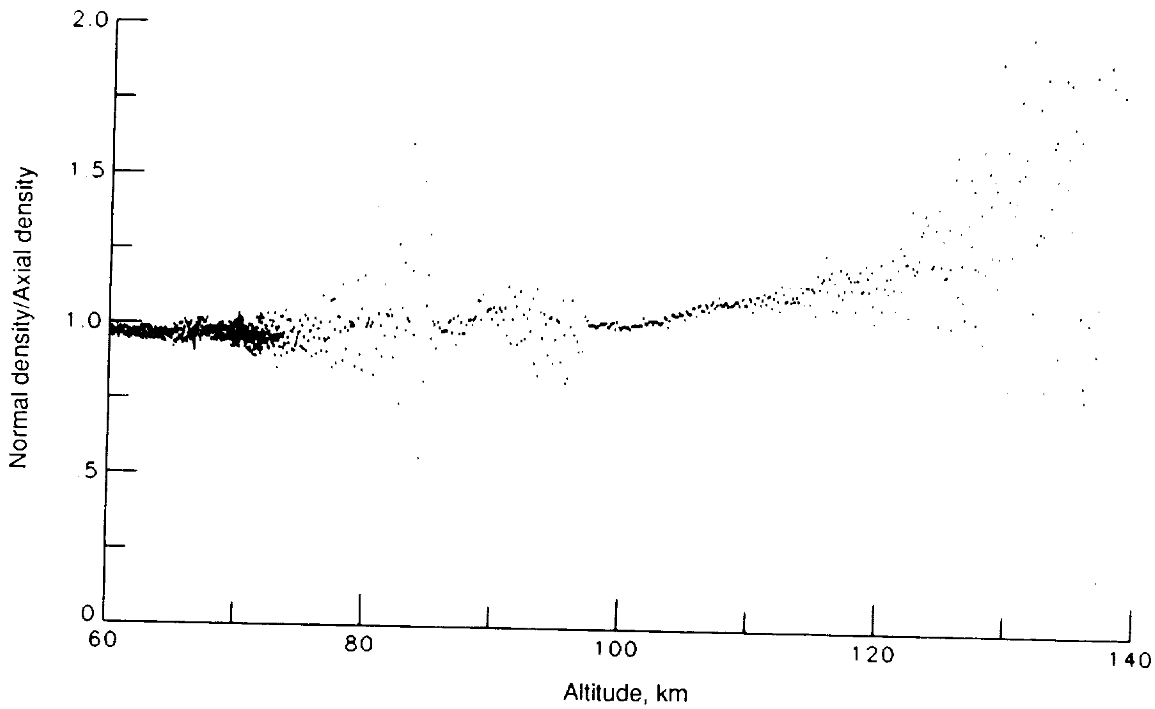


(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

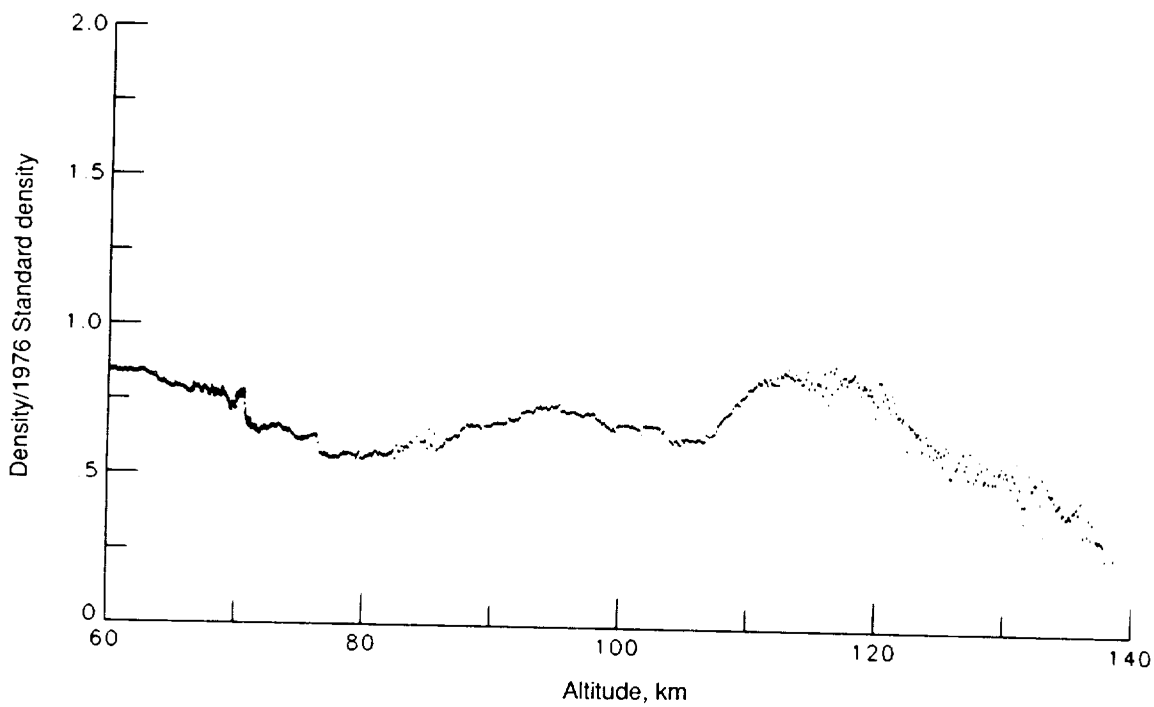


(b) Profile of calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model profile.

Figure 62. Density analysis results for STS-08.

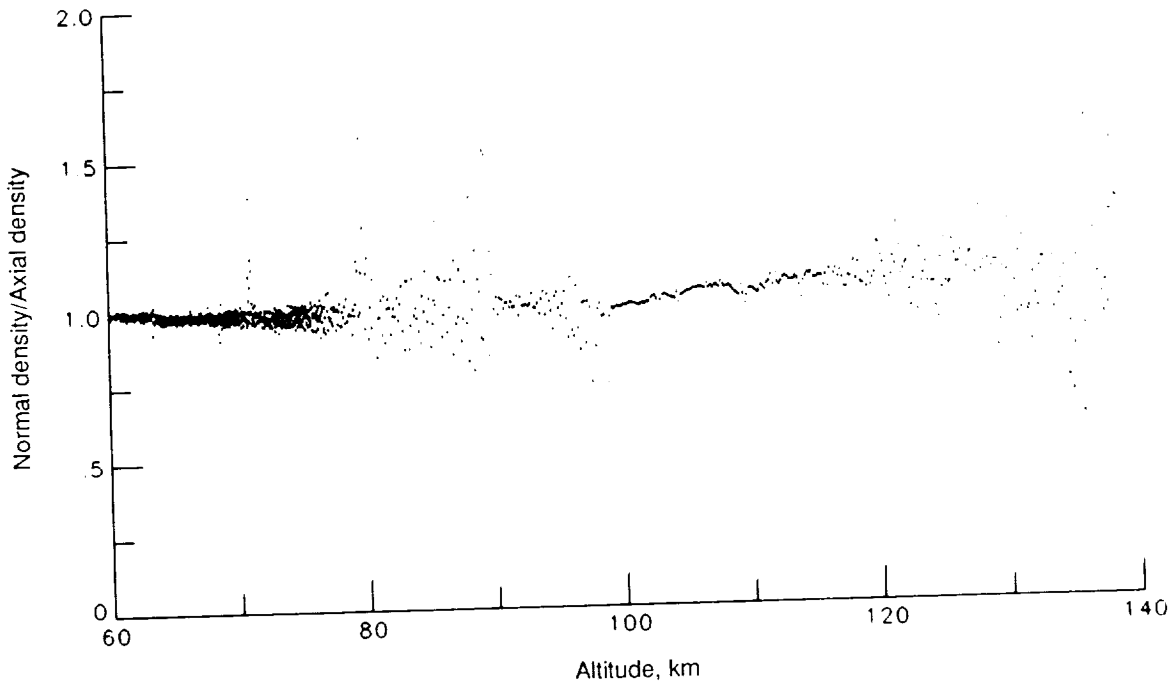


(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

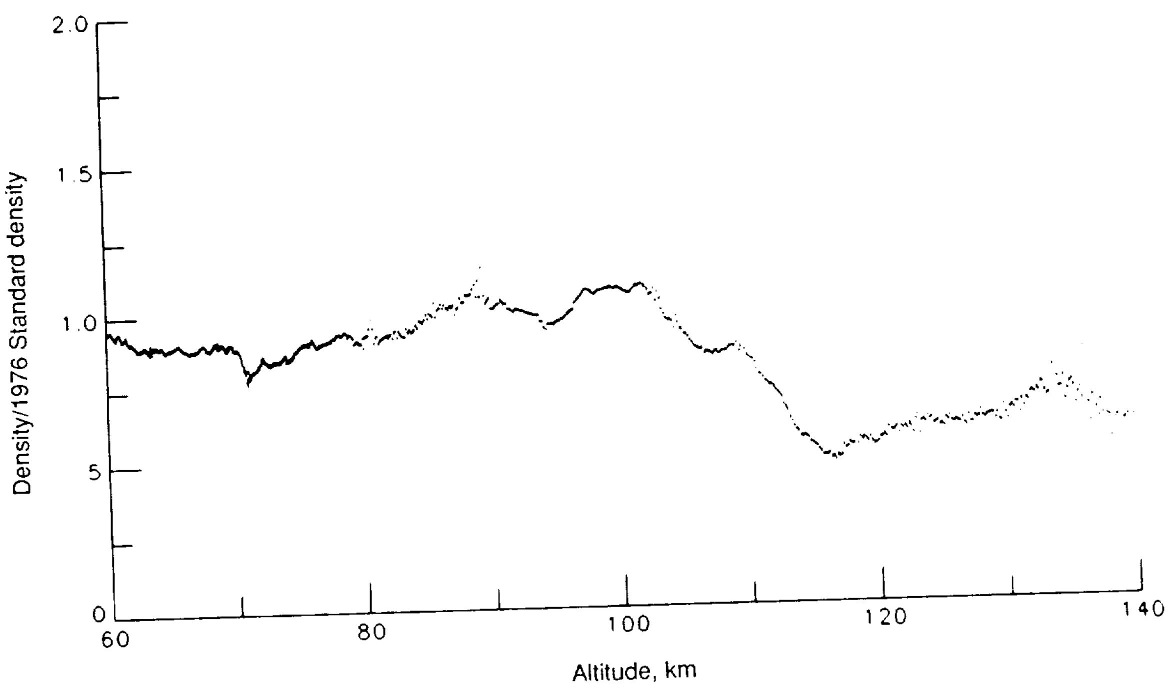


(b) Profile of calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model profile.

Figure 63. Density analysis results for STS-09.

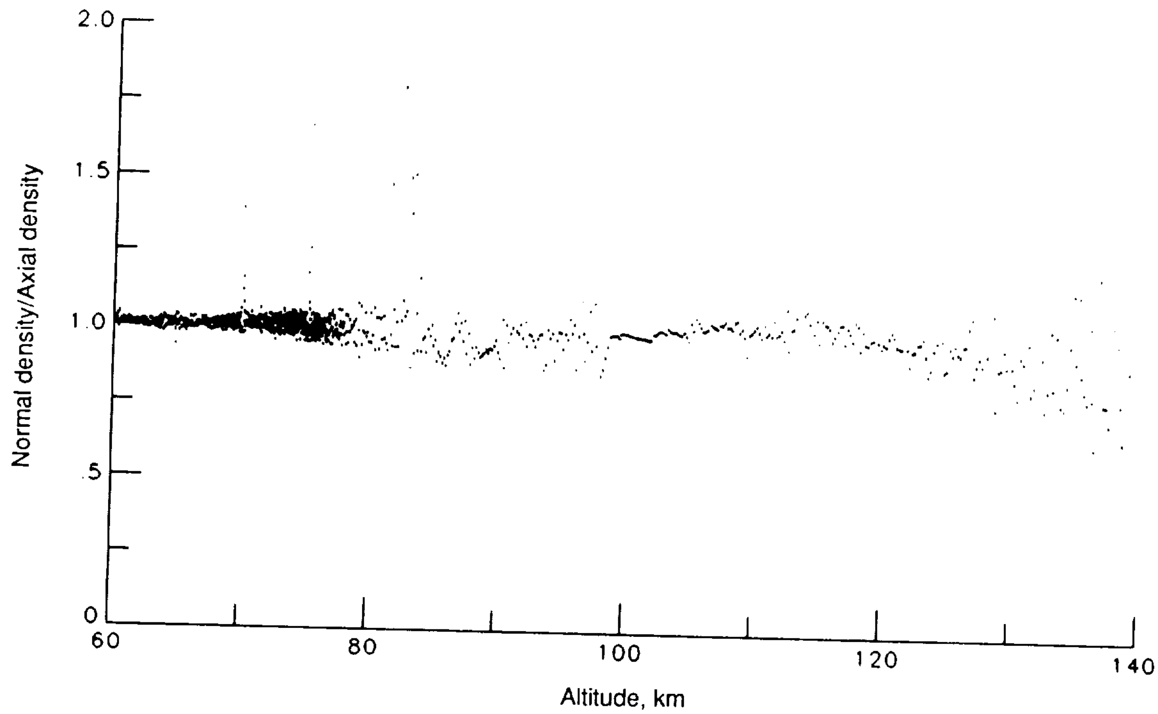


(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

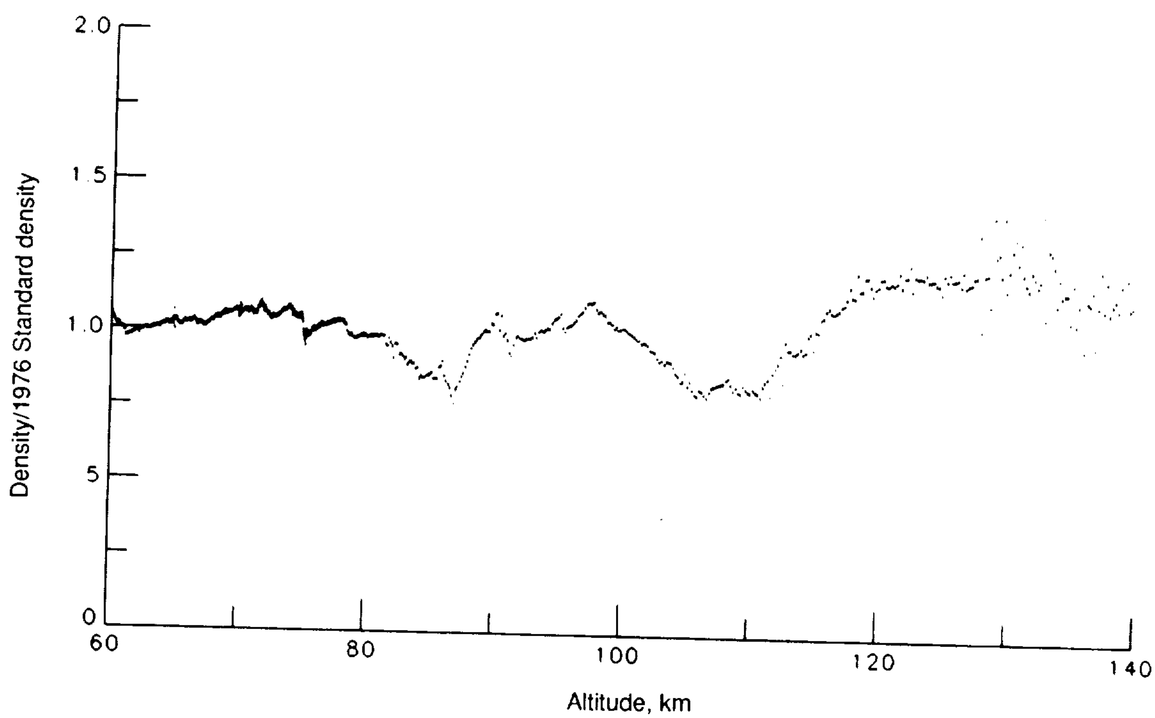


(b) Profile of calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model profile.

Figure 64. Density analysis results for STS-41B.

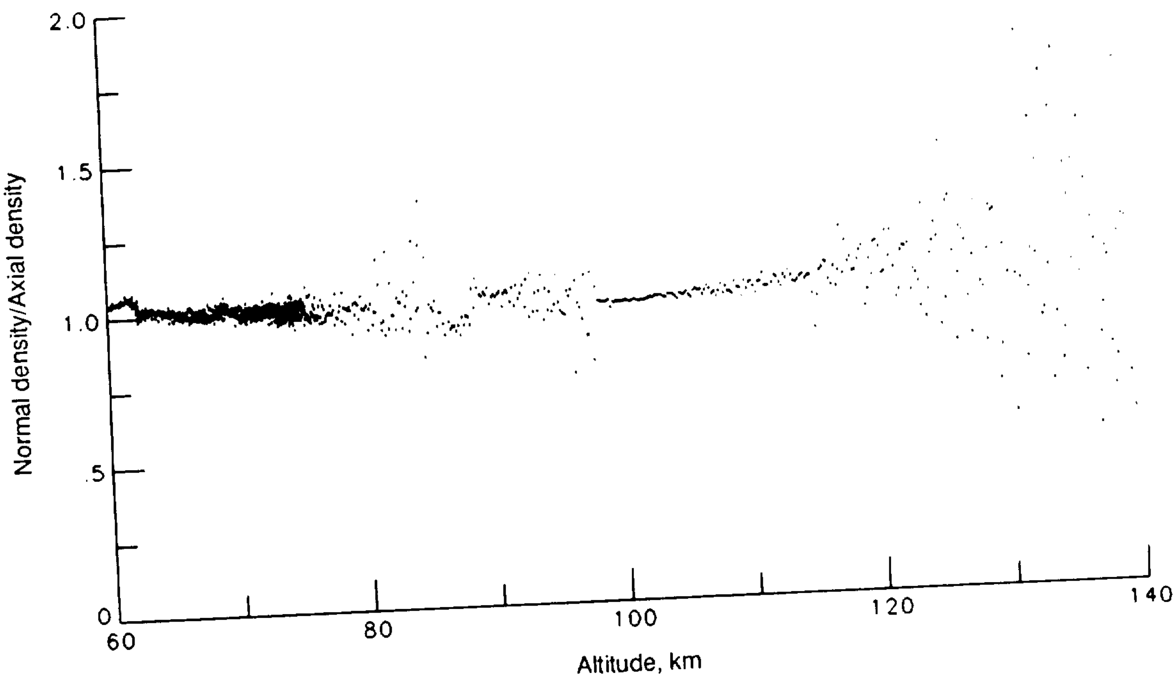


(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

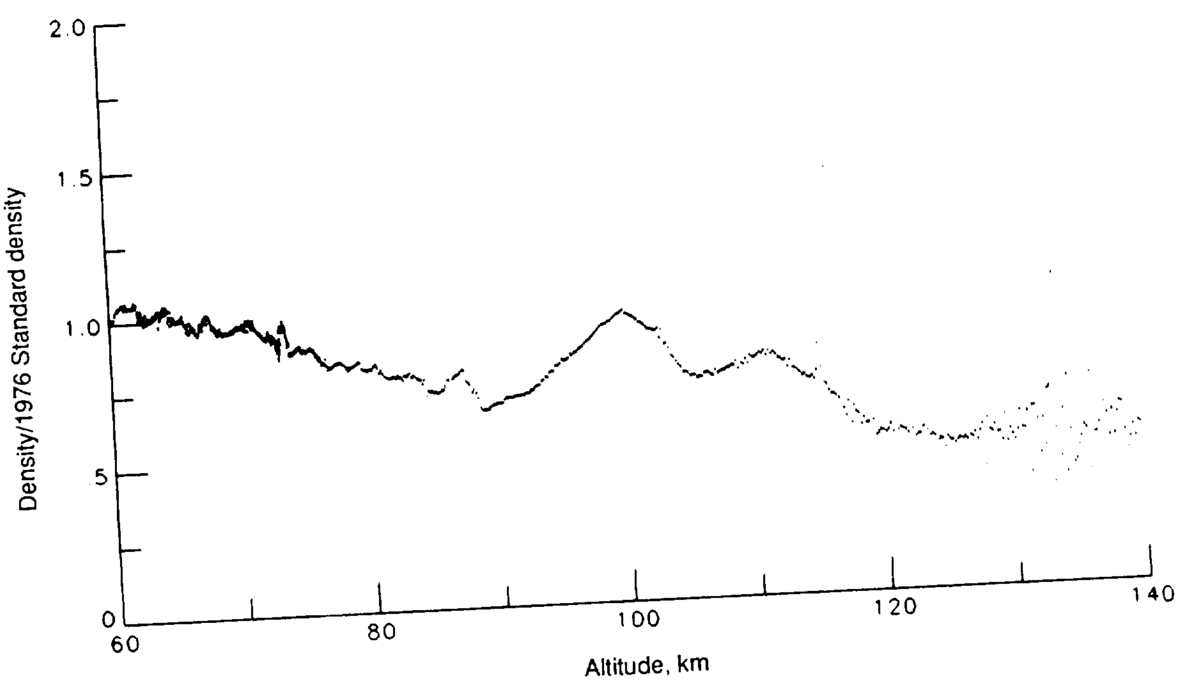


(b) Profile of calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model profile.

Figure 65. Density analysis results for STS-41C.

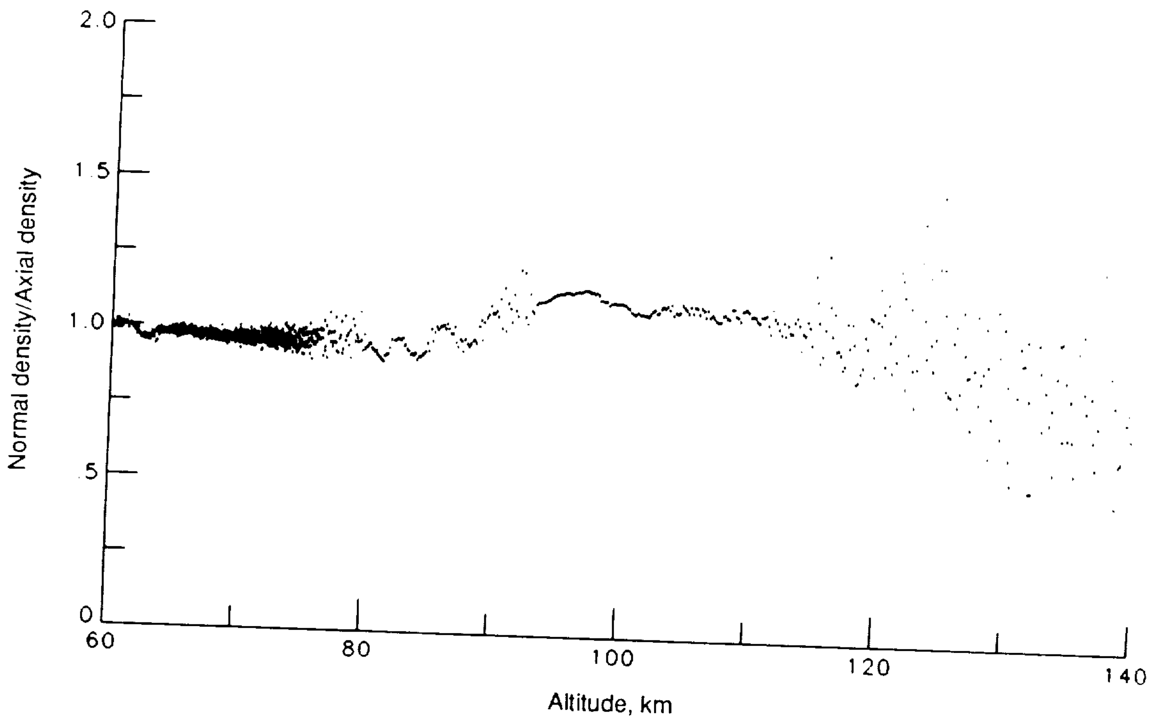


(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

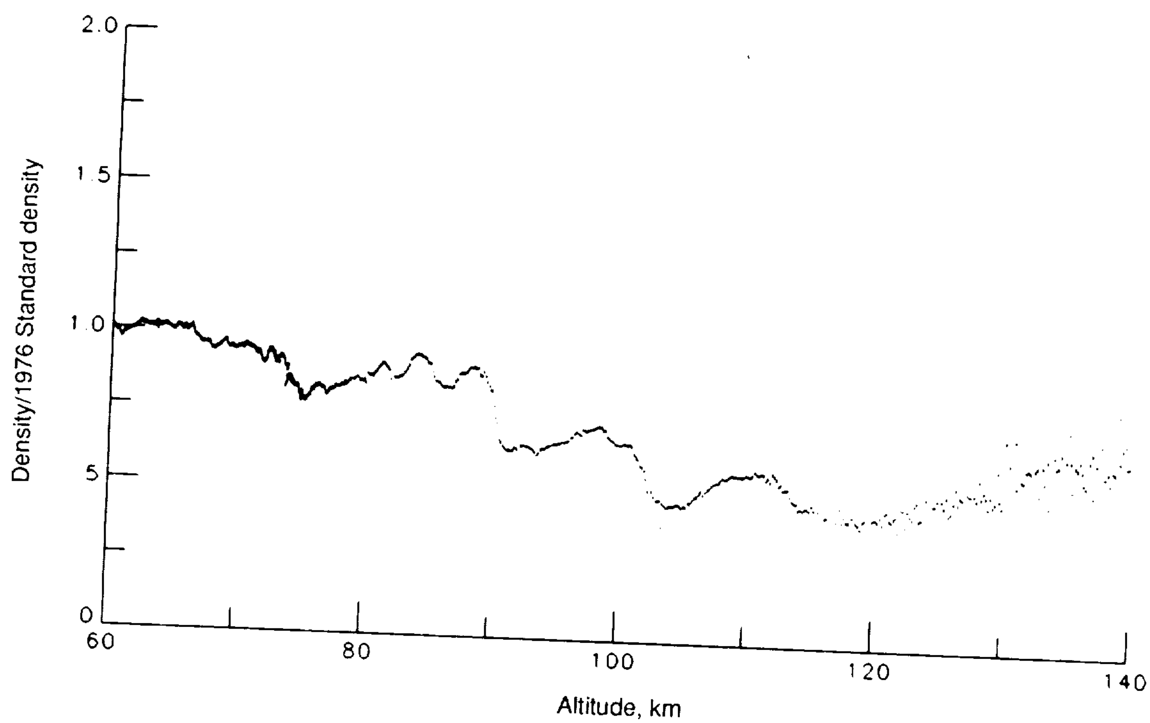


(b) Profile of calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model profile.

Figure 66. Density analysis results for STS-51B.

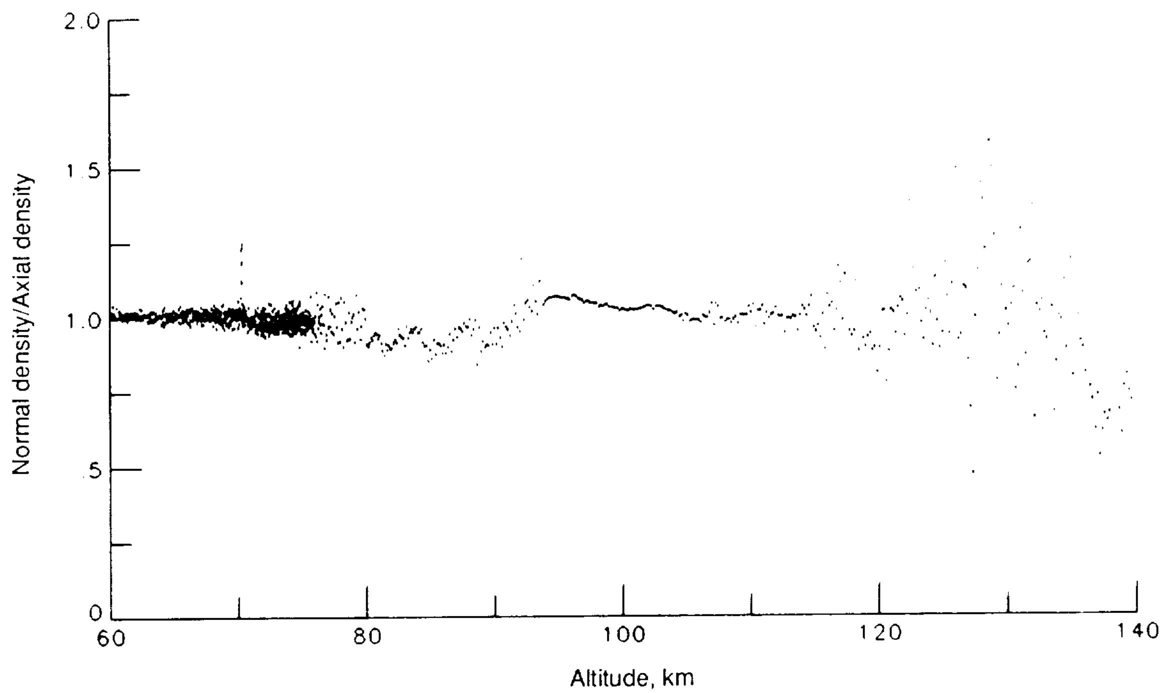


(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

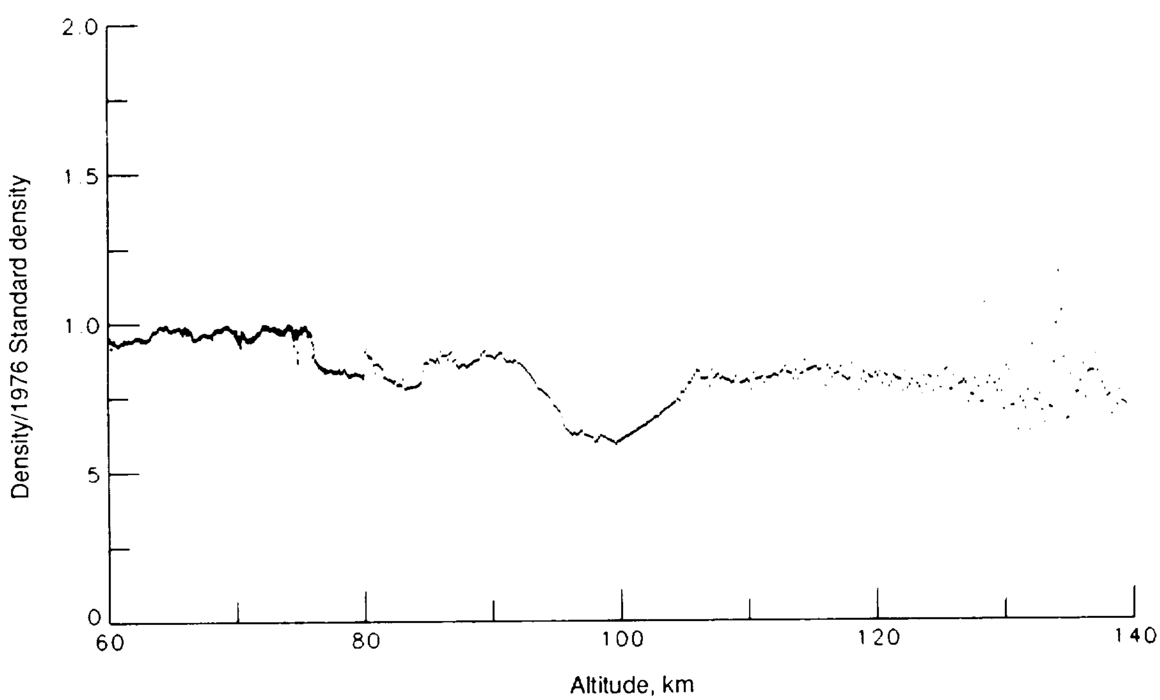


(b) Profile of calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model profile.

Figure 67. Density analysis results for STS-51F.

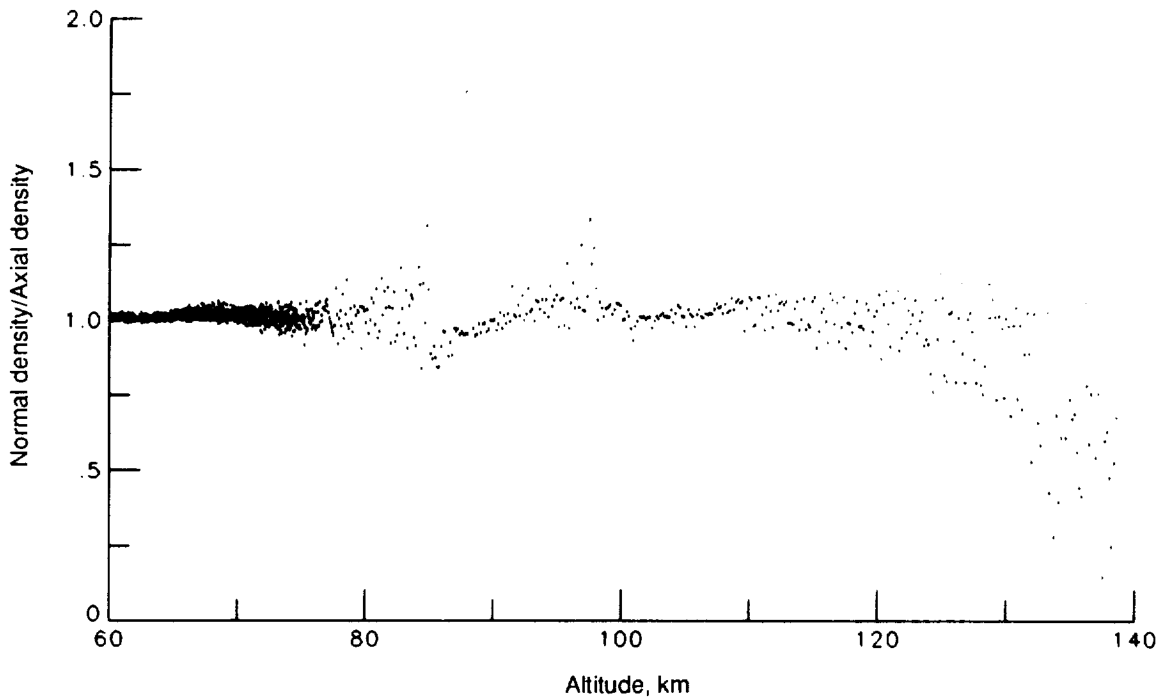


(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

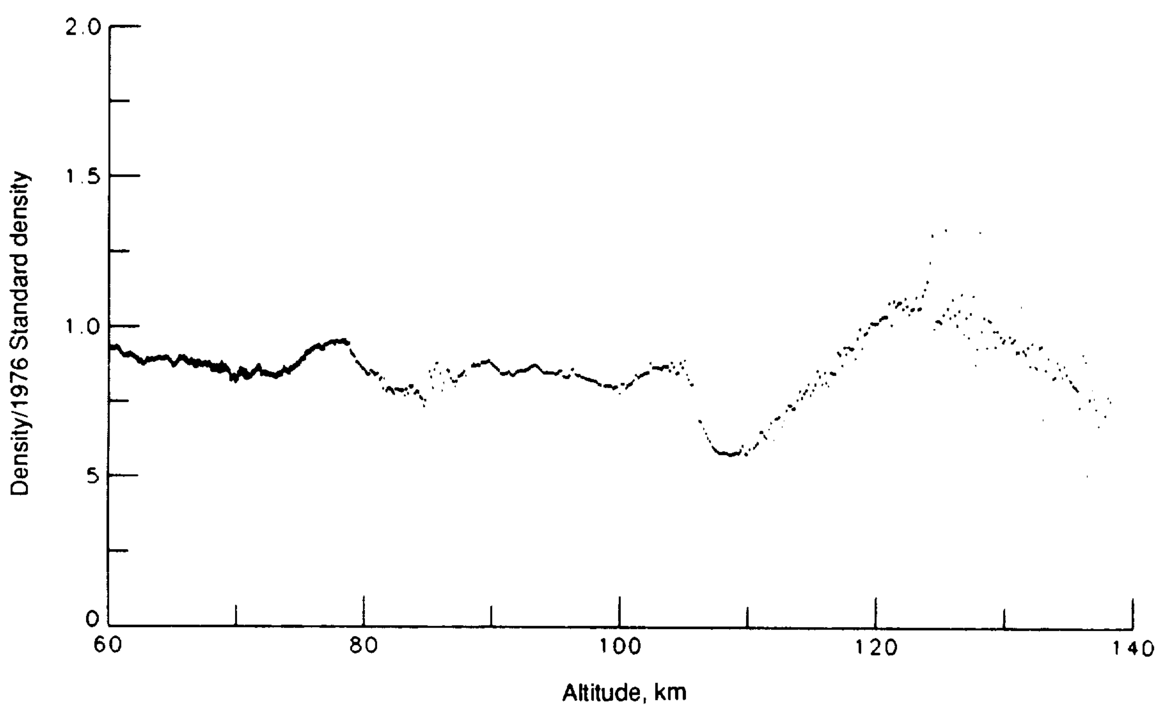


(b) Profile of calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model profile.

Figure 68. Density analysis results for STS-61A.

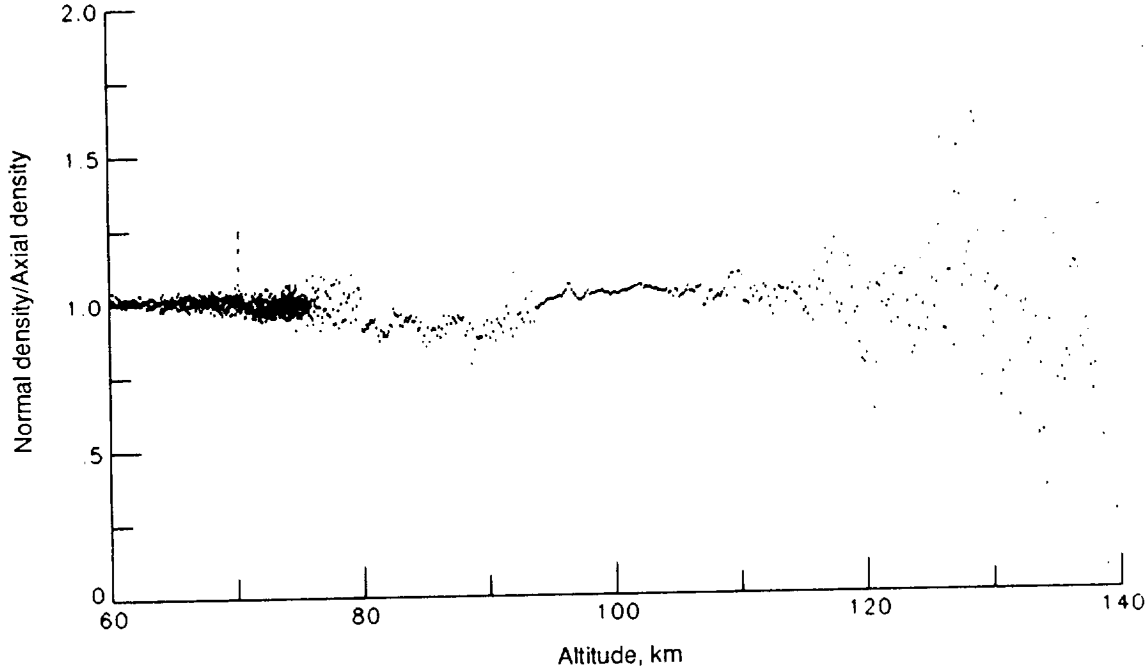


(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

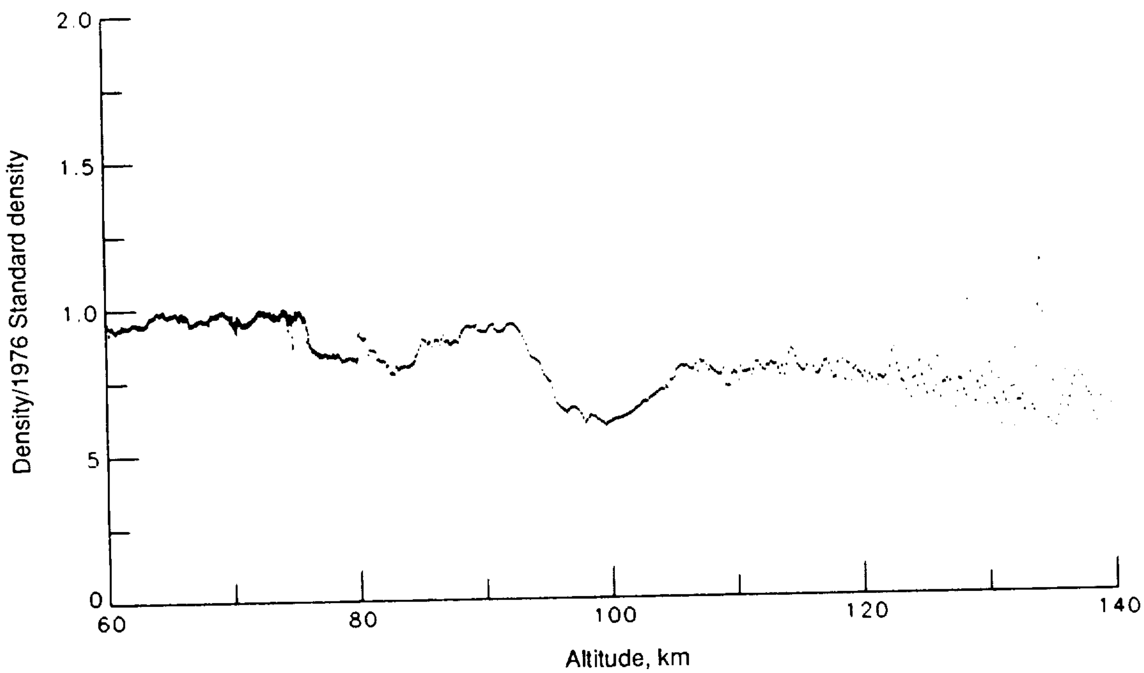


(b) Profile of calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model profile.

Figure 69. Density analysis results for STS-61C.

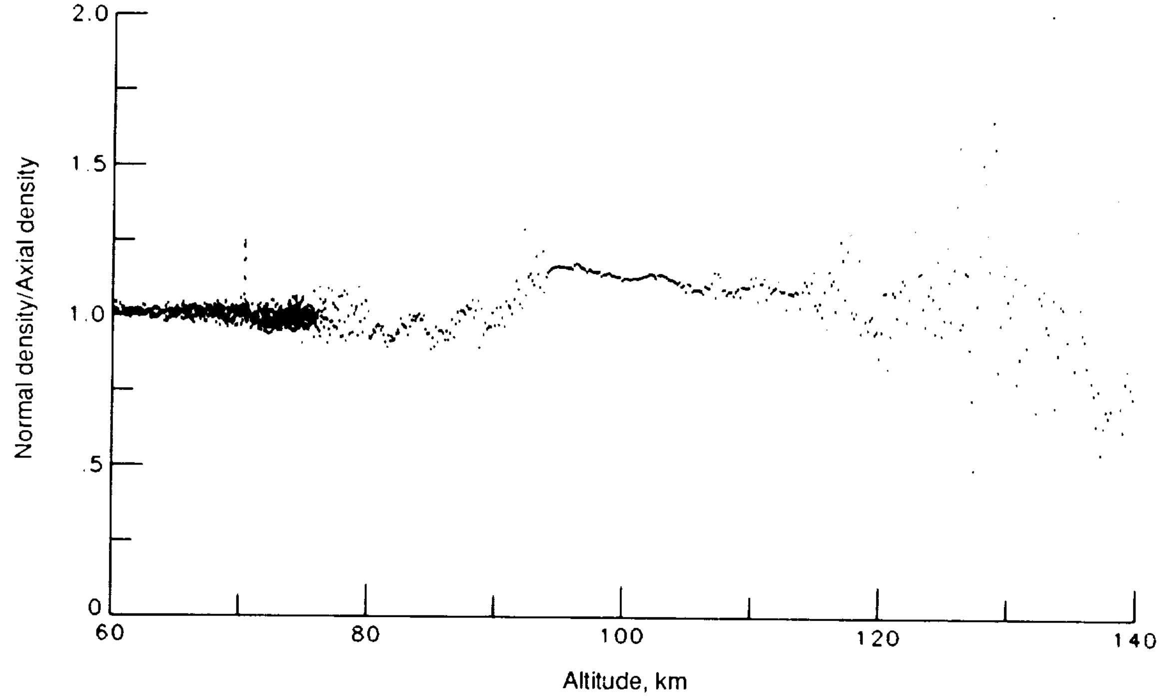


(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

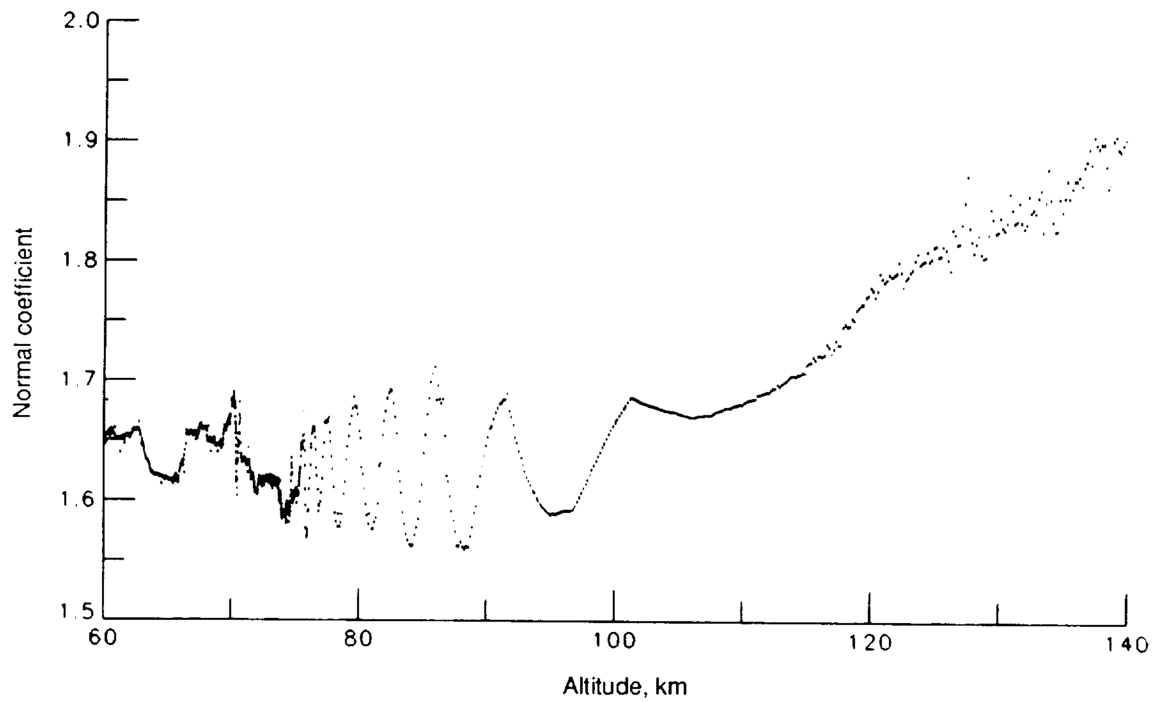


(b) Calculated density normalized to 1976 U.S. Standard Atmosphere (ref. 1) model estimate.

Figure 70. Density analysis results for STS-61A simulating correction of -1° misalignment.

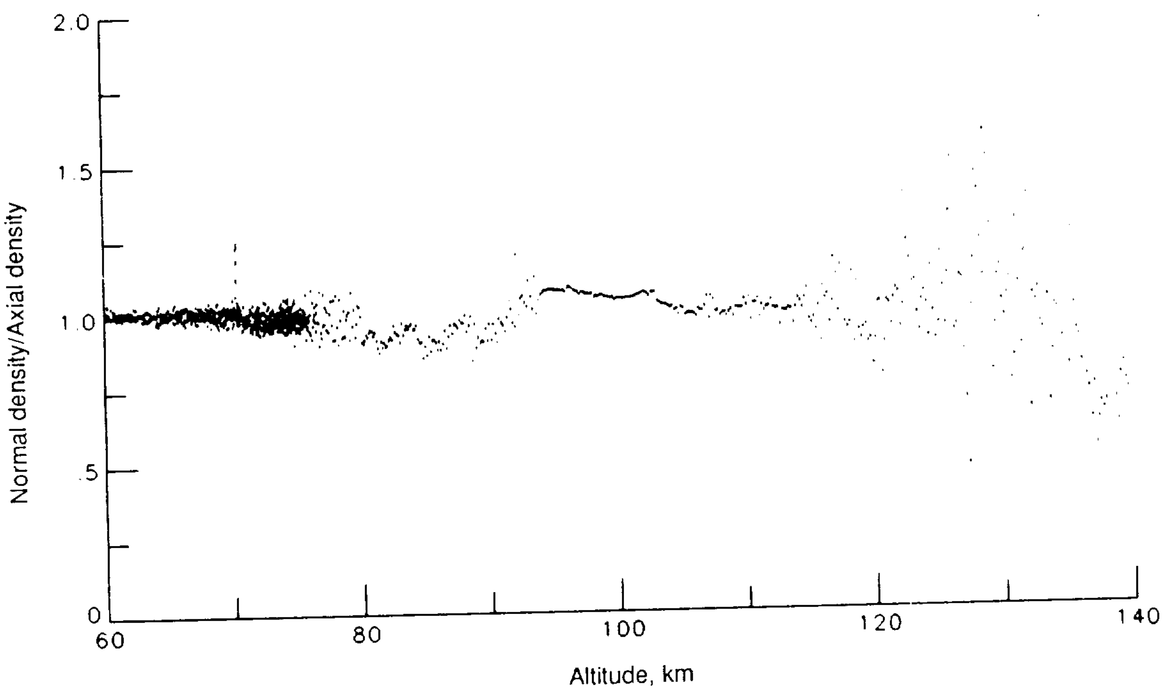


(a) Profile of ratio of calculated normal component of density to calculated axial component of density.

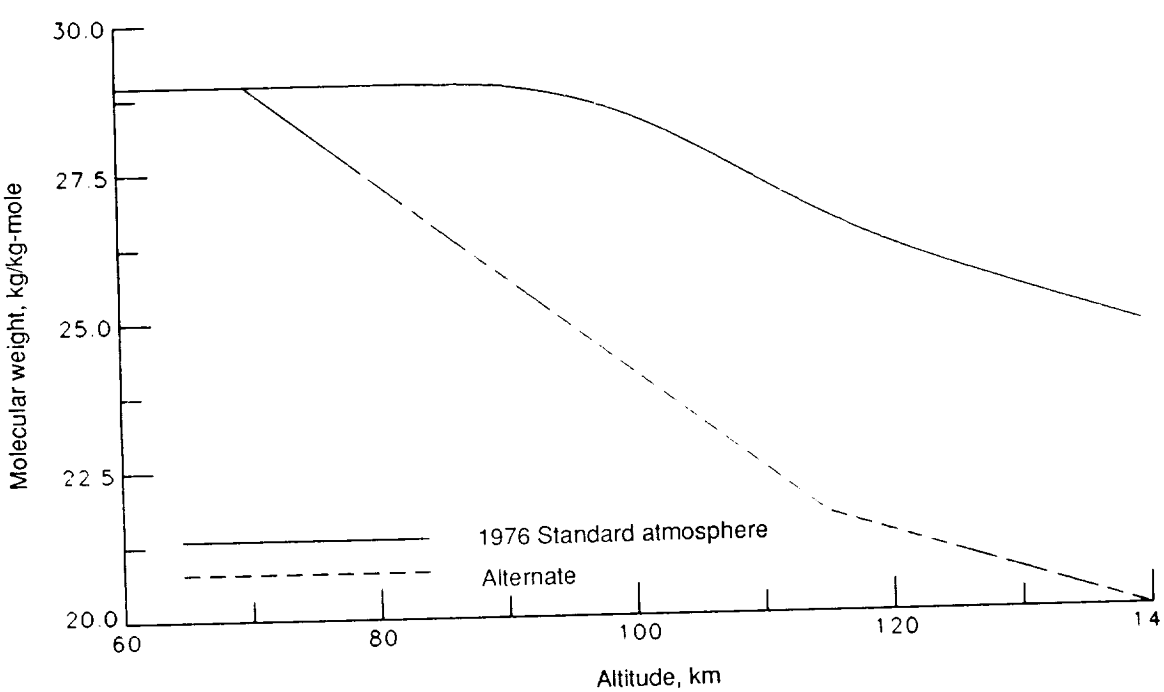


(b) Profile of normal coefficient.

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.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE April 1992	3. REPORT TYPE AND DATES COVERED Reference Publication

4. TITLE AND SUBTITLE
The High Resolution Accelerometer Package (HiRAP) Flight Experiment Summary for the First 10 Flights

5. FUNDING NUMBERS
WU 506-48-11-01

6. AUTHOR(S)
Robert C. Blanchard, K. T. Larman, and M. Barrett

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23665-5225

8. PERFORMING ORGANIZATION REPORT NUMBER
L-16900

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA RP-1267

11. SUPPLEMENTARY NOTES
Blanchard: Langley Research Center, Hampton, VA; Larman and Barrett: Lockheed Engineering & Sciences Company, Hampton, VA.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified
Subject Category 01

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
The High Resolution Accelerometer Package (HiRAP) instrument is a triaxial, orthogonal system of gas-damped accelerometers with a resolution of $1 \times 10^{-6}g$ ($1\ \mu g$). The purpose of HiRAP is to measure the low-frequency component of the total acceleration along the orbiter vehicle (OV) body axes 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. Instrument bias and orbiter mechanical system acceleration effects were incorporated into one bulk bias. The bulk bias was subtracted from the acceleration measurements to produce aerodynamic descent data sets for all 10 flights. The aerodynamic acceleration data sets were input to an aerodynamic coefficient model. The aerodynamic acceleration data and coefficient model were used to estimate the atmospheric density for the altitude range of 140 to 60 km and a downrange distance of 600 km. For 8 of 10 flights results from this model agree with expected results. For the results that do not agree with expected results, a variety of error sources have been explored.

14. SUBJECT TERMS
Rarefied-flow aerodynamics; Accelerometer measurements; Upper atmosphere

15. NUMBER OF PAGES
316

16. PRICE CODE
A14

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

NASA-Langley, 1992

National Aeronautics and
Space Administration
Code JTT
Washington, D.C.
20546-0001

Official Business
Penalty for Private Use, $300

SPECIAL FOURTH-CLASS RATE
POSTAGE & FEES PAID
NASA
PERMIT No. G27

NASA

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return