IN-35
77043
P.206

NASA
Technical
Memorandum

NASA TM-103573

A REAL-TIME RECURSIVE FILTER FOR THE ATTITUDE DETERMINATION OF THE SPACELAB INSTRUMENT POINTING SUBSYSTEM

By M.E. West

Structures and Dynamics Laboratory
Science and Engineering Directorate

February 1992

(NASA-TM-103573) A REAL-TIME RECURSIVE FILTER FOR THE ATTITUDE DETERMINATION OF THE SPACELAB INSTRUMENT POINTING SUBSYSTEM (NASA) 206 p CSCL 14B G3/35

N92-19920

Unclas
0077043

National Aeronautics and
Space Administration

George C. Marshall Space Flight Center

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-103573		

4. Title and Subtitle	5. Report Date
A Real-Time Recursive Filter for the Attitude Determination of the Spacelab Instrument Pointing Subsystem	February 1992
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
M.E. West	10. Work Unit No.

9. Performing Organization Name and Address	11. Contract or Grant No.
George C. Marshall Space Flight Center Marshall Space Flight Center, Alabama 35812	13. Type of Report and Period Covered Technical Memorandum

12. Sponsoring Agency Name and Address	14. Sponsoring Agency Code
National Aeronautics and Space Administration Washington, DC 20546	

15. Supplementary Notes

Prepared by Structures and Dynamics Laboratory, Science and Engineering Directorate.

16. Abstract

A real-time estimation filter which reduces sensitivity to system variations and reduces the amount of preflight computation is developed for the instrument pointing subsystem (IPS). The IPS is a three-axis stabilized platform developed to point various astronomical observation instruments aboard the shuttle. Currently, the IPS utilizes a linearized Kalman filter (LKF), with premission defined gains, to compensate for system drifts and accumulated attitude errors. Since the a priori gains are generated for an expected system, variations result in a suboptimal estimation process.

This report compares the performance of three real-time estimation filters with the current LKF implementation. An extended Kalman filter and a second-order Kalman filter are developed to account for the system nonlinearities, while a linear Kalman filter implementation assumes that the nonlinearities are negligible. The performance of each of the four estimation filters are compared with respect to accuracy, stability, settling time, robustness, and computational requirements. It is shown, that for the current IPS pointing requirements, the linear Kalman filter provides improved robustness over the LKF with less computational requirements than the two real-time nonlinear estimation filters.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Nonlinear Estimation Kalman Filter Attitude Determination Gyro Modeling	Unclassified – Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	208	NTIS

For sale by the National Technical Information Service, Springfield, VA 22161-2171

TABLE OF CONTENTS

PRECEDING PAGE BLANK NOT FILMED

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF TABLES

TECHNICAL MEMORANDUM

A REAL-TIME RECURSIVE FILTER FOR THE ATTITUDE DETERMINATION OF THE SPACELAB INSTRUMENT POINTING SUBSYSTEM

I. INTRODUCTION

The instrument pointing subsystem (IPS) is a three-axis stabilized platform which was developed to point observation experiments with stability and accuracy requirements beyond the capability of NASA's space transportation system. The IPS was developed by Dornier System under contract to the European Space Agency and was delivered to NASA in 1984 [1]. It was designed as a multipurpose pointing instrument with the capability to accommodate scientific instruments of various masses and configurations. In July and August of 1985, the IPS was flown aboard the shuttle to observe solar phenomena as part of the Spacelab-2 mission. The IPS quiescent performance for a typical objective during the Spacelab-2 mission was measured to be 0.05 arcsec accuracy and 0.5 arcsec standard deviation in line-of-sight (LOS) [2]. The IPS will be flown on the Astro-1 mission, and that mission will view various ultraviolet targets.

The IPS uses a multirate, multivariable digital control system. A 25-Hz control loop, which is the coarse pointing or gyro-only control pointing mode, utilizes gyro measurements in a proportional-integral-derivative (PID) feedback to reject the shuttle disturbances such as man motion and thruster firings. An accelerometer feed forward loop is also used in the 25-Hz loop to suppress disturbances.

The second control loop is a 1-Hz attitude determination filter (ADF) which is only operational during fine pointing. The ADF processes inertial measurements generated by an optical sensor package (OSP) to compensate for inherent gyro drifts, biases, and the resulting attitude errors accumulated during gyro-only control. Three star trackers provide the OSP's inertial measurement outputs from the position of an identified star in each tracker's field-of-view (FOV). The choice of IPS attitude representation and inclusion of second-order dynamics in the system models introduces nonlinearities into the closed loop which complicates the ADF task. Due to IPS's data systems memory constraints, the ADF is a suboptimal estimation filter which employs premission derived time-varying gains. The precomputed gains are generated using expected initial conditions, which can result in nonoptimal overshoot and settling time for actual attitude errors. Nominally, the ADF processes attitude measurements for the three star trackers, hence the loss of a star tracker's measurements results in erroneous estimations. The implementation of a real-time ADF allows the filter to adapt to actual conditions and reduces the amount of preflight computation.

The performance of four estimation methods are compared in this report. The first method is the current IPS estimation filter which is a linearized Kalman filter (LKF) [3]. The LKF linearizes the filter states about a prespecified state. The second method is the linear Kalman filter (KF) [4] in which it will be assumed that the nonlinearities are negligible. The third estimation filter is an extended Kalman filter (EKF) in which system states are linearized about the estimated states. The final estimation filter that will be presented is a second-order Kalman filter (SOKF) [5]

which includes second-order terms in the linearization process, unlike the LKF and EKF which include only first-order terms. The performance of each of these filters will be simulated, and their relative merits compared with respect to accuracy, stability, settling time, computational requirements, and robustness.

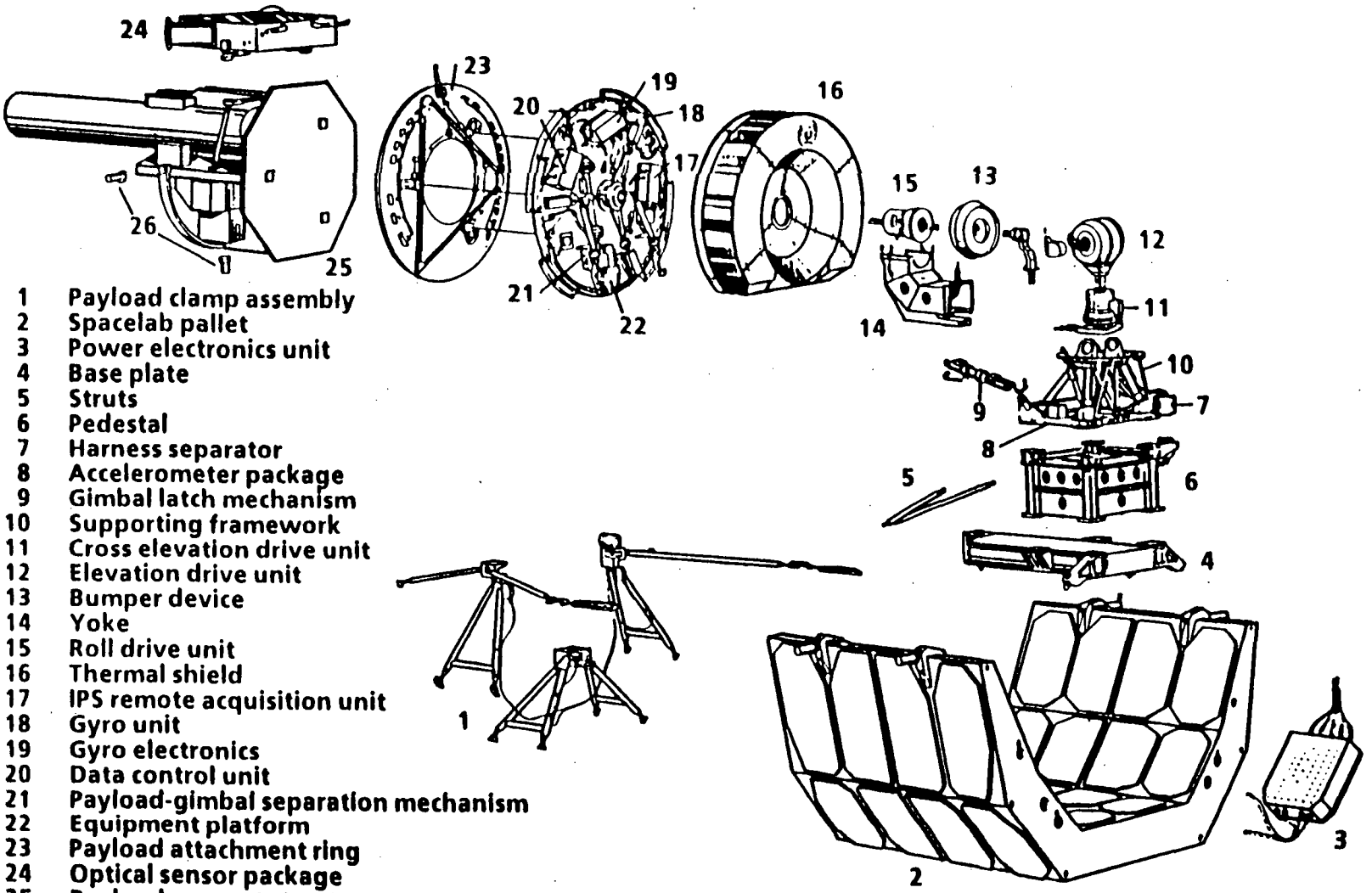
II. SYSTEM DESCRIPTION

The IPS is a complex inertial pointing mount. Its support structure has integrated actuator and sensor hardware along with the necessary electronics, data processors, and software to allow pointing to, and tracking of, a variety of astronomical targets such as stars, gas clouds, planets, and comets. It uses a unique three-axis gimbal design which allows scientific instruments to be attached to the extremities of payloads, as opposed to the typical gimbal design which attaches to the center-of-gravity of the payload. The IPS gimbal configuration is often referred to as an "inside-out" configuration and can result in nonorthogonal gimbal geometry for some IPS attitudes. The IPS gimbal structure design allows for a large range of motions and for a variety of payload masses and inertias [1]. An exploded view of the IPS components, taken from the "Instrument Pointing System Spacelab Systems Training Manual," [6] is presented in figure 1.

A. Operational Description

IPS operations were divided into separate modes in order to meet the variety of desired pointing and tracking requirements and the software memory constraints. The IPS has four operational modes which include 18 subfunctions. The operational modes are activation/deactivation, stellar, solar, and Earth, and at least two of these modes are typically employed in each IPS mission. During launch and landing the IPS payload is physically separated from the gimbal structure in a stowed configuration so as to reduce loads on the structure. The activation/deactivation mode attaches the payload to the gimbals, activates the electronics and measurement instruments, and erects the IPS to within the IPS cone of operation with a 90° elevation maneuver. A 60° cone about the 90° elevation and the 0° cross-elevation gimbal angle configuration specifies the operational range of the IPS. The stellar, solar, and Earth operational modes have the IPS pointing to a guide star, the Sun, or the Earth, respectively. The results of simulations presented subsequently are for the Astro-1 configuration, which is a stellar mission. Therefore, a brief description of the IPS operations necessary to obtain a stellar objective follows.

Once the IPS has been maneuvered so that it is within the cone of operation, the stellar pointing mode is enabled. Because of memory constraints, the stellar mode is divided into seven independent subfunctions which are better known as memory configurations (MC). Whenever a desired pointing objective is more than 4° from the current attitude, the slew MC is utilized to rotate the IPS payload to the proper attitude. A slew maneuver can be accomplished by either a gimbal angle command or an inertial attitude command. All slew maneuvers are performed with gyro-only feedback control. Once the pointing objective is within the star trackers' FOV, one of three basic MC's can be used. Two star identification configurations which have different probabilities of success, and therefore different time requirements, comprise the first basic configuration. An



1 Payload clamp assembly
2 Spacelab pallet
3 Power electronics unit
4 Base plate
5 Struts
6 Pedestal
7 Harness separator
8 Accelerometer package
9 Gimbal latch mechanism
10 Supporting framework
11 Cross elevation drive unit
12 Elevation drive unit
13 Bumper device
14 Yoke
15 Roll drive unit
16 Thermal shield
17 IPS remote acquisition unit
18 Gyro unit
19 Gyro electronics
20 Data control unit
21 Payload-gimbal separation mechanism
22 Equipment platform
23 Payload attachment ring
24 Optical sensor package
25 Payload support structure
26 Payload attachment flanges

Figure 1. Exploded view of IPS components.

optical sensor package calibration operation, which is usually only scheduled on the first acquisition, is the second basic configuration. The last basic MC is comprised of combinations of fine pointing, manual offset pointing, and scanning.

For Astro-1 the fine pointing and manual pointing control configurations will be the primary mode of operation for periods of astronomical observations. The hardware and software necessary to accomplish these complex operations will be described in the next section.

B. Hardware Description

The primary hardware component which makes a three-axis pointing mount possible is the IPS gimbals structure assembly (GSA) [6]. The GSA contains three identical torque drive units (DU's) [7], whose outer housings and inner shafts form the gimbal mechanism depicted in figure 2; one DU for each rotational degrees-of-freedom (DOF). Each DU contains two redundant frameless, brushless, direct current (dc) torque motors that generate torque on the motor housing with respect to the DU shaft. The elevation drive unit (EDU) shaft is attached to the Spacelab support structure, and the EDU housing is connected to the cross-elevation drive unit (XDU) shaft. A yoke structure connects the XDU and the roll drive unit (RDU) housings. The yoke was designed to produce a gimbal geometry which results in the three axes of rotation intersecting at a point on the elevation shaft. Since the RDU housing is attached to the yoke, the RDU shaft is connected to the equipment platform and is free to rotate the payload about the IPS LOS. A single-speed and multispeed resolver set is provided with each torque motor. The single-speed resolver outputs relative attitude measurements used by the control system to generate attitude commands and coordinate transformations. Commutation of the torque motors is provided by the multispeed resolver. Each DU is capable of providing 30 Newton-meters (Nm) maximum torque, but the torques are limited to a maximum of 27.12 Nm in LOS and 11.16 Nm in the roll axis [2]. The IPS is constrained to a 30° half-cone angle for observation, but is capable of performing $\pm 180°$ roll orientations.

Mounted on the IPS lower support framework is an accelerometer package (ACP) [8] consisting of three analog force pendulums in an orthogonal configuration. The ACP outputs are filtered by a high-pass analog filter to remove alternating current (ac) coupling, and a low-pass filter to reduce aliasing due to sampling. The output of the low-pass filter is sampled and held at a 50-Hz frequency before being acquired by the control unit. The controller utilizes the ACP measurements in a feed forward path to assist in suppressing the shuttle vibration environment.

A three-axis strap-down inertial reference unit, manufactured by Feranti, is mounted on the underside of the equipment platform above the RDU. The gyro package (GP) [9] uses four single-DOF pulse-balanced rate integrating gyroscopes in the rate mode. The four wheels are arranged so that three are orthogonal while the fourth is skewed to provide redundancy in each axis if a single wheel were to fail. The GP outputs a 16-bit data word for each axis at 100 Hz. The GP will saturate, but not overflow, when the IPS or the shuttle rates reach 3°/s. The delta angle output by the GP is read by the digital controller every 10 ms and is the primary inertial reference.

The final inertial measurement unit is the OSP [10] which consists of three fixed head star trackers (FHST's). Each FHST uses a photomultiplier tube which calculates a star's inertial coordinates from the deflection currents required to direct the incoming photons. A boresight FHST is

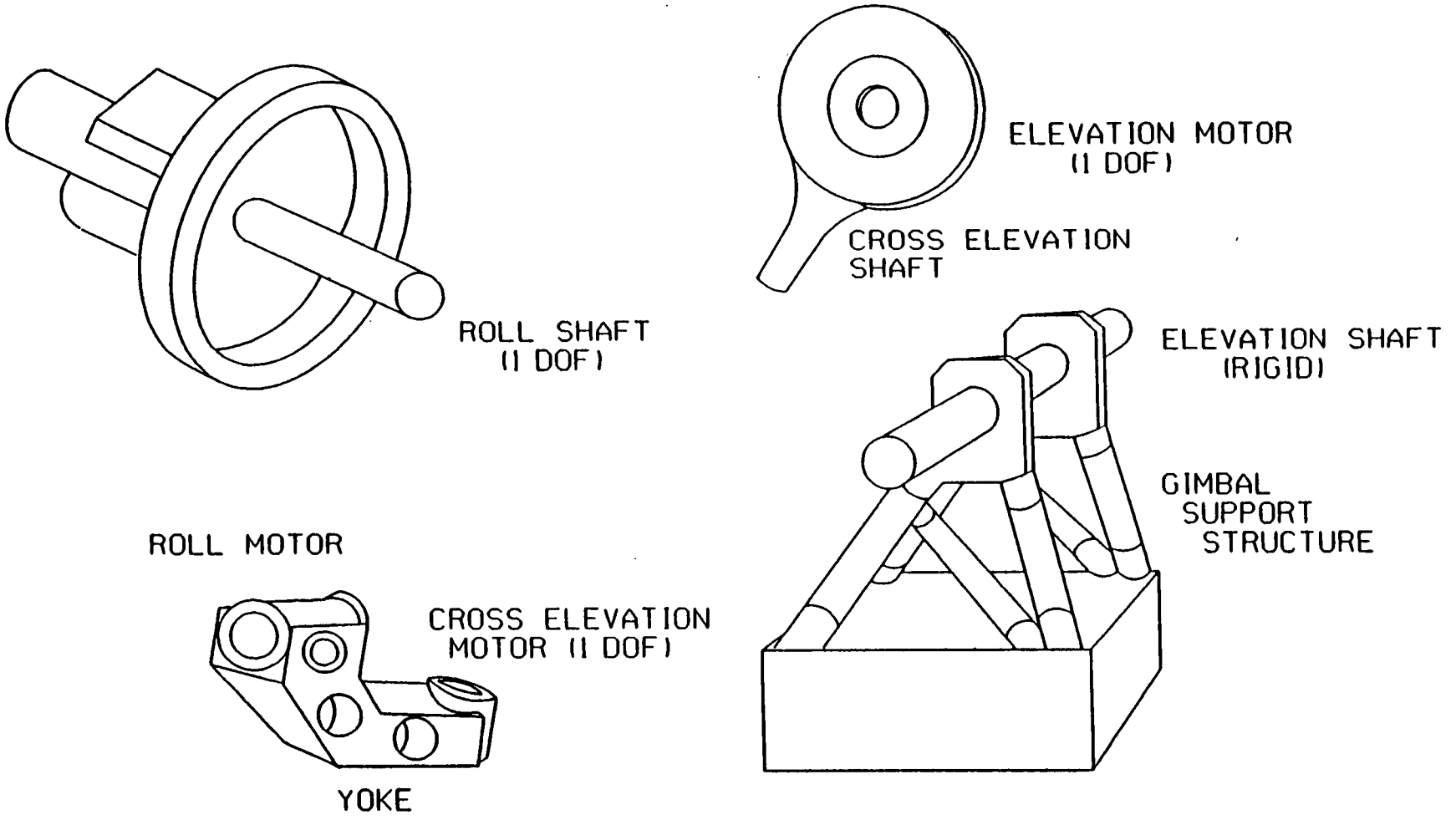


Figure 2. IPS gimbal structure.

aligned along the IPS platform LOS while the other two FHST's are skewed an angle of $\pm\alpha$ from the boresight tracker. The angle α is dependent upon whether the IPS mission is to be solar (α = 45°) or stellar (α = 12°). Each FHST is capable of outputing y and z focal plane coordinates for one or two stars. The FHST measurements are averaged 18 times over a 1-s interval before being sent to the control unit to be processed by the ADF. The two skew trackers provide the observability for the IPS roll attitude and roll drifts.

The last IPS hardware system discussed herein is the data processing system. Three computer systems and their respective interfaces comprise the IPS automatic data processing system. The data control unit (DCU) is a 16-bit fixed point processor which performs the primary 25-Hz control computations. An active thermal control system is also implemented in the DCU. The Spacelab command and data management system (CDMS) provides IPS operational mode definition, IPS command generation, and telemetry operations. A subsystem computer (SSC) of the CDMS performs the 1-Hz ADF processing. An experiment computer (EC) provides data processing for the scientific instruments. The EC can also utilize measurements from scientific instruments to generate attitude commands to the DCU or sensor substitution measurements to the SSC. This completes the hardware description of the IPS.

C. Software Description

The software tasks required to achieve the complex IPS missions include operation mode sequencing, command generation, control and stabilization, telemetry, experiment data processing, and data storage. The tasks are divided among the three computer systems described in the hardware description section. The operational mode sequence is specified by the user and executed by the CDMS computer. Loading of the required MC and generation of proper controller commands are tasks of the mode sequencing software. One advantage of different MC's is the capability of varying the control gains, depending upon the operational mode, while using a fixed controller structure.

Control and stabilization software is executed for all IPS operational modes. The IPS uses an adaptable multirate, multivariable digital controller. Control parameters can be varied to satisfy the different requirements of the operational modes by changing the MC's. The primary control loop [11], called the fast loop, is executed in the DCU at 25 Hz. It is comprised of a PID feedback loop and an acceleration feed forward path for each gimbal axis. Figure 3 presents a simplified block diagram of the continuous plant model and the discrete IPS PID feedback controller, where the "^" denotes the discrete equivalent of the continuous parameter. The IPS is modeled by a second-order system with negligible damping and stiffness. Therefore, to obtain a system with a desirable time response, a derivative feedback term is utilized to introduce damping into the system. To meet the pointing and stability requirements, the derivative feedback gain was designed to produce a system behavior between a critically damped and an overdamped system. Also to meet the fine pointing requirements, an integral feedback path is implemented. The integral feedback increases the IPS system order by one, which enables the IPS control system to follow an acceleration input with zero steady-state error. The fast loop is executed in all operational modes except during emergency conditions, when an analog controller is used to return the IPS to the separated launch and landing configuration. For an IPS slew maneuver, the only control requirements are stability and an attitude end condition of less than 4°. Typically, for the slew maneuver, the integral feedback gain is small to improve the response of the maneuver.

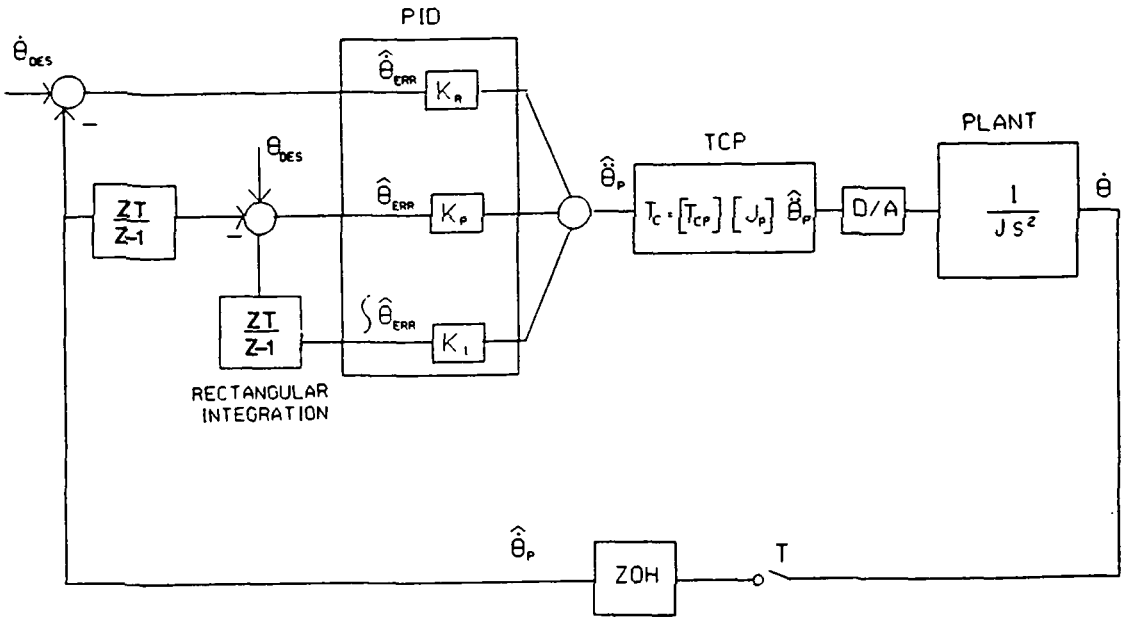


Figure 3. Simplified IPS feedback control block diagram.

During the fine pointing operations the 1-Hz ADF, which provides attitude updates and system drift estimates to the fast loop, is executed. Control system requirements for the fine pointing operations include high pointing stability and accuracy with small rate errors. Therefore, in contrast to the slew maneuver gains in the fast loop, a significant integral term is used in the slow loop along with the proportional and derivative terms to improve steady-state error performance. Table 1 gives the Astro-1 pointing requirements used to design the control parameters. The fast loop and ADF structures will be discussed next.

The 25-Hz software, executed in the DCU, is divided into two 50-Hz minor cycles. Interrupts are not used by the DCU to synchronize read/write operations with the sensors and actuators. Instead, scheduling of the data transfers was accomplished by separating the read/write

Table 1. IPS pointing accommodations.

	Requirement	Design Goal
Accuracy		
Line of sight (2 axes)	7 arcsec	7 arcsec
Roll	20 arcmin	1 arcmin
Stability		
Quiescent LOS	None	0.75 arcsec (rms)
Roll	None	1.8 arcsec (rms)
Crew Motion LOS	None	9.1 arcsec (peak)
VCRS thruster firing LOS	None	3.3 arcsec (peak)

operations by a specific amount of DCU code. The time required to execute the DCU code determines when the input/output operations occur. The fast-loop control loop, which is presented in figure 4, is dependent upon gyro measurements for IPS stability and control. The DCU reads the gyro package outputs every 10 ms, where three gyro reads occur in the first minor cycle and the remaining read is performed in the second. In each minor cycle, two sequential gyro measurements are averaged and then divided by 20 ms to form a rate measurement. IPS uses pulse-rebalance rate integrating gyros in a rate mode, which means the outputs of the gyros are the delta angles sensed since the last measurement. Therefore, in order to produce a rate measurement, the averaged gyro measurements are divided by the total time elapsed for both measurements. The resultant rate signal in each minor cycle is transformed from an internal gyro coordinate system to the IPS platform coordinate system, the transformation being dependent upon which gyro wheels are operational. Then, in each minor cycle the 50-Hz rate measurements are processed in a normalized second-order prefilter with a 4-Hz bandwidth to suppress high frequency signals not suppressed by the averaging operation. This is done in order to eliminate aliasing effects, which are caused by overlapping of high frequency information about the Nyquist frequency (one half of the sample frequency whenever sampling occurs). Only in the first minor cycle prefilter processing is an output equation computed, thereby producing a 25-Hz prefilter output. The prefilter states are passed between the two minor cycle prefilter processes. Shown in the upper right corner of figure 4 is an expanded view of the prefilter processing. Once the 25-Hz rate measurement has been obtained, the signal is compensated by a system drift estimate, which is the output of the ADF.

Division of the control algorithm into two minor cycles is shown by the dashed line in figure 4, where the blocks inside the dashed line comprise the calculations executed during the second minor cycle. The first minor cycle begins with the reading of the fourth gyro measurement, averaging with an initial third gyro measurement, and continues with the first 50-Hz aliasing filter processing to form the drift compensated rate measurement. The desired rate $\dot{\theta}_{DES}$ generated in the CDMS is compared to the rate signal to produce a rate error $\hat{\theta}_{ERR}$. The LOS rate errors are then processed by a normalized seventh-order digital filter. Three second-order filters and one first-order filter in state space representation are cascaded to form the seventh-order filters in the elevation and cross-elevation axes. The filters were designed to achieve stability with an approximate 1-Hz bandwidth. To improve attenuation of high gain structural vibrations, the seventh-order filters include notches at approximately 3 Hz. In order to obtain desired attenuation of structural resonances, the frequency of the vibration modes must be known accurately. If the actual resonant frequency does not correspond to the modeled frequency, or if the frequency varies due to changes in mass or inertia, then degradation in controller response may be observed. For the Astro-1 mission, the analytical rate loop gain and phase margins are approximately 2 decibels (dB) and 20 Hz, respectively, for the worst IPS inertia configuration. Outputs of the LOS rate filter along with the unfiltered roll channel rate error form the derivative input to the PID control law.

Initial integral and proportional signals, which are generally calculated in the second minor cycle of the previous 25-Hz interval, are combined with the derivative signal to form a control command in the platform coordinate frame. The command is transformed into the gimbal coordinate system and is decoupled to produce commands in the individual gimbal axes. When a nonzero cross-elevation gimbal angle is present, the IPS gimbal configuration becomes nonorthogonal, which results in cross coupling between the roll and elevation gimbals. In order to prevent motion in the undesired coupled axis, a command is generated to resist the unwanted motion. The decoupling is accomplished via a matrix T_{cp} which transforms the desired control

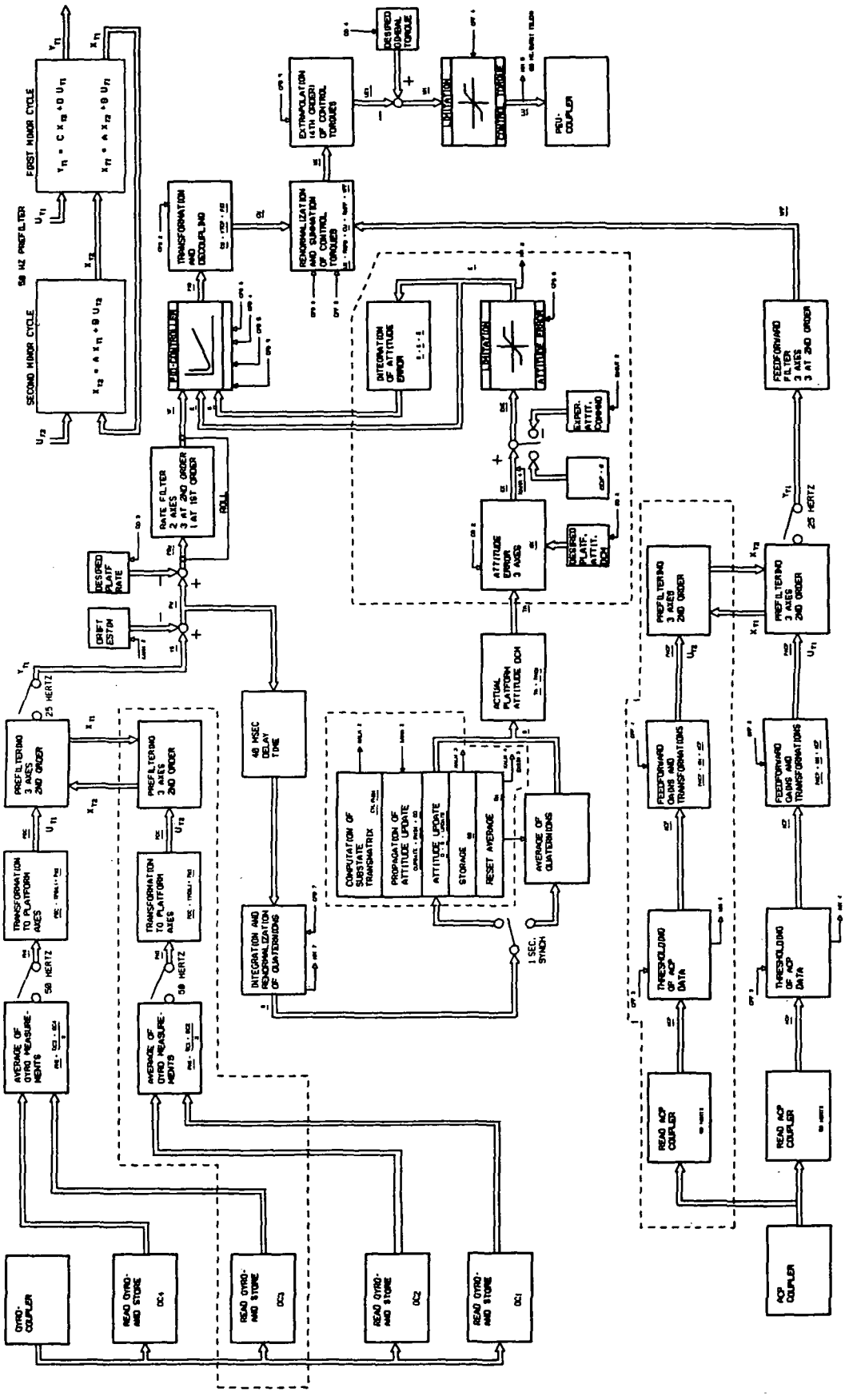


Figure 4. IPS fast-loop controller.

command from the IPS platform coordinate system (P) to the actuator coordinate system (C). To produce a torque signal, the command is multiplied by a constant IPS inertia matrix (J_p) for the 90/0/0 gimbal configuration. The inertia matrix consists of all the components above the RDU housing and is calculated about the IPS center of rotation. Since the actual IPS inertia matrix is a function of the gimbal configuration, use of a constant inertia matrix effectively increases the IPS feedback gains when a significant cross-elevation attitude is commanded. Therefore, pointing objectives which include significant cross-elevation attitudes exhibit degraded stability performance. The inertia matrix and the similarity transformation matrix are combined in the 3×3 matrix T_{cp} shown in figure 3. Finally, the PID control torque is summed with a signal from the acceleration loop before being applied to the torque motors.

The acceleration feed forward path receives inputs from the ACP, which is located on the IPS support structure. Accelerations are measured at 50 Hz and then transformed to the IPS platform coordinate system. A 50-Hz prefilter with a 2-Hz bandwidth is used to prevent aliasing effects when the acceleration measurements are processed by the controller at 25 Hz. The acceleration prefilters are of the same form as the rate prefilters, where a 50-Hz signal is processed in both minor cycles with a 25-Hz output in only the first minor cycle. The accelerations are then processed by three second-order filters to stabilize the accelerometer path before they are combined with the PID torque commands. The first gyro read and store is performed during the accelerometer filter calculations. The combined torque commands are output to an electronics unit to be applied to the torque motors.

After the torque commands are output, the fast loop position and integral error signals are computed. As a result, an inherent 40 ms delay is present in the controller design. This delay was incorporated into the simulations presented here. To determine the pointing error, the IPS inertial attitude must be obtained from the rate measurements. Euler's kinematic representation, which is presented in section IV, is employed to determine a quaternion rate. The quaternion rate is numerically integrated using a rectangular method to form the IPS inertial quaternion attitude. The rectangular integration method was chosen in order to meet DCU timing and memory constraints. A rectangular numerical integration method approximates a function's definite integral by multiplying a specified time interval by a value of the function at a fixed time during the integration interval and summing the result with the integral of the previous interval. The rectangular integration method will only produce the correct definite integral of a constant function. If the function to be integrated is a ramp or higher order function, then an integration error will exist. During fine pointing, the IPS rates are small, and the use of rectangular integration results in negligible errors. However, during IPS and orbiter maneuvers, integration of the measured time-varying rates produces significant integration errors. The integration error produced during the slew maneuvers will contribute to the attitude error which is present at the beginning of an observation period.

Once the attitude quaternion has been obtained by rectangular integration of the quaternion rates, the attitude quaternion is then renormalized. At this point a quaternion sum is computed for use by the ADF as an average, then the second gyro read and store is performed. The last calculation of the first minor cycle is the generation of the computed platform attitude from the quaternion in the form of a direction cosine matrix (DCM). At the completion of the first minor cycle, the DCU waits for a synch pulse before beginning the second minor cycle.

ORIGINAL PAGE IS
OF POOR QUALITY

The second minor cycle begins with the computation of the attitude error signal $\hat{\theta}_{ERR}$ from the computed platform attitude and a desired attitude θ_{DES} supplied by the CDMS. Next the attitude errors are limited to prevent gyro saturation and then summed to form the integral error signal $\int\hat{\theta}_{ERR}$. The position and integral error signals are processed in the next minor cycle. Averaging of the first and second gyro reads is then performed and input into the second minor cycle rate prefilter. The second 50-Hz acceleration measurement is acquired, and the acceleration prefiltering is performed. This completes the 25-Hz control processing. The second minor cycle continues with execution of telemetry processing and thermal control processing. Also, during fine pointing operations, a 1-Hz branch is executed; this computes the state transition matrix to be used by the CDMS for ADF processing. Section IV will present the method for calculating the state transition matrix from the current and 1-s-old IPS quaternion attitudes. The state transition matrix is used in both the DCU and in the CDMS to propagate the state estimate to the desired time interval. Also, during the 1-Hz computation cycle, the quaternion is updated by the ADF estimate. After completion of the second minor cycle, the DCU again waits for another 20 ms synch pulse before beginning the next 25-Hz control cycle.

Processing of the 1-Hz ADF software by the SSC during fine pointing operations produces the second of the multirate control loops. As specified in the fast-loop controller description, the ADF provides the DCU with IPS attitude updates and system drift estimates once every second. Inputs required by the ADF are the OSP measurements, the state transition matrix components calculated in the DCU, and the pointing objective star direction vectors. The remainder of this report will present the development and simulation of the four estimation filters.

III. ESTIMATION FILTER THEORY

Estimation is the determination of a physical system's state when the system and its output measurements are corrupted by stochastic noise processes. Optimal estimation produces a minimum expected error estimate based upon several factors: a cost or performance index, uncertainty of system model, statistical characteristics of system and measurement noise, and initial conditions. Smoothing, filtering, and prediction are the three types of estimation problems. To perform smoothing, measurements before and after the time point of interest must be processed to estimate the system's current states. Calculation of the state estimate at the end of a period of noisy measurements is referred to as filtering. Prediction is the processing of measurements to estimate the state of a system at some future time [4].

The theory of estimation can be applied to a wide range of problems, such as target tracking, digital image enhancement, vehicle attitude determination, materials processing, and estimation of nuclear reactor states. The estimation problem discussed in this report will be the determination of a pointing platform's inertial attitude using filtering techniques.

A. Historical Survey

This section is not intended to give a complete review on estimation theory. There are several papers which are devoted to the development of estimation theory along with extensive

bibliographies: Kailath [12], Sorenson [13], and Lefferts [14]. Textbooks by Jazwinski [15], Gelb [4], and Brown [16] treat estimation theory with varying degrees of mathematical rigor.

The first published work on estimation theory is credited to Legendre in 1805 [17], but it has since been recognized that Gauss, at the age of 18, employed recursive estimation techniques in 1795 to aid astronomers in locating the asteroid Ceres. Gauss published his results in his book "Theoria Motus" [18] in 1809. The techniques developed by Gauss are known today as least-squares estimation. Fisher [19] developed the maximum likelihood approach more than a hundred years later. In 1942, Kolmogorov [20] and Wiener [21] independently developed the linear minimum mean-square estimation theory based upon classical transform function techniques. Wiener's least-squares method was applied to the linear continuous-time stationary process. In 1960 Kalman [22] published the paper "A New Approach to Linear Filtering and Prediction Problems" which, because of its discrete-time recursive nature, was readily implemented in the digital computer. The recursive equations which became known as the Kalman filter are developed using state space techniques. Since the development of the Kalman filter in 1960, the use of estimation theory has been applied to various problems.

B. Kalman Filter Theory

The majority of material presented in this section is based upon the theory presented by Gelb [4], with portions from Brown [16]. Real systems have many sources of noise or errors which contribute to the uncertainty of their states. Estimation is the blending of noisy measurements with the imperfectly modeled state to obtain a better prediction of the state. Therefore, it can be seen that an estimator requires a model of the system. In the linear case, the discrete-time model is given by

$$x_k = \Phi_{k,k-1} x_{k-1} + w_{k-1} \quad , \tag{1}$$

where x_k is the system's $n \times 1$ state vector and w_k is the system error, or noise, vector. Also, $\Phi_{k,k-1}$ represents the discretized state transition matrix or the solution of the unforced dynamical system, and t_k defines the discrete time point at which the state vector is available. Since the estimation algorithm is implemented on a digital computer, a discrete-time system representation is utilized. In order to base an estimate upon measurements, a relationship between the measurements and the states must be established. For the case of the linear, discrete time measurement, the state to measurement relationship is

$$z_k = H_k x_k + v_k \quad , \tag{2}$$

where z_k is a discrete $m \times 1$ measurement vector, H_k is the $m \times n$ observation matrix, and v_k is the measurement error vector.

The choice of an estimation scheme depends on the type of system which must be evaluated and the amount of information that the designer has about the system. For a system which has a stationary uncertainty in the knowledge of the system, a simple observer could be implemented to estimate the desired states. A process is stationary if its probability distribution and associated statistics do not vary with time. For the situation where a system's uncertainty is stationary, the feedback gain of the observed state is time invariant. The minimization of the function

$$J_k = (z_k - H_k\hat{x}_k)^T(z_k - H_k\hat{x}_k) \ , \tag{3}$$

where $\hat{x}_k$ is the $n\times 1$ estimate of the state at time t_k, is referred to as least-squares estimation. Generally, estimation by least-squares requires no statistical knowkedge of the system's states x_k or the measurements z_k, consequently both processes receive equal weighting. On the other hand, the maximum likelihood method assumes some knowledge of the properties of the measurement vector represented by the random process v_k. Minimization of the cost function

$$J_k = (z_k - H_k\hat{x}_k)^T R^{-1}(z_k - H_k\hat{x}_k) \ , \tag{4}$$

where R is a positive semidefinite matrix based upon the statistical properties of v_k, yields a system estimate $\hat{x}_k$ which is indirectly related to the uncertainty of z_k. A Bayesian estimation approach weights a cost function using the conditional probability $p_k(x_k|z_k)$ of the system and measurement vectors. For a minimum variance Bayesian estimate, the cost function is represented by the second moment of the estimation error

$$J_k = \int_{-\infty}^{+\infty} (\hat{x}_k - x_k)^T(\hat{x}_k - x_k)p_k(x_k|z_k)dx_k \ . \tag{5}$$

Equation (5) is also referred to as the expectation of the square of the estimation error

$$J_k = E[(\hat{x}_k - x_k)^T(\hat{x}_k - x_k)] \ . \tag{6}$$

Development of the discrete Kalman filter in 1960 resulted in a recursive minimum variance Bayesian estimator, which uses a blending of noisy measurements with previous state estimates to produce the best state estimate based on all current information. For the LKF, the discrete-time system represented by equation (1) is contaminated by the noise process w_k which has a zero mean, Gaussian distribution. Since w_k is a stochastic process, its statistical properties can be written as

$$E[w_k] = 0 \qquad \text{and} \qquad E[w_k w_i^T] = \begin{cases} Q_k & i=k \\ 0 & i\neq k \end{cases} \ , \tag{7}$$

where i and k denote the times at which the state vector is available. The observation model for the KF given by equation (2) is corrupted by a sequence of Gaussian distributed random variables v_k. Also, the measurement noise vector is uncorrelated with, and independent of, the system noise. Statistical properties of v_k are presented in equation (8)

$$E[v_k] = 0 \quad \text{and} \quad E[v_k v_i^T] = \begin{cases} R_k & i=k \\ 0 & i \neq k \end{cases} , \tag{8}$$

where i and k denote the times at which the measurements occur. Gaussian distributions are assumed for both noise processes based on the central limit theorem. The central limit theorem can essentially be expressed as follows: the sum or combination of small independent stochastic processes exhibit an approximate Gaussian distribution regardless of the individual process distributions [16]. Given an initial estimate of the state denoted as $\hat{x}_0^-$, the linear state estimate update can be written as

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k\hat{x}_k^-) . \tag{9}$$

A parameter with a superscript plus sign ($^+$) denotes that the parameter is updated or estimated during the current measurement interval, whereas the superscript minus sign ($^-$) represents a variable which was calcuulated during the previous measurement interval and propagated to the current measurement or filter cycle. The performance criterion for choosing K_k is the minimization of the error covariance matrix. Substitution of equations (2) and (9) into equation (6) yields the error covariance matrix P_k as follows [16]

$$\begin{aligned} P_k^+ = E\{&[(x_k - \hat{x}_k^-) - K_k(H_k x_k + v_k - H_k\hat{x}_k^-)][(x_k - \hat{x}_k^-) \\ &- K_k(H_k x_k + v_k - H_k\hat{x}_k^-)]^T\} . \end{aligned} \tag{10}$$

Equation (10) can be rearranged into the following form

$$P_k^+ = E\{[(I - K_k H_k)(x_k - \hat{x}_k^-) - K_k v_k][(I - K_k H_k)(x_k - \hat{x}_k^-) - K_k v_k]^T\} . \tag{11}$$

Next, expanding equation (11) produces

$$\begin{aligned} P_k^+ = E\{&(I - K_k H_k)(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T(I - K_k H_k)^T - (I - K_K H_k)(x_k - \hat{x}_k^-)v_k^T K_k^T \\ &- K_k v_k(x_k - \hat{x}_k^-)^T(I - K_k H_k)^T + K_k v_k v_k^T K_k^T\} . \end{aligned} \tag{12}$$

Noting that the predicted error covariance matrix can be written as

$$P_k^- = E\{(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T\} \quad . \tag{13}$$

Also, recall that the measurement noise vector is uncorrelated with the system error and that the expectation of the sensor noise is given by equation (8), then by performing the expectation of equation (12), the system error covariance can be expressed as

$$P_k^+ = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \quad . \tag{14}$$

Equation (14) is the general expression for updating the error covariance matrix which is valid for any gain K_k for which equation (9) holds. The FHST tracking rate limitation constrains the actual flight ADF gains to a suboptimal estimation which requires the use of equation (14) in the gain generation program. For the purpose of this report the star tracker rate limitations will be neglected, which will allow the use of optimal filter gains.

There are several methods which may be used to determine an optimal choice of K_k, Sorenson [23] and Brown [16] use a completion of the square method to minimize the trace of the error covariance matrix. The more frequently used method of minimizing the diagonal of P_k using matrix differentiation is presented here [4]

$$\frac{\partial \text{trace}(P_k)}{\partial K_k} = 0 \quad . \tag{15}$$

First, the terms of P_k can be expanded and regrouped to form

$$P_k^+ = P_k^- - K_k H_k P_k^- - P_k^- H_k K_k^T + K_k (H_k P_k^- H_k) K_k^T + K_k R_k K_k^T \quad . \tag{16}$$

Utilizing the relationship that the trace$[AB]$ = trace$[BA]$ and that P_k is a symmetric matrix, then the trace$[K_k H_k P_k]$ = trace$[P_k H_k K_k]$ and, therefore, the traceP_k can be written as

$$\text{trace}P_k^+ = \text{trace}P_k^- - 2\text{trace}[K_k H_k P_k^-] + \text{trace}[K_k (H_k P_k^- H_k^T + R_k) K_k^T] \quad . \tag{17}$$

Taking the partial of equation (17) with respect to K_k and using the following relationships

$$\frac{\partial \text{trace}[AB]}{\partial A} = B^T \qquad \frac{\partial \text{trace}[ABA^T]}{\partial A} = 2AB \quad , \tag{18}$$

then the expression for an optimal choice of K_k can be stated as follows

$$-2P_k^- H_k^T + 2K_k(H_k P_k^- H_k^T + R_k) = 0 \ . \tag{19}$$

Equation (19) can be solved for K_k to obtain the generally recognized form of the Kalman gain matrix

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1} \ . \tag{20}$$

Substitution of equation (20) into equation (16) with minimal simplification results in

$$P_k^+ = P_k^- - P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^- \ . \tag{21}$$

Realizing that the second term can be replaced with $K_k H_k P_k^-$, then equation (21) can be rewritten in the frequently used Kalman covariance update equation

$$P_k^+ = [I - K_k H_k] P_k^- \ . \tag{22}$$

Equations (9), (20), and (22) make up the Kalman filter update equations which are processed whenever a new measurement is available. The state estimate and the error covariance of the previous measurement are required by the update equations. In a dynamic system, the state estimate and the error covariance at the previous measurement are not good measures of the system state. Therefore, the previous estimate and covariances must be propagated to the current time. Typically, the state estimate can be propagated via the state transition matrix generated from the system model

$$\hat{x}_k^- = \Phi_{k,k-1} \hat{x}_{k-1}^+ \ , \tag{23}$$

where $\Phi_{k,k-1}$ is the state transition matrix (STM), and the system noise process is neglected to produce an unbiased predicted state. An unbiased predicted state is generated due to the assumed zero mean Gaussian distribution of the system's noise vector w_k, since the best estimate of the noise process at time t_k is its expected value $E[w_k] = 0$ [24]. Propagation of the error covariance can be accomplished by first expressing the system error as the difference between system state and the state estimate at time t_k

$$e_k = x_k - \hat{x}_k^- \ . \tag{24}$$

Substituting equation (23) and equation (1) into equation (24) yields the system error in terms of the system's state and state estimate of the previous filter interval

$$e_k = \Phi_{k,k-1}x_{k-1} + w_{k-1} - \Phi_{k,k-1}\hat{x}_{k-1}^{+} \,. \tag{25}$$

The predicted error covariance matrix P_k^- can be found by taking the second moment of equation (25) as follows

$$\begin{aligned} P_k^- &= E[e_k e_k^T] \\ &= E\{[\Phi_{k,k-1}(x_{k-1} - \hat{x}_{k-1}^{+}) + w_{k-1}]\ [\Phi_{k,k-1}(x_{k-1} - \hat{x}_{k-1}^{+}) + w_{k-1}]^T\} \,. \end{aligned} \tag{26}$$

Expanding the terms of equation (26) gives

$$\begin{aligned} P_k^- = E[&\Phi_{k,k-1}(x_{k-1} - \hat{x}_{k-1}^{+})(x_{k-1} - \hat{x}_{k-1}^{+})^T\Phi_{k,k-1}{}^T + \Phi_{k,k-1}(x_{k-1} - \hat{x}_{k-1}^{+})w_{k-1}{}^T \\ &+ w_{k-1}(x_{k-1} - \hat{x}_{k-1}^{+})^T\Phi_{k,k-1}{}^T] + w_{k-1}w_{k-1}{}^T \,. \end{aligned} \tag{27}$$

Recall that the system noise and the system error are uncorrelated and that the expectation of the quadratic term is the updated error covariance of the previous filter cycle, then, by performing the expectation of equation (27), the predicted error covariance can be rewritten as

$$P_k^- = \Phi_{k,k-1}P_{k-1}{}^{+}\Phi_{k,k-1}{}^T + Q_{k-1} \,, \tag{28}$$

where Q_{k-1} is given by equation (7). Figure 5 shows the complete recursive Kalman filter process in a flow diagram for a linear system.

C. Nonlinear Filter Theory

As was stated in the introduction, the use of quaternions for attitude representation and the multibody rotational dynamics result in nonlinear system and observation models. The nonlinear continuous system can be described by the differential equation

$$\dot{x}(t) = f(x(t),t) + w(t) \,, \tag{29}$$

and the nonlinear discrete measurement equation can be represented by

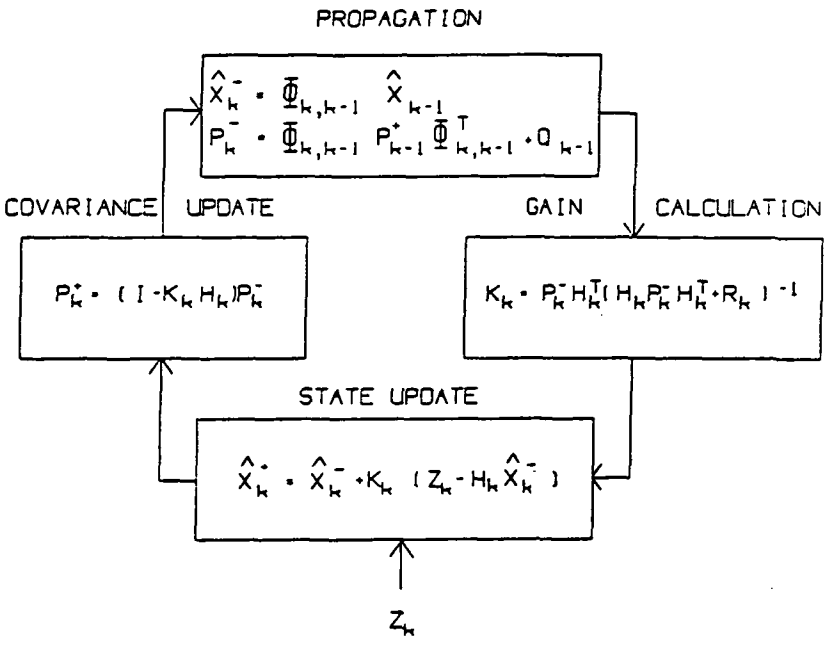


Figure 5. Flow diagram for the linear KF.

$$z_k = h_k(x_k) + v_k \ . \tag{30}$$

For all three nonlinear estimation filters presented in this report, the task of obtaining approximate solutions to the nonlinear representations is accomplished using truncated Taylor series approximations. The extensions of the linear KF theory for each of the nonlinear approximations are presented in the following sections.

D. Linearized Kalman Filter

Dornier deveoped the LKF based upon the state deviation method presented by Jazwinski [15]. The state deviation or perturbation is given by

$$\partial x(t) = x(t) - \bar{x}(t) \ , \tag{31}$$

where $\bar{x}(t)$ is a known reference trajectory about which equation (29) is linearized. Substitution of equation (31) into equation (29) yields

$$\partial \dot{x}(t) = f(x(t),t) - f(\bar{x}(t),t) + w(t) \ . \tag{32}$$

A Taylor series approximation of equation (32), truncated after the second term, gives

$$f(x(t),t) - f(\bar{x}(t),t) = F(\bar{x}(t_0),t)\partial x(t) \quad , \tag{33}$$

where $F(\bar{x}(t_0),t)$ is a Jacobian matrix evaluated along the reference trajectory: i.e..

$$F(\bar{x}(t_0),t) = \frac{\partial f_i(\bar{x}(t),t)}{\partial x_j}\Big|_{x=\bar{x}} \quad . \tag{34}$$

Next the relationship between the state deviation vector ∂x_k and the measurement vector z_k is developed by forming a nominal measurement vector z_k. The nominal measurement vector is obtained by linearizing equation (30) about the reference trajectory $\bar{x}(t)$ using a Taylor series approximation. Truncation of the linearized nominal measurement equation after the second term yields

$$\bar{z}_k = h_k(\bar{x}_k) + H_k(\bar{x}_k)\partial x_k + v_k \quad , \tag{35}$$

where $H_k(\bar{x}_k)$ is a Jacobian of the nonlinear observation matrix evaluated along the reference trajectory given by

$$H_k(\bar{x}_k) = \frac{\partial h_k(x)}{\partial x}\Big|_{x=\bar{x}_k} \quad . \tag{36}$$

Once a linearized representation of the system and observation models have been obtained, they can be substituted into the appropriate equations in the linear KF derivation to produce the LKF update equations

$$\partial\hat{x}_k^+ = \partial\hat{x}_k^- + K_k[z_k - h(\bar{x}_k) - H_k(\bar{x}_k)\partial\hat{x}_k^-] \quad , \tag{37}$$

$$P_k^+ = [I - K_k H_k(\bar{x}_k)]P_k^- \quad , \tag{38}$$

$$K_k = P_k^- H_k^T(\bar{x}_k)[H_k(\bar{x}_k)P_k^- H_k^T(\bar{x}_k) + R_k]^{-1} \quad , \tag{39}$$

and the propagation equations

$$\partial\hat{x}_k^- = \Phi_{k,k-1}\partial\hat{x}_k^+ \quad , \tag{40}$$

$$P_k^- = \Phi_{k,k-1} P_k^+ \Phi_{k,k-1} + Q_{k-1} \quad . \tag{41}$$

Once the state deviation estimate has been obtained, the state estimate can be updated by

$$\hat{x}_k^+ = \hat{x}_k^- + \partial\hat{x}_k^+ \quad . \tag{42}$$

The use of a known reference trajectory about which the estimation filter models are linearized allows the computation of the gains a priori. Therefore, only equations (37), (40), and (42) are necessary to implement the LKF in the IPS flight software. Of course the state transition matrix must be computed to satisfy equation (39).

E. Extended Kalman Filter

To obtain the EKF, the nonlinear system equation (29) and the nonlinear observation equation (30) are linearized about the predicted state estimate $\hat{x}^-$. According to Gelb [4], the truncated Taylor series approximations using the predicted estimate are found to be

$$f(x(t),t) = f(\hat{x}(t),t) + F(\hat{x}(t),t)(x_k - \hat{x}(t)) \quad , \tag{43}$$

and

$$h_k(x_k) = h_k(\hat{x}_k^-) + H_k(\hat{x}_k^-)(x_k - \hat{x}_k^-) \quad , \tag{44}$$

where

$$F(\hat{x}(t),t) = \frac{\partial f_i(x(t),t)}{\partial x_j}\Big|_{x=\hat{x}^-} \qquad \text{and} \qquad H_k(\hat{x}_k^-) = \frac{\partial h_k(x)}{\partial x}\Big|_{x=\hat{x}_k^-} \quad . \tag{45}$$

For the EKF the full state vector is employed as opposed to the state deviation vector used in the LKF. As with the LKF, substitution of equations (43) and (44) into the system and observation models in the derivation of the linear KF produces the EKF update equations as follows

$$\hat{x}_k^+ = \hat{x}_k^- + K_k[z_k - h(\hat{x}_k^-)] \quad , \tag{46}$$

$$K_k = P_k^- H_k^T(\hat{x}_k^-)[H_k(\hat{x}_k^-)P_k^- H_k^T(\hat{x}_k^-) + R_k]^{-1} \quad , \tag{47}$$

$$P_k^+ = [I - K_k H_k(\hat{x}_k^-)]P_k^- \quad . \tag{48}$$

Also the EKF propagation equations are determined to be

$$\hat{x}_k^- = \Phi_{k,k-1}\hat{x}_{k-1}^+ \quad , \tag{49}$$

$$P_k^- = \Phi_{k,k-1}P_k^+\Phi_{k,k-1} + Q_{k-1} \quad . \tag{50}$$

Since the EKF is linearized about the predicted state estimate, equations (46) through (50) must be computed during one EKF estimation interval.

F. Second-Order Kalman Filter

The material presented in this section is mainly derived from the paper by Vathsal, "Spacecraft Attitude Determination Using a Second-Order Nonlinear Filter" [5], although some notation is consistent with Gelb [4]. As with the EKF, the SOKF is obtained by linearizing the nonlinear system and observation models about the predicted state estimate $\hat{x}^-$ using a Taylor series expansion. However, for the SOKF, second-order terms are considered significant and are included in the approximation

$$f(x(t),t) = f(\hat{x}(t),t) - F(\hat{x}(t),t)\tilde{x}(t) + \frac{1}{2}\partial^2(f,\tilde{x}(t)\tilde{x}^T(t)) \quad , \tag{51}$$

and

$$h_k(x_k) = h_k(\hat{x}_k) - H_k(\hat{x}_k^-)\tilde{x}_k^- + \frac{1}{2}\partial^2(h_k,\tilde{x}_k^-\tilde{x}_k^{-T}) \quad . \tag{52}$$

The third terms in equations (51) and (52) can be written as vectors of the traces of Hermitian matrices, of the respective models, and the system covariance P_k^-. Equation (53) gives the ith Hermitian matrix which corresponds to the second partial of the ith element of the observation vector z_k with respect to the filter states x_k

$$\tilde{H}_i = \frac{\partial^2 h_k(\hat{x}_k^-)_i}{\partial x_n \, \partial x_m} \quad . \tag{53}$$

Substitution of the linearized equations into the linear KF development yields the SOKF update equations

$$\hat{x}_k^+ = \hat{x}_k^- + K_k[z_k - h_k(\hat{x}_k^-) - A_k] \ , \tag{54}$$

$$K_k = P_k^- H_k^T(\hat{x}_k^-)[H_k(\hat{x}_k^-)P_k^- H_k^T(\hat{x}_k^-) + R_k + B_k]^{-1} \ , \tag{55}$$

and

$$P_k^+ = [I - K_k H_k(\hat{x}_k^-)]P_k^- \ , \tag{56}$$

where the ith element of the vector A_k is

$$[A_k]_i = \frac{1}{2} \text{trace}\{\tilde{H}_i(\hat{x}_k^-)P_k^-\} \ , \tag{57}$$

and B_k is the matrix whose ijth element is given by

$$[B_k]_{ij} = \frac{1}{2} \text{trace}[\tilde{H}_i(\hat{x}_k^-)P_k^-\tilde{H}_j(\hat{x}_k^-)P_k^-] \ . \tag{58}$$

The propagation equations of the SOKF are the same as for the EKF assuming the STM was generated including the second-order system terms.

IV. IPS ATTITUDE DETERMINATION

The desire to view different astronomical bodies and the occultation of objectives due to the shuttle's orbit required the capability to perform IPS and/or orbiter inertial attitude maneuvers. In order to minimize the time between astronomical observations, the maneuvers are performed at rates which exceed the IPS optical sensors' tracking capability. Therefore, the inertial attitude is maintained by the gyros and the attitude calculations performed in the DCU. Both the unknown gyro drift and DCU numerical errors result in an accumulated attitude error unobserved by the IPS control system. To improve the IPS pointing performance, an estimation filter was implemented using the optical sensor to determine the system drifts and the attitude errors accumulated during gyro-only control. Use of the optical sensor requires that the estimation filter be executed during fine pointing operations, when the IPS platform inertial rates are expected to be within the sensors tracking capability.

The first three terms of the estimation filter are the inertial attitude of the IPS. The attitude of a vehicle can be described using various methods such as DCM [25], Euler angles [26], or by quaternions [27]. A quaternion is a vector of dimension 4×1 which can define a coordinate transformation between two frames by one rotation about the axis which is unique to both coordinate systems. Quaternions were selected as the method to determine IPS's attitude because they require fewer variables than DCM's or Euler angles and are not subject to the trigonometric singularities commonly known as gimbal lock. Normalization of the quaternion vector allows reduction of the system's states through the relationship

$$q_4 = \sqrt{1 - q_1^2 - q_2^2 - q_3^3} \ . \tag{59}$$

Since the system drifts D_i contribute significantly to the accumulated attitude errors, inclusion of three states to estimate the drifts in each IPS axis will improve the system performance. The system model shown in equation (60) has been widely used to determine a vehicle's attitude

$$x^T = [\ q_1 \quad q_2 \quad q_3 \quad D_1 \quad D_2 \quad D_3\ ] \ . \tag{60}$$

When Dornier System developed the ADF, which is presently implemented in the IPS, thermal deflections of the star trackers were considered to significantly impact the attitude measurements. As a result, a 10-state estimation filter was designed in which the last four elements of the filter states were misalignments between star trackers. Further analysis of the star trackers' thermal deflection characteristics have since found the misalignments due to thermal gradients to be negligible. Therefore, the six-element state representation shown in equation (60) will be utilized throughout this report. Once the IPS states to be estimated are defined, then the system model shown in equation (29) must be developed to determine the propagation of the states and error covariance matrix of the estimation filter. Most systems require the development of an analytical model to specify the state trajectory. But, vehicle attitude estimation utilizes an applied modeling technique which substitutes rate measurements in place of the dynamic model, eliminating the need to obtain a solution of the dynamic equations [28]. Generally, a gyro package integrates the dynamic equations mechanically, which leaves only the definition and integration of the dynamically coupled kinematic equations to define the motion of the IPS. A well-known kinematic attitude representation attributed to Euler can be written in the differential form using quaternions as

$$\dot{q}(t) = \frac{1}{2}\Omega(t)q(t) \ , \tag{61}$$

where $\Omega(t)$ is the matrix of vehicle body rates ω_i

$$\Omega(t) = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \tag{62}$$

which are outputs of a gyro package. If you assume $\Omega(t)$ is constant over the sample period, then the unforced solution $q(t)$ to the continuous state attitude equation (61) can be written as

$$q(t) = e^{1/2\Omega(t-t_0)} \; q(t_0) \; . \tag{63}$$

To find the discrete time solution to equation (61), let τ be the sample interval between measurements and substitute into equation (63) to obtain

$$q(t_k+1) = e^{1/2\Omega_k\tau} \; q(t_k) \; , \tag{64}$$

which can be rewritten as

$$q_k = \Phi_{k,k-1} q_{k-1} \; , \tag{65}$$

where $\Phi_{k,k-1}$ is the discrete time STM, which can be considered a quaternion matrix. If the quaternion at time t_{k-1} represents a rotation of the IPS platform coordinate system with respect to the fixed inertial frame and the quaternion q_k represents a different platform to inertial transformation at time t_k, then the state transition matrix is a transformation between the two platform coordinate systems. A quaternion which is the culmination of successive rotations can be represented by the multiplication of quaternions which define the individual rotations

$$q_T = q_1 q_2 \; . \tag{66}$$

The quaternion multiplication expressed by equation (66) was shown by Euler [29] to be

$$\begin{aligned} q_{T1} &= q_{11}q_{21} - q_{12}q_{22} - q_{13}q_{23} - q_{14}q_{24} \\ q_{T2} &= q_{11}q_{22} + q_{12}q_{21} + q_{13}q_{24} - q_{14}q_{23} \\ q_{T3} &= q_{11}q_{23} - q_{12}q_{24} + q_{13}q_{21} + q_{14}q_{22} \\ q_{T4} &= q_{11}q_{24} + q_{12}q_{23} - q_{13}q_{22} + q_{14}q_{21} \; , \end{aligned} \tag{67}$$

where the first subscript shown in equation (67) defines the quaternion rotation and the second subscript is the quaternion component. It can be shown that an equivalent expression for equation (66) can be obtained from the multiplication of the "quaternion matrix" [29], which is composed of the elements of the quaternion q_1, with the quaternion vector q_2.

$$\begin{pmatrix} q_{T1} \\ q_{T2} \\ q_{T3} \\ q_{T4} \end{pmatrix} = \begin{pmatrix} q_{11} & -q_{12} & -q_{13} & -q_{14} \\ q_{12} & q_{11} & -q_{14} & q_{13} \\ q_{13} & q_{14} & q_{11} & -q_{12} \\ q_{14} & -q_{13} & q_{12} & q_{11} \end{pmatrix} \begin{pmatrix} q_{21} \\ q_{22} \\ q_{23} \\ q_{24} \end{pmatrix} \tag{68}$$

From Ickes [29], the quaternion q_T can also be found from

$$\begin{pmatrix} q_{T1} \\ q_{T2} \\ q_{T3} \\ q_{T4} \end{pmatrix} = \begin{pmatrix} q_{21} & -q_{22} & -q_{23} & -q_{24} \\ q_{22} & q_{21} & q_{24} & -q_{23} \\ q_{23} & -q_{24} & q_{21} & q_{22} \\ q_{24} & q_{23} & -q_{22} & q_{21} \end{pmatrix} \begin{pmatrix} q_{11} \\ q_{12} \\ q_{13} \\ q_{14} \end{pmatrix} \tag{69}$$

The matrix given in equation (69) is referred to as the quaternion transmuted matrix. Assuming that the gyro package measurements are output at a fixed sample rate and that the vehicle's rates are constant over the sample period, then a closed-form solution is known for the Euler's differential kinematic equation. However, Dornier System implemented a numerical approximation to the solution of the state transition matrix [30]. First the updated attitude quaternion q_{k-1} is stored at the beginning of a 1-Hz ADF cycle. The 25-Hz control loop removes the updated attitude error during the 1-s ADF interval. When the 1-Hz interval is completed, the current quaternion q_k calculated in the fast loop is the gyro model of the predicted ADF estimate. Since the IPS attitude state transition between two estimation filter cycles is essentially a rotation between two attitudes, the state transition matrix can be expressed as quaternion matrix. Therefore, equation (65) can be rewritten in the following form

$$q_k = \begin{bmatrix} \Phi_0 & \Phi_3 & -\Phi_2 & \Phi_1 \\ -\Phi_3 & \Phi_0 & \Phi_1 & \Phi_2 \\ \Phi_2 & -\Phi_1 & \Phi_0 & \Phi_3 \\ -\Phi_1 & -\Phi_2 & -\Phi_3 & \Phi_0 \end{bmatrix} q_{k-1} \, . \tag{70}$$

The equivalent relationship using the transmuted quaternion matrix is

$$q_k = \begin{bmatrix} q_1 & q_4 & -q_3 & q_2 \\ q_2 & q_3 & q_4 & -q_1 \\ q_3 & -q_2 & q_1 & q_4 \\ q_4 & -q_1 & -q_2 & -q_3 \end{bmatrix} \begin{bmatrix} \Phi_0 \\ \Phi_1 \\ \Phi_2 \\ \Phi_3 \end{bmatrix} , \tag{71}$$

where the elements of the matrix are elements of the quaternion q_{k-1} saved from the end of the previous filter cycle. That is

$$q_k = q_{k-1}\Phi_{k,k-1} . \tag{72}$$

It can be shown that any quaternion matrix is an orthogonal matrix, therefore its transpose is equal to its inverse, and as a result equation (72) can be shown as

$$\Phi_{k,k-1} = q_{k-1}{}^T q_k . \tag{73}$$

Equation (73) is the calculation of the quaternion components which represent a Euler axis rotation between the IPS platform attitude at two different time intervals. Once the components of $\Phi_{k,k-1}$ are determined, they can be substituted into equation (70) to define the state transition matrix. According to Dornier, the method presented for determining the elements of $\Phi_{k,k-1}$ was simulation verified and has been implemented in the current IPS software. The STM computation method given above was also used in the estimation algorithms presented in this report in an effort to maintain consistent simulations. The STM development given has only been for the attitude states. System drifts are assumed to be constant, which allows the propagation of the drifts with the identity transformation. This completes the definition of the system model and the state transition matrix.

Next, the development of the IPS observation equations will be presented. First, the implementation of the linear, discrete Kalman equations are assumed to be valid due to the small rotations and rates experienced during IPS fine pointing operations. A linear relationship between the filter states and the measurement vector was developed as a part of this study and was based on the following observations. Recall that the first three components of the state vector are the complex components of the quaternion which is an eigenaxis rotation from the actual IPS platform attitude to the desired inertial attitude. For a small rotation (θ less than 10°), the complex quaternion components can be simplified to [31]

$$[q_1 \; q_2 \; q_3]^T = \left[\frac{\theta_x}{2} \quad \frac{\theta_y}{2} \quad \frac{\theta_z}{2}\right]^T , \tag{74}$$

where θ_i are the components of the rotation about the gimbal axes. A small pitch attitude error results in a rotation about the IPS y axis which corresponds to a z tracker measurement and a q_3 state estimate. A small yaw attitude error produces a rotation about the y axis, a z tracker measurement, and a q_2 state estimate. From equation (74) it can be seen that for a small angle the star tracker measurements will be twice the corresponding quaternion element. Both pitch and yaw attitude errors are seen equally by all three trackers. A small roll attitude error will be seen as a z measurement of equal magnitude, but opposite in sign, by each skew star tracker. The roll quaternion q_1 is related to the skew tracker measurements by twice the sine of the trackers's skew angle ($2\sin 12° \approx 0.4$). Finally, since the drifts are included in the accumulated attitude errors, the observation matrix drift terms are zero. The complete observation matrix for the linear KF implementation is presented in equation (75).

$$H(t) = \begin{pmatrix} 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 \\ 0.4 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 \\ -0.4 & 2 & 0 & 0 & 0 & 0 \end{pmatrix} . \tag{75}$$

A listing of the discrete Kalman filter is presented in appendix C.

Next the observation model of the three nonlinear filters is developed. The observation model widely used for inertial star trackers in the determination of a vehicle's attitude transforms a star's reference direction vector in inertial space S_r to a predicted measurement direction vector in the sensor coordinate frame S_s as given in equation (76) [28]

$$S_s = TA(q)S_r . \tag{76}$$

An ephemeris, a catalog of star positions on certain dates, is used to define the components of the reference star direction vector for the desired mission pointing objectives. The matrix $A(q)$ is the inertial attitude transformation from the desired inertial attitude to the actual IPS body attitude, where the well-known nonlinear quaternion representation of $A(q)$ is given by

$$A(q) = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_1q_4 + q_2q_3) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \tag{77}$$

The matrix T in equation (76) is a fixed transformation from the IPS body coordinate frame to the sensor coordinate system given by

$$T = \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{78}$$

For the boresight FHST, the matrix T is the identity matrix, while the skew tracker transformations are $\pm\alpha$ rotations about the boresight z axis. The rotation α is dependent upon the IPS mission configuration; α is 45° for solar missions and 12° for stellar missions. A 12° rotation of the skew trackers is implemented for the simulation presented in this report.

Once the predicted star direction vector S_s has been defined by equation (76), then the relationship between the direction vector and the star tracker outputs must be determined. The FHST measures two geometrical coordinates y_f and z_f in the sensor focal plane. The measurements are proportional to the angles defined by the IPS LOS and the star direction vector as shown in figure 6. Before being output by the FHST, the geometrical coordinates are divided by the optical focal length, which forms the direction tangents of the angles α_y and α_z.

From figure 6 it can be observed that the star direction vector in the sensor coordinate frame can also be represented by the direction tangents of the angles α_y and α_z [32]

$$\tan\alpha_y = \frac{S_{S3}}{S_{S1}} \qquad \text{and} \qquad \tan\alpha_z = \frac{S_{S2}}{S_{S1}} \tag{79}$$

From equations (30) and (79), the nonlinear observation model for the IPS for a single star tracker can be written as

$$z_k^T = h_k^T + v_k^T = [S_{S3}/S_{S1} \quad S_{S2}/S_{S1}]^T + v_k^T \ . \tag{80}$$

To implement the three nonlinear filters, equation (80) must first be linearized, which requires the computation of partial derivatives of the observation model with respect to the reference state vector. The SOKF also requires the calculation of the second partials of the observation model. Since the measurement vector z_k consists of two measurements from three FHST, where the transformation T for the separate star trackers are independent of the estimation filter state vector, the partials of the observation equation can be derived for a general star tracker as follows. Let H_1 be the α_y component of the measurement vector and H_2 be the α_z component; then using equation (80) and the differentiation chain rule, the partials for the general measurement equation can be expressed as

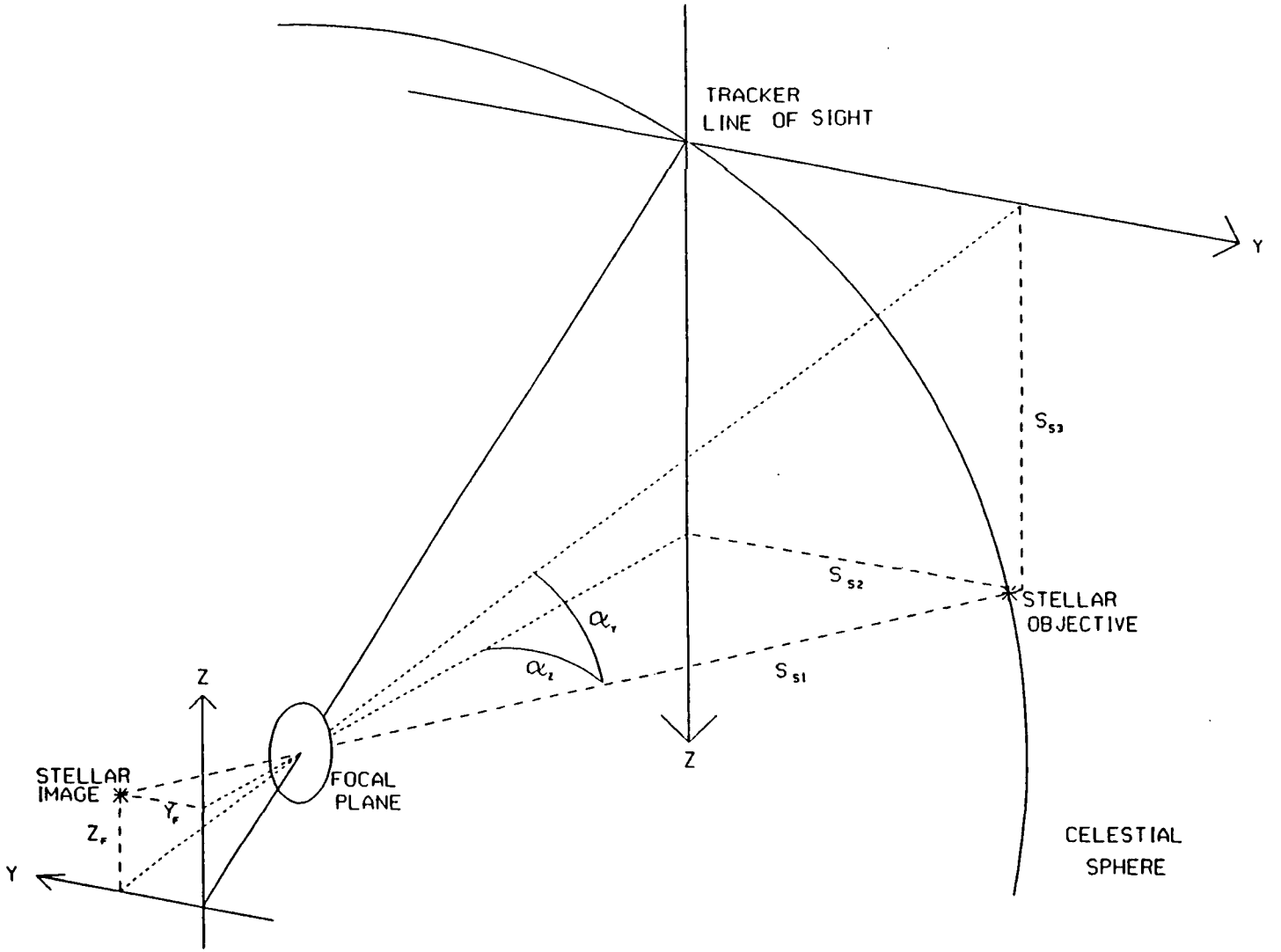


Figure 6. Projection of star tracker measurements.

$$\frac{\partial H_i}{\partial x_m} = \frac{1}{S_1^2}\left\{S_1\frac{\partial S_{i+1}}{\partial x_m} - S_{i+1}\frac{\partial S_1}{\partial x_m}\right\} \qquad \begin{matrix} i = 1,2 \\ m = 1,\ldots,6 \end{matrix} \quad . \tag{81}$$

and the second partials can be written as

$$\frac{\partial^2 H_i}{\partial x_j \partial x_m} = \frac{1}{S_1^3}\left\{S_1^2\frac{\partial^2 S_{i+1}}{\partial x_j \partial x_m} - S_1 S_{i+1}\frac{\partial^2 S_1}{\partial x_j \partial x_m} + 2S_{i+1}\left(\frac{\partial S_1}{\partial x_j}\right)\left(\frac{\partial S_1}{\partial x_m}\right) - S_1\left[\left(\frac{\partial S_1}{\partial x_j}\right)\left(\frac{\partial S_{i+1}}{\partial x_m}\right) + \left(\frac{\partial S_{i+1}}{\partial x_j}\right)\left(\frac{\partial S_1}{\partial x_m}\right)\right]\right\} \tag{82}$$

$$i = 1,2 \quad j = 1,\ldots,6 \quad m = 1,\ldots,6 \quad .$$

where i is the LOS component of the measurement vector, and j and m are the components of the estimation filter state vector. Noting that the observation model is independent of the system drifts, and therefore the partials of h_m with respect to the drifts are zero, allows the computation of the partial derivatives of the observation model only with respect to the quaternions.

For the LKF, the implementation of equation (81) by Dornier is found in reference 3. Since the EKF also requires only the first partials of the observation matrix, Dornier's implementation of equation (81) was utilized in the coding of the EKF simulation. However, since the SOKF requires both the first and second partials of the nonlinear observation model, equations (81) and (82) were implemented as follows. Equation (76) can be rewritten as

$$S = [B]r \ , \tag{83}$$

where B is a 3×9 matrix whose elements are combinations of elements from T and S_r. The components of r are elements of the matrix $A(q)$, i.e., $r_1 = A_{11}$, $r_2 = A_{12},\ldots,$ $r_8 = A_{32}$, $r_9 = A_{33}$. Hence, S can be expressed as the product of a matrix B, whose elements $b_{i,j}$ are independent of quaternions, and a vector whose elements are functions of quaternions. Therefore

$$\frac{\partial S_i}{\partial q_m} = \sum_{n=1}^{9} b_{i,n}\frac{\partial r_n}{\partial q_m} \qquad \begin{matrix} i = 1,2,3 \\ m = 1,2,3 \end{matrix} \tag{84}$$

and

$$\frac{\partial^2 S_i}{\partial q_j \partial q_m} = \sum_{n=1}^{9} b_{i,n}\frac{\partial^2 r_n}{\partial q_j \partial q_m} \qquad \begin{matrix} i = 1,2,3 \\ j = 1,2,3 \\ m = 1,2,3 \end{matrix} \tag{85}$$

Performing the partial derivative of r with respect to the quaternions results in the 9×3 matrix W given by

$$W^T = \begin{pmatrix} 0 & w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & -w_5 & w_4 \\ w_7 & w_8 & -w_9 & w_{10} & 0 & w_6 & w_9 & w_2 & w_7 \\ w_{11} & w_{12} & w_{10} & -w_{12} & w_{11} & w_1 & w_8 & w_3 & 0 \end{pmatrix} \tag{86}$$

where the 27 elements of $W_{i,j}$ are defined by 12 unique partials w_n, not including constants. These are given by:

$$\begin{aligned} w_1 &= (2/q_4)(q_2q_4 - q_1q_3) & w_7 &= -4q_2 \\ w_2 &= (2/q_4)(q_1q_2 + q_3q_4) & w_8 &= (2/q_4)(q_1q_4 - q_2q_3) \\ w_3 &= (2/q_4)(q_1q_3 + q_2q_4) & w_9 &= (2/q_4)(q_4^2 - q_2^2) \\ w_4 &= -4q_1 & w_{10} &= (2/q_4)(q_1q_4 + q_2q_3) \\ w_5 &= (2/q_4)(q_4^2 - q_1^2) & w_{11} &= -4q_3 \\ w_6 &= (2/q_4)(q_3q_4 - q_1q_2) & w_{12} &= (2/q_4)(q_4^2 - q_3^2) \end{aligned} \tag{87}$$

For example the element in row 3 column 2, $W_{3,2} = -w_9$, is the first partial of r_3 with respect to q_2, and $W_{8,1} = -w_5$ is the first partial of r_8 with respect to q_1.

The matrix Δ is obtained from taking the second partial of the vector r with respect to the quaternions.

$$\Delta = \begin{pmatrix} 0 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & -4 \\ -\delta_1 & \delta_2 & -\delta_3 & \delta_2 & -\delta_4 & -\delta_5 & -\delta_3 & -\delta_5 & -\delta_6 \\ \delta_7 & \delta_8 & \delta_9 & \delta_8 & \delta_{10} & \delta_4 & \delta_9 & \delta_4 & \delta_5 \\ \delta_1 & \delta_9 & \delta_3 & \delta_9 & \delta_4 & \delta_5 & \delta_3 & \delta_5 & \delta_6 \\ -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4 \\ -\delta_{11} & -\delta_7 & -\delta_1 & -\delta_7 & -\delta_8 & \delta_2 & -\delta_1 & \delta_2 & -\delta_3 \\ -\delta_7 & -\delta_8 & \delta_2 & -\delta_8 & -\delta_{10} & -\delta_4 & \delta_2 & -\delta_4 & -\delta_5 \\ \delta_{11} & \delta_7 & \delta_1 & \delta_7 & \delta_8 & \delta_9 & \delta_1 & \delta_9 & \delta_3 \\ -4 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{88}$$

where the 9×9 matrix Δ has 11 unique second partials δ_n, not including constants, defined by

$$
\begin{aligned}
\delta_1 &= 2\frac{q_3}{q_4}\left(1+\frac{q_1^2}{q_4^2}\right) & \delta_7 &= 2\frac{q_2}{q_4}\left(1+\frac{q_1^2}{q_4^2}\right)\\
\delta_2 &= 2\left(1-\frac{q_1q_2q_3}{q_4^3}\right) & \delta_8 &= 2\frac{q_1}{q_4}\left(1+\frac{q_2^2}{q_4^2}\right)\\
\delta_3 &= 2\frac{q_1}{q_4}\left(1+\frac{q_3^2}{q_4^2}\right) & \delta_9 &= 2\left(1+\frac{q_1q_2q_3}{q_4^3}\right)\\
\delta_4 &= 2\frac{q_3}{q_4}\left(1+\frac{q_2^2}{q_4^2}\right) & \delta_{10} &= 2\frac{q_2}{q_4}\left(3+\frac{q_2^2}{q_4^2}\right)\\
\delta_5 &= 2\frac{q_2}{q_4}\left(1+\frac{q_3^2}{q_4^2}\right) & \delta_{11} &= 2\frac{q_1}{q_4}\left(3+\frac{q_1^2}{q_4^2}\right)\\
\delta_6 &= 2\frac{q_3}{q_4}\left(3+\frac{q_3^2}{q_4^2}\right)
\end{aligned}
\tag{89}
$$

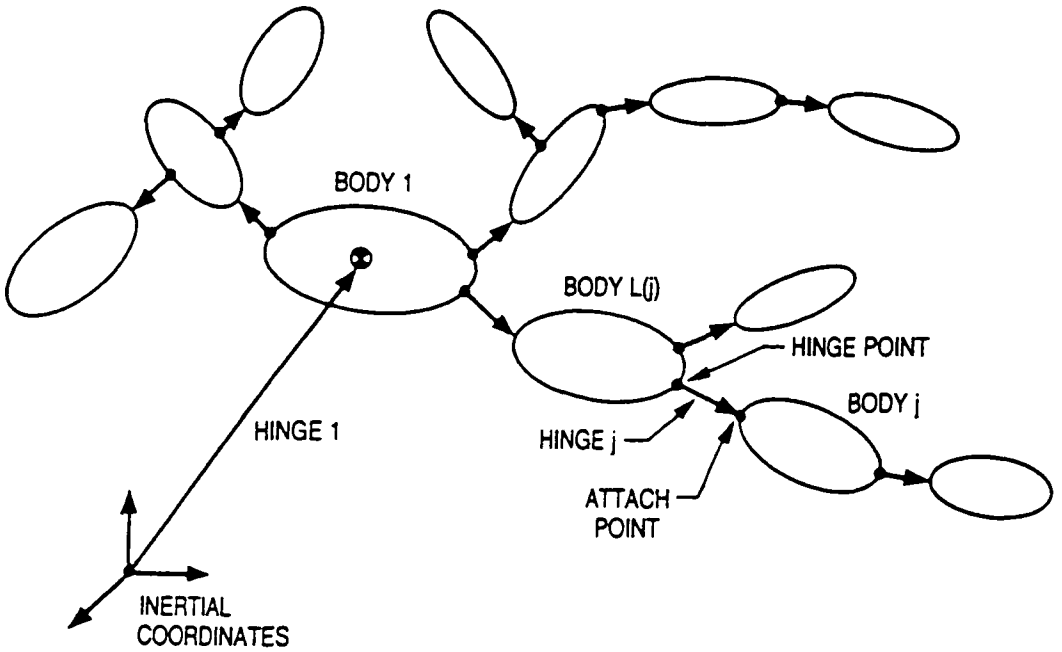
Columns 1 through 9 of Δ represent the second partials with respect to ∂q_1^2, $\partial q_2\partial q_1,\ldots,\partial q_2\partial q_3$, and ∂q_3^2, respectively. For example the element in row 7 and column 6 of the matrix Δ is given by

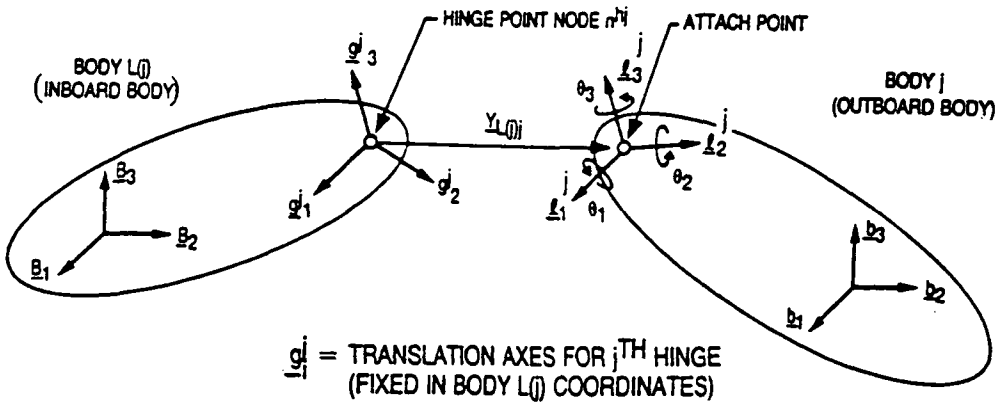
$$\Delta_{7,6} = -\delta_4 = \frac{\partial^2 r_7}{\partial q_3 \partial q_2} \tag{90}$$

The first partials of r given in W and the second partials of r given in Δ are independent of the star tracker. Substitution of W and Δ into equations (84) and (85) yields the partials of the nonlinear equation (76), which are dependent upon the star tracker transformation matrix T and the desired star reference direction vector S_r contained in the matrix B. Therefore, solving equations (84) and (85) for each of the three FHST's and substituting the resulting partials into equations (81) and (82) forms the Jacobian and Hermitian matrices of the IPS observation model. Appendices D through F present the listings for the implementation of the three nonlinear estimation filters LKF, EKF, and SOKF, respectively.

V. SIMULATION DESCRIPTION

Earlier IPS fine pointing simulations were constrained to small deviations about the desired pointing objective. The simulations utilized a fixed IPS structural configuration which did not include nonlinear effects due to varying inertias. Additionally, the nonlinear accelerations were neglected in the kinematic representation. Verification of the IPS control system performance and stability were accomplished by simulating 11 separate IPS gimbal configurations. In order to verify the nonlinear effects on the estimation filters and to improve the verification process, a new IPS simulation was developed as part of the effort of this research. To account for the nonlinear rotational accelerations and varying inertias, a multibody simulation program called TREETOPS was used to model the system equations of motion [33,34]. TREETOPS was developed by Dynacs, supported in part by NASA, to model complex structures such as robot arms, large space structures, and even biosystems in a tree structure as shown in figure 7.





$\underline{g}^j_i$ = TRANSLATION AXES FOR j^{TH} HINGE (FIXED IN BODY L(j) COORDINATES)

$\underline{\ell}^j_i$ = ROTATION AXES FOR j^{TH} HINGE

$\underline{Y}_{L(j)j}$ = TRANSLATION OF j^{TH} HINGE

$\underline{B}_i$ = BASIS VECTORS OF BODY L (j)

$\underline{b}_i$ = BASIS VECTORS OF BODY j

Figure 7. Tree structure for a multibody system.

TREETOPS is written in FORTRAN and will run on several computer systems. The simulations presented herein were performed on a VAX 780 series computer. An interactive setup program called TREESET is provided with TREETOPS which assists the user in:

1. The definition of the structure, attach points, and boundary conditions of individual bodies

2. The definition of sensor and actuator types as well as their locations

3. The definition of two program resident controllers.

TREETOPS can also input NASTRAN (a finite element modeling program) data to define the system's individual body's masses, inertias, and flexibilities. Along with the discrete and continuous program defined controllers, a user supplied controller written in FORTRAN can be easily incorporated into the TREETOPS simulation.

A. IPS Simulation Model

A five-body TREETOPS model was developed to simulate the various IPS operations, shown in figure 8. The first body defined for the IPS simulation is an on-orbit model of the shuttle *Columbia* with the payload bay doors open and the shuttle payload, not including the IPS. Making up the second body is the Spacelab pallet and the IPS gimbal support structure. Body two is rigidly attached to body one. The elevation torque motor housing and cross-elevation torque motor shaft form body three. Body four includes the cross-elevation and roll torque motor housings and the yoke. Body five consists of the roll torque motor shaft, IPS instrumentation, and observational equipment. Each gimbal is constrained to one rotational DOF. The mass properties of each body were obtained from test-verified NASTRAN simulations. Models of the sensors and actuators provided with TREETOPS were compatible with IPS hardware. A user-defined controller was developed to model the 25-Hz digital control loop, the necessary command generators, and the ADF. Figure 9 gives the complete TREETOPS block diagram for the IPS simulation. The 25-Hz control loop and the command generators are in the FORTRAN subroutine USDC which is presented in appendix B. The remaining discussion in this section will present the equations implemented for each estimation filter and the FORTRAN subroutines which contain the equations.

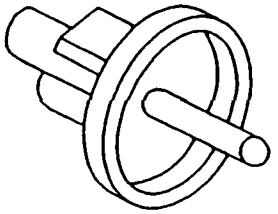
B. Linear Kalman Filter

For the KF, all the equations are contained in the subroutine LADF which is found in appendix C. Only the KF equations implemented in the simulations are presented here, the derivation of the equations can be found in section III-B and section IV.

1. Subroutine LADF

A. State transition matrix equation (73)

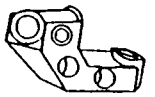
$$\Phi_{k,k-1} = q_{k-1}{}^T q_k \tag{91}$$



INERTIA MATRIX (kg/m^2)

1496	49.7	-116.6
	4484	146.4
		4127

BODY 5 MASS = 3284.1 kg (1 DOF)



INERTIA MATRIX (kg/m^2)

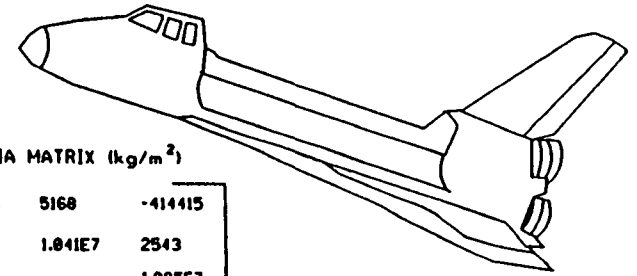
8.1324	-1.3554	-28.463
	16.265	0
		9.4878

BODY 4 MASS = 105.2 kg (1 DOF)

INERTIA MATRIX (kg/m^2)

5.4216	0	-0.1324
	5.4216	0
		1.3554

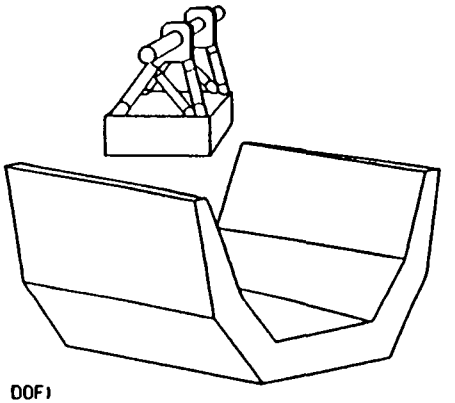
BODY 3 MASS = 84 kg (1 DOF)



INERTIA MATRIX (kg/m^2)

1.362E6	5168	-414415
	1.041E7	2543
		1.085E7

BODY 1 MASS = 103025.2 kg (6 DOF)



INERTIA MATRIX (kg/m^2)

7544	665.7	-661.6
	20290	-196.6
		24311

BODY 2 MASS = 4408 kg (0 DOF)

Figure 8. IPS five-body model.

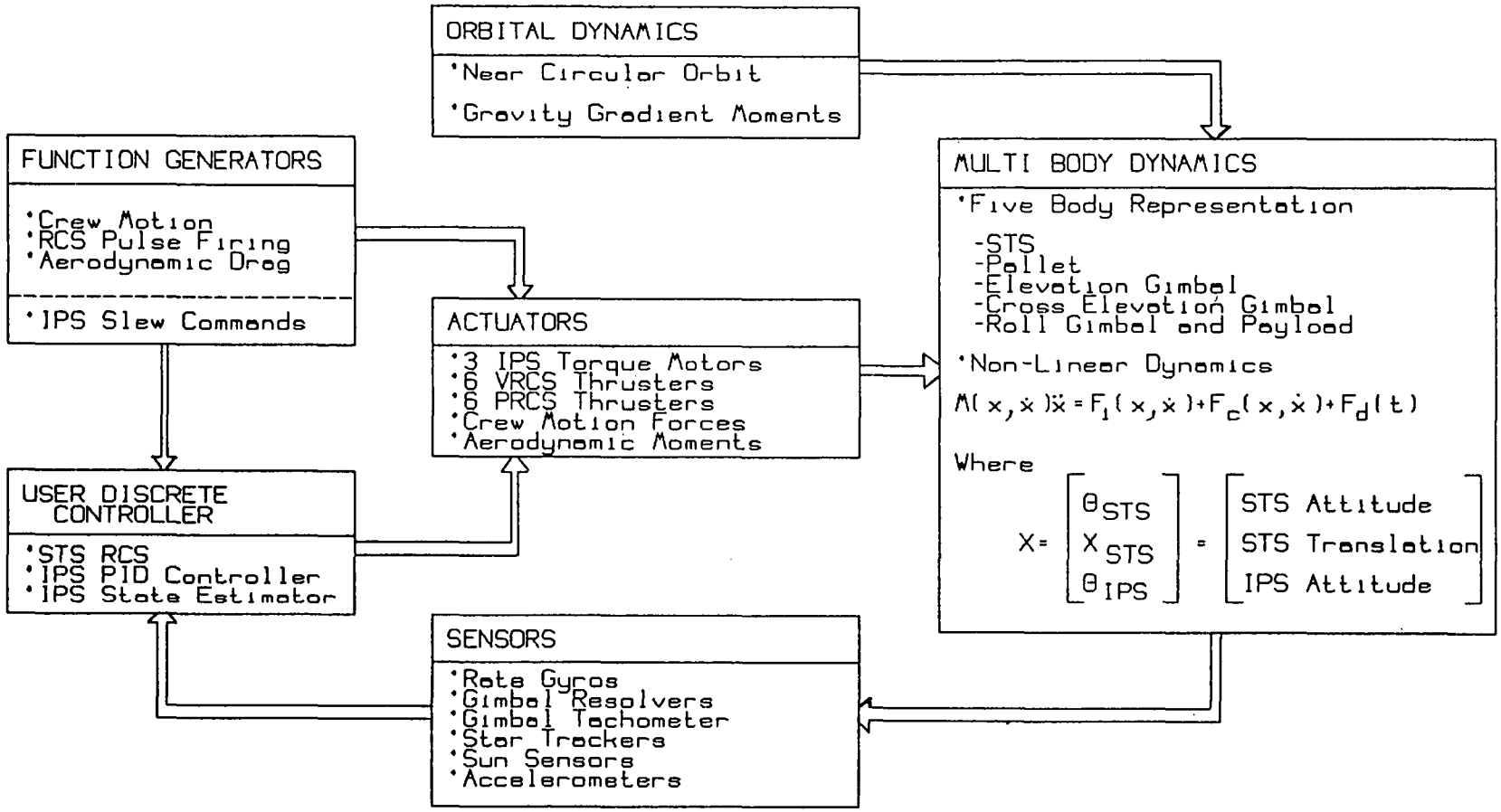


Figure 9. IPS TREETOPS block diagram.

B. Observation matrix equation (75)

$$H(t) = \begin{pmatrix} 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 \\ 0.4 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 \\ -0.4 & 2 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{92}$$

C. State propagation equation (23)

$$\hat{x}_k^- = \Phi_{k,k-1}\hat{x}_{k-1}^+ \tag{93}$$

D. Covariance propagation equation (28)

$$P_k^- = \Phi_{k,k-1}P_{k-1}^+\Phi_{k,k-1}^T + Q_{k-1} \tag{94}$$

E. State update equation (9)

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k\hat{x}_k^-) \tag{95}$$

F. Covariance update equation (22)

$$P_k^+ = [I - K_kH_k]P_k^- \ . \tag{96}$$

C. Linearized Kalman Filter

The LKF coding is divided into three subroutines: LADF, SUBOPT, and GAINMA as provided by Dornier. Derivation of the LKF equations is provided in section III-D and section IV. Implementation of the LKF equations by subroutine is found in appendix D.

1. LADF

A. State transition matrix equation (73)

$$\Phi_{k,k-1} = q_{k-1}{}^T q_k \tag{97}$$

B. State propagation equation (39)

$$\partial\hat{x}_k^- = \Phi_{k,k-1}\partial\hat{x}_{k-1}^+ \tag{98}$$

C. Observation matrix equation (81)

$$\frac{\partial H_i}{\partial x_m} = \frac{1}{S_1{}^2}\left\{S_1\frac{\partial S_{i+1}}{\partial x_m} - S_{i+1}\frac{\partial S_1}{\partial x_m}\right\} \qquad \begin{matrix} i = 1,2 \\ m = 1,\ldots,6 \end{matrix} \tag{99}$$

2. SUBOPT

State update equation (38)

$$\partial\hat{x}_k{}^+ = \partial\hat{x}_k^- + K_k[z_k - h(\bar{x}_k) - H_k(\bar{x}_k)\partial\hat{x}_k^-] \tag{100}$$

3. GAINMA

Processing of time-varying precomputed gains.

D. Extended Kalman Filter

For the EKF, the equations are implemented in three subroutines: LADF, STATE, and GAINMA, and can be found in appendix E. Section III-E and section IV present the derivation of the EKF equations. The state propagation is obtained by the numerical solution of Euler's kinematic equation performed in the fast loop.

1. LADF

A. State transition matrix equationn (73)

$$\Phi_{k,k-1} = q_{k-1}{}^T q_k \tag{101}$$

B. Observation matrix equation (81)

$$\frac{\partial H_i}{\partial x_m} = \frac{1}{S_1^2}\left\{S_1\frac{\partial S_{i+1}}{\partial x_m} - S_{i+1}\frac{\partial S_1}{\partial x_m}\right\} \qquad \begin{matrix} i = 1,2 \\ m = 1,\ldots,6 \end{matrix} \tag{102}$$

2. GAINMA

A. Covariance propagation equation (49)

$$P_k^- = \Phi_{k,k-1}P_k^+\Phi_{k,k-1} + Q_{k-1} \tag{103}$$

B. Gain calculation equation (46)

$$K_k = P_k^- H_k^T(\hat{x}_k^-)[H_k(\hat{x}_k^-)P_k^- H_k^T(\hat{x}_k^-) + R_k]^{-1} \tag{104}$$

3. STATE

State update equation (45)

$$\hat{x}_k^+ = \hat{x}_k^- + K_k[z_k - h(\hat{x}_k^-)] \ . \tag{105}$$

E. Second-Order Kalman Filter

The SOKF equations are implemented in four subroutines: LADF, OBST, GAINS, and STATE, and can be found in appendix F. Derivations of the equations are presented in section III-F and section IV. As with the EKF, the state propagation is obtained by numerical solution of Euler's kinematic equation which was presented in section IV.

1. LADF

State transition matrix equation (73)

$$\Phi_{k,k-1} = q_{k-1}^T q_k \tag{106}$$

2. OBST

Observation matrix equations (81) and (82)

$$\frac{\partial H_i}{\partial x_m} = \frac{1}{S_1^2}\left\{S_1\frac{\partial S_{i+1}}{\partial x_m} - S_{i+1}\frac{\partial S_1}{\partial x_m}\right\} \qquad \begin{matrix} i = 1,2 \\ m = 1,\ldots,6 \end{matrix} \tag{107}$$

$$\frac{\partial^2 H_i}{\partial x_j \partial x_m} = \frac{1}{S_1^3}\left\{S_1^2\frac{\partial^2 S_{i+1}}{\partial x_j \partial x_m} - S_1 S_{i+1}\frac{\partial^2 S_1}{\partial x_j \partial x_m} + 2S_{i+1}\left(\frac{\partial S_1}{\partial x_j}\right)\left(\frac{\partial S_1}{\partial x_m}\right)\right.$$
$$\left. - S_1\left[\left(\frac{\partial S_1}{\partial x_j}\right)\left(\frac{\partial S_{i+1}}{\partial x_m}\right) + \left(\frac{\partial S_{i+1}}{\partial x_j}\right)\left(\frac{\partial S_1}{\partial x_m}\right)\right]\right\} \tag{108}$$

$$i = 1,2 \quad j = 1,\ldots,6 \quad m = 1,\ldots,6$$

3. GAINS

A. Covariance propagation equation (49)

$$P_k^- = \Phi_{k,k-1} P_k^+ \Phi_{k,k-1} + Q_{k-1} \tag{109}$$

B. Gain calculation equation (55)

$$K_k = P_k^- H_k^T(\hat{x}_k^-)[H_k(\hat{x}_k^-)P_k^- H_k^T(\hat{x}_k^-) + R_k + B_k]^{-1} \tag{110}$$

C. Covariance update equation (56)

$$P_k^+ = [I - K_k H_k(\hat{x}_k^-)]P_k^- \tag{111}$$

4. STATE

State update equation (54)

$$\hat{x}_k^+ = \hat{x}_k^- + K_k[z_k - h_k(\hat{x}_k^-) - A_k] \ . \tag{112}$$

Input data required by each of the estimation filters is provided in appendices C through F along with the source code for each filter. The data presented is for the simulation of a nominal

IPS fine pointing objective. Also provided in the appendices are samples of a job control language (JCL), for the Vax 780, which defines the input and output assignments and performs a batch execution of the desired simulation.

VI. SIMULATIONS RESULTS

The simulation results for the four estimation filters will be presented here and discussed with regard to performance and sensitivity to parameter variations. Performance of the filters will be observed first for the nominal pointing objective, which consists of pointing the IPS in the vertical or 90/0/0 gimbal configuration and requires measurements from all three FHST's to be utilized. Operation of IPS with measurements available from all three FHST's is referred to as the nominal pointing case because the LKF gains were generated expecting a star in each of the FHST's. The parameter variations considered for sensitivity purposes include:

1. Star tracker loss

2. Large attitude errors (1°)

3. No system drift estimation.

For each of the simulations presented, except for the large attitude error cases, an initial attitude error of 0.3°, or approximately 1,000 arcsec, will be introduced into all three IPS DOF. For the large attitude error simulations, a 1° offset is input into each gimbal axis. Also, for each case, a drift of approximately 3 arcsec/s is included in each IPS platform axes. To compare the performance of the estimation filters, plots of the three axes attitude responses, of the system drift rate estimations, and of the commanded torque requirements will be presented. Comparison of the four estimation algorithms steady-state attitude responses for all three IPS gimbal axes is presented for the nominal and star loss simulations. A mean steady-state pointing attitude and the standard deviation of the pointing attitude about the mean will be tabulated from the simulations. Also the mean steady-state torque requirements will be given for the simulations with no system drift estimation. Finally, memory and timing requirements for each filter will be discussed.

A. Nominal Pointing

To compare the performance of the four estimation filters, 100-s TREETOPS simulations were executed with attitude errors and system drifts in all three IPS DOF in order to simulate expected flight conditions. Figures 10, 11, and 12 show the IPS attitude time responses for the individual IPS platform axes: elevation (EL), cross-elevation (XEL), and roll, respectively. The legend for the attitude plots is as follows:

○– – – –○ solid line—linear KF

×– – – –× short dashed line—LKF

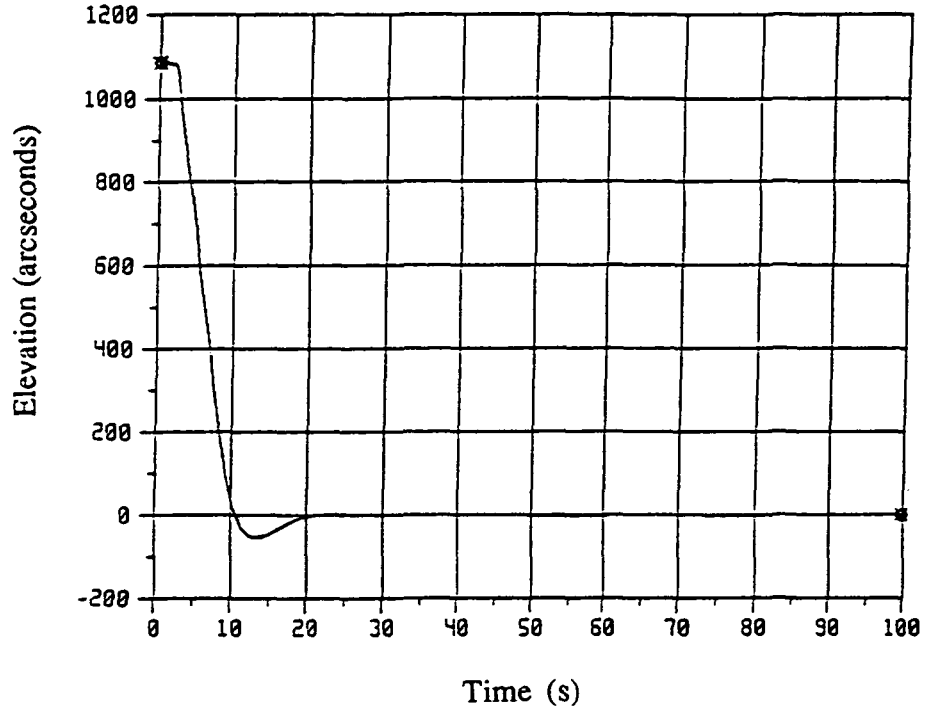


Figure 10. IPS nominal elevation response.

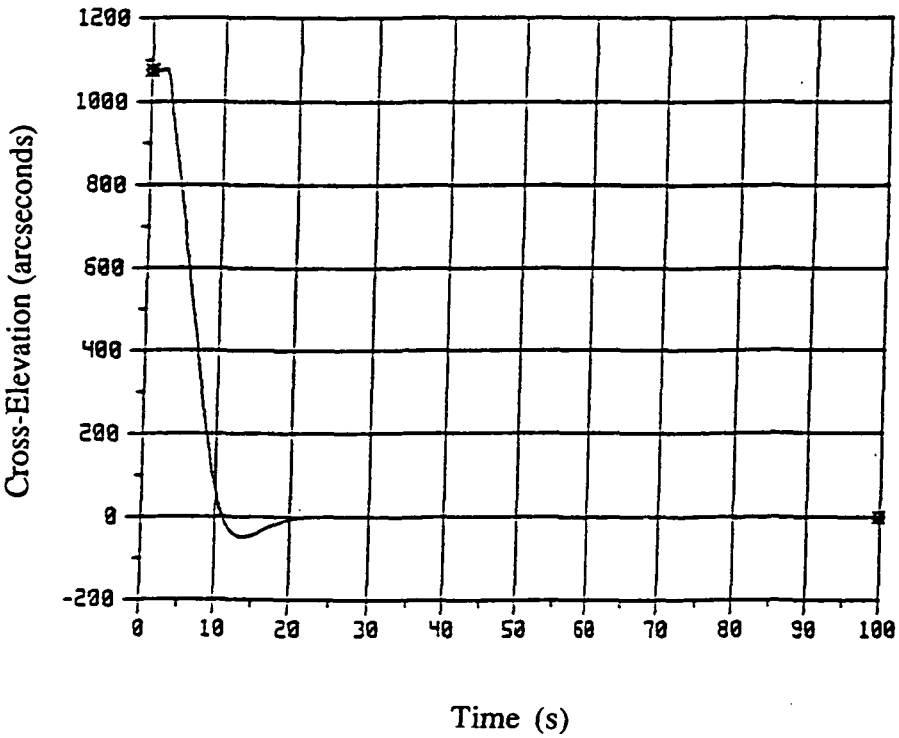


Figure 11. IPS nominal cross-elevation response.

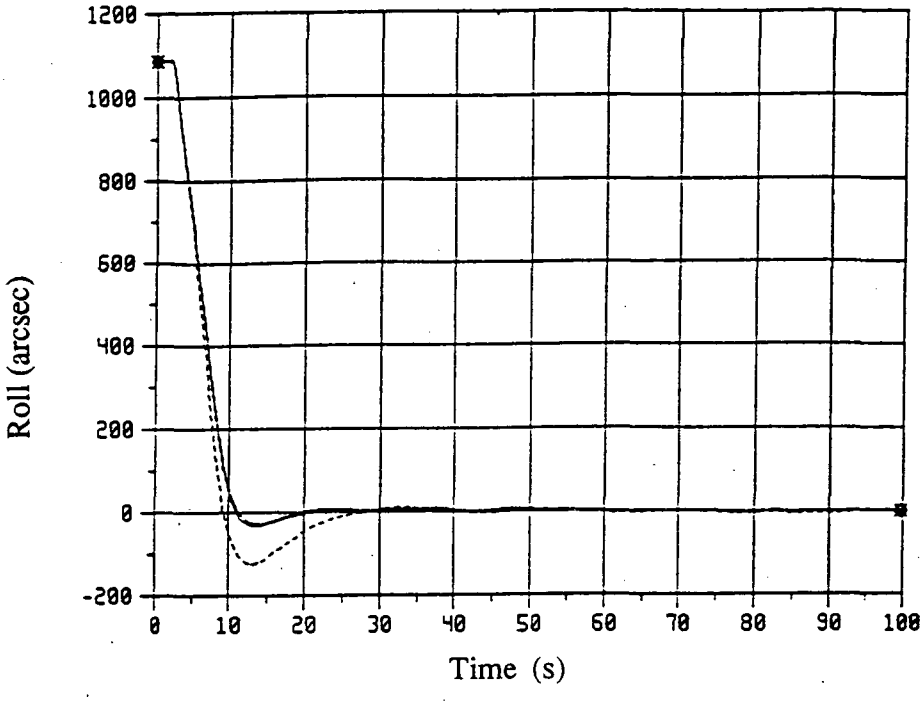


Figure 12. IPS nominal roll response.

Δ– — –Δ long, short dashed line—EKF

□— —□ long dashed line—SOKF.

For the IPS LOS, represented by figures 10 and 11, the performance of all four estimation filters are indistinguishable. A reduced time scale plot for the XEL attitude response is shown in figure 13 to observe the behavior of each filter separately. Evaluation of the IPS roll attitude response in figure 12 shows that the LKF has a faster rise time, larger overshoot, and a longer settling time than the real time filters. Hence, the LKF has a less desirable transient response than the other three estimation algorithms.

System drift estimation by the separate filters are compared for the individual IPS platform axes in figures 14, 15, and 16. The drift rate estimates are plotted in arcsecond per second, using the same legend as that used for the attitude plots. For the nominal pointing case, the three real-time estimation filters show little difference in the IPS system drift determination, while the LKF shows initial undesired transients before converging to the correct estimate of the system drifts.

Further indication of the four ADF time response performances can be obtained from the torque commands generated by the control system found in appendix A. Figures A-1, A-2, and A-3 present the torque outputs, in Newton-meters, for the three IPS torque motors. Each figure contains individual plots of the torque commands generated by the separate estimation filters. Again, the LOS torque commands for the four filters appear very similar, while the LKF roll

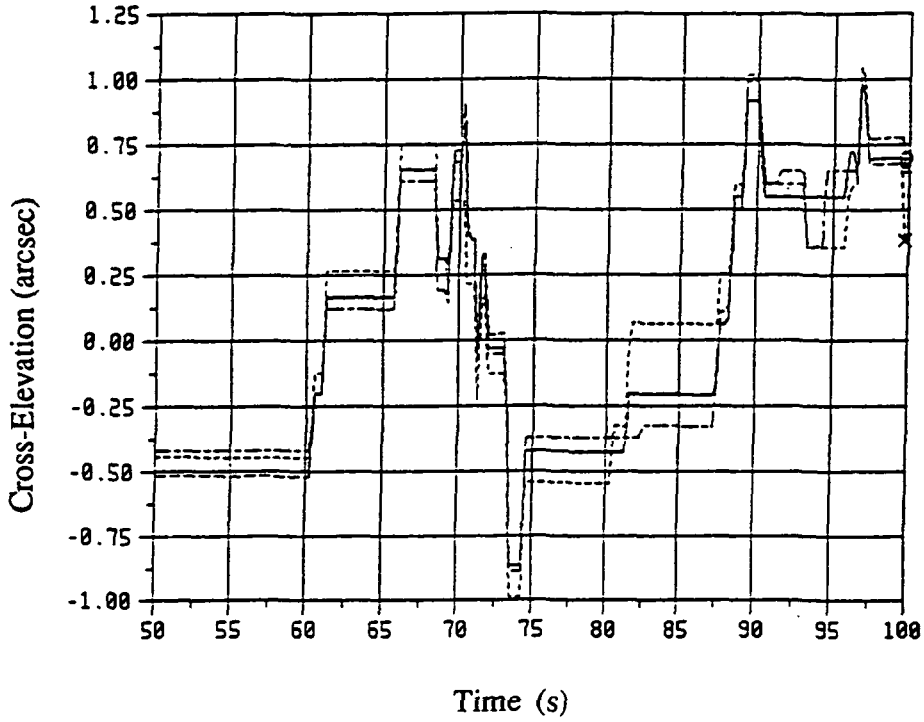


Figure 13: IPS nominal cross-elevation response (reduced time scale).

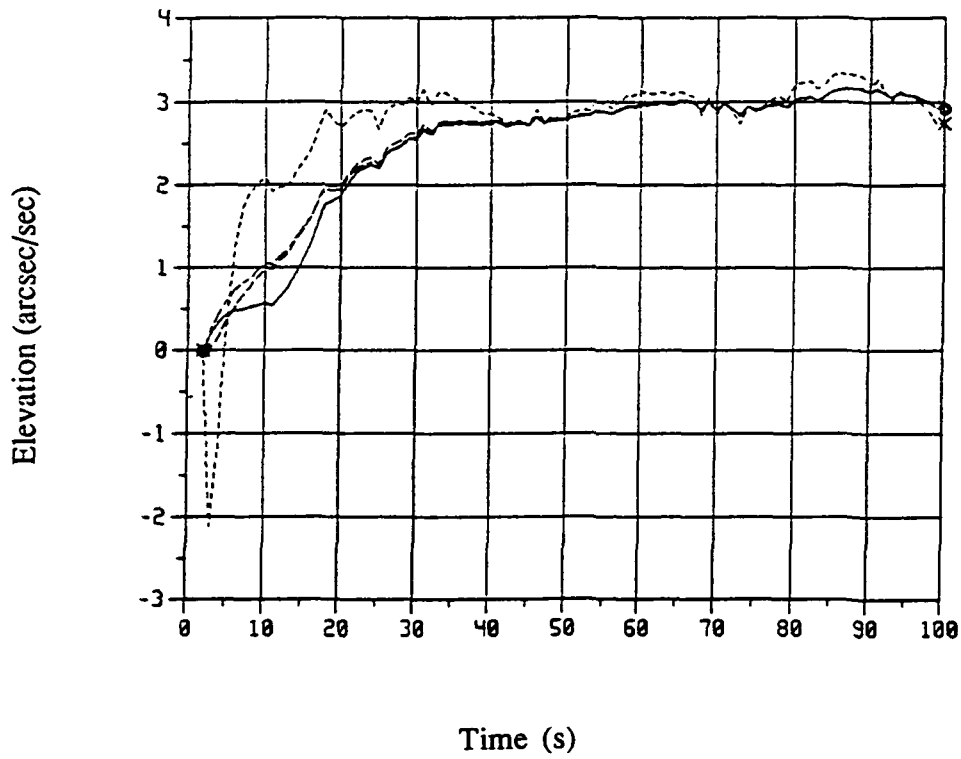


Figure 14. IPS nominal elevation drift estimation.

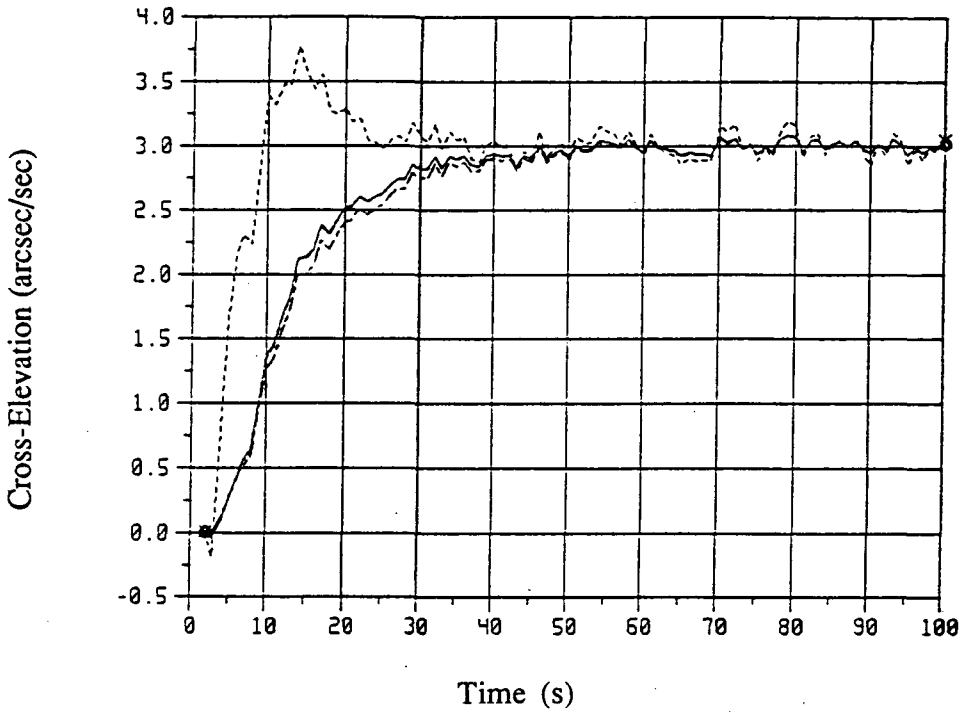


Figure 15. IPS nominal cross-elevation drift estimation.

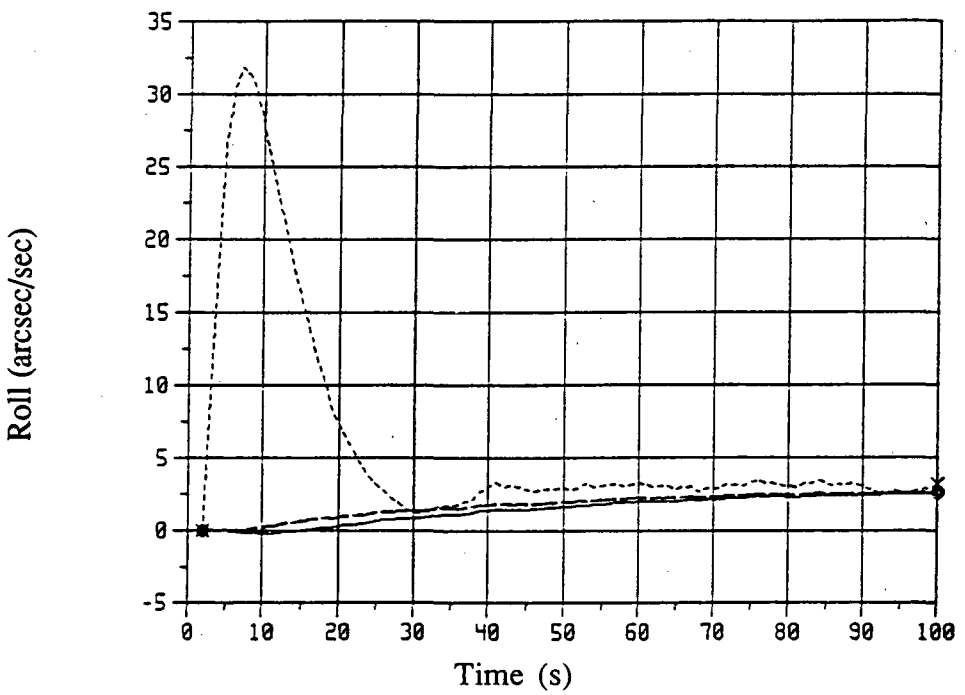


Figure 16. IPS nominal roll drift estimation.

torque exhibits an undesired oscillation. Figure A-4 shows the roll torque commands on a smaller time scale. Since the states of the estimation filters are inputs to the fast loop controller, the torque plots provide an indication of the effect of the ADF estimates upon the 25-Hz control loop.

B. Star Loss

The simulations presented in this section are for IPS pointing objectives when measurements from the right skew star tracker are not available. The lack of FHST measurements can arise for several reasons, such as star tracker component failure, no unique bright stars in the tracker FOV, or unexpected star brightness variability. Figures 17, 18, and 19 show the attitude responses for the four estimation filters. For all three DOF, the three real-time filters exhibit similar responses, while the LKF shows varying degrees of degraded stability depending upon the gimbal axis observed. Also, a comparison of the nominal fine pointing operation versus the two tracker fine pointing operation is presented in figure 20 for a cross-elevation attitude error. All four estimation filters are included in the figure for both simulations. It can be readily observed that the LKF is sensitive to a loss of a star trackers's measurements, while the three real-time filters show no noticeable sensitivity.

Review of the torque command plots for the two tracker simulations in figures A-5, A-6, and A-7 show increased oscillations in all three axes of the LKF simulations. The real-time filters show no appreciable variation of the torque commands. Figures 21, 22, and 23 present the estimation of the system drifts by the four estimation filters for the three IPS platform axes. The LKF

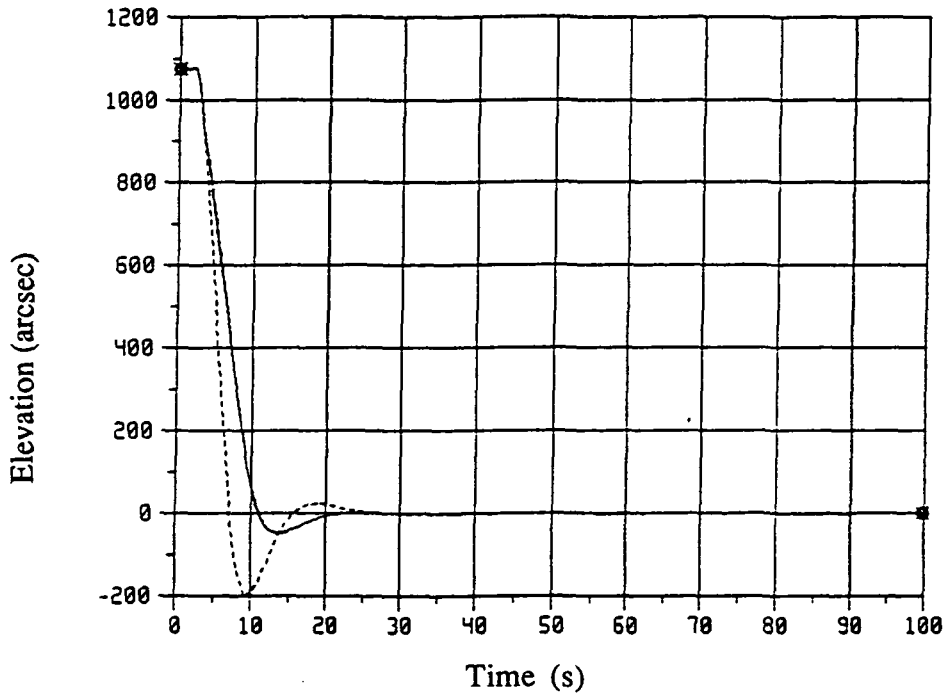


Figure 17. IPS two tracker elevation response.

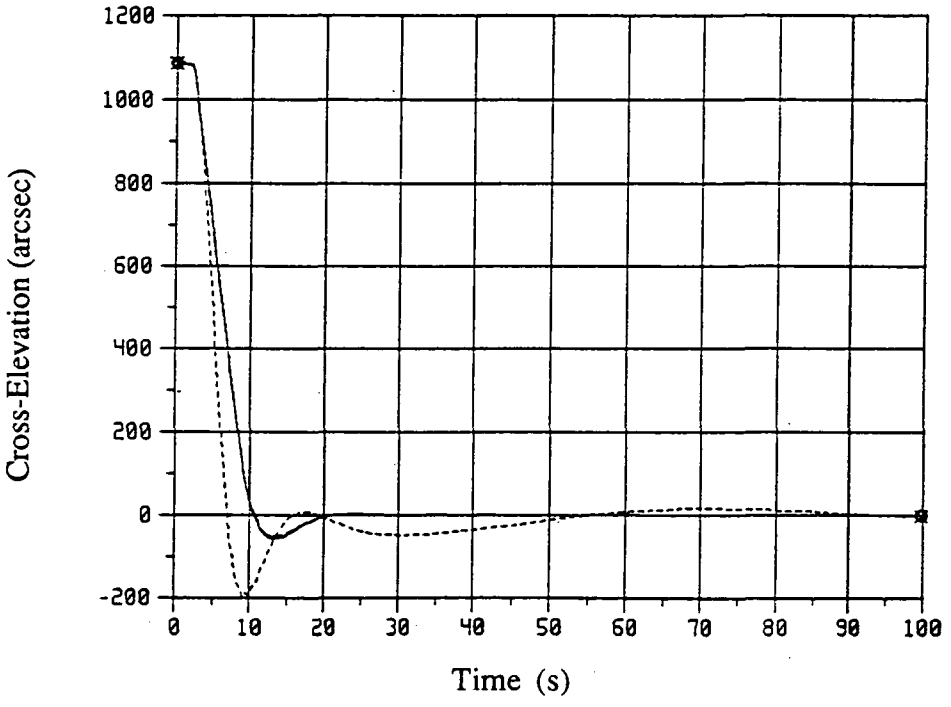


Figure 18. IPS two tracker cross-elevation response.

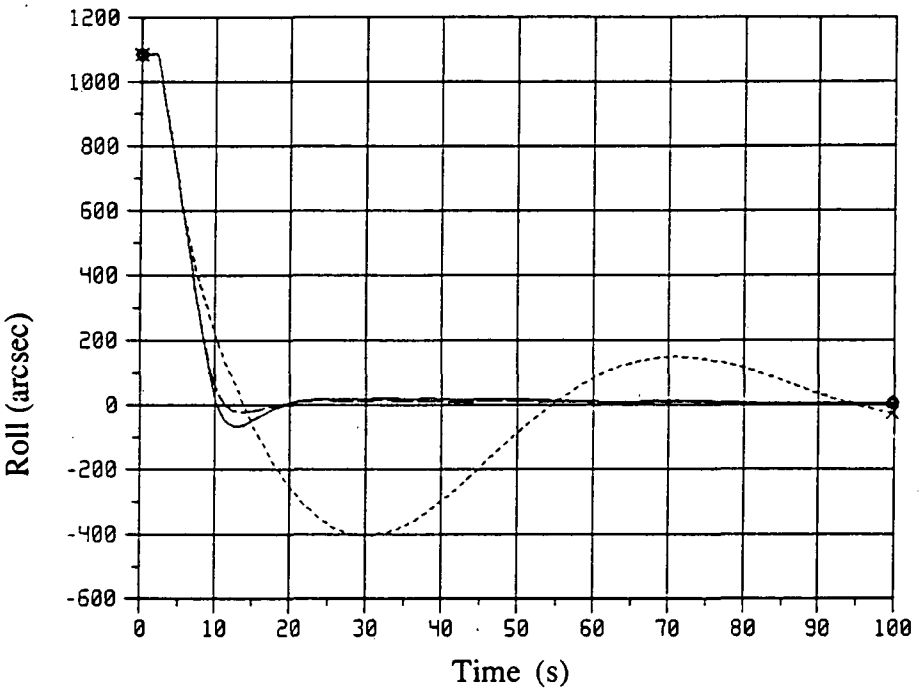


Figure 19. IPS two tracker roll response.

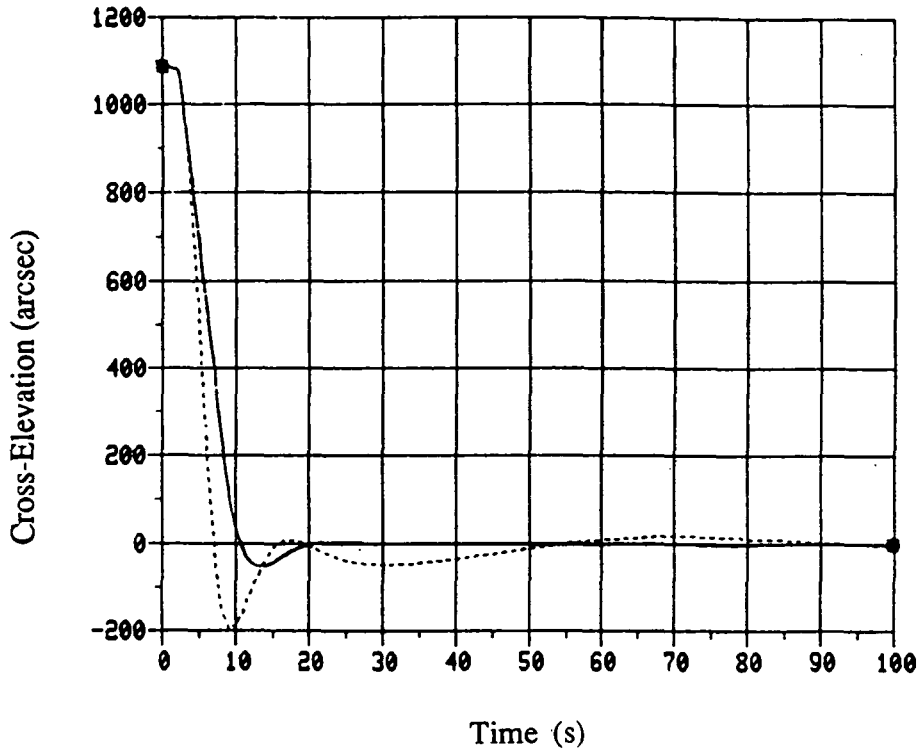


Figure 20. Nominal versus two tracker response.

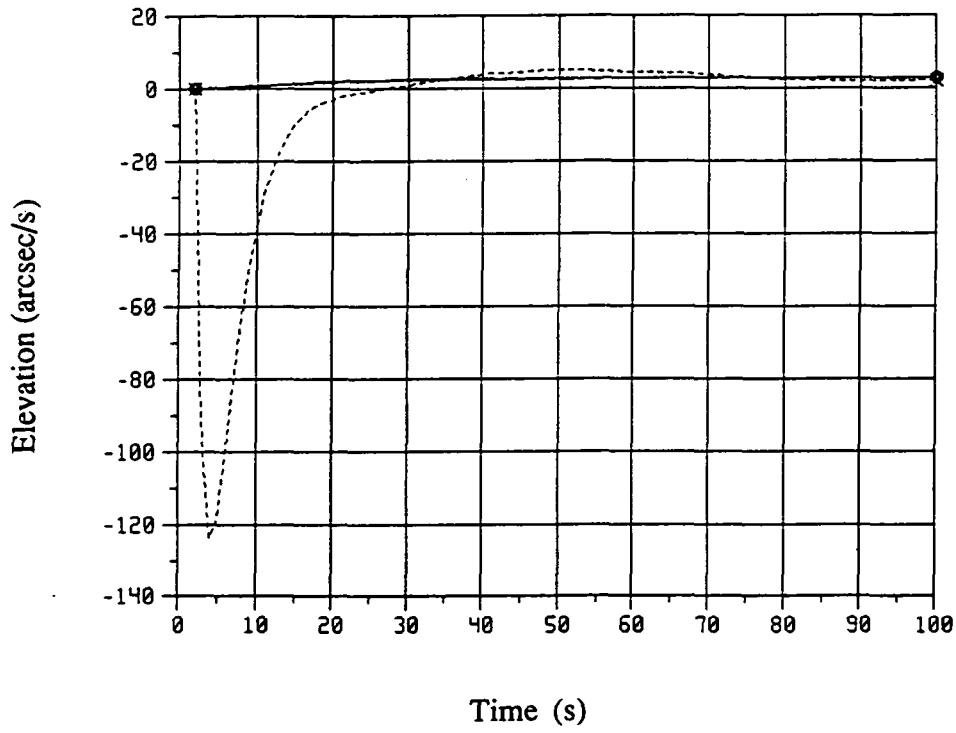


Figure 21. IPS two tracker elevation drift estimation.

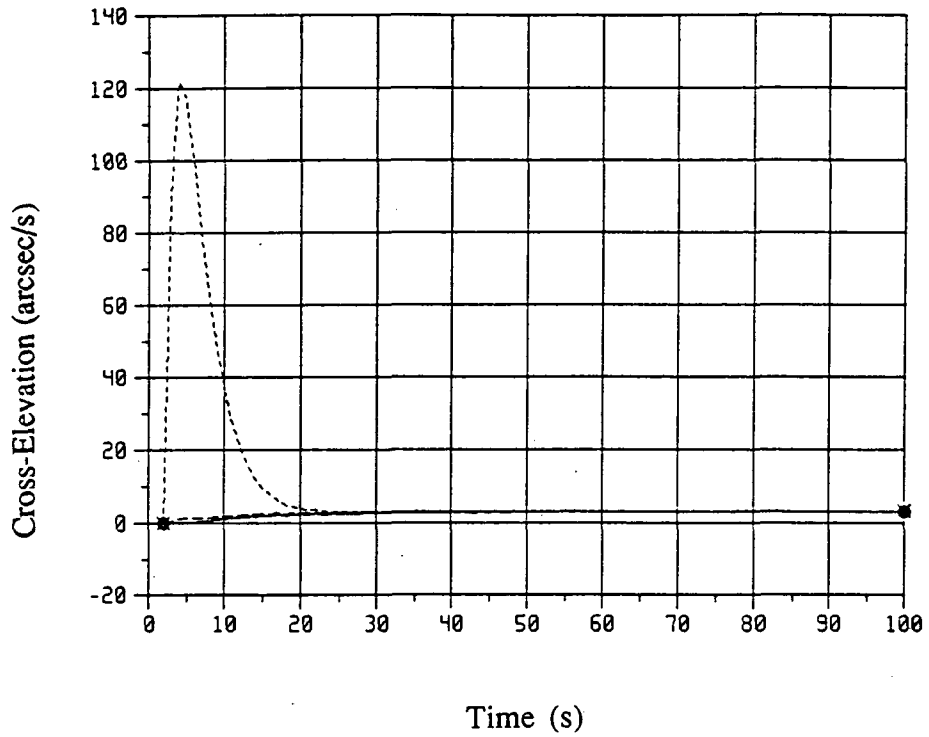


Figure 22. IPS two tracker cross-elevation drift estimation.

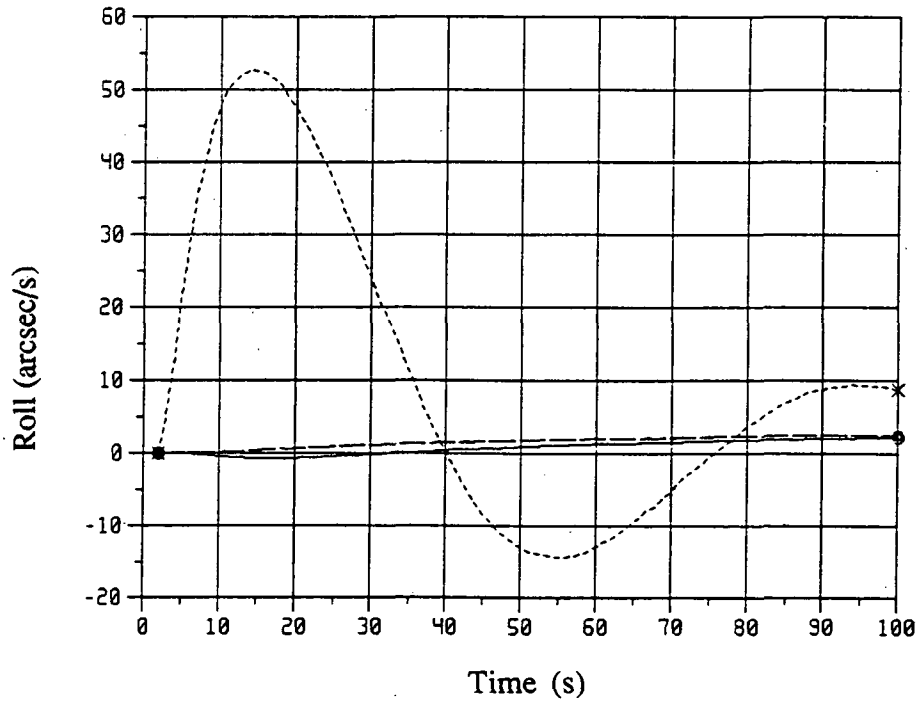


Figure 23. IPS two tracker roll drift estimation.

drift estimates have large initial transients before converging to the modeled system drift rate. Estimation of the system drifts by the real-time filters are observed to be more well behaved. Overall it can be seen that the fixed gain schedule of the LKF produces a system which does not compensate for a deviation from the assumed system definition, such as a loss of star tracker measurements.

C. One-Degree Attitude Error

In this section the sensitivity of the estimation filters to a larger than expected initial attitude error and the loss of a star tracker's measurements is presented. Comparison of the three IPS axes attitude errors for the four estimation filters is presented in figures 24, 25, and 26. Again the three real-time filters produce desirable and similar time responses, while the reduced stability of the LKF is more pronounced. In figure 24 the apparent improved performance of the LKF filter is due to an erroneous initial drift estimate, which will be discussed further in section VI-D. Figure 27 shows a reduced time scale of the cross-elevation attitude response. As in previous sections, plots of the simulated IPS torque commands for the 1° simulations can be found in appendix A. The LKF produces large torque commands over a longer duration than the three real-time filter implementations, thereby requiring more power capability.

Figures 28, 29, and 30 are plots of the drift estimates by the four filters. Estimation of the LOS system drifts by the three real-time filters converges to the actual drift values in a smooth trajectory, while the LKF initially produces large erroneous drift estimates before settling to the proper values. Determination of the roll drift by the EKF and SOKF converge smoothly, while the KF roll drift estimation shows a small erroneous transient before approaching the simulated drift value. Estimation of the roll system drift by the LKF is shown to have an undesirably large transient, with a long settling time.

The three real-time estimation filters show no noticeable sensitivity to a large unexpected initial attitude error. The small sensitivity observed in the KF can be attributed to the loss of a star tracker's measurements as was shown in section VI-B. However, the sensitivity of the LKF to a loss of star tracker measurement was shown to be dependent upon the magnitude of the initial attitude error.

D. System Drift Estimation

The objective of this section is twofold; first, to show the need to estimate the system drifts, and second, to show the effect of erroneous drift estimates. A close examination of the previous sections shows a correlation between the erroneous initial estimation of the system drifts and the variation of the attitude time response from the nominal trajectory. To determine the effect of the system drift on the attitude determination problem, the drift estimates were not utilized to correct the biased gyro measurements. As with the previous simulations, the 3 arcsec/s system drift is introduced into the system. Also, the left skew tracker measurements are not processed in the simulations presented in this section.

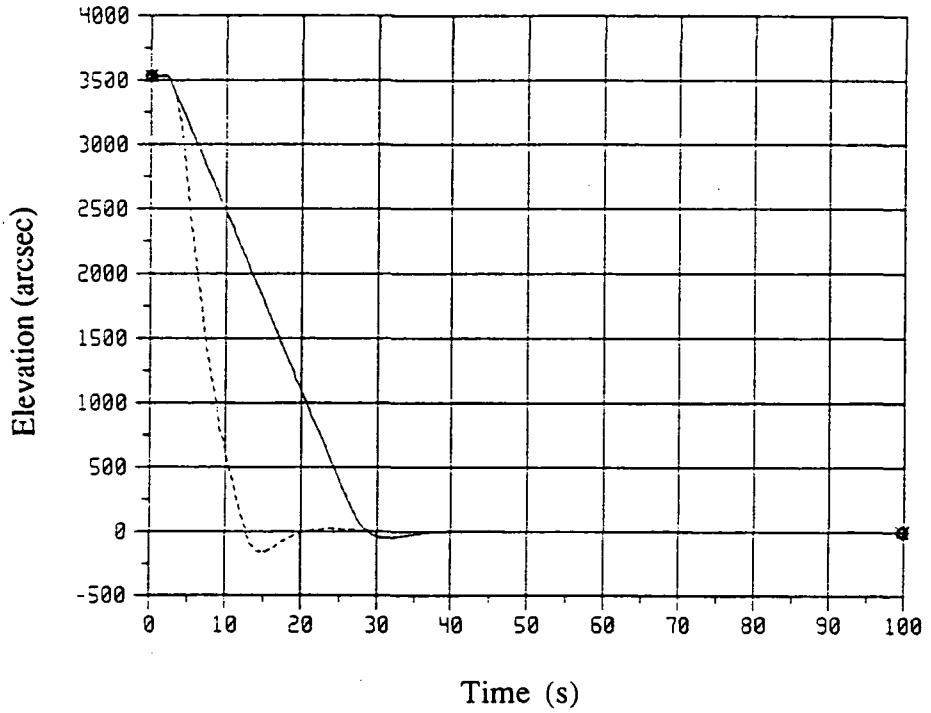


Figure 24. IPS 1° elevation response.

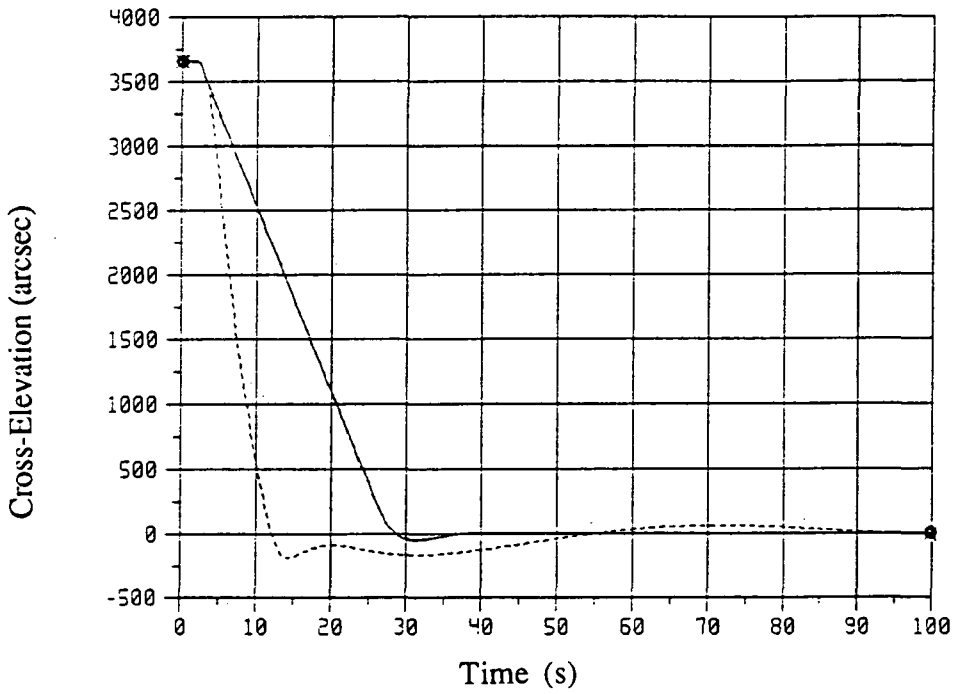


Figure 25. IPS 1° cross-elevation response.

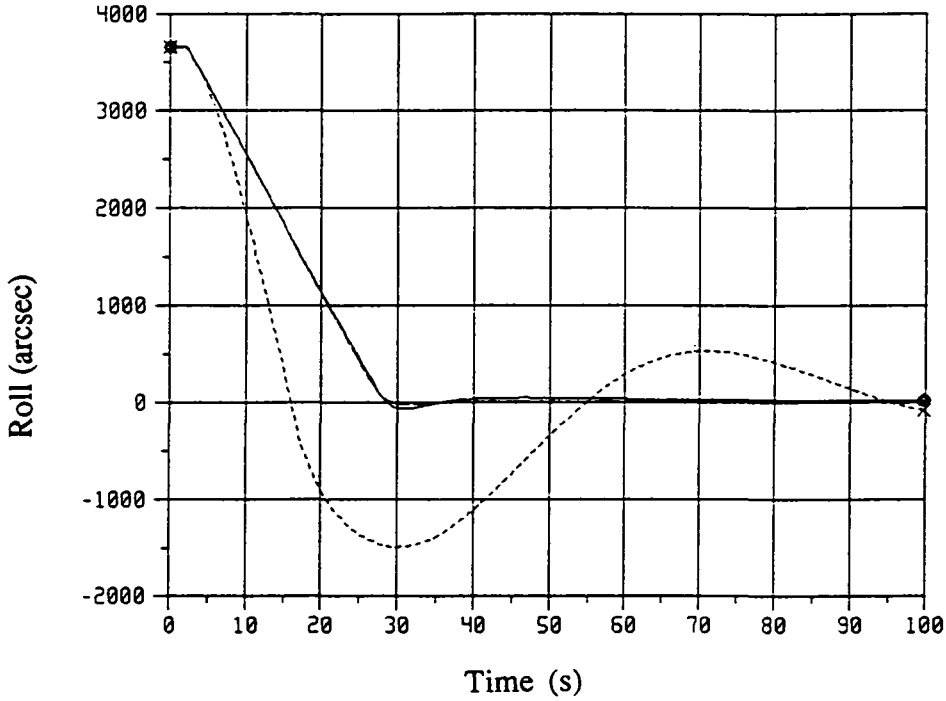


Figure 26. IPS 1° roll response.

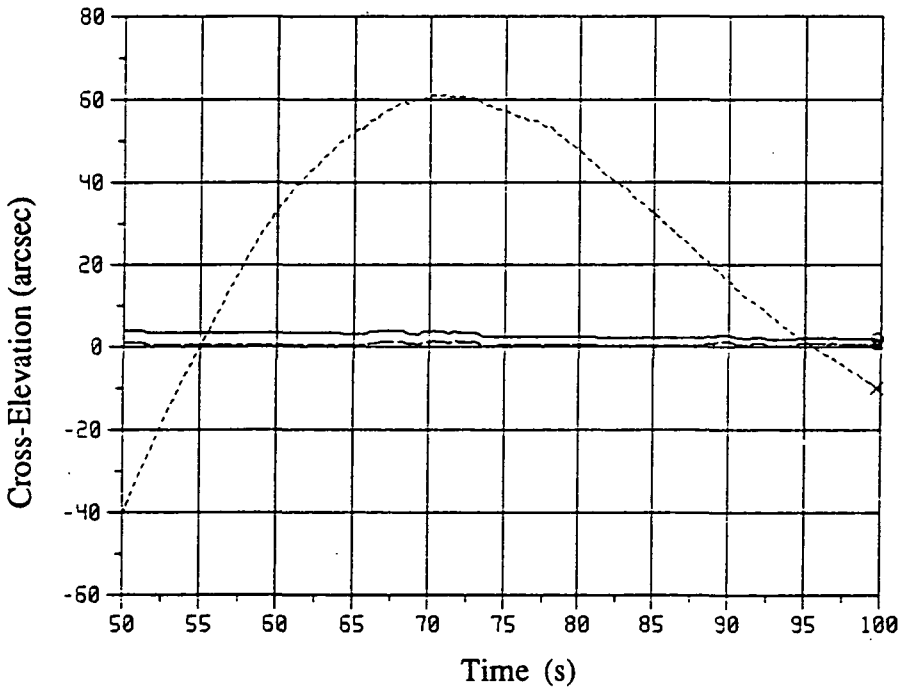


Figure 27. IPS 1° cross-elevation response (reduced time scale).

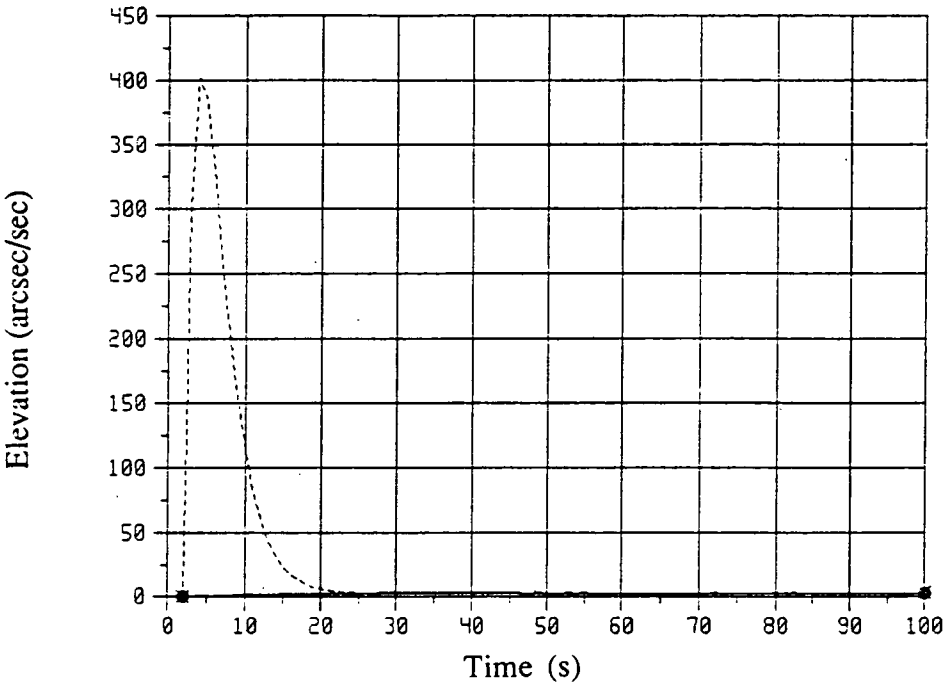


Figure 28. IPS 1° elevation drift estimation.

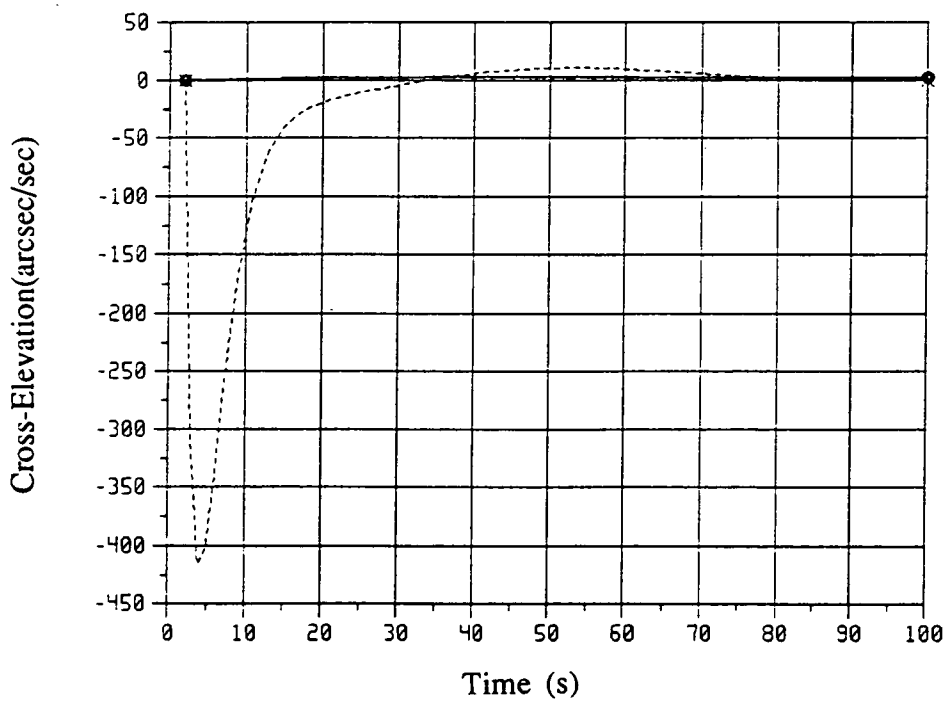


Figure 29. IPS 1° cross-elevation drift estimation.

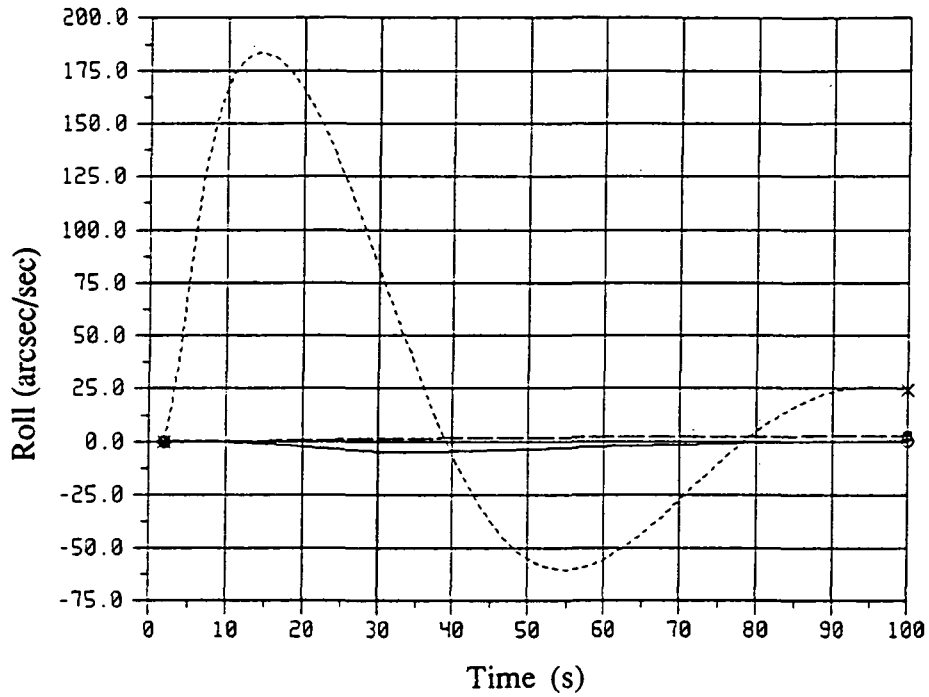


Figure 30. IPS 1° roll drift estimation.

Figures 31, 32, and 33 show the IPS attitude response for each of the IPS gimbal axes. For the LOS response, the three real-time filters produce the nominal response, while the LKF response has a slightly smaller overshoot. Also, a slight bias can be observed in the LKF's steady-state attitude. In figure 33, which shows the roll attitude response, the LKF exhibits a much longer transient phase than the three real-time filters. The KF shows a slightly larger overshoot than the two nonlinear real-time estimation filters, while a steady-state bias can be seen in all four estimation filters' attitude responses. A reduced-time scale plot of the elevation axis attitude response is presented in figure 34. In the reduced-time scale plot, a bias can also be observed in the steady-state LOS response for all four estimation filters; however, the LKF shows a larger bias than the three real-time filters.

Figure 35 compares the cross-elevation attitude response produced by drift-compensated simulations (section VI-B) versus the uncompensated drift simulations for the LKF and EKF implementations. The two EKF attitude responses are indistinguishable from each other, while the LKF response for the uncompensated drift simulation no longer shows the faster rise time than nominal and the large overshoot of the LKF drift compensated simulation. In fact, the uncompensated LKF simulation shows a slightly slower than nominal response with a smaller overshoot. It is concluded that the sensitivity observed in the initial transient of the LKF attitude response largely results from the drift estimation, and that the smooth trajectory of the real-time filter's drift estimation results in a more robust attitude response. Therefore, since the improved time response of the LKF shown in figure 24 can be attributed to an erroneous drift estimation, the behavior is undesirable.

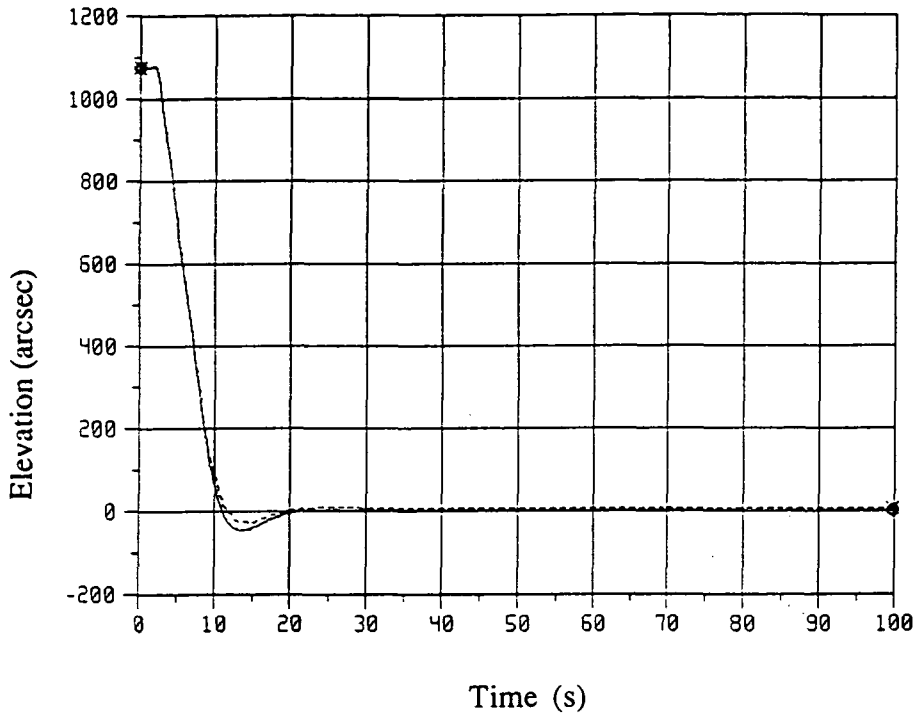


Figure 31. IPS elevation response without drift compensation.

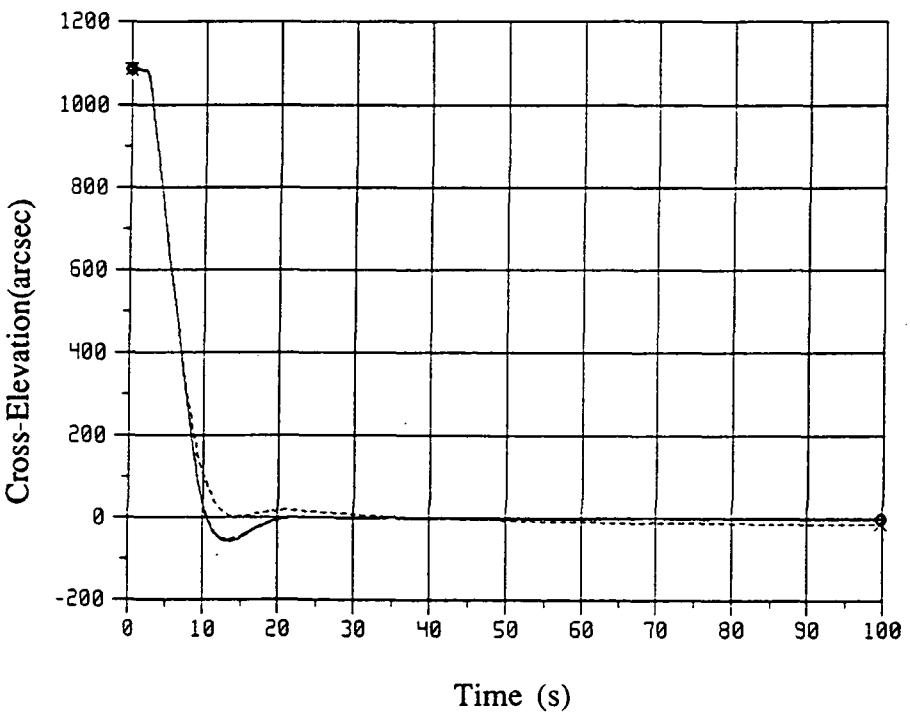


Figure 32. IPS cross-elevation response without drift compensation.

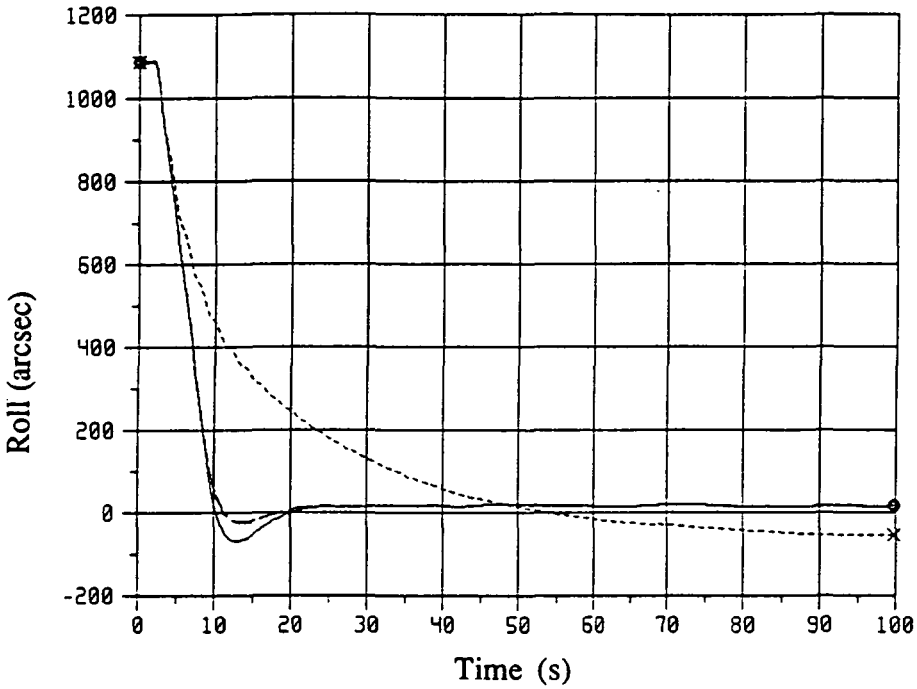


Figure 33. IPS roll response without drift compensation.

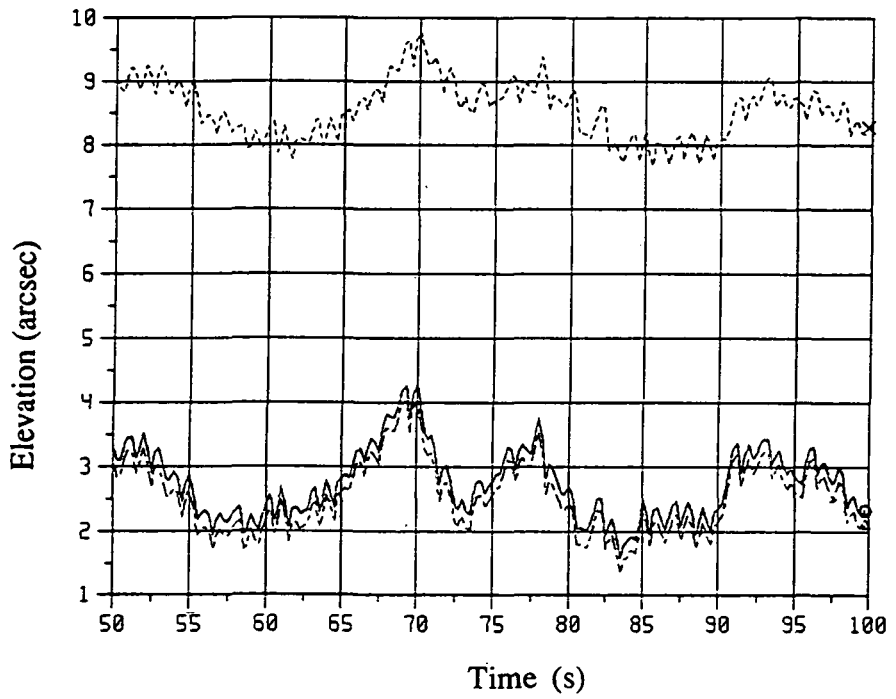


Figure 34. IPS elevation response without drift compensation (reduced time scale).

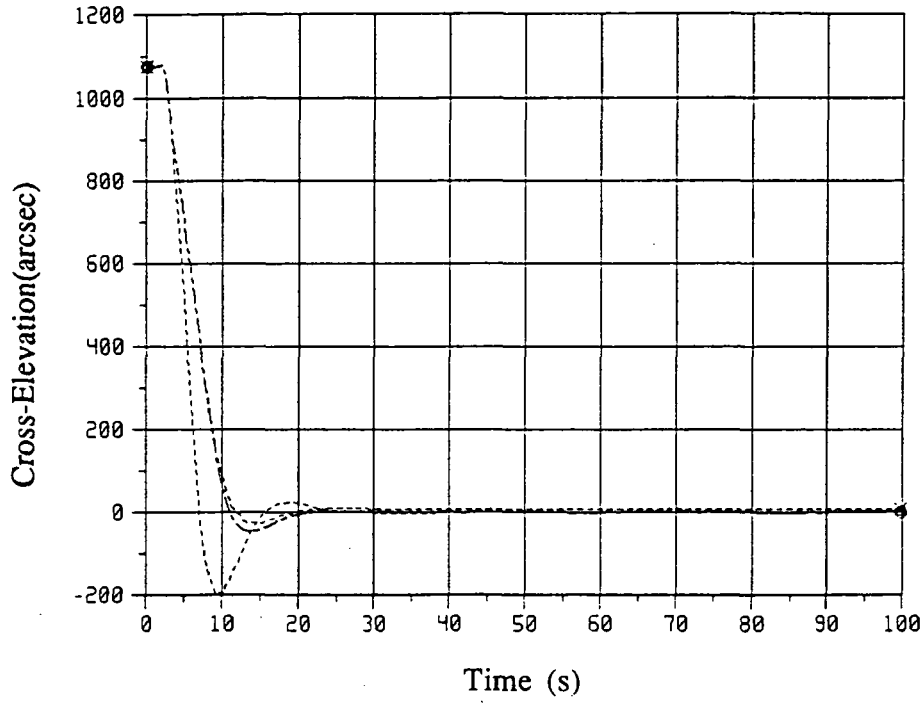


Figure 35. Comparison of drift compensated and uncompensated simulations for the LKF and EKF.

In order for the attitude errors to achieve a steady-state bias in the presence of an uncompensated drifting gyro, the estimation filters must generate a constant torque command. In appendix A, figures A-11, A-12, and A-13 give the torque commands for the three IPS gimbal axes. Each figure contains a separate plot of the four estimation filters. A non-zero bias can be found in the steady-state torque commands resulting from the attitude updates generated by each of the filters. Table 2 compares the mean steady-state value of the torque commands from the simulations of section VI-B and the uncompensated drift simulations performed for this section.

The mean values shown in table 2 are computed from the last 50 s of the respective simulations. As expected, a small increase in the mean torque commands is obtained for all estimation filters when the system drifts are not compensated in the fast loop controller.

Table 2. Comparison of torque commands for drift compensated and uncompensated simulations.

Mean Compensated/Uncompensated Torque (Nm)			
Filter	Roll	Elevation	Cross-Elevation
KF	0.008/0.01	0.015/0.09	0.003/0.08
LKF	0.014/0.02	0.006/0.09	0.004/0.08
EKF	0.005/0.01	0.016/0.09	0.002/0.08
SOKF	0.005/0.01	0.016/0.09	0.003/0.08

E. Pointing Performance

Table 3 presents the IPS steady-state fine pointing performance of the four estimation filters for both the nominal and star loss simulations. The pointing accuracy is the mean attitude in each axis given in arcseconds ($\widehat{sec}$), while the stability is the root mean square (rms) of the attitude, also given in arcseconds. The pointing accuracy and pointing stability were computed from the last 50 s of the 100-s simulations.

For the nominal simulation, the LOS (pitch and yaw axes) accuracy and stability of all four estimation filters are within the accuracy and stability requirements specified in section II and are comparable. In the roll axis, the LKF shows a better accuracy, but slightly larger variation than the three real-time filters.

The loss of a star tracker's measurements results in larger biases and variances by all four estimation filters, although, the pointing accuracy and stability for the real-time filters is considerably better than for the LKF. In fact, the LKF produces results which exceed the desired pointing performance presented in section II.

F. Data Systems Requirements

As in any evaluation of the performance of different software implementations, the performance improvements must be weighed against the implementation requirements on the data system. Table 4 presents the memory and execution requirements for the individual estimation algorithms. The memory requirements are for the compiled FORTRAN source code of each estimation filter. To obtain the execution time in milliseconds (ms) required by each estimation filter, the time before the ADF routines are called by the fast loop is differenced with the time at which execution is returned to the fast loop. An average of the execution time is performed over the 100-s simulation duration.

From table 4 it is seen that the simplicity of the linear KF greatly reduces the data system memory requirement, while the memory requirements of the nonlinear filters increase with the complexity of the estimation filter implementation; where the LKF is the simplest nonlinear implementation due to the utilization of preflight computed gains and the SOKF is the most complex due to inclusion of second-order linearization terms. As for timing requirements, the use of preflight computed gains by the LKF eliminates the need to invert a 6×6 matrix performed in the gain calculation. Therefore, the LKF has the smallest execution time requirement. As would be expected, for the real-time estimation filters the more complex the estimation filter implementation the greater the execution time. This concludes the presentation of the simulation results, the following section will present the conclusions drawn from these results.

Table 3. IPS simulation quiescent pointing performance.

Filter	Nominal Pitch	Nominal Yaw	Nominal Roll	Star Loss Pitch	Star Loss Yaw	Star Loss Roll
KF						
Accuracy	.140 arcsec	.009 arcsec	2.350 arcsec	.177 arcsec	.980 arcsec	9.770 arcsec
Stability (rms)	.518 arcsec	.488 arcsec	1.850 arcsec	.574 arcsec	.340 arcsec	3.532 arcsec
LKF						
Accuracy	.138 arcsec	.038 arcsec	0.201 arcsec	.216 arcsec	7.826 arcsec	68.129 arcsec
Stability (rms)	.550 arcsec	.490 arcsec	2.290 arcsec	.614 arcsec	7.421 arcsec	64.617 arcsec
EKF						
Accuracy	.140 arcsec	.021 arcsec	1.715 arcsec	.180 arcsec	.584 arcsec	5.804 arcsec
Stability (rms)	.520 arcsec	.486 arcsec	1.714 arcsec	.576 arcsec	.332 arcsec	2.634 arcsec
SOKF						
Accuracy	.140 arcsec	.003 arcsec	1.652 arcsec	.185 arcsec	.577 arcsec	5.791 arcsec
Stability (rms)	.517 arcsec	.492 arcsec	1.709 arcsec	.574 arcsec	.336 arcsec	2.667 arcsec

Table 4. Estimation filters data system requirements.

Filter	Memory	Timing
KF	5,275 bytes	38 ms
LKF	15,231 bytes	18 ms
EKF	16,867 bytes	42 ms
SOKF	21,760 bytes	132 ms

VII. CONCLUSIONS

The intent of this report was to develop three real-time estimation filters for the attitude determination of the IPS and compare the performance of the real-time filters with the performance of the current LKF implementation. It has been shown that the implementation of three different real-time filters is feasible and their performances provide similar results to the LKF for nominal pointing objectives. Comparison of the four estimation filters' robustness was performed by varying simulation initial conditions. Results of the parameter variations showed that the real-time filters were essentially insensitive to changes in simulation parameters, while the LKF exhibited undesirable transient responses, which indicates a sensitivity to parameter changes. Besides improved robustness, the inflight calculation of the estimation filter gains by the real-time filters reduces preflight computations, verification, and documentation thereby reducing the mission costs. The impact of the increased computational burden of the real-time filters is not available, further study will be required to determine if a one-time processor upgrade cost is necessary to meet the real-time filters timing requirements. Also, since the transient responses of the real-time filters are insensitive to anomalous pointing operations, i.e., the settling times are not increased, longer intervals of scientific data gathering is possible. Longer periods in which to obtain scientific data also effectively reduce mission costs. Therefore, it can be concluded that the implementation of a real-time estimation filter is feasible and will reduce recurring mission costs. Over a multiple mission life time of the IPS, the development cost of implementating a real-time filter may be offset by the actual reduction in mission costs. Moreoever, which real-time filter is to be implemented must also be considered. Although the linear KF showed a slight sensitivity to parameter variation, the assumption of a linear estimation process for the small angle fine pointing operations was justified. Also the cost of implementing the KF is less than for the nonlinear algorithms, primarily since the KF has fewer calculations, which results in less stringent data system requirements. If the IPS pointing requirements were altered to justify the additional performance improvements observed by accounting for the nonlinear effects, then the EKF implementation should be selected. The simulations showed that the inclusion of the second-order terms in the linearization process by the SOKF did not produce significant improvements in the IPS performance. In conclusion, based upon the current pointing requirements and estimation filter implementation costs compared with performance improvements, the suggested real-time implementation is a linear KF.

APPENDIX A

IPS Torque Commands

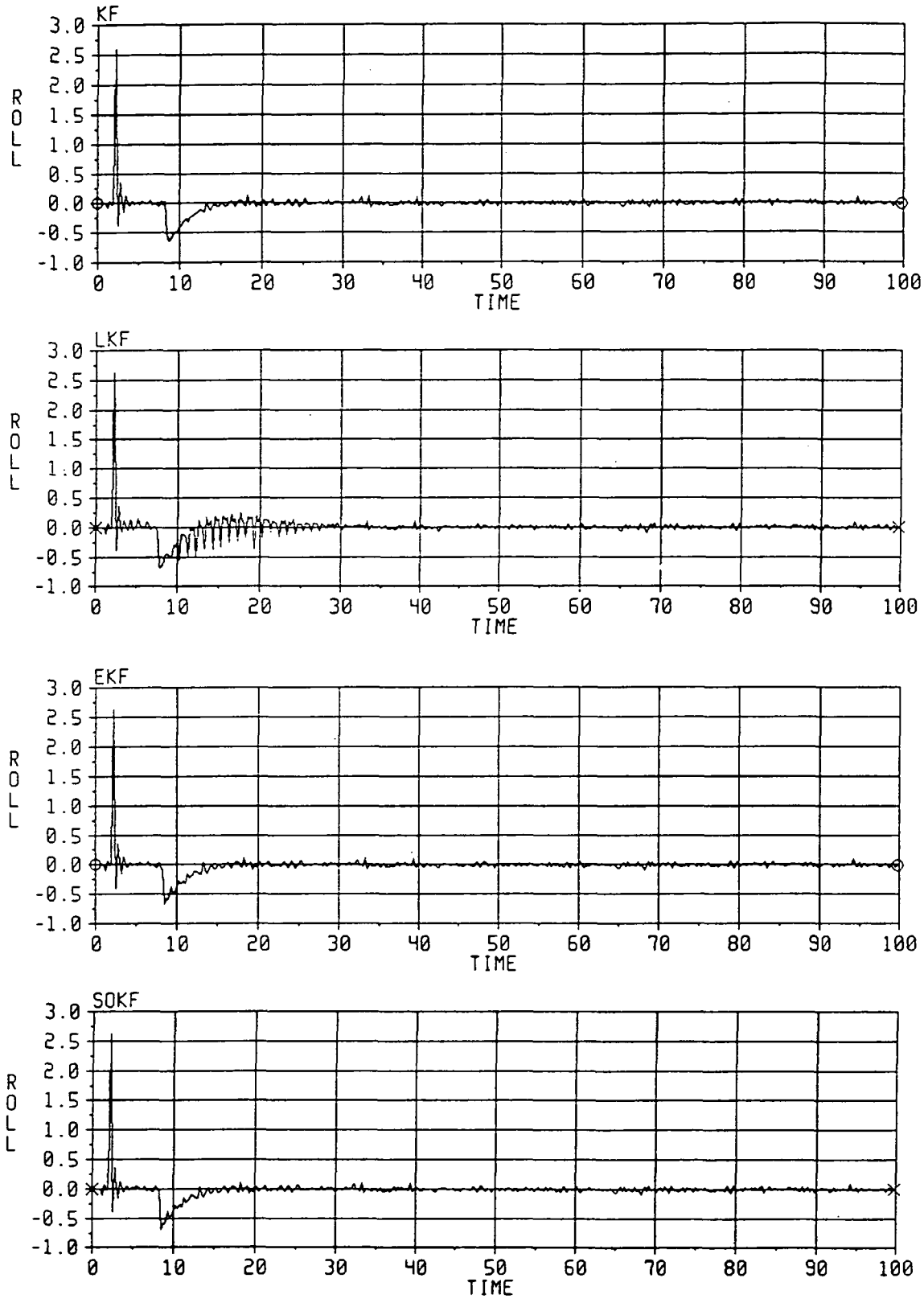


Figure A-1. IPS nominal roll torque commands.

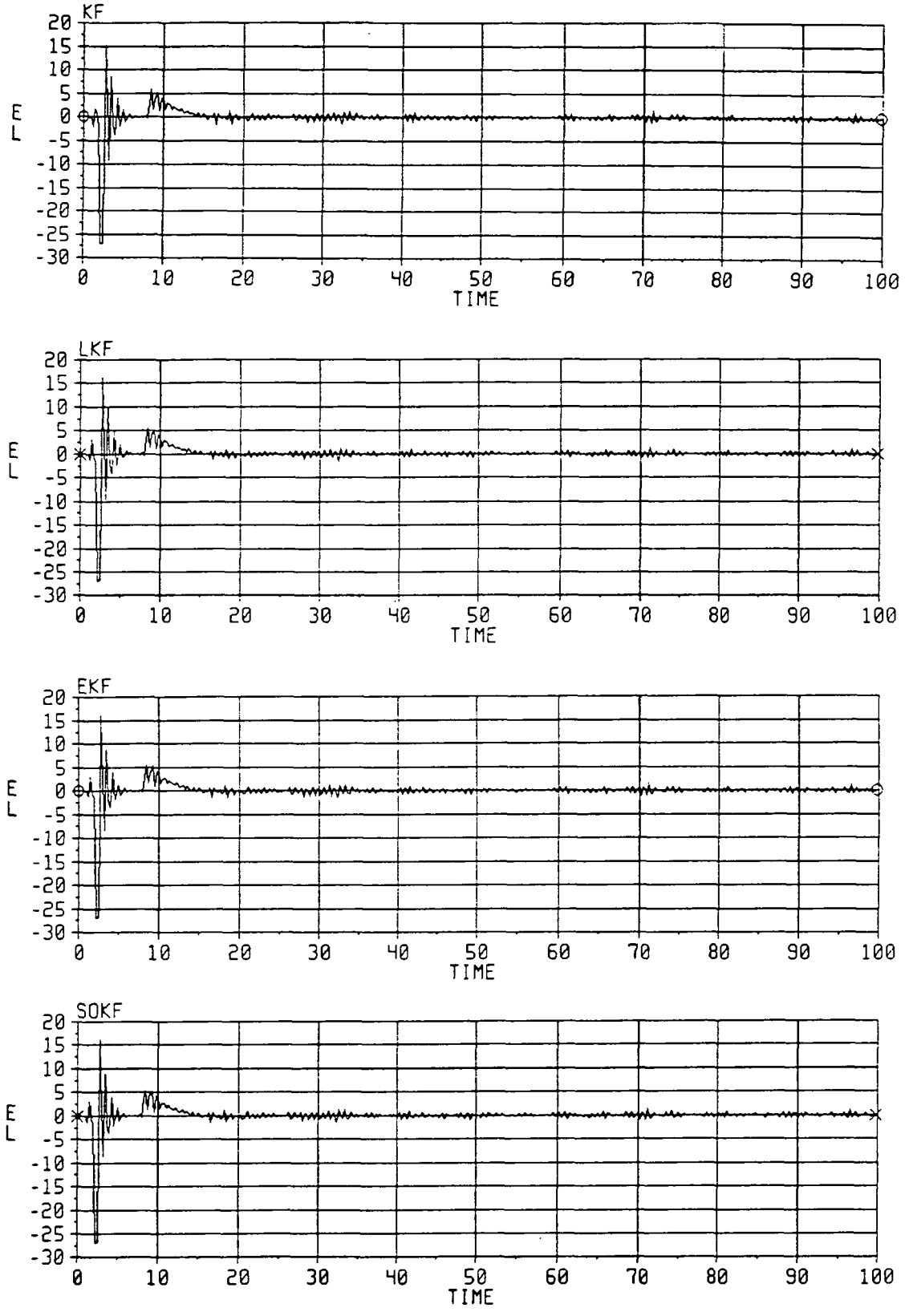


Figure A-2. IPS nominal elevation torque commands.

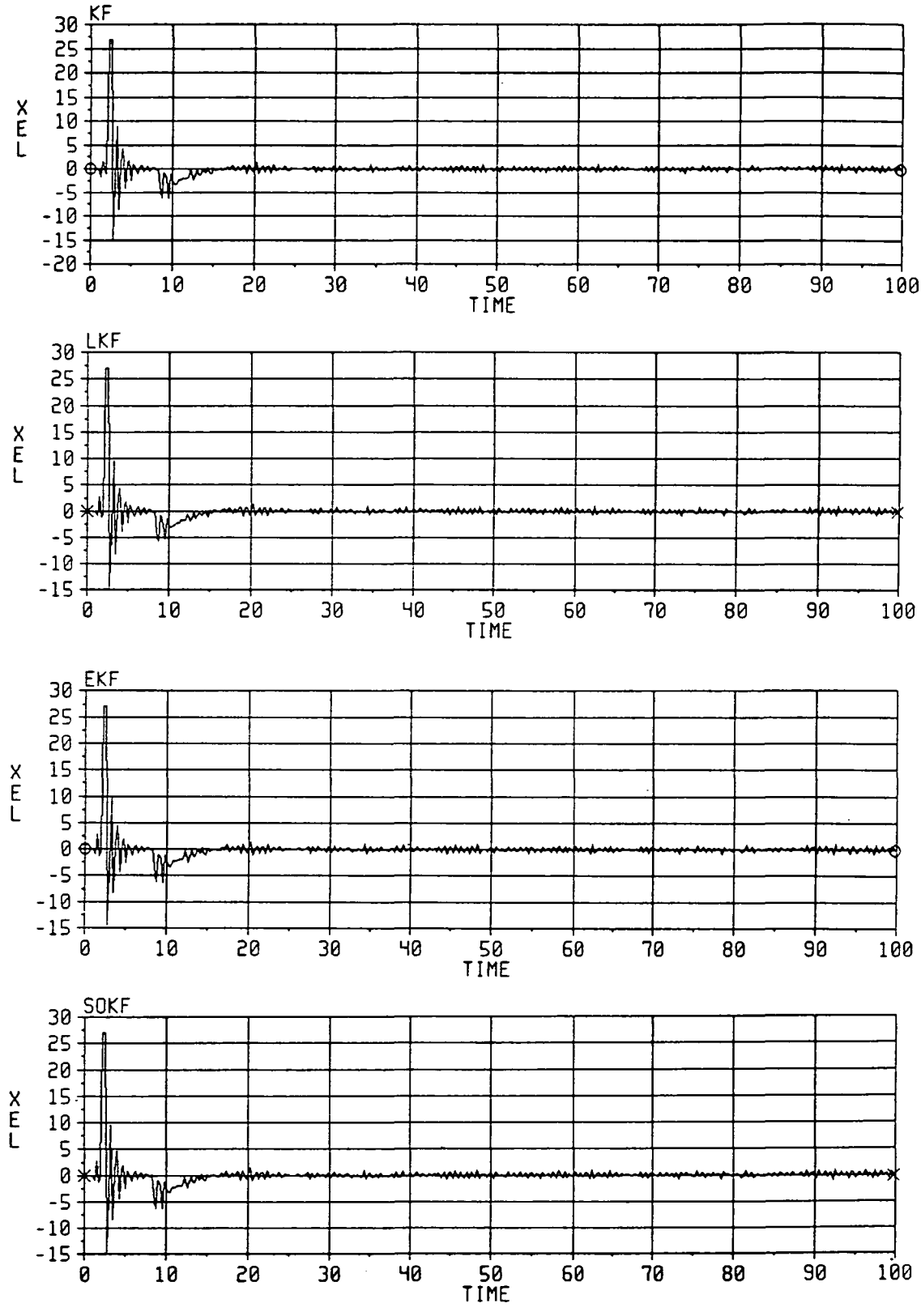


Figure A-3. IPS nominal cross-elevation torque commands.

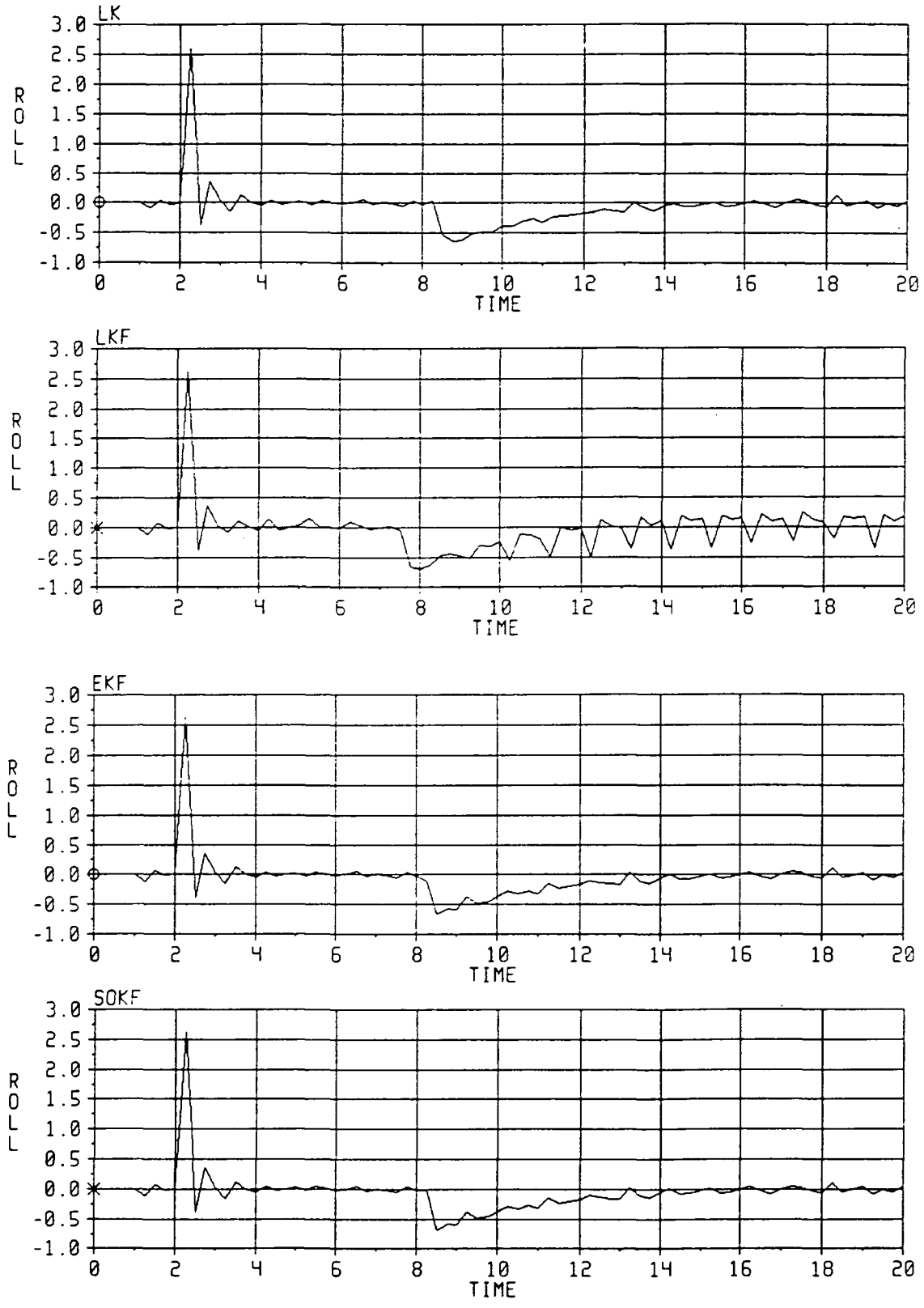


Figure A-4. IPS nominal roll torque commands reduced time scale.

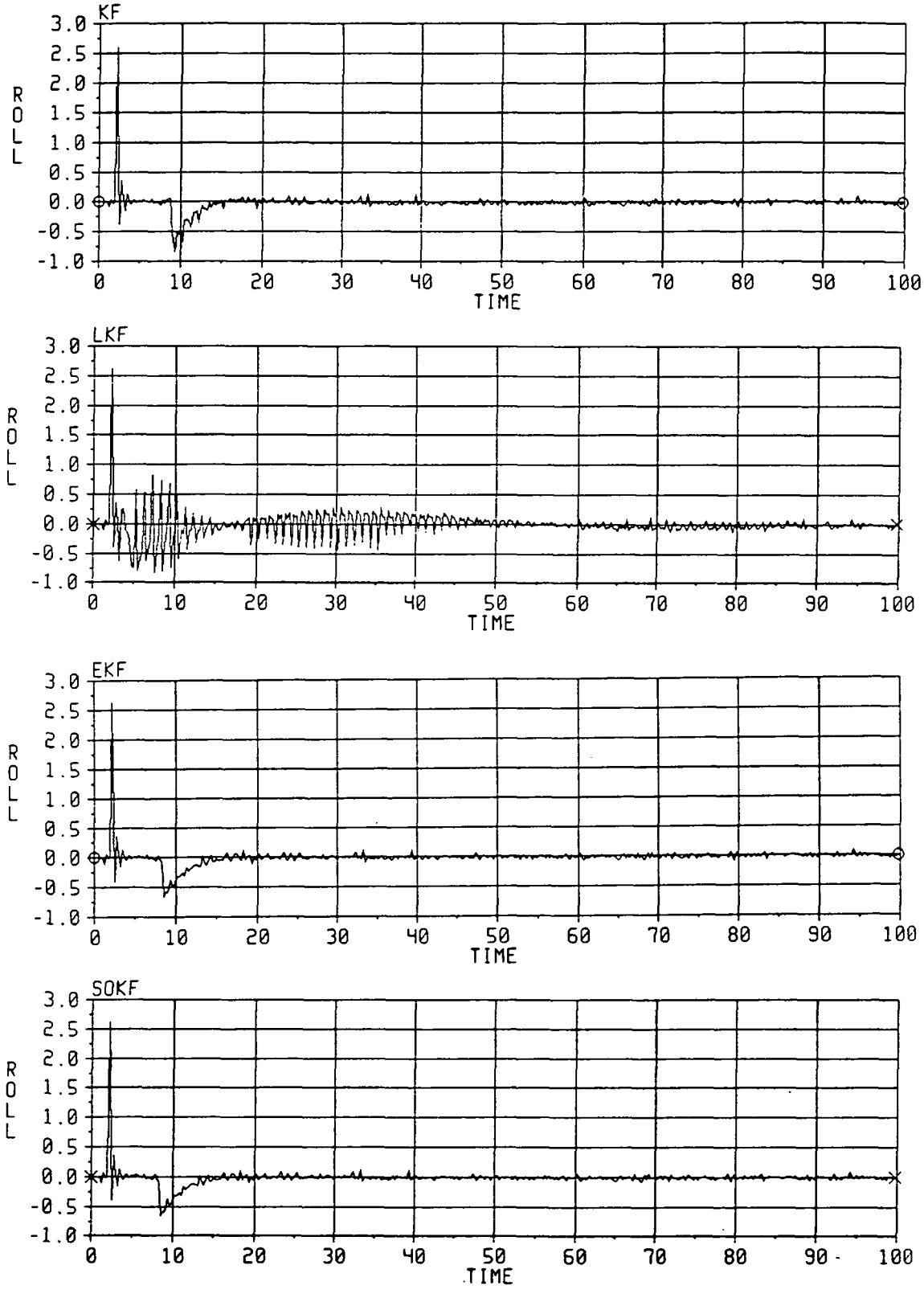


Figure A-5. IPS star loss roll torque commands.

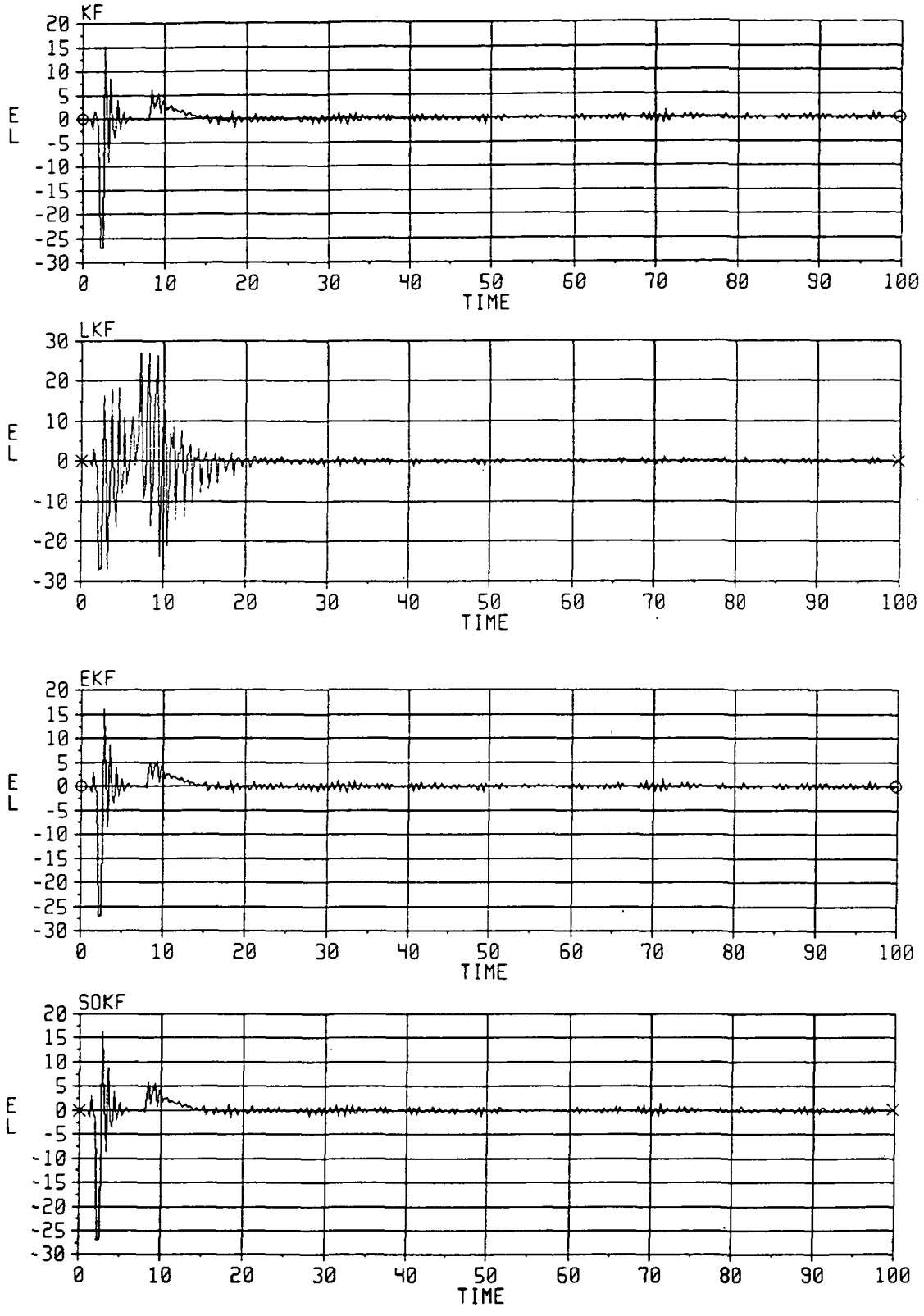


Figure A-6. IPS star loss elevation torque commands.

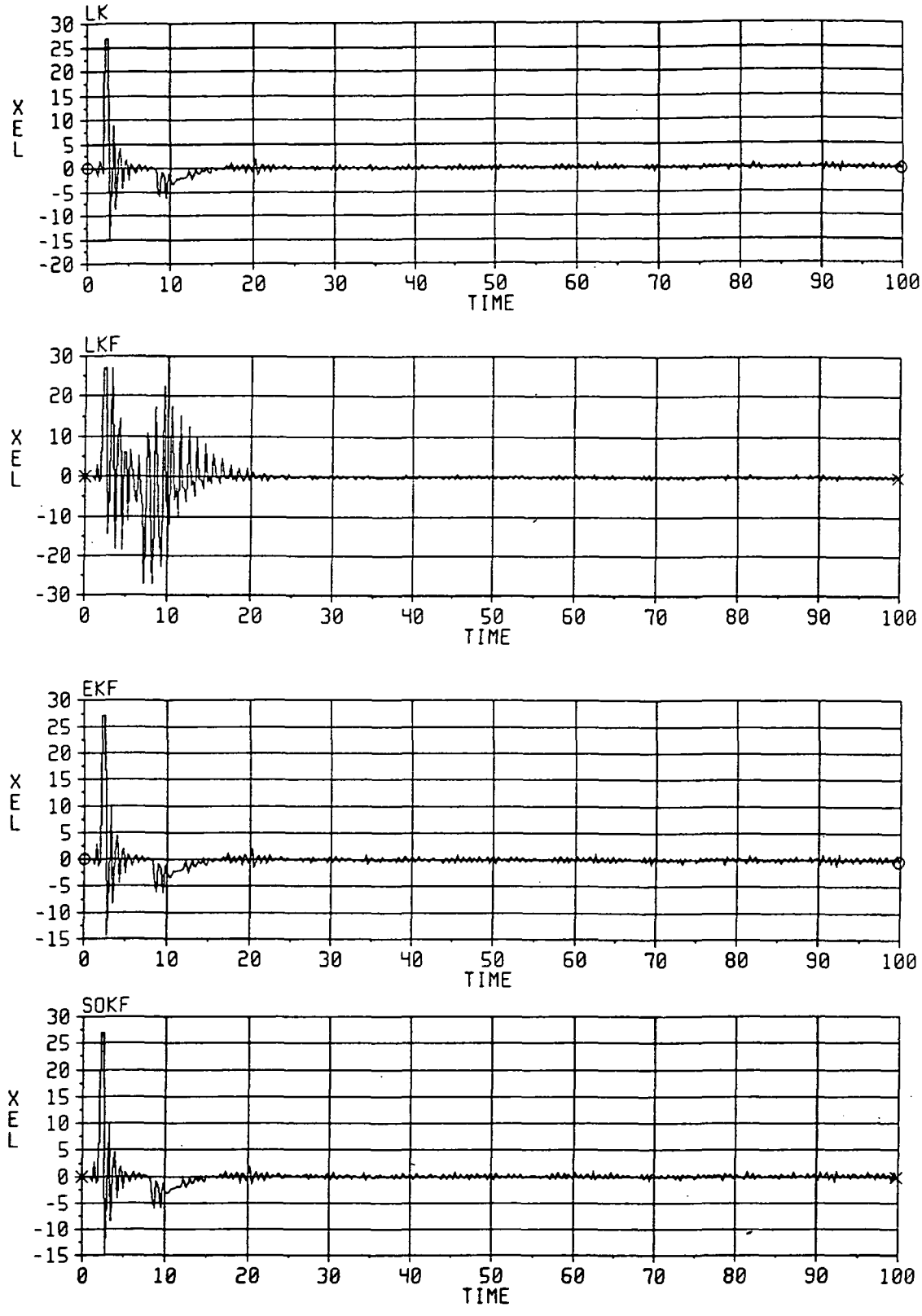


Figure A-7. IPS star loss cross-elevation torque commands.

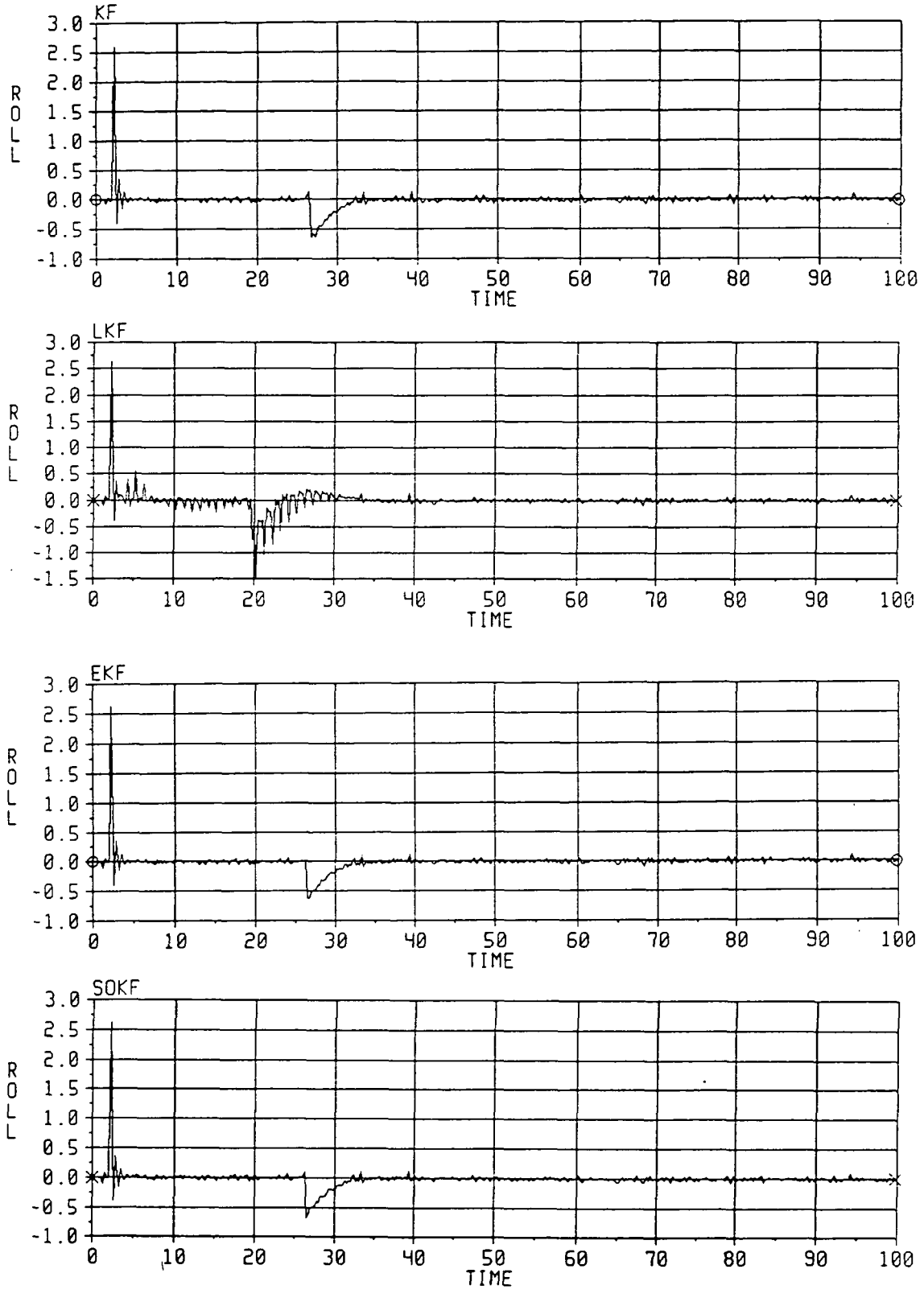


Figure A-8. IPS 1° roll torque commands.

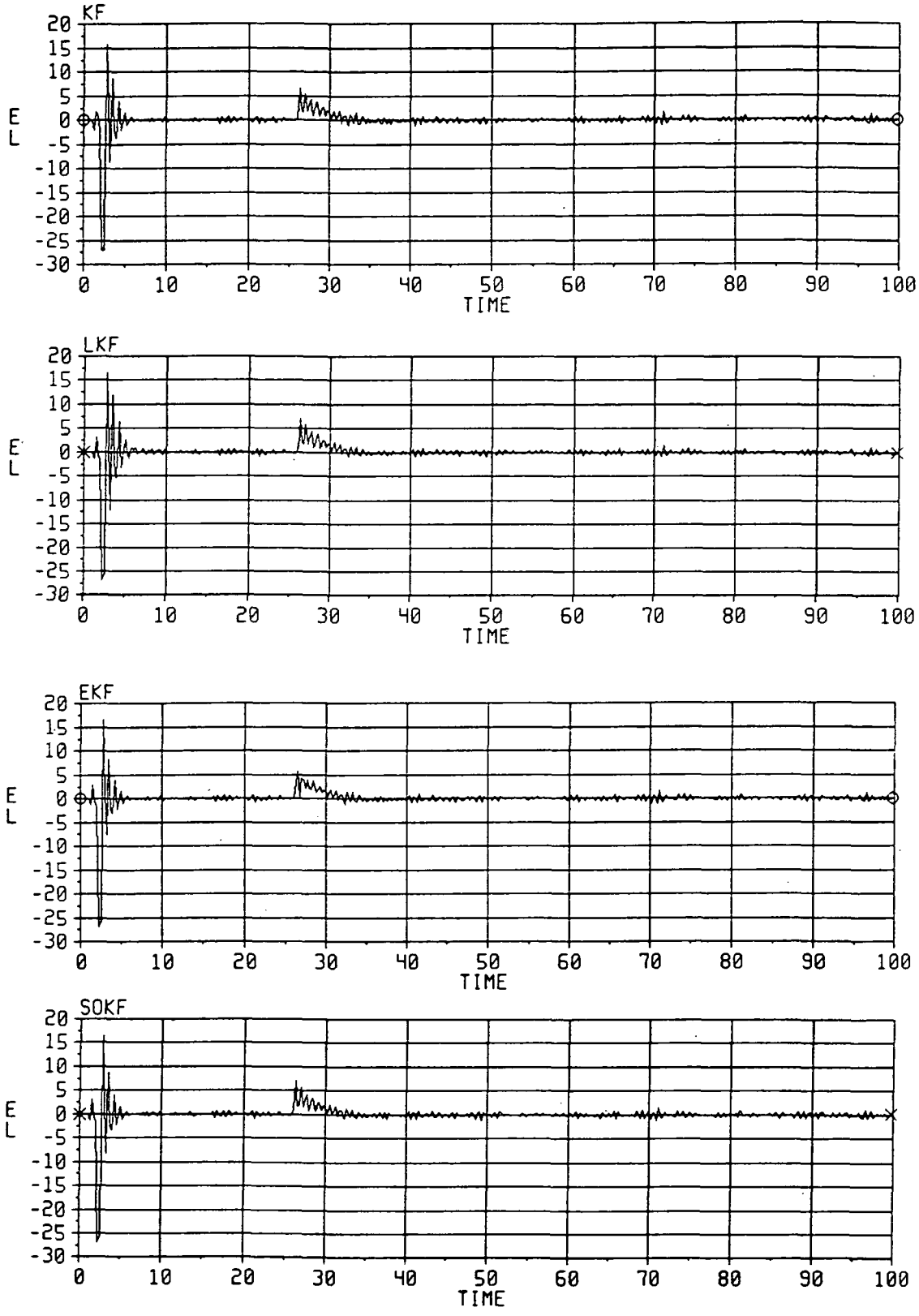


Figure A-9. IPS 1° elevation torque commands.

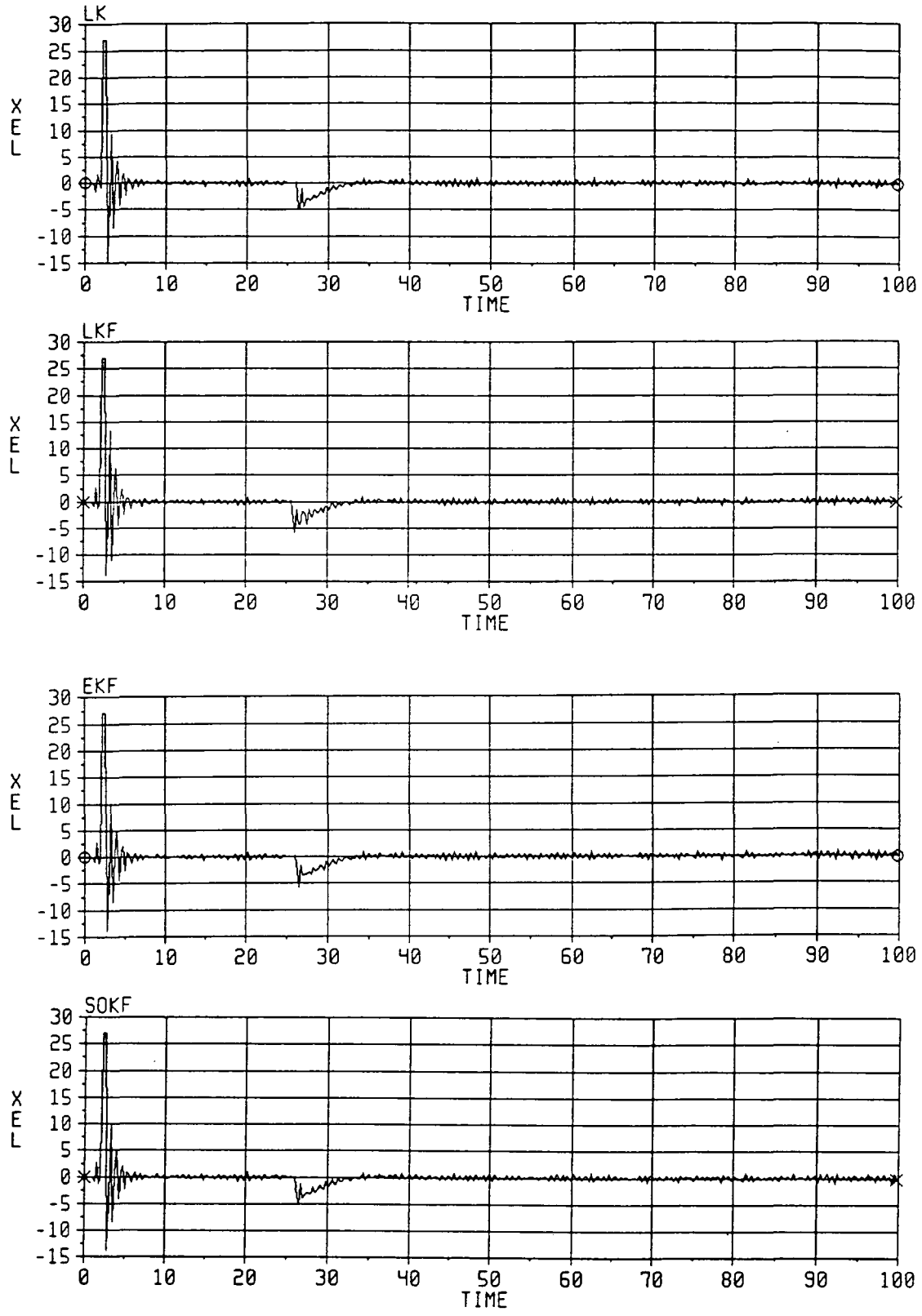


Figure A-10. IPS 1° cross-elevation torque commands.

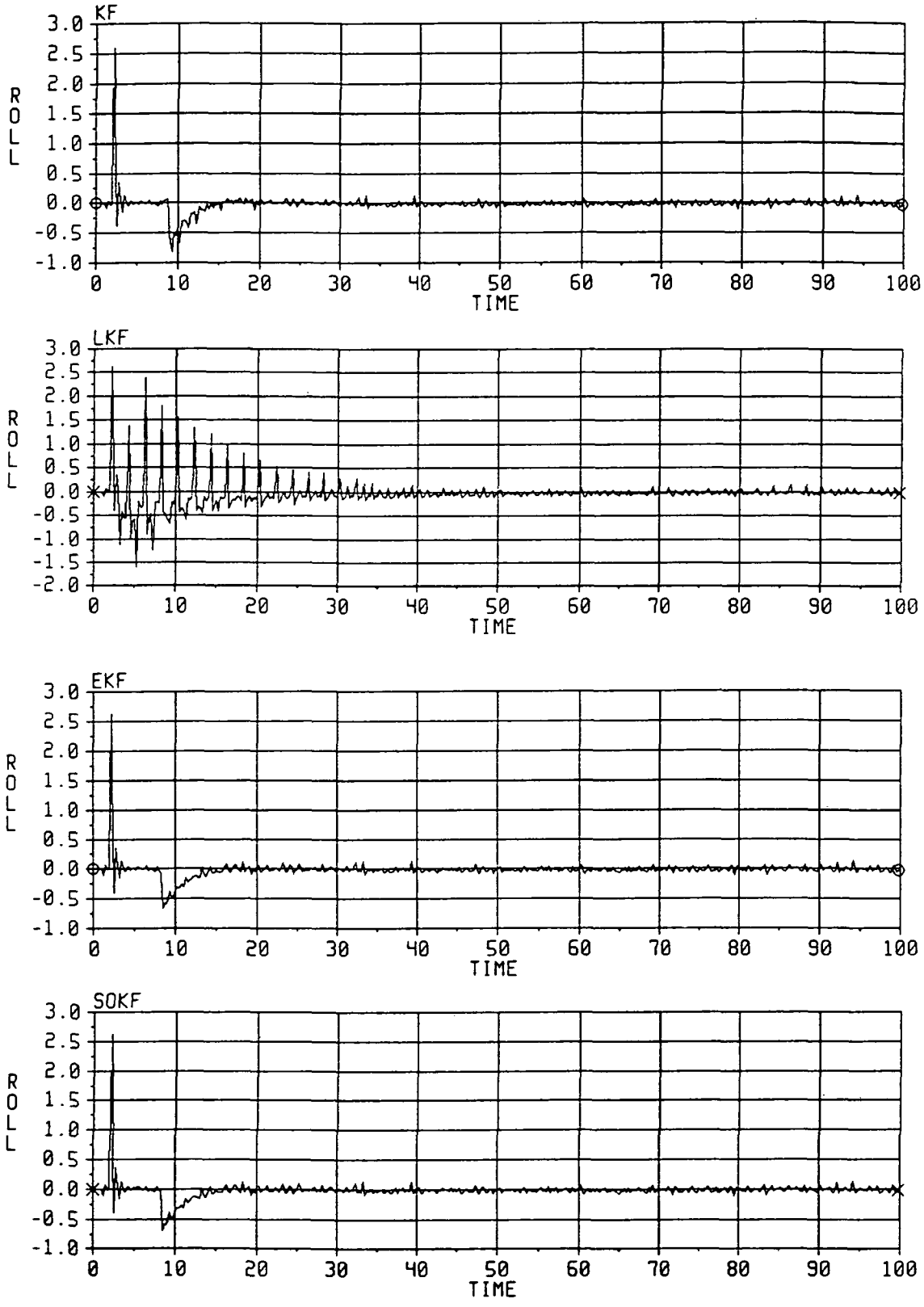


Figure A-11. IPS roll torque commands without drift compensation.

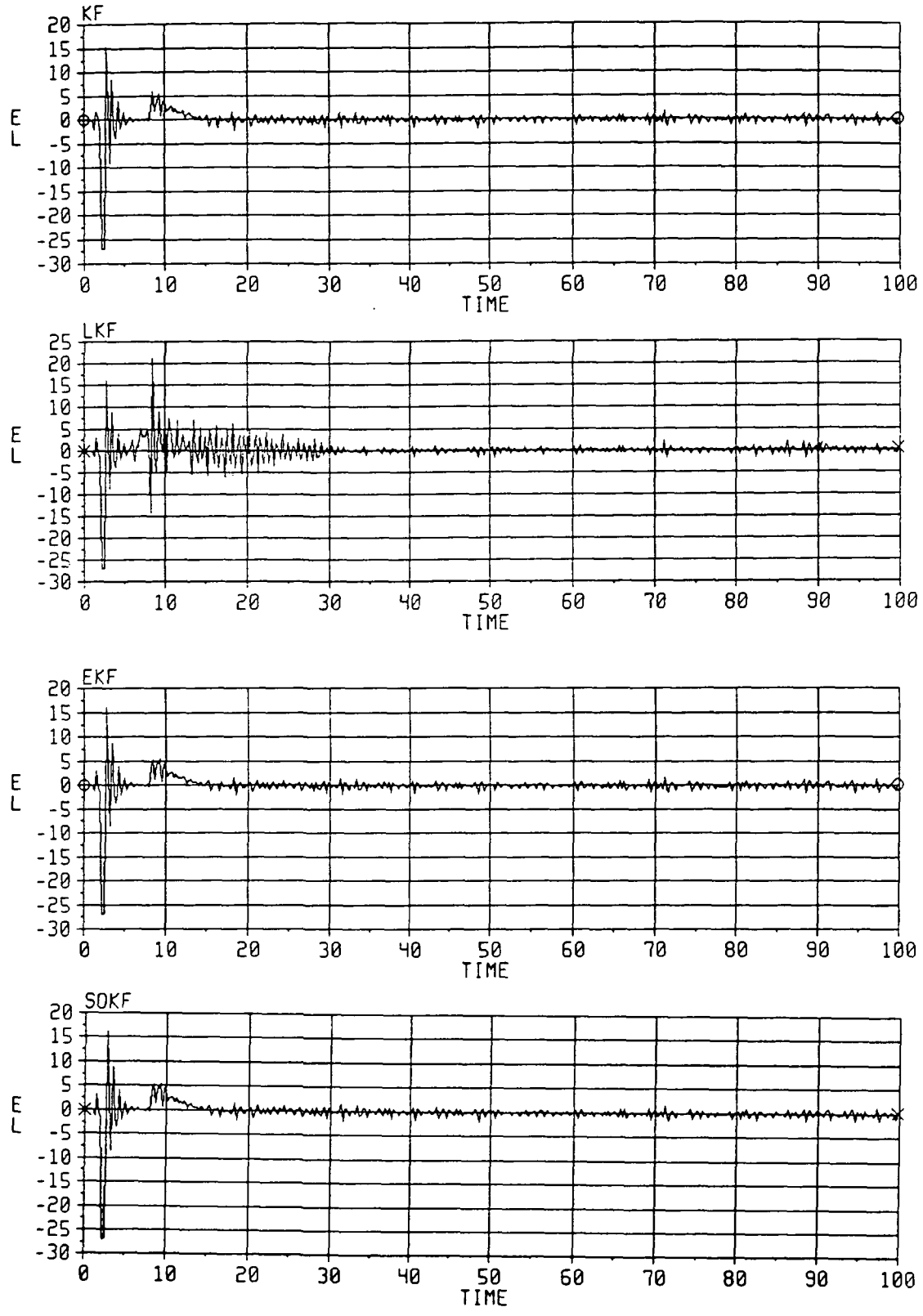


Figure A-12. IPS pitch torque commands without drift compensation.

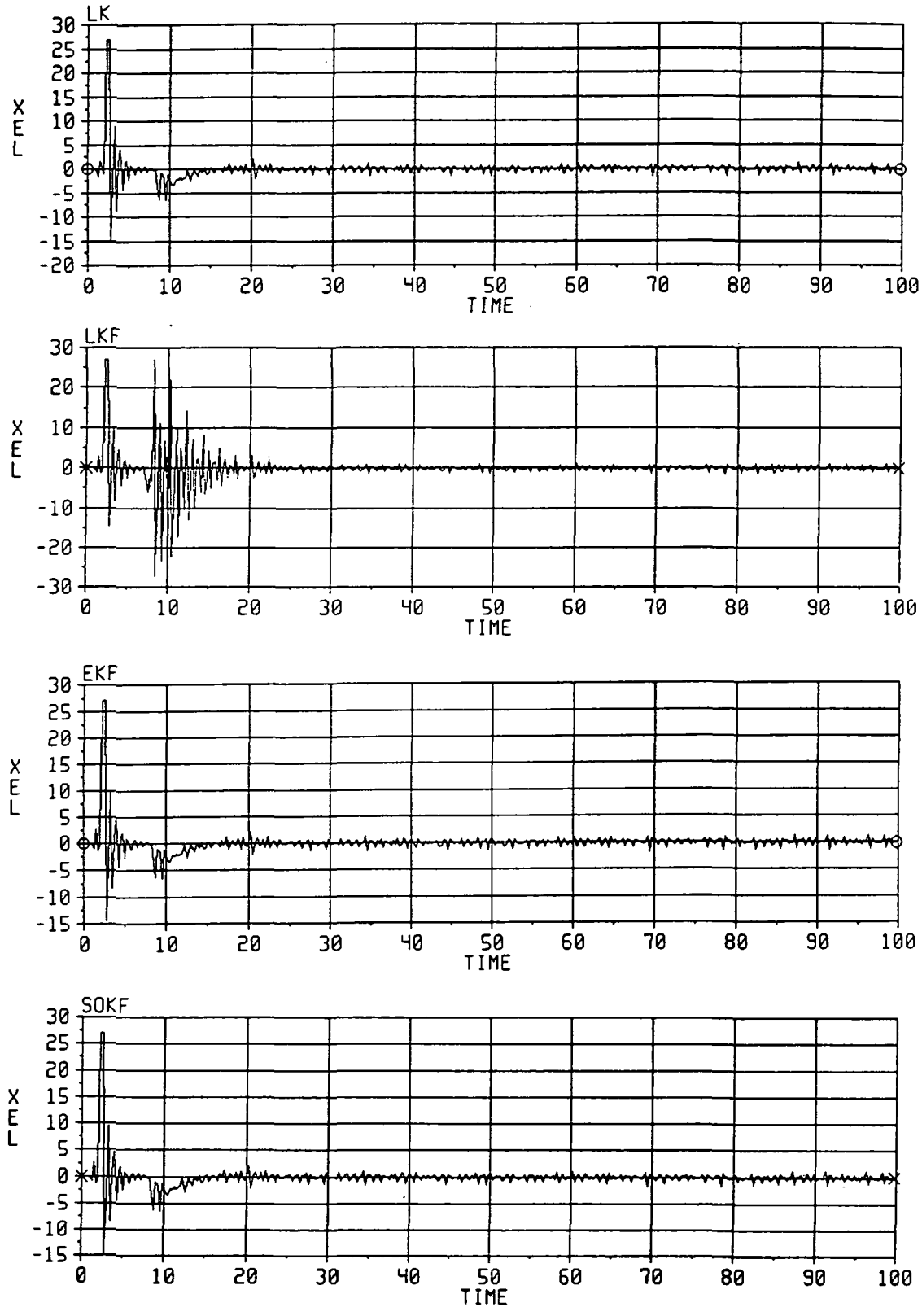


Figure A-13. IPS yaw torque commands without drift compensation.

APPENDIX B

IPS TREETOPS Simulation

USDC.FOR

```
      SUBROUTINE USDC(TIME,U,R)
      COMMON /ITIME/ NDT
      COMMON /SFLAG/ H_FLAG,H_FFDD,FIRST
      COMMON /ANGLE/ ANG(3),DESANG(3)
      COMMON/ADF/DD(3),QD(4),QM(4),QO(4),CM(4),QQ(4),VSTN(6)
      COMMON /ORB/ OMBY(3)
      REAL U(18),BETA(6),R(3),GRATE(4,3),ACCEL(3),RATE(3)
      REAL TDPO(3,3),KACP(3,3),WD(3),KTCP(3,3),GN(3),VGN(3)
      DIMENSION TMODE(10),IMODE(10),DESMOD(10,3)
      LOGICAL SAMP,FIRST,H_FLAG,H_FFDD,FOUR
      DATA FOUR/.TRUE./
      DATA  IGYRO/2/
      IF(NDT.EQ.0) READ(17,*) VGN,VSTN
      IGYRO=IGYRO+1
      IF(IGYRO.GT.4)IGYRO=1
      IF(NDT.EQ.0)NDT=1
      IF(NDT.EQ.4)ISTEP=1
      DO 10 I=1,3
      CALL RNOISE(GN(I),VGN(I))
      GRATE(IGYRO,I)=U(I)+GN(I)
      RATE(I)=GRATE(IGYRO,I)
      IF(MOD(NDT,2).EQ.0)ACCEL(I)=U(3+I)
      ANG(I)=U(6+I)
      OMBY(I)=U(15+I)
   10 CONTINUE
      FIRST=.FALSE.
      IF(TIME.LT..00001)FIRST=.TRUE.
      IF(FIRST)THEN
        CALL RMODE(NM,TMODE,IMODE,DESMOD)
        MODE=0
        DO 3 I=1,3
        DESANG(I)=0.0
        ANG(I)=0.0
        RATE(I)=0.0
        QQ(I)=0.0
    3   CONTINUE
        QQ(4)=1.0
        CALL AEDCM(TDPO)
        CALL DESRAT(WD)
        H_FLAG=.FALSE.
        H_FFDD=.FALSE.
        CALL RSLEW(RATE,ANG,WD,TDPO,DESANG,H_FLAG,H_FFDD,QQ)
      END IF
```

```
C
C
C   EXECUTE MODE-SWITCHING LOGIC
C
C
      IF(TIME.GE.TMODE(MODE+1))THEN
        MODE=MODE+1
        IF(IMODE(MODE).EQ.2.OR.IMODE(MODE).EQ.3)THEN
         DO 789 I=1,3
         DESANG(I)=DESMOD(MODE,I)/57.296
  789    CONTINUE
         IF(IMODE(MODE).EQ.2)H_FLAG=.TRUE.
         IF(IMODE(MODE).EQ.3)H_FFDD=.TRUE.
        ELSE
         IF(IMODE(MODE).EQ.4)THEN
          FOUR=.TRUE.
         ELSE
          FOUR=.FALSE.
         ENDIF
         H_FLAG=.FALSE.
         H_FFDD=.FALSE.
         IF(MOD(NDT,20).EQ.0)CALL AEDCM(TDPO)
         IF(MOD(NDT,20).EQ.0)CALL DESRAT(WD)
        END IF
      END IF
      IF(MOD(NDT,4).EQ.0)BETA(1)=U(10)
      IF(MOD(NDT,4).EQ.0)BETA(2)=U(11)
      IF(MOD(NDT,4).EQ.0)BETA(3)=U(12)
      IF(MOD(NDT,4).EQ.0)BETA(4)=U(13)
      IF(MOD(NDT,4).EQ.0)BETA(5)=U(14)
      IF(MOD(NDT,4).EQ.0)BETA(6)=U(15)
      IF(MOD(NDT,100).EQ.0)CALL ACP(ANG,KACP)
      IF(MOD(NDT,100).EQ.0)CALL TCP(ANG,KTCP)
      IF((H_FLAG.OR.H_FFDD).AND.(MOD(NDT,20).EQ.0))CALL
     $RSLEW(RATE,ANG,WD,TDPO,DESANG,H_FLAG,H_FFDD,QQ,FOUR)
      IF((IGYRO.EQ.2).OR.(IGYRO.EQ.4))
     $CALL DCU(GRATE,ACCEL,R,TDPO,WD,KTCP,KACP,ISTEP,TIME,BETA,
     $IGYRO,FOUR)
      IF(MOD(NDT,4).EQ.0)ISTEP=ISTEP+1
      NDT=NDT+1
      RETURN
      END
      FUNCTION SAMP(T,DT)
      LOGICAL SAMP
      A=T/DT
      B=ANINT(A)
      IF(ABS(A-B).LT.1.0E-5)SAMP=.TRUE.
      IF(ABS(A-B).GE.1.0E-5)SAMP=.FALSE.
      RETURN
      END
      SUBROUTINE DCM(A,B,G,H)
      DIMENSION H(3,3)
      REAL*8 SA,CA,SB,CB,SG,CG
```

```
      SA=DSIN(DBLE(A))
      CA=DCOS(DBLE(A))
      SB=DSIN(DBLE(B))
      CB=DCOS(DBLE(B))
      SG=DSIN(DBLE(G))
      CG=DCOS(DBLE(G))
      H(1,1)=SNGL(CB*CG)
      H(1,2)=SNGL(SG)
      H(1,3)=SNGL(-SB*CG)
      H(2,1)=SNGL(-CA*CB*SG+SA*SB)
      H(2,2)=SNGL(CG*CA)
      H(2,3)=SNGL(CA*SB*SG+SA*CB)
      H(3,1)=SNGL(SA*CB*SG+CA*SB)
      H(3,2)=SNGL(-SA*CG)
      H(3,3)=SNGL(-SA*SB*SG+CA*CB)
      RETURN
      END
      SUBROUTINE MULMAT(A,B,C,L,M,N)
      DIMENSION A(L,M),B(M,N),C(L,N)
      DO 1 I=1,L
      DO 1 J=1,N
      SAVE = 0.0
      DO 2 K=1,M
 2    SAVE = SAVE + A(I,K)*B(K,J)
 1    C(I,J) = SAVE
      RETURN
      END
      SUBROUTINE TRMAT(A,B,M,N)
      DIMENSION A(M,N),B(N,M)
      DO 1 I=1,N
      DO 1 J=1,M
 1    B(I,J) = A(J,I)
      RETURN
      END
      SUBROUTINE ADDMAT(A,B,C,M,N)
      DIMENSION A(M,N),B(M,N),C(M,N)
      DO 1 I=1,M
      DO 1 J=1,N
 1    C(I,J) = A(I,J) + B(I,J)
      RETURN
      END
      SUBROUTINE DCU(GRATE,ACCEL,CU,TDPO,WD,KTCP,KACP,I,TIME,BETA
     $,IGYRO,FOUR)
C...THIS PROGRAM IS THE DRIVER UNIT FOR THE FORTRAN DCU CONTROLLER
C...SERVING TO PROMPT INPUTS OF SENSOR READINGS, CONTROL FLOW OF DATA, A
C...COMPUTE CONTROL TORQUE COMMANDS
      REAL MAG(6),GRATE(4,3),ACCEL(3),PID(3),YFF(3),CU(3))
      REAL KTCP(3,3),KACP(3,3),BETA(6)
      REAL TDPO(3,3),QOM(4,4),WD(3),YS(6),SIGN(3),STN(6)
      LOGICAL SAMP,H_FLAG,H_FFDD,NOADF,FOUR
      COMMON /ITIME/ NDT
      COMMON /SFLAG/ H_FLAG,H_FFDD,FIRST
      COMMON /ANGLE/ ANG(3),DESANG(3)
```

```
      COMMON /ADF/ DD(3),QD(4),QM(4),Q01(4),CM1(4),Q(4),VSTN(6)
      COMMON /SUM/ QS(4),DRIFT(3),PE(3)
      COMMON /CFHST/ YRO(6,3),YMS(6,2),DUMMY(43)
      COMMON /OBS/ TH(6),PH(3,6),AT(6,6),P(6,6),HP(6,6)

      DATA Q01,Q/3*0.0,1.,3*0.0,1./

      DATA TMAX/27.0/
      IF(.NOT.H_FLAG)THEN
        CALL AEDCM(TDPO)
        CALL DESRAT(WD)
      ENDIF
      CALL FGYRO(GRATE,I,PID,WD,TDPO,IGYRO)
      WRITE(6,*)TIME
      CALL FACCEL(ACCEL,I,YFF,KACP,IGYRO)
C...COMPUTE CONTROL OUTPUT TORQUES, CU
      IF(IGYRO.EQ.4)THEN
      CALL MMUL(KTCP,PID,CU,3,3,1)
      C     CALL MADD(CU,YFF,CU,3,1)
      C     CALL SMUL(CU,-1.0,CU,3,1)
      ENDIF
C
C   TORQUE LIMITATION ADDED BY JMR
C
      DO 1111 IJ=1,3
      IF(CU(IJ).LT.0.0)SIGN(IJ)=-1.0
      IF(CU(IJ).GE.0.0)SIGN(IJ)=1.0
      IF(ABS(CU(IJ)).GT.TMAX)CU(IJ)=TMAX*SIGN(IJ)
 1111 CONTINUE
C
C
      DO 887 I=1,6
 887  CALL RNOISE(STN(I),VSTN(I))
      YS(1)=BETA(1)+STN(1)
      YS(2)=BETA(2)+STN(2)
      YS(3)=BETA(3)+STN(3)
      YS(4)=BETA(4)+STN(4)
      YS(5)=BETA(5)+STN(5)
      YS(6)=BETA(6)+STN(6)
      YMS(1,1)=YS(1)
      YMS(1,2)=YS(2)
      YMS(2,1)=YS(3)
      YMS(2,2)=YS(4)
      YMS(3,1)=YS(5)
      YMS(3,2)=YS(6)
      IF((H_FLAG.OR.H_FFDD).OR.FOUR)THEN
        NOADF=.TRUE.
      ELSE
        NOADF=.FALSE.
      ENDIF
      IF(MOD(NDT,100).NE.0.OR.NOADF)GO TO 747
      WRITE(6,3332)
 3332 FORMAT(1X,' ADF IS ON')
```

```
      DO 15 K=1,4
   15 QOM(1,K)=Q01(K)
      QOM(2,1)=Q01(4)
      QOM(2,2)=Q01(3)
      QOM(2,3)=-Q01(2)
      QOM(2,4)=-Q01(1)
      QOM(3,1)=-Q01(3)
      QOM(3,3)=Q01(1)
      QOM(3,4)=-Q01(2)
      QOM(4,1)=Q01(2)
      QOM(4,2)=-Q01(1)
      QOM(4,3)=Q01(4)
      QOM(4,4)=-Q01(3)
C
C-----------------------------------------------------------------------
C
C USE OF CM1 CHANGED TO CM BY ADAMS & WEST TO AVOID DELAYS
C
      CALL MULMAT(QOM,Q,CM1,4,4,1)
C
      DO 745 I=1,4
      QM(I)=QS(I)/25.
  745 QS(I)=0.0
      T1=SECNDS(0.0)
      CALL LADF
      DELTA=SECNDS(T1)
      TADF=TADF+DELTA
      DO 774 I=1,3
  774 Q(I)=Q(I)+QD(I)
      Q(4)=SQRT(1.-Q(1)**2-Q(2)**2-Q(3)**2)
      IF(NDT.GT.100)WRITE(18) TIME,Q,YMS,DD,(P(I,I),I=1,6),QD,TADF,DELTA
      DO 777 I=1,4
  777 Q01(I)=Q(I)
      DO 888 KK=1,6
      DO 888 JJ=1,2
  888 YMS(KK,JJ)=0.0
  747 CONTINUE

      RETURN
      END
      SUBROUTINE AEDCM(DCM)
      REAL DCM(3,3)
      DO 20 I=1,3
      DO 10 J=1,3
      DCM(I,J)=0.0
   10 CONTINUE
      DCM(I,I)=1.0
   20 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE DESRAT(WD)
```

```
      REAL WD(3)
      DO 10 I=1,3
      WD(I)=0.0
   10 CONTINUE
      RETURN
      END

      SUBROUTINE TCP(ANG,KTCP)
C... THIS PROGRAM COMPUTES THE TORQUE DECOUPLING MATRIX, KTCP, WHERE
C... ANG(3) IS THE VECTOR OF ROLL, ELEVATION, AND CROSS ELEVATION ANGLES
        REAL ANG(3),KTCP(3,3),TMAT(3,3),TPPPT(3,3)
        REAL TPPCT(3,3),TETA(3,3)
        DATA ALP/0.0/

C... READ TETA MATRIX FROM DATA FILE
      O       OPEN(31,FILE='TETA.DAT',STATUS='OLD')
      R       READ (31,*) ((TETA(I,J),J=1,3),I=1,3)
      C       CLOSE(31)

C... COMPUTE TRANSPOSE OF C TO PPRIME TRANSFORMATION MATRIX
      C      CR=COS(ANG(1))
      S      SR=SIN(ANG(1))
       CE=COS(ANG(2))
            SE=SIN(ANG(2))
      C     CX=COS(ANG(3))
      S     SX=SIN(ANG(3))

      T     TPPCT(1,1)=1.0
      T     TPPCT(1,2)=0.0
      T     TPPCT(1,3)=0.0
      T     TPPCT(2,1)=SX
      T     TPPCT(2,2)=CR*CX
      T     TPPCT(2,3)=-SR*CX
      T     TPPCT(3,1)=0.0
      T     TPPCT(3,2)=SR
      T     TPPCT(3,3)=CR
      C     CALL MMUL(TPPCT,TETA,TMAT,3,3,3)

C**  COMPUTE TRANSPOSE OF PPRIME TO P TRANSFORMATION
C
C
      TPPPT(1,1)=1.0
      TPPPT(1,2)=0.0
      TPPPT(1,3)=0.0
      TPPPT(2,1)=0.0
      TPPPT(2,2)=COS(ALP)
      TPPPT(2,3)=SIN(ALP)
      TPPPT(3,1)=0.0
      TPPPT(3,2)=-SIN(ALP)
      TPPPT(3,3)=COS(ALP)
C
C
      CALL MMUL(TMAT,TPPPT,KTCP,3,3,3)
```

```
       RETURN
       END
      SUBROUTINE ACP(ANG,KACP)
C... THIS PROGRAM COMPUTES THE ACP GAIN MATRIX, KACP, WHICH IS A FUNCTIO
C... OF THE ROLL, ELEVATION, AND CROSS ELEVATION ANGLES AND THE CG POSIT
C... OF THE IPS ROLL DRIVE UNIT AND PAYLOAD (STORED IN "IPS.DAT")

      REAL MASS,ANG(3)
      REAL POS(3),KACP(3,3),DSMP(3),AUX(3,3)
      REAL V1(3),V2(3),TT1(3,3),TT2(3,3),VEC(3),VTIL(3,3)

      OPEN (31,FILE='IPS.DAT',STATUS='OLD')
      READ(31,*) MASS,(POS(I),I=1,3)
      CLOSE(31)

      DSMP(1)=-MASS*POS(1)
      DSMP(2)=MASS*POS(2)
      DSMP(3)=-MASS*POS(3)

      ROLL=ANG(1)
      EL=ANG(2)
      XEL=ANG(3)

      DO 100 I=1,3
      IF (I.EQ.1) THEN
         V1(1)=0.0
         V1(2)=0.0
         V1(3)=0.0
         V2(1)=ROLL
         V2(2)=EL
         V2(3)=XEL
      ENDIF
      IF (I.EQ.2) THEN
         V1(1)=ROLL
         V1(2)=0.0
         V1(3)=XEL
         V2(1)=0.0
         V2(2)=EL
         V2(3)=0.0
      ENDIF
      IF (I.EQ.3) THEN
         V1(1)=ROLL
         V1(2)=0.0
         V1(3)=0.0
         V2(1)=0.0
         V2(2)=EL
         V2(3)=XEL
      ENDIF
      CALL TPPST(V1,TT1)
      CALL TPPST(V2,TT2)
      CALL MMUL(DSMP,TT1,VEC,1,3,3)
      CALL TILDE(VEC,VTIL)
      CALL SMUL(VTIL,-1.0,VTIL,3,3)
```

```
      CALL MMUL(VTIL,TT2,TT1,3,3,3)
      DO 50 J=1,3
50       AUX(I,J)=TT1(I,J)
100   CONTINUE
      CALL SMUL(TT2,0.0,TT2,3,3)
      TT2(1,2)=1.0
      TT2(2,3)=-1.0
      TT2(3,1)=-1.0
      CALL MMUL(AUX,TT2,KACP,3,3,3)

      RETURN
      END

      SUBROUTINE TILDE(A,B)
      REAL A(3),B(3,3)
C...FORMS THE SKEW SYMMETRIC MATRIX B OF VECTOR A
      B(1,1)=0.0
      B(1,2)=A(3)
      B(1,3)=-A(2)
      B(2,1)=-A(3)
      B(2,2)=0.0
      B(2,3)=A(1)
      B(3,1)=A(2)
      B(3,2)=-A(1)
      B(3,3)=0.0
      RETURN
      END

      SUBROUTINE TPPST(A,TT)
      REAL A(3),TT(3,3)
C...COMPUTES THE TRANSFORMATION MATRIX FROM ST TO PPRIME SYSTEM
      CR=COS(A(1))
      SR=SIN(A(1))
      CE=COS(A(2))
      SE=SIN(A(2))
      CX=COS(A(3))
      SX=SIN(A(3))

      TT(1,1)=CX*CE
      TT(1,2)=SX
      TT(1,3)=-CX*SE
      TT(2,1)=-CR*SX*CE+SR*SE
      TT(2,2)=CR*CX
      TT(2,3)=CR*SX*SE+SR*CE
      TT(3,1)=SR*SX*CE+CR*SE
      TT(3,2)=-SR*CX
      TT(3,3)=-SR*SX*SE+CR*CE

      RETURN
      END
      SUBROUTINE MMUL(A,B,C,L,M,N)
C...COMPUTES THE MATRIX PRODUCT C = A B
      REAL A(L,M),B(M,N),C(L,N)
```

```
      DO 100 I=1,L
      DO 100 J=1,N
         C(I,J)=0.0
         DO 100 K=1,M
         C(I,J)=C(I,J)+A(I,K)*B(K,J)
100   CONTINUE
      RETURN
      END
      SUBROUTINE MADD(A,B,C,M,N)
C...COMPUTES THE MATRIX SUM C = A + B
      REAL A(M,N),B(M,N),C(M,N)
      DO 100 I=1,M
      DO 100 J=1,N
         C(I,J)=A(I,J)+B(I,J)
100   CONTINUE
      RETURN
      END

      SUBROUTINE SMUL(A,SCALAR,C,M,N)
C...COMPUTES THE MATRIX PRODUCT C = A B
      REAL A(M,N),C(M,N)
      DO 100 I=1,M
      DO 100 J=1,N
         C(I,J)=SCALAR*A(I,J)
100   CONTINUE
      RETURN
      END

      SUBROUTINE FFDATA
      REAL APFA(3,2,2),BPFA(3,2),CPFA(3,2),DPFA(3)
      REAL AFF1(3,2,2),BFF1(3,2),CFF1(3,2),DFF1(3)
      REAL AFF2(3,2,2),BFF2(3,2),CFF2(3,2),DFF2(3)
      REAL AFF3(3,2,2),BFF3(3,2),CFF3(3,2),DFF3(3)
      COMMON /FFCOEF/ APFA,BPFA,CPFA,DPFA,AFF1,BFF1,CFF1,DFF1,
     $                AFF2,BFF2,CFF2,DFF2,AFF3,BFF3,CFF3,DFF3
      DATA XNORM/32767./

      OPEN(31,FILE='FFCOEF.DAT',STATUS='OLD')
C
C
C   READ FEEDFORWARD LOOP COEFFICIENTS IN STATE-SPACE
C   FORM FROM MDP BOOK
C
C
      DO 10 I=1,3
      READ(31,*)APFA(I,1,1),APFA(I,1,2),APFA(I,2,1),APFA(I,2,2)
      READ(31,*)BPFA(I,1),BPFA(I,2),CPFA(I,1),CPFA(I,2),DPFA(I)
   10 CONTINUE
      DO 20 I=1,3
      READ(31,*)AFF1(I,1,1),AFF1(I,1,2),AFF1(I,2,1),AFF1(I,2,2)
      READ(31,*)BFF1(I,1),BFF1(I,2),CFF1(I,1),CFF1(I,2),DFF1(I)
   20 CONTINUE
      DO 30 I=1,3
```

```
      READ(31,*)AFF2(I,1,1),AFF2(I,1,2),AFF2(I,2,1),AFF2(I,2,2)
      READ(31,*)BFF2(I,1),BFF2(I,2),CFF2(I,1),CFF2(I,2),DFF2(I)
   30 CONTINUE
      DO 40 I=1,3
      READ(31,*)AFF3(I,1,1),AFF3(I,1,2),AFF3(I,2,1),AFF3(I,2,2)
      READ(31,*)BFF3(I,1),BFF3(I,2),CFF3(I,1),CFF3(I,2),DFF3(I)
   40 CONTINUE
      CLOSE(31)
      DO 400 I=1,3
      DPFA(I)=DPFA(I)/XNORM
      DFF1(I)=DFF1(I)/XNORM
      DFF2(I)=DFF2(I)/XNORM
      DFF3(I)=DFF3(I)/XNORM
      DO 400 J=1,2
      BPFA(I,J)=BPFA(I,J)/XNORM
      BFF1(I,J)=BFF1(I,J)/XNORM
      BFF2(I,J)=BFF2(I,J)/XNORM
      BFF3(I,J)=BFF3(I,J)/XNORM
      CPFA(I,J)=CPFA(I,J)/XNORM
      CFF1(I,J)=CFF1(I,J)/XNORM
      CFF2(I,J)=CFF2(I,J)/XNORM
      CFF3(I,J)=CFF3(I,J)/XNORM
      DO 400 K=1,2
      APFA(I,J,K)=APFA(I,J,K)/XNORM
      AFF1(I,J,K)=AFF1(I,J,K)/XNORM
      AFF2(I,J,K)=AFF2(I,J,K)/XNORM
      AFF3(I,J,K)=AFF3(I,J,K)/XNORM
  400 CONTINUE
      RETURN
      END

      SUBROUTINE FBDATA

      REAL APF(3,2,2),BPF(3,2),CPF(3,2),DPF(3)
      REAL AFB1(2,2,2),BFB1(2,2),CFB1(2,2),DFB1(2)
      REAL AFB2(2,2,2),BFB2(2,2),CFB2(2,2),DFB2(2)
      REAL AFB3(2,2,2),BFB3(2,2),CFB3(2,2),DFB3(2)
      REAL AR(2),BR(2),CR(2),DR(2),SPSC(3),VC(3)
      REAL HKD(3),HKP(3),HKI(3),LKD(3),LKP(3),LKI(3)

      COMMON /FBCOEF/ APF,BPF,CPF,DPF,AFB1,BFB1,CFB1,DFB1,
     $                AFB2,BFB2,CFB2,DFB2,AFB3,BFB3,CFB3,DFB3,
     $                AR,BR,CR,DR,HKD,HKP,HKI,LKD,LKP,LKI,SPSC,VC

      DATA XNORM/32767./
      OPEN(31,FILE='FBCOEF.DAT',STATUS='OLD')

C
C
C   READ FILTER COEFFICIENTS IN STATE-SPACE FORM
C   FROM MDP BOOK
CC
C
```

```
C   READ COEFFICIENTS FOR RATE PREFILTER
C
C
      DO 10 I=1,3
      READ(31,*)APF(I,1,1),APF(I,1,2),APF(I,2,1),APF(I,2,2)
      READ(31,*)BPF(I,1),BPF(I,2),CPF(I,1),CPF(I,2),DPF(I)
   10 CONTINUE
C
C
C   READ COEFFICIENTS FOR 2ND ORDER RATE FILTERS
C   Y AND Z AXES ONLY
C
      DO 20 I=1,2
      READ(31,*)AFB1(I,1,1),AFB1(I,1,2),AFB1(I,2,1),AFB1(I,2,2)
      READ(31,*)BFB1(I,1),BFB1(I,2),CFB1(I,1),CFB1(I,2),DFB1(I)
   20 CONTINUE
      DO 30 I=1,2
      READ(31,*)AFB2(I,1,1),AFB2(I,1,2),AFB2(I,2,1),AFB2(I,2,2)
      READ(31,*)BFB2(I,1),BFB2(I,2),CFB2(I,1),CFB2(I,2),DFB2(I)
   30 CONTINUE
      DO 40 I=1,2
      READ(31,*)AFB3(I,1,1),AFB3(I,1,2),AFB3(I,2,1),AFB3(I,2,2)
      READ(31,*)BFB3(I,1),BFB3(I,2),CFB3(I,1),CFB3(I,2),DFB3(I)
   40 CONTINUE
C
C   READ COEFFICIENTS FOR 1ST ORDER RATE FILTER
C   Y AND Z AXES ONLY
C
C
      DO 50 I=1,2
      READ(31,*)AR(I),BR(I),CR(I),DR(I)
   50 CONTINUE
C
C
C   READ PID LOW GAINS (SLEWING)
C
      DO 300 I=1,3
      READ(31,*)LKD(I),LKP(I),LKI(I)
  300 CONTINUE
C
C   READ PID HIGH GAINS (FINE POINTING)
C
      DO 301 I=1,3
      READ(31,*)HKD(I),HKP(I),HKI(I)
  301 CONTINUE
C
C
      READ(31,*)VC(1),VC(2),VC(3)
      READ(31,*)SPSC(1),SPSC(2),SPSC(3)
      CLOSE(31)
      DO 400 I=1,2
      DPF(I)=DPF(I)/XNORM
      DFB1(I)=DFB1(I)/XNORM
```

```
      DFB2(I)=DFB2(I)/XNORM
      DFB3(I)=DFB3(I)/XNORM
      AR(I)=AR(I)/XNORM
      BR(I)=BR(I)/XNORM
      CR(I)=CR(I)/XNORM
      DR(I)=DR(I)/XNORM
      DO 400 J=1,2
      CPF(I,J)=CPF(I,J)/XNORM
      CFB1(I,J)=CFB1(I,J)/XNORM
      CFB2(I,J)=CFB2(I,J)/XNORM
      CFB3(I,J)=CFB3(I,J)/XNORM
      BPF(I,J)=BPF(I,J)/XNORM
      BFB1(I,J)=BFB1(I,J)/XNORM
      BFB2(I,J)=BFB2(I,J)/XNORM
      BFB3(I,J)=BFB3(I,J)/XNORM
      DO 400 K=1,2
      APF(I,J,K)=APF(I,J,K)/XNORM
      AFB1(I,J,K)=AFB1(I,J,K)/XNORM
      AFB2(I,J,K)=AFB2(I,J,K)/XNORM
      AFB3(I,J,K)=AFB3(I,J,K)/XNORM
  400 CONTINUE
      DPF(3)=DPF(3)/XNORM
      DO 401 J=1,2
      BPF(3,J)=BPF(3,J)/XNORM
      CPF(3,J)=CPF(3,J)/XNORM
      DO 401 K=1,2
      APF(3,J,K)=APF(3,J,K)/XNORM
  401 CONTINUE
      RETURN
      END
      SUBROUTINE FACCEL(ACCEL,ISTEP,YFF,KN,IGYRO)
C... THIS PROGRAM EXECUTES THE FEEDFORWARD (ACCELEROMETER) LOOP;
C... COMPUTING ACP GAIN MATRIX AND FILTER OUTPUTS
C...
      REAL ACCEL(3),OUT(3,4,3),IN(3,4,3),YFF(3),PACP(3)
      REAL KN(3,3),YPFA(3),YF1(3),YF2(3),YF3(3)
      LOGICAL AFIRST
      REAL APFA(3,2,2),BPFA(3,2),CPFA(3,2),DPFA(3)
      REAL AFF1(3,2,2),BFF1(3,2),CFF1(3,2),DFF1(3)
      REAL AFF2(3,2,2),BFF2(3,2),CFF2(3,2),DFF2(3)
      REAL AFF3(3,2,2),BFF3(3,2),CFF3(3,2),DFF3(3)
      DIMENSION XPF1(3),XPF2(3),XPF1N(3),XPF2N(3)
      DIMENSION XF1(3),XF2(3),X1F1N(3),X1F2N(3)
      DIMENSION X2F1(3),X2F2(3),X2F1N(3),X2F2N(3)
      DIMENSION X3F1(3),X3F2(3),X3F1N(3),X3F2N(3)
      COMMON /ACCIO/ IN,OUT
      COMMON /FFCOEF/ APFA,BPFA,CPFA,DPFA,AFF1,BFF1,CFF1,DFF1,
     $                AFF2,BFF2,CFF2,DFF2,AFF3,BFF3,CFF3,DFF3
      DATA XPF1,XPF2,XF1,XF2,X2F1,X2F2,X3F1,X3F2/24*0.0/
      DATA AFIRST/.TRUE./
      IF (.NOT.AFIRST) GO TO 30
C... CALL SUBROUTINE TO INPUT COEFFICIENTS OF TRANSFER FUNCTIONS
      CALL FFDATA
```

```
C... CALL SUBROUTINE TO OBTAIN ACP GAIN MATRIX
C... (THIS IS SENT BY SSC TO DCU AT 1 HZ UPDATES, SHOULD BE
C... RECOMPUTED AT THIS INTERVAL)
   30 CALL MMUL(KN,ACCEL,PACP,3,3,1)
C
C
C   ACCELEROMETER PREFILTER
C
      DO 100 I=1,3
      XPF1N(I)=APFA(I,1,1)*XPF1(I)+APFA(I,1,2)*XPF2(I)+
     $BPFA(I,1)*PACP(I)
      XPF2N(I)=APFA(I,2,1)*XPF1(I)+APFA(I,2,2)*XPF2(I)+
     $BPFA(I,2)*PACP(I)
      YPFA(I)=CPFA(I,1)*XPF1(I)+CPFA(I,2)*XPF2(I)+DPFA(I)*PACP(I)
      XPF1(I)=XPF1N(I)
      XPF2(I)=XPF2N(I)
  100 CONTINUE
C
C
C   SECOND ORDER ACCELEROMETER FEEDFORWARD FILTERS
C
      DO 210 I=1,3
      X1F1N(I)=AFF1(I,1,1)*XF1(I)+AFF1(I,1,2)*XF2(I)
     $+BFF1(I,1)*YPFA(I)
      X1F2N(I)=AFF1(I,2,1)*XF1(I)+AFF1(I,2,2)*XF2(I)
     $+BFF1(I,2)*YPFA(I)
      YF1(I)=CFF1(I,1)*XF1(I)+CFF1(I,2)*XF2(I)+DFF1(I)*YPFA(I)
      XF1(I)=X1F1N(I)
      XF2(I)=X1F2N(I)
  210 CONTINUE
C
C
      DO 220 I=1,3
      X2F1N(I)=AFF2(I,1,1)*X2F1(I)+AFF2(I,1,2)*X2F2(I)
     $+BFF1(I,1)*YF1(I)
      X2F2N(I)=AFF2(I,2,1)*X2F1(I)+AFF2(I,2,2)*X2F2(I)
     $+BFF2(I,2)*YF1(I)
      YF2(I)=CFF2(I,1)*X2F1(I)+CFF2(I,2)*X2F2(I)+DFF2(I)*YF1(I)
      X2F1(I)=X2F1N(I)
      X2F2(I)=X2F2N(I)
  220 CONTINUE
C
C
      DO 230 I=1,3
      X3F1N(I)=AFF3(I,1,1)*X3F1(I)+AFF3(I,1,2)*X3F2(I)
     $+BFF3(I,1)*YF2(I)
      X3F2N(I)=AFF3(I,2,1)*X3F1(I)+AFF3(I,2,2)*X3F2(I)
     $+BFF3(I,2)*YF2(I)
      YF3(I)=CFF3(I,1)*X3F1(I)+CFF3(I,2)*X3F2(I)+DFF3(I)*YF2(I)
      X3F1(I)=X3F1N(I)
      X3F2(I)=X3F2N(I)
  230 CONTINUE
C
```

```
      IF(IGYRO.EQ.4)THEN
        DO 800 I=1,3
        YFF(I)=YF3(I)
  800   CONTINUE
      ENDIF
C
      AFIRST=.FALSE.
      RETURN
      END
      SUBROUTINE FGYRO(GRATE,ISTEP,PID,WD,TDPO,IGYRO)
C... THIS PROGRAM EXECUTES THE FEEDBACK (GYRO) LOOP
      REAL GRATE(4,3),XOUT(3),OUT(3,8,3),IN(3,8,3),PID(3)
      REAL APF(3,2,2),BPF(3,2),CPF(3,2),DPF(3)
      REAL AFB1(2,2,2),BFB1(2,2),CFB1(2,2),DFB1(2),XR(2),XRN(2)
      REAL AFB2(2,2,2),BFB2(2,2),CFB2(2,2),DFB2(2),E(3),S(3)
      REAL AFB3(2,2,2),BFB3(2,2),CFB3(2,2),DFB3(2)
      REAL AR(2),BR(2),CR(2),DR(2),SPSC(3),VC(3)
      REAL XVEC(1,3),WD(3),TDPO(3,3),VEC(1,3)
      REAL HKD(3),HKP(3),HKI(3),LKD(3),LKP(3),LKI(3)
      LOGICAL H_FLAG,H_FFDD,ZERIN1,ZERIN2,ZERIN3,FIRST,GFIRST
        DIMENSION QUN(4),DELQ(4),GPFO(3)
        COMMON /ITIME/ NDT
        COMMON /SFLAG/ H_FLAG,H_FFDD,FIRST
        COMMON /ANGLE/ ANG(3),DESANG(3)
        COMMON /ADF/ DD(3),QD(4),QM(4),Q0(4),CM(4),Q(4),VSTN(6)
        COMMON /SUM/ QS(4),DRIFT(3),PE(3)

      COMMON /GYROIO/ IN,OUT
      COMMON /FBCOEF/ APF,BPF,CPF,DPF,AFB1,BFB1,CFB1,DFB1,
     $                AFB2,BFB2,CFB2,DFB2,AFB3,BFB3,CFB3,DFB3,
     $                AR,BR,CR,DR,HKD,HKP,HKI,LKD,LKP,LKI,SPSC,VC
      DIMENSION XG1(3),XG2(3),XB11(2),XB12(2),XB21(2),XB22(2),
     $          XB31(2),XB32(2),PW(3),PDW(3),YF(3),YG(3),PDC(3)
      DIMENSION XG1N(3),XG2N(3),XB11N(2),XB12N(2),XB21N(2),
     $          XB22N(2),XB31N(2),XB32N(2),YB1(2),YB2(2),YB3(2)
      DATA GFIRST/.TRUE./
      DATA XG1,XG2,XB11,XB12,XB21,XB22,XB31,XB32/18*0.0/
C...INITIALIZE FIRST TIME THROUGH
      ZERIN1=.FALSE.
      ZERIN2=.FALSE.
      ZERIN3=.FALSE.
      ELGMAX=.00425
      EHGMAX=.00116
      IF(H_FLAG.OR.H_FFDD)EMAX=ELGMAX
      IF(.NOT.(H_FLAG.OR.H_FFDD))EMAX=EHGMAX
      IF(IGYRO.EQ.2)THEN
        DO 900 I=1,3
        PDC(I)=(GRATE(1,I)+GRATE(2,I))/2.0
  900   CONTINUE
      ENDIF
      IF(IGYRO.EQ.4)THEN
        DO 901 I=1,3
        PDC(I)=(GRATE(3,I)+GRATE(4,I))/2.0
```

```
 901   CONTINUE
       ENDIF
       IF (.NOT.GFIRST) GO TO 30
         CALL FILIO


C... CALL SUBROUTINE TO INPUT STATE-SPACE COEFFICIENTS
       CALL FBDATA
C
C
C    GYRO PREFILTER ALL THREE AXES
C
C
    30 DO 100 I=1,3
       XG1N(I)=APF(I,1,1)*XG1(I)+APF(I,1,2)*XG2(I)+BPF(I,1)*PDC(I)
       XG2N(I)=APF(I,2,1)*XG1(I)+APF(I,2,2)*XG2(I)+BPF(I,2)*PDC(I)
       YG(I)=CPF(I,1)*XG1(I)+CPF(I,2)*XG2(I)+DPF(I)*PDC(I)
       XG1(I)=XG1N(I)
       XG2(I)=XG2N(I)
   100 CONTINUE
C
C
C    SUBTRACT DRIFT ESTIMATES (CALCULATED IN ADF)
C    FROM RATE SIGNALS
C
       DO 200 I=1,3
       PW(I)=YG(I)-DD(I)+DRIFT(I)
   200 CONTINUE
C
C
C   SUBTRACT DESIRED RATES FROM PREFILTERED SIGNALS
C
       DO 110 I=1,3
       PDW(I)=PW(I)-WD(I)
   110 CONTINUE
C
C
C    REPRESENT RATE FILTERS IN STATE-SPACE FORM
C    EL AND XL AXES ONLY
C
C
       DO 120 I=1,2
       XB11N(I)=AFB1(I,1,1)*XB11(I)+AFB1(I,1,2)*XB12(I)+
      $BFB1(I,1)*PDW(I+1)
       XB12N(I)=AFB1(I,2,1)*XB11(I)+AFB1(I,2,2)*XB12(I)+
      $BFB1(I,2)*PDW(I+1)
       YB1(I)=CFB1(I,1)*XB11(I)+CFB1(I,2)*XB12(I)+DFB1(I)*PDW(I+1)
       XB11(I)=XB11N(I)
       XB12(I)=XB12N(I)
   120 CONTINUE
C
C
       DO 130 I=1,2
```

```
      XB21N(I)=AFB2(I,1,1)*XB21(I)+AFB2(I,1,2)*XB22(I)+
     $BFB2(I,1)*YB1(I)
      XB22N(I)=AFB2(I,2,1)*XB21(I)+AFB2(I,2,2)*XB22(I)+
     $BFB2(I,2)*YB1(I)
      YB2(I)=CFB2(I,1)*XB21(I)+CFB2(I,2)*XB22(I)+DFB2(I)*YB1(I)
      XB21(I)=XB21N(I)
      XB22(I)=XB22N(I)
  130 CONTINUE
C
C
      DO 140 I=1,2
      XB31N(I)=AFB3(I,1,1)*XB31(I)+AFB3(I,1,2)*XB32(I)+
     $BFB3(I,1)*YB2(I)
      XB32N(I)=AFB3(I,2,1)*XB31(I)+AFB3(I,1,2)*XB32(I)+
     $BFB3(I,2)*YB2(I)
      YB3(I)=CFB3(I,1)*XB31(I)+CFB3(I,2)*XB32(I)+DFB3(I)*YB2(I)
      XB31(I)=XB31N(I)
      XB32(I)=XB32N(I)
  140 CONTINUE
C
C
C   FIRST ORDER RATE FILTER
C   EL AND XL AXES ONLY
C
      DO 150 I=1,2
      XRN(I)=AR(I)*XR(I)+BR(I)*YB3(I)
      YF(I+1)=CR(I)*XR(I)+DR(I)*YB3(I)
      XR(I)=XRN(I)
  150 CONTINUE
      YF(1)=PDW(1)
C
C
C   DETERMINE QUATERNIONS BY INTEGRATING PREFILTERED
C   RATE SIGNALS AND EVALUATE ATTITUDE AND
C   INTEGRAL LOOPS
C
      IF(IGYRO.EQ.4)GO TO 999
C
C
       GPFO(1)=PW(1)
       GPFO(2)=PW(2)
       GPFO(3)=PW(3)
C
C     CALCULATE DELTA Q BASED ON GYRO RATES
      DELQ(1)= GPFO(3)*Q(2)-GPFO(2)*Q(3)+GPFO(1)*Q(4)
      DELQ(2)=-GPFO(3)*Q(1)+GPFO(1)*Q(3)+GPFO(2)*Q(4)
      DELQ(3)= GPFO(2)*Q(1)-GPFO(1)*Q(2)+GPFO(3)*Q(4)
      DELQ(4)=-GPFO(1)*Q(1)-GPFO(2)*Q(2)-GPFO(3)*Q(3)
C      INTEGRATE QUATERNION
      Q(1)=Q(1)+.02*DELQ(1)
      Q(2)=Q(2)+.02*DELQ(2)
      Q(3)=Q(3)+.02*DELQ(3)
      Q(4)=Q(4)+.02*DELQ(4)
```

```
C      SAVE UNNORMALIZED QUATERNION FOR PLOTTING
       QUN(1)=Q(1)
       QUN(2)=Q(2)
       QUN(3)=Q(3)
       QUN(4)=Q(4)
C      CALCULATE NORMALIZATION ERROR AND RESCALE QUATERNION
       EQ=1-Q(1)*Q(1)-Q(2)*Q(2)-Q(3)*Q(3)-Q(4)*Q(4)
       SF=1+.5*EQ
       Q(1)=SF*Q(1)
       Q(2)=SF*Q(2)
       Q(3)=SF*Q(3)
       Q(4)=SF*Q(4)
       DO 249 I=1,4
 249   QS(I)=QS(I)+Q(I)
C      CALCULATE ATTITUDE DCM (TA)
       TA11=2*(Q(4)*Q(4)+Q(1)*Q(1))-1
       TA12=2*(Q(1)*Q(2)+Q(3)*Q(4))
       TA13=2*(Q(1)*Q(3)-Q(2)*Q(4))
       TA21=2*(Q(1)*Q(2)-Q(3)*Q(4))
       TA22=2*(Q(4)*Q(4)+Q(2)*Q(2))-1
       TA23=2*(Q(2)*Q(3)+Q(1)*Q(4))
C      CALCULATE ATT ERROR   E=TA*TDPO(-1)
       EKX= TA21*TDPO(3,1)+TA22*TDPO(3,2)+TA23*TDPO(3,3)
       EKY=-TA11*TDPO(3,1)-TA12*TDPO(3,2)-TA13*TDPO(3,3)
       EKZ= TA11*TDPO(2,1)+TA12*TDPO(2,2)+TA13*TDPO(2,3)
C
C
C   ATTITUDE ERROR LIMITATION FOR SLEW
C   ADDED BY JMR
C
C
       IF(EKX.LT.0.0)THEN
          SIGNX=-1.0
       ELSE
          SIGNX=1.0
       ENDIF
       IF(EKY.LT.0.0)THEN
          SIGNY=-1.0
       ELSE
          SIGNY=1.0
       ENDIF
       IF(EKZ.LT.0.0)THEN
          SIGNZ=-1.0
       ELSE
          SIGNZ=1.0
       ENDIF
       IF(ABS(EKX).GT.EMAX)THEN
          EKX=EMAX*SIGNX
          ZERIN1=.TRUE.
       ENDIF
       IF(ABS(EKY).GT.EMAX)THEN
          EKY=EMAX*SIGNY
          ZERIN2=.TRUE.
```

```
      ENDIF
      IF(ABS(EKZ).GT.EMAX)THEN
         EKZ=EMAX*SIGNZ
         ZERIN3=.TRUE.
      ENDIF
C
C
       E(1)=EKX
       E(2)=EKY
       E(3)=EKZ
C
C   EVALUATE INTEGRAL OF ATTITUDE ERROR
C
C
      DO 190 I=1,3
      S(I)=S(I)+E(I)*.04
  190 CONTINUE
C
C
C  ZERO INTEGRAL LOOP IF E > EMAX
C
      IF(ZERIN1)S(1)=0.0
      IF(ZERIN2)S(2)=0.0
      IF(ZERIN3)S(3)=0.0
C
C
C
C   COMPUTE PID FOR HIGH GAIN MODE
C   OR FOR LOW GAIN MODE
C
 999  IF(H_FLAG.OR.H_FFDD)THEN
       WRITE(6,7777)
 7777  FORMAT(1X,' LOW GAINS')
       PID(1)=SPSC(1)*VC(1)*(LKD(1)*YF(1)+LKP(1)*E(1)+LKI(1)*S(1))
       DO 803 I=2,3
       PID(I)=SPSC(I)*VC(I)*(LKD(I)*YF(I)+LKP(I)*E(I)+LKI(I)*S(I))
  803  CONTINUE
      ELSE
       WRITE(6,7778)
 7778  FORMAT(1X,' HIGH GAINS')
       PID(1)=VC(1)*(HKD(1)*YF(1)+HKP(1)*E(1)+HKI(1)*S(1))
       DO 804 I=2,3
       PID(I)=VC(I)*(HKD(I)*YF(I)+HKP(I)*E(I)+HKI(I)*S(I))
  804  CONTINUE
      ENDIF
      GFIRST=.FALSE.
      RETURN
      END
      SUBROUTINE RMODE(NM,TMODE,IMODE,DESMOD)
      DIMENSION TMODE(10),IMODE(10),DESMOD(10,3)
      READ(17,*)NM
      DO 10 I=1,NM
      READ(17,*)TMODE(I),IMODE(I),(DESMOD(I,J),J=1,3)
```

```
   10 CONTINUE
      RETURN
      END
      SUBROUTINE RSLEW(H_OMGG,H_RCUR,H_OMGD,H_TDPO,H_RESD,
     $INSLEW,GAC,QQ,FOUR)
C
      LOGICAL INSLEW,GAC,FOUR
      REAL H_RCUR(3),H_OMGG(3),H_RESD(3)
      DIMENSION H_TPP1(3,3),H_TSTS(3,3),H_OMGD(3),H_TDPO(3,3)
C
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      LOGICAL H_FLAG(32)
C
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      LOGICAL H_FFDD(28)
      INTEGER*2       H_DWN1,H_DWN5,H_CFHS,H_ODLB,H_FFDI,H_ALI1
      REAL*4          H_FFDF
C
      DIMENSION QQ(4),TEMP(3,3)
      COMMON /CBUF/   H_DWN1(59),H_DWN5(49),H_FFDI(30),
     *                H_CFHS(21),H_ODLB(27),H_ALI1,H_FFDF(21)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      INTEGER*2       H_LGYR,H_ALI2
C
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
      COMMON /CMM/    H_MMTT,H_Q(4)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      LOGICAL*1       H_GYHR,H_STRK,H_ERRT
      INTEGER*2       H_KFF, H_UNFL,H_ULIM,H_TQRN,H_TCP, H_VC, H_ALOM,
     *                H_ALS, H_OMRN,H_FIL1,
     *                H_KFFN,H_UNFN,H_ULMN,H_TRNN,H_TCPN,H_VCN,H_ALON,
     *                H_ALSN,H_ORNN,H_FL1N,H_ALI4,
     *                H_MXRT,H_CNTS,H_DCR,H_ALI3,H_SPIN,H_EPON,H_EPOF,
     *                H_SWBO,H_MPMX,H_TUTO
      INTEGER*4       H_EMAX,H_QSHT,H_EMXN,H_QSHN
      REAL*4          H_TPE1,H_TSTA,H_TSTS,H_TPP1,H_TS13,H_TS12,H_MPA,
     *                H_DDIN,H_MPC1,H_MPC2,H_MPC3,H_COMX,H_ADMX,H_TQMX,
     *                H_JROL,H_JLAT,H_DSLW,H_DCO,H_DSEP,H_SCNA,H_SCNG,
     *                H_MPCD,H_EHMX,H_PHSO,H_SITO,H_GYCH,H_CO1,H_CO3,
     *                H_CO4,H_CO5,H_CO7,H_CO8,H_CI1,H_CI5,H_CI6,H_DST,
     *                H_TETA,H_RULL,H_RUUL,H_STLL,H_STUL,H_SPSC,H_SRMX,
     *                H_ROMX,H_ESMX
```

```
C
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
       REAL H_OMGN(3)
C
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      REAL*4          H_TAPO,H_TOO1,H_TSHO,H_TSO1,H_TE1R,H_TA1R,H_TNPO,
     *                H_TSBY,H_TAPS,H_TTPO,H_TOS1
C
      COMMON /CDCM/   H_TAPO(3,3),H_TOO1(3,3),H_TSHO(3,3),H_TSO1(3,3),
     *                H_TE1R(3,3),H_TA1R(3,3),H_TNPO(3,3),H_TSBY(3,3),
     *                H_TAPS(3,3),H_TTPO(3,3),H_TOS1(3,3)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON /ORB/ OMBY(3)
      COMMON /COSV/   T_BY_I(3,3),TBYI(3,3),TA(3,3),T_PP_I(3,3)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     *-------------------------------------------------------------*
C     *               INTERNAL VARIABLES                            *
C     *-------------------------------------------------------------*
C
      DIMENSION OMEGAL(3),OMGP(3),RESDES(3),DUM(3),Q(4),HILFSA(3),
     *          TDFP1S(3,3),TDFPP(3,3),TSTART(3,3),TP1G(3,3),
     *          TLAT(3,3),TROL(3,3),ODP(3,3),CALC1(3,3),CALC2(3,3),
     *          TTTPO(3,3),TAPO(3,3),TAP1ST(3,3),TPP1(3,3),TDPO(3,3),
     *          TSTSH(3,3),ROTP(3),QORB(4),TORB(3,3)
C
      INTEGER   Z,I,R,JPSI,VZ1,VZ2
C
      REAL*8    Q,QORB
      REAL      OMEGAL,OMGP,RESDES,DUM,TDFP1S,TDFPP,HILFSA,TSTART,
     *          TP1G,TLAT,TROL,ODP,CALC1,THLDM,THL2DM,THRDA,SLEWMX,
     *          THLDA,THRA,THLA,TH,TH2DM,THDM,THA,THDA,PSI,THR,REDU,
     *          THL,THRDM,THR2DM,LY,LZ,SEL,CEL,SXL,CXL,SRO,CRO,RANGE,
     *          TTTPO,TAPO,TAP1ST,TPP1,TDPO,TSTSH,TORB,ROTP,CALC2,
     *          DELTA,ENDCON,ROLCUR,ROLCOR,DELTAT,PI,LENGTH,DIAGON
C
C------------------------
C  ADDED BY ADAMS
C
      REAL LX,THRG,THRP
      DATA THRG,THRP/0.0,0.0/
C
C--------------------------
C
      LOGICAL   LOGO,LOG1,ROLAVO
C
```

```
      DATA      THLDM,THRDA,THLDA,THRA,THLA,TH,TH2DM,THDM,THA,THL2DM,
     *          THDA,PSI,THR,THL,THRDM,LY,LZ,SEL,CEL,SXL,THR2DM,
     *          CXL,SRO,CRO,DELTA,ENDCON,ROLCUR,ROLCOR,DELTAT/29*0.0/,
     *          Z,I,R,JPSI/4*0/,SLEWMX,REDU/2*0.0/,
     *          ROLAVO/.TRUE./,RANGE/0.0174533/,
     *          PI/3.1415927/
C
      DATA      OMEGAL,OMGP,RESDES,DUM,Q,HILFSA,ROTP/22*0.0/,
     *          TDFP1S,TDFPP,TSTART,ODP,CALC1,CALC2/54*0.0/,
     *          TP1G/1.,3*0.,.7071068,-.7071068,0.,2*.7071068/,
     *          TLAT/1.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,1.0/,
     *          TROL/1.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,1.0/,
     *          TORB/1.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,1.0/,
     *          QORB/0.0,0.0,0.0,1.0/
C
      DATA H_TPP1/1.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,1.0/
      DATA H_TSTS/1.0,0.0,0.0,0.0,-1.0,0.0,0.0,0.0,-1.0/
      DATA H_TAPO/1.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,1.0/
      DATA H_TTPO/1.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,1.0/
      DATA H_TAPS/0.0,0.0,-1.0,0.0,-1.0,0.0,-1.0,0.0,0.0/
C
C
      DATA H_CNTS,H_LGYR,H_TQMX,H_ADMX,H_JROL,H_JLAT,H_DSLW/
     $25,5,10.0,.0157,2524.5,8145.0,.00349/
      DATA H_FFDD(28)/.FALSE./
C
C   SET NON-VARYING SLEW PARAMETERS
C
      H_FLAG(2)=INSLEW
      H_FFDD(7)=GAC
C
C
C
C   COMPUTE H_TAPO
      IF(.NOT.(H_FLAG(2).OR.H_FFDD(7)))GO TO 443
C
      TEMP(1,1)=2.0*(QQ(1)**2.0+QQ(4)**2.0-.5)
      TEMP(1,2)=2.0*(QQ(1)*QQ(2)+QQ(3)*QQ(4))
      TEMP(1,3)=2.0*(QQ(1)*QQ(3)-QQ(2)*QQ(4))
      TEMP(2,1)=2.0*(QQ(1)*QQ(2)-QQ(3)*QQ(4))
      TEMP(2,2)=2.0*(QQ(2)**2.0+QQ(4)**2.0-.5)
      TEMP(2,3)=2.0*(QQ(2)*QQ(3)+QQ(1)*QQ(4))
      TEMP(3,1)=2.0*(QQ(1)*QQ(3)+QQ(2)*QQ(4))
      TEMP(3,2)=2.0*(QQ(2)*QQ(3)-QQ(1)*QQ(4))
      TEMP(3,3)=2.0*(QQ(3)**2.0+QQ(4)**2.0-.5)
      CALL TRANSP(TEMP,H_TAPO)
C
C
      CALL DCM(H_RESD(1),H_RESD(2),H_RESD(3),TEMP)
      CALL TRANSP(TEMP,H_TTPO)
      CALL DCM(H_RCUR(1),H_RCUR(2),H_RCUR(3),TEMP)
      CALL TRANSP(TEMP,H_TAPS)
C
```

```
C
C
C     *-------------------------------------------------*
C     *            TRANSPOSE  MATRIX                    *
C     *    T-TT-PO,T-A-P-O,T-A-P1-ST,T-P-P1,T-ST-SH     *
C     *-------------------------------------------------*
 443  DO 444 N=1,3
         DO 555 K=1,3
            TTTPO(N,K)  = H_TTPO(K,N)
            TAPO(N,K)   = H_TAPO(K,N)
            TAP1ST(N,K) = H_TAPS(K,N)
            TPP1(N,K)   = H_TPP1(K,N)
            TSTSH(N,K)  = H_TSTS(K,N)
  555    CONTINUE
  444 CONTINUE
C
C     ==================================================
C
C     *-----------------------------------------------------------*
C     *          GIMBAL-ANGLE-COMMAND-MODE  ?                     *
C     *-----------------------------------------------------------*
C
      IF ( .NOT.H_FFDD(7) )  GOTO 10
C
C        *-----------------------------------------------------------*
C        *      AUTOMATIC RESTART OF TRAJECTORY WHEN COMMANDED       *
C        *      GIMBAL-ANGLES ARE CHANGED                            *
C        *-----------------------------------------------------------*
C
         LOGO=(H_RESD(1).NE.RESDES(1)).OR.(H_RESD(2).NE.RESDES(2)).OR.
     *        (H_RESD(3).NE.RESDES(3))
         IF ( .NOT.LOGO ) GOTO 20
            ROLAVO = .TRUE.
            I=H_CNTS
            DO 15 N=1,3
               RESDES(N)=H_RESD(N)
   15       CONTINUE
   20    CONTINUE
C
C        *-----------------------------------------------------------*
C        *     GENERATE DESIRED PLATFORM ATTITUDE DCM  ( TDFPP )     *
C        *     VERSUS CURRENT PLATFORM ATTITUDE BY MEANS OF ACTUAL   *
C        *     AND DESIRED GIMBAL ATTITUDE                           *
C        *-----------------------------------------------------------*
C
         SEL = SIN ( H_RESD(2) )
         CEL = COS ( H_RESD(2) )
         SXL = SIN ( H_RESD(3) )
         CXL = COS ( H_RESD(3) )
         SRO = SIN ( H_RESD(1) )
         CRO = COS ( H_RESD(1) )
         TDFP1S(1,1) = CXL*CEL
         TDFP1S(1,2) = SXL
```

```
      TDFP1S(1,3) = -CXL*SEL
      TDFP1S(2,1) = -CRO*SXL*CEL+SRO*SEL
      TDFP1S(2,2) = CRO*CXL
      TDFP1S(2,3) = CRO*SXL*SEL+SRO*CEL
      TDFP1S(3,1) = SRO*SXL*CEL+CRO*SEL
      TDFP1S(3,2) = -SRO*CXL
      TDFP1S(3,3) = -SRO*SXL*SEL+CRO*CEL
      CALL MAMA  (TPP1,TDFP1S,TDFPP,3)
      CALL MAMAT (TDFPP,TAP1ST,CALC1,3)
      CALL MAMAT (CALC1,TPP1,TDFPP,3)
C
C     *--------------------------*
C     *   DESIRED ROLL ANGLE     *
C     *--------------------------*
C
      GOTO 30
   10 CONTINUE
      IF ( H_FLAG(2) ) GOTO 40
C
C        *----------------------------------*
C        *    NO EXECUTION OF SLEWING S/W   *
C        *----------------------------------*
C
         ROLAVO = .TRUE.
         I=H_CNTS
         RETURN
   40    CONTINUE
C
C     *-------------------------------------------------------*
C     *   GENERATE DESIRED PLATFORM ATTITUDE  ( DCM TDFPP )   *
C     *   VERSUS CURRENT PLATFORM ATTITUDE BY MEANS OF ACTUAL *
C     *   AND DESIRED INERTIAL ATTITUDE                       *
C     *-------------------------------------------------------*
C
      CALL MAMAT (TTTPO,TAPO,TDFPP,3)
C------------------------
C  ADDED BY ADAMS
C
C  DESIRED GIMBAL-ANGLES IN FORM OF DCM
C
      CALL MATMA(TPP1,TDFPP,CALC1,3)
      CALL MAMA(CALC1,TPP1,CALC2,3)
      CALL MAMA(CALC2,TAP1ST,TDFP1S,3)
C
C  CALCULATE DESIRED GIMBAL-ROLL FROM DCM
C
      ROLDES = ATAN2(-TDFP1S(3,2),TDFP1S(2,2))
C
C-------------------------------
   30 CONTINUE
C
C     *-------------------------------------*
C     *   DESIRED ROLL-ANGLE ( IN SLEW )    *
```

```
C     *--------------------------------------*
C     *   CONTROL OF SLEWING END CONDITION   *
C     *--------------------------------------*
C
      IF (.NOT.( H_FFDD(28) .AND. H_FLAG(2) ) ) GOTO 905
         LENGTH = SQRT( H_OMGN(1) * H_OMGN(1) + H_OMGN(2) * H_OMGN(2) +
     *                  H_OMGN(3) * H_OMGN(3) )
         DELTA = LENGTH * H_CNTS * 0.2
         GOTO 910
  905 CONTINUE
         DELTA = 0.0
  910 CONTINUE
      ENDCON = H_DSLW + DELTA
C
      DUM(1) = ABS ( TDFPP(1,2) )
      DUM(2) = ABS ( TDFPP(1,3) )
      DUM(3) = ABS ( TDFPP(2,3) )
C
      DIAGON = TDFPP(1,1) + TDFPP(2,2) + TDFPP(3,3) - 1.0
      IF ( DIAGON .GE. 0.0 ) VZ1 = 1
      IF ( DIAGON .LT. 0.0 ) VZ1 = -1
      LOGO = VZ1 .LT. 0.0
      LOG1 = ENDCON .LE. ( AMAX1( DUM(1),DUM(2),DUM(3) ) )
      IF ( .NOT.( LOGO .OR. LOG1 ) ) GOTO 50
C
   51    IF ( .NOT.(I .EQ. H_CNTS) ) GOTO 60
C        --- RESET INTEGRATIONS COUNTER ---
            I=1
C        --- BRANCH TO TRAJECTORY INITIALISATION ---
            R=1
            GOTO 70
   60    CONTINUE
C        --- BRANCH TO TRAJECTORY INTIGRATION ---
            R=2
   70    CONTINUE
         GOTO 80
   50 CONTINUE
C        --- INDICATES END OF SLEWING ---
         H_FLAG(6)=.TRUE.
         IF ( H_FFDD(7) ) GOTO 51
         I=H_CNTS
C        --- SETS DESIRED PLATFORM ATTITUDE TO ZERO ---
         DO 75 N=1,3
            H_OMGD(N)=0.0
   75    CONTINUE
      INSLEW=.FALSE.
      FOUR=.TRUE.
         RETURN
   80 CONTINUE
      GOTO (111,222),R
C
C     *-----------------------------------------------*
C     *   CASE: R=1                                   *
```

```
C     *   INITIALISATION FOR TRAJECTORY INTEGRATION    *
C     *------------------------------------------------*
C
C        ****************************************************************
C        *  TRANSFORM THE ORBITER-ROTATION INTO PLATFORM-COORDINATES  *
C        *     ---  ONLY IN GIMBAL-COMMAND-MOD ---                     *
C        ****************************************************************
C
C----------------------------------------
C
C   STATEMENT 111 MOVED TO HERE BY ADAMS TO AGREE
C   WITH NEW PATCH
C
C-------------------------------------------------
C
 111  CONTINUE
C
         IF ( .NOT.H_FFDD(7) ) GOTO 915
            HILFSA(1) = -OMBY(1)
            HILFSA(2) =  OMBY(2)
            HILFSA(3) = -OMBY(3)
            CALL MAMA ( TPP1,TAP1ST,CALC1,3 )
            CALL MAMA ( CALC1,TSTSH,CALC2,3 )
            CALL MAVE ( CALC2,HILFSA,ROTP,3 )
            GOTO 920
  915    CONTINUE
            ROTP(1) = 0.0
            ROTP(2) = 0.0
            ROTP(3) = 0.0
  920    CONTINUE
C
C       ***********************************************************
C
C     --- INITIALISE DCM FOR RECURSIVE ATTITUDE AND RATE PROCESSING ---
C-111 CONTINUE
         DO 120 N=1,3
            DO 140 K=1,3
               TSTART(N,K) = TAPO(N,K)
  140       CONTINUE
  120    CONTINUE
C
C        *------------------------------------------------*
C        *    LATERAL -MOTION:                            *
C        *        EULER LATERAL ANGLE            : THL     *
C        *        EULER LATERAL ACCELARATION     : THL2DM  *
C        *------------------------------------------------*
C
         IF ( .NOT.( ABS(TDFPP(1,1)) .GT. 1.0 ) )  GOTO 85
            IF ( TDFPP(1,1) .GE. 0.0 ) TDFPP(1,1)=1
            IF ( TDFPP(1,1) .LT. 0.0 ) TDFPP(1,1)=-1
   85    CONTINUE
C-----   THL = ARCOS ( TDFPP(1,1) )
         THL =  ACOS ( TDFPP(1,1) )
```

```
        IF ( .NOT.( THL .GT. H_DSLW ) )  GOTO 90
C          --- NON SINGULAR CASE ---
           LY = -TDFPP(1,3)/SIN(THL)
           LZ =  TDFPP(1,2)/SIN(THL)
           GOTO 100
   90   CONTINUE
C          --- SINGULAR CASE ---
           LY=0
           LZ=0
  100   CONTINUE
        THL2DM = H_TQMX/H_JLAT
C
C       *-------------------------------------------------*
C       *    ROLL-MOTION :                                *
C       *       EULER ROLL ANGLE            : THR         *
C       *       EULER ROLL ACCELARATION     : THR2DM      *
C       *-------------------------------------------------*
C
        Q(1) = 0
        Q(2) = LY * SIN(THL/2)
        Q(3) = LZ * SIN(THL/2)
        Q(4) = COS(THL/2)
        CALL QTODCM (Q,TLAT)
        CALL MAMAT (TDFPP,TLAT,TROL,3)
C----   THR = ATAN2 ( TROL(2,3) , TROL(2,2) )
C-------------------------------
C  ADDED BY ADAMS
C
      THRP = ATAN2 ( TROL(2,3) , TROL(2,2) )
C
      IF(.NOT.ABS(THRP).GT.H_DSLW) GO TO 150
      LX = 1
      GO TO 151
 150  LX = 0
 151  CONTINUE
C
C-------------------------------
        THR2DM = H_TQMX/H_JROL
C
C       *-------------------------------------------------*
C       *   RATE INITIALISATION OF TRAJECTORY BY MEANS    *
C       *   ACTUAL GYRO RATES                             *
C       *-------------------------------------------------*
C
        DO 200 N=1,3
           OMEGAL(N)=H_OMGG(N)
  200   CONTINUE
        IF ( .NOT.( H_LGYR .EQ. 2 ) )  GOTO 210
C          --- GYRO X IS OFF ---
           OMEGAL(1)=(-1)*H_OMGG(2)+(-1)*H_OMGG(3)+SQRT(3.0)*H_OMGG(4)
  210   CONTINUE
        IF ( .NOT.( H_LGYR .EQ. 3 ) )  GOTO 220
C          --- GYRO Y IS OFF ---
```

```
            OMEGAL(2)=(-1)*H_OMGG(1)+(-1)*H_OMGG(3)+SQRT(3.0)*H_OMGG(4)
  220     CONTINUE
          IF ( .NOT.( H_LGYR .EQ. 4 ) )  GOTO 230
C            --- GYRO Z IS OFF ---
            OMEGAL(3)=(-1)*H_OMGG(1)+(-1)*H_OMGG(2)+SQRT(3.0)*H_OMGG(4)
  230     CONTINUE
C
          ROLCUR = H_RCUR(1)
C
          CALL MAMA (TPP1,TP1G,CALC1,3)
          CALL MAVE (CALC1,OMEGAL,OMGP,3)
C
          OMGP(1)=OMGP(1)-ROTP(1)
          OMGP(2)=OMGP(2)-ROTP(2)
          OMGP(3)=OMGP(3)-ROTP(3)
C
          THRDA = OMGP(1)
          THLDA = LY * OMGP(2) + LZ * OMGP(3)
C
C         *-------------------------------------------------*
C         *   RESET ROLL-AND LATERAL EULER ANGLES TO ZERO    *
C         *   FOR INTEGRATIONS RESTART                       *
C         *-------------------------------------------------*
C
          THRA=0
          THLA=0
C
C         ****************************************
C         *  AVOIDANCE OF MECHANICAL ROLL-STOP AT  *
C         *     GIMBAL-ROLL 180 DEG.               *
C         ****************************************
C
C
C-----------------------------
C  ADDED BY ADAMS
C
      THRG = ROLDES - ROLCUR
      IF(.NOT.ABS(THRG).GT.PI) GO TO 231
      THRG = THRG - 2.0 * PI * SIN(THRG)
 231  CONTINUE
C
C---------------------------------
          IF ( .NOT.ROLAVO ) GOTO 830
            ROLAVO = .FALSE.
C
C----       IF ( .NOT.( ABS(ROLCUR + THR) .GT. PI ) ) GOTO 840
            IF ( .NOT.( ABS(ROLCUR +THRG) .GT. PI ) ) GOTO 840
C
C----           IF ( THR .GE. 0.0 ) VZ1 = 1
C----           IF ( THR .LT. 0.0 ) VZ1 = -1
                IF ( THRG.GE. 0.0 ) VZ1 = 1
                IF ( THRG.LT. 0.0 ) VZ1 = -1
C
```

```
               ROLCOR = -2 * PI * VZ1
               GOTO 850
  840       CONTINUE
               ROLCOR = 0.0
  850       CONTINUE
  830    CONTINUE
         IF ( .NOT.( ROLCOR .NE. 0.0 ) ) GOTO 860
C
C----       IF ( THR .GE. 0.0 ) VZ1 =1
C----       IF ( THR .LT. 0.0 ) VZ1 = -1
            IF ( THRG.GE. 0.0 ) VZ1 =1
            IF ( THRG.LT. 0.0 ) VZ1 = -1
C
            IF ( ROLCOR .GE. 0.0 ) VZ2 = 1
            IF ( ROLCOR .LT. 0.0 ) VZ2 = -1
         IF ( VZ1 .EQ. VZ2 ) ROLCOR =0.0
  860    CONTINUE
C
C----    THR = THR + ROLCOR
C
C-----------------------------
C  ADDED BY ADAMS
C
      IF((ABS(THRG).LT.(PI/2.0)).AND.(ROLCOR.EQ.0.0)) GO TO 861
      THR = THRG + ROLCOR
      GO TO 862
 861  THR = THRP
 862  CONTINUE
C
C-------------------------------
C
C       ************************************************************
C       *  ADAPTION OF MAX SLEWRATE WHEN THE Q-GYRO IS RUNNING  *
C       ************************************************************
C
         LENGTH = SQRT( ROTP(1)**2 + ROTP(2)**2 + ROTP(3)**2 )
         SLEWMX = H_ADMX - LENGTH
         IF ( H_LGYR .EQ. 5 ) GOTO 301
         IF (.NOT.(ABS(THR).GT.RANGE .AND. ABS(THL).GT.RANGE)) GOTO 301
            REDU = 0.5
            GOTO 302
  301    CONTINUE
            REDU = 1.0
  302    CONTINUE
         THRDM = REDU * SLEWMX
         THLDM = REDU * SLEWMX
C
C        ***************************************
C        *  LIMITATION OF INITIAL EULER-RATES  *
C        ***************************************
C
         IF (.NOT.( ABS(THRDA) .GT. THRDM ) ) GOTO 810
             IF ( THRDA .GE. 0.0 ) VZ1 = 1
```

```
            IF ( THRDA .LT. 0.0 ) VZ1 = -1
            THRDA = THRDM * VZ1
  810    CONTINUE
         IF (.NOT.( ABS(THLDA) .GT. THLDM ) ) GOTO 820
            IF ( THLDA .GE. 0.0 ) VZ1 = 1
            IF ( THLDA .LT.0.0 ) VZ1 = -1
            THLDA = THLDM * VZ1
  820    CONTINUE
C
         GOTO 250
  222 CONTINUE
C
C        *-----------------------------------*
C        *   CASE :  R=2                     *
C        *   INCREMENT INTEGRATION COUNTER   *
C        *-----------------------------------*
C
         I=I+1
  250 CONTINUE
C
C     *-----------------------------------*
C     *   INTEGRATION OF EULER ANGLES     *
C     *   Z=1 :  LATERAL AXIS             *
C     *   Z=2 :  ROLL AXIS                *
C     *-----------------------------------*
C
      DO 300 Z=1,2
         IF ( .NOT.(Z .EQ. 1) ) GOTO 310
            THDA  = THLDA
            THA   = THLA
            TH    = THL
            TH2DM = THL2DM
            THDM  = THLDM
            GOTO 320
  310    CONTINUE
            THDA  = THRDA
            THA   = THRA
            TH    = THR
            TH2DM = THR2DM
            THDM  = THRDM
  320    CONTINUE
C
C        *-------------------------------------------------*
C        *    CONTROL CONDITION FOR TORQUE DIRECTION       *
C        *    SWITCHING CURVE DUE TO PONTRJAGIN PRINCIPLE  *
C        *-------------------------------------------------*
C
         PSI= (-0.5/TH2DM) * THDA * ABS(THDA) + (TH-THA)
C
C        *-------------------------------*
C        *    EULER RATE INTEGRATION     *
C        *-------------------------------*
C
```

```
        IF ( PSI .GE. 0.0 )  JPSI=1
        IF ( PSI .LT. 0.0 )  JPSI=-1
        THDA = THDA + 0.2 * TH2DM * JPSI
C
C       *--------------------------*
C       *  EULER RATE LIMITATION   *
C       *--------------------------*
C
C----   IF ( ABS(THDA) .GT. THDM ) THDA=THDM*JPSI
        IF ( ABS(THDA) .GE. THDM ) THDA=THDM*JPSI
C
C       *--------------------------*
C       *  EULER ANGLE INTEGRATION *
C       *--------------------------*
C
        THA = THA + 0.2 * THDA
        IF ( .NOT.( Z .EQ. 1 ) )  GOTO 330
C
C          *-----------------------------------------------*
C          *   LATERAL MOTION IN FORM OF DIRECTION COSINE  *
C          *-----------------------------------------------*
C
           THLDA = THDA
           THLA  = THA
           Q(1) = 0.0
           Q(2) = LY*SIN(THA/2)
           Q(3) = LZ*SIN(THA/2)
           Q(4) = COS(THA/2)
           CALL QTODCM (Q,TLAT)
           GOTO 340
  330   CONTINUE
C
C          *-----------------------------------------------*
C          *   ROLL MOTION IN FORM OF DIRECTION COSINE     *
C          *-----------------------------------------------*
C
           THRDA = THDA
           THRA  = THA
           TROL(1,1) = 1.0
           TROL(1,2) = 0.0
           TROL(1,3) = 0.0
           TROL(2,1) = 0.0
           TROL(2,2) = COS(THA)
C
C----       TROL(2,3) = SIN(THA)
           TROL(2,3) = LX * SIN(THA)
C
           TROL(3,1) = 0.0
           TROL(3,2) = -TROL(2,3)
           TROL(3,3) = COS(THA)
  340   CONTINUE
  300 CONTINUE
C
```

```
C     *---------------------------------------*
C     *   DESIRED PLATFORM ATTITUDE DCM SENT TO  *
C     *   DCU FASTLOOP CONTROLER                *
C     *---------------------------------------*
C
      DELTAT = 0.1 * I
      QORB(1) = ROTP(1) * DELTAT
      QORB(2) = ROTP(2) * DELTAT
      QORB(3) = ROTP(3) * DELTAT
      QORB(4) = DSQRT( 1.0 - QORB(1) * QORB(1) - QORB(2) * QORB(2) -
     *                QORB(3) * QORB(3) )
      CALL QTODCM ( QORB,TORB )
C
      CALL MAMA (TORB,TROL,CALC1,3)
      CALL MAMA (CALC1,TLAT,ODP,3)
      CALL MAMA (ODP,TSTART,TDPO,3)
C
C     *-------------------------------*
C     *   TRANSPOSE MATRIX  T-D-P-O   *
C     *-------------------------------*
C
      DO 666 N=1,3
         DO 777 K=1,3
            H_TDPO(K,N) = TDPO(N,K)
  777    CONTINUE
  666 CONTINUE
C
C
C  PRINT LOOP ADDED BY JMR 8/28/89
C
C
      DO 6789 III=1,3
 6789 CONTINUE
C
C     ================================
C
C     *-------------------------------------*
C     *   DESIRED PLATFORM RATE SENT TO     *
C     *   DCU FASTLOOP CONTROLLER           *
C     *-------------------------------------*
C
C
C---- HILFSA(1) = THRDA + ROTP(1)
      HILFSA(1) = LX * THRDA + ROTP(1)
C
      HILFSA(2) = LY*THLDA + ROTP(2)
      HILFSA(3) = LZ*THLDA + ROTP(3)
      CALL MAVE (ODP,HILFSA,H_OMGD,3)
      RETURN
      END
      SUBROUTINE  MAMA (A,B,C ,N)
C
C     ***********************************************************************
```

```
C     MATRIX * MATRIX A*B=C
C     ************************************************************************
C
      DIMENSION A(N,N),B(N,N),C(N,N)
      DO  1  I= 1,N
      DO  1  J= 1,N
      C(I,J)= 0.
      DO  1  K= 1,N
      C(I,J)= C(I,J) + A(I,K)*B(K,J)
    1 CONTINUE
C
      RETURN
      END
      SUBROUTINE MATMA (A,B,C ,N)
C
C     ************************************************************************
C     MATRIX TRANSPONIERT * MATRIX  AT*B=C
C     ************************************************************************
C
      DIMENSION A(N,N),B(N,N),C(N,N)
      DO  1  I= 1,N
      DO  1  J= 1,N
      C(I,J)= 0.
      DO  1  K= 1,N
      C(I,J)= C(I,J) + A(K,I)*B(K,J)
    1 CONTINUE
C
      RETURN
      END
      SUBROUTINE  MAMAT (A,B,C ,N)
C
C     ************************************************************************
C     MATRIX * MATRIX TRANSPONIERT  A*BT=C
C     ************************************************************************
C
      DIMENSION A(N,N),B(N,N),C(N,N)
      DO  1  I= 1,N
      DO  1  J= 1,N
      C(I,J)= 0.
      DO  1  K= 1,N
      C(I,J)= C(I,J) + A(I,K)*B(J,K)
    1 CONTINUE
C
      RETURN
      END
      SUBROUTINE  MAVE (A,B,C,N)
C     ************************************************************************
C     MATRIX * VEKTOR A*B=C
C     ************************************************************************
      DIMENSION A(N,N),B(N),C(N)
      DO  1  I= 1,N
      C(I)= 0.
      DO  1  J= 1,N
```

```
      C(I)= C(I) + A(I,J)*B(J)
    1 CONTINUE
C
      RETURN
      END
      SUBROUTINE QTODCM(Q,TA)
      REAL*4 TA(3,3)
      REAL*8 Q(4),A(4,4)
      DO 23 K=1,4
   23 A(K,K)=Q(K)*Q(K)
      DO 24 K=1,3
      KK=K+1
      DO 24 L=KK,4
   24 A(K,L)=Q(K)*Q(L)
      TA(1,1)=SNGL( A(1,1)-A(2,2)-A(3,3)+A(4,4))
      TA(2,2)=SNGL(-A(1,1)+A(2,2)-A(3,3)+A(4,4))
      TA(3,3)=SNGL(-A(1,1)-A(2,2)+A(3,3)+A(4,4))
      TA(1,2)=SNGL(2.D0*(A(1,2)+A(3,4)))
      TA(1,3)=SNGL(2.D0*(A(1,3)-A(2,4)))
      TA(2,1)=SNGL(2.D0*(A(1,2)-A(3,4)))
      TA(2,3)=SNGL(2.D0*(A(2,3)+A(1,4)))
      TA(3,1)=SNGL(2.D0*(A(1,3)+A(2,4)))
      TA(3,2)=SNGL(2.D0*(A(2,3)-A(1,4)))
      RETURN
      END
      SUBROUTINE DCM(A,B,G,H)
      DIMENSION H(3,3)
      REAL*8 SA,CA,SB,CB,SG,CG
      SA=DSIN(DBLE(A))
      CA=DCOS(DBLE(A))
      SB=DSIN(DBLE(B))
      CB=DCOS(DBLE(B))
      SG=DSIN(DBLE(G))
      CG=DCOS(DBLE(G))
      H(1,1)=SNGL(CB*CG)
      H(1,2)=SNGL(SG)
      H(1,3)=SNGL(-SB*CG)
      H(2,1)=SNGL(-CA*CB*SG+SA*SB)
      H(2,2)=SNGL(CG*CA)
      H(2,3)=SNGL(CA*SB*SG+SA*CB)
      H(3,1)=SNGL(SA*CB*SG+CA*SB)
      H(3,2)=SNGL(-SA*CG)
      H(3,3)=SNGL(-SA*SB*SG+CA*CB)
      RETURN
      END
      SUBROUTINE TRANSP(A,B)
      DIMENSION A(3,3),B(3,3)
      DO 10 I=1,3
      DO 10 J=1,3
      B(I,J)=A(J,I)
   10 CONTINUE
      RETURN
      END
```

IPS TREESET Output

IPSREV7.INT

```
          TREETOPS REV 7  03/22/89

          SIM CONTROL

 1 SI   0 Title                                                IPS
 2 SI   0 Simulation stop time                                 100
 3 SI   0 Plot data interval                                   .25
 4 SI   0 Integration type (R,S or U)                          R
 5 SI   0 Step size (sec)                                      .01
 6 SI   0 Sandia integration absolute and relative error
 7 SI   0 Linearization option  (L,Z or N)                     N
 8 SI   0 Restart option        (Y/N)                          N
 9 SI   0 Contact force computation option     (Y/N)           N
10 SI   0 Constraint force computation option  (Y/N)           N
11 SI   0 Small angle speedup option (All,Bypass,First,Nth)    F
12 SI   0 Mass matrix speedup option (All,Bypass,First,Nth)    10
13 SI   0 Non-Linear  speedup option (All,Bypass,First,Nth)    A
14 SI   0 Constraint  speedup option (All,Bypass,First,Nth)    A
15 SI   0 Constraint stabilization option  (Y/N)              N
16 SI   0 Stabilization epsilon



          BODY


17 BO   1 Body ID number                                      1
18 BO   1 Type (Rigid,Flexible,NASTRAN)                       R
19 BO   1 Number of modes
20 BO   1 Modal calculation option (0, 1 or 2)
21 BO   1 Model Reduction Method (NO,MS,MC,CC,QM,CV)
22 BO   1 NASTRAN data file FORTRAN unit number (40-60)
23 BO   1 Number of augmented nodes (0 if none)
24 BO   1 Damping matrix option (NS,CD,HL,SD)
25 BO   1 Constant damping ratio
26 BO   1 Low frequency, High frequency ratios
27 BO   1 Mode ID number, damping ratio
28 BO   1 Conversion factors: Length,Mass,Force
29 BO   1 Inertia reference node (0=Bdy Ref Frm; 1=mass cen) 1
30 BO   1 Moments of inertia (kg-m2) Ixx,Iyy,Izz          13619E+7,.1041E+8,
                                                          .10854E+8
31 BO   1 Products of inertia (kg-m2) Ixy,Ixz,Iyz          5168,-414415,2543
32 BO   1 Mass (kg)                                        103025.2
33 BO   1 Number of Nodes                                  9
34 BO   1 Node ID, Node coord. (meters) x,y,z              1,0,0,0
35 BO   1 Node ID, Node coord. (meters) x,y,z             2,3.772,-.0773,.117
36 BO   1 Node ID, Node coord. (meters) x,y,z             3,14.371,.01,-1.923
37 BO   1 Node ID, Node coord. (meters) x,y,z               4,19.34,1.52,.61
```

```
38 BO   1 Node ID, Node coord. (meters) x,y,z                5,19.34,-1.52,.61
39 BO   1 Node ID, Node coord. (meters) x,y,z              6,-12.17,3.81,-2.16
40 BO   1 Node ID, Node coord. (meters) x,y,z              7,-12.17,-3.81,-2.16
41 BO   1 Node ID, Node coord. (meters) x,y,z               8,-12.17,3.0,-2.07
42 BO   1 Node ID, Node coord. (meters) x,y,z               9,-12.17,-3.0,-2.07
43 BO   1 Node ID, Node structual joint ID

44 BO   2 Body ID number                                    2
45 BO   2 Type (Rigid,Flexible,NASTRAN)                     R
46 BO   2 Number of modes
47 BO   2 Modal calculation option (0, 1 or 2)
48 BO   2 Model Reduction Method (NO,MS,MC,CC,QM,CV)
49 BO   2 NASTRAN data file FORTRAN unit number (40-60)
50 BO   2 Number of augmented nodes (0 if none)
51 BO   2 Damping matrix option (NS,CD,HL,SD)
52 BO   2 Constant damping ratio
53 BO   2 Low frequency, High frequency ratios
54 BO   2 Mode ID number, damping ratio
55 BO   2 Conversion factors: Length,Mass,Force
56 BO   2 Inertia reference node (0=Bdy Ref Frm; 1=mass cen) 1
57 BO   2 Moments of inertia (kg-m2) Ixx,Iyy,Izz            7544,20290,24311
58 BO   2 Products of inertia (kg-m2) Ixy,Ixz,Iyz         665.7,-661.6,-196.6
59 BO   2 Mass (kg)                                        4408.7
60 BO   2 Number of Nodes                                  3
61 BO   2 Node ID, Node coord. (meters) x,y,z               1,0,0,0
62 BO   2 Node ID, Node coord. (meters) x,y,z            2,1.113,.0958,-1.072
63 BO   2 Node ID, Node coord. (meters) x,y,z               3,.846,.576,-.267
64 BO   2 Node ID, Node structual joint ID

65 BO   3 Body ID number                                    3
66 BO   3 Type (Rigid,Flexible,NASTRAN)                     R
67 BO   3 Number of modes
68 BO   3 Modal calculation option (0, 1 or 2)
69 BO   3 Model Reduction Method (NO,MS,MC,CC,QM,CV)
70 BO   3 NASTRAN data file FORTRAN unit number (40-60)
71 BO   3 Number of augmented nodes (0 if none)
72 BO   3 Damping matrix option (NS,CD,HL,SD)
73 BO   3 Constant damping ratio
74 BO   3 Low frequency, High frequency ratios
75 BO   3 Mode ID number, damping ratio
76 BO   3 Conversion factors: Length,Mass,Force
77 BO   3 Inertia reference node (0=Bdy Ref Frm; 1=mass cen) 1
78 BO   3 Moments of inertia (kg-m2) Ixx,Iyy,Izz          5.4216,5.4216,1.3554
79 BO   3 Products of inertia (kg-m2) Ixy,Ixz,Iyz           0,-8.1324,0
80 BO   3 Mass (kg)                                        84.4
81 BO   3 Number of Nodes                                  3
82 BO   3 Node ID, Node coord. (meters) x,y,z               1,0,0,0
83 BO   3 Node ID, Node coord. (meters) x,y,z         2,-.00279,.00686,.33604
84 BO   3 Node ID, Node coord. (meters) x,y,z        3,-.00279,.00686,-.14656
85 BO   3 Node ID, Node structual joint ID

86 BO   4 Body ID number                                    4
87 BO   4 Type (Rigid,Flexible,NASTRAN)                     R
```

```
 88 BO   4 Number of modes
 89 BO   4 Modal calculation option (0, 1 or 2)
 90 BO   4 Model Reduction Method (NO,MS,MC,CC,QM,CV)
 91 BO   4 NASTRAN data file FORTRAN unit number (40-60)
 92 BO   4 Number of augmented nodes (0 if none)
 93 BO   4 Damping matrix option (NS,CD,HL,SD)
 94 BO   4 Constant damping ratio
 95 BO   4 Low frequency, High frequency ratios
 96 BO   4 Mode ID number, damping ratio
 97 BO   4 Conversion factors: Length,Mass,Force
 98 BO   4 Inertia reference node (0=Bdy Ref Frm; 1=mass cen) 1
 99 BO   4 Moments of inertia (kg-m2) Ixx,Iyy,Izz          8.1324,16.265,9.4878
100 BO   4 Products of inertia (kg-m2) Ixy,Ixz,Iyz           -1.3554,-28.463,0
101 BO   4 Mass (kg)                                        105.2
102 BO   4 Number of Nodes                                  3
103 BO   4 Node ID, Node coord. (meters) x,y,z              1,0,0,0
104 BO   4 Node ID, Node coord. (meters) x,y,z             2,.309,-.0102,-.292
105 BO   4 Node ID, Node coord. (meters) x,y,z             3,-.311,-.00102,.191
106 BO   4 Node ID, Node structual joint ID

107 BO   5 Body ID number                                  5
108 BO   5 Type (Rigid,Flexible,NASTRAN)                   R
109 BO   5 Number of modes
110 BO   5 Modal calculation option (0, 1 or 2)
111 BO   5 Model Reduction Method (NO,MS,MC,CC,QM,CV)
112 BO   5 NASTRAN data file FORTRAN unit number (40-60)
113 BO   5 Number of augmented nodes (0 if none)
114 BO   5 Damping matrix option (NS,CD,HL,SD)
115 BO   5 Constant damping ratio
116 BO   5 Low frequency, High frequency ratios
117 BO   5 Mode ID number, damping ratio
118 BO   5 Conversion factors: Length,Mass,Force
119 BO   5 Inertia reference node (0=Bdy Ref Frm; 1=mass cen) 1
120 BO   5 Moments of inertia (kg-m2) Ixx,Iyy,Izz            2496,4404,4127
121 BO   5 Products of inertia (kg-m2) Ixy,Ixz,Iyz          44.7,-116.6,146.4
122 BO   5 Mass (kg)                                       3284.1
123 BO   5 Number of Nodes                                 2
124 BO   5 Node ID, Node coord. (meters) x,y,z             1,0,0,0
125 BO   5 Node ID, Node coord. (meters) x,y,z            2,-1.436,.0254,.014
126 BO   5 Node ID, Node structual joint ID

           HINGE


127 HI   1 Hinge ID number                                 1
128 HI   1 Inboard Body ID, Outboard Body ID               0   1
129 HI   1 "p" Node ID, "q" Node ID                        0,1
130 HI   1 Number of rotation DOFs, Rotation option (F or G) 3,G
131 HI   1 L1 unit vector in inboard body coord. x,y,z     1,0,0
132 HI   1 L1 unit vector in outboard body coord. x,y,z    1,0,0
133 HI   1 L2 unit vector in inboard body coord. x,y,z
134 HI   1 L2 unit vector in outboard body coord. x,y,z
135 HI   1 L3 unit vector in inboard body coord. x,y,z     0,0,1
```

```
136 HI   1 L3 unit vector in outboard body coord. x,y,z       0,0,1
137 HI   1 Initial rotation angles (deg)                     0,0,0
138 HI   1 Initial rotation rates (deg/sec)                  0,0,0
139 HI   1 Rotation stiffness (newton-meters/rad)            0,0,0
140 HI   1 Rotation damping (newton-meters/rad/sec)          0,0,0
141 HI   1 Null torque angles (deg)                         0,0,0
142 HI   1 Number of translation DOFs                       3
143 HI   1 First translation unit vector   g1               1,0,0
144 HI   1 Second translation unit vector  g2               0,1,0
145 HI   1 Third translation unit vector   g3               0,0,1
146 HI   1 Initial translation (meters)                     0,0,0
147 HI   1 Initial translation velocity (meters/sec)        0,0,0
148 HI   1 Translation stiffness (newtons/meters)           0,0,0
149 HI   1 Translation damping (newtons/meter/sec)          0,0,0
150 HI   1 Null force translations                          0,0,0

151 HI   2 Hinge ID number                                  2
152 HI   2 Inboard Body ID, Outboard Body ID                1   2
153 HI   2 "p" Node ID, "q" Node ID                         2,1
154 HI   2  No of rotation DOFs, Hinge 1 rotation option(F/G) 0
155 HI   2 L1 unit vector in inboard body coord. x,y,z      1,0,0
156 HI   2 L1 unit vector in outboard body coord. x,y,z     1,0,0
157 HI   2 L2 unit vector in inboard body coord. x,y,z
158 HI   2 L2 unit vector in outboard body coord. x,y,z
159 HI   2 L3 unit vector in inboard body coord. x,y,z      0,0,1
160 HI   2 L3 unit vector in outboard body coord. x,y,z     0,0,1
161 HI   2 Initial rotation angles (deg)                    0,0,0
162 HI   2 Initial rotation rates (deg/sec)
163 HI   2 Rotation stiffness (newton-meters/rad)
164 HI   2 Rotation damping (newton-meters/rad/sec)
165 HI   2 Null torque angles (deg)
166 HI   2 Number of translation DOFs                       0
167 HI   2 First translation unit vector   g1               1,0,0
168 HI   2 Second translation unit vector  g2               0,1,0
169 HI   2 Third translation unit vector   g3               0,0,1
170 HI   2 Initial translation (meters)                     0,0,0
171 HI   2 Initial translation velocity (meters/sec)
172 HI   2 Translation stiffness (newtons/meters)
173 HI   2 Translation damping (newtons/meter/sec)
174 HI   2 Null force translations

175 HI   3 Hinge ID number                                  3
176 HI   3 Inboard Body ID, Outboard Body ID                2   3
177 HI   3 "p" Node ID, "q" Node ID                         2   3
178 HI   3  Number of rotation DOFs                         1
179 HI   3 L1 unit vector in inboard body coord. x,y,z      0,-1,0
180 HI   3 L1 unit vector in outboard body coord. x,y,z     0,1,0
181 HI   3 L2 unit vector in inboard body coord. x,y,z
182 HI   3 L2 unit vector in outboard body coord. x,y,z
183 HI   3 L3 unit vector in inboard body coord. x,y,z      1,0,0
184 HI   3 L3 unit vector in outboard body coord. x,y,z     -1,0,0
185 HI   3 Initial rotation angles (deg)                    120.3,0,0
186 HI   3 Initial rotation rates (deg/sec)                 0.
```

```
187 HI   3 Rotation stiffness (newton-meters/rad)              0
188 HI   3 Rotation damping (newton-meters/rad/sec)           0
189 HI   3 Null torque angles (deg)                           0
190 HI   3 Number of translation DOFs                         0
191 HI   3 First translation unit vector   g1                 1,0,0
192 HI   3 Second translation unit vector  g2                 0,1,0
193 HI   3 Third translation unit vector   g3                 0,0,1
194 HI   3 Initial translation (meters)                       0,0,0
195 HI   3 Initial translation velocity (meters/sec)
196 HI   3 Translation stiffness (newtons/meters)
197 HI   3 Translation damping (newtons/meter/sec)
198 HI   3 Null force translations

199 HI   4 Hinge ID number                                    4
200 HI   4 Inboard Body ID, Outboard Body ID                  3   4
201 HI   4 "p" Node ID, "q" Node ID                           2   3
202 HI   4  Number of rotation DOFs                           1
203 HI   4 L1 unit vector in inboard body coord. x,y,z        0,0,1
204 HI   4 L1 unit vector in outboard body coord. x,y,z       0,0,1
205 HI   4 L2 unit vector in inboard body coord. x,y,z
206 HI   4 L2 unit vector in outboard body coord. x,y,z
207 HI   4 L3 unit vector in inboard body coord. x,y,z        0,1,0
208 HI   4 L3 unit vector in outboard body coord. x,y,z       0,1,0
209 HI   4 Initial rotation angles (deg)                      19.7,0,0
210 HI   4 Initial rotation rates (deg/sec)                   0
211 HI   4 Rotation stiffness (newton-meters/rad)             0
212 HI   4 Rotation damping (newton-meters/rad/sec)           0
213 HI   4 Null torque angles (deg)                           0
214 HI   4 Number of translation DOFs                         0
215 HI   4 First translation unit vector   g1                 1,0,0
216 HI   4 Second translation unit vector  g2                 0,1,0
217 HI   4 Third translation unit vector   g3                 0,0,1
218 HI   4 Initial translation (meters)                       0,0,0
219 HI   4 Initial translation velocity (meters/sec)
220 HI   4 Translation stiffness (newtons/meters)
221 HI   4 Translation damping (newtons/meter/sec)
222 HI   4 Null force translations

223 HI   5 Hinge ID number                                    5
224 HI   5 Inboard Body ID, Outboard Body ID                  4   5
225 HI   5 "p" Node ID, "q" Node ID                           2   2
226 HI   5  Number of rotation DOFs                           1
227 HI   5 L1 unit vector in inboard body coord. x,y,z        1,0,0
228 HI   5 L1 unit vector in outboard body coord. x,y,z       1,0,0
229 HI   5 L2 unit vector in inboard body coord. x,y,z
230 HI   5 L2 unit vector in outboard body coord. x,y,z
231 HI   5 L3 unit vector in inboard body coord. x,y,z        0,0,1
232 HI   5 L3 unit vector in outboard body coord. x,y,z       0,0,1
233 HI   5 Initial rotation angles (deg)                      -.3,0,0
234 HI   5 Initial rotation rates (deg/sec)                   0
235 HI   5 Rotation stiffness (newton-meters/rad)             0
236 HI   5 Rotation damping (newton-meters/rad/sec)           0
237 HI   5 Null torque angles (deg)                           0
```

```
238 HI   5 Number of translation DOFs                          0
239 HI   5 First translation unit vector   g1                  1,0,0
240 HI   5 Second translation unit vector  g2                  0,1,0
241 HI   5 Third translation unit vector   g3                  0,0,1
242 HI   5 Initial translation (meters)                        0,0,0
243 HI   5 Initial translation velocity (meters/sec)
244 HI   5 Translation stiffness (newtons/meters)
245 HI   5 Translation damping (newtons/meter/sec)
246 HI   5 Null force translations

           SENSOR


247 SE   1 Sensor ID number                                    1
248 SE   1 Type (G,R,AN,V,P,AC,T,I,SU,ST,IM,P3,V3,CR,CT)      G
249 SE   1 Mounting point body ID, Mounting point node ID      5,1
250 SE   1 Second mounting point body ID, Second node ID
251 SE   1 Input axis unit vector (IA) x,y,z                   1,0,0
252 SE   1 Mounting point Hinge index, Axis index
253 SE   1 First focal plane unit vector  (Fp1) x,y,z
254 SE   1 Second focal plane unit vector (Fp2) x,y,z
255 SE   1 Sun/Star unit vector (Us) x,y,z
256 SE   1 Euler Angle Sequence (1-6)
257 SE   1 CMG ID number and Gimbal number

258 SE   2 Sensor ID number                                    2
259 SE   2 Type (G,R,AN,V,P,AC,T,I,SU,ST,IM,P3,V3,CR,CT)      G
260 SE   2 Mounting point body ID, Mounting point node ID      5,1
261 SE   2 Second mounting point body ID, Second node ID
262 SE   2 Input axis unit vector (IA) x,y,z                   0,1,0
263 SE   2 Mounting point Hinge index, Axis index
264 SE   2 First focal plane unit vector  (Fp1) x,y,z
265 SE   2 Second focal plane unit vector (Fp2) x,y,z
266 SE   2 Sun/Star unit vector (Us) x,y,z
267 SE   2 Euler Angle Sequence (1-6)
268 SE   2 CMG ID number and Gimbal number

269 SE   3 Sensor ID number                                    3
270 SE   3 Type (G,R,AN,V,P,AC,T,I,SU,ST,IM,P3,V3,CR,CT)      G
271 SE   3 Mounting point body ID, Mounting point node ID      5,1
272 SE   3 Second mounting point body ID, Second node ID
273 SE   3 Input axis unit vector (IA) x,y,z                   0,0,1
274 SE   3 Mounting point Hinge index, Axis index
275 SE   3 First focal plane unit vector  (Fp1) x,y,z
276 SE   3 Second focal plane unit vector (Fp2) x,y,z
277 SE   3 Sun/Star unit vector (Us) x,y,z
278 SE   3 Euler Angle Sequence (1-6)
279 SE   3 CMG ID number and Gimbal number

280 SE   4 Sensor ID number                                    4
281 SE   4 Type (G,R,AN,V,P,AC,T,I,SU,ST,IM,P3,V3,CR,CT)      AC
282 SE   4 Mounting point body ID, Mounting point node ID      2,3
283 SE   4 Second mounting point body ID, Second node ID
```

```
284 SE   4 Input axis unit vector (IA) x,y,z                    0,0,-1
285 SE   4 Mounting point Hinge index, Axis index
286 SE   4 First focal plane unit vector  (Fp1) x,y,z
287 SE   4 Second focal plane unit vector (Fp2) x,y,z
288 SE   4 Sun/Star unit vector (Us) x,y,z
289 SE   4 Euler Angle Sequence (1-6)
290 SE   4 CMG ID number and Gimbal number

291 SE   5 Sensor ID number                                     5
292 SE   5 Type (G,R,AN,V,P,AC,T,I,SU,ST,IM,P3,V3,CR,CT)       AC
293 SE   5 Mounting point body ID, Mounting point node ID      2,3
294 SE   5 Second mounting point body ID, Second node ID
295 SE   5 Input axis unit vector (IA) x,y,z                   -1,0,0
296 SE   5 Mounting point Hinge index, Axis index
297 SE   5 First focal plane unit vector  (Fp1) x,y,z
298 SE   5 Second focal plane unit vector (Fp2) x,y,z
299 SE   5 Sun/Star unit vector (Us) x,y,z
300 SE   5 Euler Angle Sequence (1-6)
301 SE   5 CMG ID number and Gimbal number

302 SE   6 Sensor ID number                                     6
303 SE   6 Type (G,R,AN,V,P,AC,T,I,SU,ST,IM,P3,V3,CR,CT)       AC
304 SE   6 Mounting point body ID, Mounting point node ID      2,3
305 SE   6 Second mounting point body ID, Second node ID
306 SE   6 Input axis unit vector (IA) x,y,z                   0,1,0
307 SE   6 Mounting point Hinge index, Axis index
308 SE   6 First focal plane unit vector  (Fp1) x,y,z
309 SE   6 Second focal plane unit vector (Fp2) x,y,z
310 SE   6 Sun/Star unit vector (Us) x,y,z
311 SE   6 Euler Angle Sequence (1-6)
312 SE   6 CMG ID number and Gimbal number

313 SE   7 Sensor ID number                                     7
314 SE   7 Type (G,R,AN,V,P,AC,T,I,SU,ST,IM,P3,V3,CR,CT)       G
315 SE   7 Mounting point body ID, Mounting point node ID      3,1
316 SE   7 Second mounting point body ID, Second node ID
317 SE   7 Input axis unit vector (IA) x,y,z                   0,1,0
318 SE   7 Mounting point Hinge index, Axis index
319 SE   7 First focal plane unit vector  (Fp1) x,y,z
320 SE   7 Second focal plane unit vector (Fp2) x,y,z
321 SE   7 Sun/Star unit vector (Us) x,y,z
322 SE   7 Euler Angle Sequence (1-6)
323 SE   7 CMG ID number and Gimbal number

324 SE   8 Sensor ID number                                     8
325 SE   8 Type (G,R,AN,V,P,AC,T,I,SU,ST,IM,P3,V3,CR,CT)       G
326 SE   8 Mounting point body ID, Mounting point node ID      4,1
327 SE   8 Second mounting point body ID, Second node ID
328 SE   8 Input axis unit vector (IA) x,y,z                   0,0,1
329 SE   8 Mounting point Hinge index, Axis index
330 SE   8 First focal plane unit vector  (Fp1) x,y,z
331 SE   8 Second focal plane unit vector (Fp2) x,y,z
332 SE   8 Sun/Star unit vector (Us) x,y,z
```

```
333 SE   8 Euler Angle Sequence (1-6)
334 SE   8 CMG ID number and Gimbal number

335 SE   9 Sensor ID number                                        9
336 SE   9 Type (G,R,AN,V,P,AC,T,I,SU,ST,IM,P3,V3,CR,CT)           R
337 SE   9 Mounting point body ID, Mounting point node ID
338 SE   9 Second mounting point body ID, Second node ID
339 SE   9 Input axis unit vector (IA) x,y,z
340 SE   9 Mounting point Hinge index, Axis index                  5,1
341 SE   9 First focal plane unit vector  (Fp1) x,y,z
342 SE   9 Second focal plane unit vector (Fp2) x,y,z
343 SE   9 Sun/Star unit vector (Us) x,y,z
344 SE   9 Euler Angle Sequence (1-6)
345 SE   9 CMG ID number and Gimbal number

346 SE  10 Sensor ID number                                       10
347 SE  10 Type (G,R,AN,V,P,AC,T,I,SU,ST,IM,P3,V3,CR,CT)          R
348 SE  10 Mounting point body ID, Mounting point node ID
349 SE  10 Second mounting point body ID, Second node ID
350 SE  10 Input axis unit vector (IA) x,y,z
351 SE  10 Mounting point Hinge index, Axis index                 3,1
352 SE  10 First focal plane unit vector  (Fp1) x,y,z
353 SE  10 Second focal plane unit vector (Fp2) x,y,z
354 SE  10 Sun/Star unit vector (Us) x,y,z
355 SE  10 Euler Angle Sequence (1-6)
356 SE  10 CMG ID number and Gimbal number

357 SE  11 Sensor ID number                                       11
358 SE  11 Type (G,R,AN,V,P,AC,T,I,SU,ST,IM,P3,V3,CR,CT)          R
359 SE  11 Mounting point body ID, Mounting point node ID
360 SE  11 Second mounting point body ID, Second node ID
361 SE  11 Input axis unit vector (IA) x,y,z
362 SE  11 Mounting point Hinge index, Axis index                 4,1
363 SE  11 First focal plane unit vector  (Fp1) x,y,z
364 SE  11 Second focal plane unit vector (Fp2) x,y,z
365 SE  11 Sun/Star unit vector (Us) x,y,z
366 SE  11 Euler Angle Sequence (1-6)
367 SE  11 CMG ID number and Gimbal number

368 SE  12 Sensor ID number                                       12
369 SE  12 Type (G,R,AN,V,P,AC,T,I,SU,ST,IM,P3,V3,CR,CT)          ST
370 SE  12 Mounting point body ID, Mounting point node ID         5,2
371 SE  12 Second mounting point body ID, Second node ID
372 SE  12 Input axis unit vector (IA) x,y,z
373 SE  12 Mounting point Hinge index, Axis index
374 SE  12 First focal plane unit vector  (Fp1) x,y,z             0,1,0
375 SE  12 Second focal plane unit vector (Fp2) x,y,z             0,0,1
376 SE  12 Sun/Star unit vector (Us) x,y,z                  .46984631,-.342020143,
                                                            -.813797681
377 SE  12 Euler Angle Sequence (1-6)
378 SE  12 CMG ID number and Gimbal number

379 SE  13 Sensor ID number                                       13
```

```
380 SE  13 Type (G,R,AN,V,P,AC,T,I,SU,ST,IM,P3,V3,CR,CT)      ST
381 SE  13 Mounting point body ID, Mounting point node ID     5,2
382 SE  13 Second mounting point body ID, Second node ID
383 SE  13 Input axis unit vector (IA) x,y,z
384 SE  13 Mounting point Hinge index, Axis index
385 SE  13 First focal plane unit vector  (Fp1) x,y,z     .20791169,.9781476,0.
386 SE  13 Second focal plane unit vector (Fp2) x,y,z         0,0,1
387 SE  13 Sun/Star unit vector (Us) x,y,z              .495134034,-.139173101,
                                                       -.857597304
388 SE  13 Euler Angle Sequence (1-6)
389 SE  13 CMG ID number and Gimbal number

390 SE  14 Sensor ID number                                    14
391 SE  14 Type (G,R,AN,V,P,AC,T,I,SU,ST,IM,P3,V3,CR,CT)      ST
392 SE  14 Mounting point body ID, Mounting point node ID     5,2
393 SE  14 Second mounting point body ID, Second node ID
394 SE  14 Input axis unit vector (IA) x,y,z
395 SE  14 Mounting point Hinge index, Axis index
396 SE  14 First focal plane unit vector  (Fp1) x,y,z     -.20791169,.9781476,0
397 SE  14 Second focal plane unit vector (Fp2) x,y,z         0,0,1
398 SE  14 Sun/Star unit vector (Us) x,y,z              .424024048,-.529919264,
                                                       -.734431194
399 SE  14 Euler Angle Sequence (1-6)
400 SE  14 CMG ID number and Gimbal number

401 SE  15 Sensor ID number                                    15
402 SE  15 Type (G,R,AN,V,P,AC,T,I,SU,ST,IM,P3,V3,CR,CT)      T
403 SE  15 Mounting point body ID, Mounting point node ID
404 SE  15 Second mounting point body ID, Second node ID
405 SE  15 Input axis unit vector (IA) x,y,z
406 SE  15 Mounting point Hinge index, Axis index             1,1
407 SE  15 First focal plane unit vector  (Fp1) x,y,z
408 SE  15 Second focal plane unit vector (Fp2) x,y,z
409 SE  15 Sun/Star unit vector (Us) x,y,z
410 SE  15 Euler Angle Sequence (1-6)
411 SE  15 CMG ID number and Gimbal number

412 SE  16 Sensor ID number                                    16
413 SE  16 Type (G,R,AN,V,P,AC,T,I,SU,ST,IM,P3,V3,CR,CT)      T
414 SE  16 Mounting point body ID, Mounting point node ID
415 SE  16 Second mounting point body ID, Second node ID
416 SE  16 Input axis unit vector (IA) x,y,z
417 SE  16 Mounting point Hinge index, Axis index             1,2
418 SE  16 First focal plane unit vector  (Fp1) x,y,z
419 SE  16 Second focal plane unit vector (Fp2) x,y,z
420 SE  16 Sun/Star unit vector (Us) x,y,z
421 SE  16 Euler Angle Sequence (1-6)
422 SE  16 CMG ID number and Gimbal number

423 SE  17 Sensor ID number                                    17
424 SE  17 Type (G,R,AN,V,P,AC,T,I,SU,ST,IM,P3,V3,CR,CT)      T
425 SE  17 Mounting point body ID, Mounting point node ID
426 SE  17 Second mounting point body ID, Second node ID
```

```
427 SE  17 Input axis unit vector (IA) x,y,z
428 SE  17 Mounting point Hinge index, Axis index            1,3
429 SE  17 First focal plane unit vector  (Fp1) x,y,z
430 SE  17 Second focal plane unit vector (Fp2) x,y,z
431 SE  17 Sun/Star unit vector (Us) x,y,z
432 SE  17 Euler Angle Sequence (1-6)
433 SE  17 CMG ID number and Gimbal number

434 SE  18 Sensor ID number                                  18
435 SE  18 Type (G,R,AN,V,P,AC,T,I,SU,ST,IM,P3,V3,CR,CT)     ST
436 SE  18 Mounting point body ID, Mounting point node ID    5,1
437 SE  18 Second mounting point body ID, Second node ID
438 SE  18 Input axis unit vector (IA) x,y,z
439 SE  18 Mounting point Hinge index, Axis index
440 SE  18 First focal plane unit vector  (Fp1) x,y,z       1,0,0
441 SE  18 Second focal plane unit vector (Fp2) x,y,z       0,1,0
442 SE  18 Sun/Star unit vector (Us) x,y,z                  .866025403,0,.5
443 SE  18 Euler Angle Sequence (1-6)
444 SE  18 CMG ID number and Gimbal number

           ACTR


445 AC   1 Actuator ID number                               1
446 AC   1 Type (J,H,MO,T,B,MA,SG,DG,W,L,M1-M7)             J
447 AC   1 Actuator location; Node or Hinge (N or H)        1,1
448 AC   1 Mounting point body ID number, node ID number    1,1
449 AC   1 Second mounting point body ID, second node ID
450 AC   1 Output axis unit vector x,y,z                    1,0,0
451 AC   1 Mounting point Hinge index, Axis index
452 AC   1 Rotor spin axis unit vector x,y,z
453 AC   1 Initial rotor momentum, H
454 AC   1 Outer gimbal - angle(deg),inertia,friction(D,S,N)
455 AC   1 Outer gimbal axis unit vector x,y,z
456 AC   1 Outer gim friction (Tfi,Tgfo,GAM) or (Tfi,M,D,Kf)
457 AC   1 Inner gimbal - angle(deg),inertia,friction(D,S,N)
458 AC   1 Inner gimbal axis unit vector x,y,z
459 AC   1 Inner gim friction (Tfi,Tgfo,GAM) or (Tfi,M,D,Kf)
460 AC   1 Initial length and rate, y(to) and ydot(to)
461 AC   1 Constants; K1 or wo, n or zeta, Kg, Jm
462 AC   1 Non-linearities; TLim, Tco, Dz

463 AC   2 Actuator ID number                               2
464 AC   2 Type (J,H,MO,T,B,MA,SG,DG,W,L,M1-M7)             J
465 AC   2 Actuator location; Node or Hinge (N or H)        1,1
466 AC   2 Mounting point body ID number, node ID number    1,1
467 AC   2 Second mounting point body ID, second node ID
468 AC   2 Output axis unit vector x,y,z                    0,1,0
469 AC   2 Mounting point Hinge index, Axis index
470 AC   2 Rotor spin axis unit vector x,y,z
471 AC   2 Initial rotor momentum, H
472 AC   2 Outer gimbal - angle(deg),inertia,friction(D,S,N)
473 AC   2 Outer gimbal axis unit vector x,y,z
```

```
474 AC   2 Outer gim friction (Tfi,Tgfo,GAM) or (Tfi,M,D,Kf)
475 AC   2 Inner gimbal - angle(deg),inertia,friction(D,S,N)
476 AC   2 Inner gimbal axis unit vector x,y,z
477 AC   2 Inner gim friction (Tfi,Tgfo,GAM) or (Tfi,M,D,Kf)
478 AC   2 Initial length and rate, y(to) and ydot(to)
479 AC   2 Constants; K1 or wo, n or zeta, Kg, Jm
480 AC   2 Non-linearities; TLim, Tco, Dz

481 AC   3 Actuator ID number                                  3
482 AC   3 Type (J,H,MO,T,B,MA,SG,DG,W,L,M1-M7)               J
483 AC   3 Actuator location; Node or Hinge (N or H)          1,1
484 AC   3 Mounting point body ID number, node ID number      1,1
485 AC   3 Second mounting point body ID, second node ID
486 AC   3 Output axis unit vector x,y,z                      0,0,1
487 AC   3 Mounting point Hinge index, Axis index
488 AC   3 Rotor spin axis unit vector x,y,z
489 AC   3 Initial rotor momentum, H
490 AC   3 Outer gimbal - angle(deg),inertia,friction(D,S,N)
491 AC   3 Outer gimbal axis unit vector x,y,z
492 AC   3 Outer gim friction (Tfi,Tgfo,GAM) or (Tfi,M,D,Kf)
493 AC   3 Inner gimbal - angle(deg),inertia,friction(D,S,N)
494 AC   3 Inner gimbal axis unit vector x,y,z
495 AC   3 Inner gim friction (Tfi,Tgfo,GAM) or (Tfi,M,D,Kf)
496 AC   3 Initial length and rate, y(to) and ydot(to)
497 AC   3 Constants; K1 or wo, n or zeta, Kg, Jm
498 AC   3 Non-linearities; TLim, Tco, Dz

499 AC   4 Actuator ID number                                  4
500 AC   4 Type (J,H,MO,T,B,MA,SG,DG,W,L,M1-M7)               MO
501 AC   4 Actuator location; Node or Hinge (N or H)          1,1
502 AC   4 Mounting point body ID number, node ID number      1,1
503 AC   4 Second mounting point body ID, second node ID
504 AC   4 Output axis unit vector x,y,z                      1,0,0
505 AC   4 Mounting point Hinge index, Axis index
506 AC   4 Rotor spin axis unit vector x,y,z
507 AC   4 Initial rotor momentum, H
508 AC   4 Outer gimbal - angle(deg),inertia,friction(D,S,N)
509 AC   4 Outer gimbal axis unit vector x,y,z
510 AC   4 Outer gim friction (Tfi,Tgfo,GAM) or (Tfi,M,D,Kf)
511 AC   4 Inner gimbal - angle(deg),inertia,friction(D,S,N)
512 AC   4 Inner gimbal axis unit vector x,y,z
513 AC   4 Inner gim friction (Tfi,Tgfo,GAM) or (Tfi,M,D,Kf)
514 AC   4 Initial length and rate, y(to) and ydot(to)
515 AC   4 Constants; K1 or wo, n or zeta, Kg, Jm
516 AC   4 Non-linearities; TLim, Tco, Dz

517 AC   5 Actuator ID number                                  5
518 AC   5 Type (J,H,MO,T,B,MA,SG,DG,W,L,M1-M7)               MO
519 AC   5 Actuator location; Node or Hinge (N or H)          1,1
520 AC   5 Mounting point body ID number, node ID number      1,1
521 AC   5 Second mounting point body ID, second node ID
522 AC   5 Output axis unit vector x,y,z                      0,1,0
523 AC   5 Mounting point Hinge index, Axis index
```

```
524 AC   5 Rotor spin axis unit vector x,y,z
525 AC   5 Initial rotor momentum, H
526 AC   5 Outer gimbal - angle(deg),inertia,friction(D,S,N)
527 AC   5 Outer gimbal axis unit vector x,y,z
528 AC   5 Outer gim friction (Tfi,Tgfo,GAM) or (Tfi,M,D,Kf)
529 AC   5 Inner gimbal - angle(deg),inertia,friction(D,S,N)
530 AC   5 Inner gimbal axis unit vector x,y,z
531 AC   5 Inner gim friction (Tfi,Tgfo,GAM) or (Tfi,M,D,Kf)
532 AC   5 Initial length and rate, y(to) and ydot(to)
533 AC   5 Constants; K1 or wo, n or zeta, Kg, Jm
534 AC   5 Non-linearities; TLim, Tco, Dz

535 AC   6 Actuator ID number                                  6
536 AC   6 Type (J,H,MO,T,B,MA,SG,DG,W,L,M1-M7)               MO
537 AC   6 Actuator location; Node or Hinge (N or H)          1,1
538 AC   6 Mounting point body ID number, node ID number      1,1
539 AC   6 Second mounting point body ID, second node ID
540 AC   6 Output axis unit vector x,y,z                      0,0,1
541 AC   6 Mounting point Hinge index, Axis index
542 AC   6 Rotor spin axis unit vector x,y,z
543 AC   6 Initial rotor momentum, H
544 AC   6 Outer gimbal - angle(deg),inertia,friction(D,S,N)
545 AC   6 Outer gimbal axis unit vector x,y,z
546 AC   6 Outer gim friction (Tfi,Tgfo,GAM) or (Tfi,M,D,Kf)
547 AC   6 Inner gimbal - angle(deg),inertia,friction(D,S,N)
548 AC   6 Inner gimbal axis unit vector x,y,z
549 AC   6 Inner gim friction (Tfi,Tgfo,GAM) or (Tfi,M,D,Kf)
550 AC   6 Initial length and rate, y(to) and ydot(to)
551 AC   6 Constants; K1 or wo, n or zeta, Kg, Jm
552 AC   6 Non-linearities; TLim, Tco, Dz

553 AC   7 Actuator ID number                                  7
554 AC   7 Type (J,H,MO,T,B,MA,SG,DG,W,L,M1-M7)               T
555 AC   7 Actuator location; Node or Hinge (N or H)
556 AC   7 Mounting point body ID number, node ID number
557 AC   7 Second mounting point body ID, second node ID
558 AC   7 Output axis unit vector x,y,z
559 AC   7 Mounting point Hinge index, Axis index            5,1
560 AC   7 Rotor spin axis unit vector x,y,z
561 AC   7 Initial rotor momentum, H
562 AC   7 Outer gimbal - angle(deg),inertia,friction(D,S,N)
563 AC   7 Outer gimbal axis unit vector x,y,z
564 AC   7 Outer gim friction (Tfi,Tgfo,GAM) or (Tfi,M,D,Kf)
565 AC   7 Inner gimbal - angle(deg),inertia,friction(D,S,N)
566 AC   7 Inner gimbal axis unit vector x,y,z
567 AC   7 Inner gim friction (Tfi,Tgfo,GAM) or (Tfi,M,D,Kf)
568 AC   7 Initial length and rate, y(to) and ydot(to)
569 AC   7 Constants; K1 or wo, n or zeta, Kg, Jm
570 AC   7 Non-linearities; TLim, Tco, Dz

571 AC   8 Actuator ID number                                  8
572 AC   8 Type (J,H,MO,T,B,MA,SG,DG,W,L,M1-M7)               T
573 AC   8 Actuator location; Node or Hinge (N or H)
```

```
574 AC   8 Mounting point body ID number, node ID number
575 AC   8 Second mounting point body ID, second node ID
576 AC   8 Output axis unit vector x,y,z
577 AC   8 Mounting point Hinge index, Axis index              3,1
578 AC   8 Rotor spin axis unit vector x,y,z
579 AC   8 Initial rotor momentum, H
580 AC   8 Outer gimbal - angle(deg),inertia,friction(D,S,N)
581 AC   8 Outer gimbal axis unit vector x,y,z
582 AC   8 Outer gim friction (Tfi,Tgfo,GAM) or (Tfi,M,D,Kf)
583 AC   8 Inner gimbal - angle(deg),inertia,friction(D,S,N)
584 AC   8 Inner gimbal axis unit vector x,y,z
585 AC   8 Inner gim friction (Tfi,Tgfo,GAM) or (Tfi,M,D,Kf)
586 AC   8 Initial length and rate, y(to) and ydot(to)
587 AC   8 Constants; K1 or wo, n or zeta, Kg, Jm
588 AC   8 Non-linearities; TLim, Tco, Dz

589 AC   9 Actuator ID number                                 9
590 AC   9 Type (J,H,MO,T,B,MA,SG,DG,W,L,M1-M7)              T
591 AC   9 Actuator location; Node or Hinge (N or H)
592 AC   9 Mounting point body ID number, node ID number
593 AC   9 Second mounting point body ID, second node ID
594 AC   9 Output axis unit vector x,y,z
595 AC   9 Mounting point Hinge index, Axis index              4,1
596 AC   9 Rotor spin axis unit vector x,y,z
597 AC   9 Initial rotor momentum, H
598 AC   9 Outer gimbal - angle(deg),inertia,friction(D,S,N)
599 AC   9 Outer gimbal axis unit vector x,y,z
600 AC   9 Outer gim friction (Tfi,Tgfo,GAM) or (Tfi,M,D,Kf)
601 AC   9 Inner gimbal - angle(deg),inertia,friction(D,S,N)
602 AC   9 Inner gimbal axis unit vector x,y,z
603 AC   9 Inner gim friction (Tfi,Tgfo,GAM) or (Tfi,M,D,Kf)
604 AC   9 Initial length and rate, y(to) and ydot(to)
605 AC   9 Constants; K1 or wo, n or zeta, Kg, Jm
606 AC   9 Non-linearities; TLim, Tco, Dz

607 AC  10 Actuator ID number                                10
608 AC  10 Type(J,H,MO,T,B,MA,SG,DG,W,L,M1-M7)              J
609 AC  10 Actuator location; Node or Hinge (N or H)
610 AC  10 Mounting point body ID number, node ID number     1,3
611 AC  10 Second mounting point body ID, second node ID
612 AC  10 Output axis unit vector x,y,z                     0,1,0
613 AC  10 Mounting point Hinge index, Axis index
614 AC  10 Rotor spin axis unit vector x,y,z
615 AC  10 Initial rotor momentum, H
616 AC  10 Outer gimbal- angle(deg),inertia,friction(D,S,B,N)
617 AC  10 Outer gimbal axis unit vector x,y,z
618 AC  10 Out gim fric (Tfi,Tgfo,GAM)/(Tfi,M,D,Kf)/(m,M,B,k)
619 AC  10 Inner gimbal- angle(deg),inertia,friction(D,S,B,N)
620 AC  10 Inner gimbal axis unit vector x,y,z
621 AC  10 In gim fric (Tfi,Tgfo,GAM)/(Tfi,M,D,Kf)/(m,M,B,k)
622 AC  10 Initial length and rate, y(to) and ydot(to)
623 AC  10 Constants; K1 or wo, n or zeta, Kg, Jm
624 AC  10 Non-linearities; TLim, Tco, Dz
```

```
625 AC  11 Actuator ID number                                    11
626 AC  11 Type(J,H,MO,T,B,MA,SG,DG,W,L,M1-M7)                   J
627 AC  11 Actuator location; Node or Hinge (N or H)
628 AC  11 Mounting point body ID number, node ID number         1,4
629 AC  11 Second mounting point body ID, second node ID
630 AC  11 Output axis unit vector x,y,z                         .033,-.696,-.717
631 AC  11 Mounting point Hinge index, Axis index
632 AC  11 Rotor spin axis unit vector x,y,z
633 AC  11 Initial rotor momentum, H
634 AC  11 Outer gimbal- angle(deg),inertia,friction(D,S,B,N)
635 AC  11 Outer gimbal axis unit vector x,y,z
636 AC  11 Out gim fric (Tfi,Tgfo,GAM)/(Tfi,M,D,Kf)/(m,M,B,k)
637 AC  11 Inner gimbal- angle(deg),inertia,friction(D,S,B,N)
638 AC  11 Inner gimbal axis unit vector x,y,z
639 AC  11 In gim fric (Tfi,Tgfo,GAM)/(Tfi,M,D,Kf)/(m,M,B,k)
640 AC  11 Initial length and rate, y(to) and ydot(to)
641 AC  11 Constants; K1 or wo, n or zeta, Kg, Jm
642 AC  11 Non-linearities; TLim, Tco, Dz

643 AC  12 Actuator ID number                                    12
644 AC  12 Type(J,H,MO,T,B,MA,SG,DG,W,L,M1-M7)                   J
645 AC  12 Actuator location; Node or Hinge (N or H)
646 AC  12 Mounting point body ID number, node ID number         1,5
647 AC  12 Second mounting point body ID, second node ID
648 AC  12 Output axis unit vector x,y,z                         -.033,.696,-.717
649 AC  12 Mounting point Hinge index, Axis index
650 AC  12 Rotor spin axis unit vector x,y,z
651 AC  12 Initial rotor momentum, H
652 AC  12 Outer gimbal- angle(deg),inertia,friction(D,S,B,N)
653 AC  12 Outer gimbal axis unit vector x,y,z
654 AC  12 Out gim fric (Tfi,Tgfo,GAM)/(Tfi,M,D,Kf)/(m,M,B,k)
655 AC  12 Inner gimbal- angle(deg),inertia,friction(D,S,B,N)
656 AC  12 Inner gimbal axis unit vector x,y,z
657 AC  12 In gim fric (Tfi,Tgfo,GAM)/(Tfi,M,D,Kf)/(m,M,B,k)
658 AC  12 Initial length and rate, y(to) and ydot(to)
659 AC  12 Constants; K1 or wo, n or zeta, Kg, Jm
660 AC  12 Non-linearities; TLim, Tco, Dz

661 AC  13 Actuator ID number                                    13
662 AC  13 Type(J,H,MO,T,B,MA,SG,DG,W,L,M1-M7)                   J
663 AC  13 Actuator location; Node or Hinge (N or H)
664 AC  13 Mounting point body ID number, node ID number         1,6
665 AC  13 Second mounting point body ID, second node ID
666 AC  13 Output axis unit vector x,y,z                         0,-1.0,-.025
667 AC  13 Mounting point Hinge index, Axis index
668 AC  13 Rotor spin axis unit vector x,y,z
669 AC  13 Initial rotor momentum, H
670 AC  13 Outer gimbal- angle(deg),inertia,friction(D,S,B,N)
671 AC  13 Outer gimbal axis unit vector x,y,z
672 AC  13 Out gim fric (Tfi,Tgfo,GAM)/(Tfi,M,D,Kf)/(m,M,B,k)
673 AC  13 Inner gimbal- angle(deg),inertia,friction(D,S,B,N)
674 AC  13 Inner gimbal axis unit vector x,y,z
```

```
675 AC  13 In gim fric (Tfi,Tgfo,GAM)/(Tfi,M,D,Kf)/(m,M,B,k)
676 AC  13 Initial length and rate, y(to) and ydot(to)
677 AC  13 Constants; K1 or wo, n or zeta, Kg, Jm
678 AC  13 Non-linearities; TLim, Tco, Dz

679 AC  14 Actuator ID number                                  14
680 AC  14 Type(J,H,MO,T,B,MA,SG,DG,W,L,M1-M7)                J
681 AC  14 Actuator location; Node or Hinge (N or H)
682 AC  14 Mounting point body ID number, node ID number      1,7
683 AC  14 Second mounting point body ID, second node ID
684 AC  14 Output axis unit vector x,y,z                      0,1.0,-.025
685 AC  14 Mounting point Hinge index, Axis index
686 AC  14 Rotor spin axis unit vector x,y,z
687 AC  14 Initial rotor momentum, H
688 AC  14 Outer gimbal- angle(deg),inertia,friction(D,S,B,N)
689 AC  14 Outer gimbal axis unit vector x,y,z
690 AC  14 Out gim fric (Tfi,Tgfo,GAM)/(Tfi,M,D,Kf)/(m,M,B,k)
691 AC  14 Inner gimbal- angle(deg),inertia,friction(D,S,B,N)
692 AC  14 Inner gimbal axis unit vector x,y,z
693 AC  14 In gim fric (Tfi,Tgfo,GAM)/(Tfi,M,D,Kf)/(m,M,B,k)
694 AC  14 Initial length and rate, y(to) and ydot(to)
695 AC  14 Constants; K1 or wo, n or zeta, Kg, Jm
696 AC  14 Non-linearities; TLim, Tco, Dz

697 AC  15 Actuator ID number                                  15
698 AC  15 Type(J,H,MO,T,B,MA,SG,DG,W,L,M1-M7)                J
699 AC  15 Actuator location; Node or Hinge (N or H)
700 AC  15 Mounting point body ID number, node ID number      1,8
701 AC  15 Second mounting point body ID, second node ID
702 AC  15 Output axis unit vector x,y,z                      .027,-.287,-.958
703 AC  15 Mounting point Hinge index, Axis index
704 AC  15 Rotor spin axis unit vector x,y,z
705 AC  15 Initial rotor momentum, H
706 AC  15 Outer gimbal- angle(deg),inertia,friction(D,S,B,N)
707 AC  15 Outer gimbal axis unit vector x,y,z
708 AC  15 Out gim fric (Tfi,Tgfo,GAM)/(Tfi,M,D,Kf)/(m,M,B,k)
709 AC  15 Inner gimbal- angle(deg),inertia,friction(D,S,B,N)
710 AC  15 Inner gimbal axis unit vector x,y,z
711 AC  15 In gim fric (Tfi,Tgfo,GAM)/(Tfi,M,D,Kf)/(m,M,B,k)
712 AC  15 Initial length and rate, y(to) and ydot(to)
713 AC  15 Constants; K1 or wo, n or zeta, Kg, Jm
714 AC  15 Non-linearities; TLim, Tco, Dz

715 AC  16 Actuator ID number                                  16
716 AC  16 Type(J,H,MO,T,B,MA,SG,DG,W,L,M1-M7)                J
717 AC  16 Actuator location; Node or Hinge (N or H)
718 AC  16 Mounting point body ID number, node ID number      1,9
719 AC  16 Second mounting point body ID, second node ID
720 AC  16 Output axis unit vector x,y,z                      .027,.287,-.958
721 AC  16 Mounting point Hinge index, Axis index
722 AC  16 Rotor spin axis unit vector x,y,z
723 AC  16 Initial rotor momentum, H
724 AC  16 Outer gimbal- angle(deg),inertia,friction(D,S,B,N)
```

```
725 AC  16 Outer gimbal axis unit vector x,y,z
726 AC  16 Out gim fric (Tfi,Tgfo,GAM)/(Tfi,M,D,Kf)/(m,M,B,k)
727 AC  16 Inner gimbal- angle(deg),inertia,friction(D,S,B,N)
728 AC  16 Inner gimbal axis unit vector x,y,z
729 AC  16 In gim fric (Tfi,Tgfo,GAM)/(Tfi,M,D,Kf)/(m,M,B,k)
730 AC  16 Initial length and rate, y(to) and ydot(to)
731 AC  16 Constants; K1 or wo, n or zeta, Kg, Jm
732 AC  16 Non-linearities; TLim, Tco, Dz

           CONTROLLER


733 CO   1 Controller ID number                              1
734 CO   1 Controller type (CB,CM,DB,DM,UC,UD)               CB
735 CO   1 Sample time (sec)
736 CO   1 Number of inputs, Number of outputs               4,9
737 CO   1 Number of states
738 CO   1 Output No., Input type (I,S,T), Input ID, Gain    1,I,1,0
739 CO   1 Output No., Input type (I,S,T), Input ID, Gain    2,I,1,10
740 CO   1 Output No., Input type (I,S,T), Input ID, Gain    3,I,1,0
741 CO   1 Output No., Input type (I,S,T), Input ID, Gain    4,I,1,0
742 CO   1 Output No., Input type (I,S,T), Input ID, Gain    5,I,1,0
743 CO   1 Output No., Input type (I,S,T), Input ID, Gain    6,I,1,0
744 CO   1 Output No., Input type (I,S,T), Input ID, Gain    7,T,33,.20627E+6
745 CO   1 Output No., Input type (I,S,T), Input ID, Gain    8,T,34,.20627E+6
746 CO   1 Output No., Input type (I,S,T), Input ID, Gain    9,T,35,.20627E+6

747 CO   2 Controller ID number                              2
748 CO   2 Controller type (CB,CM,DB,DM,UC,UD)               UD
749 CO   2 Sample time (sec)                                 .01
750 CO   2 Number of inputs, Number of outputs               21,9
751 CO   2 Number of states
752 CO   2 Output No., Input type (I,S,T), Input ID, Gain

           TRANSFER FUN


753 TR  33 Transfer function ID number                      33
754 TR  33 Controller ID number                             1
755 TR  33 Input type (I,S or T), Input ID number           I,2
756 TR  33 Order of numerator                               0
757 TR  33 Numerator coefficients (4 per line max)          1
758 TR  33 Order of denominator                             1
759 TR  33 Denominator coefficients (4 per line max)        0,1
760 TR  33 Transfer function gain                           1.0

761 TR  34 Transfer function ID number                      34
762 TR  34 Controller ID number                             1
763 TR  34 Input type (I,S or T), Input ID number           I,3
764 TR  34 Order of numerator                               0
765 TR  34 Numerator coefficients (4 per line max)          1
766 TR  34 Order of denominator                             1
767 TR  34 Denominator coefficients (4 per line max)        0,1
```

```
768 TR  34 Transfer function gain                          1.0

769 TR  35 Transfer function ID number                     35
770 TR  35 Controller ID number                            1
771 TR  35 Input type (I,S or T), Input ID number          I,4
772 TR  35 Order of numerator                              0
773 TR  35 Numerator coefficients (4 per line max)         1
774 TR  35 Order of denominator                            1
775 TR  35 Denominator coefficients (4 per line max)       0,1
776 TR  35 Transfer function gain                          1.0

           FUNCTION GEN


777 FU   1 Function generator ID number                    1
778 FU   1 Type (ST,RA,PU,SA,SI,US,NO,DO)                  PU
779 FU   1 Amplitude                                       1.0
780 FU   1 Slope
781 FU   1 Start time (sec)                                1.0
782 FU   1 Stop time (sec)                                 1.08
783 FU   1 Frequency (rad/sec)
784 FU   1 Phase shift (deg)
785 FU   1 Array location
786 FU   1 Mean
787 FU   1 Variance
788 FU   1 Pulse width (sec)
789 FU   1 Second pulse start time (sec)

790 FU   2 Function generator ID number                    2
791 FU   2 Type (ST,RA,PU,SA,SI,US,NO,DO)                  DO
792 FU   2 Amplitude                                       80.0
793 FU   2 Slope
794 FU   2 Start time (sec)                                90
795 FU   2 Stop time (sec)
796 FU   2 Frequency (rad/sec)
797 FU   2 Phase shift (deg)
798 FU   2 Array location
799 FU   2 Mean,Seed
800 FU   2 Variance
801 FU   2 Pulse width (sec)                               .5
802 FU   2 Second pulse start time (sec)                   92.1

803 FU   3 Function generator ID number                    3
804 FU   3 Type (ST,RA,PU,SA,SI,US,NO,DO)                  PU
805 FU   3 Amplitude                                       2.5
806 FU   3 Slope
807 FU   3 Start time (sec)                                30.0
808 FU   3 Stop time (sec)                                 130.0
809 FU   3 Frequency (rad/sec)
810 FU   3 Phase shift (deg)
811 FU   3 Array location
812 FU   3 Mean,Seed
813 FU   3 Variance
```

```
814 FU   3 Pulse width (sec)
815 FU   3 Second pulse start time (sec)

           INTERCONNECT


816 IN   1 Interconnect ID number                          1
817 IN   1 Source type(S,C or F),Source ID,Source row #    F,1 1
818 IN   1 Destination type(A or C),Dest ID,Dest row #     C,1,1
819 IN   1 Gain                                            0.0

820 IN   2 Interconnect ID number                          2
821 IN   2 Source type(S,C or F),Source ID,Source row #    C,1,1
822 IN   2 Destination type(A or C),Dest ID,Dest row #     A,1 1
823 IN   2 Gain                                            1

824 IN   3 Interconnect ID number                          3
825 IN   3 Source type(S,C or F),Source ID,Source row #    C,1,2
826 IN   3 Destination type(A or C),Dest ID,Dest row #     A,2 1
827 IN   3 Gain                                            1

828 IN   4 Interconnect ID number                          4
829 IN   4 Source type(S,C or F),Source ID,Source row #    C,1,3
830 IN   4 Destination type(A or C),Dest ID,Dest row #     A,3 1
831 IN   4 Gain                                            1

832 IN   5 Interconnect ID number                          5
833 IN   5 Source type(S,C or F),Source ID,Source row #    C,1,4
834 IN   5 Destination type(A or C),Dest ID,Dest row #     A,4 1
835 IN   5 Gain                                            1

836 IN   6 Interconnect ID number                          6
837 IN   6 Source type(S,C or F),Source ID,Source row #    C,1,5
838 IN   6 Destination type(A or C),Dest ID,Dest row #     A,5 1
839 IN   6 Gain                                            1

840 IN   7 Interconnect ID number                          7
841 IN   7 Source type(S,C or F),Source ID,Source row #    C,1,6
842 IN   7 Destination type(A or C),Dest ID,Dest row #     A,6 1
843 IN   7 Gain                                            1

844 IN   8 Interconnect ID number                          8
845 IN   8 Source type(S,C or F),Source ID,Source row #    S,1 1
846 IN   8 Destination type(A or C),Dest ID,Dest row #     C,1,2
847 IN   8 Gain                                            1

848 IN   9 Interconnect ID number                          9
849 IN   9 Source type(S,C or F),Source ID,Source row #    S,2 1
850 IN   9 Destination type(A or C),Dest ID,Dest row #     C,1,3
851 IN   9 Gain                                            1

852 IN  10 Interconnect ID number                          10
853 IN  10 Source type(S,C or F),Source ID,Source row #    S,3 1
```

```
854 IN  10 Destination type(A or C),Dest ID,Dest row #       C,1,4
855 IN  10 Gain                                              1

856 IN  14 Interconnect ID number                            14
857 IN  14 Source type(S,C or F),Source ID,Source row #      S,1 1
858 IN  14 Destination type(A or C),Dest ID,Dest row #       C,2,1
859 IN  14 Gain                                              1.0

860 IN  15 Interconnect ID number                            15
861 IN  15 Source type(S,C or F),Source ID,Source row #      S,2 1
862 IN  15 Destination type(A or C),Dest ID,Dest row #       C,2,2
863 IN  15 Gain                                              1.0

864 IN  16 Interconnect ID number                            16
865 IN  16 Source type(S,C or F),Source ID,Source row #      S,3 1
866 IN  16 Destination type(A or C),Dest ID,Dest row #       C,2,3
867 IN  16 Gain                                              1.0

868 IN  17 Interconnect ID number                            17
869 IN  17 Source type(S,C or F),Source ID,Source row #      S,4 1
870 IN  17 Destination type(A or C),Dest ID,Dest row #       C,2,4
871 IN  17 Gain                                              1.0

872 IN  18 Interconnect ID number                            18
873 IN  18 Source type(S,C or F),Source ID,Source row #      S,5 1
874 IN  18 Destination type(A or C),Dest ID,Dest row #       C,2,5
875 IN  18 Gain                                              1.0

876 IN  19 Interconnect ID number                            19
877 IN  19 Source type(S,C or F),Source ID,Source row #      S,6 1
878 IN  19 Destination type(A or C),Dest ID,Dest row #       C,2,6
879 IN  19 Gain                                              1.0

880 IN  20 Interconnect ID number                            20
881 IN  20 Source type(S,C or F),Source ID,Source row #      C,2,1
882 IN  20 Destination type(A or C),Dest ID,Dest row #       A,7,1
883 IN  20 Gain                                              1

884 IN  21 Interconnect ID number                            21
885 IN  21 Source type(S,C or F),Source ID,Source row #      C,2,2
886 IN  21 Destination type(A or C),Dest ID,Dest row #       A,8,1
887 IN  21 Gain                                              1

888 IN  22 Interconnect ID number                            22
889 IN  22 Source type(S,C or F),Source ID,Source row #      C,2,3
890 IN  22 Destination type(A or C),Dest ID,Dest row #       A,9,1
891 IN  22 Gain                                              1

892 IN  23 Interconnect ID number                            23
893 IN  23 Source type(S,C, or F),Source ID,Source row #     S,9,1
894 IN  23 Destination type(A or C),Dest ID,Dest row #       C,2,7
895 IN  23 Gain                                              1.0
```

```
896 IN  24 Interconnect ID number                          24
897 IN  24 Source type(S,C, or F),Source ID,Source row #   S,10,1
898 IN  24 Destination type(A or C),Dest ID,Dest row #     C,2,8
899 IN  24 Gain                                           1.0

900 IN  25 Interconnect ID number                          25
901 IN  25 Source type(S,C, or F),Source ID,Source row #   S,11,1
902 IN  25 Destination type(A or C),Dest ID,Dest row #     C,2,9
903 IN  25 Gain                                           1.0

904 IN  26 Interconnect ID number                          26
905 IN  26 Source type(S,C, or F),Source ID,Source row #   S,12,1
906 IN  26 Destination type(A or C),Dest ID,Dest row #     C,2,10
907 IN  26 Gain                                           1.0

908 IN  27 Interconnect ID number                          27
909 IN  27 Source type(S,C, or F),Source ID,Source row #   S,12,2
910 IN  27 Destination type(A or C),Dest ID,Dest row #     C,2,11
911 IN  27 Gain                                           1.0

912 IN  28 Interconnect ID number                          28
913 IN  28 Source type(S,C, or F),Source ID,Source row #   S,13,1
914 IN  28 Destination type(A or C),Dest ID,Dest row #     C,2,12
915 IN  28 Gain                                           1.0

916 IN  29 Interconnect ID number                          29
917 IN  29 Source type(S,C, or F),Source ID,Source row #   S,13,2
918 IN  29 Destination type(A or C),Dest ID,Dest row #     C,2,13
919 IN  29 Gain                                           1.0

920 IN  30 Interconnect ID number                          30
921 IN  30 Source type(S,C, or F),Source ID,Source row #   S,14,1
922 IN  30 Destination type(A or C),Dest ID,Dest row #     C,2,14
923 IN  30 Gain                                           0.

924 IN  31 Interconnect ID number                          31
925 IN  31 Source type(S,C, or F),Source ID,Source row #   S,14,2
926 IN  31 Destination type(A or C),Dest ID,Dest row #     C,2,15
927 IN  31 Gain                                           0.0

928 IN  32 Interconnect ID number                          32
929 IN  32 Source type(S,C, or F),Source ID,Source row #   F,2,1
930 IN  32 Destination type(A or C),Dest ID,Dest row #     A,10,1
931 IN  32 Gain                                           0.0

932 IN  33 Interconnect ID number                          33
933 IN  33 Source type(S,C, or F),Source ID,Source row #   S,15,1
934 IN  33 Destination type(A or C),Dest ID,Dest row #     C,2,16
935 IN  33 Gain                                           1.0

936 IN  34 Interconnect ID number                          34
937 IN  34 Source type(S,C, or F),Source ID,Source row #   S,16,1
938 IN  34 Destination type(A or C),Dest ID,Dest row #     C,2,17
```

```
939 IN  34 Gain                                                1.0

940 IN  35 Interconnect ID number                              35
941 IN  35 Source type(S,C, or F),Source ID,Source row #       S,17,1
942 IN  35 Destination type(A or C),Dest ID,Dest row #         C,2,18
943 IN  35 Gain                                                1.0

944 IN  36 Interconnect ID number                              36
945 IN  36 Source type(S,C, or F),Source ID,Source row #       C,2,4
946 IN  36 Destination type(A or C),Dest ID,Dest row #         A,11,1
947 IN  36 Gain                                                1.0

948 IN  37 Interconnect ID number                              37
949 IN  37 Source type(S,C, or F),Source ID,Source row #       C,2,5
950 IN  37 Destination type(A or C),Dest ID,Dest row #         A,12,1
951 IN  37 Gain                                                1.0

952 IN  38 Interconnect ID number                              38
953 IN  38 Source type(S,C, or F),Source ID,Source row #       C,2,6
954 IN  38 Destination type(A or C),Dest ID,Dest row #         A,13,1
955 IN  38 Gain                                                1.0

956 IN  39 Interconnect ID number                              39
957 IN  39 Source type(S,C, or F),Source ID,Source row #       C,2,7
958 IN  39 Destination type(A or C),Dest ID,Dest row #         A,14,1
959 IN  39 Gain                                                1.0

960 IN  40 Interconnect ID number                              40
961 IN  40 Source type(S,C, or F),Source ID,Source row #       C,2,8
962 IN  40 Destination type(A or C),Dest ID,Dest row #         A,15,1
963 IN  40 Gain                                                1.0

964 IN  41 Interconnect ID number                              41
965 IN  41 Source type(S,C, or F),Source ID,Source row #       C,2,9
966 IN  41 Destination type(A or C),Dest ID,Dest row #         A,16,1
967 IN  41 Gain                                                1.0

968 IN  42 Interconnect ID number                              42
969 IN  42 Source type(S,C, or F),Source ID,Source row #       F,3,1
970 IN  42 Destination type(A or C),Dest ID,Dest row #         C,2,19
971 IN  42 Gain                                                0.0

972 IN  43 Interconnect ID number                              43
973 IN  43 Source type(S,C, or F),Source ID,Source row #       F,3,1
974 IN  43 Destination type(A or C),Dest ID,Dest row #         C,2,20
975 IN  43 Gain                                                0.0

976 IN  44 Interconnect ID number                              44
977 IN  44 Source type(S,C, or F),Source ID,Source row #       F,3,1
978 IN  44 Destination type(A or C),Dest ID,Dest row #         C,2,21
979 IN  44 Gain                                                0.0
```

Fast Loop Controller Parameters

FBCOEF.DAT

```
17958.,13318.,-5905.,25208.
2808.,17286.,22439.,2515.,8.

17958.,13318.,-5905.,25208.
2808.,17286.,22439.,2515.,8.

17958.,13318.,-5905.,25208.
2808.,17286.,22439.,2515.,8.

6977.,26797.,-8976.,11144.
5878.,29268.,21256.,-28637.,30300.

6977.,26797.,-8976.,11144.
5878.,29268.,21256.,-28637.,30300.

4447.,32440.,-7917.,7196.
5505.,32760.,15561.,-10102.,22781.

6767.,30801.,-8306.,4875.
7783.,32132.,17735.,-24945.,30548.

6767.,27258.,-9386.,4875.
6887.,32132.,15692.,5649.,15013.

6767.,27258.,-9386.,4875.
6887.,32132.,15692.,5649.,15013.

-7561.,31706.,16029.,20179.

-7561.,31697.,16034.,20180.

2.0     0.02    0.0023

1.795   0.02    0.0023

1.795   0.02    0.0023

1.666   1.0     0.2304

1.0     1.0     0.3021

1.0     1.0     0.3021

3.45    4.5     4.5

0.33    0.4     0.4
```

FFCOEF.DAT

```
27129.,5655.,-4259.,26945.
810.,9456.,31627.,480.,15.

27129.,5655.,-4259.,26945.
810.,9456.,31627.,480.,15.

27129.,5655.,-4259.,26945.
810.,9456.,31627.,480.,15.

15845.,14483.,-14315.,10722.
5832.,27171.,22952.,-8323.,17518.

16697.,13979.,-11696.,16869.
4270.,22488.,25536.,-12990.,19124.

15845.,14483.,-14315.,10722.
5832.,27171.,22827.,-9985.,18705.

15267.,15635.,-14975.,7386.
6967.,28582.,22679.,4377.,9356.

16697.,12570.,-13007.,16869.
4338.,25411.,24624.,-438.,10307.

15845.,13053.,-15883.,10722.
5934.,30677.,23477.,3421.,8770.

12211.,23117.,-14845.,-5580.
11685.,25867.,21123.,14539.,8642.

15845.,13285.,-15606.,10722.
6040.,30677.,23626.,8641.,4442.

12211.,23081.,-14867.,-5580.
11667.,25867.,21156.,14539.,8644.
```

Command File to Link IPS Object Modules

XUSDC.COM

```
$LINK USDC,ADFUTIL,[-]QUATRN,[-]ADFS,[-]GAINS,[-]OBST,[-]NOISE,
SYS$USER3:[RAKOCZY]OUTPUT1,SYS$USER3:[RAKOCZY]TREETOPS
```

IPS TREETOPS Plot File Discription

IPSREV7.XRF

```
      TREETOPS Rev 4  4/17/88

                              IPS

   NBODY=   5   NSEN=  18  NACT=  16

   IOFF:    1    4   22   47   69   85    0    0    0    0

   KOFF:    0    9    0    0    0

   NOFF:    0    4    0    0    0

   LENGTH=   237


 NBODY=    5  NSEN=   18  NACT=   16  NFUN=    3  NCONT=    2  NUTOT=   25
 NRTOT=   18

  OUTPUT FILE REFENCE TABLE:

   1     TIME
   2     RSG            1
   3     RSG            2
   4     RSG            3
   5     RC            1
   6     RC            2
   7     RC            3
   8     RC            4
   9     RC            5
  10     RC            6
```

11	RC	7
12	RC	8
13	RC	9
14	RC	10
15	RC	11
16	RC	12
17	RC	13
18	RC	14
19	RC	15
20	RC	16
21	RC	17
22	RC	18
23	UC	1
24	UC	2
25	UC	3
26	UC	4
27	UC	5
28	UC	6
29	UC	7
30	UC	8
31	UC	9
32	UC	10
33	UC	11
34	UC	12
35	UC	13
36	UC	14
37	UC	15
38	UC	16
39	UC	17
40	UC	18
41	UC	19
42	UC	20
43	UC	21
44	UC	22
45	UC	23
46	UC	24
47	UC	25
48	RP	1
49	RP	2
50	RP	3
51	RP	4
52	RP	5
53	RP	6
54	RP	7
55	RP	8
56	RP	9
57	RP	10
58	RP	11
59	RP	12
60	RP	13
61	RP	14
62	RP	15
63	RP	16

64	RP		17					
65	RP		18					
66	RP		19					
67	RP		20					
68	RP		21					
69	RP		22					
70	UP		1					
71	UP		2					
72	UP		3					
73	UP		4					
74	UP		5					
75	UP		6					
76	UP		7					
77	UP		8					
78	UP		9					
79	UP		10					
80	UP		11					
81	UP		12					
82	UP		13					
83	UP		14					
84	UP		15					
85	UP		16					
86	THETA	HINGE	1	AXIS	1			
87	THETA	HINGE	1	AXIS	2			
88	THETA	HINGE	1	AXIS	3			
89	THDOT	HINGE	1	AXIS	1			
90	THDOT	HINGE	1	AXIS	2			
91	THDOT	HINGE	1	AXIS	3			
92	DDJJBJ	BODY	1	NODE	1	AXIS	1	
93	DDJJBJ	BODY	1	NODE	1	AXIS	2	
94	DDJJBJ	BODY	1	NODE	1	AXIS	3	
95	DDJJBJ	BODY	1	NODE	2	AXIS	1	
96	DDJJBJ	BODY	1	NODE	2	AXIS	2	
97	DDJJBJ	BODY	1	NODE	2	AXIS	3	
98	DDJJBJ	BODY	1	NODE	3	AXIS	1	
99	DDJJBJ	BODY	1	NODE	3	AXIS	2	
100	DDJJBJ	BODY	1	NODE	3	AXIS	3	
101	DDJJBJ	BODY	1	NODE	4	AXIS	1	
102	DDJJBJ	BODY	1	NODE	4	AXIS	2	
103	DDJJBJ	BODY	1	NODE	4	AXIS	3	
104	DDJJBJ	BODY	1	NODE	5	AXIS	1	
105	DDJJBJ	BODY	1	NODE	5	AXIS	2	
106	DDJJBJ	BODY	1	NODE	5	AXIS	3	
107	DDJJBJ	BODY	1	NODE	6	AXIS	1	
108	DDJJBJ	BODY	1	NODE	6	AXIS	2	
109	DDJJBJ	BODY	1	NODE	6	AXIS	3	
110	DDJJBJ	BODY	1	NODE	7	AXIS	1	
111	DDJJBJ	BODY	1	NODE	7	AXIS	2	
112	DDJJBJ	BODY	1	NODE	7	AXIS	3	
113	DDJJBJ	BODY	1	NODE	8	AXIS	1	
114	DDJJBJ	BODY	1	NODE	8	AXIS	2	
115	DDJJBJ	BODY	1	NODE	8	AXIS	3	
116	DDJJBJ	BODY	1	NODE	9	AXIS	1	

117	DDJJBJ	BODY	1	NODE	9	AXIS	2
118	DDJJBJ	BODY	1	NODE	9	AXIS	3
119	WJI	BODY	1	AXIS	1		
120	WJI	BODY	1	AXIS	2		
121	WJI	BODY	1	AXIS	3		
122	YJI	HINGE	1	AXIS	1		
123	YJI	HINGE	1	AXIS	2		
124	YJI	HINGE	1	AXIS	3		
125	YDOTJI	HINGE	1	AXIS	1		
126	YDOTJI	HINGE	1	AXIS	2		
127	YDOTJI	HINGE	1	AXIS	3		
128	QBODY	BODY	1	PARA	1		
129	QBODY	BODY	1	PARA	2		
130	QBODY	BODY	1	PARA	3		
131	QBODY	BODY	1	PARA	4		
132	RHJ	BODY	1	AXIS	1		
133	RHJ	BODY	1	AXIS	2		
134	RHJ	BODY	1	AXIS	3		
135	DDJJBJ	BODY	2	NODE	1	AXIS	1
136	DDJJBJ	BODY	2	NODE	1	AXIS	2
137	DDJJBJ	BODY	2	NODE	1	AXIS	3
138	DDJJBJ	BODY	2	NODE	2	AXIS	1
139	DDJJBJ	BODY	2	NODE	2	AXIS	2
140	DDJJBJ	BODY	2	NODE	2	AXIS	3
141	DDJJBJ	BODY	2	NODE	3	AXIS	1
142	DDJJBJ	BODY	2	NODE	3	AXIS	2
143	DDJJBJ	BODY	2	NODE	3	AXIS	3
144	WJI	BODY	2	AXIS	1		
145	WJI	BODY	2	AXIS	2		
146	WJI	BODY	2	AXIS	3		
147	YJI	HINGE	2	AXIS	1		
148	YJI	HINGE	2	AXIS	2		
149	YJI	HINGE	2	AXIS	3		
150	YDOTJI	HINGE	2	AXIS	1		
151	YDOTJI	HINGE	2	AXIS	2		
152	YDOTJI	HINGE	2	AXIS	3		
153	QBODY	BODY	2	PARA	1		
154	QBODY	BODY	2	PARA	2		
155	QBODY	BODY	2	PARA	3		
156	QBODY	BODY	2	PARA	4		
157	RHJ	BODY	2	AXIS	1		
158	RHJ	BODY	2	AXIS	2		
159	RHJ	BODY	2	AXIS	3		
160	THETA	HINGE	3	AXIS	1		
161	THDOT	HINGE	3	AXIS	1		
162	DDJJBJ	BODY	3	NODE	1	AXIS	1
163	DDJJBJ	BODY	3	NODE	1	AXIS	2
164	DDJJBJ	BODY	3	NODE	1	AXIS	3
165	DDJJBJ	BODY	3	NODE	2	AXIS	1
166	DDJJBJ	BODY	3	NODE	2	AXIS	2
167	DDJJBJ	BODY	3	NODE	2	AXIS	3
168	DDJJBJ	BODY	3	NODE	3	AXIS	1
169	DDJJBJ	BODY	3	NODE	3	AXIS	2

170	DDJJBJ	BODY	3	NODE	3	AXIS	3
171	WJI	BODY	3	AXIS	1		
172	WJI	BODY	3	AXIS	2		
173	WJI	BODY	3	AXIS	3		
174	YJI	HINGE	3	AXIS	1		
175	YJI	HINGE	3	AXIS	2		
176	YJI	HINGE	3	AXIS	3		
177	YDOTJI	HINGE	3	AXIS	1		
178	YDOTJI	HINGE	3	AXIS	2		
179	YDOTJI	HINGE	3	AXIS	3		
180	QBODY	BODY	3	PARA	1		
181	QBODY	BODY	3	PARA	2		
182	QBODY	BODY	3	PARA	3		
183	QBODY	BODY	3	PARA	4		
184	RHJ	BODY	3	AXIS	1		
185	RHJ	BODY	3	AXIS	2		
186	RHJ	BODY	3	AXIS	3		
187	THETA	HINGE	4	AXIS	1		
188	THDOT	HINGE	4	AXIS	1		
189	DDJJBJ	BODY	4	NODE	1	AXIS	1
190	DDJJBJ	BODY	4	NODE	1	AXIS	2
191	DDJJBJ	BODY	4	NODE	1	AXIS	3
192	DDJJBJ	BODY	4	NODE	2	AXIS	1
193	DDJJBJ	BODY	4	NODE	2	AXIS	2
194	DDJJBJ	BODY	4	NODE	2	AXIS	3
195	DDJJBJ	BODY	4	NODE	3	AXIS	1
196	DDJJBJ	BODY	4	NODE	3	AXIS	2
197	DDJJBJ	BODY	4	NODE	3	AXIS	3
198	WJI	BODY	4	AXIS	1		
199	WJI	BODY	4	AXIS	2		
200	WJI	BODY	4	AXIS	3		
201	YJI	HINGE	4	AXIS	1		
202	YJI	HINGE	4	AXIS	2		
203	YJI	HINGE	4	AXIS	3		
204	YDOTJI	HINGE	4	AXIS	1		
205	YDOTJI	HINGE	4	AXIS	2		
206	YDOTJI	HINGE	4	AXIS	3		
207	QBODY	BODY	4	PARA	1		
208	QBODY	BODY	4	PARA	2		
209	QBODY	BODY	4	PARA	3		
210	QBODY	BODY	4	PARA	4		
211	RHJ	BODY	4	AXIS	1		
212	RHJ	BODY	4	AXIS	2		
213	RHJ	BODY	4	AXIS	3		
214	THETA	HINGE	5	AXIS	1		
215	THDOT	HINGE	5	AXIS	1		
216	DDJJBJ	BODY	5	NODE	1	AXIS	1
217	DDJJBJ	BODY	5	NODE	1	AXIS	2
218	DDJJBJ	BODY	5	NODE	1	AXIS	3
219	DDJJBJ	BODY	5	NODE	2	AXIS	1
220	DDJJBJ	BODY	5	NODE	2	AXIS	2
221	DDJJBJ	BODY	5	NODE	2	AXIS	3
222	WJI	BODY	5	AXIS	1		

```
223     WJI     BODY    5     AXIS    2
224     WJI     BODY    5     AXIS    3
225     YJI     HINGE   5     AXIS    1
226     YJI     HINGE   5     AXIS    2
227     YJI     HINGE   5     AXIS    3
228     YDOTJI  HINGE   5     AXIS    1
229     YDOTJI  HINGE   5     AXIS    2
230     YDOTJI  HINGE   5     AXIS    3
231     QBODY   BODY    5     PARA    1
232     QBODY   BODY    5     PARA    2
233     QBODY   BODY    5     PARA    3
234     QBODY   BODY    5     PARA    4
235     RHJ     BODY    5     AXIS    1
236     RHJ     BODY    5     AXIS    2
237     RHJ     BODY    5     AXIS    3
 CONTACT FORCES WILL NOT BE COMPUTED
 CONSTRAINT FORCES WILL NOT BE COMPUTED
```

APPENDIX C

Linear Kalman Filter Listing and Data

PRECEDING PAGE BLANK NOT FILMED

IPS KF Batch Command File

TREEBATCH.COM

```
$ JOB
$ SET DEF [ROBINSON.TREETOPS]
$    ASSIGN/NOLOG IPSREV7.INT FOR001
$    ASSIGN/NOLOG IPSREV7.OUT FOR009
$    ASSIGN/NOLOG IPSREV7.RES FOR003
$    ASSIGN/NOLOG IPSREV7.MAT FOR004
$    ASSIGN/NOLOG IPSREV7.LOG FOR006
$    ASSIGN/NOLOG IPSREV7.MDK FOR019
$    ASSIGN/NOLOG IPSREV7.FLX FOR011
$    ASSIGN/NOLOG IPSREV7.FLN FOR013
$    ASSIGN/NOLOG IPSREV7.AUX FOR039
$    ASSIGN/NOLOG IPSREV7.XRF FOR012
$    ASSIGN/NOLOG TCATEMP.DAT  FOR025
$    ASS ASTP FOR015
$    ASS MSUBCO FOR016
$    ASS IPSL   FOR017
$    ASS PLP  FOR018
$    ASS DEFAULT.DFF FOR085
$    RUN [ROBINSON.TREETOPS.FILES]LCON.EXE
$    DEASSIGN FOR001
$    DEASSIGN FOR009
$    DEASSIGN FOR003
$    DEASSIGN FOR004
$    DEASSIGN FOR006
$    DEASSIGN FOR019
$    DEASSIGN FOR011
$    DEASSIGN FOR013
$    DEASSIGN FOR039
$    DEASSIGN FOR012
$    DEASSIGN FOR025
$    DEASSIGN FOR085
$    RENAME TCATEMP.DAT IPSREV7.MAT
$    DIR/DATE/SIZE IPSREV7.OUT
$ COPY IPSREV7.OUT IPSREV7.PLT
```

ADFR.FOR

```
      SUBROUTINE LADF
     DIMENSION PO(6),RNI(6),QNI(6),X(6),Z(6),HXM(6)
    1,ZHX(6),GZHX(6),DUM1(100),HPHTRO(6,6),IYO(6)
     DIMENSION P(6,6),QN(6,6),RN(6,6),H(6,6),HT(6,6),PHT(6,6),HPHT(6,6)
    *,HPHTR(6,6),HPHTRI(6,6),G(6,6),PHI(4,3),GH(6,6),IGH(6,6),PP(6,6)
    *,PHIT(6,6),PPPT(6,6),ID(6,6),PH(6,6)
     DIMENSION QOM(4,4),QU(3),PHIM(4,4),XI(3),PH4(3)
     COMMON/ADF/ D(3),QD(4),QMA(4),QO1(4),CM1(4),Q(4),VSTN(6)
     COMMON/CFHST/ YRO(6,3),YMS(6),DUMMY(43)
     COMMON/SUM/ QS(4),DRIFT(3),PE(3)
     COMMON/TEMP/ HX(6)
     REAL ID,IGH
C
     IF(INIT.GT.0)GO TO 5
     READ(17,*)  DRIFT,PO,RNI,QNI,IYO
C    WRITE(2,*) PE,PO,RNI,QNI,NADF
     DO 10 I=1,6
     ID(I,I)=1.0
     QN(I,I)=QNI(I)
     P(I,I)=PO(I)
 10  CONTINUE
     DO 20 I=1,6
     RN(I,I)=RNI(I)
 20   CONTINUE
C    WRITE(10) T,Q,X,YMS
C
C    OBSERVATION MATRIX
C
     H(1,2)=0.
     H(1,3)=-2.
     H(2,2)=2
     H(2,3)=0.
C
     H(3,1)=0.0
     H(3,2)=0.
     H(3,3)=-2.
     H(4,1)=0.4
     H(4,2)=2.
     H(4,3)=0.
     H(5,1)=0.0
     H(5,2)=0.
     H(5,3)=-2.
     H(6,1)=-0.4
     H(6,2)=2.
     H(6,3)=0.
     INIT=1
     GO TO  999
  5  CONTINUE
C
     CALL SSTM(CM1(2),CM1(3),CM1(4),CM1(1),PHIM)
```

```
      DO 11 I=1,3
   11 CO(I)=QO1(I)/QO1(4)
   11 CONTINUE
      DO 12 I=1,3
      DO 12 J=1,4
      PHI(J,I)=PHIM(J,I)-PHIM(J,4)*XI(I)
   12 CONTINUE
C      DO 13 J=1,3
C      PH4(J)=PHIM(4,J)-PHIM(4,4)*XI(J)
C  13  CONTINUE
      CP=-.5
      DO 300 I=1,3
      PH(I,I+3)=Q(4)*CP
      DO 300 J=1,3
  300 PH(I,J)=PHI(I,J)
      PH(3,5)=Q(1)*CP
      PH(1,6)=Q(2)*CP
      PH(2,4)=Q(3)*CP
      PH(1,5)=-PH(2,4)
      PH(2,6)=-PH(3,5)
      PH(3,4)=-PH(1,6)
      PH(4,4)=1.0
      PH(5,5)=1.0
C     DO 14 I=1,4
C     QO1(I)=Q(I)
C 14  CONTINUE
C -----------------------------------------
      DO 25 J=1,6
      IF(IYO(J).EQ.1) GO TO 24
      Z(J)=0.0
      RN(J,J)=9999.
      GO TO 25
   24 Z(J)=YMS(J)
   25 CONTINUE
   30 CONTINUE
C     STATE ESTIMATE EXTRAPOLATION
C     X+ = X-
      CALL MULMAT(PH,X,X,6,6,1)
      DO 222 I=1,3
  222 X(I)=Q(I)
C     END STATE EXTRAPOLATION
C
C     ERROR COVARIANCE EXTRAPOLATION
C
C     P+ = P-
      CALL MULMAT(PH,P,PP,6,6,6)
      CALL TRMAT(PH,PHIT,6,6)
      CALL MULMAT(PP,PHIT,PPPT,6,6,6)
      CALL ADDMAT(PPPT,QN,P,6,6)
C
C     END COVARIANCE EXTRAPOLATION
C
C     GAIN MATRIX GENERATION
```

```
C
      CALL TRMAT(H,HT,6,6)
      CALL MULMAT(P,HT,PHT,6,6,6)
      CALL MULMAT(H,PHT,HPHT,6,6,6)
      CALL ADDMAT(HPHT,RN,HPHTR,6,6)
      DO 31 J=1,6
      DO 31 K=1,6
      HPHTRO(J,K)=HPHTR(J,K)
 31   CONTINUE
      CALL MATINV(HPHTRO,HPHTRI,6,6,IER)
      CALL MULMAT(PHT,HPHTRI,G,6,6,6)
C
C      END GAIN GENERATION
C
C     STATE ESTIMATE UPDATE
C
      CALL MULMAT(H,X,HX,6,6,1)
      DO 40 J=1,6
 40   HXM(J)=-HX(J)
      CALL ADDMAT(Z,HXM,ZHX,6,1)
      CALL MULMAT(G,ZHX,GZHX,6,6,1)
      CALL ADDMAT(X,GZHX,X,6,1)
C
C     END STATE ESTIMATE UPDATE
C
C     ERROR COVARIANCE UPDATE
C
      CALL MULMAT(G,H,GH,6,6,6)
      DO 32 L=1,6
      DO 32 K=1,6
      GH(L,K)=-GH(L,K)
 32   CONTINUE
      CALL ADDMAT(ID,GH,IGH,6,6)
      CALL MULMAT(IGH,P,P,6,6,6)
C
C     END COVARIANCE UPDATE
C
      DO  50 J=1,3
      QD(J)=GZHX(J)
      D(J)=X(J+3)
 50   CONTINUE
      QD(4)=SQRT(1.-QD(1)**2-QD(2)**2-QD(3)**2)
C     WRITE(10) T,QD,YMS,X(4),X(5),X(6)
C     OUTPUT T,DET,H,Q,P,G
 999  CONTINUE
C      DO 777 I=1,4
C 777  Q01(I)=Q(I)
      RETURN
      END
```

IPS KF Input Data

IPSL.DAT

```
.5E-5 .5E-5 .5E-5
.5E-5 .5E-5 .5E-5 .5E-5 .5E-5 .5E-5
2
0. 1 0. 0. 0.
100. 2 0. 0. 0.
1.45E-5 1.45E-5 1.45E-5
.007 .005 .005 1.E-7 1.E-7 1.E-7 4.85E-6 4.85E-6 4.85E-6 4.85E-6 4.85E-6 4.85E-6
1.212E-6 1.212E-6 1.212E-6 1.E-9 1.E-9 1.E-9
1 1 1 1 0 0
```

APPENDIX D

Linearized Kalman Filter Listing and Data

IPS LKF Batch Command File

TREEBF.COM

```
$ JOB
$ SET DEF [ROBINSON.TREETOPS]
$    ASSIGN/NOLOG IPSREV7.INT FOR001
$    ASSIGN/NOLOG IPSREV7.OUT FOR009
$    ASSIGN/NOLOG IPSREV7.RES FOR003
$    ASSIGN/NOLOG IPSREV7.MAT FOR004
$    ASSIGN/NOLOG IPSREV7.LOG FOR006
$    ASSIGN/NOLOG IPSREV7.MDK FOR019
$    ASSIGN/NOLOG IPSREV7.FLX FOR011
$    ASSIGN/NOLOG IPSREV7.FLN FOR013
$    ASSIGN/NOLOG IPSREV7.AUX FOR039
$    ASSIGN/NOLOG IPSREV7.XRF FOR012
$    ASSIGN/NOLOG TCATEMP.DAT  FOR025
$    ASS ASTP FOR015
$    ASS OPTNL FOR016
$    ASS IPSE   FOR017
$    ASS PLOTP  FOR018
$    ASSIGN/NOLOG DEFAULT.DFF.1 FOR085
$    RUN [ROBINSON.TREETOPS.FILES]FCON.EXE
$    DEASSIGN FOR001
$    DEASSIGN FOR009
$    DEASSIGN FOR003
$    DEASSIGN FOR004
$    DEASSIGN FOR006
$    DEASSIGN FOR019
$    DEASSIGN FOR011
$    DEASSIGN FOR013
$    DEASSIGN FOR039
$    DEASSIGN FOR012
$    DEASSIGN FOR025
$    DEASSIGN FOR085
$    RENAME TCATEMP.DAT IPSREV7.MAT
$    DIR/DATE/SIZE IPSREV7.OUT
$ COPY IPSREV7.OUT IPSREV7.PLT
```

ADFP.FOR

```
      SUBROUTINE LADF
C
      REAL KA
      LOGICAL LTHDIS,LSTMES
      DIMENSION TDS2S1(3,3),TDS3S1(3,3),XI(3),PH(3,6),QMS(4),PH4(3)
      DIMENSION A(4,4),QMS2(4),H4(9,3),AZ(4,4),CMS(4),XD1(3),PH1(4,3)
      DIMENSION A3(3,3),XD6(6),XD3(3),XDH(10)
CCC   DIMENSION A3(3,3),XD6(6),XD7(3),XD3(3),XDH(10),XDHO(10)
CCC  1,XDH6(6),XDH1(3)
C
C-------------------------------------------------------
C
C NEW ARRAYS ADDED FOR USE BY DELAYS ADDED BY ADAMS & WEST
C
CCC   DIMENSION QD11(4),QD12(4),D11(3),D12(3)
CCC   DATA QD11,D11/ 4*0.0,  3*0.0 /
CCC   DIMENSION QD(4),D(3)
CCC   DATA QD,XD1,XD3,XDHO,XDH/30*0.0/
C
C-------------------------------------------------------
C
      COMMON /ADF/ D(3),QD(4),QM(4),Q0(4),CM(4),Q(4)
CCCC  COMMON /FILTER/ DI(3),QDI(4),QM(4),Q0(4),CM(4),Q(4)
      COMMON /KALMAN/ ND(6),NF(6,2),NTR,AM(4,4),Q0A(4),QMA(4),NZ,
     *TM(3,3),IZ,TS2S1(3,3),TS3S1(3,3),YD(2),H6(9,3),XD(10),
     * CS(3,10,2),C(3,10,2),KA(3,10,2),PHI(3,3),FAC,FAC1,FAC2,PMAX,
     * LTHDIS
      COMMON /TEST/ ITEST(10)
      COMMON/COSUB/YMSA(6,2),T0S2S1(3,3),T0S3S1(3,3)
      COMMON/COGAIN/TGAIN1(2),TGAIN2(2),TGAIN3(2),C1(3,10,2),C2(3,10,2),
     1              JSST
C
      COMMON/COG2/ IG
      DATA TDS2S1,TDS3S1 /18*0./,LSTMES/.FALSE./
      NAMELIST/TEST8/NTR,XD,AM,D,PHI,CM,QD,Q0,QM,Q0A,QMA
C
C     REINITIALIZATION OF THE STATE DEVIATION VECTOR AT THE BEGINNING OF
C     EACH FILTER INTERVAL (INPUT FOR SUBOPT FOR THE FIRST STAR),
C     BECAUSE THE FILTER EQUATIONS ARE PROGRAMMED FOR THE STATE VECTOR
C     UPDATES, NOT FOR THE STATE VECTOR ITSELF
C
C
C-------------------------------------------------------------------
C
C    INITIALIZATION ADDED BY ADAMS & WEST
C
      DO 10 I=1,4
      Q0A(I) = Q0(I)
      QMA(I) = Q(I)
 10   CONTINUE
```

```
      CALL SSTM(CM(2),CM(3),CM(4),CM(1),AM)
C     CALL MAVE(AM,QD,QDI,4)
C     DO 15 I=1,3
C15   DI(I)=D(I)
C-----------------------------------------------------------------
C
C      DO 50 I=1,10
C   50 XD(I)=0.
C
C
C     MODIFICATION OF THE SUBSTATE TRANSITION MATRIX (REF.SECT.4.6.1)
C
   70 CQ=1./Q0A(4)
      DO 100 I=1,3
      XI(I)=Q0A(I)*CQ
      DO 100 J=1,4
  100 PH1(J,I)=AM(J,I)-AM(J,4)*XI(I)
C     PRINT 150
C 150 FORMAT(' PH1')
C     PRINT 155,((PH1(J,I),J=1,4),I=1,3)
C 155 FORMAT(4G20.6)
      DO 200 I=1,3
  200 PH4(I)=PH1(4,I)
C     PRINT 220
C 220 FORMAT(' PH4')
C     PRINT 155,(PH4(I),I=1,3)
C
C     COMPUTATION OF THE STATE TRANSITION MATRIX (REF. SECT.4.6.1 C)
C
      CP=-.5
      DO 300 I=1,3
      PH(I,I+3)=QMA(4)*CP
      DO 300 J=1,3
  300 PH(I,J)=PH1(I,J)
      PH(3,5)=QMA(1)*CP
      PH(1,6)=QMA(2)*CP
      PH(2,4)=QMA(3)*CP
      PH(1,5)=-PH(2,4)
      PH(2,6)=-PH(3,5)
      PH(3,4)=-PH(1,6)
C     PRINT 350
C 350 FORMAT(' PH')
C     NORMALIZATION OF MEAN QUATERNION VECTORS (REF. SECT.4.6.1 D)
C
      DO 1450 I=1,6
 1450 XD6(I)=XD(I)
      DO 50 I=1,10
   50 XD(I)=0.
      CALL GMPRD(PH,XD6,XD1,3,6,1)
      DO 51 I=1,3
   51 XDH(I)=XD1(I)
      CP=0.
      DO 400 I=1,4
```

```
  400 CP=CP+QMA(I)*QMA(I)
      CP=1./SQRT(CP)
      DO 500 I=1,4
  500 QMS(I)=QMA(I)*CP
C
C     COMPUTATION OF THE DIRECTION COSINE MATRIX FROM THE IPS INERTIAL
C     REFERENCE SYSTEM TO THE IPS PLATFORM SYSYTEM RESULTING FROM THE
C     MEAN QUATERNION VECTOR OF THE PREVIOUS FILTER INTERVAL
C     (REF. SECT. 4.6.1 E)
C
      DO 600 I=1,4
      DO 600 J=1,4
  600 A(I,J)=QMS(I)*QMS(J)
      DO 700 I=1,3
  700 TM(I,I)=2.*(A(4,4)+A(I,I))-1.
      TM(1,2)=2.*(A(1,2)+A(3,4))
      TM(1,3)=2.*(A(3,1)-A(2,4))
      TM(2,1)=2.*(A(2,1)-A(3,4))
      TM(2,3)=2.*(A(2,3)+A(1,4))
      TM(3,1)=2.*(A(1,3)+A(2,4))
      TM(3,2)=2.*(A(2,3)-A(1,4))
C
C     COMPUTATION OF THE FHST INDEPENDENT PART OF THE
C     OBSERVATION MATRIX (REF. SECT. 4.6.2)
C
      CP=2./QMS(4)
      DO 800 I=1,3
      DO 800 J=1,3
  800 A(I,J)=A(I,J)*CP
      DO 900 I=1,4
  900 QMS2(I)=QMS(I)*2.
      H4(1,1)=0.
      H4(1,2)=-QMS(2)*4.
      H4(1,3)=-QMS(3)*4.
      H4(2,1)=QMS2(2)-A(1,3)
      H4(2,2)=QMS2(1)-A(2,3)
      H4(2,3)=QMS2(4)-A(3,3)
      H4(3,1)=QMS2(3)+A(1,2)
      H4(3,2)=-QMS2(4)+A(2,2)
      H4(3,3)=QMS2(1)+A(2,3)
      H4(4,1)=QMS2(2)+A(1,3)
      H4(4,2)=QMS2(1)+A(2,3)
      H4(4,3)=-QMS2(4)+A(3,3)
      H4(5,1)=-4.*QMS(1)
      H4(5,2)=0.
      H4(5,3)=-4.*QMS(3)
      H4(6,1)=QMS2(4)-A(1,1)
      H4(6,2)=QMS2(3)-A(1,2)
      H4(6,3)=QMS2(2)-A(1,3)
      H4(7,1)=QMS2(3)-A(1,2)
      H4(7,2)=QMS2(4)-A(2,2)
      H4(7,3)=QMS2(1)-A(2,3)
      H4(8,1)=-QMS2(4)+A(1,1)
```

```
      H4(8,2)=QMS2(3)+A(1,2)
      H4(8,3)=QMS2(2)+A(1,3)
      H4(9,1)=-4.*QMS(1)
      H4(9,2)=-4.*QMS(2)
      H4(9,3)=0.
      DO 1000 I=1,4
 1000 AZ(1,I)=QOA(I)
      AZ(2,1)=QOA(4)
      AZ(2,2)=QOA(3)
      AZ(2,3)=-QOA(2)
      AZ(2,4)=-QOA(1)
      AZ(3,1)=-QOA(3)
      AZ(3,2)=QOA(4)
      AZ(3,3)=QOA(1)
      AZ(3,4)=-QOA(2)
      AZ(4,1)=QOA(2)
      AZ(4,2)=-QOA(1)
      AZ(4,3)=QOA(4)
      AZ(4,4)=-QOA(3)
      CALL GMPRD(AZ,QMS,CMS,4,4,1)
      CALL SSTM(CMS(2),CMS(3),CMS(4),CMS(1),A)
      DO 1100 I=1,3
      DO 1100 J=1,3
 1100 A3(I,J)=A(I,J)-A(I,4)*XI(J)
      CALL GMPRD(H4,A3,H6,9,3,3)
C
C     DEFINITION OF THE CORRECTION FACTOR FOR THE ESTIMATION
C     OF THE FHST THERMAL DISTORTIONS
C
      IF(LTHDIS)GOTO 1170
      FAC1=0.
      FAC2=0.
 1170 IF((ND(2)+ND(5)).EQ.0)FAC1=0.
      IF((ND(3)+ND(6)).EQ.0)FAC2=0.
C
C     REFRESHING OF THE SKEWED/BORESIGHTED FHST DCM'S (REF. SECT. 4.6.3)
C
      DO 1200 I=1,3
      TDS2S1(I,I)=1.
 1200 TDS3S1(I,I)=1.
      TDS2S1(1,2)=PHI(2,3)
      TDS2S1(3,1)=PHI(2,2)
      TDS2S1(1,3)=-PHI(2,2)
      TDS2S1(2,1)=-PHI(2,3)
      TDS3S1(1,2)=PHI(3,3)
      TDS3S1(3,1)=PHI(3,2)
      TDS3S1(1,3)=-PHI(3,2)
      TDS3S1(2,1)=-PHI(3,3)
      CALL GMPRD(TDS2S1,TOS2S1,TS2S1,3,3,3)
      CALL GMPRD(TDS3S1,TOS3S1,TS3S1,3,3,3)
C
C     SUBOPTIMAL FILTER, LOOP FOR THE SIX STARS (REF. SECT. 4.6.4)
C
```

```
      NTR=NTR+1
      NZ=0
C
 1300 CONTINUE
C
      NZ=NZ+1
      IZ=NZ
      IF(NZ.GT.3) IZ=NZ-3
      DO 1400 I=1,2
 1400 NF(NZ,I)=ND(NZ)
      IF(ND(NZ).NE.1) GO TO 1405
      CALL SUBOPT(XDH,XDHO,LSTMES)
 1405 CONTINUE
C     PRINT 1410
C1410 FORMAT(' XD')
C     PRINT 1420,(XD(I),I=1,10)
C1420 FORMAT(10G12.4)
C
      IF(NZ.NE.6) GOTO 1300
C
C---- UPDATE IS NOT DONE IN THE FIRST FILTER STEP , BECAUSE STAR POSI-
C     TION MEASUREMENT DATA ARE NOT YET AVAILABLE ----------------------
C
      IF (.NOT.LSTMES) GOTO 2000
C
C     PREDICTION AND PREPARATION OF FILTER UPDATE DATA (REF. SECT.4.6.5)
C
C      DO 1450 I=1,6
C     XDH6(I)=XDH(I)
C 1450 XD6(I)=XD(I)
C      CALL GMPRD(PH,XD6,XD1,3,6,1)
C     CALL GMPRD(PH,XDH6,XDH1,3,6,1)
      DO 1500 I=1,3
C     XDH(I)=XDH1(I)
C      XD(I)=XD1(I)
C      XD3(I)=XD6(I)
 1500 QD(I)=XD(I)
C      DO 49 I=1,10
C     XDHO(I)=XDH(I)
C 49   XDH(I)=XD(I)
C     PRINT 1520
C1520 FORMAT(' XD3')
C     PRINT 155,(XD3(I),I=1,3)
C      CALL GMPRD(PH4,XD3,QD(4),1,3,1)
C     PRINT 1530
C1530 FORMAT(' QD=')
C     PRINT 155,(QD(I),I=1,4)
      DO 1600 I=1,3
 1600 D(I)=D(I)+XD(I+3)
C
      DO 1700 I=2,3
      PHI(2,I)=PHI(2,I)+XD(I+5)
 1700 PHI(3,I)=PHI(3,I)+XD(I+7)
```

```
      DO 1701 I=2,3
      IF((ABS(PHI(2,I))).GT.PMAX) PHI(2,I)=PMAX*ABS(PHI(2,I))/PHI(2,I)
 1701  IF((ABS(PHI(3,I))).GT.PMAX) PHI(3,I)=PMAX*ABS(PHI(3,I))/PHI(3,I)
C
      IF(ITEST(8).EQ.0.AND.(D(1).LT..1.AND.D(2).LT..1.AND.D(3).LT..1.
     * AND.QD(1).LT..1.AND.QD(2).LT..1.AND.QD(3).LT..1)) GOTO 2000
      WRITE(6,TEST8)
 2000 CONTINUE
C
      LSTMES = .TRUE.
C
      IG=IG+1
      IF(ITEST(10).EQ.1) ITEST(9)=1
      RETURN
      END
      SUBROUTINE SUBOPT(XDH,XDHO,LSTMES)
C
      LOGICAL LTHDIS,LSTMES
      REAL KA
      DIMENSION YRA(3),YRP(3),T(3,3),YRS(3),YRN(2),H(14),H3(2,9)
CCC   DIMENSION HM(2,10),YDD(2),H23(2,3)
      DIMENSION HM(2,10),YDD(2),H23(2,3),XDH(10),XDHO(10)
C
      COMMON/TRANS/TA_P_I(3,3),TD_P_I(3,3),DUMMY1(27)
      COMMON/KALMAN/ND(6),NF(6,2),NTR,AM(4,4),QOA(4),QMA(4),NZ,TM(3,3),
     *IZ,TS2S1(3,3),TS3S1(3,3),YD(2),H6(9,3),XD(10),CS(3,10,2),C(3,10,2)
     *,KA(3,10,2),PHI(3,3),FAC,FAC1,FAC2,PMAX,LTHDIS
      COMMON/CFHST/YRO(6,3),YMS(6,2),DUMMY(43)
      COMMON/TEST/ITEST(10)
      COMMON/COSUB/YMSA(6,2),TOS2S1(3,3),TOS3S1(3,3)
      COMMON/CSUBOP/TSETTL,EFLCH1,EFLCH2
      COMMON/COG2/IG
      COMMON/COGAIN/TGAIN1(2),TGAIN2(2),TGAIN3(2),DUMMY2(120),
     *JSST,HOEQU1,HOEQU2,HOEQU3,HOEDR1,HOEDR2,HOEDR3,
     *AHYPV(3,10,2),AHYPN(3,10,2),BHYPV(3,10,2),BHYPN(3,10,2),
     *CHYPV(3,10,2),CHYPN(3,10,2),TMAXMA(3,10,2)
C
      NAMELIST/TEST7/NTR,NZ,YRN,YD,YDD,XD
C**********************************************************************
C     FHST DATA PROCESSING (REF. SECT. 4.6.4.4)
C**********************************************************************
C     YRO         THE COLUMNS OF YRO ARE THE STAR DIRECTION VECTORS
C                 IN THE IPS INERTIAL REFERENCE SYSTEM
C                 YRO IS COMPUTED IN 'FHST'
C     TM          DCM FROM THE I- TO THE P-SYSTEM BASED ON THE GYRO DATA
C                 TM IS COMPUTED IN 'KAFILT'
C     TS2S1,TS3S1  TRANSFORMATION MATRICES FROM THE BORESIGHTED SENSOR
C                   SYSTEM (S1) TO THE SKEWED SENSOR SYSTEM (S2,S3)
C                   AS USED IN THE ADF, BASED ON THE ESTIMATED OF THE
C                   MISALIGNMENTS
      C1=.7071068
C
      DO 50 I=1,3
```

```
   50 YRA(I)=YRO(NZ,I)
C
C---- YRP = STAR DIRECTION IN THE P-SYSTEM BASED ON THE GYRO DATA
C           YRP = TM * YRA
C
      CALL GMPRD(TM,YRA,YRP,3,3,1)
C
      GOTO (300,400,500),IZ
C
C---- TRANSFORMATION MATRIX FROM P TO S1 SYSTEM
C
  300 DO 350 I=1,3
      DO 350 J=1,3
  350 T(I,J)=0.
      DO 370 I=1,3
  370 T(I,I)=1.
C
      GOTO 600
C
C---- TRANSFORMATION MATRIX FROM S1 TO S2 SYSTEM
C
  400 DO 450 I=1,3
      DO 450 J=1,3
  450 T(I,J)=TS2S1(I,J)
C
      GOTO 600
C
C---- TRANSFORMATION MATRIX FROM S1 TO S3 SYSTEM
C
  500 DO 550 I=1,3
      DO 550 J=1,3
  550 T(I,J)=TS3S1(I,J)
C
  600 CONTINUE
C
      CALL GMPRD(T,YRP,YRS,3,3,1)
C
      CX=1./YRS(1)
      CX2=CX*CX
C
C---- YRN = NOMINAL MEASUREMENT VECTOR; THE NOMINAL P-SYSTEM IS
C                          DETERMINED FROM THE GYRO MEASUREMENTS--------

      DO 700 I=1,2
  700 YRN(I)=YRS(I+1)*CX
C
C************* MEASUREMENT DEVIATION VECTOR ****************************
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  NEXT 2 CARDS FOR YMSA MOVED UP TO HERE BY MARK WEST
C
      YMSA(NZ,1)=YMS(NZ,1)
```

```
      YMSA(NZ,2)=YMS(NZ,2)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      DO 800 I=1,2
  800 YD(I) = YMSA(NZ,I) - YRN(I)
  806 CONTINUE
C
C-----YMSA(NZ,1)=C1*(YMS(NZ,1)+YMS(NZ,2))
C-----YMSA(NZ,2)=C1*(-YMS(NZ,1)+YMS(NZ,2))
C
C---- FIRST STEP : FILTER IS NOT PROCESSED , DATA ARE NOT YET AVAILABLE
C
      IF (.NOT.LSTMES) GO TO 2000
C
C**** LIMIT CHECK FOR OSP FAILURE DETECTION (REF. SECT. 4.6.4.2 E) *****
C
      GO TO 950
      EF=EFLCH1
      IF(NTR.GE.TSETTL)EF=EFLCH2
      DO 900 I=1,2
  900 IF(ABS(YD(I)).GT.EF) NF(NZ,I)=0
C
      IF((NF(NZ,1)*NF(NZ,2)).NE.0) GOTO 950
      PRINT 930,NTR,EFLCH1,EFLCH2,TSETTL
  930 FORMAT(1H0,'LIMIT CHECK: SENSOR FAILURE AT NTR =',I4,/,1H ,
     1'EFLCH1=',G12.4,5X,'EFLCH2=',G12.4,5X,'TSETTL=',G12.4)
      WRITE(6,TEST7)
      STOP
C
C**** COMPUTATION OF THE OBSERVATION MATRIX (REF. SECT.4.6.4.4) ********
C
  950 H(1)=-YRN(1)*CX
      H(2)=-YRN(2)*CX
      H(3)=H(1)-CX*PHI(IZ,3)
      H(4)=CX+H(1)*PHI(IZ,3)
      H(5)=-H(1)*PHI(IZ,2)
      H(6)=H(2)+CX*PHI(IZ,2)
      H(7)=H(2)*PHI(IZ,3)
      H(8)=CX-H(2)*PHI(IZ,2)
      DO 1000 I=1,3
      H(I+8)=H(3)*T(1,I)+H(4)*T(2,I)+H(5)*T(3,I)
 1000 H(I+11)=H(6)*T(1,I)+H(7)*T(2,I)+H(8)*T(3,I)
C
      DO 1100 I=1,3
      H3(1,I)=H(9)*YRA(I)
      H3(1,I+3)=H(10)*YRA(I)
      H3(1,I+6)=H(11)*YRA(I)
      H3(2,I)=H(12)*YRA(I)
      H3(2,I+3)=H(13)*YRA(I)
 1100 H3(2,I+6)=H(14)*YRA(I)
      CALL GMPRD(H3,H6,H23,2,9,3)
C
```

```
      DO 1120 I=1,2
      DO 1120 J=1,3
 1120 HM(I,J)=H23(I,J)
C
      DO 1150 I=1,2
      DO 1150 J=4,10
 1150 HM(I,J)=0.
C
C---- COMPUTE ESTIMATET MEASUREMENT DEVIATION : YDD = HM * XDH
C
      GOTO (1200,1300,1400),IZ
C
 1200 DO 1250 I=1,2
      YDD(I)=0.
      DO 1250 J=1,3
 1250 YDD(I)=YDD(I)+HM(I,J)*XDH(J)
C
      GOTO 1500
 1300 HM(1,7)=YRS(2)*YRS(3)*CX2
      HM(1,8)=-YRS(2)*YRS(2)*CX2-1.
      HM(2,7)=YRS(3)*YRS(3)*CX2+1.
      HM(2,8)=-HM(1,7)
      DO 1350 I=1,2
      YDD(I)=0.
C
      DO 1350 J=7,8
 1350 YDD(I)=YDD(I)+HM(I,J)*XDH(J)
C
      DO 1370 I=1,2
      DO 1370 J=1,3
 1370 YDD(I)=YDD(I)+HM(I,J)*XDH(J)
C
      GOTO 1500
 1400 HM(1,9)=YRS(2)*YRS(3)*CX2
      HM(1,10)=-YRS(2)*YRS(2)*CX2-1.
      HM(2,9)=YRS(3)*YRS(3)*CX2+1.
      HM(2,10)=-HM(1,9)
C
      DO 1450 I=1,2
      YDD(I)=0.
      DO 1450 J=9,10
 1450 YDD(I)=YDD(I)+HM(I,J)*XDH(J)
C
      DO 1470 I=1,2
      DO 1470 J=1,3
 1470 YDD(I)=YDD(I)+HM(I,J)*XDH(J)
C
 1500 CONTINUE
      IF((NZ.GT.3).AND.ND(IZ).NE.0) GOTO 1600
C
C**** COMPUTATION OF THE SUBOPTIMAL GAIN MATRIX (REF. SECT. 4.6.4.5) ***
C
      CALL GAINMA
```

```
 1600 CONTINUE
C
C**** STATE DEVIATION VECTOR UPDATE (REF. SECT. 4.6.4.6) ***************
C
      DO 1700 I=1,10
      DO 1700 J=1,2
 1700 XD(I)=XD(I)+KA(IZ,I,J)*(YD(J)-YDD(J))
C
      IF(ITEST(7).NE.0)WRITE(6,TEST7)
C
 2000 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE GAINMA
C
      REAL KA
C
      COMMON/COGAIN/TGAIN1(2),TGAIN2(2),TGAIN3(2),C1(3,10,2),C2(3,10,2),
     1              JSST,HOEQU1,HOEQU2,HOEQU3,HOEDR1,HOEDR2,
     2              HOEDR3,AHYPV(3,10,2),AHYPN(3,10,2),BHYPV(3,10,2),
     3              BHYPN(3,10,2),CHYPV(3,10,2),CHYPN(3,10,2),
     4              TMAXMA(3,10,2)
      COMMON/KALMAN/DUMMY1(18),NTR,DUMMY2(34),IZ,DUMMY3(177),KA(3,10,2),
     1              DUMMY4(9),FAC,FAC1,FAC2,PMAX,DUMMY5
      COMMON/COG1/G(500,10,6),JSUBOP
      COMMON/COG2/IG
C
C********** STATIONARY GAINS
C
      IG1=IG
      IF(IG.GT.KEND)IG1=KEND
C********** DECISION FOR THE SORT OF GAINS  *************************
C
C---- JSUBOP=0 : OPTIMAL GAINS
      IF(JSUBOP.EQ.0) GO TO 1000
C
C---- JSUBOP=1 : STAIRCASE GAINS (IN CASE OF RATE LIMITATION)
      IF(JSUBOP.EQ.1) GO TO 2000
C
C---- JSUBOP=2 : PARABOLIC GAINS (CASE WITHOUT RATE LIMITATION)
      IF(JSUBOP.EQ.2) GO TO 3000
C
C---- JSUBOP=3 : HYPERBOLIC GAINS (IN CASE OF RATE LIMITATION)
      IF(JSUBOP.EQ.3) GO TO 4000
C
C********    CALCULATION OF OPTIMAL GAINS    **********************
C
 1000 CONTINUE
C
      DO 1200 I=1,10
      DO 1200 J=1,2
```

```
C
      J1=2*(IZ-1)+J
      KA(IZ,I,J)=G(IG,I,J1)
C
 1200 CONTINUE
C
      GO TO 6000
C
C**************  CALCULATION OF STAIRCASE GAINS    **********************
C
C---- COMPUTATION OF THE GAINS FROM THE PARAMETERMATRIX C1 -------------
C
 2000 T=IG
      DO 2020 K=1,2
      KA(IZ,J,K)=C1(IZ,J,K)
C
 2020 CONTINUE
C
C---- GAIN REDUCTION FOR THE QUATERNIONS ----------------------------
C
      DO 2500 J=1,3
C
      IF(T.LT.TGAIN1(1)) GO TO 2100
      IF(T.LT.TGAIN2(1)) GO TO 2200
      IF(T.LT.TGAIN3(1)) GO TO 2300
      GO TO 2500
C
C---- 0 < T < TGAIN1(1) -----------------------
 2100 DO 2110 K=1,2
 2110 KA(IZ,J,K)=HOEQU1*KA(IZ,J,K)
      GO TO 2500
C
C---- TGAIN1(1) < T < TGAIN2(1) ---------------
 2200 DO 2210 K=1,2
C---- SPRUNG
 2210 KA(IZ,J,K)=HOEQU2*KA(IZ,J,K)
      GO TO 2500
C
C---- TGAIN2(1) < T < TGAIN3(1) ---------------
 2300 DO 2310 K=1,2
 2310 KA(IZ,J,K)=HOEQU3*KA(IZ,J,K)
C
 2500 CONTINUE
C
C---- GAIN REDUCTION FOR DRIFTS -----------------------------------
C
      DO 2800 J=4,6
C
      IF(T.LT.TGAIN1(1))  GO TO 2550
      IF(T.LT.TGAIN2(1))  GO TO 2600
      IF(T.LT.TGAIN3(1))  GO TO 2700
      GO TO 2800
C
```

```
C---- 0 < T < TGAIN1(1)
 2550 DO 2560 K=1,2
 2560 KA(IZ,J,K)=HOEDR1*KA(IZ,J,K)
      GO TO 2800
C
C---- TGAIN1(1) < T < TGAIN2(1)
 2600 DO 2610 K=1,2
C---- SPRUNG
 2610 KA(IZ,J,K)=HOEDR2*KA(IZ,J,K)
      GO TO 2800
C
C---- TGAIN2(1) < T < TGAIN3(1)
 2700 DO 2710 K=1,2
 2710 KA(IZ,J,K)=HOEDR3*KA(IZ,J,K)
C
 2800 CONTINUE
      GO TO 6000
C
C*************************************************************************
C
C     CALCULATION OF THE PARABOLIC GAINS (NORMAL CASE)
C
C*************************************************************************
C
 3000 DO 3900 J=1,10
C
      T=IG
C
C---- QUATERNIONS (J=1,2,3) ---------------------------------------
      IF(J.GT.3)GOTO 3100
      IF(T.LT.TGAIN1(1))   T=TGAIN1(1)
      IF(T.GT.TGAIN2(1))   T=TGAIN3(1)
      GO TO 3500
C
C---- DRIFTS (J=4,5,6) --------------------------------------------
 3100 IF(J.GT.6.) GO TO 3200
      IF(T.LT.(TGAIN1(2)))   T=TGAIN1(2)
      IF(T.GT.(TGAIN2(2)))   T=TGAIN3(2)
      GO TO 3500
C
C---- MISALIGNEMENTS (J=7,8,9,10) ---------------------------------
 3200 IF(T.LT.TGAIN1(2))T=TGAIN1(2)
      IF(T.GT.TGAIN2(2))T=TGAIN3(2)
C
 3500 CONTINUE
C
C---- COMPUTATION OF THE GAINS FROM THE PARAMETERMATRICES C1,C2 --------
C
      DO 3900 K=1,2
      KA(IZ,J,K)=C1(IZ,J,K)*T**C2(IZ,J,K)
C
 3900 CONTINUE
C
```

```
      GO TO 6000
C
C*********************************************************************
C
C     COMPUTATION OF THE HYPERBOLIC GAINS  (IN CASE OF RATE LIMITATION)
C
C*********************************************************************
C
 4000 T=IG
C
      DO 4900 J=1,10
      DO 4900 K=1,2
      JG1=TMAXMA(IZ,J,K)
      IF(IG.GT.JG1) GO TO 4500
C
C---- COMPUTATION OF THE GAINS FOR T <= TMAX
C
      KA(IZ,J,K)=AHYPV(IZ,J,K)/(T+BHYPV(IZ,J,K))+CHYPV(IZ,J,K)
      GO TO 4900
C
C---- COMPUTATION OF THE GAINS FOR T > TMAX
C
 4500 KA(IZ,J,K)=AHYPN(IZ,J,K)/(T+BHYPN(IZ,J,K))+CHYPN(IZ,J,K)
C
 4900 CONTINUE
 6000 CONTINUE
C
C---- REDEFINITION OF MISALIGNEMENT GAINS ----------------------------
C
      DO 6090 J=7,8
      DO 6090 K=1,2
 6090 KA(IZ,J,K)=FAC1*KA(IZ,J,K)
C
      DO 6095 J=9,10
      DO 6095 K=1,2
 6095 KA(IZ,J,K)=FAC2*KA(IZ,J,K)
C
C*********************************************************************
C     REDEFINITION OF KA FOR SOLAR POINTING
C*********************************************************************
C
      IF(IZ.EQ.1.OR.JSST.EQ.2) GO TO 6100
      KA(IZ,2,1)=0.
      KA(IZ,2,2)=0.
      KA(IZ,3,1)=0.
      KA(IZ,3,2)=0.
      KA(IZ,5,1)=0.
      KA(IZ,5,2)=0.
      KA(IZ,6,1)=0.
      KA(IZ,6,2)=0.
 6100 CONTINUE
C
      RETURN
```

```
C
      END
C
C
C
C
C
```

IPS LKF Input Data

IPSE.DAT

```
.5E-5 .5E-5 .5E-5
.5E-5 .5E-5 .5E-5 .5E-5 .5E-5 .5E-5
2
0. 1 0. 0. 0.
100. 2 0. 0. 0.
1.45E-5 1.45E-5 1.45E-5
.007 .005 .005 1.E-7 1.E-7 1.E-7 4.85E-6 4.85E-6 4.85E-6 4.85E-6 4.85E-6 4.85E-6
1.212E-6 1.212E-6 1.212E-6 1.E-9 1.E-9 1.E-9
```

ASTP.DAT

```
 &NFILT
 LATTUP=T, P_I=-1500., 900., 900., NTR=-2, JSUBOP=3
 MODE(1)=1,    BJAS(1,1)=0.0,    SIGMAN(1,1)=.75,    YS(1,1)=0.,
               BJAS(1,2)=0.0,    SIGMAN(1,2)=.75,    YS(1,2)=0.,
 MODE(2)=1,    BJAS(2,1)=0.0,    SIGMAN(2,1)=1.0,    YS(2,1)=0.,
               BJAS(2,2)=0.0,    SIGMAN(2,2)=0.0,    YS(2,2)=0.,
 MODE(3)=0,    BJAS(3,1)=0.0,    SIGMAN(3,1)=1.0,    YS(3,1)=0.,
               BJAS(3,2)=0.0,    SIGMAN(3,2)=1.0,    YS(3,2)=0.,
 MODE(4)=0,    BJAS(4,1)=0.0,    SIGMAN(4,1)=.75,    YS(4,1)=0.,
               BJAS(4,2)=0.0,    SIGMAN(4,2)=.75,    YS(4,2)=0.,
 MODE(5)=0,    BJAS(5,1)=0.0,    SIGMAN(5,1)=1.,    YS(5,1)=0.,
               BJAS(5,2)=0.0,    SIGMAN(5,2)=1.,    YS(5,2)=0.,
 MODE(6)=0,    BJAS(6,1)=0.0,    SIGMAN(6,1)=1.,    YS(6,1)=0.,
               BJAS(6,2)=0.0,    SIGMAN(6,2)=1.,    YS(6,2)=0.,
 ITEST=0,0,0,0,0,0,0,0,0,0, LTHDIS=T, ALPHO=12.0, BETA=-45,
 FAC=1.,  JREAD=16,  JSST=2,
 DEPHIX=0.0, 0.0,   DEPHIY=0.0,0.0,   DEPHIZ=0.0,0.0,
 DRTHX=  2*0.,   DRTHY= 2*0.,   DRTHZ= 0., 0.,
 TSETTL=60.0,  EFLCH1=2000.0,  EFLCH2=90.0,
 LFHST=1, ALP11=7.36, ALP12=127.158, ALP2=0.21875, FDTTR=2.38, WEFF=25.,
 FOVR=72.5,
 FOVDY=0.0, FOVDZ=0.0, DSUN=1920.0, DNEGST=200.0, DSCAN=36.75, RNTNS=0.3,
 &END
```

APPENDIX E

Extended Kalman Filter Listing and Data

PRECEDING PAGE BLANK NOT FILMED

IPS EKF Batch Command File

TREEBE.COM

```
$ JOB
$ SET DEF [ROBINSON.TREETOPS]
$    ASSIGN/NOLOG IPSREV7.INT FOR001
$    ASSIGN/NOLOG IPSREV7.OUT FOR009
$    ASSIGN/NOLOG IPSREV7.RES FOR003
$    ASSIGN/NOLOG IPSREV7.MAT FOR004
$    ASSIGN/NOLOG IPSREV7.LOG FOR006
$    ASSIGN/NOLOG IPSREV7.MDK FOR019
$    ASSIGN/NOLOG IPSREV7.FLX FOR011
$    ASSIGN/NOLOG IPSREV7.FLN FOR013
$    ASSIGN/NOLOG IPSREV7.AUX FOR039
$    ASSIGN/NOLOG IPSREV7.XRF FOR012
$    ASSIGN/NOLOG TCATEMP.DAT  FOR025
$    ASS ASTP FOR015
$    ASS MSUBCO FOR016
$    ASS IPSE   FOR017
$    ASS PLOTE  FOR018
$    ASS/NOLOG DEFAULT.DFF FOR085
$    RUN [ROBINSON.TREETOPS.FILES]ECON.EXE
$    DEASSIGN FOR001
$    DEASSIGN FOR009
$    DEASSIGN FOR003
$    DEASSIGN FOR004
$    DEASSIGN FOR006
$    DEASSIGN FOR019
$    DEASSIGN FOR011
$    DEASSIGN FOR013
$    DEASSIGN FOR039
$    DEASSIGN FOR012
$    DEASSIGN FOR025
$    DEASSIGN FOR085
$    RENAME TCATEMP.DAT IPSREV7.MAT
$    DIR/DATE/SIZE IPSREV7.OUT
$ COPY IPSREV7.OUT IPSREV7.PLT
```

ADFE.FOR

```
      SUBROUTINE LADF
C
      REAL KA
      LOGICAL LTHDIS,LSTMES
      DIMENSION TDS2S1(3,3),TDS3S1(3,3),XI(3),QMS(4),PH4(3)
      DIMENSION A(4,4),QMS2(4),H4(9,3),AZ(4,4),CMS(4),XD1(3),PH1(4,3)
      DIMENSION A3(3,3),XD6(6),XD3(3),XDH(10)
CCC   DIMENSION A3(3,3),XD6(6),XD7(3),XD3(3),XDH(10),XDHO(10)
CCC  1,XDH6(6),XDH1(3)
C
C-------------------------------------------------------
C
C NEW ARRAYS ADDED FOR USE BY DELAYS ADDED BY ADAMS & WEST
C
CCC   DIMENSION QD11(4),QD12(4),D11(3),D12(3)
CCC   DATA QD11,D11/ 4*0.0,  3*0.0 /
CCC   DIMENSION QD(4),D(3)
CCC   DATA QD,XD1,XD3,XDHO,XDH/30*0.0/
C
C-------------------------------------------------------
C
       COMMON /OBS/ HT(6,10),PH(3,6)
      COMMON /ADF/ D(3),QD(4),QM(4),QO(4),CM(4),Q(4)
CCCC  COMMON /FILTER/ DI(3),QDI(4),QM(4),QO(4),CM(4),Q(4)
      COMMON /KALMAN/ ND(6),NF(6,2),NTR,AM(4,4),QOA(4),QMA(4),NZ,
     *TM(3,3),IZ,TS2S1(3,3),TS3S1(3,3),YD(2),H6(9,3),XD(10),
     * CS(3,10,2),C(3,10,2),KA(3,10,2),PHI(3,3),FAC,FAC1,FAC2,PMAX,
     * LTHDIS
      COMMON /TEST/ ITEST(10)
      COMMON/COSUB/YMSA(6,2),TOS2S1(3,3),TOS3S1(3,3)
C
      DATA TDS2S1,TDS3S1 /18*0./,LSTMES/.FALSE./
      NAMELIST/TEST8/NTR,XD,AM,D,PHI,CM,QD,QO,QM,QOA,QMA
C
C     REINITIALIZATION OF THE STATE DEVIATION VECTOR AT THE BEGINNING OF
C     EACH FILTER INTERVAL (INPUT FOR SUBOPT FOR THE FIRST STAR),
C     BECAUSE THE FILTER EQUATIONS ARE PROGRAMMED FOR THE STATE VECTOR
C     UPDATES, NOT FOR THE STATE VECTOR ITSELF
C
C
C-------------------------------------------------------------------
C
C    INITIALIZATION ADDED BY ADAMS & WEST
C
      DO 10 I=1,4
      QOA(I) = QO(I)
      QMA(I) = Q(I)
 10   CONTINUE
      CALL SSTM(CM(2),CM(3),CM(4),CM(1),AM)
C     CALL MAVE(AM,QD,QDI,4)
```

```
C     DO 15 I=1,3
C15   DI(I)=D(I)
C-----------------------------------------------------------------------
C
      DO 50 I=1,10
   50 XD(I)=0.
C
C
C     MODIFICATION OF THE SUBSTATE TRANSITION MATRIX (REF.SECT.4.6.1)
C
   70 CQ=1./QOA(4)
      DO 100 I=1,3
      XI(I)=QOA(I)*CQ
      DO 100 J=1,4
  100 PH1(J,I)=AM(J,I)-AM(J,4)*XI(I)
C     PRINT 150
C 150 FORMAT(' PH1')
C     PRINT 155,((PH1(J,I),J=1,4),I=1,3)
C 155 FORMAT(4G20.6)
      DO 200 I=1,3
  200 PH4(I)=PH1(4,I)
C     PRINT 220
C 220 FORMAT(' PH4')
C     PRINT 155,(PH4(I),I=1,3)
C
C     COMPUTATION OF THE STATE TRANSITION MATRIX (REF. SECT.4.6.1 C)
C
      CP=-.5
      DO 300 I=1,3
      PH(I,I+3)=QMA(4)*CP
      DO 300 J=1,3
  300 PH(I,J)=PH1(I,J)
      PH(3,5)=QMA(1)*CP
      PH(1,6)=QMA(2)*CP
      PH(2,4)=QMA(3)*CP
      PH(1,5)=-PH(2,4)
      PH(2,6)=-PH(3,5)
      PH(3,4)=-PH(1,6)
C     PRINT 350
C 350 FORMAT(' PH')
C     NORMALIZATION OF MEAN QUATERNION VECTORS (REF. SECT.4.6.1 D)
C
      CP=0.
      DO 400 I=1,4
  400 CP=CP+QMA(I)*QMA(I)
      CP=1./SQRT(CP)
      DO 500 I=1,4
  500 QMS(I)=QMA(I)*CP
C
C     COMPUTATION OF THE DIRECTION COSINE MATRIX FROM THE IPS INERTIAL
C     REFERENCE SYSTEM TO THE IPS PLATFORM SYSYTEM RESULTING FROM THE
C     MEAN QUATERNION VECTOR OF THE PREVIOUS FILTER INTERVAL
C     (REF. SECT. 4.6.1 E)
```

```
C
      DO 600 I=1,4
      DO 600 J=1,4
  600 A(I,J)=QMS(I)*QMS(J)
      DO 700 I=1,3
  700 TM(I,I)=2.*(A(4,4)+A(I,I))-1.
      TM(1,2)=2.*(A(1,2)+A(3,4))
      TM(1,3)=2.*(A(3,1)-A(2,4))
      TM(2,1)=2.*(A(2,1)-A(3,4))
      TM(2,3)=2.*(A(2,3)+A(1,4))
      TM(3,1)=2.*(A(1,3)+A(2,4))
      TM(3,2)=2.*(A(2,3)-A(1,4))
C
C     COMPUTATION OF THE FHST INDEPENDENT PART OF THE
C     OBSERVATION MATRIX (REF. SECT. 4.6.2)
C
      CP=2./QMS(4)
      DO 800 I=1,3
      DO 800 J=1,3
  800 A(I,J)=A(I,J)*CP
      DO 900 I=1,4
  900 QMS2(I)=QMS(I)*2.
      H4(1,1)=0.
      H4(1,2)=-QMS(2)*4.
      H4(1,3)=-QMS(3)*4.
      H4(2,1)=QMS2(2)-A(1,3)
      H4(2,2)=QMS2(1)-A(2,3)
      H4(2,3)=QMS2(4)-A(3,3)
      H4(3,1)=QMS2(3)+A(1,2)
      H4(3,2)=-QMS2(4)+A(2,2)
      H4(3,3)=QMS2(1)+A(2,3)
      H4(4,1)=QMS2(2)+A(1,3)
      H4(4,2)=QMS2(1)+A(2,3)
      H4(4,3)=-QMS2(4)+A(3,3)
      H4(5,1)=-4.*QMS(1)
      H4(5,2)=0.
      H4(5,3)=-4.*QMS(3)
      H4(6,1)=QMS2(4)-A(1,1)
      H4(6,2)=QMS2(3)-A(1,2)
      H4(6,3)=QMS2(2)-A(1,3)
      H4(7,1)=QMS2(3)-A(1,2)
      H4(7,2)=QMS2(4)-A(2,2)
      H4(7,3)=QMS2(1)-A(2,3)
      H4(8,1)=-QMS2(4)+A(1,1)
      H4(8,2)=QMS2(3)+A(1,2)
      H4(8,3)=QMS2(2)+A(1,3)
      H4(9,1)=-4.*QMS(1)
      H4(9,2)=-4.*QMS(2)
      H4(9,3)=0.
      DO 1000 I=1,4
 1000 AZ(1,I)=QOA(I)
      AZ(2,1)=QOA(4)
      AZ(2,2)=QOA(3)
```

```
      AZ(2,3)=-Q0A(2)
      AZ(2,4)=-Q0A(1)
      AZ(3,1)=-Q0A(3)
      AZ(3,2)=Q0A(4)
      AZ(3,3)=Q0A(1)
      AZ(3,4)=-Q0A(2)
      AZ(4,1)=Q0A(2)
      AZ(4,2)=-Q0A(1)
      AZ(4,3)=Q0A(4)
      AZ(4,4)=-Q0A(3)
      CALL GMPRD(AZ,QMS,CMS,4,4,1)
      CALL SSTM(CMS(2),CMS(3),CMS(4),CMS(1),A)
      DO 1100 I=1,3
      DO 1100 J=1,3
 1100 A3(I,J)=A(I,J)-A(I,4)*XI(J)
      CALL GMPRD(H4,A3,H6,9,3,3)
C
C     DEFINITION OF THE CORRECTION FACTOR FOR THE ESTIMATION
C     OF THE FHST THERMAL DISTORTIONS
C
      IF(LTHDIS)GOTO 1170
      FAC1=0.
      FAC2=0.
 1170 IF((ND(2)+ND(5)).EQ.0)FAC1=0.
      IF((ND(3)+ND(6)).EQ.0)FAC2=0.
C
C     REFRESHING OF THE SKEWED/BORESIGHTED FHST DCM'S (REF. SECT. 4.6.3)
C
      DO 1200 I=1,3
      TDS2S1(I,I)=1.
 1200 TDS3S1(I,I)=1.
      TDS2S1(1,2)=PHI(2,3)
      TDS2S1(3,1)=PHI(2,2)
      TDS2S1(1,3)=-PHI(2,2)
      TDS2S1(2,1)=-PHI(2,3)
      TDS3S1(1,2)=PHI(3,3)
      TDS3S1(3,1)=PHI(3,2)
      TDS3S1(1,3)=-PHI(3,2)
      TDS3S1(2,1)=-PHI(3,3)
      CALL GMPRD(TDS2S1,T0S2S1,TS2S1,3,3,3)
      CALL GMPRD(TDS3S1,T0S3S1,TS3S1,3,3,3)
C
C     SUBOPTIMAL FILTER, LOOP FOR THE SIX STARS (REF. SECT. 4.6.4)
C
      NTR=NTR+1
      NZ=0
C
      CALL OBST
 1300 CONTINUE
C
      NZ=NZ+1
      IZ=NZ
      IF(NZ.GT.3) IZ=NZ-3
```

```
      DO 1400 I=1,2
 1400 NF(NZ,I)=ND(NZ)
      IF(ND(NZ).NE.1) GO TO 1405
      CALL SUBOPT(XDH,XDHO,LSTMES)
 1405 CONTINUE
C     PRINT 1410
C1410 FORMAT(' XD')
C     PRINT 1420,(XD(I),I=1,10)
C1420 FORMAT(10G12.4)
C
      IF(NZ.NE.6) GOTO 1300
C
C---- UPDATE IS NOT DONE IN THE FIRST FILTER STEP , BECAUSE STAR POSI-
C     TION MEASUREMENT DATA ARE NOT YET AVAILABLE ----------------------
C
      IF (.NOT.LSTMES) GOTO 2000
C
C     PREDICTION AND PREPARATION OF FILTER UPDATE DATA (REF. SECT.4.6.5)
C
      DO 1450 I=1,6
C     XDH6(I)=XDH(I)
 1450 XD6(I)=XD(I)
      CALL GMPRD(PH,XD6,XD1,3,6,1)
C     CALL GMPRD(PH,XDH6,XDH1,3,6,1)
      DO 1500 I=1,3
C     XDH(I)=XDH1(I)
      XD(I)=XD1(I)
      XD3(I)=XD6(I)
 1500 QD(I)=XD1(I)
      DO 49 I=1,10
C     XDHO(I)=XDH(I)
 49   XDH(I)=XD(I)
C     PRINT 1520
C1520 FORMAT(' XD3')
C     PRINT 155,(XD3(I),I=1,3)
      CALL GMPRD(PH4,XD3,QD(4),1,3,1)
C     PRINT 1530
C1530 FORMAT(' QD=')
C     PRINT 155,(QD(I),I=1,4)
      DO 1600 I=1,3
 1600 D(I)=D(I)+XD(I+3)
C
      DO 1700 I=2,3
      PHI(2,I)=PHI(2,I)+XD(I+5)
 1700 PHI(3,I)=PHI(3,I)+XD(I+7)
      DO 1701 I=2,3
      IF((ABS(PHI(2,I))).GT.PMAX) PHI(2,I)=PMAX*ABS(PHI(2,I))/PHI(2,I)
 1701  IF((ABS(PHI(3,I))).GT.PMAX) PHI(3,I)=PMAX*ABS(PHI(3,I))/PHI(3,I)
C
      IF(ITEST(8).EQ.0.AND.(D(1).LT..1.AND.D(2).LT..1.AND.D(3).LT..1.
     * AND.QD(1).LT..1.AND.QD(2).LT..1.AND.QD(3).LT..1)) GOTO 2000
      WRITE(6,TEST8)
 2000 CONTINUE
```

```
C
      LSTMES = .TRUE.
C
      IF(ITEST(10).EQ.1) ITEST(9)=1
      RETURN
      END
      SUBROUTINE SUBOPT(XDH,XDHO,LSTMES)
C
      LOGICAL LTHDIS,LSTMES
      REAL KA
      DIMENSION HM(2,10),YDD(2),H23(2,3),XDH(10),XDHO(10)
C
      COMMON /OBS/ HT(6,10),PH(3,6)
      COMMON/TRANS/TA_P_I(3,3),TD_P_I(3,3),DUMMY1(27)
      COMMON/KALMAN/ND(6),NF(6,2),NTR,AM(4,4),QOA(4),QMA(4),NZ,TM(3,3),
     *IZ,TS2S1(3,3),TS3S1(3,3),YD(2),H6(9,3),XD(10),CS(3,10,2),C(3,10,2)
     *,KA(3,10,2),PHI(3,3),FAC,FAC1,FAC2,PMAX,LTHDIS
      COMMON/CFHST/YRO(6,3),YMS(6,2),DUMMY(43)
      COMMON/TEST/ITEST(10)
      COMMON/COSUB/YMSA(6,2),TOS2S1(3,3),TOS3S1(3,3)
      COMMON/CSUBOP/TSETTL,EFLCH1,EFLCH2
      COMMON/COG2/YDA(2,6),IG
C
      NAMELIST/TEST7/NTR,NZ,YRN,YD,YDD,XD
C*********************************************************************
C     FHST DATA PROCESSING (REF. SECT. 4.6.4.4)
C*********************************************************************
C     YRO         THE COLUMNS OF YRO ARE THE STAR DIRECTION VECTORS
C                 IN THE IPS INERTIAL REFERENCE SYSTEM
C                 YRO IS COMPUTED IN 'FHST'
C     TM          DCM FROM THE I- TO THE P-SYSTEM BASED ON THE GYRO DATA
C                 TM IS COMPUTED IN 'KAFILT'
C     TS2S1,TS3S1  TRANSFORMATION MATRICES FROM THE BORESIGHTED SENSOR
C                   SYSTEM (S1) TO THE SKEWED SENSOR SYSTEM (S2,S3)
C                   AS USED IN THE ADF, BASED ON THE ESTIMATED OF THE
C                   MISALIGNMENTS
      IF (.NOT.LSTMES) GO TO 2000
C
C     DO 1120 J=1,3
C     HM(1,J)=HT(IZ*2-1,J)
C1120 HM(2,J)=HT(IZ*2,J)
CC
C     DO 1150 I=1,2
C     DO 1150 J=4,10
C1150 HM(I,J)=0.
CC
C----CCOMPUTE ESTIMATET MEASUREMENT DEVIATION : YDD = HM * XDH
CC
C     GOTO (1200,1300,1400),IZ
CC
C1200 DO 1250 I=1,2
C     YDD(I)=0.
C     DO 1250 J=1,3
```

```
C1250 YDD(I)=YDD(I)+HM(I,J)*XDH(J)
CC
C     GOTO 1500
C1300 HM(1,7)=HT(3,7)
C     HM(1,8)=HT(3,8)
C     HM(2,7)=HT(4,7)
C     HM(2,8)=HT(4,8)
C     DO 1350 I=1,2
C     YDD(I)=0.
CC
C     DO 1350 J=7,8
C1350 YDD(I)=YDD(I)+HM(I,J)*XDH(J)
CC
C     DO 1370 I=1,2
C     DO 1370 J=1,3
C1370 YDD(I)=YDD(I)+HM(I,J)*XDH(J)
CC
C     GOTO 1500
C1400 HM(1,9)=HT(5,9)
C     HM(1,10)=HT(5,10)
C     HM(2,9)=HT(6,9)
C     HM(2,10)=HT(6,10)
CC
C     DO 1450 I=1,2
C     YDD(I)=0.
C     DO 1450 J=9,10
C1450 YDD(I)=YDD(I)+HM(I,J)*XDH(J)
CC
C     DO 1470 I=1,2
C     DO 1470 J=1,3
C1470 YDD(I)=YDD(I)+HM(I,J)*XDH(J)
CC
C1500 CONTINUE
      IF((NZ.GT.3).AND.ND(IZ).NE.0) GOTO 1600
C
C**** COMPUTATION OF THE SUBOPTIMAL GAIN MATRIX (REF. SECT. 4.6.4.5) ***
C
      CALL GAINMA
 1600 CONTINUE
C
C**** STATE DEVIATION VECTOR UPDATE (REF. SECT. 4.6.4.6) ***************
C
      DO 1700 I=1,10
      DO 1700 J=1,2
 1700 XD(I)=XD(I)+KA(IZ,I,J)*YDA(J,IZ)
C1700 XD(I)=XD(I)+KA(IZ,I,J)*(YD(J)-YDD(J))
C
      IF(ITEST(7).NE.0)WRITE(6,TEST7)
C
 2000 CONTINUE
C
      RETURN
      END
```

```
C
```

GAINMA.FOR

```
      SUBROUTINE GAINMA
      REAL KA
      COMMON /OBS/ TH(6,10),PH(3,6)
      COMMON /CFHST/ YR0(6,3),YMS(6,2),DUMMY(43)
      COMMON/KALMAN/IYO(6),NF(6,2),NTR,AM(4,4),QOA(4),QMA(4),NZ,TM(3,3),
     *IZ,TS2S1(3,3),TS3S1(3,3),YD(2),H6(9,3),XD(10),CS(3,10,2),C(3,10,2)
     *,KA(3,10,2),DUM(3,3),FAC,FAC1,FAC2,PMAX,LTHDIS
      DIMENSION PO(6),RNI(6),QNI(6)
     1,DUM1(100),HPHTRO(6,6),IYO(6)
      DIMENSION P(6,6),QN(6,6),RN(6,6),H(6,6),HT(6,6),PHT(6,6),HPHT(6,6)
     *,HPHTR(6,6),HPHTRI(6,6),G(6,6),PHI(6,6),GH(6,6),IGH(6,6),PP(6,6)
     *,PHIT(6,6),PPPT(6,6),ID(6,6)
      REAL ID,IGH
C
      IF(INIT.GT.0)GO TO 5
      READ(17,*) PO,RNI,QNI
C     WRITE(2,*) PO,RNI,QNI
      DO 15 I=1,3
      DO 15 J=1,10
      DO 15 K=1,2
 15   KA(I,J,K)=0.0
      DO 16 I=1,6
      DO 16 J=1,6
      PHI(I,J)=0.0
 16   PHI(I,I)=1.0
      DO 10 I=1,6
      ID(I,I)=1.0
      QN(I,I)=QNI(I)
      P(I,I)=PO(I)
 10   CONTINUE
      DO 20 I=1,6
      RN(I,I)=RNI(I)
  20   CONTINUE
C     WRITE(10) T,Q,X,YMS
      INIT=1
   5  CONTINUE
      IF(NZ.GT.1.AND.IYO(1).EQ.1) GO TO 999
      IF(NZ.GT.2.AND.IYO(1).EQ.0) GO TO 999
      DO 17 I=1,6
      DO 17 J=1,6
17    H(I,J)=TH(I,J)
      DO 30 I=1,3
      DO 30 J=1,6
      PHI(I,J)=PH(I,J)
  30  CONTINUE
```

```
      DO 24 J=0,2
      IF(IYO(J+1).EQ.1) GOTO 24
      DO 26 I=1,2
 26   RN(I+J*2,I+J*2)=99999.
  24  CONTINUE
C
C
C     ERROR COVARIANCE EXTRAPOLATION
C
C     P+ = P-
      CALL MULMAT(PHI,P,PP,6,6,6)
      CALL TRMAT(PHI,PHIT,6,6)
      CALL MULMAT(PP,PHIT,PPPT,6,6,6)
      CALL ADDMAT(PPPT,QN,P,6,6)
C
C     END COVARIANCE EXTRAPOLATION
C
C     GAIN MATRIX GENERATION
C
      CALL TRMAT(H,HT,6,6)
      CALL MULMAT(P,HT,PHT,6,6,6)
      CALL MULMAT(H,PHT,HPHT,6,6,6)
      CALL ADDMAT(HPHT,RN,HPHTR,6,6)
      DO 31 J=1,6
      DO 31 K=1,6
      HPHTRO(J,K)=HPHTR(J,K)
 31   CONTINUE
      IZERO=0
      CALL MATINV(HPHTRO,HPHTRI,6,6,IER)
      CALL MULMAT(PHT,HPHTRI,G,6,6,6)
C
C      END GAIN GENERATION
C
C
C     ERROR COVARIANCE UPDATE
C
      CALL MULMAT(G,H,GH,6,6,6)
      DO 32 L=1,6
      DO 32 K=1,6
      GH(L,K)=-GH(L,K)
 32   CONTINUE
      CALL ADDMAT(ID,GH,IGH,6,6)
      CALL MULMAT(IGH,P,P,6,6,6)
C
C     END COVARIANCE UPDATE
C
      DO 998 JZ=0,2
      DO 998 I=1,6
      DO 998 J=1,2
 998  KA(JZ+1,I,J)=G(I,J+JZ*2)
 999  CONTINUE
      RETURN
      END
```

OBSTE.FOR

```
      SUBROUTINE OBST
      REAL KA
      DIMENSION YRA(3),YRP(3),T(3,3),YRS(3),YRN(2),H(14),H3(2,9)
       DIMENSION H23(2,3)
C
       COMMON/OBS/ HT(6,10),PH(3,6)
      COMMON/TRANS/TA_P_I(3,3),TD_P_I(3,3),DUMMY1(27)
      COMMON/KALMAN/ND(6),NF(6,2),NTR,AM(4,4),QOA(4),QMA(4),NZ,TM(3,3),
     *IZ,TS2S1(3,3),TS3S1(3,3),YD(2),H6(9,3),XD(10),CS(3,10,2),C(3,10,2)
     *,KA(3,10,2),PHI(3,3),FAC,FAC1,FAC2,PMAX,LTHDIS
      COMMON/CFHST/YRO(6,3),YMS(6,2),DUMMY(43)
      COMMON/COFHST/ YSJ(6,3)
      COMMON/COSUB/YMSA(6,2),TOS2S1(3,3),TOS3S1(3,3)
      COMMON/COG2/YDA(2,6),IG
C
      NAMELIST/TEST7/NTR,NZ,YRN,YD,YDD,XD
C*********************************************************************
C     FHST DATA PROCESSING (REF. SECT. 4.6.4.4)
C*********************************************************************
C     YRO         THE COLUMNS OF YRO ARE THE STAR DIRECTION VECTORS
C                 IN THE IPS INERTIAL REFERENCE SYSTEM
C                 YRO IS COMPUTED IN 'FHST'
C     TM          DCM FROM THE I- TO THE P-SYSTEM BASED ON THE GYRO DATA
C                 TM IS COMPUTED IN 'KAFILT'
C     TS2S1,TS3S1  TRANSFORMATION MATRICES FROM THE BORESIGHTED SENSOR
C                   SYSTEM (S1) TO THE SKEWED SENSOR SYSTEM (S2,S3)
C                   AS USED IN THE ADF, BASED ON THE ESTIMATED OF THE
C                   MISALIGNMENTS
      DO 999 IZ=1,3
C
      DO 50 I=1,3
   50 YRA(I)=YRO(IZ,I)
C
C---- YRP = STAR DIRECTION IN THE P-SYSTEM BASED ON THE GYRO DATA
C           YRP = TM * YRA
C
      CALL GMPRD(TM,YRA,YRP,3,3,1)
C
      GOTO (300,400,500),IZ
C
C---- TRANSFORMATION MATRIX FROM P TO S1 SYSTEM
C
  300 DO 350 I=1,3
      DO 350 J=1,3
  350 T(I,J)=0.
      DO 370 I=1,3
  370 T(I,I)=1.
C
      GOTO 600
C
C---- TRANSFORMATION MATRIX FROM S1 TO S2 SYSTEM
C
```

```
  400 DO 450 I=1,3
      DO 450 J=1,3
  450 T(I,J)=TS2S1(I,J)
C
      GOTO 600
C
C---- TRANSFORMATION MATRIX FROM S1 TO S3 SYSTEM
C
  500 DO 550 I=1,3
      DO 550 J=1,3
  550 T(I,J)=TS3S1(I,J)
C
  600 CONTINUE
C
      CALL GMPRD(T,YRP,YRS,3,3,1)
C
      CX=1./YRS(1)
      CX2=CX*CX
C
C---- YRN = NOMINAL MEASUREMENT VECTOR; THE NOMINAL P-SYSTEM IS
C                          DETERMINED FROM THE GYRO MEASUREMENTS--------

      DO 700 I=1,2
  700 YRN(I)=YRS(I+1)*CX
C
C************** MEASUREMENT DEVIATION VECTOR ****************************
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  NEXT 2 CARDS FOR YMSA MOVED UP TO HERE BY MARK WEST
C
      YMSA(IZ,1)=YMS(IZ,1)
      YMSA(IZ,2)=YMS(IZ,2)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      DO 800 I=1,2
      YD(I) = YMSA(IZ,I) - YRN(I)
 800  YDA(I,IZ)=YD(I)
C 800 YD(I) = YMSA(IZ,I)
  806 CONTINUE
C
C**** COMPUTATION OF THE OBSERVATION MATRIX (REF. SECT.4.6.4.4) ********
C
      H(1)=-YRN(1)*CX
      H(2)=-YRN(2)*CX
      H(3)=H(1)-CX*PHI(IZ,3)
      H(4)=CX+H(1)*PHI(IZ,3)
      H(5)=-H(1)*PHI(IZ,2)
      H(6)=H(2)+CX*PHI(IZ,2)
      H(7)=H(2)*PHI(IZ,3)
      H(8)=CX-H(2)*PHI(IZ,2)
      DO 1000 I=1,3
```

```
      H(I+8)=H(3)*T(1,I)+H(4)*T(2,I)+H(5)*T(3,I)
 1000 H(I+11)=H(6)*T(1,I)+H(7)*T(2,I)+H(8)*T(3,I)
C
      DO 1100 I=1,3
      H3(1,I)=H(9)*YRA(I)
      H3(1,I+3)=H(10)*YRA(I)
      H3(1,I+6)=H(11)*YRA(I)
      H3(2,I)=H(12)*YRA(I)
      H3(2,I+3)=H(13)*YRA(I)
 1100 H3(2,I+6)=H(14)*YRA(I)
      CALL GMPRD(H3,H6,H23,2,9,3)
      DO 777 J=1,3
      HT(IZ*2-1,J)=H23(1,J)
 777  HT(IZ*2,J)=H23(2,J)
      GOTO (1200,1300,1400) IZ
 1300 HT(3,7)=YRS(2)*YRS(3)*CX2
      HT(3,8)=-YRS(2)*YRS(2)*CX2-1.
      HT(4,7)=YRS(3)*YRS(3)*CX2+1.
      HT(4,8)=-HT(3,7)
      GOTO 1200
1400  HT(5,9)=YRS(2)*YRS(3)*CX2
      HT(5,10)=-YRS(2)*YRS(2)*CX2-1.
      HT(6,9)=YRS(3)*YRS(3)*CX2+1.
      HT(6,10)=-HT(5,9)
1200  CONTINUE
 999  CONTINUE
      RETURN
      END
```

IPS EKF Input Data

IPSE.DAT

```
.5E-5 .5E-5 .5E-5
.5E-5 .5E-5 .5E-5 .5E-5 .5E-5 .5E-5
2
0. 1 0. 0. 0.
100. 2 0. 0. 0.
1.45E-5 1.45E-5 1.45E-5
.007 .005 .005 1.E-7 1.E-7 1.E-7 4.85E-6 4.85E-6 4.85E-6 4.85E-6 4.85E-6 4.85E-6
1.212E-6 1.212E-6 1.212E-6 1.E-9 1.E-9 1.E-9
```

ASTP.DAT

```
 &NFILT
 LATTUP=T, P_I=-1500., 900., 900., NTR=-2, JSUBOP=3
 MODE(1)=1,    BJAS(1,1)=0.0,    SIGMAN(1,1)=.75,    YS(1,1)=0.,
               BJAS(1,2)=0.0,    SIGMAN(1,2)=.75,    YS(1,2)=0.,
 MODE(2)=1,    BJAS(2,1)=0.0,    SIGMAN(2,1)=1.0,    YS(2,1)=0.,
               BJAS(2,2)=0.0,    SIGMAN(2,2)=0.0,    YS(2,2)=0.,
 MODE(3)=0,    BJAS(3,1)=0.0,    SIGMAN(3,1)=1.0,    YS(3,1)=0.,
               BJAS(3,2)=0.0,    SIGMAN(3,2)=1.0,    YS(3,2)=0.,
 MODE(4)=0,    BJAS(4,1)=0.0,    SIGMAN(4,1)=.75,    YS(4,1)=0.,
               BJAS(4,2)=0.0,    SIGMAN(4,2)=.75,    YS(4,2)=0.,
 MODE(5)=0,    BJAS(5,1)=0.0,    SIGMAN(5,1)=1.,    YS(5,1)=0.,
               BJAS(5,2)=0.0,    SIGMAN(5,2)=1.,    YS(5,2)=0.,
 MODE(6)=0,    BJAS(6,1)=0.0,    SIGMAN(6,1)=1.,    YS(6,1)=0.,
               BJAS(6,2)=0.0,    SIGMAN(6,2)=1.,    YS(6,2)=0.,
 ITEST=0,0,0,0,0,0,0,0,0,0, LTHDIS=T, ALPHO=12.0, BETA=-45,
 FAC=1.,  JREAD=16,  JSST=2,
 DEPHIX=0.0, 0.0,   DEPHIY=0.0,0.0,   DEPHIZ=0.0,0.0,
 DRTHX=  2*0.,   DRTHY= 2*0.,   DRTHZ= 0., 0.,
 TSETTL=60.0,  EFLCH1=2000.0,  EFLCH2=90.0,
 LFHST=1, ALP11=7.36, ALP12=127.158, ALP2=0.21875, FDTTR=2.38, WEFF=25.,
 FOVR=72.5,
 FOVDY=0.0, FOVDZ=0.0, DSUN=1920.0, DNEGST=200.0, DSCAN=36.75, RNTNS=0.3,
 &END
```

APPENDIX F

Second-Order Kalman Filter Listing and Data

PRECEDING PAGE BLANK NOT FILMED

TREEBS.COM

```
$ JOB
$ SET DEF [ROBINSON.TREETOPS]
$    ASSIGN/NOLOG IPSREV7.INT FOR001
$    ASSIGN/NOLOG IPSREV7.OUT FOR009
$    ASSIGN/NOLOG IPSREV7.RES FOR003
$    ASSIGN/NOLOG IPSREV7.MAT FOR004
$    ASSIGN/NOLOG IPSREV7.LOG FOR006
$    ASSIGN/NOLOG IPSREV7.MDK FOR019
$    ASSIGN/NOLOG IPSREV7.FLX FOR011
$    ASSIGN/NOLOG IPSREV7.FLN FOR013
$    ASSIGN/NOLOG IPSREV7.AUX FOR039
$    ASSIGN/NOLOG IPSREV7.XRF FOR012
$    ASSIGN/NOLOG TCATEMP.DAT  FOR025
$    ASS ASTP FOR015
$    ASS MSUBCO FOR016
$    ASS IPSE   FOR017
$    ASS PLOTS  FOR018
$    ASS/NOLOG DEFAULT.DFF FOR085
$    RUN [ROBINSON.TREETOPS.THE]USDC
$    DEASSIGN FOR001
$    DEASSIGN FOR009
$    DEASSIGN FOR003
$    DEASSIGN FOR004
$    DEASSIGN FOR006
$    DEASSIGN FOR019
$    DEASSIGN FOR011
$    DEASSIGN FOR013
$    DEASSIGN FOR039
$    DEASSIGN FOR012
$    DEASSIGN FOR025
$    DEASSIGN FOR085
$    RENAME TCATEMP.DAT IPSREV7.MAT
$    DIR/DATE/SIZE IPSREV7.OUT
$ COPY IPSREV7.OUT IPSREV7.PLT
```

ADFS.FOR

```
      SUBROUTINE LADF
C
      REAL KA
      LOGICAL LTHDIS,LSTMES
      DIMENSION TDS2S1(3,3),TDS3S1(3,3),XI(3),QMA(4),PH4(3)
      DIMENSION A(4,4),QMS2(4),H4(9,3),AZ(4,4),CMS(4),XD1(3),PH1(4,3)
      DIMENSION A3(3,3),XD6(6),XD3(3),XDH(10)
CCC   DIMENSION A3(3,3),XD6(6),XD7(3),XD3(3),XDH(10),XDHO(10)
CCC  1,XDH6(6),XDH1(3)
C
C-------------------------------------------------------
C
C NEW ARRAYS ADDED FOR USE BY DELAYS ADDED BY ADAMS & WEST
C
CCC   DIMENSION QD11(4),QD12(4),D11(3),D12(3)
CCC   DATA QD11,D11/ 4*0.0,  3*0.0 /
CCC   DIMENSION QD(4),D(3)
CCC   DATA QD,XD1,XD3,XDHO,XDH/30*0.0/
C
C-------------------------------------------------------
C
       COMMON /OBS/ HT(6),PH(3,6),AT(6,6),P(6,6),HP(6,6)
      COMMON /ADF/ D(3),QD(4),QM(4),QO(4),CM(4),Q(4),VSTN(6)
CCCC  COMMON /FILTER/ DI(3),QDI(4),QM(4),QO(4),CM(4),Q(4)
      COMMON /KALMAN/ ND(6),NF(6,2),NTR,AM(4,4),QOA(4),QMS(4),NZ,
     *TM(3,3),IZ,TS2S1(3,3),TS3S1(3,3),YD(2),H6(9,3),XD(10),
     * CS(3,10,2),C(3,10,2),KA(3,10,2),PHI(3,3),FAC,FAC1,FAC2,PMAX,
     * LTHDIS
      COMMON /TEST/ ITEST(10)
      COMMON/COSUB/YMSA(6,2),TOS2S1(3,3),TOS3S1(3,3)
C
      DATA TDS2S1,TDS3S1 /18*0./,LSTMES/.FALSE./
      NAMELIST/TEST8/NTR,XD,AM,D,PHI,CM,QD,QO,QM,QOA,QMA
C
C     REINITIALIZATION OF THE STATE DEVIATION VECTOR AT THE BEGINNING OF
C     EACH FILTER INTERVAL (INPUT FOR SUBOPT FOR THE FIRST STAR),
C     BECAUSE THE FILTER EQUATIONS ARE PROGRAMMED FOR THE STATE VECTOR
C     UPDATES, NOT FOR THE STATE VECTOR ITSELF
C
C
C-------------------------------------------------------------------
C
C    INITIALIZATION ADDED BY ADAMS & WEST
C
      DO 10 I=1,4
      QOA(I) = QO(I)
      QMA(I) = Q(I)
 10   CONTINUE
      CALL SSTM(CM(2),CM(3),CM(4),CM(1),AM)
C     CALL MAVE(AM,QD,QDI,4)
```

```
C     DO 15 I=1,3
C15   DI(I)=D(I)
C-----------------------------------------------------------------------
C
      DO 50 I=1,10
   50 XD(I)=0.
C
C
C     MODIFICATION OF THE SUBSTATE TRANSITION MATRIX (REF.SECT.4.6.1)
C
   70 CQ=1./QOA(4)
      DO 100 I=1,3
      XI(I)=QOA(I)*CQ
      DO 100 J=1,4
  100 PH1(J,I)=AM(J,I)-AM(J,4)*XI(I)
C     PRINT 150
C 150 FORMAT(' PH1')
C     PRINT 155,((PH1(J,I),J=1,4),I=1,3)
C 155 FORMAT(4G20.6)
      DO 200 I=1,3
  200 PH4(I)=PH1(4,I)
C     PRINT 220
C 220 FORMAT(' PH4')
C     PRINT 155,(PH4(I),I=1,3)
C
C     COMPUTATION OF THE STATE TRANSITION MATRIX (REF. SECT.4.6.1 C)
C
      CP=-.5
      DO 300 I=1,3
      PH(I,I+3)=QMA(4)*CP
      DO 300 J=1,3
  300 PH(I,J)=PH1(I,J)
      PH(3,5)=QMA(1)*CP
      PH(1,6)=QMA(2)*CP
      PH(2,4)=QMA(3)*CP
      PH(1,5)=-PH(2,4)
      PH(2,6)=-PH(3,5)
      PH(3,4)=-PH(1,6)
C     PRINT 350
C 350 FORMAT(' PH')
C     NORMALIZATION OF MEAN QUATERNION VECTORS (REF. SECT.4.6.1 D)
C
      CP=0.
      DO 400 I=1,4
  400 CP=CP+QMA(I)*QMA(I)
      CP=1./SQRT(CP)
      DO 500 I=1,4
  500 QMS(I)=QMA(I)*CP
C
C     COMPUTATION OF THE DIRECTION COSINE MATRIX FROM THE IPS INERTIAL
C     REFERENCE SYSTEM TO THE IPS PLATFORM SYSYTEM RESULTING FROM THE
C     MEAN QUATERNION VECTOR OF THE PREVIOUS FILTER INTERVAL
C     (REF. SECT. 4.6.1 E)
```

```
C
      DO 600 I=1,4
      DO 600 J=1,4
  600 A(I,J)=QMS(I)*QMS(J)
      DO 700 I=1,3
  700 TM(I,I)=2.*(A(4,4)+A(I,I))-1.
      TM(1,2)=2.*(A(1,2)+A(3,4))
      TM(1,3)=2.*(A(3,1)-A(2,4))
      TM(2,1)=2.*(A(2,1)-A(3,4))
      TM(2,3)=2.*(A(2,3)+A(1,4))
      TM(3,1)=2.*(A(1,3)+A(2,4))
      TM(3,2)=2.*(A(2,3)-A(1,4))
C
C
C     DEFINITION OF THE CORRECTION FACTOR FOR THE ESTIMATION
C     OF THE FHST THERMAL DISTORTIONS
C
      IF(LTHDIS)GOTO 1170
      FAC1=0.
      FAC2=0.
 1170 IF((ND(2)+ND(5)).EQ.0)FAC1=0.
      IF((ND(3)+ND(6)).EQ.0)FAC2=0.
C
C     REFRESHING OF THE SKEWED/BORESIGHTED FHST DCM'S (REF. SECT. 4.6.3)
C
      DO 1200 I=1,3
      TDS2S1(I,I)=1.
 1200 TDS3S1(I,I)=1.
      TDS2S1(1,2)=PHI(2,3)
      TDS2S1(3,1)=PHI(2,2)
      TDS2S1(1,3)=-PHI(2,2)
      TDS2S1(2,1)=-PHI(2,3)
      TDS3S1(1,2)=PHI(3,3)
      TDS3S1(3,1)=PHI(3,2)
      TDS3S1(1,3)=-PHI(3,2)
      TDS3S1(2,1)=-PHI(3,3)
      CALL GMPRD(TDS2S1,TOS2S1,TS2S1,3,3,3)
      CALL GMPRD(TDS3S1,TOS3S1,TS3S1,3,3,3)
C
C     SUBOPTIMAL FILTER, LOOP FOR THE SIX STARS (REF. SECT. 4.6.4)
C
      NTR=NTR+1
      NZ=0
C
C     CALL OBST
 1300 CONTINUE
C
      NZ=NZ+1
      IZ=NZ
      IF(NZ.GT.3) IZ=NZ-3
      DO 1400 I=1,2
 1400 NF(NZ,I)=ND(NZ)
      IF(ND(NZ).NE.1) GO TO 1405
```

```
      CALL SUBOPT(XDH,XDHO,LSTMES)
 1405 CONTINUE
C     PRINT 1410
C1410 FORMAT(' XD')
C     PRINT 1420,(XD(I),I=1,10)
C1420 FORMAT(10G12.4)
C
      IF(NZ.NE.6) GOTO 1300
C
C---- UPDATE IS NOT DONE IN THE FIRST FILTER STEP , BECAUSE STAR POSI-
C     TION MEASUREMENT DATA ARE NOT YET AVAILABLE ----------------------
C
      IF (.NOT.LSTMES) GOTO 2000
C
C     PREDICTION AND PREPARATION OF FILTER UPDATE DATA (REF. SECT.4.6.5)
C
C      DO 1450 I=1,6
C     XDH6(I)=XDH(I)
C 1450 XD6(I)=XD(I)
C      CALL GMPRD(PH,XD6,XD1,3,6,1)
C     CALL GMPRD(PH,XDH6,XDH1,3,6,1)
      DO 1500 I=1,3
C     XDH(I)=XDH1(I)
C      XD(I)=XD1(I)
C      XD3(I)=XD6(I)
 1500 QD(I)=XD(I)
C      DO 49 I=1,10
C     XDHO(I)=XDH(I)
C 49   XDH(I)=XD(I)
C     PRINT 1520
C1520 FORMAT(' XD3')
C     PRINT 155,(XD3(I),I=1,3)
C      CALL GMPRD(PH4,XD3,QD(4),1,3,1)
C     PRINT 1530
C1530 FORMAT(' QD=')
C     PRINT 155,(QD(I),I=1,4)
      DO 1600 I=1,3
 1600 D(I)=D(I)+XD(I+3)
C
      DO 1700 I=2,3
      PHI(2,I)=PHI(2,I)+XD(I+5)
 1700 PHI(3,I)=PHI(3,I)+XD(I+7)
      DO 1701 I=2,3
      IF((ABS(PHI(2,I))).GT.PMAX) PHI(2,I)=PMAX*ABS(PHI(2,I))/PHI(2,I)
 1701  IF((ABS(PHI(3,I))).GT.PMAX) PHI(3,I)=PMAX*ABS(PHI(3,I))/PHI(3,I)
C
      IF(ITEST(8).EQ.0.AND.(D(1).LT..1.AND.D(2).LT..1.AND.D(3).LT..1.
     * AND.QD(1).LT..1.AND.QD(2).LT..1.AND.QD(3).LT..1)) GOTO 2000
      WRITE(6,TEST8)
 2000 CONTINUE
C
      LSTMES = .TRUE.
C
```

```
      IF(ITEST(10).EQ.1) ITEST(9)=1
      RETURN
      END
      SUBROUTINE SUBOPT(XDH,XDHO,LSTMES)
C
      LOGICAL LTHDIS,LSTMES
      REAL KA
      DIMENSION YDD(2),XDH(10),XDHO(10)
C
      COMMON /OBS/ HT(6),PH(3,6),AT(6,6),P(6,6),HP(6,6)
      COMMON/TRANS/TA_P_I(3,3),TD_P_I(3,3),DUMMY1(27)
      COMMON/KALMAN/ND(6),NF(6,2),NTR,AM(4,4),QOA(4),QMA(4),NZ,TM(3,3),
     *IZ,TS2S1(3,3),TS3S1(3,3),YD(2),H6(9,3),XD(10),CS(3,10,2),C(3,10,2)
     *,KA(3,10,2),PHI(3,3),FAC,FAC1,FAC2,PMAX,LTHDIS
      COMMON/CFHST/YRO(6,3),YMS(6,2),DUMMY(43)
      COMMON/TEST/ITEST(10)
      COMMON/COSUB/YMSA(6,2),TOS2S1(3,3),TOS3S1(3,3)
      COMMON/CSUBOP/TSETTL,EFLCH1,EFLCH2
      COMMON/COG2/YDA(2,6),IG
C
      NAMELIST/TEST7/NTR,NZ,YRN,YD,YDD,XD
C*********************************************************************
C     FHST DATA PROCESSING (REF. SECT. 4.6.4.4)
C*********************************************************************
C     YRO         THE COLUMNS OF YRO ARE THE STAR DIRECTION VECTORS
C                 IN THE IPS INERTIAL REFERENCE SYSTEM
C                 YRO IS COMPUTED IN 'FHST'
C     TM          DCM FROM THE I- TO THE P-SYSTEM BASED ON THE GYRO DATA
C                 TM IS COMPUTED IN 'KAFILT'
C     TS2S1,TS3S1  TRANSFORMATION MATRICES FROM THE BORESIGHTED SENSOR
C                   SYSTEM (S1) TO THE SKEWED SENSOR SYSTEM (S2,S3)
C                   AS USED IN THE ADF, BASED ON THE ESTIMATED OF THE
C                   MISALIGNMENTS
      IF (.NOT.LSTMES) GO TO 2000
C
      IF((NZ.GT.3).AND.ND(IZ).NE.0) GOTO 1600
C
C**** COMPUTATION OF THE SUBOPTIMAL GAIN MATRIX (REF. SECT. 4.6.4.5) ***
C
      CALL GAINMA
 1600 CONTINUE
C
C
C---- COMPUTE ESTIMATET MEASUREMENT DEVIATION : YDD = HM * XDH
C
      GOTO (1200,1300,1400),IZ
C
 1200 YDD(1)=HT(1)
      YDD(2)=HT(2)
C
      GOTO 1500
 1300 YDD(1)=HT(3)
      YDD(2)=HT(4)
```

```
C
      GOTO 1500
 1400 YDD(1)=HT(5)
      YDD(2)=HT(6)
C
 1500 CONTINUE
C     IF((NZ.GT.3).AND.ND(IZ).NE.0) GOTO 1600
C
C**** COMPUTATION OF THE SUBOPTIMAL GAIN MATRIX (REF. SECT. 4.6.4.5) ***
C
C     CALL GAINMA
C1600 CONTINUE
C
C**** STATE DEVIATION VECTOR UPDATE (REF. SECT. 4.6.4.6) ***************
C
      DO 1700 I=1,10
      DO 1700 J=1,2
 1700 XD(I)=XD(I)+KA(IZ,I,J)*(YDA(J,IZ)-YDD(J))
C
      IF(ITEST(7).NE.0)WRITE(6,TEST7)
C
 2000 CONTINUE
C
      RETURN
      END
```

GAINS.FOR

```
       SUBROUTINE GAINMA
      REAL KA
      COMMON/ADF/ DD(3),QD(4),QM(4),QO1(4),CM1(4),Q(4),VSTN(6)
      COMMON/OBS/ TH(6),PH(3,6),AT(6,6),P(6,6),HP(6,6)
      COMMON /CFHST/ YR0(6,3),YMS(6,2),DUMMY(43)
      COMMON/KALMAN/IYO(6),NF(6,2),NTR,AM(4,4),QOA(4),QMA(4),NZ,TM(3,3),
     *IZ,TS2S1(3,3),TS3S1(3,3),YD(2),H6(9,3),XD(10),CS(3,10,2),C(3,10,2)
     *,KA(3,10,2),DUM(3,3),FAC,FAC1,FAC2,PMAX,LTHDIS
      DIMENSION PO(6),RNI(6),QNI(6)
     1,DUM1(100),HPHTRO(6,6),IYO(6)
      DIMENSION QN(6,6),RN(6,6),H(6,6),HT(6,6),PHT(6,6),HPHT(6,6)
     *,HPHTR(6,6),HPHTRI(6,6),G(6,6),PHI(6,6),GH(6,6),IGH(6,6),PP(6,6)
     *,PHIT(6,6),PPPT(6,6),ID(6,6)
      DIMENSION HPHTRAT(6,6),DUM2(100),DDM(6,6)
      REAL ID,IGH
C
      IF(INIT.GT.0)GO TO 5
      READ(17,*) PO,RNI,QNI
C     WRITE(2,*) PO,RNI,QNI
      DO 15 I=1,3
      DO 15 J=1,10
```

```
      DO 15 K=1,2
 15   KA(I,J,K)=0.0
      DO 16 I=1,6
      DO 16 J=1,6
      PHI(I,J)=0.0
 16   PHI(I,I)=1.0
      DO 10 I=1,6
      ID(I,I)=1.0
      QN(I,I)=QNI(I)
      P(I,I)=PO(I)
 10   CONTINUE
      DO 20 I=1,6
      RN(I,I)=RNI(I)
  20   CONTINUE
      INIT=1
C     WRITE(18) INIT,Q,YMS,DD,(P(I,I),I=1,6),QD
      INIT=1
   5  CONTINUE
      IF(NZ.GT.1.AND.IYO(1).EQ.1) GO TO 999
      IF(NZ.GT.2.AND.IYO(1).EQ.0) GO TO 999
      DO 30 I=1,3
      DO 30 J=1,6
 30   PHI(I,J)=PH(I,J)
C     ERROR COVARIANCE EXTRAPOLATION
C
C     P+ = P-
      CALL MULMAT(PHI,P,PP,6,6,6)
      CALL TRMAT(PHI,PHIT,6,6)
      CALL MULMAT(PP,PHIT,PPPT,6,6,6)
      CALL ADDMAT(PPPT,QN,P,6,6)
C
C     END COVARIANCE EXTRAPOLATION
      CALL OBST
      DO 17 I=1,6
      DO 17 J=1,6
17    H(I,J)=HP(I,J)
C     DO 30 I=1,3
C     DO 30 J=1,6
C     PHI(I,J)=PH(I,J)
C 30  CONTINUE
      DO 24 J=0,2
      IF(IYO(J+1).EQ.1) GOTO 24
      DO 26 I=1,2
 26   RN(I+J*2,I+J*2)=99999.
  24  CONTINUE
C
C
C     ERROR COVARIANCE EXTRAPOLATION
C
C     P+ = P-
C     CALL MULMAT(PHI,P,PP,6,6,6)
C     CALL TRMAT(PHI,PHIT,6,6)
C     CALL MULMAT(PP,PHIT,PPPT,6,6,6)
```

```
C     CALL ADDMAT(PPPT,QN,P,6,6)
C
C     END COVARIANCE EXTRAPOLATION
C
C     GAIN MATRIX GENERATION
C
      CALL TRMAT(H,HT,6,6)
      CALL MULMAT(P,HT,PHT,6,6,6)
      CALL MULMAT(H,PHT,HPHT,6,6,6)
      CALL ADDMAT(HPHT,RN,HPHTR,6,6)
      CALL ADDMAT(HPHTR,AT,HPHTRAT,6,6)
      DO 31 J=1,6
      DO 31 K=1,6
      HPHTRO(J,K)=HPHTRAT(J,K)
 31   CONTINUE
      IZERO = 0
      CALL MATINV(HPHTRO,HPHTRI,6,6,IER)
      CALL GMPRD(HPHTRO,HPHTRI,DDM,6,6,6)
      IF(IER.NE.0) PRINT IER

      CALL MULMAT(PHT,HPHTRI,G,6,6,6)
C
C      END GAIN GENERATION
C
C
C     ERROR COVARIANCE UPDATE
C
      CALL MULMAT(G,H,GH,6,6,6)
      DO 32 L=1,6
      DO 32 K=1,6
      GH(L,K)=-GH(L,K)
 32   CONTINUE
      CALL ADDMAT(ID,GH,IGH,6,6)
      CALL MULMAT(IGH,P,P,6,6,6)
C
C     END COVARIANCE UPDATE
C
      DO 998 JZ=0,2
      DO 998 I=1,6
      DO 998 J=1,2
 998  KA(JZ+1,I,J)=G(I,J+JZ*2)
 999  CONTINUE
      RETURN
      END
```

OBSTS.FOR

```
      SUBROUTINE OBST
      REAL KA
```

```
      DIMENSION YRA(3),YRP(3),T(3,3),YRS(3),YRN(2)
      DIMENSION H2P(6,6),HPP(6,6),H2PO(6,6),H2PP(6,6),HPOP(6,6)
      DIMENSION HPHP(6,6),HPHP1(6,6),HPHP2(6,6)
      DIMENSION A(9),B(3,9),SH(3),BH(2),DH1(3),DH2(3),
     1DH3(3),DDH1(9),DDH2(9),DDH3(9),PH1(3),PH2(3),DPH1(9),DPH2(9),
     2PA(9,3),DPA(9,9),DUM(9),FUM(3)
       COMMON /OBS/ HT(6),PH(3,6),AT(6,6),PM(6,6),HP(6,6)
      COMMON/TRANS/TA_P_I(3,3),TD_P_I(3,3),DUMMY1(27)
      COMMON/KALMAN/ND(6),NF(6,2),NTR,AM(4,4),QOA(4),QMA(4),NZ,TM(3,3),
     *IZ,TS2S1(3,3),TS3S1(3,3),YD(2),H6(9,3),XD(10),CS(3,10,2),C(3,10,2)
     *,KA(3,10,2),PHI(3,3),FAC,FAC1,FAC2,PMAX,LTHDIS
      COMMON/CFHST/YRO(6,3),YMS(6,2),DUMMY(43)
      COMMON/COFHST/ YSJ(6,3)
      COMMON/COSUB/YMSA(6,2),TOS2S1(3,3),TOS3S1(3,3)
      COMMON/COG2/YDA(2,6),IG
C
      NAMELIST/TEST7/NTR,NZ,YRN,YD,YDD,XD
C*************************************************************************
C     FHST DATA PROCESSING (REF. SECT. 4.6.4.4)
C*************************************************************************
C     YRO         THE COLUMNS OF YRO ARE THE STAR DIRECTION VECTORS
C                 IN THE IPS INERTIAL REFERENCE SYSTEM
C                 YRO IS COMPUTED IN 'FHST'
C     TM          DCM FROM THE I- TO THE P-SYSTEM BASED ON THE GYRO DATA
C                 TM IS COMPUTED IN 'KAFILT'
C     TS2S1,TS3S1  TRANSFORMATION MATRICES FROM THE BORESIGHTED SENSOR
C                   SYSTEM (S1) TO THE SKEWED SENSOR SYSTEM (S2,S3)
C                   AS USED IN THE ADF, BASED ON THE ESTIMATED OF THE
C                   MISALIGNMENTS
C
      Q4S=QMA(4)*QMA(4)
      Q1S=QMA(1)*QMA(1)
      Q2S=QMA(2)*QMA(2)
      Q3S=QMA(3)*QMA(3)
      A(1)=1.-2.*(Q2S+Q3S)
      A(2)=2.*(QMA(1)*QMA(2)+QMA(3)*QMA(4))
      A(3)=2.*(QMA(1)*QMA(3)-QMA(2)*QMA(4))
      A(4)=2.*(QMA(1)*QMA(2)-QMA(3)*QMA(4))
      A(5)=1.-2.*(Q1S+Q3S)
      A(6)=2.*(QMA(2)*QMA(3)+QMA(1)*QMA(4))
      A(7)=2.*(QMA(1)*QMA(3)+QMA(2)*QMA(4))
      A(8)=2.*(QMA(2)*QMA(3)-QMA(1)*QMA(4))
      A(9)=1.-2.*(Q1S+Q2S)
      PA(1,1)=0.
      PA(1,2)=-4.*QMA(2)
      PA(1,3)=-4.*QMA(3)
      PA(2,1)=2.*(QMA(2)-QMA(1)*QMA(3)/QMA(4))
      PA(2,2)=2.*(QMA(1)-QMA(2)*QMA(3)/QMA(4))
      PA(2,3)=2.*(QMA(4)-Q3S/QMA(4))
      PA(3,1)=2.*(QMA(3)+QMA(1)*QMA(2)/QMA(4))
      PA(3,2)=-2.*(QMA(4)-Q2S/QMA(4))
      PA(3,3)=2.*(QMA(1)+QMA(2)*QMA(3)/QMA(4))
      PA(4,1)=2.*(QMA(2)+QMA(1)*QMA(3)/QMA(4))
```

```
PA(4,2)=2.*(QMA(1)+QMA(2)*QMA(3)/QMA(4))
PA(4,3)=-2.*(QMA(4)-Q3S/QMA(4))
PA(5,1)=-4.*QMA(1)
PA(5,2)=0.
PA(5,3)=-4.*QMA(3)
PA(6,1)=2.*(QMA(4)-Q1S/QMA(4))
PA(6,2)=2.*(QMA(3)-QMA(1)*QMA(2)/QMA(4))
PA(6,3)=2.*(QMA(2)-QMA(1)*QMA(3)/QMA(4))
PA(7,1)=2.*(QMA(3)-QMA(1)*QMA(2)/QMA(4))
PA(7,2)=2.*(QMA(4)-Q2S/QMA(4))
PA(7,3)=2.*(QMA(1)-QMA(2)*QMA(3)/QMA(4))
PA(8,1)=-2.*(QMA(4)-Q1S/QMA(4))
PA(8,2)=2.*(QMA(3)+QMA(1)*QMA(2)/QMA(4))
PA(8,3)=2.*(QMA(2)+QMA(1)*QMA(3)/QMA(4))
PA(9,1)=-4.*QMA(1)
PA(9,2)=-4.*QMA(2)
PA(9,3)=0.
R1=QMA(1)/QMA(4)
R2=QMA(2)/QMA(4)
R3=QMA(3)/QMA(4)
R4=QMA(1)*QMA(2)*QMA(3)/(Q4S*QMA(4))
DPA(1,1)=0.
DPA(1,2)=0.
DPA(1,3)=0.
DPA(1,4)=0.
DPA(1,5)=-4.
DPA(1,6)=0.
DPA(1,7)=0.
DPA(1,8)=0.
DPA(1,9)=-4.
DPA(2,1)=-2.*R3*(1.+Q1S/Q4S)
DPA(2,2)=2.*(1.-R4)
DPA(2,3)=-2.*R1*(1.+Q3S/Q4S)
DPA(2,4)=DPA(2,2)
DPA(2,5)=-2.*R3*(1.+Q2S/Q4S)
DPA(2,6)=-2.*R2*(1.+Q3S/Q4S)
DPA(2,7)=DPA(2,3)
DPA(2,8)=DPA(2,6)
DPA(2,9)=-2.*R3*(3.+Q3S/Q4S)
DPA(3,1)=2.*R2*(1.+Q1S/Q4S)
DPA(3,2)=2.*R1*(1.+Q2S/Q4S)
DPA(3,3)=2.*(1.+R4)
DPA(3,4)=DPA(3,2)
DPA(3,5)=2.*R2*(3.+Q2S/Q4S)
DPA(3,6)=2.*R3*(1.+Q2S/Q4S)
DPA(3,7)=DPA(3,3)
DPA(3,8)=DPA(3,6)
DPA(3,9)=2.*R2*(1.+Q3S/Q4S)
DPA(4,1)=2.*R3*(1.+Q1S/Q4S)
DPA(4,2)=2.*(1.+R4)
DPA(4,3)=2.*R1*(1.+Q3S/Q4S)
DPA(4,4)=DPA(4,2)
DPA(4,5)=2.*R3*(1.+Q2S/Q4S)
```

```
      DPA(4,6)=2.*R2*(1.+Q3S/Q4S)
      DPA(4,7)=DPA(4,3)
      DPA(4,8)=DPA(4,6)
      DPA(4,9)=2.*R3*(3.+Q3S/Q4S)
      DPA(5,1)=-4.
      DPA(5,2)=0.
      DPA(5,3)=0.
      DPA(5,4)=0.
      DPA(5,5)=0.
      DPA(5,6)=0.
      DPA(5,7)=0.
      DPA(5,8)=0.
      DPA(5,9)=-4.
      DPA(6,1)=-2.*R1*(3.+Q1S/Q4S)
      DPA(6,2)=-2.*R2*(1.+Q1S/Q4S)
      DPA(6,3)=-2.*R3*(1.+Q1S/Q4S)
      DPA(6,4)=DPA(6,2)
      DPA(6,5)=-2.*R1*(1.+Q2S/Q4S)
      DPA(6,6)=2.*(1.-R4)
      DPA(6,7)=DPA(6,3)
      DPA(6,8)=DPA(6,6)
      DPA(6,9)=-2.*R1*(1.+Q3S/Q4S)
      DPA(7,1)=-2.*R2*(1.+Q1S/Q4S)
      DPA(7,2)=-2.*R1*(1.+Q2S/Q4S)
      DPA(7,3)=2.*(1.-R4)
      DPA(7,4)=DPA(7,2)
      DPA(7,5)=-2.*R2*(3.+Q2S/Q4S)
      DPA(7,6)=-2.*R3*(1.+Q2S/Q4S)
      DPA(7,7)=DPA(7,3)
      DPA(7,8)=DPA(7,6)
      DPA(7,9)=-2.*R2*(1.+Q3S/Q4S)
      DPA(8,1)=2.*R1*(3.+Q1S/Q4S)
      DPA(8,2)=2.*R2*(1.+Q1S/Q4S)
      DPA(8,3)=2.*R3*(1.+Q1S/Q4S)
      DPA(8,4)=DPA(8,2)
      DPA(8,5)=2.*R1*(1.+Q2S/Q4S)
      DPA(8,6)=2.*(1.+R4)
      DPA(8,7)=DPA(8,3)
      DPA(8,8)=DPA(8,6)
      DPA(8,9)=2.*R1*(1.+Q3S/Q4S)
      DPA(9,1)=-4.
      DPA(9,2)=0.
      DPA(9,3)=0.
      DPA(9,4)=0.
      DPA(9,5)=-4.
      DPA(9,6)=0.
      DPA(9,7)=0.
      DPA(9,8)=0.
      DPA(9,9)=0.
      DO 49 I=1,6
      DO 49 J=1,6
      AT(I,J)=0.0
   49 H2P(I,J)=0.0
```

```
      DO 999 IZT=1,3
      IF(ND(IZT).NE.1) GO TO 999
C
      DO 50 I=1,3
   50 YRA(I)=YRO(IZT,I)
C
C---- YRP = STAR DIRECTION IN THE P-SYSTEM BASED ON THE GYRO DATA
C           YRP = TM * YRA
C
      CALL GMPRD(TM,YRA,YRP,3,3,1)
C
      GOTO (300,400,500),IZT
C
C---- TRANSFORMATION MATRIX FROM P TO S1 SYSTEM
C
  300 DO 350 I=1,3
      DO 350 J=1,3
  350 T(I,J)=0.
      DO 370 I=1,3
  370 T(I,I)=1.
C
      IZT1=1
      GOTO 600
C
C---- TRANSFORMATION MATRIX FROM S1 TO S2 SYSTEM
C
  400 DO 450 I=1,3
      DO 450 J=1,3
  450 T(I,J)=TS2S1(I,J)
C
      IZT1=3
      GOTO 600
C
C---- TRANSFORMATION MATRIX FROM S1 TO S3 SYSTEM
C
  500 DO 550 I=1,3
      DO 550 J=1,3
  550 T(I,J)=TS3S1(I,J)
C
      IZT1=5
  600 CONTINUE
C
      CALL GMPRD(T,YRP,YRS,3,3,1)
C
      CX=1./YRS(1)
C     CX2=CX*CX
C
C---- YRN = NOMINAL MEASUREMENT VECTOR; THE NOMINAL P-SYSTEM IS
C                          DETERMINED FROM THE GYRO MEASUREMENTS--------

      DO 700 I=1,2
  700 YRN(I)=YRS(I+1)*CX
C
```

```
C************* MEASUREMENT DEVIATION VECTOR ****************************
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  NEXT 2 CARDS FOR YMSA MOVED UP TO HERE BY MARK WEST
C
      YMSA(IZT,1)=YMS(IZT,1)
      YMSA(IZT,2)=YMS(IZT,2)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     DO 800 I=1,2
C 800 YD(I) = YMSA(IZT,I) - YRN(I)
C 800 YD(I) = YMSA(IZT,I)
  806 CONTINUE
C
C**** COMPUTATION OF THE OBSERVATION MATRIX (REF. SECT.4.6.4.4) ********
C
      DO 10 I=1,3
      DO 10 J=1,3
      K=J+3
      L=J+6
      B(I,J)=T(I,1)*YRA(J)
      B(I,K)=T(I,2)*YRA(J)
   10 B(I,L)=T(I,3)*YRA(J)
      CALL MATVEC(3,9,B,A,SH)
      DO 30 I=1,3
      DO 20 J=1,9
   20 DUM(J)=PA(J,I)
      CALL MATVEC(3,9,B,DUM,FUM)
      DH1(I)=FUM(1)
      DH2(I)=FUM(2)
   30 DH3(I)=FUM(3)
      DO 45 I=1,9
      DO 40 J=1,9
   40 DUM(J)=DPA(J,I)
      CALL MATVEC(3,9,B,DUM,FUM)
      DDH1(I)=FUM(1)
      DDH2(I)=FUM(2)
   45 DDH3(I)=FUM(3)
      BH(1)=SH(2)/SH(1)
      BH(2)=SH(3)/SH(1)
      DO 800 I=1,2
  800 YDA(I,IZT) = YMSA(IZT,I) - BH(I)
C  800 YDA(I,IZT) = YMSA(IZT,I)
      H12=SH(1)*SH(2)
      H13=SH(1)*SH(3)
      H1S=SH(1)*SH(1)
      H1C=H1S*SH(1)
      DO 60 I=1,3
      PH1(I)=(SH(1)*DH2(I)-SH(2)*DH1(I))/H1S
   60 PH2(I)=(SH(1)*DH3(I)-SH(3)*DH1(I))/H1S
      HP(IZT1,1)=PH1(1)
```

```
      HP(IZT1,2)=PH1(2)
      HP(IZT1,3)=PH1(3)
      HP(IZT1+1,1)=PH2(1)
      HP(IZT1+1,2)=PH2(2)
      HP(IZT1+1,3)=PH2(3)
      DO 70 I=1,3
      J=I+3
      K=I+6
      DPH1(I)=(H1S*DDH2(I)-H12*DDH1(I)+2.*SH(2)*DH1(1)*DH1(I)-
     1SH(1)*(DH2(1)*DH1(I)+DH1(1)*DH2(I)))/H1C
      DPH2(I)=(H1S*DDH3(I)-H13*DDH1(I)+2.*SH(3)*DH1(1)*DH1(I)-
     1SH(1)*(DH3(1)*DH1(I)+DH1(1)*DH3(I)))/H1C
      DPH1(J)=(H1S*DDH2(J)-H12*DDH1(J)+2.*SH(2)*DH1(2)*DH1(I)-
     1SH(1)*(DH2(2)*DH1(I)+DH1(2)*DH2(I)))/H1C
      DPH2(J)=(H1S*DDH3(J)-H13*DDH1(J)+2.*SH(3)*DH1(2)*DH1(I)-
     1SH(1)*(DH3(2)*DH1(I)+DH1(2)*DH3(I)))/H1C
      DPH1(K)=(H1S*DDH2(K)-H12*DDH1(K)+2.*SH(2)*DH1(3)*DH1(I)-
     1SH(1)*(DH2(3)*DH1(I)+DH1(3)*DH2(I)))/H1C
      DPH2(K)=(H1S*DDH3(K)-H13*DDH1(K)+2.*SH(3)*DH1(3)*DH1(I)-
     1SH(1)*(DH3(3)*DH1(I)+DH1(3)*DH3(I)))/H1C
   70 CONTINUE
      H2P(1,1)=DPH1(1)
      H2P(1,2)=DPH1(2)
      H2P(1,3)=DPH1(3)
      H2P(2,1)=DPH1(4)
      H2P(2,2)=DPH1(5)
      H2P(2,3)=DPH1(6)
      H2P(3,1)=DPH1(7)
      H2P(3,2)=DPH1(8)
      H2P(3,3)=DPH1(9)
      CALL GMPRD(H2P,PM,HPP,6,6,6)
      H2PT=0.0
      DO 25 I=1,6
   25 H2PT=H2PT+HPP(I,I)*0.5
      HT(IZT1)=H2PT
      DO 26 I=1,6
      DO 26 J=1,6
   26 H2PO(I,J)=H2P(I,J)
C
C     BEGIN CALCULATION FOR Z MEASUREMENT
C
      H2P(1,1)=DPH2(1)
      H2P(1,2)=DPH2(2)
      H2P(1,3)=DPH2(3)
      H2P(2,1)=DPH2(4)
      H2P(2,2)=DPH2(5)
      H2P(2,3)=DPH2(6)
      H2P(3,1)=DPH2(7)
      H2P(3,2)=DPH2(8)
      H2P(3,3)=DPH2(9)
      CALL GMPRD(H2P,PM,HPP,6,6,6)
      H2PT=0.0
      DO 35 I=1,6
```

```
 35   H2PT=H2PT+HPP(I,I)*0.5
      HT(IZT1+1)=H2PT
C
C   CALCULATION OF A TILDE MATRIX FOR GAIN CALCULATIONS
C
      CALL GMPRD(H2PO,PM,HPOP,6,6,6)
      CALL GMPRD(H2P,PM,H2PP,6,6,6)
      CALL GMPRD(HPOP,HPOP,HPHP,6,6,6)
      CALL GMPRD(HPOP,H2PP,HPHP1,6,6,6)
      CALL GMPRD(H2PP,H2PP,HPHP2,6,6,6)
      HPT1=0.0
      HPT2=0.0
      HPT3=0.0
      DO 55 I=1,6
      HPT1=HPT1+HPHP(I,I)*0.5
      HPT2=HPT2+HPHP1(I,I)*0.5
 55   HPT3=HPT3+HPHP2(I,I)*0.5
      AT(IZT1,IZT1)=HPT1
      AT(IZT1,IZT1+1)=HPT2
      AT(IZT1+1,IZT1)=HPT2
      AT(IZT1+1,IZT1+1)=HPT3
 999  CONTINUE
 777  RETURN
      END
```

IPS SOKF Input Data

IPSE.DAT

```
.5E-5 .5E-5 .5E-5
.5E-5 .5E-5 .5E-5 .5E-5 .5E-5 .5E-5
2
0. 1 0. 0. 0.
100. 2 0. 0. 0.
1.45E-5 1.45E-5 1.45E-5
.007 .005 .005 1.E-7 1.E-7 1.E-7 4.85E-6 4.85E-6 4.85E-6 4.85E-6 4.85E-6 4.85E-6
1.212E-6 1.212E-6 1.212E-6 1.E-9 1.E-9 1.E-9
```

ASTP.DAT

```
 &NFILT
 LATTUP=T, P_I=-1500., 900., 900., NTR=-2, JSUBOP=3
 MODE(1)=1,     BJAS(1,1)=0.0,     SIGMAN(1,1)=.75,     YS(1,1)=0.,
                BJAS(1,2)=0.0,     SIGMAN(1,2)=.75,     YS(1,2)=0.,
 MODE(2)=1,     BJAS(2,1)=0.0,     SIGMAN(2,1)=1.0,     YS(2,1)=0.,
                BJAS(2,2)=0.0,     SIGMAN(2,2)=0.0,     YS(2,2)=0.,
 MODE(3)=0,     BJAS(3,1)=0.0,     SIGMAN(3,1)=1.0,     YS(3,1)=0.,
                BJAS(3,2)=0.0,     SIGMAN(3,2)=1.0,     YS(3,2)=0.,
 MODE(4)=0,     BJAS(4,1)=0.0,     SIGMAN(4,1)=.75,     YS(4,1)=0.,
                BJAS(4,2)=0.0,     SIGMAN(4,2)=.75,     YS(4,2)=0.,
 MODE(5)=0,     BJAS(5,1)=0.0,     SIGMAN(5,1)=1.,     YS(5,1)=0.,
                BJAS(5,2)=0.0,     SIGMAN(5,2)=1.,     YS(5,2)=0.,
 MODE(6)=0,     BJAS(6,1)=0.0,     SIGMAN(6,1)=1.,     YS(6,1)=0.,
                BJAS(6,2)=0.0,     SIGMAN(6,2)=1.,     YS(6,2)=0.,
 ITEST=0,0,0,0,0,0,0,0,0,0, LTHDIS=T, ALPHO=12.0, BETA=-45,
 FAC=1.,  JREAD=16,  JSST=2,
 DEPHIX=0.0, 0.0,   DEPHIY=0.0,0.0,   DEPHIZ=0.0,0.0,
 DRTHX=  2*0.,   DRTHY= 2*0.,   DRTHZ= 0., 0.,
 TSETTL=60.0,  EFLCH1=2000.0,  EFLCH2=90.0,
 LFHST=1, ALP11=7.36, ALP12=127.158, ALP2=0.21875, FDTTR=2.38, WEFF=25.,
 FOVR=72.5,
 FOVDY=0.0, FOVDZ=0.0, DSUN=1920.0, DNEGST=200.0, DSCAN=36.75, RNTNS=0.3,
 &END
```

REFERENCES

1. Kosters, Bernd: "The First Mission of the Instrument Pointing System." AAS Guidance and Control Conference, AAS 86-053, 1986.

2. Vallely, D.P.: "IPS Control and Performance for Spacelab-2 on Mission 51F." NASA Internal Memorandum No. ED12-85-112, George C. Marshall Space Flight Center, Huntsville, Alabama, October 31, 1985.

3. Leibold, G.: "IPS Attitude Measurement Software." VII IFAC Symposium on Automatic Control in Space, Oxford, 1979, pp. 149–154.

4. Gelb, A. (ed): "Applied Optimal Estimation." MIT Press, Cambridge, Massachusetts, 1974.

5. Vathsal, S.: "Spacecraft Attitude Determination Using a Second-Order Nonlinear Filter." Journal of Guidance, Control, and Dynamics, vol. 10, No. 6, November–December 1987, pp. 559–566.

6. _____: "Instrument Pointing System Spacelab Systems Training Manual." Missions Operations Directorate, Training Division, Systems Training Branch, Lyndon B. Johnson Space Center, Houston, Texas, 1986.

7. _____: "Brushless Torque Motor Specification." NASA Technical Document IPS-DS-SP-041, Marshall Space Flight Center, 1985.

8. _____: "Data Electronics Assembly." NASA Technical Document IPS-DS-SP-0100, Marshall Space Flight Center, 1985.

9. _____: "Gyro Package Specification." NASA Technical Document IPS-DS-SP-0032, Marshall Space Flight Center, 1985.

10. _____: "Fixed Head Star/Sun Tracker Specification." NASA Technical Document, IPS-DS-SP-0031, Marshall Space Flight Center, 1985.

11. _____: "IPS Control Software Requirements." NASA Technical Document IPS-DS-TN-0464, Marshall Space Flight Center, 1985.

12. Kailath, T.: " A View of Three Decades of Linear Filtering Theory." IEEE Transactions on Information Theory, vol. IT-20, No. 2, March 1974, pp. 146–181.

13. Sorenson, H.W.: "Least-Squares Estimation From Gauss to Kalman." IEEE Spectrum, vol. 7, July 1970, pp. 63–68.

14. Lefferts, E.J., Markley, F.L., and Shuster, M.D.: "Kalman Filtering for Spacecraft Attitude Estimation." Journal of Guidance, Control, and Dynamics, vol. 5, September–October 1982, pp. 417–429.

15. Jazwinski, A.: "Stochastic Processes and Filtering Theory." Academic Press, New York, 1970.

16. Brown, R.G.: "Introduction to Random Signal Analysis and Kalman Filtering." John Wiley and Sons, New York, 1983.

17. Legendre, A.M.: "Methode des moindres quarres, pour trouver le milieu le plus probable entre les resultats de differentes observations." Mem. Inst. France, 1805, pp. 149–154.

18. Gauss, K.F.: "Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientum." Hamburg, 1809 (Translated: Dover, New York, 1963).

19. Fisher, R.A.: "On an Absolute Criterion for Fitting Frequency Curves." Messenger of Math, vol. 41, 1912, p. 155.

20. Kolmogorov, A.N.: "Interpolation and Extrapolation von Stationaren Zufalligen Folgen." Bull. Acad. Sci., USSR, Ser. Math 5, 1941, pp. 3–14.

21. Wiener, N.: "The Extrapolation, Interpolation, and Smoothing of Stationary Time Series." John Wiley and Sons, New York, 1949.

22. Kalman, R.: "A New Approach to Linear Filtering and Prediction Problems." Paper 59-IRD-11 presented at ASME Instruments and Regulators Conference, March 29–April 2, 1959.

23. Sorenson, H.W.: "Kalman Filtering Techniques." Advances in Control Systems, vol. 3, edited by C.T. Leondes, Academic Press, New York, 1966.

24. Bryson, A.E., Jr., and Ho, Y.: "Applied Optimal Control." Hemisphere Publishing Corp., New York, 1975.

25. Bar-Itzhack, I.Y., and Reiner, J.: "Recursive Attitude Determination From Vector Observations: Direction Cosine Matrix Identification." Journal of Guidance, Control, and Dynamics, vol. 7, No. 1, 1984, pp. 51–56.

26. Bar-Itzhack, I.Y., and Idan, M.: "Recursive Determination From Vector Observations: Euler Angle Estimation." Journal of Guidance, Control, and Dynamics, vol. 10, No. 2, 1987, pp. 152–157.

27. Bar-Itzhack, I.Y., and Oshman, Y.: "Attitude Determination From Vector Observations: Quaternion Estimation." IEEE Transactions on Aerospace and Electronic Systems, vol. AES-21, No. 1, 1985, pp. 128–135.

28. Wertz, J.R. (ed).: "Spacecraft Attitude Determination and Control." D. Reidel Publishing Company, Dordrecht, Holland, vol. 73, 1978.

29. Ickes, B.P.: "A New Method for Performing Digital Control System Attitude Computations Using Quaternions." AIAA Journal, vol. 8, January 1970, pp. 13–17.

30. _____: "A Linearized Kalman Filter Method for the IPS Attitude Determination, and Covariance Analysis With Suboptimal Gains Precalculation of the Filter Gains." NASA Technical Document IPS-DS-TN-0354, Marshall Space Flight Center, 1985.

31. McElroy, T.: "Euler Symmetric Parameters Revisited." NASA Technical Document H2V-415-HEAO-S-73-463, Marshall Space Flight Center, 1973.

32. _____: "Coordinate Systems for IPS as Used in Flight and Check-Out." NASA Technical Document IPS-DS-TN-0012, Marshall Space Flight Center, 1984.

33. Singh, R.P., Vander Voort, R.J., and Likins, P.W.: "Dynamics of Flexible Bodies in Tree Topology—A Computer Oriented Approach." Proceedings of the AIAA Dynamics Specialist Conference, Palm Springs, California, May 1984, pp. 327–337

34. _____: "Users Manual for TREETOPS, A Control System Simulation for Structures With a Tree Topology." NASA Contract NAS-36287, Marshall Space Flight Center, April 1990.

APPROVAL

A REAL-TIME RECURSIVE FILTER FOR THE ATTITUDE DETERMINATION OF THE SPACELAB INSTRUMENT POINTING SUBSYSTEM

By M.E. West

The information in this report has been reviewed for technical content. Review of any information concerning Department of Defense or nuclear energy activities or programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

JAMES C. BLAIR
Director, Structures and Dynamics Laboratory

☆ U.S. GOVERNMENT PRINTING OFFICE 1992—631-060/60071