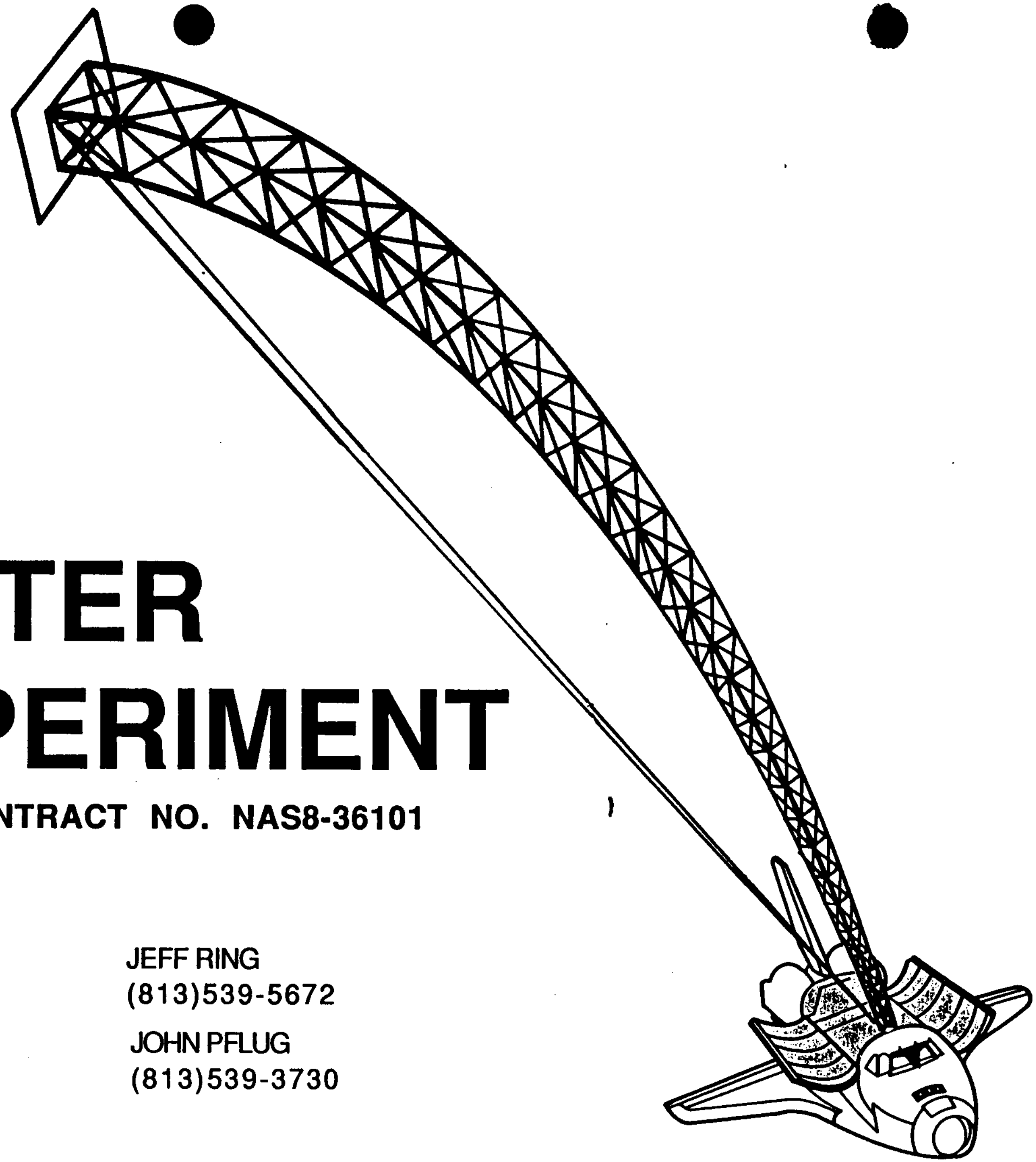


PINHOLE OCCULTER EXPERIMENT

CONTRACT NO. NAS8-36101

JEFF RING
(813)539-5672

JOHN PFLUG
(813)539-3730

Honeywell

SPACE & STRATEGIC
AVIONICS DIVISION
CLEARWATER, FLORIDA

POF BRIEFING OUTLINE

- Introduction
- Program Accomplishments
- Summary of Results(Orbiter/POF Baseline)
- Conclusions
- Recommendations
- Orbiter/POF
    - TREETOPS Modeling
    - Compensator Design
    - Summary of Results
- Space Station/POF
    - TREETOPS Modeling
    - Summary of Results
- Appendix
    - TREETOPS Time Histories
    - TREETOPS Files
    - Crew Motion Supplemental Data
    - HONEY-X Design and Analysis Command Files

INTRODUCTION

- The results in this document are the output of a feasibility study undertaken between July 1986 and February 1987.

- The conclusions reached are influenced by the fidelity of the math models used. In particular, the stochastic crew motion models are conservative representations of the "real world". Also, the Orbiter RCS deadband vs. disturbance frequency study is dependant on the orientation of the Shuttle wrt the orbit plane.

1986 PROGRAM ACCOMPLISHMENTS

- Improved IPS Gimbal Model - replaced 3 DOF hinge with 3 single DOF hinges enabling modeling of gimbal masses.

- Improved Crew Motion Disturbance Model - Replaced deterministic simulated Crew wall pushoff and stop with a stochastic representation of a variety of both low level and high level motion activities.

- Utilized the existing intricate Honeywell Shuttle on-orbit simulation to study the effects of Orbiter attitude deadband size on POF performance.

- Furthered our understanding of the maximum performance expected from the current actuator/sensor set through the study of the controllability and observability of the system modes.

- Utilized the powerful nonlinear time domain program(TREETOPS) developed by Honeywell. Through the use of this tool we were able to quickly obtain the system dynamics describing the complex multibody flexible structures studied in this effort.

- Utilized the fast and efficient computer aided design tool HONEY-X developed by Honeywell to design and evaluate our multivariable compensator for stability, robustness and performance.

- Applied a state of the art compensator design methodology(<u>L</u>inear <u>Q</u>uadratic <u>G</u>aussian/<u>L</u>oop <u>T</u>ransfer <u>R</u>ecovery) first introduced by Honeywell.

- Examined for the first time the tolerance required on knowledge of the POF boom flexible mode frequencies to insure stability. This measure of system robustness was obtained using the Structured Uncertainty Analysis(u test) developed at Honeywell.

SUMMARY OF RESULTS

O PERFORMANCE

	Elevation	Cross Elevation	Roll
Band Width	0.658 rad/sec	0.641 rad/sec	0.389 rad/sec
Dominant Damping Ratio	0.687	0.7045	0.463
T_S : Step Response Settle Time	11 sec	12 sec	22 sec
$\delta_{\ell-rms}$	1.0 arc-sec-rms	1.85 arc-sec-rms	2.6 arc-sec-rms
δ_{ℓ}	2.5 arc-sec	5.0 arc-sec	5.0 arc-sec
δ_{h-rms}	1.4 arc-sec-rms	2.4 arc-sec-rms	2.7 arc-sec-rms
δ_{h}	3.5 arc-sec	5.5 arc-sec	5.5 arc-sec

δ : Line Of Sight Error h: High Level Crew Motion ℓ: Low Level Crew Motion

Results For Step Response Are Consistent With Those Presented For The Elevation And Cross Elevation Axis In The 1985 Study.

Honeywell

SUMMARY OF RESULTS

J.F.P 1/26/87

O **STABILITY**

	Elevation	Cross Elevation	Roll
Gain Margin	3 DB	3 DB	35 DB
Phase Margin	40 Degrees	40 Degrees	45 Degrees

O **PRELIMINARY ROBUSTNESS**

oo Results From Structured Uncertainty Analysis For 10% Uncertainty In The Three Modal Frequencies

$\mu = 2.2$

$\therefore$ *will tolerate up to* 4.54 % *uncertainty on modal frequency*

Conclusions

1. With The Current Set Of Sensors And Actuators,We Fall Just Short Of The Current Goals For The Control Of The POF Mast Flexure.

   O Bandwidth/Performance Implications

   oo The Existing POF System/Controller Exhibits Performance Outside The 1.0 arc-sec Pointing Requirement. A less conservative model of crew motion however, might bring pointing performance back within spec.

   O Stability/Robustness Implications

   oo Stability And Robustness Margins Of The Existing POF System Need Improvement.

   - Model Uncertainty/Unmodeled Effects Need To Be Addressed.

2. The Disturbances Due To RCS Jet Firings For Attitude Corrections May Be Significantly Decreased By Increasing The RCS Attitude Deadbands.

   - Effects Of Increased Jet-On-Time On Settling Time And Sensor Field Of View Need To Be Examined.

THERE ARE SEVERAL OPTIONS AVAILABLE TO ADDRESS THE ABOVE PROBLEM

Recommendations

1. Define Model Uncertainty / Tolerance Data Base

2. Define And Examine Options For Improving System Performance

   - Limiting Crew Motion

   - Adding Passive Structural Damping Devices

   - Adding Additional Sensors/Actuators To Control Mast Flexure

2a. Redesign "Optimal" controller To Optimize Performance/Stability Robustness Within Model Uncertainty Spec.

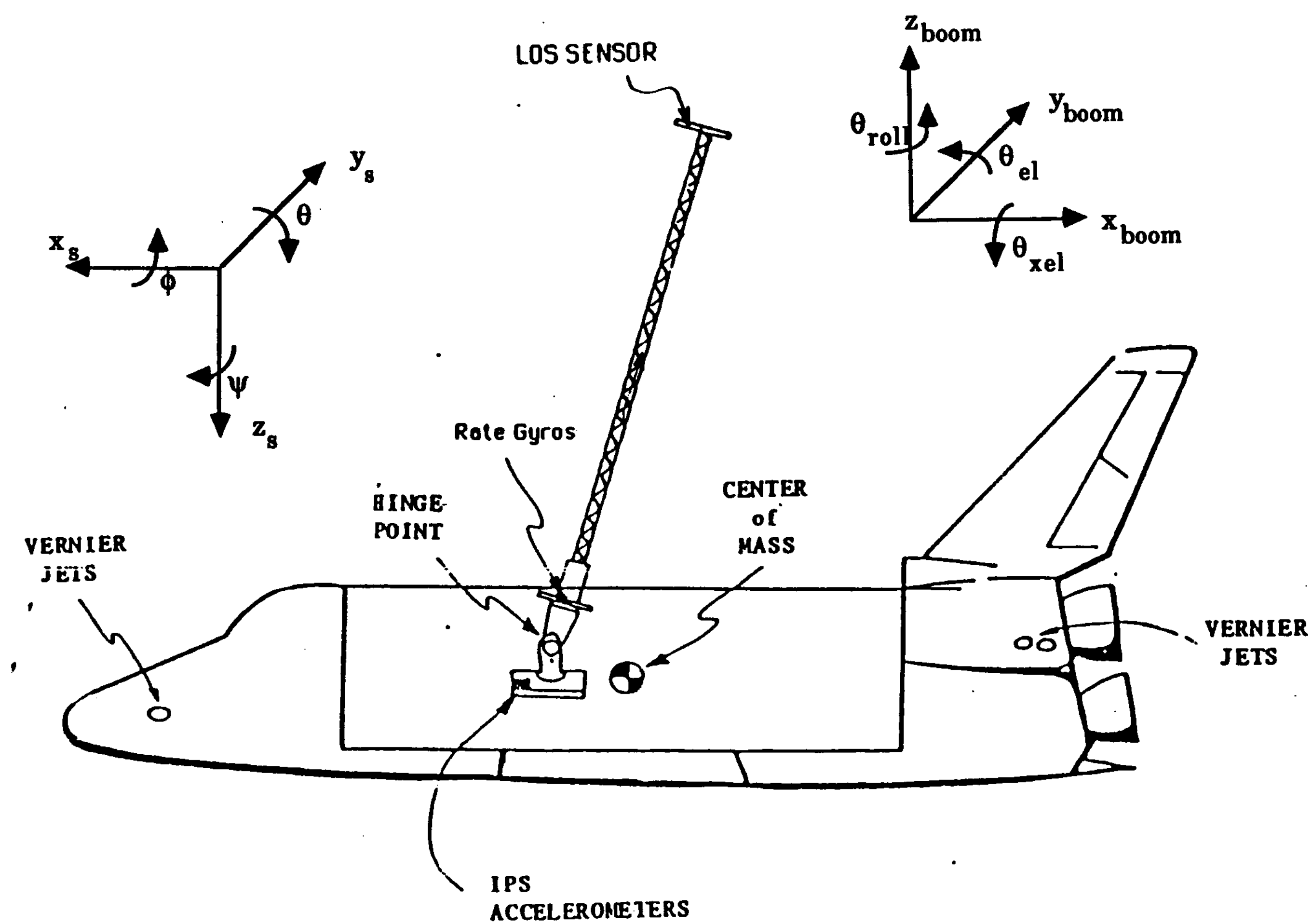
3. If Adequate Performance/Stability Is Available, Design "Classical" Controller To Simplify Implementation And Improve Useability Of Existing IPS Controller Hardware And Software.

4. Examine Sensor Field-Of-View And Settle Time Issues Associated With Increased RCS Deadband.

- Tasks 1,2,2a required to show feasibility of concept

* Tasks 3,4 would be follow on tasks to optimize the performance and implementability.

SPACE SHUTTLE ORBITER WITH POF



Control Objective

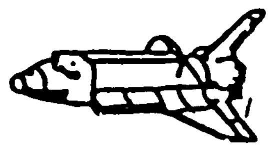
O Hold Line Of Sight Error To Less Than 1.0 arc-sec

TREETOPS MODELING

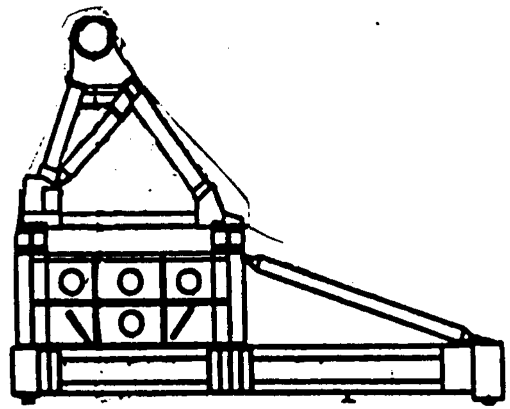
- Body Input Data
- Hinge Input Data
- Sensor Input Data
- Actuator Input Data
- Linear Models
- Controller Input Data
- Function Generation
- Interconnect Definition

ORBITER/POF STRUCTURE TOPOLOGY

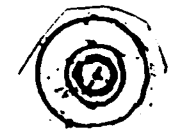
SPACE SHUTTLE
(BODY 1)



IPS STRUCTURE UP
TO EDU (BODY 2)

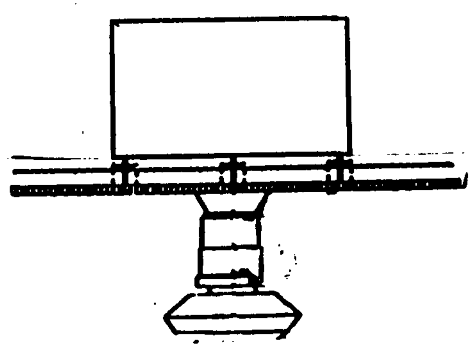


IPS FROM OUTBOARD PART OF
EDU TO INBOARD PART OF XDU
(BODY 3)

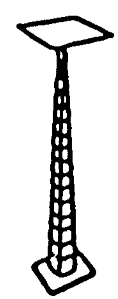


IPS FROM OUTBOARD PART OF
XDU TO INBOARD PART OF RDU
(BODY 4)

IPS FROM OUTBOARD PART OF
RDU TO BOOM INTERFACE
(BODY 5)



P/QF BOOM + MASK
(BODY 6)

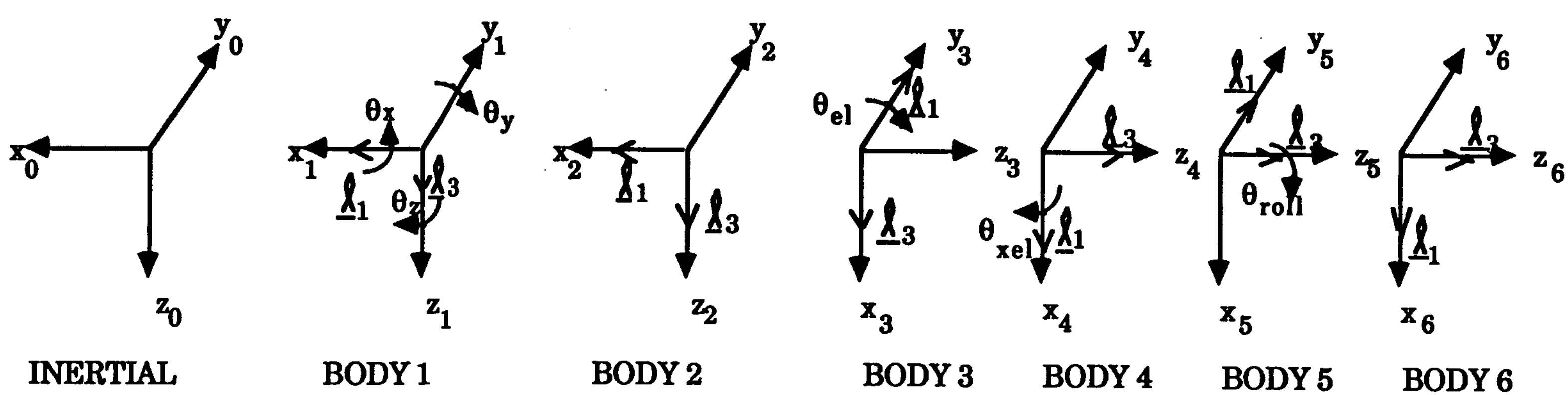


ORBITER/POF TREETOPS BODY INPUT DATA

		1	2	3	4	5	6
BODY#:		1	2	3	4	5	6
TYPE:		Rigid	Rigid	Rigid	Rigid	Rigid	Flex (6 modes)
MASS(kg):		$.10344 \times 10^6$	4358.	84.4	105.2	1875.	69.87
MOMENTS OF INERTIA($kg\text{-}m^2$):	I_{xx}	1.3951×10^6	7307.	1.3554	9.4878	12691.	65391.
	I_{yy}	10.69×10^6	19805.	5.4216	16.265	12354.	65416.
	I_{zz}	11.064×10^6	23648.	5.4216	8.1324	1903.	81.
PRODUCTS OF INERTIA($kg\text{-}m^2$):	I_{xy}	-3177.1	645.2	0.	0.	0.	0.
	I_{xz}	$-.46457 \times 10^6$	-669.6	-8.1324	-28.463	0.	0.
	I_{yz}	262.9	-146.4	0.	-1.3554	0.	0.
ATTACH POINT(m):		(0,0,0)	(0,0,0)	(-.147,-.00686,-.003)	(.191,.00102,-.311)	(0,0,0)	(0,0,0)
MASS CENTER(m):		(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,.5761)	(0,0,29.92)
HINGE POINT(m):		(4.209,-.1135,.201)	(1.125,.1046,-1.075)	(.336,-.00686,-.003)	(-.292,.00102,.3)	(0,0,4.3198)	
SENSOR/ACTUATOR MOUNTING POINTS(m):							
F5R VRCS JET		(19.34,1.52,-.61)					
F5L VRCS JET		(19.34,-1.52,-.61)					
R5R VRCS JET		(-12.17,3.81,2.16)					
L5L VRCS JET		(-12.17,-3.81,2.16)					
R5D VRCS JET		(-12.17,3.0,2.07)					
L5D VRCS JET		(-12.17,-3.0,2.07)					
CREW DISTURBANCE		(13.966,0.,-1.858)					
IPS ACCELEROMETERS			(.858,.5846,-.27)				
IPS RATE GYROS						(0,0,1)	
POF LOS SENSOR							(0,0,32.)

ORBITER/POF TREETOPS HINGE INPUT DATA

HINGE ID#	1	2	3	4	5	6
INBOARD BODY ID#	0	1	2	3	4	5
OUTBOARD BODY ID#	1	2	3	4	5	6
ROTATIONAL DOF:	3	0	1	1	1	0
ROTATION AXIS:						
λ_1 WRT INBOARD BODY	(1,0,0)	(1,0,0)	(0,1,0)	(1,0,0)	(0,0,1)	(1,0,0)
$\underline{\lambda}_1$ WRT OUTBOARD BODY	(1,0,0)	(1,0,0)	(0,1,0)	(1,0,0)	(0,0,1)	(1,0,0)
λ_3 WRT INBOARD BODY	(0,0,1)	(0,0,1)	(0,0,1)	(0,0,1)	(0,1,0)	(0,0,1)
$\underline{\lambda}_3$ WRT OUTBOARD BODY	(0,0,1)	(0,0,1)	(1,0,0)	(0,0,1)	(0,1,0)	(0,0,1)
INITIAL ROTATION ANGLE(deg)	(0,0,0)		-90.	0.	0.	0.
TRANSLATIONAL DOF:	3	0	0	0	0	0



ORBITER/POF TREETOPS SENSOR INPUT DATA

SENSOR ID#	SENSOR	MOUNTING POINT BODY ID#	MOUNTING POINT NODE ID#	INPUT AXIS
1	IPS RATE GYRO(roll axis)	5	3	0,0,1
2	IPS RATE GYRO(elevation axis)	5	3	0,1,0
3	IPS RATE GYRO(cross elevation axis)	5	3	1,0,0
4	IPS ACCELEROMETER(x)	2	3	1,0,0
5	IPS ACCELEROMETER(y)	2	3	0,1,0
6	IPS ACCELEROMETER(z)	2	3	0,0,1
99	POF LINE OF SIGHT	6	2	0,0,-1

ORBITER/POF TREETOPS ACTUATOR INPUT DATA

ACTUATOR ID#	ACTUATOR	HINGE ID#	MOUNTING POINT HINGE AXIS ID#	MOUNTING POINT BODY ID#	MOUNTING POINT NODE ID#	OUTPUT AXIS
1	F5R VRCS JET			1	3	(-.03265,-.69625,-.71706)
2	F5L VRCS JET			1	4	(-.03265,.69625,-.71706)
3	R5R VRCS JET			1	5	(0.,-.99967,-.02582)
4	LFL VRCS JET			1	6	(0.,.99967,-.02582)
5	R5D VRCS JET			1	7	(.02656,-.28662,-.95768)
6	L5D VRSC JET			1	8	(.02656,.28662,-.95768)
7	IPS TORQUE MOTOR(roll)	5	1			
8	IPS TORQUE MOTOR(elevation)	3	1			
9	IPS TORQUE MOTOR(cross elevation)	4	1			
10	CREW MOTION FORCE(x)			1	9	(1,0,0)
11	CREW MOTION FORCE(y)			1	9	(0,1,0)
12	CREW MOTION FORCE(z)			1	9	(0,0,1)

PINHOLE OCCULTER STATE VECTOR DEFINITION

(FULL STATE MODEL)

STATE VECTOR

X

X_1	$\dot{\theta}_1^1$	Orbiter rotation wrt Inertial frame
X_2	$\dot{\theta}_2^1$	
X_3	$\dot{\theta}_3^1$	
X_4	$\dot{y}_1^1$	Orbiter translation wrt Inertial frame
X_5	$\dot{y}_2^1$	
X_6	$\dot{y}_3^1$	
X_7	$\dot{\theta}_1^3$	Elevation gimbal rate
X_8	$\dot{\theta}_1^4$	X Elevation gimbal rate
X_9	$\dot{\theta}_1^5$	Roll gimbal rate
X_{10}	$\dot{\eta}_1^6$	Bending modes of P/OF Boom (Body 6)
X_{11}	$\dot{\eta}_2^6$	
X_{12}	$\dot{\eta}_3^6$	
X_{13}	$\dot{\eta}_4^6$	
X_{14}	$\dot{\eta}_5^6$	
X_{15}	$\dot{\eta}_6^6$	
X_{16}	$\dot{\eta}_7^6$	
X_{17}	$\dot{\eta}_8^6$	
X_{18}	θ_1^1	Generalized Coordinates corresponding to the above Generalized Velocities
X_{19}	θ_2^1	
X_{20}	θ_3^1	
X_{21}	y_1^1	
X_{22}	y_2^1	
X_{23}	y_3^1	
X_{24}	θ_1^3	
X_{25}	θ_1^4	
X_{26}	θ_1^5	
X_{27}	η_1^6	
X_{28}	η_2^6	
X_{29}	η_3^6	
X_{30}	η_4^6	
X_{31}	η_5^6	
X_{32}	η_6^6	
X_{33}	η_7^6	
X_{34}	η_8^6	

INPUT(Actuator Commands)

U

U_1	J_1	Orbiter VRCS Jets
U_2	J_2	
U_3	J_3	
U_4	J_4	
U_5	J_5	
U_6	J_6	
U_7	T_x	IPS Torque Motors
U_8	T_y	
U_9	T_z	
U_{10}	J^x	Jets simulating crew motion along Orbiter body axes
U_{11}	J^y	
U_{12}	J_z	

OUTPUT(Sensors)

Y

Y_1	$\dot{\theta}_z$	IPS Rate Gyros
Y_2	$\dot{\theta}_y$	
Y_3	$\dot{\theta}_x$	
Y_4	a_x	IPS Accelerometers
Y_5	a_y	
Y_6	a_z	
Y_7	$\dot{\theta}_{gy}$	Aux. Gimbal Rate Gyros
Y_8	$\dot{\theta}_{gz}$	
Y_9	β_x	P/OF Mask Tilt
Y_{10}	β_y	
Y_{11}	β_z	
Y_{12}	x_1	P/OF Boom Tip Deflection
Y_{13}	x_2	
Y_{14}	δ_1	Line of Sight Angle
Y_{15}	δ_2	

PINHOLE OCCULTER STATE VECTOR DEFINITION

(ELEVATION GIMBAL AXIS REDUCED ORDER MODEL)

STATE VECTOR

X

X_1	$\dot{\theta}^3_1$	Elevation gimbal rate
X_2	$\dot{\eta}^6_1$	P/OF Bending Modes
X_3	$\dot{\eta}^6_4$	
X_4	$\dot{\eta}^6_6$	
X_5	$\dot{\eta}^6_8$	
X_6	θ^3_1	Generalized Coordinates corres- ponding to the above Generalized Velocities
X_7	η^6_1	
X_8	η^6_4	
X_9	η^6_6	
X_{10}	η^6_8	

INPUT(Actuator Commands)

U

U_1 $[T_y]$ IPS Pitch Torque Motor

OUTPUT(Sensors)

Y

Y_1	$\dot{\theta}_y$	Pitch rate gyro
Y_2	β_y	Mask tilt
Y_3	x_1	Boom Tip Deflection
Y_4	δ_2	LOS Angle

(CROSS ELEVATION GIMBAL AXIS REDUCED ORDER MODEL)

STATE VECTOR

X

X_1	$\dot{\theta}^4_1$	X Elevation Gimbal Rate
X_2	$\dot{\eta}^6_2$	P/OF Bending Modes
X_3	$\dot{\eta}^6_5$	
X_4	$\dot{\eta}^6_7$	
X_5	θ^4_1	Generalized Coordinates corres- ponding to the above Generalize Velocities
X_6	η^6_2	
X_7	η^6_5	
X_8	η^6_7	

INPUT(Actuator Commands)

U

U_1 $[T_Z]$ IPS Yaw Torque Motor

Y

Y_1	$\dot{\theta}_x$	Yaw rate gyro
Y_2	β_x	Mask tilt
Y_3	x_2	Boom Tip Deflection
Y_4	δ_1	LOS Angle

(ROLL GIMBAL AXIS REDUCED ORDER MODEL)

STATE VECTOR

X

X_1	$\dot{\theta}^5_1$	Roll gimbal rate
X_2	$\dot{\eta}^6_3$	P/OF Torsional Bending Mode
X_3	θ^5_1	Generalized Coordinates
X_4	η^6_3	

INPUT(Actuator Commands)

U

U_1 $[T_X]$ IPS Roll Torque Motor

Y

Y_1	$\dot{\theta}_z$	Roll rate gyro
Y_2	β_z	Mask twist

ORBITER/POF TREETOPS CONTROLLER FORM AND FUNCITON

- CONTINUOUS CONTROLLER:
  - Sensor Dynamics
  - Actuator Dynamics
  - Crew Motion Shaping Filter

- USER CONTROLLER:
  - Sensor Errors
  - Actuator Errors
  - Vernier RCS Orbiter Attitude Control Law

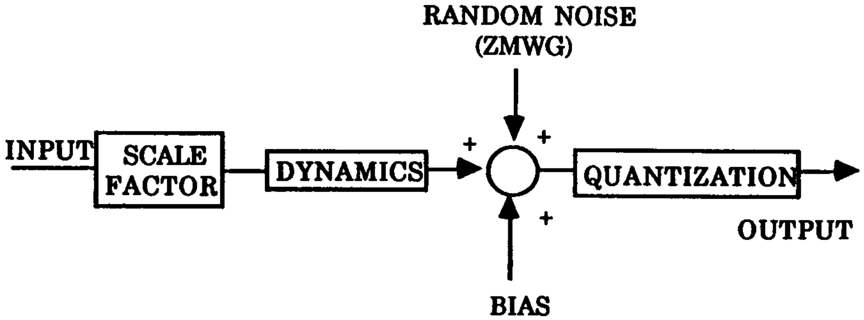
    Phase Plane

    Jet Select

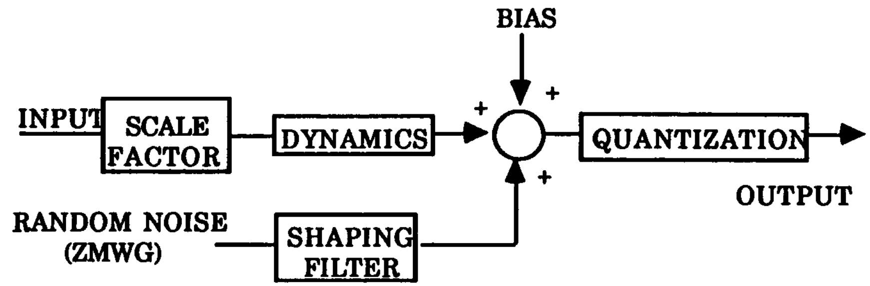
    State Estimator

    Attitude Processor
  - IPS 3 Axis Gimbal Pointing Sampled Data Control Law

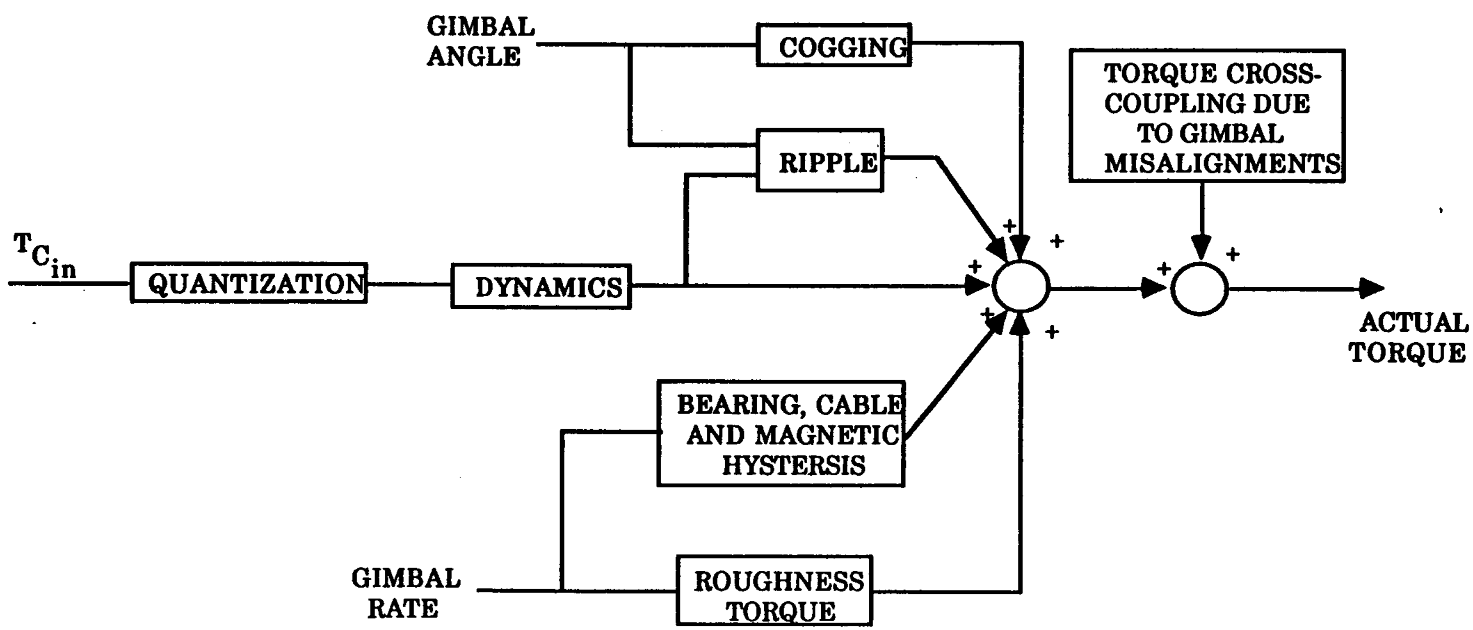
SENSOR/ACTUATOR ERROR MODEL STRUCTURE



LOS & LASER INTERFEROMETER ERROR SOURCE MODEL

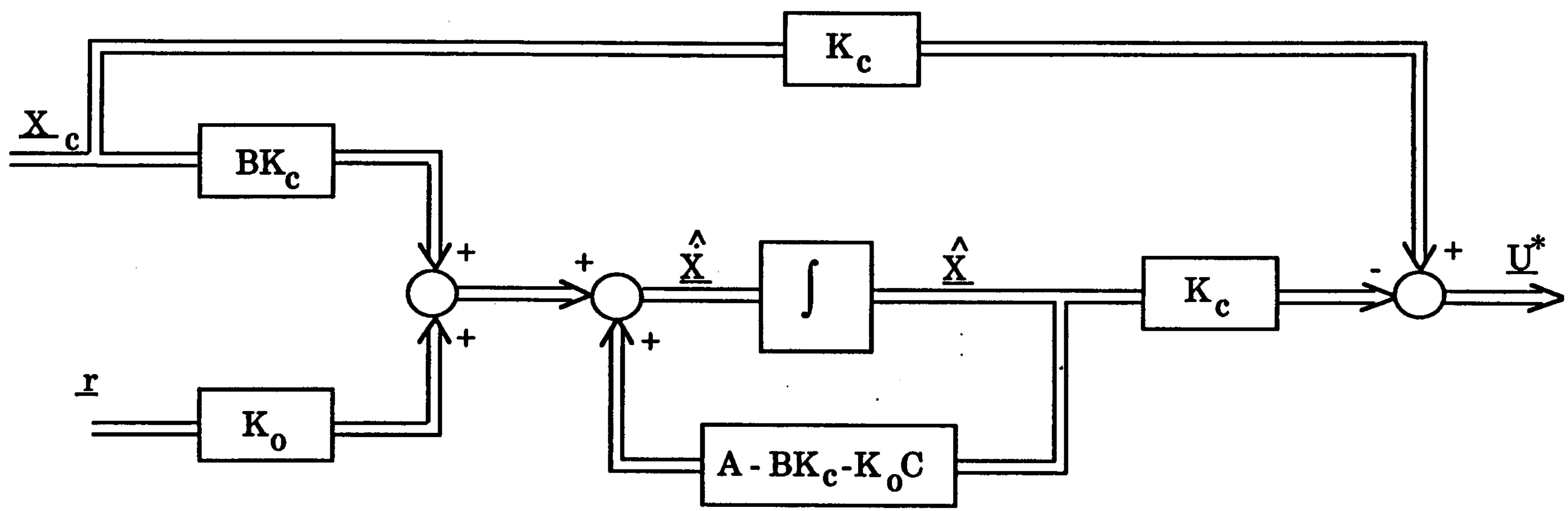


RATE GYRO ERROR SOURCE MODEL



TORQUE MOTOR ERROR SOURCE

IPS 3 AXIS GIMBAL POINTING COMPENSATOR MODEL(CONTINUOUS)



$A' = (A - BK_c - K_oC)$

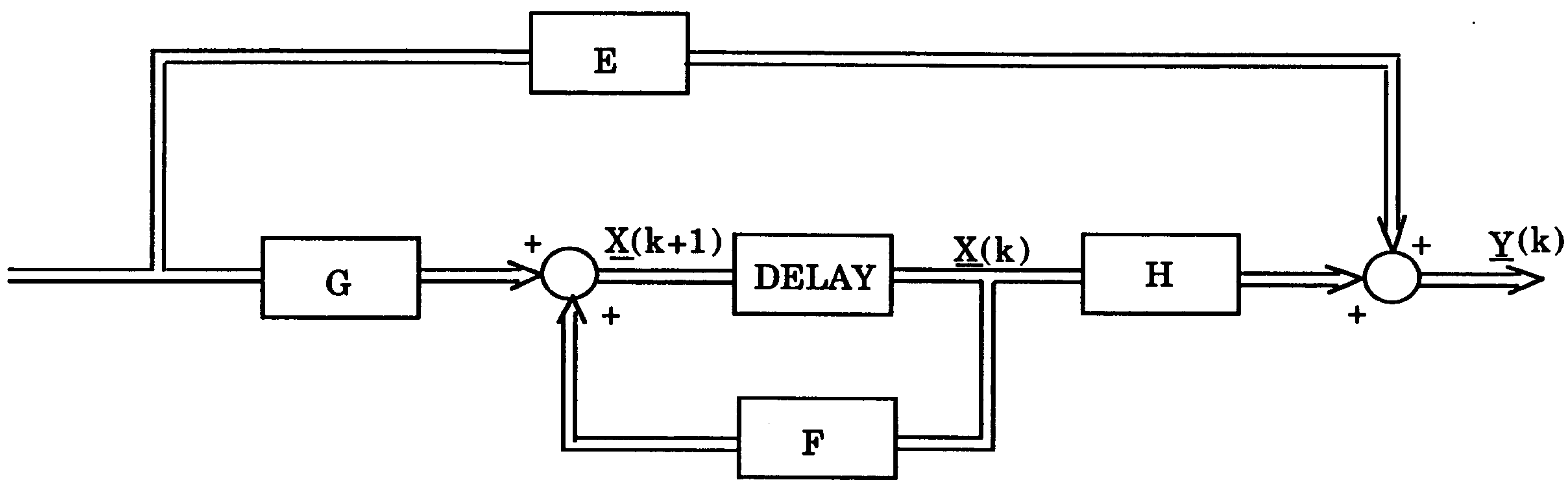
$B' = BK_c$

$C' = -K_c$

$D' = (K_c \mid 0)$

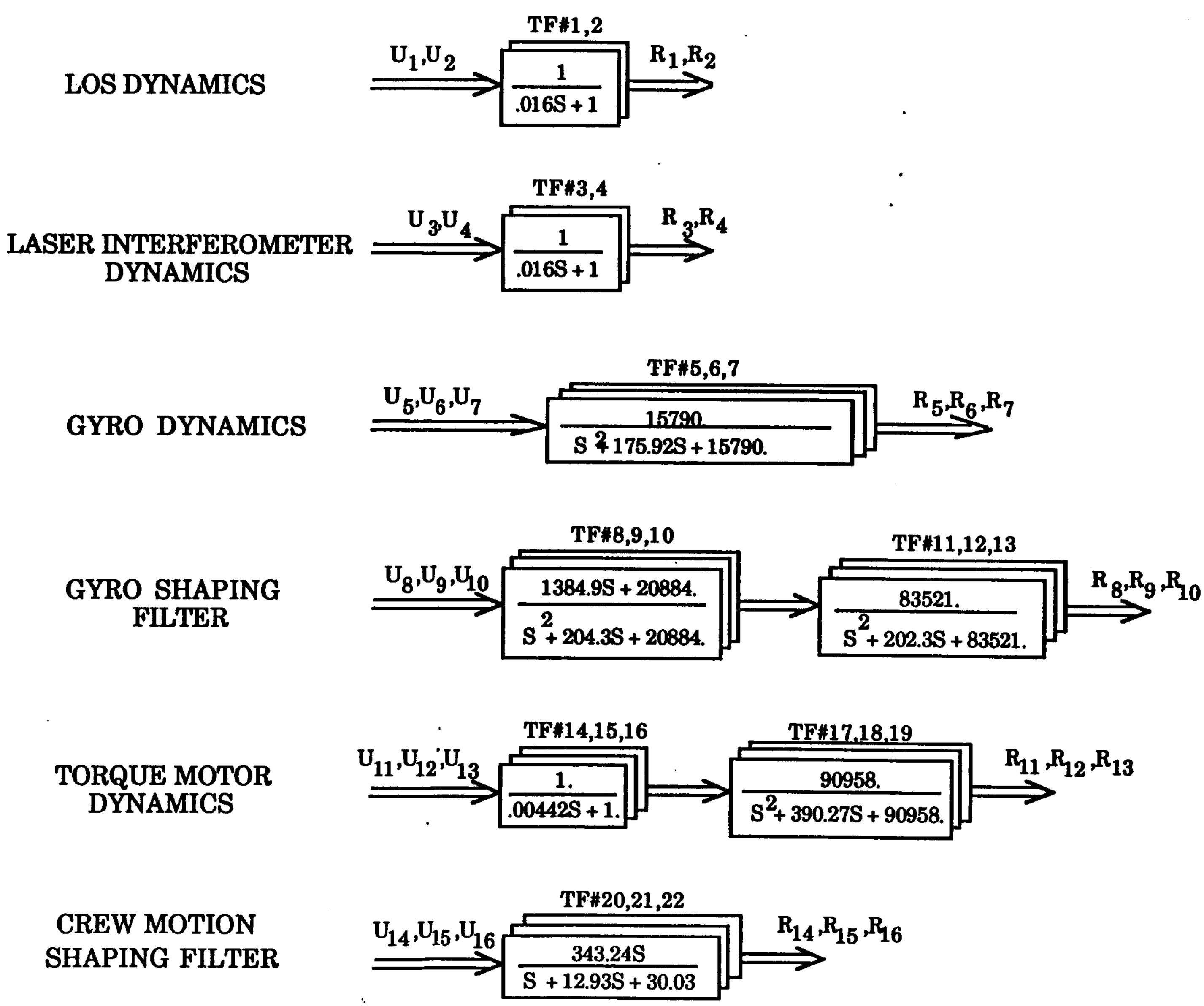
$$\underline{U} = \begin{bmatrix} \underline{X}_c \\ \underline{r} \end{bmatrix}$$

IPS 3 AXIS GIMBAL POINTING COMPENSATOR MODEL(DISCRETE)



$$U(k) = \begin{bmatrix} \underline{X}_c \\ \cdots \\ \underline{r} \end{bmatrix}$$

ORBITER/POF TREETOPS CONTINUOUS CONTROLLER DEFINITION



ORBITER/POF TREETOPS CONTINUOUS CONTROLLER INPUT DATA

OUTPUT DATA

OUTPUT#	SOURCE TYPE	SOURCE ID#	GAIN
1	T	1	1.
2	T	2	1.
3	T	3	1.
4	T	4	1.
5	T	5	1.
6	T	6	1.
7	T	7	1.
8	T	8	1.
9	T	9	1.
10	T	10	1.
11	T	11	1.
12	T	12	1.
13	T	13	1.
14	T	14	0.
15	T	15	0.
16	T	16	0.

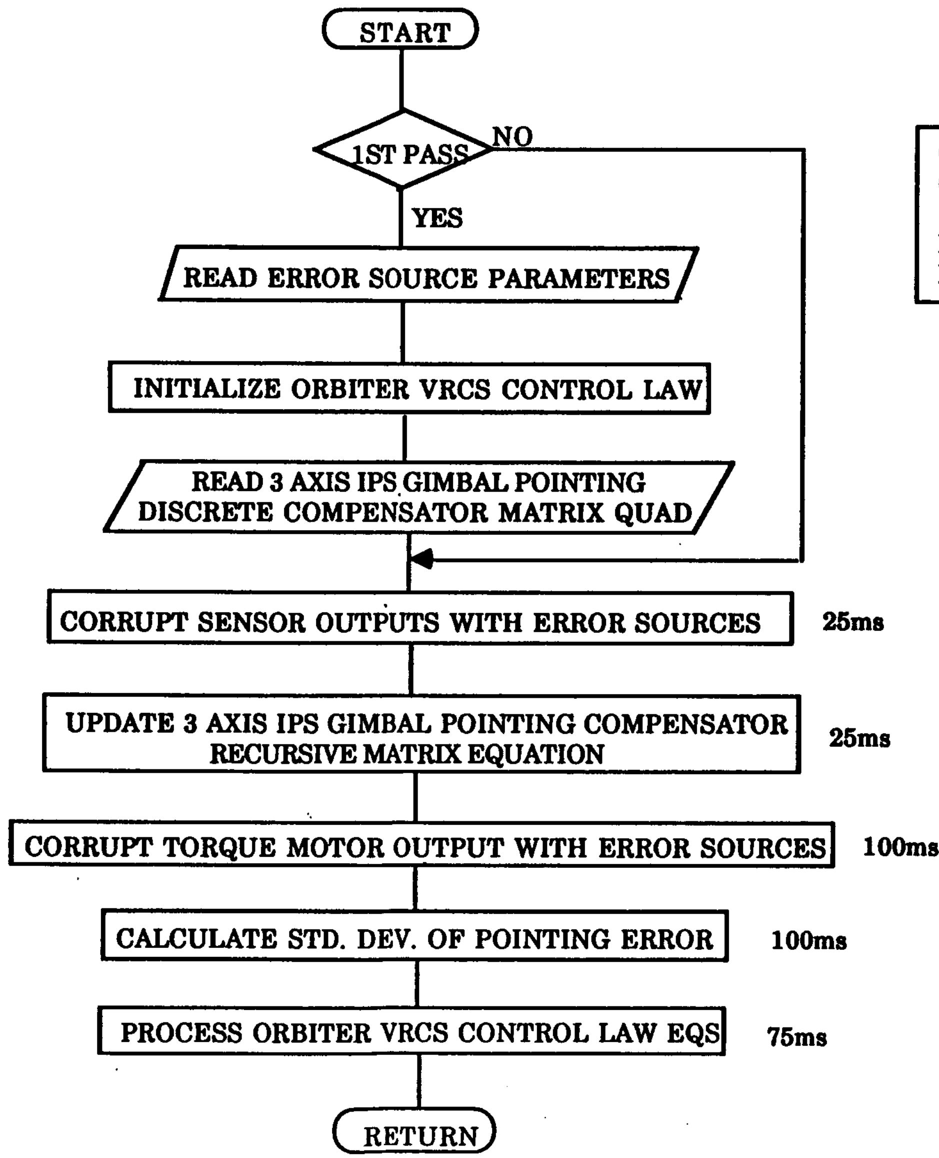
T = TRANSFER FUNCTION

I = CONTROLLER INPUT

TRANSFER FUNCTION DATA

T#	SOURCE TYPE	SOURCE ID#	NUMERATOR ORDER	NUMERATOR COEFFICIENTS	DENOMINATOR ORDER	DENOMINATOR COEFFICIENTS
1	I	1	0	1.	1	1.,.016
2	I	2	0	1.	1	1.,.016
3	I	3	0	1.	1	1.,.016
4	I	4	0	1.	1	1.,.016
5	I	5	0	15790.	2	15790.,175.92,1.
6	I	6	0	15790.	2	15790.,175.92,1.
7	I	7	0	15790.	2	15790.,175.92,1.
8	I	8	1	20884.,1384.9	2	20884.,204.3,1.
9	I	9	1	20884.,1384.9	2	20884.,204.3,1.
10	I	10	1	20884.,1384.9	2	20884.,204.3,1.
11	T	8	0	83521.	2	83521.,202.3,1.
12	T	9	0	83521.	2	83521.,202.3,1.
13	T	10	0	83521.	2	83521.,202.3,1.
14	I	11	0	1.	1	1.,.00442
15	I	12	0	1.	1	1.,.00442
16	I	13	0	1.	1	1.,.00442
17	T	14	0	90958.	2	90958.,390.27,1.
18	T	15	0	90958.	2	90958.,390.27,1.
19	T	16	0	90958.	2	90958.,390.27,1.
20	I	14	1	0.,343.24	2	30.03,12.93,1.
21	I	15	1	0.,343.24	2	30.03,12.93,1.
22	I	16	1	0.,343.24	2	30.03,12.93,1.

ORBITER/POF TREETOPS USER CONTROLLER DEFINITION



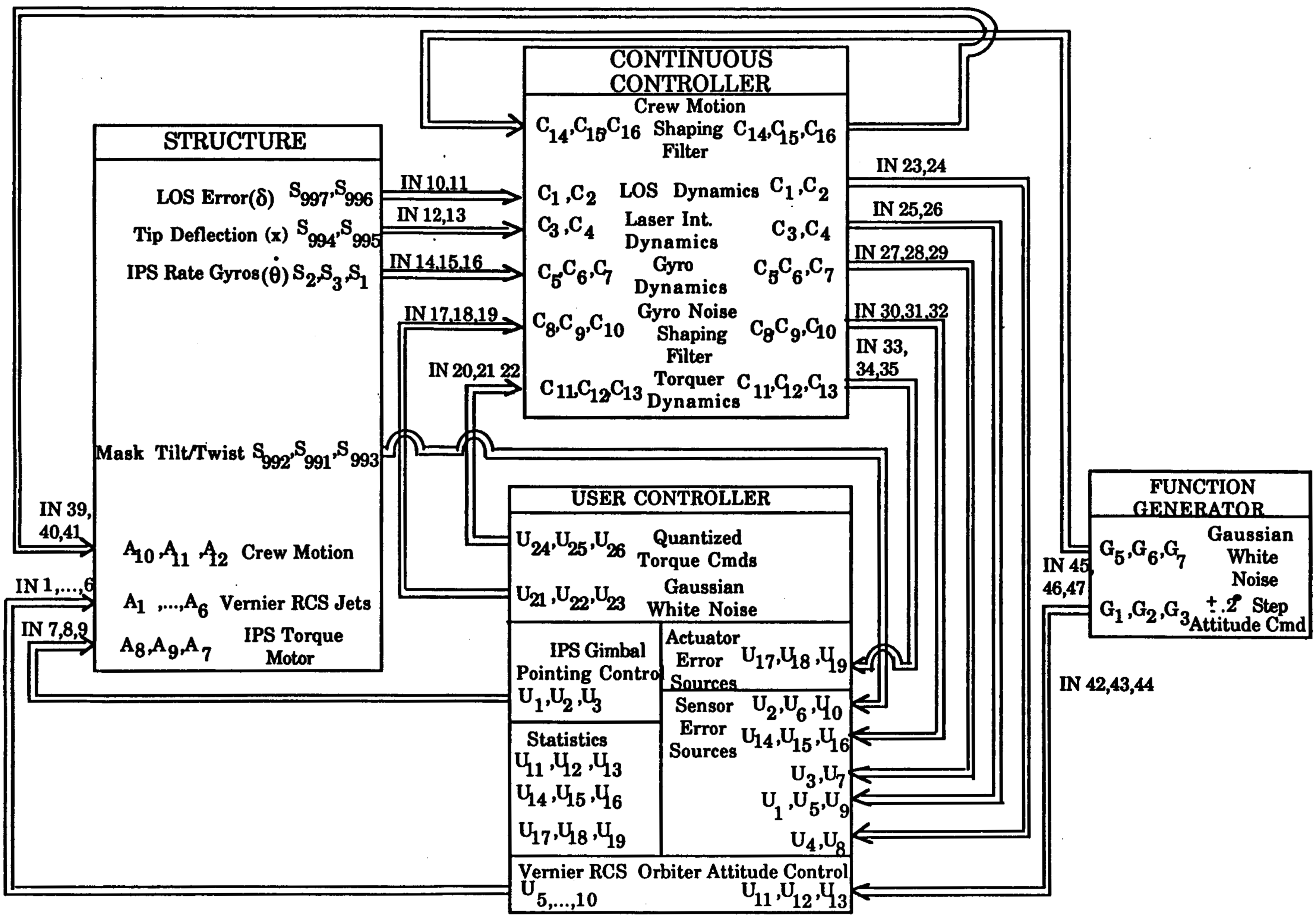
INPUT DATA

CONTROLLER TYPE = USER
CONTROLLER UPDATE INTERVAL = .025sec
NUMBER OF INPUTS = 10
NUMBER OF OUTPUTS=19

ORBITER/POF TREETOPS FUNCTION GENERATOR INPUT DATA

FUNCTION GENERATOR ID#	TYPE	MAGNITUDE	DESCRIPTION
1	STEP	-.2°	3 AXIS ORBITER VRCS ATTITUDE DISTURBANCE
2	STEP	+.2°	
3	STEP	-.2°	
5	USER DEFINED		
6	USER DEFINED		GAUSSIAN WHITE NOISE ($\mu = 0$, $\sigma = 1.$)
7	USER DEFINED		

ORBITER/POF TREETOPS INTERCONNECT DEFINITION



COMPENSATOR DESIGN

- Problem Description
- Disturbance Environment
- Controllability and Observability
- System Dynamics
- LQG/LTR Design
    - Elevation Axis
        - Structured Uncertainty Analysis
    - Cross Elevation Axis
    - Roll Axis

Problem Description

Sensors

- O Line Of Sight (LOS) Sensor
  - oo LOS Error
  - oo Mask Tilt
  - oo Tip Deflection
- O IPS Rate Gyro
- O IPS Accelerometer

Actuators

- O Three Instrument Pointing System (IPS) Gimbal Torque Motors
- O Six Shuttle Reaction Jets

Dominant Disturbances

- O Crew Motion
- O Reaction Control System (RCS) Jet Firings For Attitude Corrections

Control Objective

- O Hold Line Of Sight Error To Less Than 1.0 arc-sec

DOMINANT DISTURBANCES
CREW MOTION

O Crew Activities Being Modeled

Low Level Crew Motion[1]
- Console Operation
- Respiration, Coughing, Sneezing
- Arm Motion
- Leg Motion
- Swaying Motion

High Level Crew Motion[1]
- Bowing Motion
- Arm Flapping
- Crouch and Pushoff
- Soaring
  - o One Man Normal and One Man Forceful
  - o Two Man Normal and Two Man Forceful

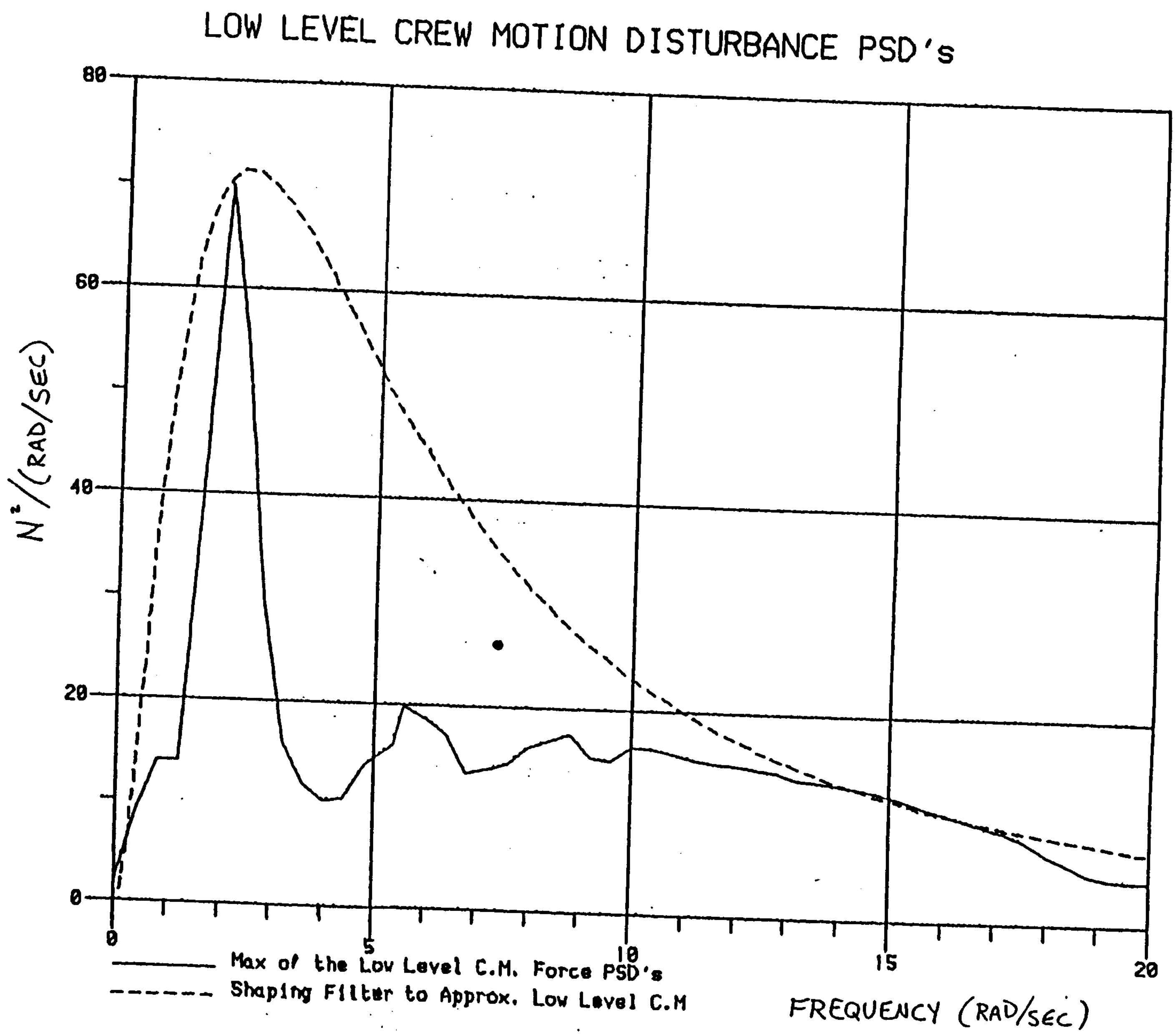
O Modeled As Shaped White Noise[2]

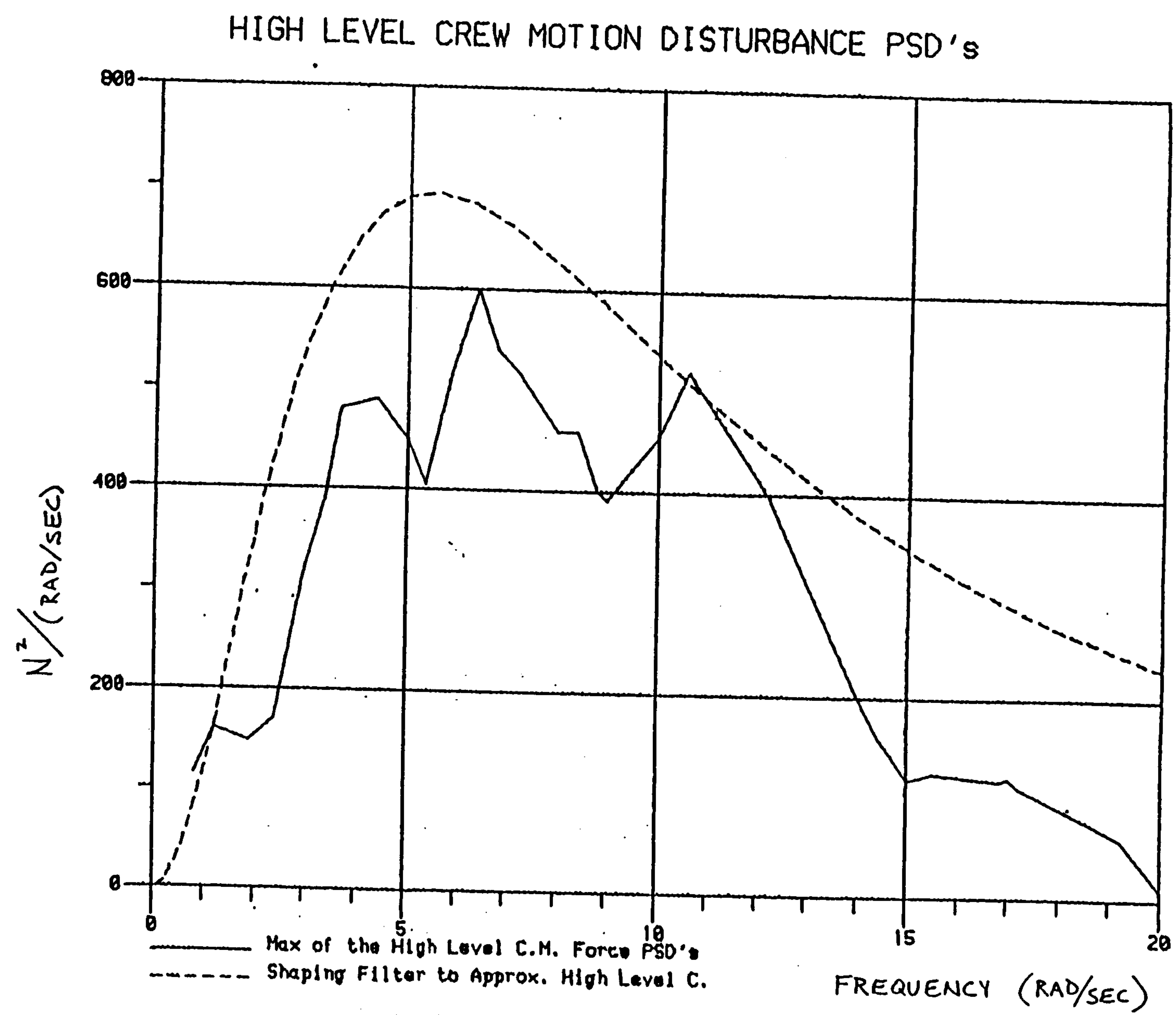
Shaping Filter

$$G(s) = \frac{2 k \zeta \omega_n s}{s^2 + 2 \zeta \omega_n s + \omega_n^2}$$

1. "Handbook on Astronaut Crew Motion Disturbance For Control System Design ", NASA Reference Publication 1025.

2. February 24, 1986, To: J.F.L. Lee, From: M. J. Kaindl,"Stochastic Crew Motion Models For Flexible Space Station Response Analysis"


LOW LEVEL CREW MOTION DISTURBANCE PSD's
80
60
40
20
0
N²/(RAD/SEC)
0
5
10
15
20
FREQUENCY (RAD/SEC)
Max of the Low Level C.M. Force PSD's
Shaping Filter to Approx. Low Level C.M


HIGH LEVEL CREW MOTION DISTURBANCE PSD's
800
600
400
200
0
$N^2/(RAD/SEC)$
0
5
10
15
20
Max of the High Level C.M. Force PSD's
Shaping Filter to Approx. High Level C.
FREQUENCY (RAD/SEC)

Honeywell

DOMINANT DISTURBANCES
RCS JET FIRING STUDY

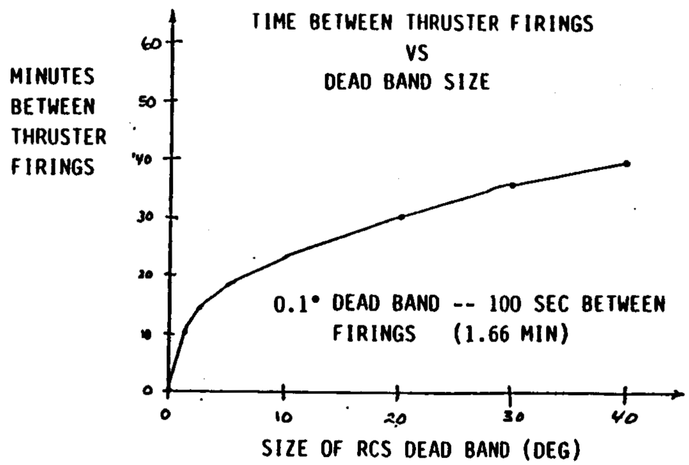
J.F.P. 1/27/87

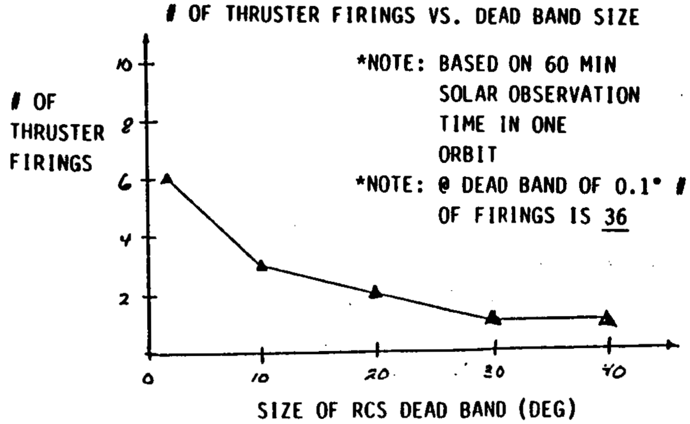
OBJECTIVE : Minimize Number Of Jet Firings And Therefore Disturbances From Jets.

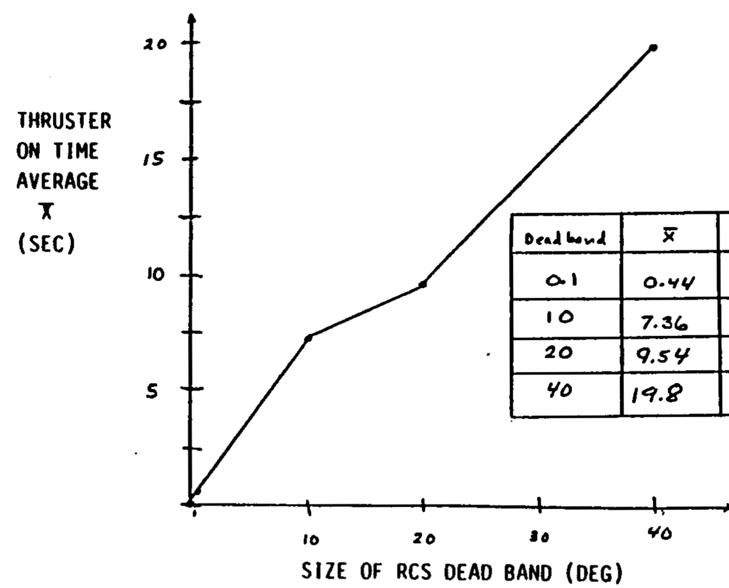
The Effects Of Increased Deadband Size On The Number And Duration Of Jet Firings

O Summary Of Results

>> **Disturbances Due To RCS Jet Firings For Attitude Corrections May Be Significantly Decreased By Increasing The RCS Attitude Deadband.**





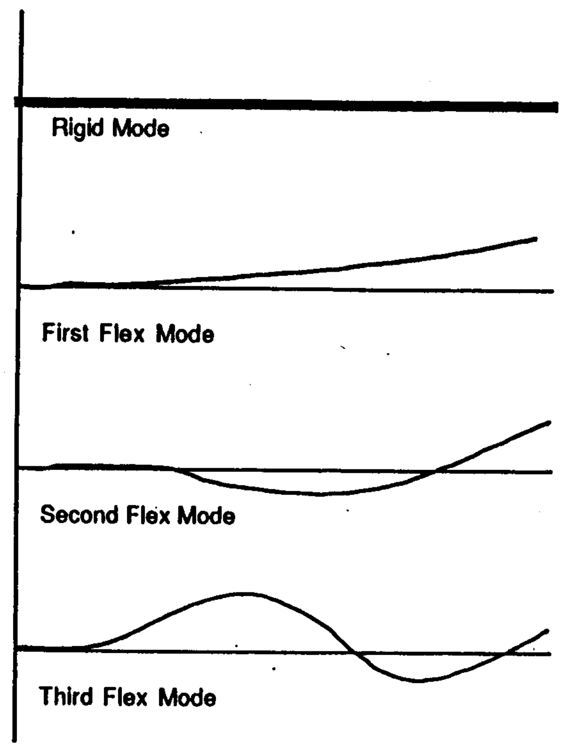


Deadband	$\bar{X}$	σ	# Samp.
0.1	0.44	0.4	28
10	7.36	5.11	4
20	9.54	8.26	4
40	19.8	17.01	3

* RCS MINIMUM ON TIME = 0.08 SEC.

Controllability And Observability Of System Modes

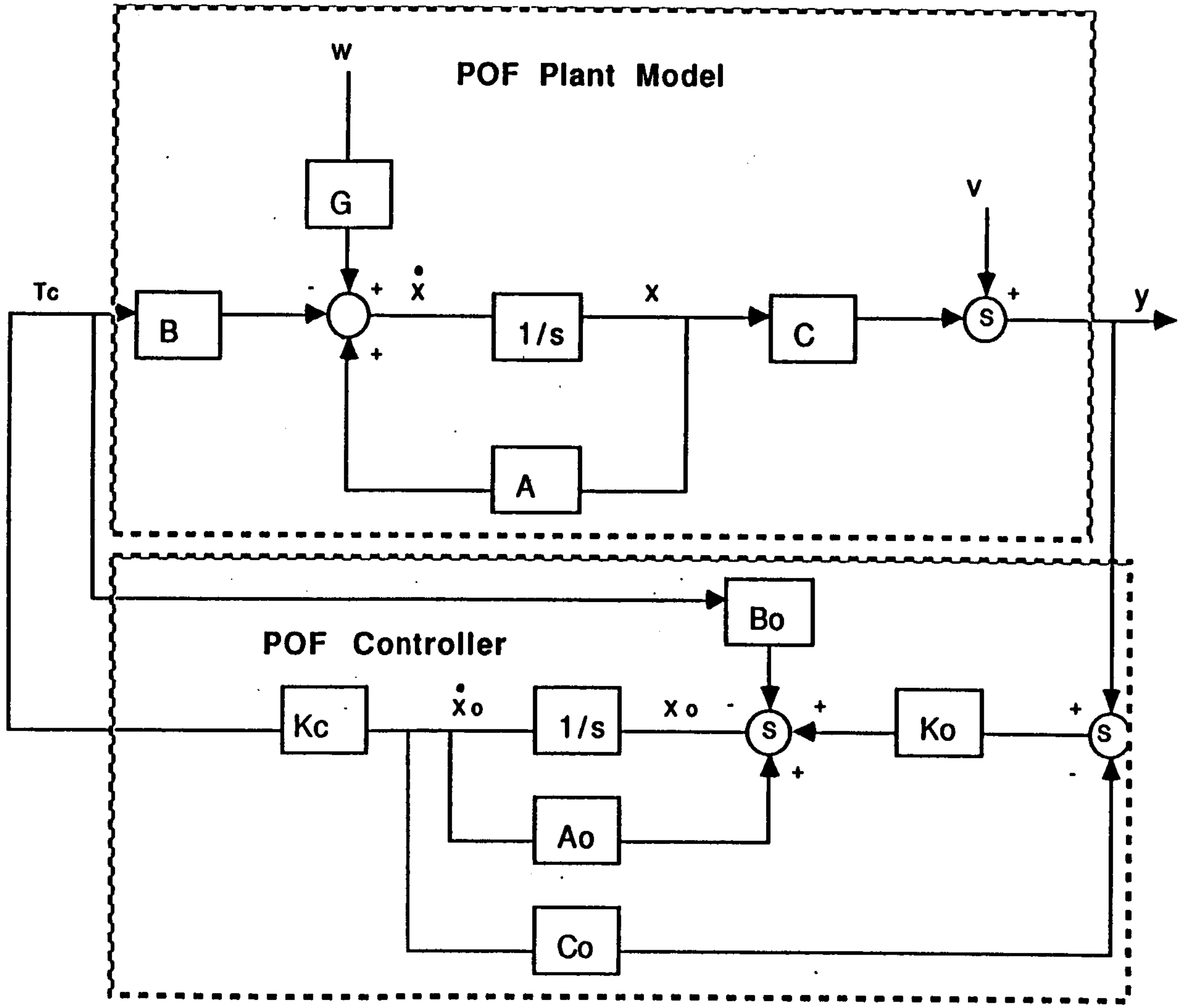
PINHOLE OCCULTER MODE SHAPES



Mode	Controllability Index	Observability Index
Rigid Mode	1.0	1.0
First Flex Mode	8.5 E-17	4.6 E-5
Second Flex Mode	1.8 E-17	6.3 E-2
Third Flex Mode	1.8 E-19	2.72 E-4

- O All Three Modes Are Observable Through The Line Of Sight Sensor.
- O Only The Rigid Body Mode Of The Beam Is Controllable With The Existing IPS Gimbal Actuator.

O System And Controller Block Diagram



Ko : Kalman Filter Gains Kc : Regulator Gains Tc : Controller Torque

Reduced Order Optimal Controller Design And Analysis

O State Space Equations

$$\begin{bmatrix} \dot{\underline{x}} \\ \dot{\underline{x}}_o \end{bmatrix} = \begin{bmatrix} A & -B\,Kc \\ Ko\,C & Ao - Bo\,Kc - Ko\,Co \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{x}_o \end{bmatrix} + \begin{bmatrix} B \\ B_o \end{bmatrix} T_c$$

$$y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{x}_o \end{bmatrix}$$

Optimal Controller Design And Analysis

Regulator Design

O Cost Function : Minimize The Steady State Error Covariance Of The Line Of Sight Error (LOS Error $\equiv \delta$).

$$J = \int x^T Q x + x^T Q R u + u^T R u \, dt$$

Where

$Q = \beta^T \alpha^2 \beta$ $\qquad$ α : *penalty function on* δ

$R = 1.0$ n-m $\qquad$ $\beta : \delta = \beta x$

O The Value Of α, (The Penalty Function On LOS Error δ) Is Constrained By The Stability Of :

1) The Linear Quadratic Regulator For The Reduced Order Model (Kc)*

2) The Overall Controller For The Reduced Order Model*

3) The Full Order Closed Loop System

* Stability Not Guaranteed Because Mast Flex Mode States Are Not Fully Controllable.

Optimal Controller Design And Analysis

Estimator Design

O Cost Function : Minimize The Error Due To Disturbance Effects And Sensor Noise Covariance (w, v).

$$J = \int \underline{w}^T q \, \underline{w} + \underline{v}^T R \, \underline{v} \, dt \qquad R = I \qquad I : \textit{identity matrix}$$

Where The System Is

$$\dot{x} = A_0 \, x_0 + B_0 \, u + G \, \underline{w} \qquad \underline{w} : \textit{system disturbance vector}$$

$$z = C_0 \, x_0 + D_0 \, u + \underline{v} \qquad \underline{v} : \textit{sensor noise vector}$$

and

$$q = diag \{ \rho G \} \qquad \rho : \frac{\textit{disturbance level}}{\textit{noise level}} \qquad G : \textit{disturbance input vector}$$

O The Value Of ρ Is Constrained By The Stability Of :

1) The Overall Controller For The Reduced Order Model*

3) The Full Order Closed Loop System

* Stability Not Guaranteed Because Mast Flex Mode States Are Not Fully Controllable.

Optimal Controller Design And Analysis
Elevation Axis

O The Following Regulator And Estimator Weights Where Found To Yield The Maximum System Performance For The Elevation Axis *.

α : *weighting on line of sight error* $= 4.0 \times 10^4$

ρ : *weighting on disturbance level to sensor noise level* $= 5.0 \times 10^6$

* Note : *performance limited by inability to control flex modes with existing actuator.*

O These Weightings Yielded The Following Performance, Stability And Robustness Measures.

oo Performance Measures

B/W	ξ	$\delta\ell$-rms	$\delta\ell$	δh-rms	δh	Ts
0.658 rad/s	0.687	1.0 arc-sec-rms	2.5 arc-sec	1.4 arc-sec-rms	3.5 arc-sec	11 sec

oo Stability Margins

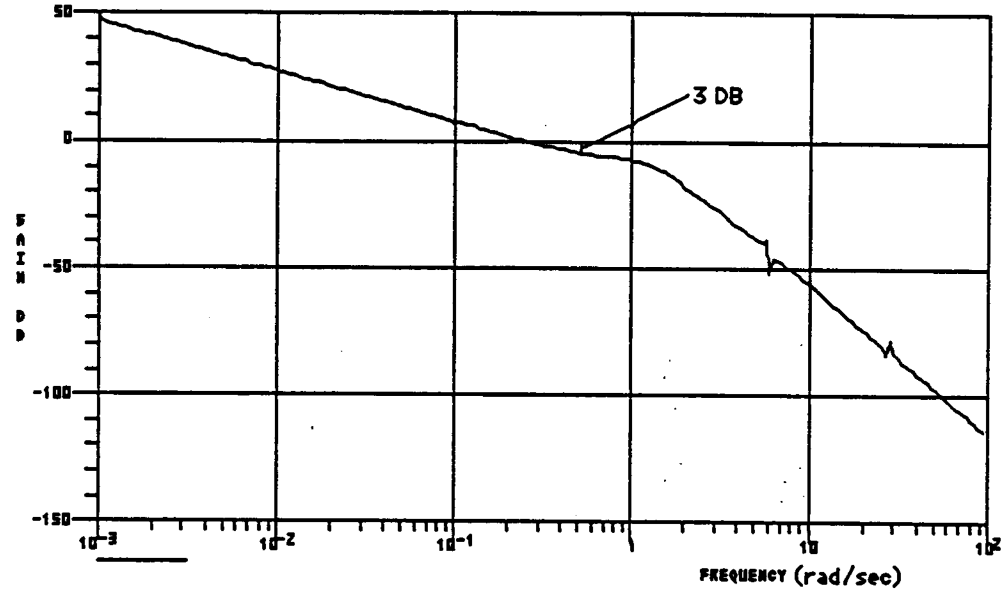
gain margin = 3 *db*   *phase margin* = 40 *degrees*

oo Robustness - From Structured Uncertainty , μ Analysis

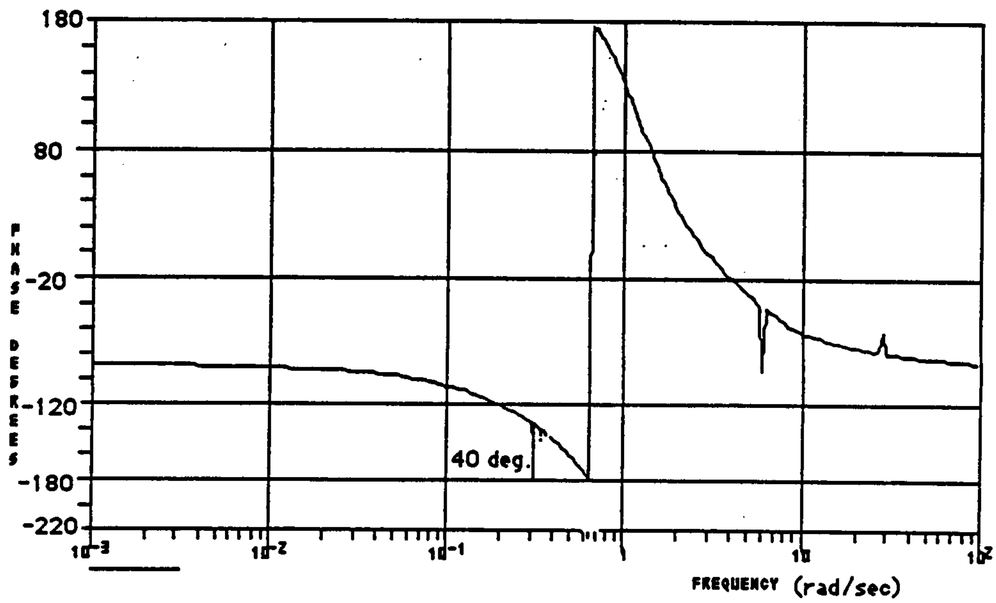
$\mu = 2.2^{**}$ {∴ *will tolerate up to* 4.54% *uncertainty on modal frequency*}

** *for* 10% *uncertainty in the three modal frequencies.*

HONEYWELL | ELEVATION AXIS | J.F.P. 1/29/87



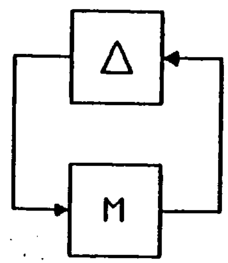
BODE GAIN PLOT



BODE PHASE PLOT

Honeywell

STRUCTURED UNCERTAINTY ANALYSIS – μ TEST

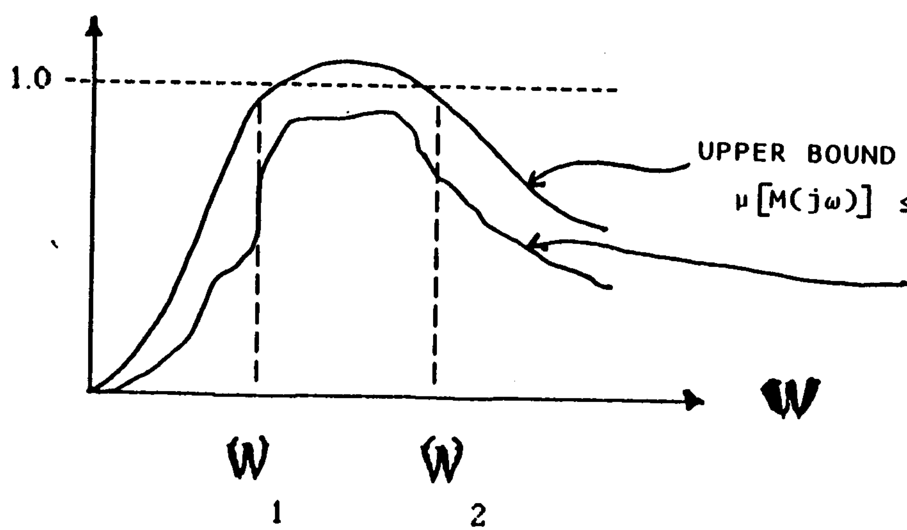


Δ: UNCERTAINTY MATRIX
- STIFFNESS, DAMPING
- AERODYNAMIC COEF

M: CLOSED LOOP SYSTEM

STABILITY CRITERIA

- DET $(I + M\Delta) \neq 0$ (NON-SINGULAR) OVER Δ SET
- TRUE IF $\mu(M(j\omega)) < 1$ $\quad \mu(M) = [\min_{\Delta \epsilon X} \{\sigma(\Delta) | \det(I + M\Delta) = 0\}]^{-1}$



UPPER BOUND ON $\mu(M)$ IS

$$\mu[M(j\omega)] \leq \min_{D \epsilon \mathcal{D}} \bar{\sigma}(DMD^{-1})$$

LOWER BOUND ON $\mu(M)$ IS

$$\max_{u \epsilon U} \rho(Mu) \leq \mu[M(j\omega)]$$

ρ : SPECTRAL RADIUS ~ MAX $|\lambda_i|$
u,D: UNITARY MATRIX ~ SQUARE ORTHONORMAL MATRIX

References

[1] J.C. Doyle, *Lecture Notes*, 1984 ONR/Honeywell Workshop on Advances in Multivariable Control

[2] J.C. Doyle, "Analysis of feedback systems with structured uncertainty," *IEE Proceedings, Part D*, V129, No. 6, Nov., 1982

[3] J.C. Doyle, J.E. Wall, and G. Stein, "Performance and robustness analysis with structured uncertainty," 1982 CDC

[4] J.C. Doyle, "Synthesis of robust controllers and filters," 1983 CDC

[5] C.C. Chu and J.C. Doyle, "On inner-outer and spectral factorizations," 1984 CDC

[6] C.C. Chu and J.C. Doyle, "The general distance problem in H_∞ synthesis," 1985 CDC

[7] I.M. Horowitz, *Synthesis of Feedback Systems*, 1963

[8] G. Zames, "Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms and approximate inverses," *IEEE Trans. Auto. Control*, AC-26, 1981

[9] G. Zames and B.A. Francis, "Feedback, minimax sensitivity, and optimal robustness," *IEEE Trans. Auto. Control*, AC-28, April, 1983

[10] M. Vidyasagar, to appear

[12] G. Zames, "On the input-output stability of nonlinear time-varying feedback systems, parts I and II," *IEEE Trans. Auto. Control*, AC-11, 1966

[13] J.C. Doyle and G. Stein, "Multivariable feedback design: comcepts for a classical/modern synthesis," *IEEE Trans. Auto. Control*, AC-26, Feb., 1981

[14] M. Fan and A. Tits, "Characterization and efficient computation of the structured singular value," submitted to *IEEE Trans. Auto. Control*

[15] M.G. Safonov and J.C. Doyle, "Minimizing conservativeness of robustness singular values," in S.G. Tzefestas (ed.), *Multivariable Control*, 1984, D. Reidel Publishing Company

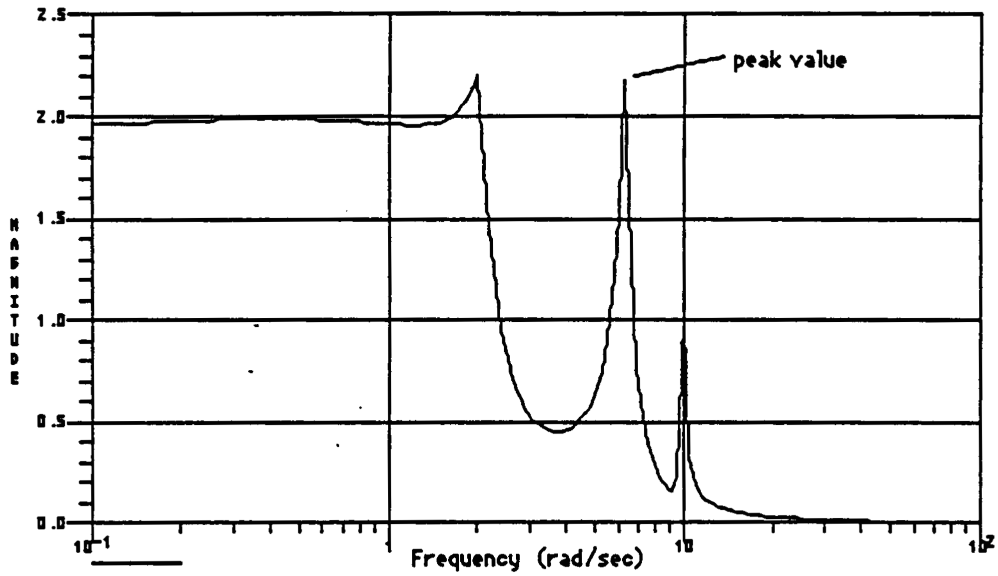
[16] D.C. Youla, H.A. Jabr, and J.J. Bongiorno, Jr., "Modern Wiener-Hopf design of optimal controllers: Part II", *IEEE Trans. Auto. Control*, AC-21, 1976

[17] C. Davis, W.M. Kahan, and H.F. Weinberger, "Norm-preserving dilations and their applications to optimal error bounds," *SIAM J. Num. Anal.*, 1982

[18] K. Glover, "All optimal Hankel-norm approximations of linear multivariable systems and their L^∞-error bounds," *Int. J. Control*, 1984

[19] M.G. Safonov, "Optimal diagonal scaling for infinity norm opimization," 1985 ACC

HONEYWELL   ELEVATION AXIS MU-PLOT   J.F.P. 1/29/8



Mu-Plot For 10% Uncertainty In Three Modal Frequencies

* Shows System Stable With Up To 4.54% Uncertainty On Modal Frequency.

1/2.2 * 10% = 4.54%

Optimal Controller Design And Analysis
Cross Elevation Axis

O The Following Regulator And Estimator Weights Where Found To Yield The Maximum System Performance For The Cross Elevation Axis *.

α : *weighting on line of sight error* $= 4.0 \times 10^4$

ρ : *weighting on disturbance level to sensor noise level* $= 5.0 \times 10^6$

* Note : *performance limited by inability to control flex modes with existing actuator.*

O These Weightings Yielded The Following Performance And Stability

oo Performance Measures

B/W	ξ	δl-rms	δl	δh-rms	δh	Ts
0.641 rad/s	0.705	1.85 arc-sec-rms	5.0 arc-sec	2.4 arc-sec-rms	5.5 arc-sec	12 sec

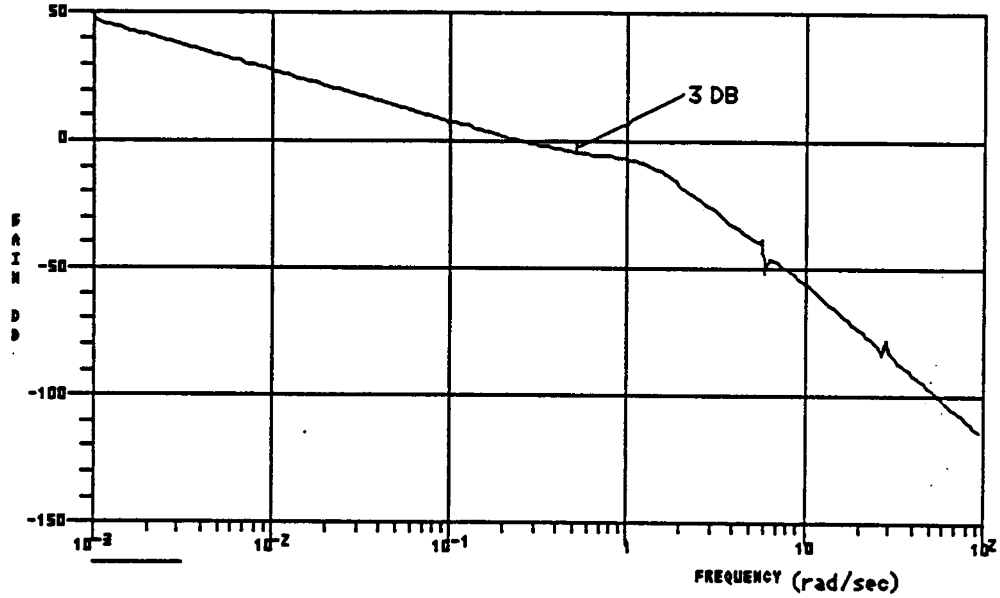
oo Stability Margins

gain margin = 3 *db* *phase margin* = 40 *degrees*

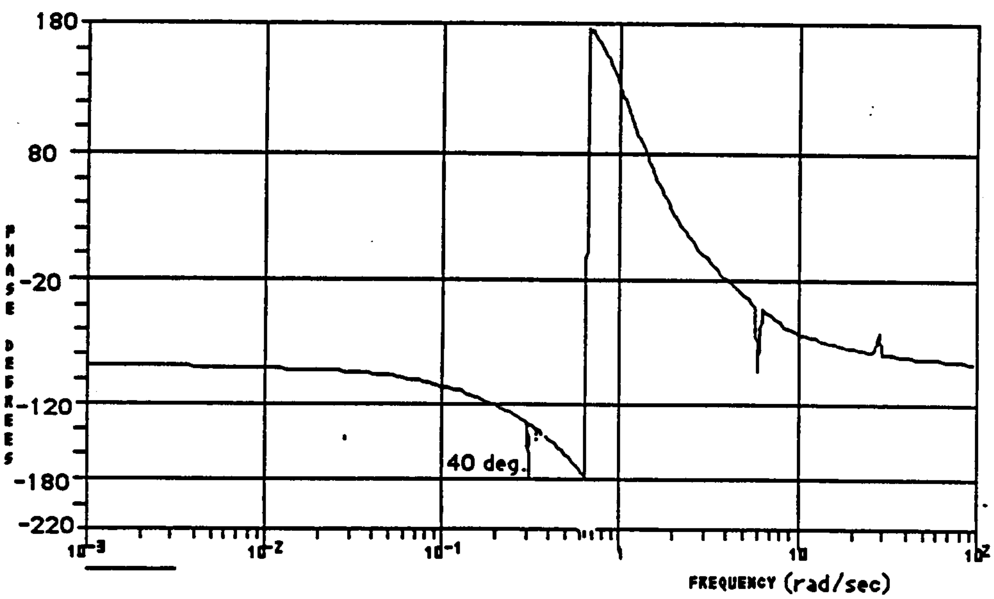
HONEYWELL

CROSS ELEVATION AXIS

J.F.P. 1/29/87



BODE GAIN PLOT



BODE PHASE PLOT

Optimal Controller Design And Analysis
Roll Axis

O The Following Regulator And Estimator Weights Where Found To Yield The Maximum System Performance For The Roll Axis *.

α : *weighting on line of sight error* $= 1.04 \times 10^6$

ρ : *weighting on disturbance level to sensor noise level* $= 1.0 \times 10^5$

* Note : *performance limited by inability to control flex modes with existing actuator.*

O These Weightings Yielded The Following Performance And Stability

oo Performance Measures

B/W	ξ	δl-rms	δl	δh-rms	δh	Ts
0..389 rad/s	0..463	2.6 arc-sec-rms	5.0 arc-sec	2.7 arc-sec-rms	5.5 arc-sec	22 sec

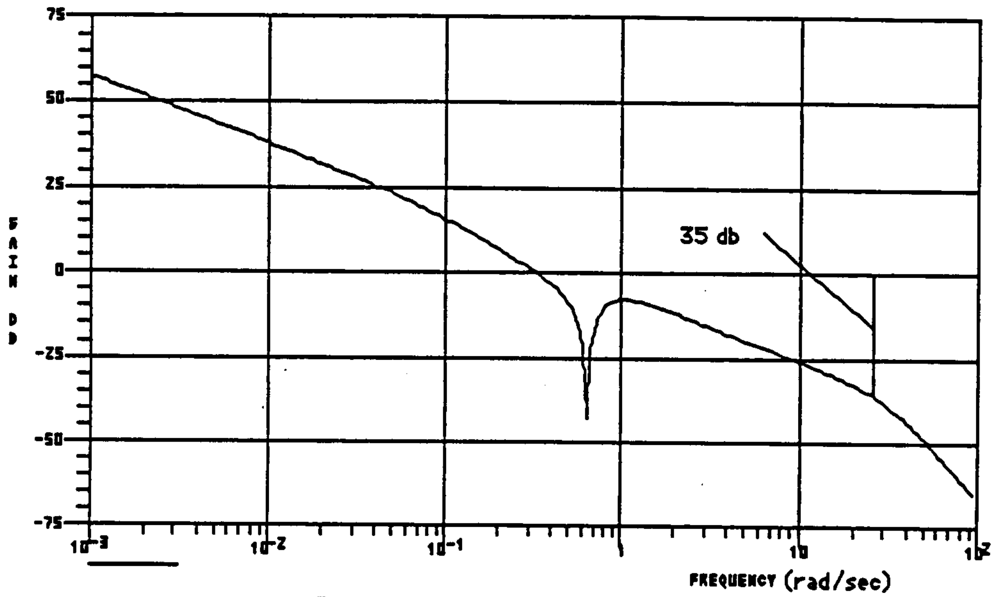
oo Stability Margins

gain margin = 35 *db*　　*phase margin* = 45 *degrees*

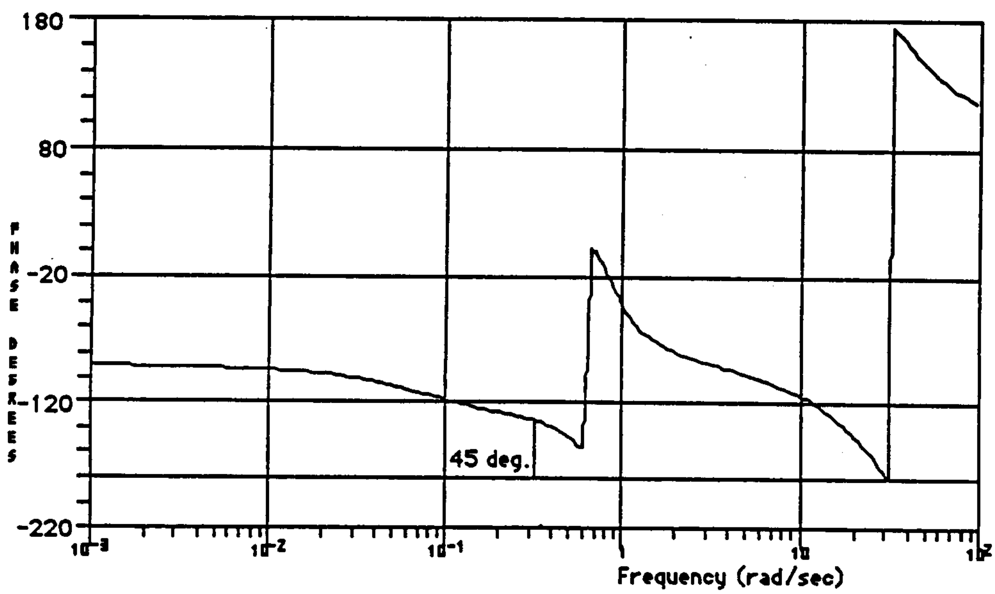
HONEYWELL

ROLL AXIS

J.F.P. 1/29/87



BODE GAIN PLOT



BODE PHASE PLOT

Honeywell

SUMMARY OF RESULTS

J.F.P 1/26/87

O PERFORMANCE

	Elevation	Cross Elevation	Roll
Band Width	0.658 rad/sec	0.641 rad/sec	0.389 rad/sec
Dominant Damping Ratio	0.687	0.7045	0.463
T_s : Step Response Settle Time	11 sec	12 sec	22 sec
$\delta_{\ell-rms}$	1.0 arc-sec-rms	1.85 arc-sec-rms	2.6 arc-sec-rms
δ_{ℓ}	2.5 arc-sec	5.0 arc-sec	5.0 arc-sec
δ_{h-rms}	1.4 arc-sec-rms	2.4 arc-sec-rms	2.7 arc-sec-rms
δ_{h}	3.5 arc-sec	5.5 arc-sec	5.5 arc-sec

δ : Line Of Sight Error h: High Level Crew Motion ℓ: Low Level Crew Motion

Results For Step Response Are Consistent With Those Presented For The Elevation And Cross Elevation Axis In The 1985 Study.

SUMMARY OF RESULTS

J.F.P 1/26/87

O STABILITY

	Elevation	Cross Elevation	Roll
Gain Margin	3 DB	3 DB	35 DB
Phase Margin	40 Degrees	40 Degrees	45 Degrees

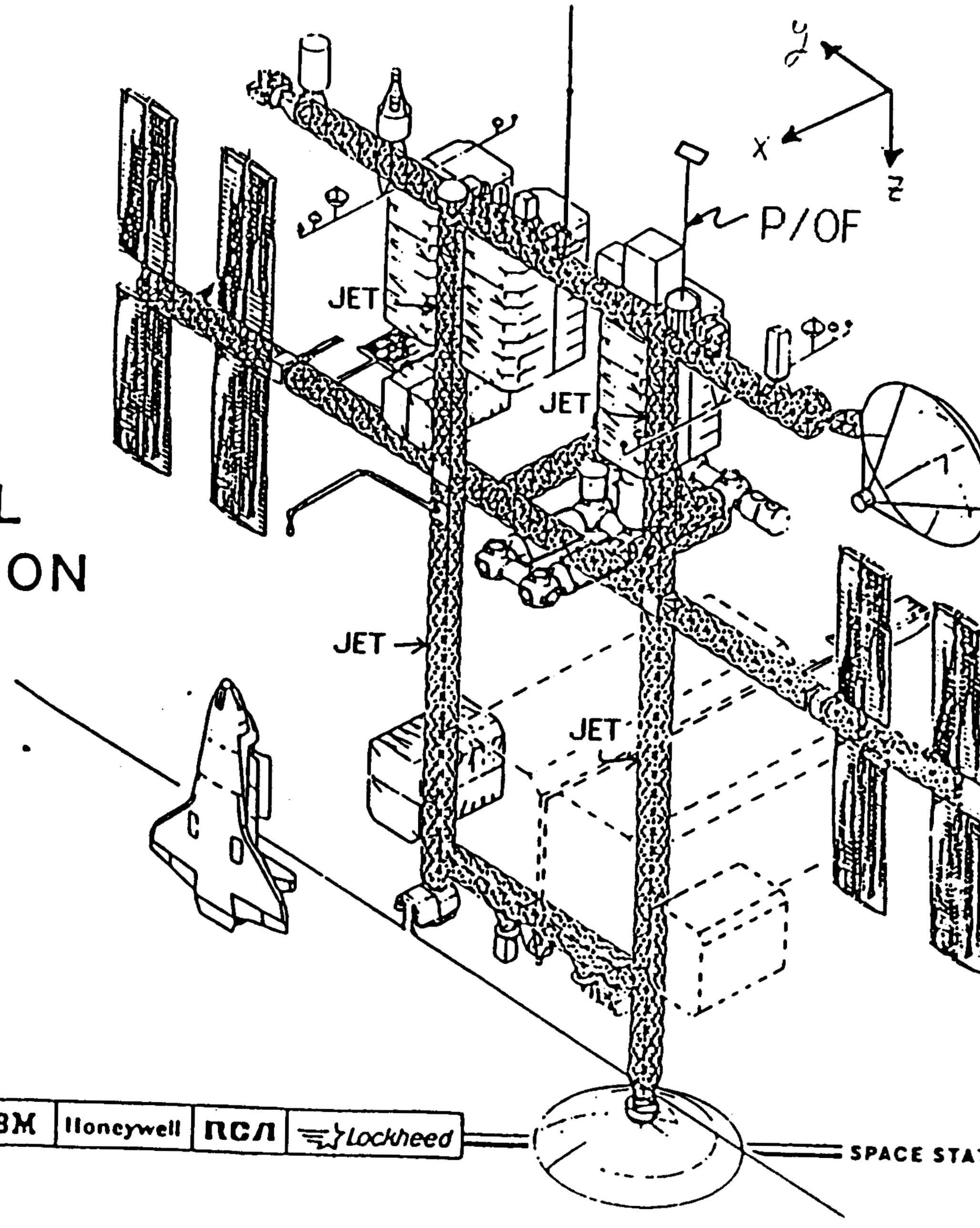
O PRELIMINARY ROBUSTNESS

oo Results From Structured Uncertainty Analysis For 10% Uncertainty In The Three Modal Frequencies

$$\mu = 2.2$$

$\therefore$ *will tolerate up to* 4.54 % *uncertainty on modal frequency*

DUAL KEEL SPACE STATION WITH POF

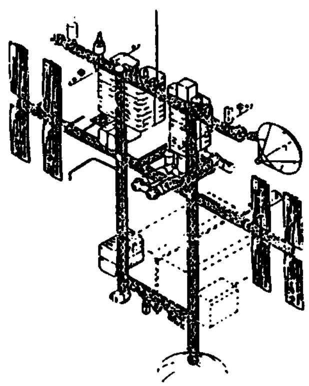


TREETOPS MODELING

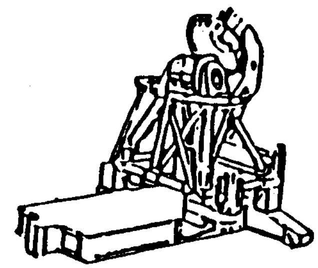
- Body Input Data
- Hinge Input Data
- Sensor/Actuator Input Data
- Controller Input Data
- Function Generation
- Interconnect Definition

SPACE STATION/POF STRUCTURE TOPOLOGY

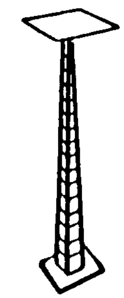
SPACE STATION
(Body 1)

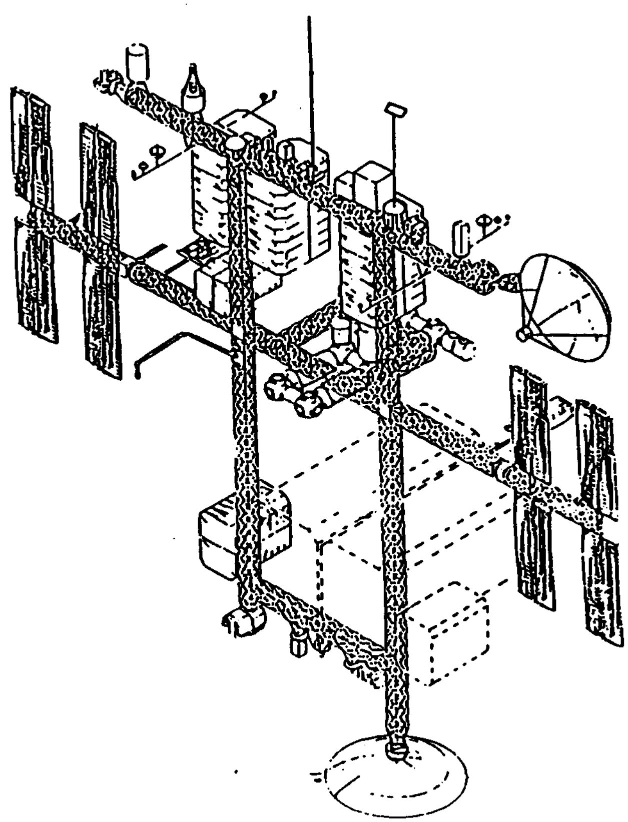


INSTRUMENT
POINTING SYSTEM
(IPS) (Body 2)



PINHOLE OCCULTER
FACILITY (Body 3)



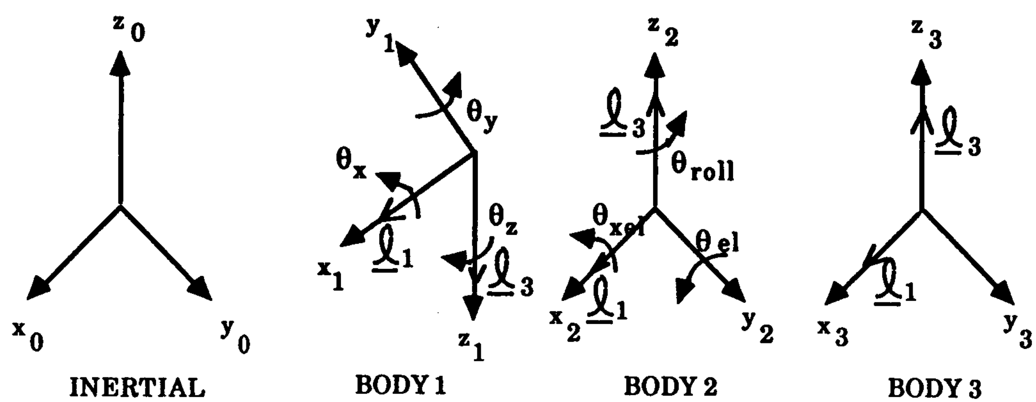


SPACE STATION/POF TREETOPS BODY INPUT DATA

BODY#:		1	2	3
TYPE:		Flex (94 modes)	Rigid	Flex (8 modes)
MASS(kg):		$.3071 \times 10^{6}$	1875.	69.81
MOMENTS OF INERTIA($kg\text{-}m^2$):	I_{xx}	$.29885 \times 10^{9}$	12691.	65391.
	I_{yy}	$.232 \times 10^{9}$	12354.	65416.
	I_{zz}	$.94365 \times 10^{9}$	1903.	81.
PRODUCTS OF INERTIA($kg\text{-}m^2$):	I_{xy}	$-.4497 \times 10^{6}$	0.	0.
	I_{xz}	$.71475 \times 10^{6}$	0.	0.
	I_{yz}	$.94365 \times 10^{7}$	0.	0.
ATTACH POINT(m):		(-.4444,.40549,-5.6515)	(0.,0.,0.)	(0.,0.,0.)
MASS CENTER(m):		(-.4444,.40549,-5.6515)	(0.,0.,.5761)	(0.,0.,29.92)
HINGE POINT(m):		(0,-17.831,-38.405)	(0.,0.,4.3198)	
SENSOR/ACTUATOR MOUNTING POINTS(m):				
CREW DISTURBANCE		(-21.275,-3.556,4.1148)		
PITCH JETS(4)		(0,$\pm$17.831,$\pm$21.946)		
IPS RATE GYROS			(0.,0.,1.)	
POF LOS SENSOR				(0.,0.,32.)

SPACE STATION/POF TREETOPS HINGE INPUT DATA

HINGE ID#	1	2	3
INBOARD BODY ID#	0	1	2
OUTBOARD BODY ID#	1	2	3
ROTATION AXIS:	3	3	0
$\underline{\ell}_1$ WRT INBOARD BODY	(1,0,0)	(1,0,0)	(1,0,0)
$\underline{\ell}_1$ WRT OUTBOARD BODY	(1,0,0)	(1,0,0)	(1,0,0)
$\underline{\ell}_3$ WRT INBOARD BODY	(0,0,1)	(0,0,-1)	(0,0,1)
$\underline{\ell}_3$ WRT OUTBOARD BODY	(0,0,-1)	(0,0,1)	(0,0,1)
INITIAL ROTATION ANGLE(deg):	0.	0.	0.
TRANSLATIONAL DOF:	3	0	0



SPACE STATION/POF TREETOPS SENSOR/ACTUATOR INPUT DATA

		MOUNTING POINT		
SENSOR ID#	SENSOR	BODY ID#	NODE ID#	INPUT AXIS
3	IPS RATE GYRO(cross elevation axis)	2	3	(1,0,0)
4	IPS RATE GYRO(elevation axis)	2	3	(0,1,0)
5	IPS RATE GYRO(roll axis)	2	3	(0,0,1)
99	POF LINE OF SIGHT	3	2	(0,0,1)

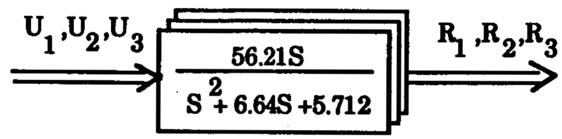
		MOUNTING POINT				
ACTUATOR ID#	ACTUATOR	HINGE ID#	HINGE AXIS ID#	BODY ID#	NODE ID#	OUTPUT AXIS
1	IPS TORQUE MOTOR(roll)	2	3			
2	IPS TORQUE MOTOR(elevation)	2	2			
3	IPS TORQUE MOTOR(cross elevation)	2	1			
4	PITCH JET #1			1	4	(1,0,0)
5	PITCH JET #2			1	5	(1,0,0)
6	PITCH JET #3			1	6	(1,0,0)
11	PITCH JET #4			1	7	(1,0,0)
12	CREW MOTION FORCE(x)			1	2	(1,0,0)
13	CREW MOTION FORCE(y)			1	2	(0,1,0)
14	CREW MOTION FORCE(z)			1	2	(0,0,1)

SPACE STATION/POF TREETOPS CONTROLLER FORM AND FUNCTION

- CONTINUOUS CONTROLLER
  - **Crew Motion Shaping Filter**
- USER CONTROLLER
  - **IPS 3 Axis Gimbal Pointing Sampled Data Control Law**

SPACE STATION/POF TREETOPS CONTROLLER DATA DEFINITION

CONTINUOUS CONTROLLER



OUTPUT DATA

OUTPUT#	SOURCE TYPE	SOURCE ID#	GAIN
1	T	1	10.954
2	T	2	10.954
3	T	3	10.954

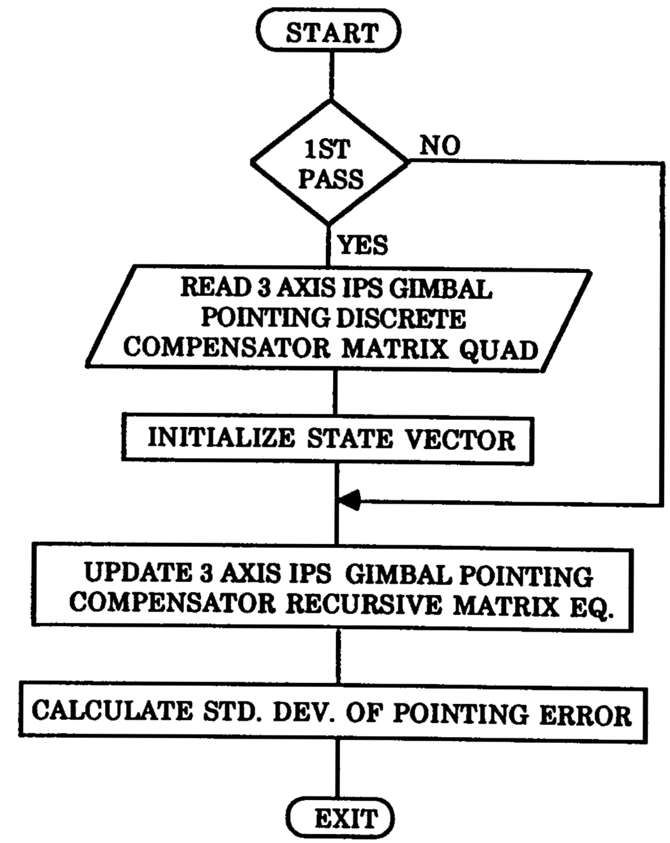
TRANSFER FUNCTION DATA

T#	SOURCE TYPE	SOURCE ID#	NUMERATOR ORDER	NUMERATOR COEFFICIENTS	DENOMINATOR ORDER	DENOMINATOR COEEFICIENTS
1	I	1	1	0.,56.21	2	5.712,6.64,1.
2	I	2	1	0.,56.21	2	5.712,6.64,1.
3	I	3	1	0.,56.21	2	5.712,6.64,1.

T = TRANSFER FUNCTION

I = CONTROLLER INPUT

USER CONTROLLER



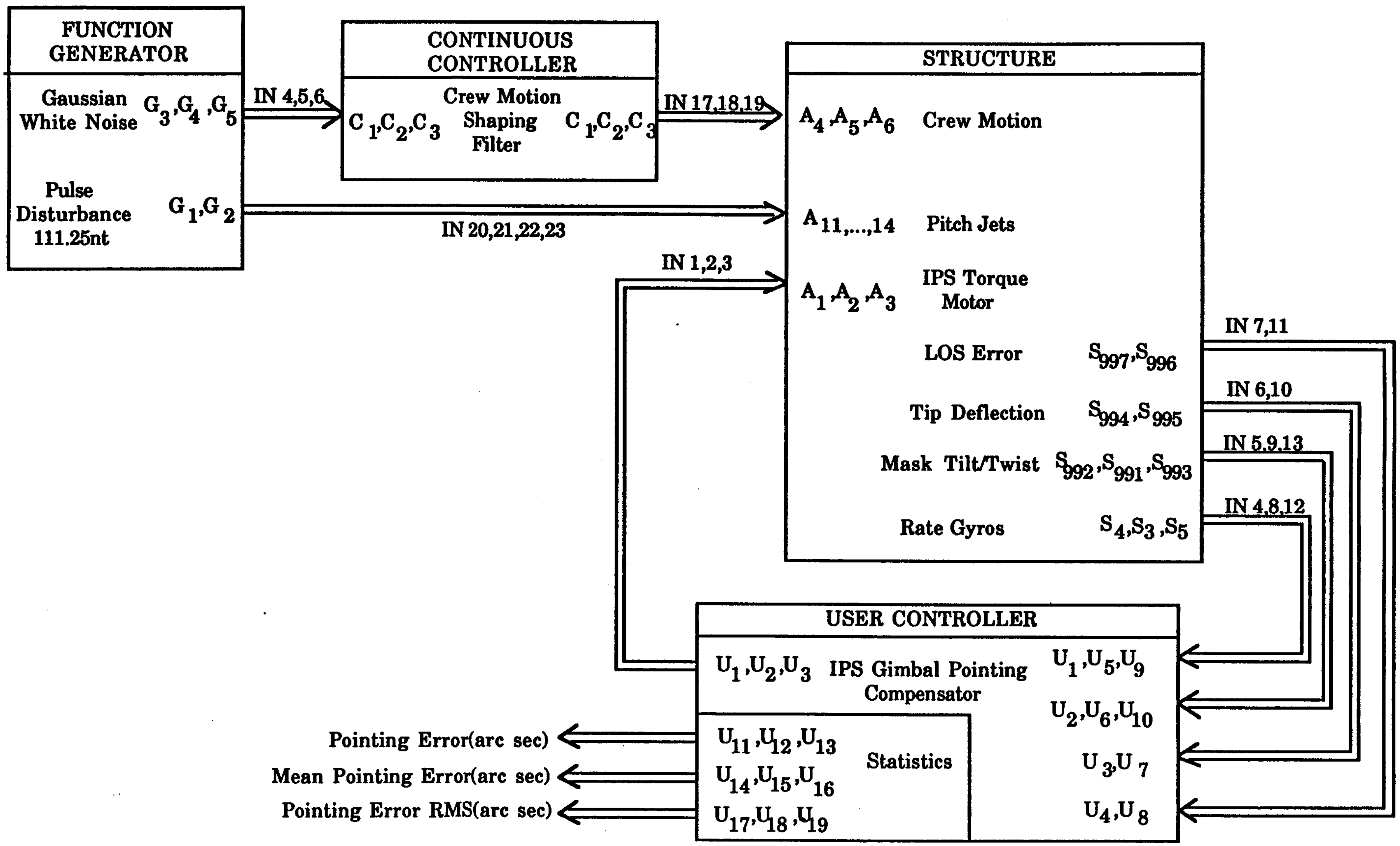
INPUT DATA

CONTROLLER TYPE = USER
CONTROLLER UPDATE INTERVAL = .025sec
NUMBER OF INPUTS = 13
NUMBER OF OUTPUTS = 20

SPACE STATION/POF TREETOPS FUNCTION GENERATOR INPUT DATA

FUNCTION GENERATOR ID#	TYPE	START TIME(sec)	STOP TIME(sec)	MAGNITUDE	DESCRIPTION
1	PULSE	0.	1.	111.25 n	PITCH JET DISTURBANCE
2	PULSE	0.	1.	-111.25 n	
3	USER DEFINED				
4	USER DEFINED				GAUSSIAN WHITE NOISE (μ =0, σ =1)
5	USER DEFINED				

SPACE STATION/POF TREETOPS INTERCONNECT DEFINITION



SPACE STATION/POF COMPENSATOR DISCUSSION

- Compensator used was not tailored to the Space Station/POF problem.

- The quick look at POF performance with the Space Station as an alternate mounting base was achieved by utilizing the compensator designed for the Orbiter/POF configuration.

- The performance obtained was not optimal because the Orbiter and Space Station dynamics are different.

PERFORMANCE

	Elevation	Cross Elevation
$\delta_{\ell-rms}$	1.0 arc-sec-rms	0.82 arc-sec-rms
δ_{ℓ}	1.8 arc-sec	0.9 arc-sec

δ : Line Of Sight Error h: High Level Crew Motion ℓ: Low Level Crew Motion

APPENDIX

- TREETOPS Time Histories
- TREETOPS Files
- Crew Motion Supplemental Data
- HONEY-X Design and Analysis Command Files

TREETOPS PERFORMANCE ANALYSIS - TIME HISTORIES

ORBITER/POF

- 3 Axis Orbiter VRCS Attitude Maneuver
  - Orbiter attitude rates
  - Orbiter attitude angles
  - POF elevation axis pointing error and torque motor output(old vs. new)
  - POF cross elevation axis pointing error and torque motor output(old vs. new)
  - POF roll axis pointing error and torque motor output
  - 3 axis pointing error standard deviation

- Low Level Crew Disturbance
  - Crew motion force(x)
  - POF elevation axis pointing error and torque motor output
  - POF cross elevation axis pointing error and torque motor output
  - POF roll axis pointing error and torque motor output
  - 3 axis pointing error standard deviation

- High Level Crew Disturbance
  - Crew motion force(x)
  - POF elevation axis pointing error and torque motor output
  - POF cross elevation axis pointing error and torque motor output
  - POF roll axis pointing error and torque motor output
  - 3 axis pointing error standard deviation

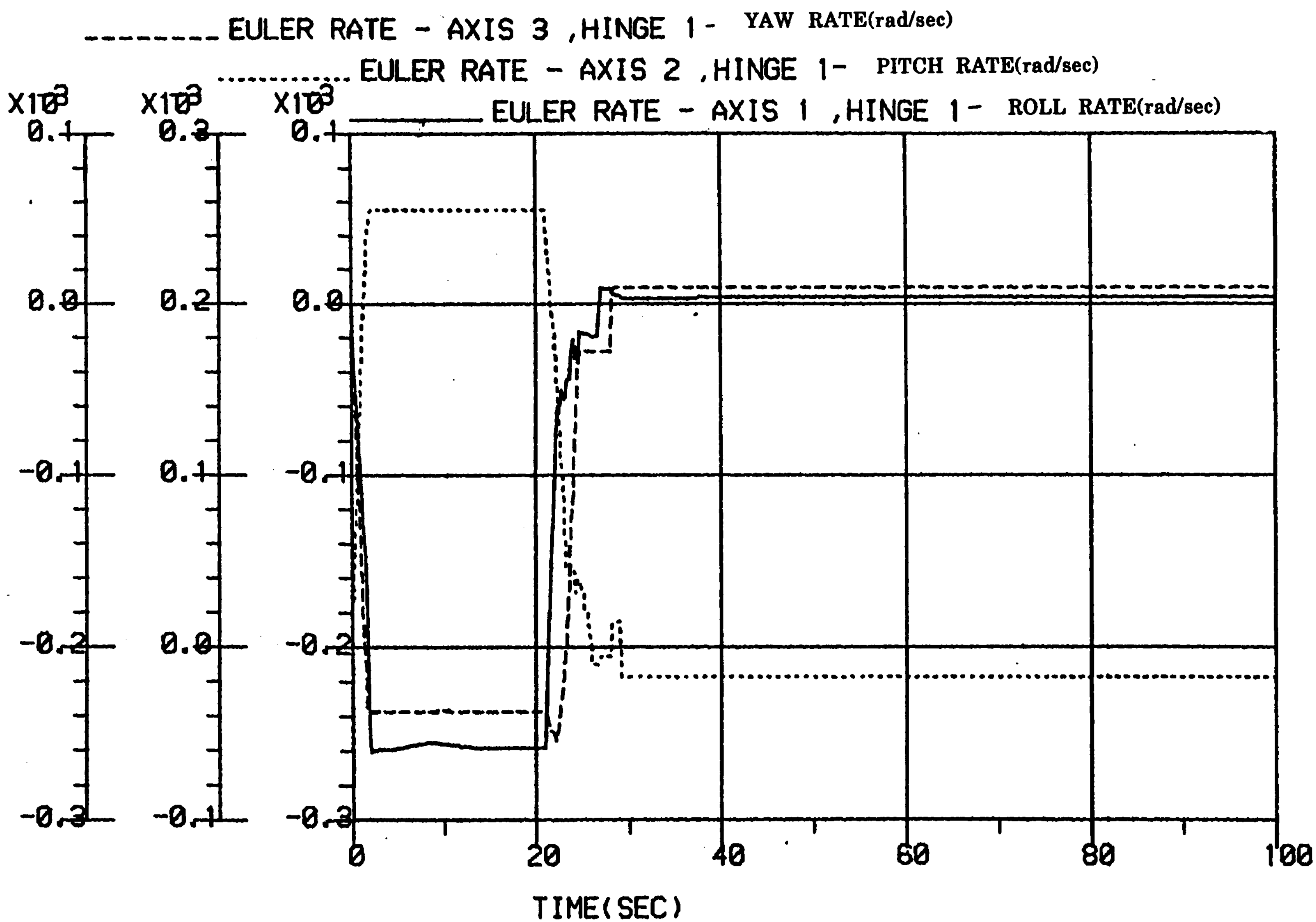
SPACE STATION/POF

- Pitch Jet Maneuver
  - Space station attitude rates
  - Space station attitude
  - POF elevation axis pointing error and torque motor output
  - POF cross elevation axis pointing error and torque motor output

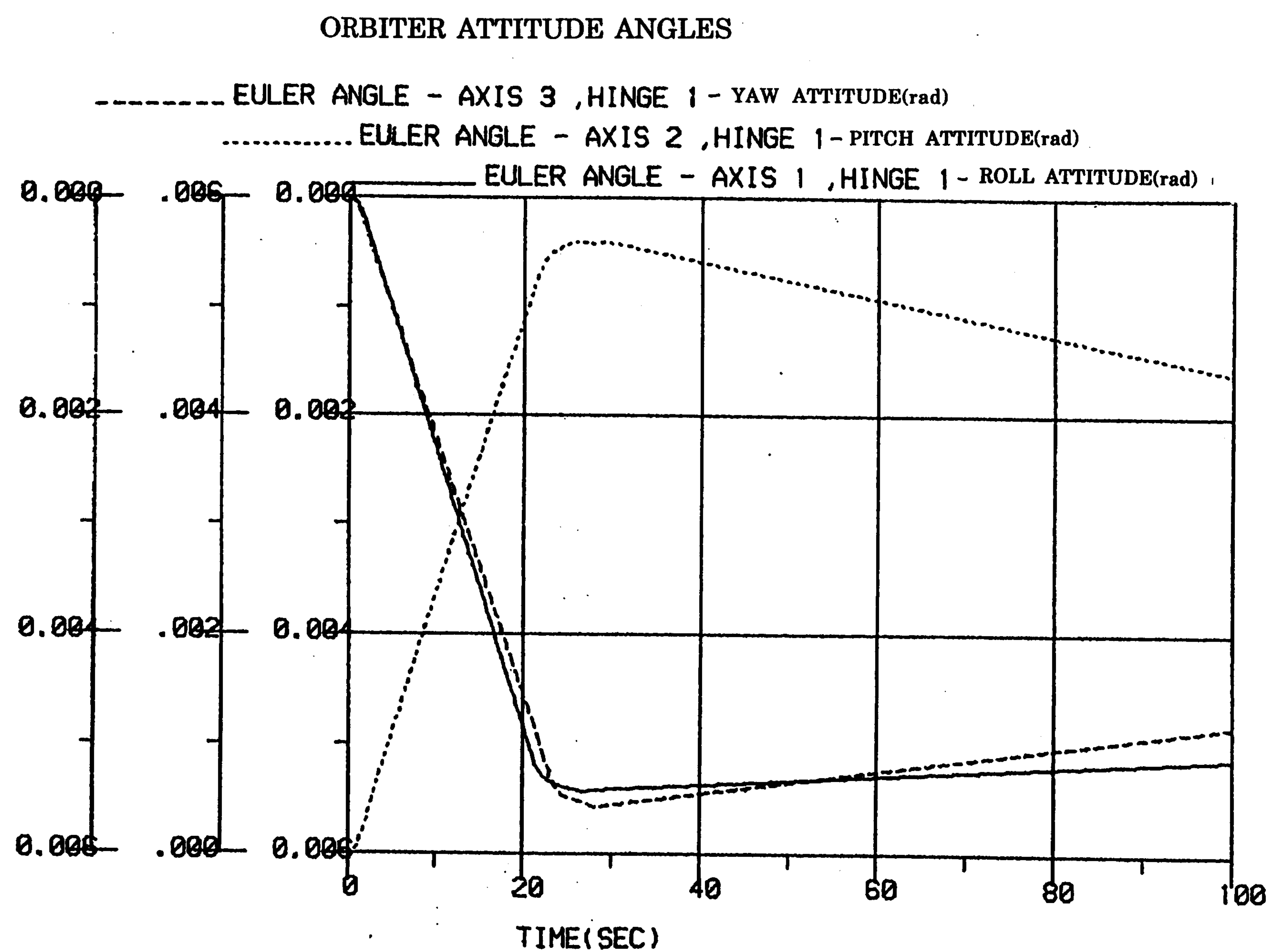
- Low Level Crew Disturbance
  - Crew motion force
  - POF elevation axis pointing error and torque motor output
  - POF cross elevation axis pointing error and torque motor output
  - Pointing error standard deviation

3 AXIS ORBITER VRCS ATTITUDE MANEUVER($\theta cmd_x = -.2°, \theta cmd_y = +.2°, \theta cmd_z = +.2°$)

ORBITER ATTITUDE RATES



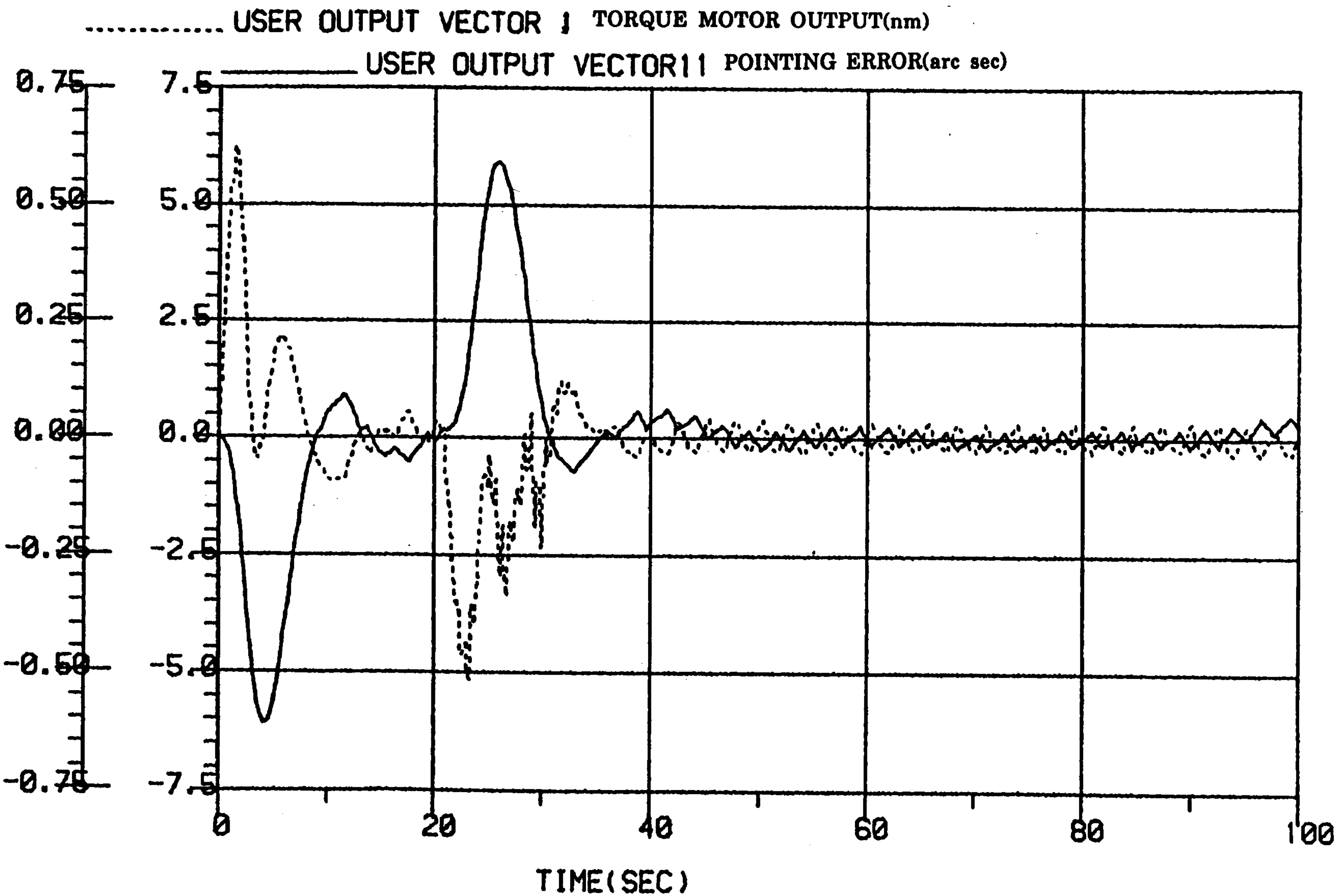
3 AXIS ORBITER VRCS ATTITUDE MANEUVER($\theta cmd_x = -.2°$, $\theta cmd_y = +.2°$, $\theta cmd_z = +.2°$)



3 AXIS ORBITER VRCS ATTITUDE MANEUVER($\theta cmd_x = -.2°, \theta cmd_y = +.2°, \theta cmd_z = +.2°$)

POF ELEVATION AXIS

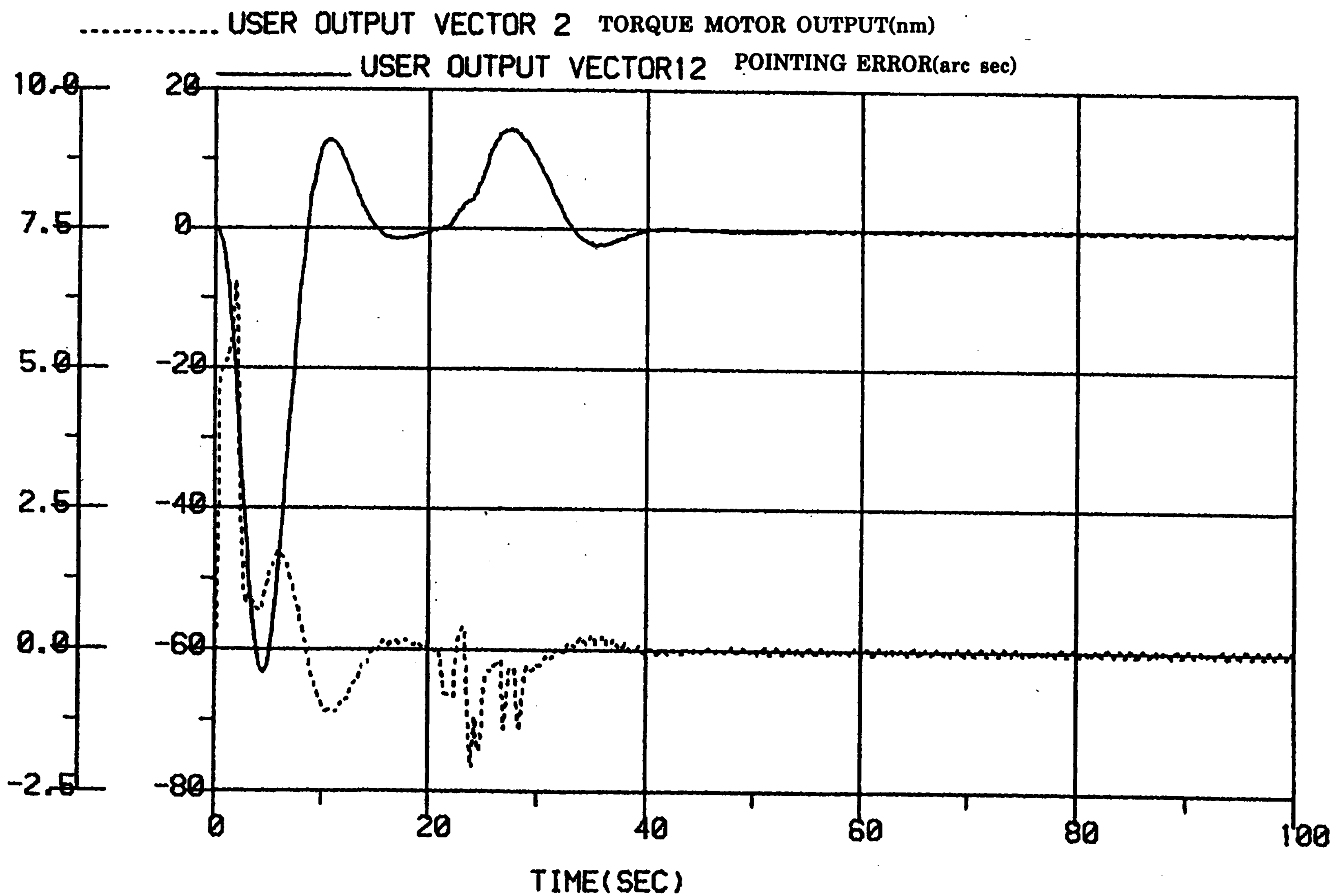
1986 BASELINE



3 AXIS ORBITER VRCS ATTITUDE MANEUVER($\theta cmd_x = -.2^\circ$, $\theta cmd_y = +.2^\circ$, $\theta cmd_z = +.2^\circ$)

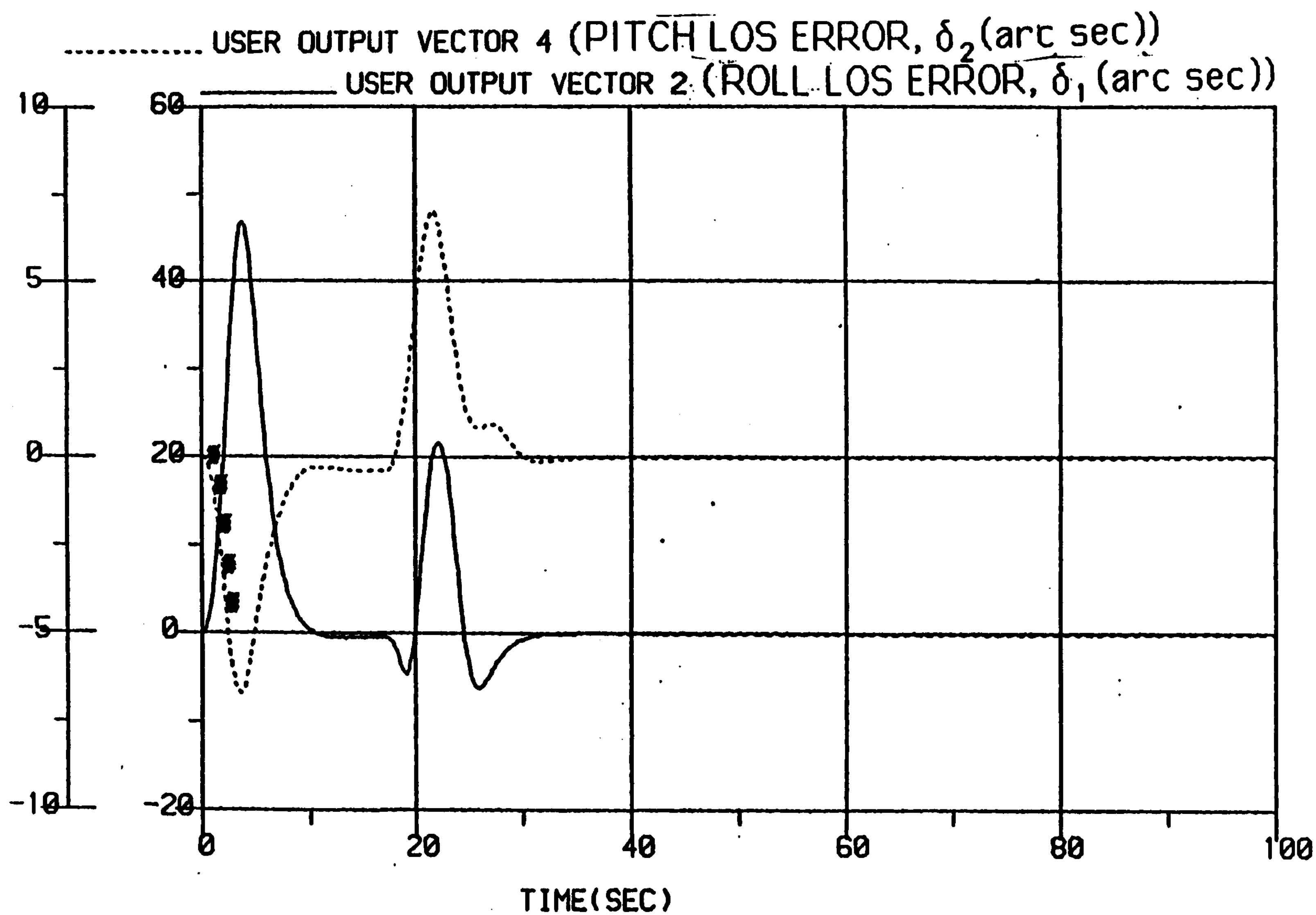
POF CROSS ELEVATION AXIS

1986 BASELINE



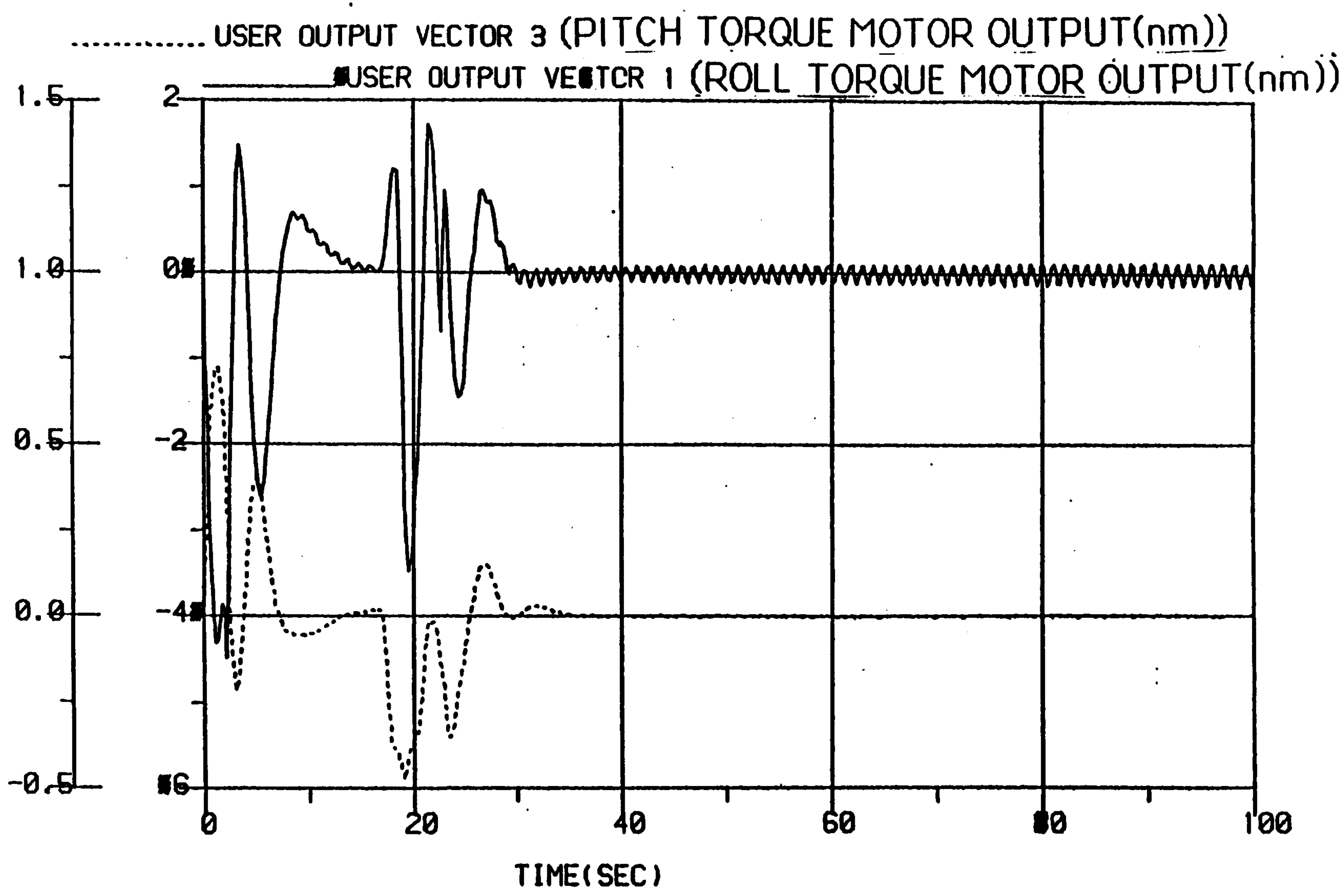
3 AXIS ORBITER VRCS ATTITUDE MANEUVER($\theta cmd_x = -.2^\circ$, $\theta cmd_y = +.2^\circ$, $\theta cmd_z = +.2^\circ$)

1985 BASELINE

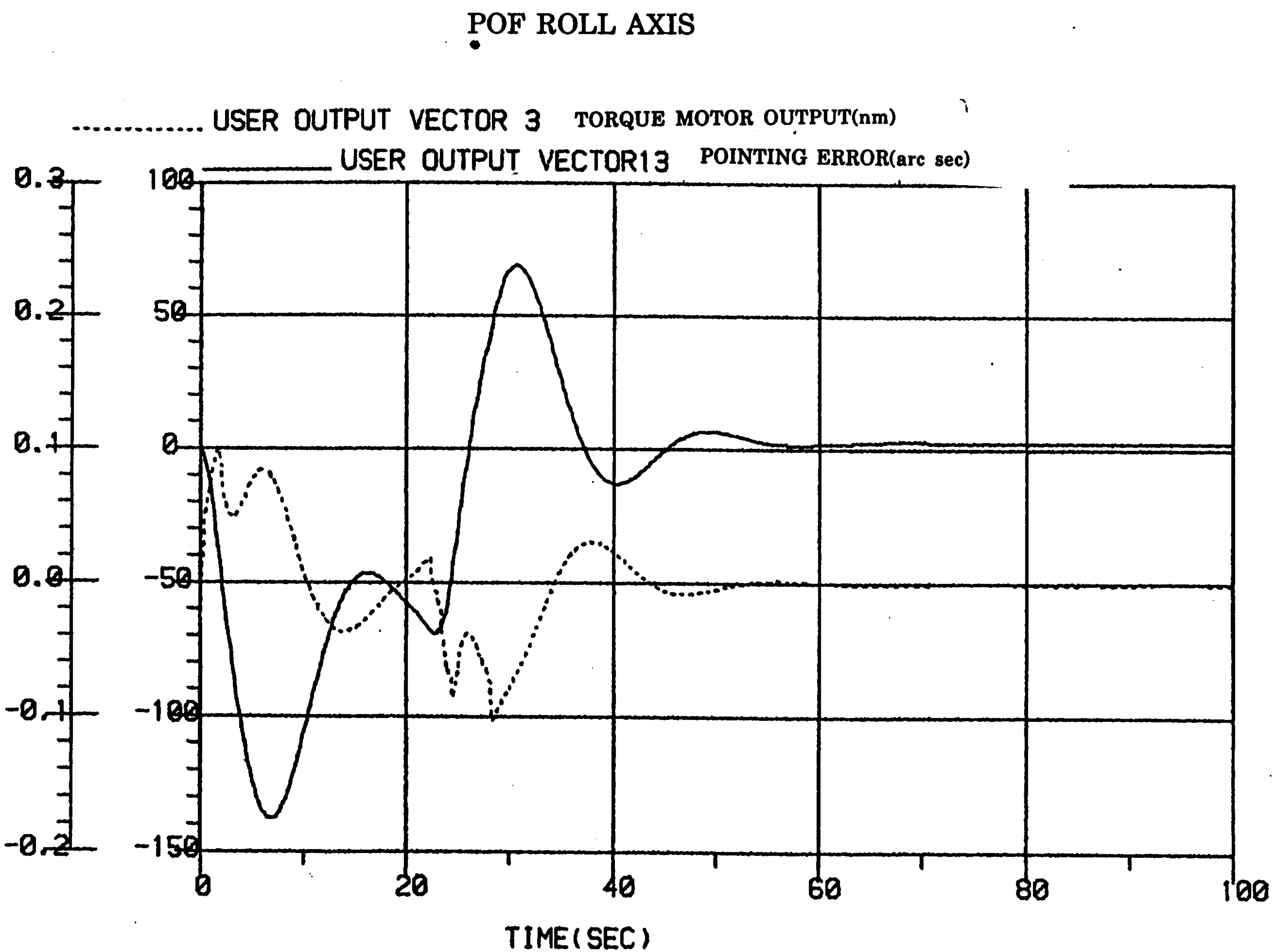


3 AXIS ORBITER VRCS ATTITUDE MANEUVER($\theta cmd_x = -.2^\circ, \theta cmd_y = +.2^\circ, \theta cmd_z = +.2^\circ$)

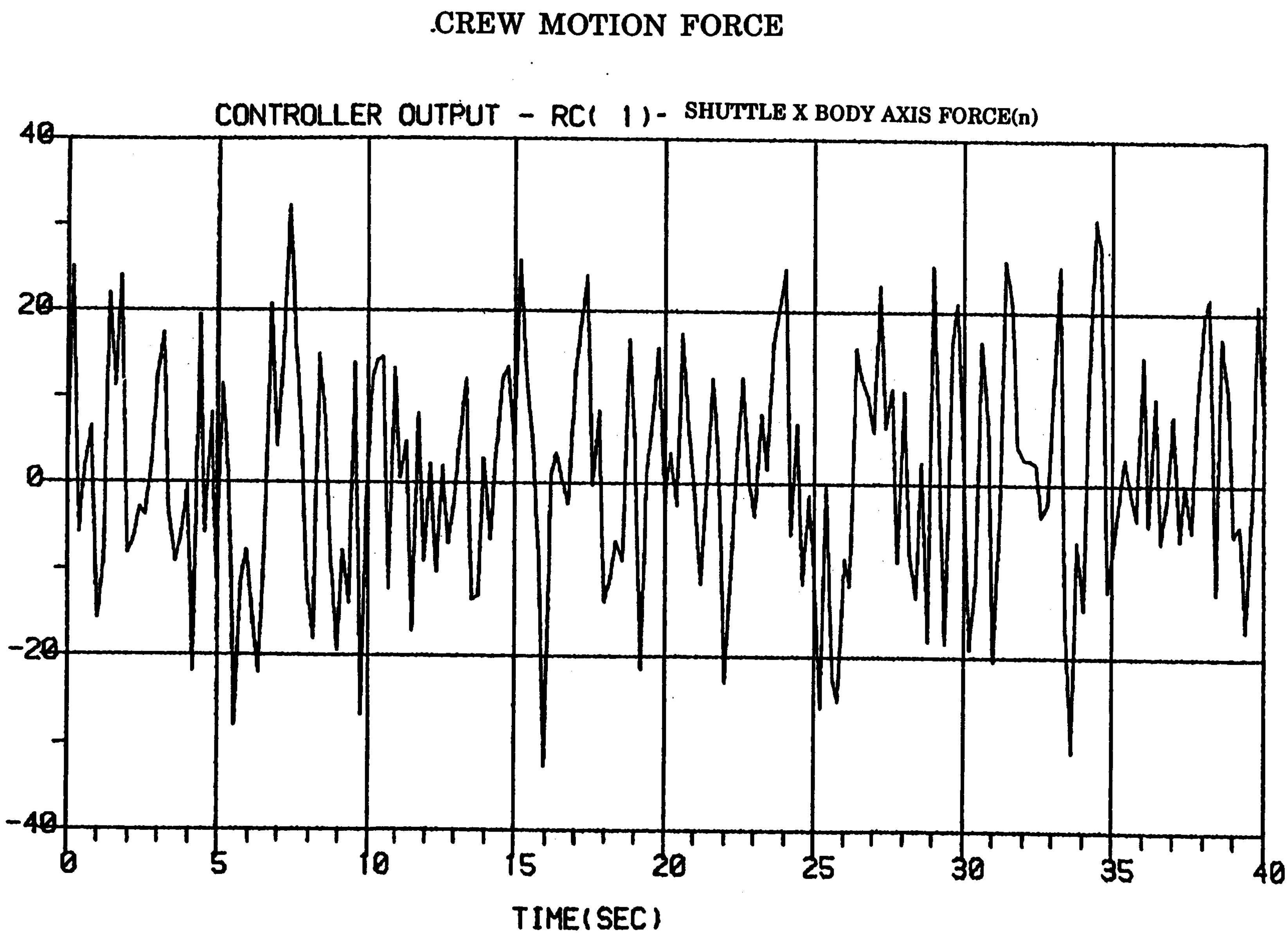
1985 BASELINE



3 AXIS ORBITER VRCS ATTITUDE MANEUVER($\theta cmd_x = -.2^\circ$, $\theta cmd_y = +.2^\circ$, $\theta cmd_z = +.2^\circ$)

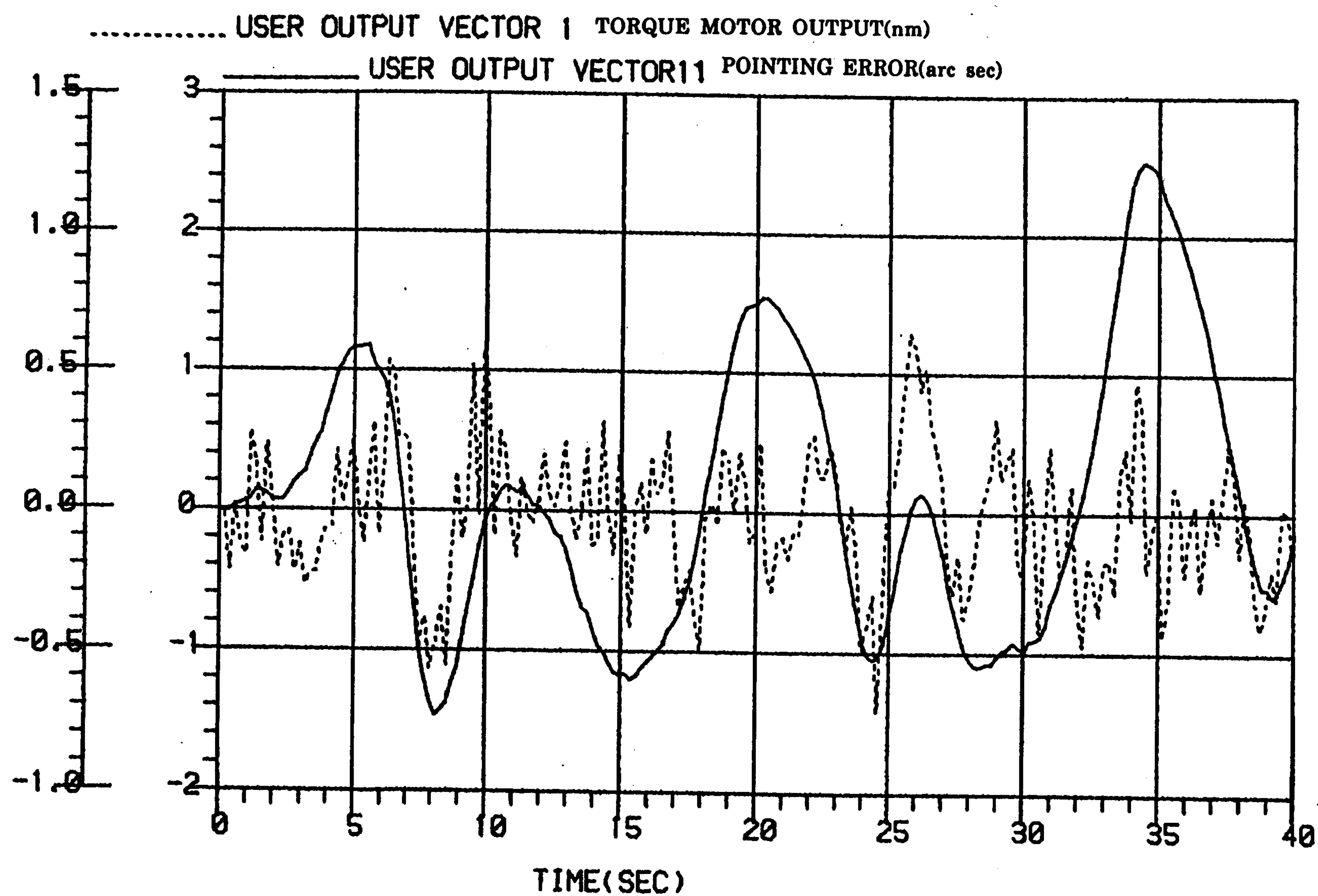


LOW LEVEL CREW MOTION FORCE



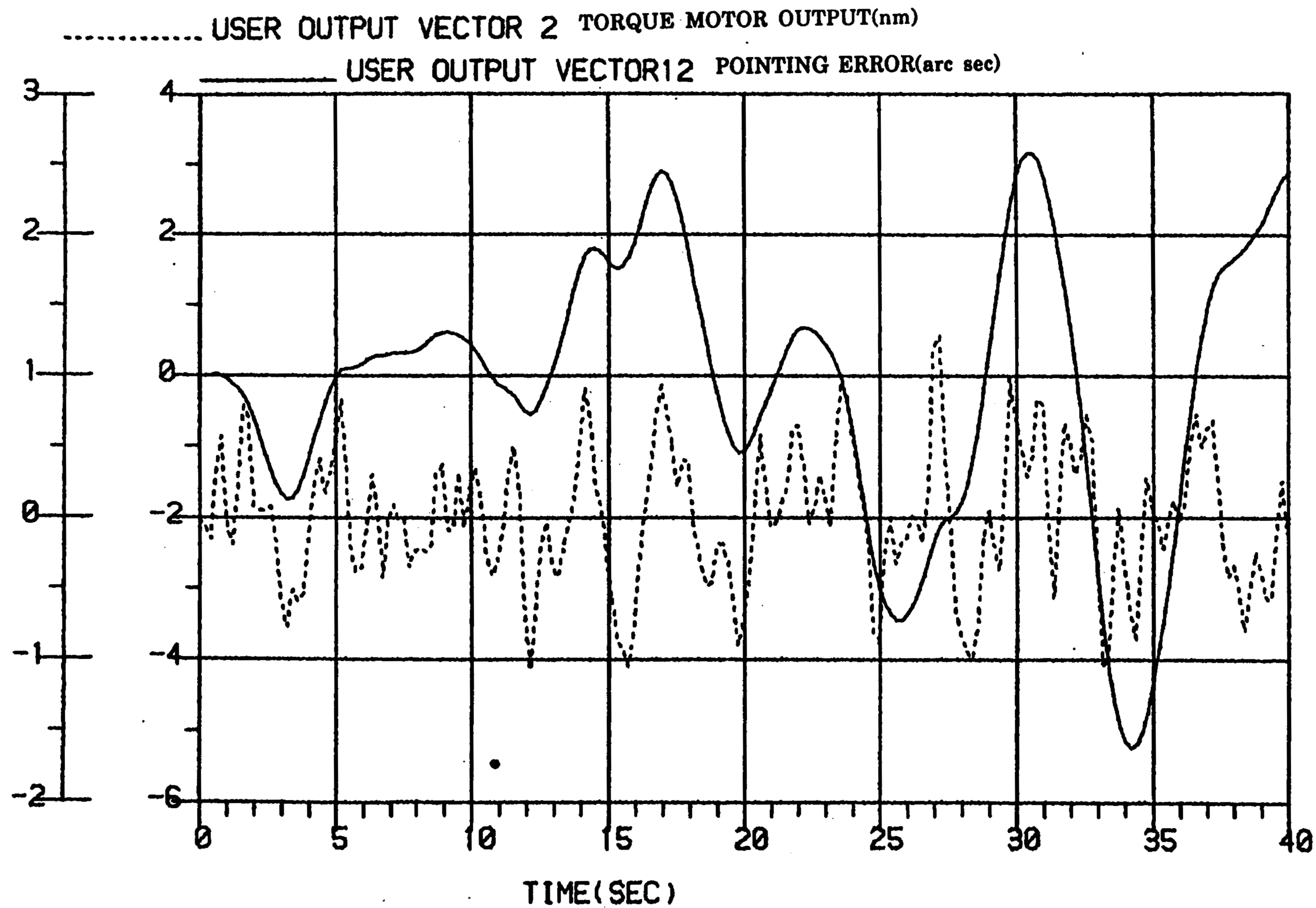
LOW LEVEL CREW MOTION FORCE

POF ELEVATION AXIS



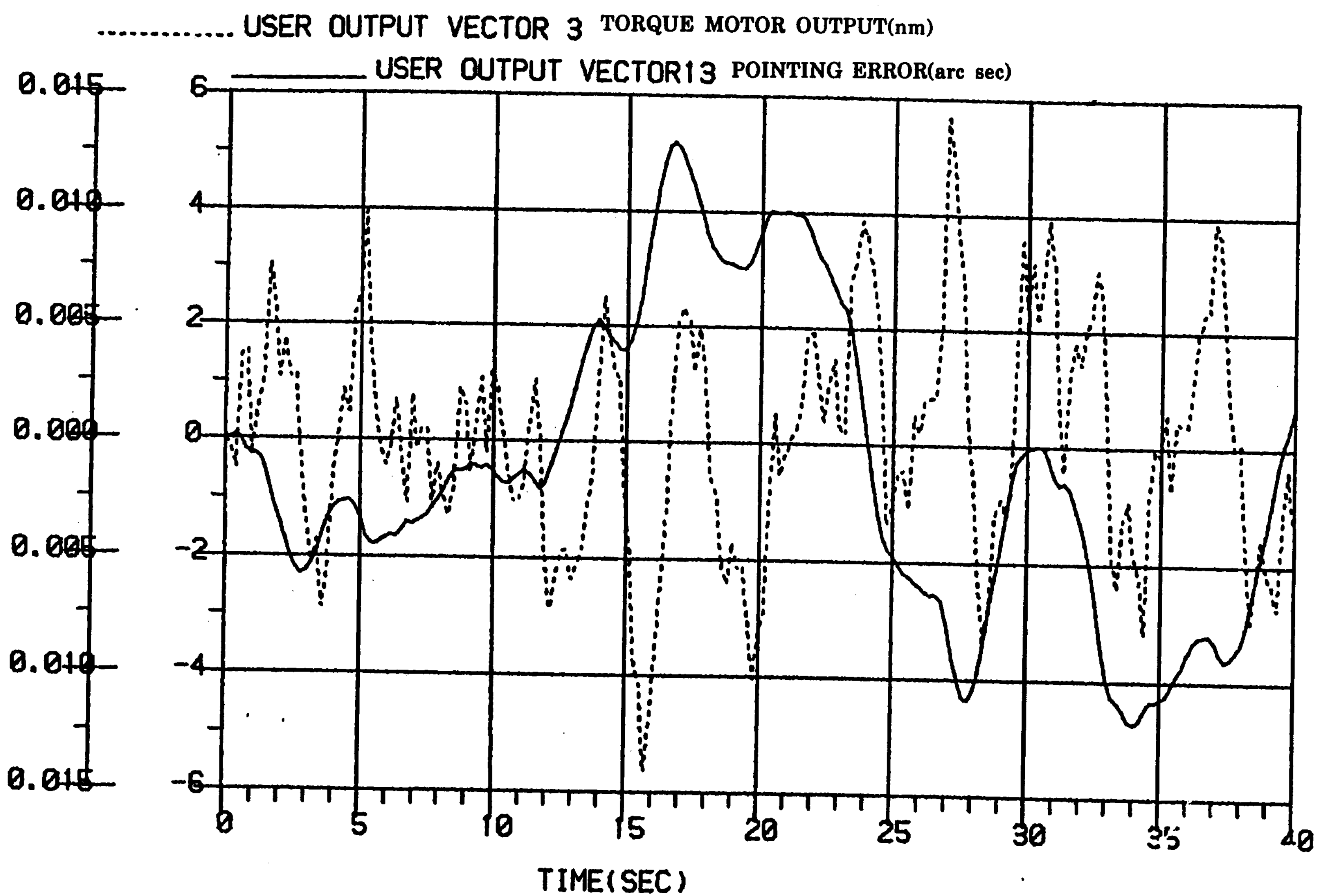
LOW LEVEL CREW MOTION FORCE

POF CROSS ELEVATION AXIS



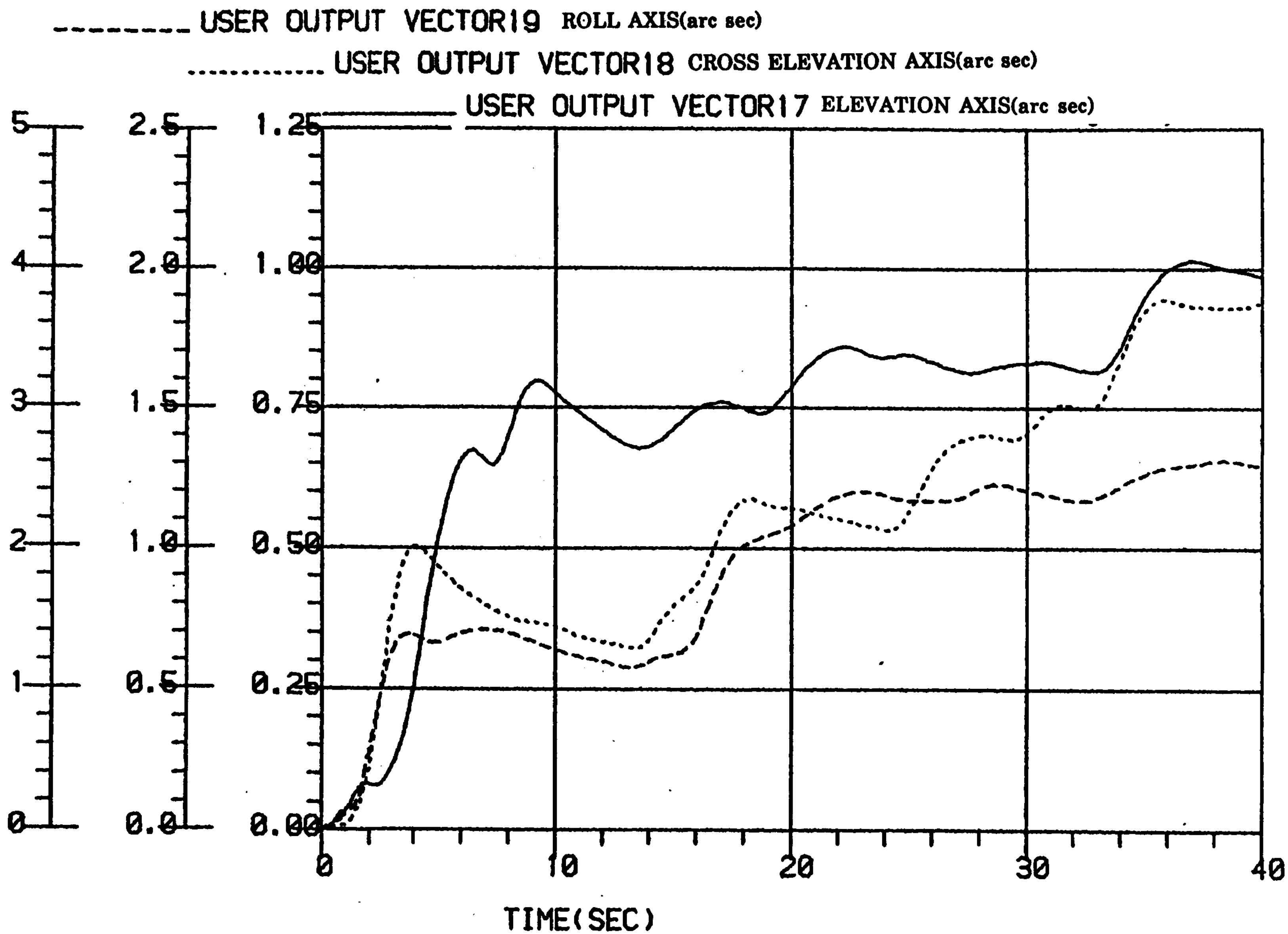
LOW LEVEL CREW MOTION FORCE

POF ROLL AXIS



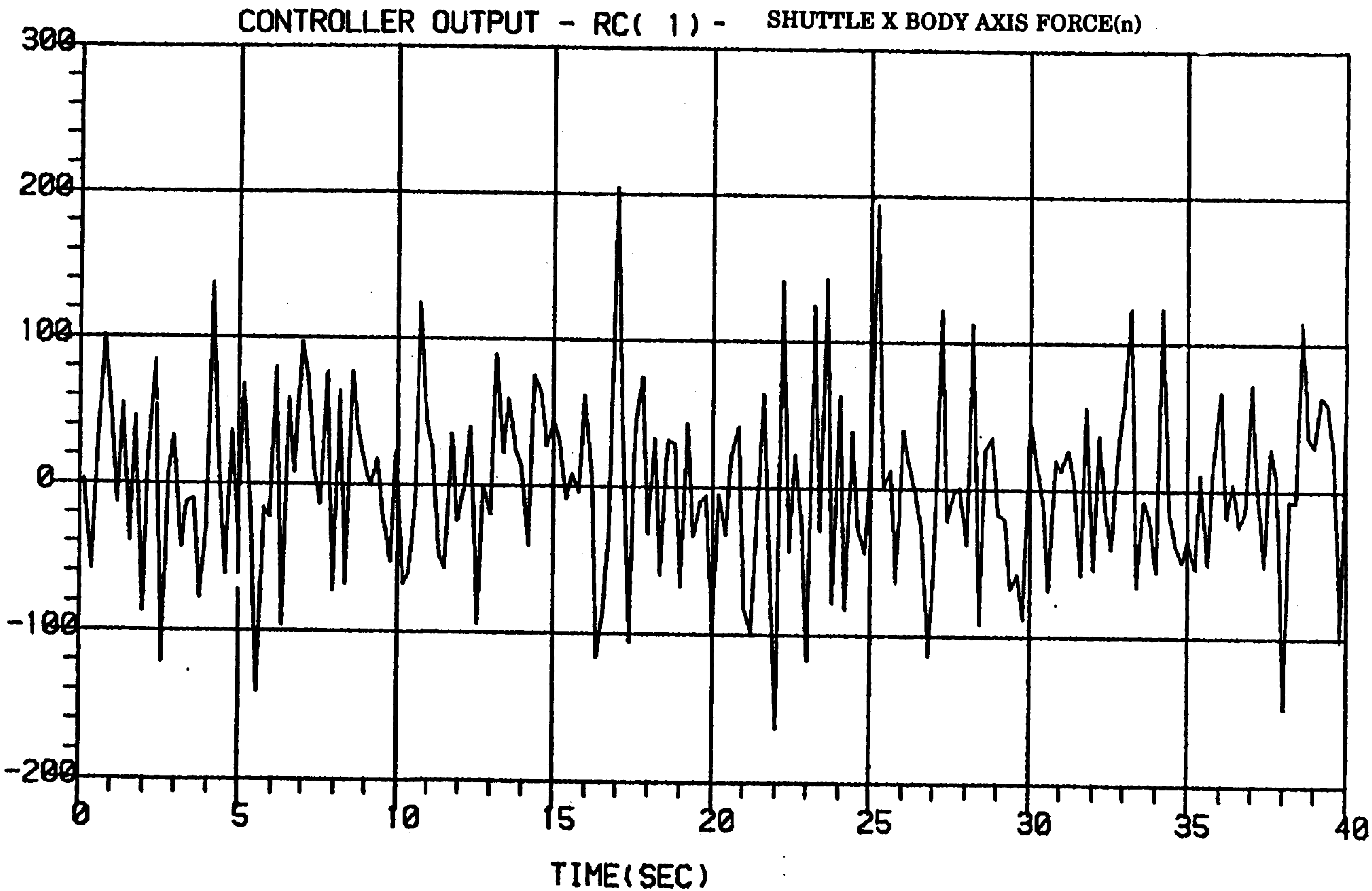
LOW LEVEL CREW MOTION FORCE

POINTING STANDARD DEVIATION



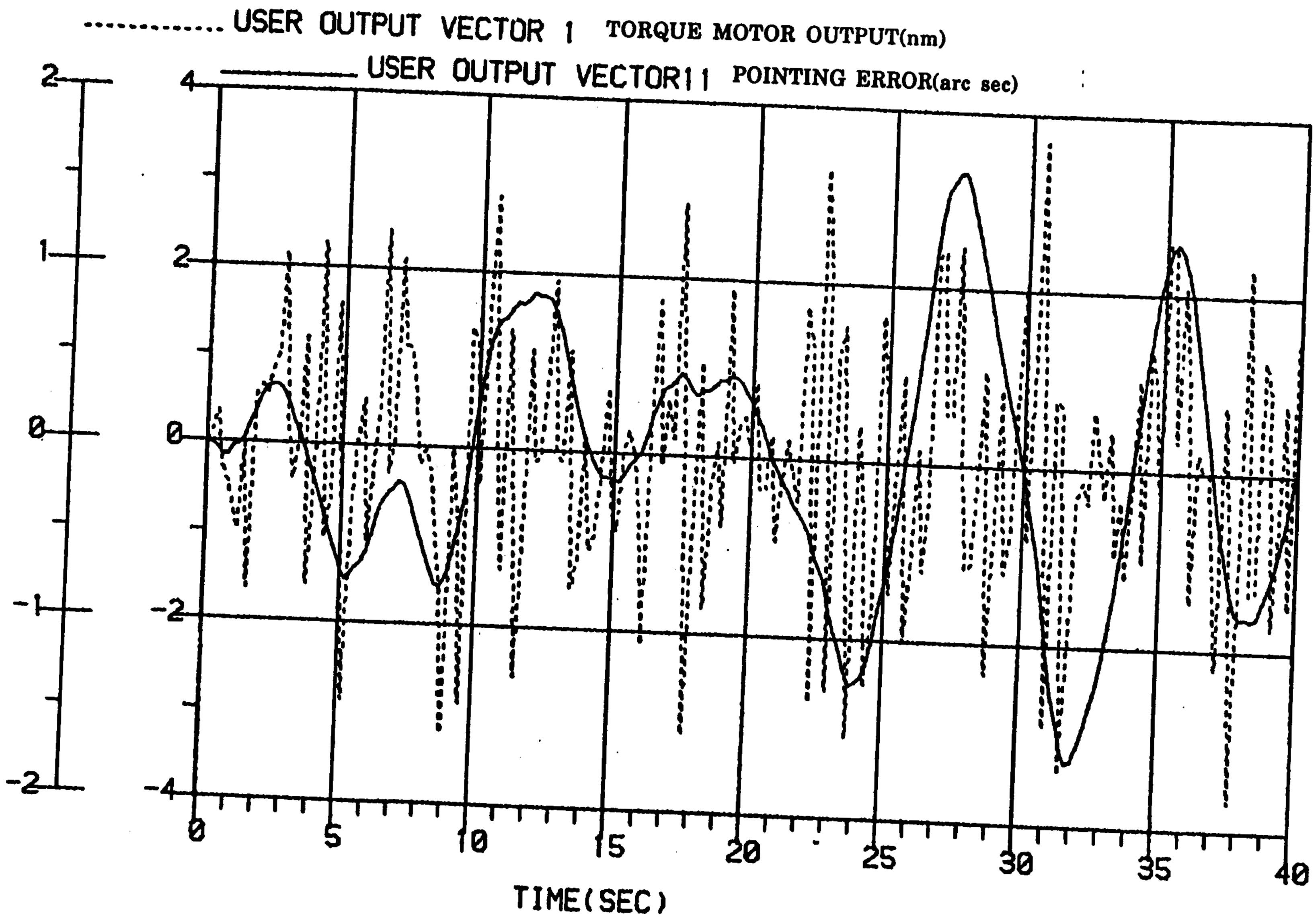
HIGH LEVEL CREW MOTION FORCE

CREW MOTION FORCE



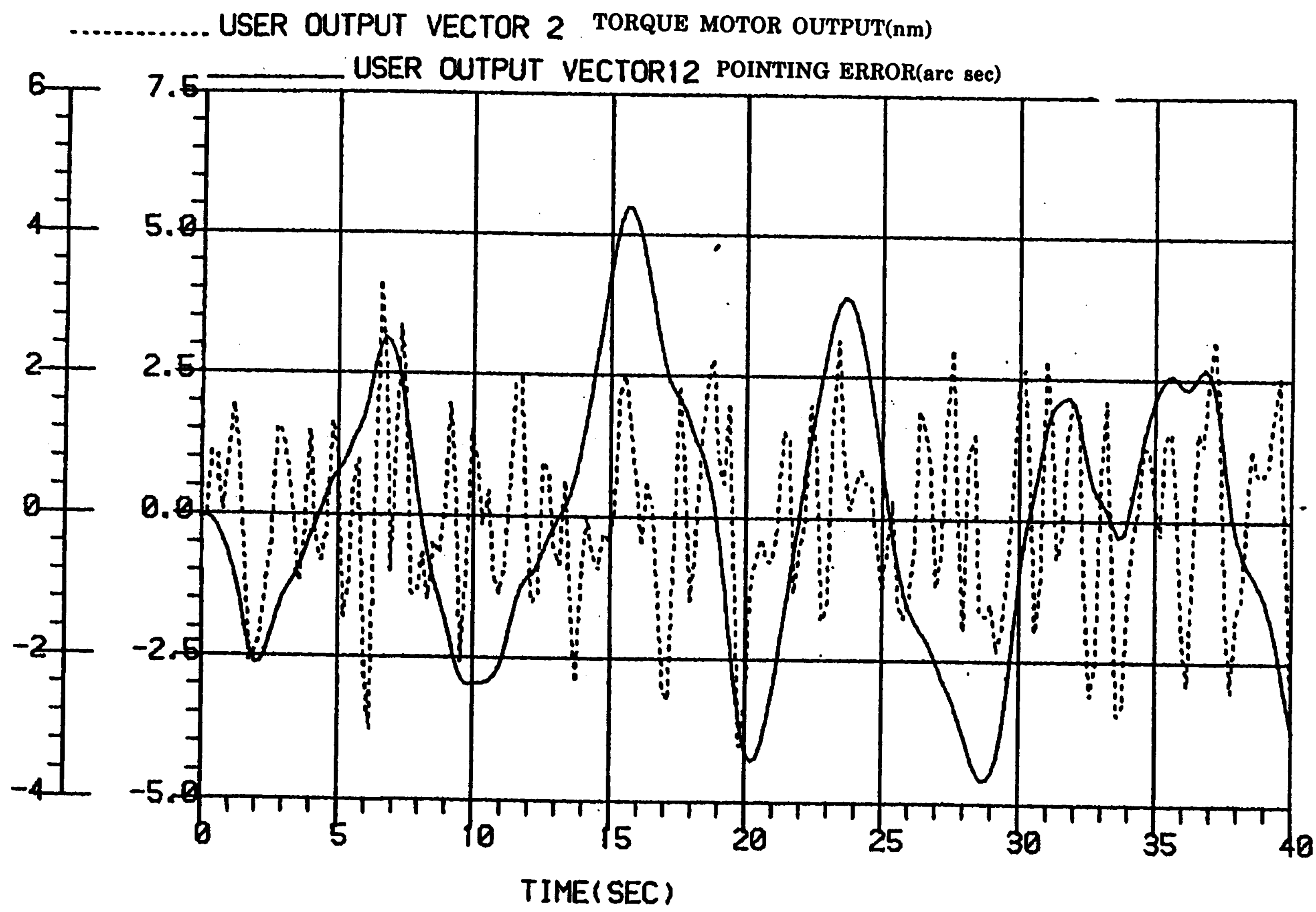
HIGH LEVEL CREW MOTION FORCE

POF ELEVATION AXIS



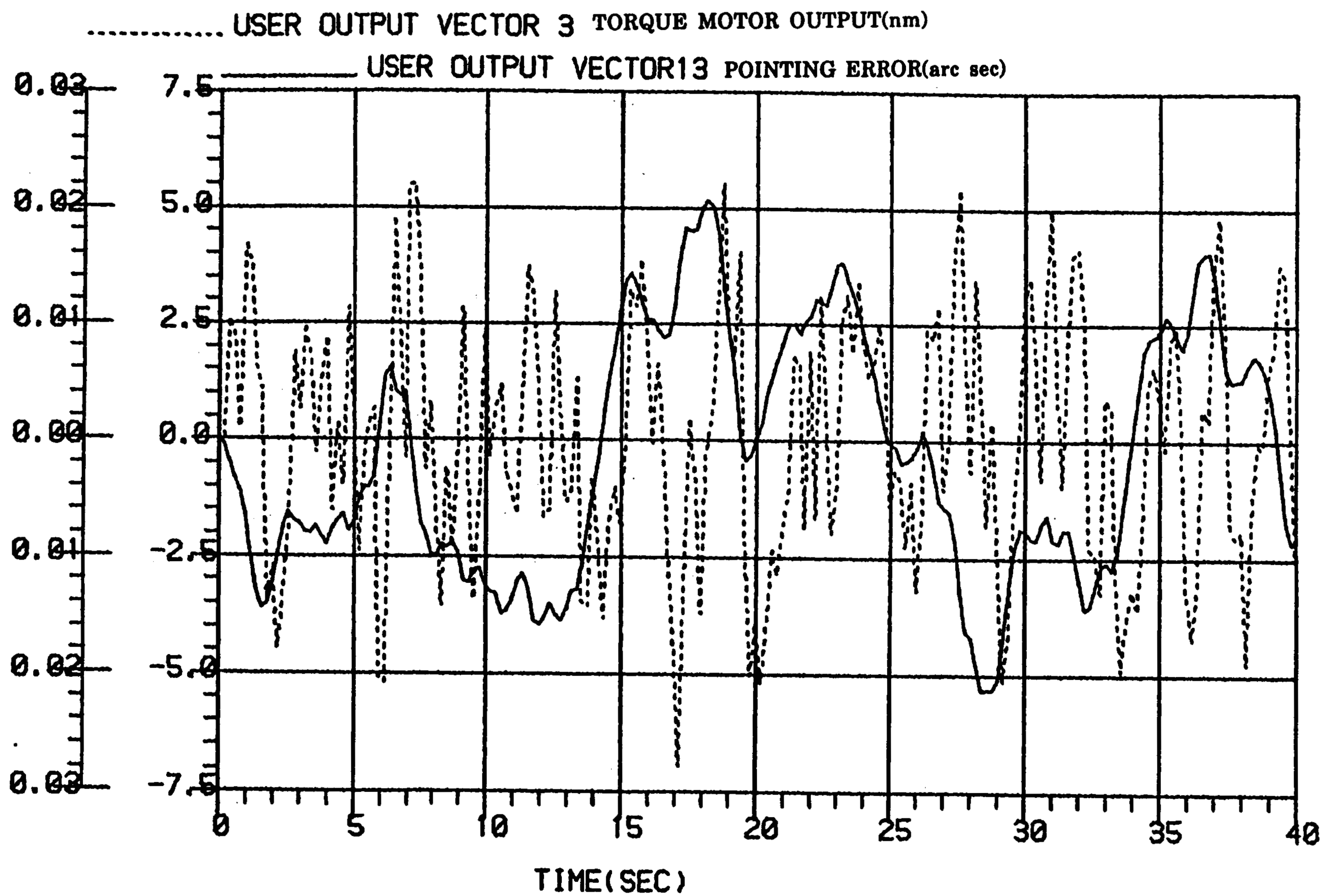
HIGH LEVEL CREW MOTION FORCE

POF CROSS ELEVATION AXIS



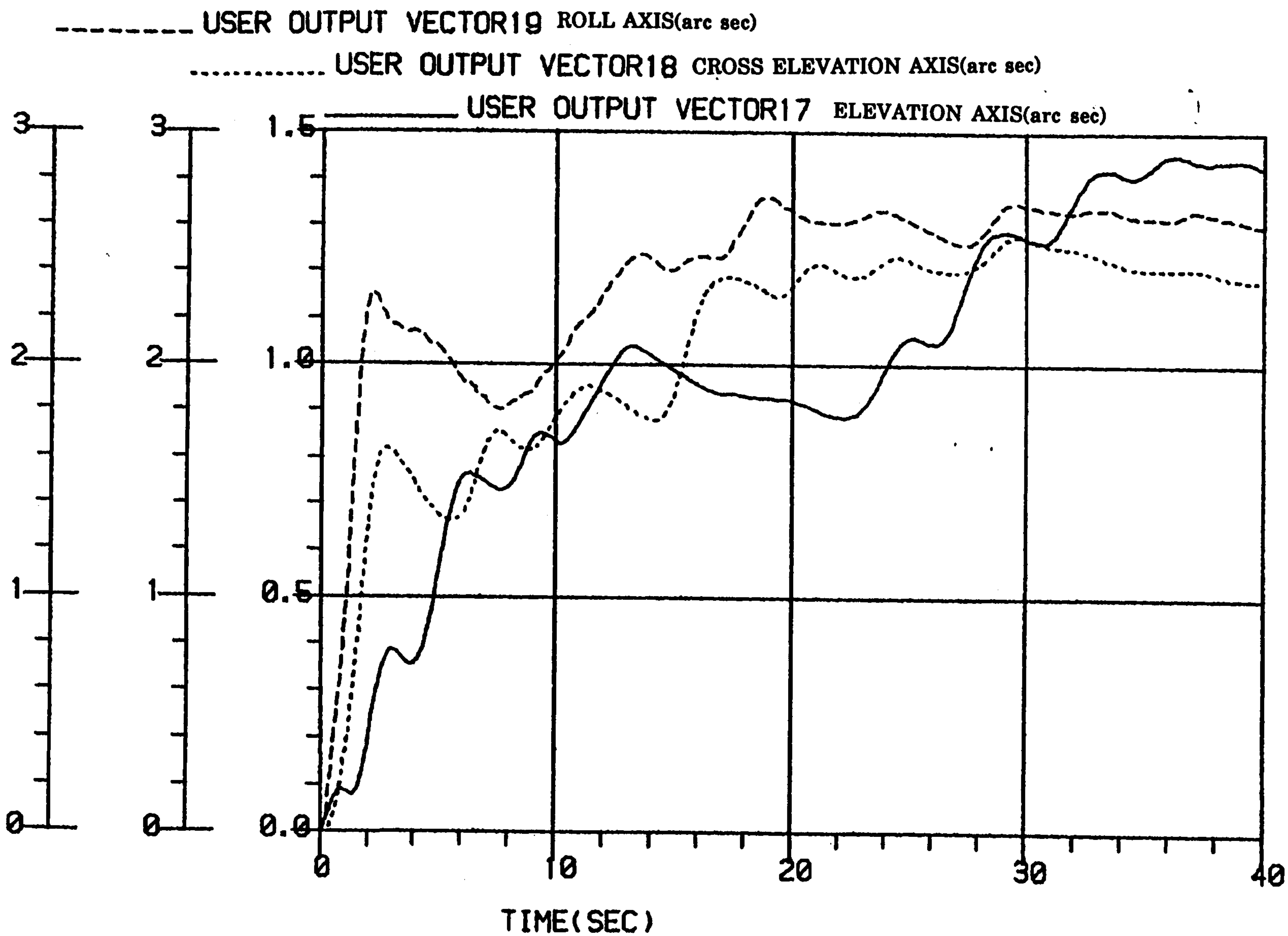
HIGH LEVEL CREW MOTION FORCE

POF ROLL AXIS



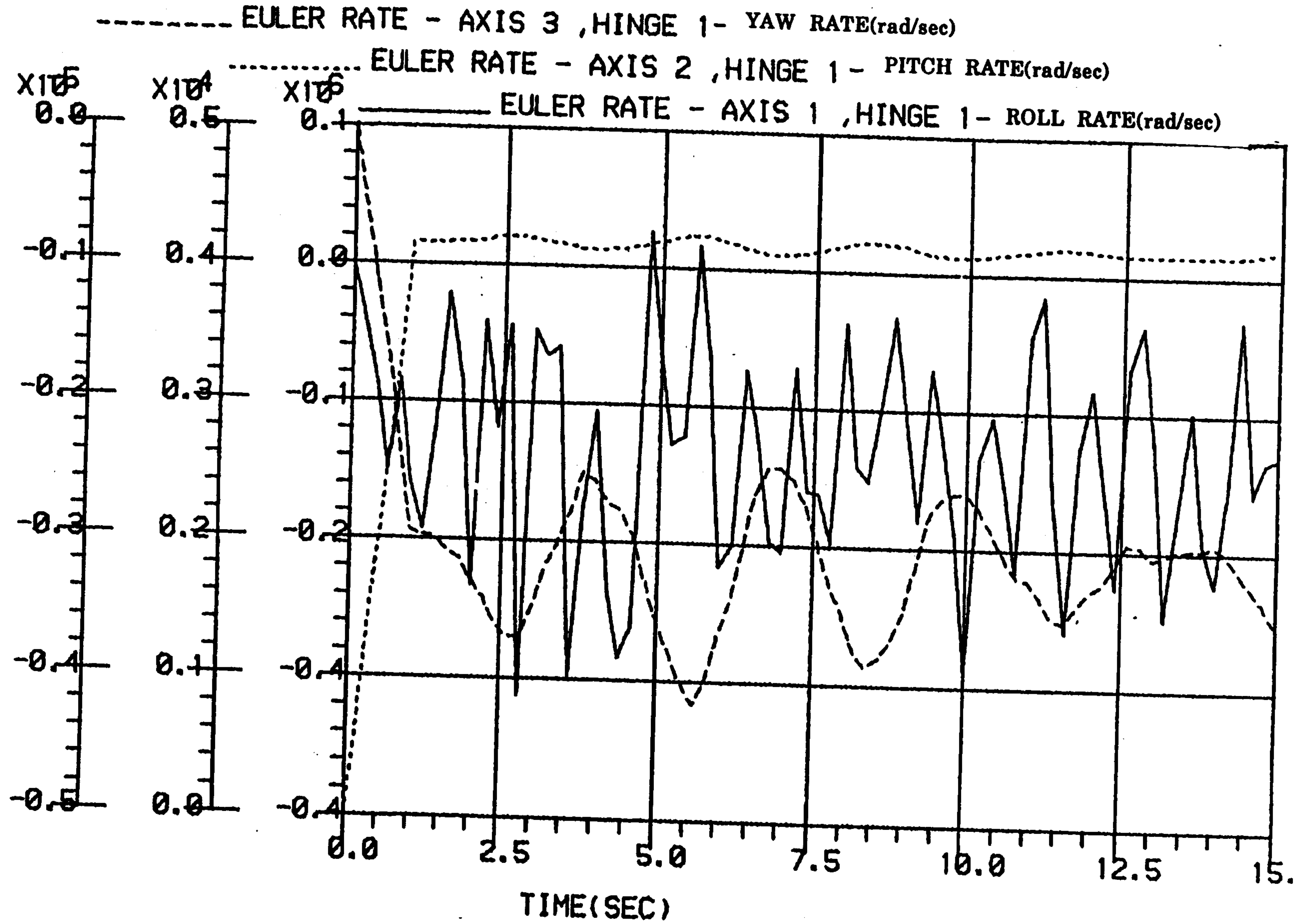
HIGH LEVEL CREW MOTION FORCE

POINTING STANDARD DEVIATION



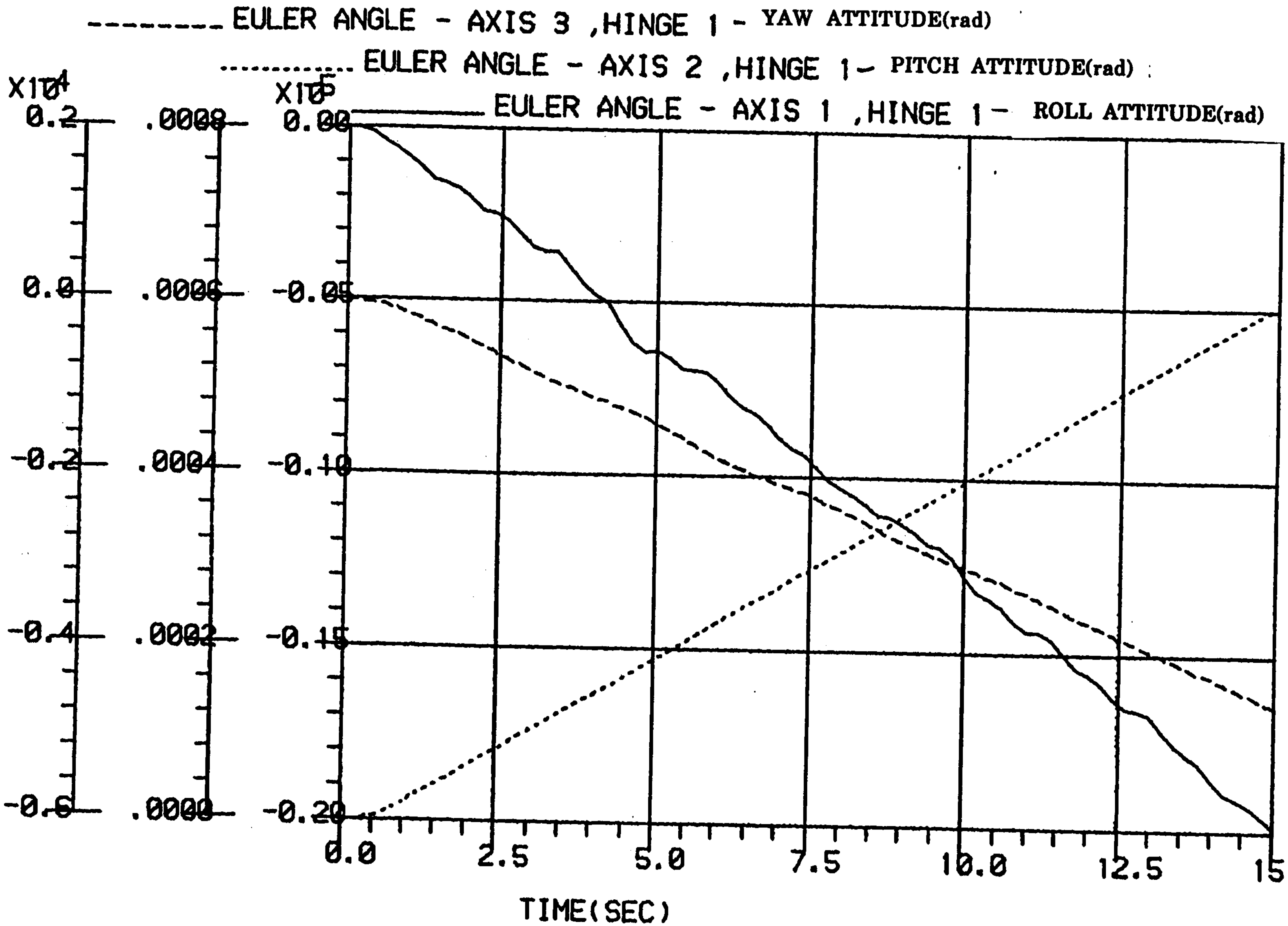
SPACE STATION PITCH JET DISTURBANCE

ATTITUDE RATES



0

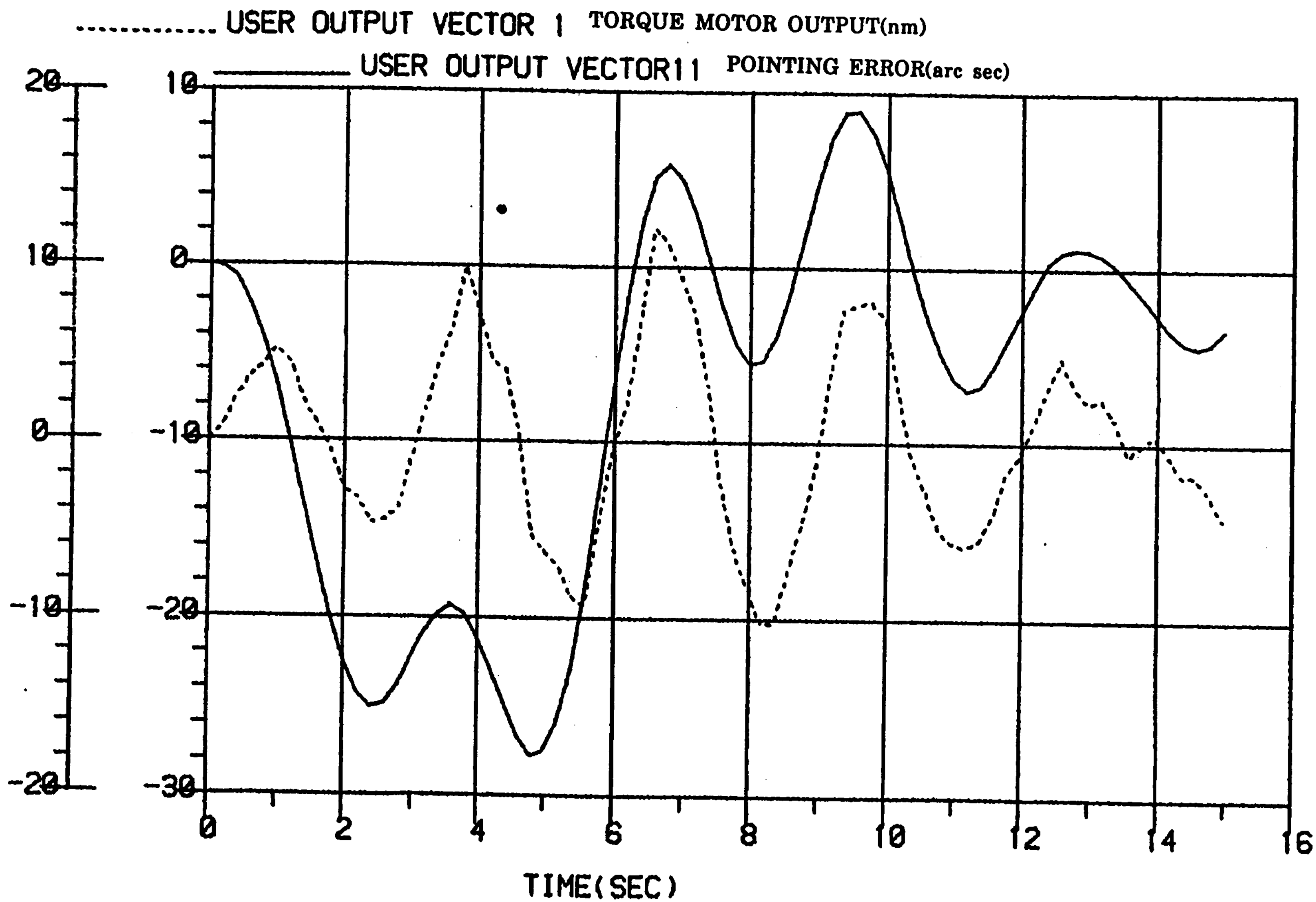
SPACE STATION PITCH JET DISTURBANCE



0

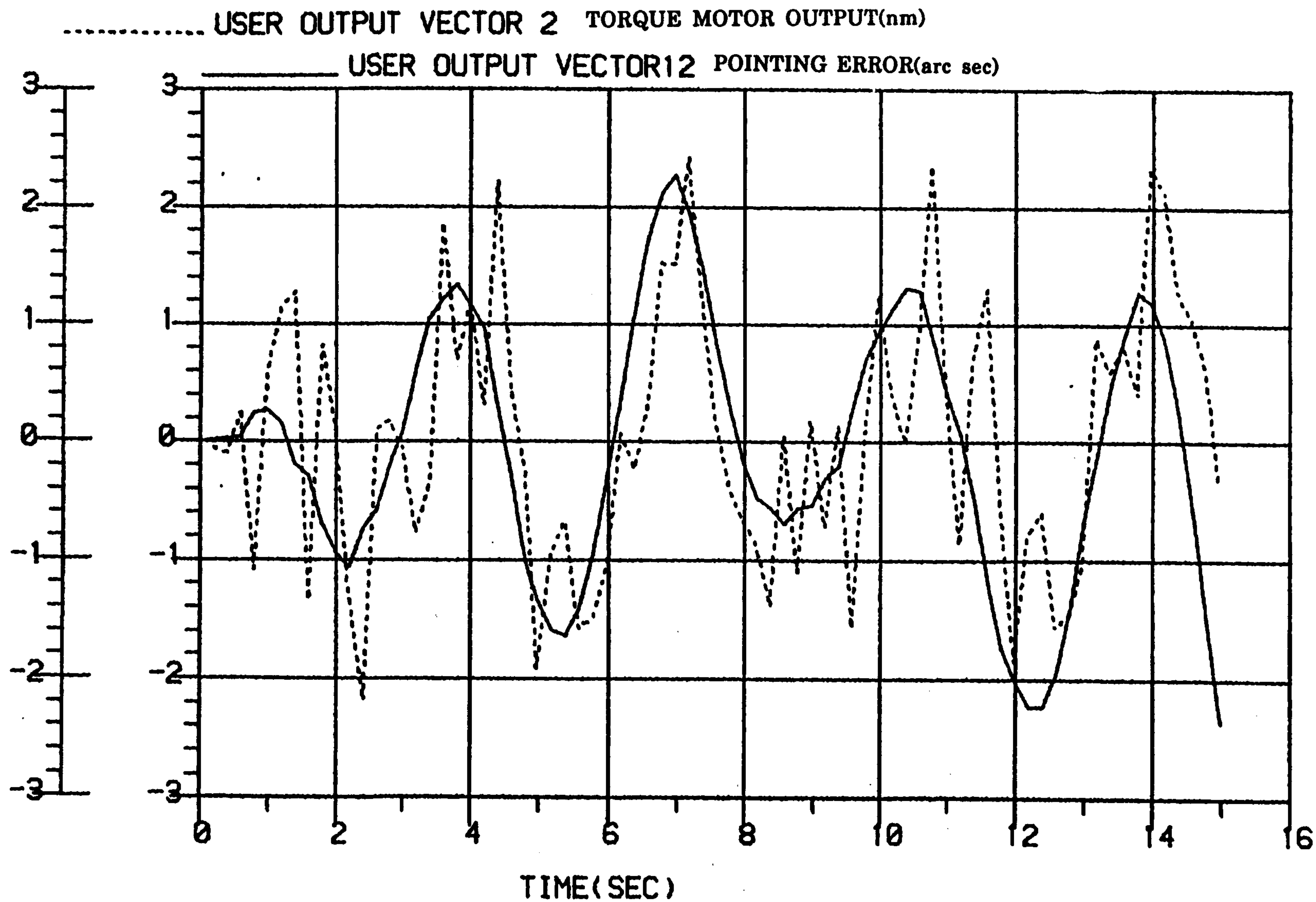
SPACE STATION PITCH JET DISTURBANCE

POF ELEVATION AXIS



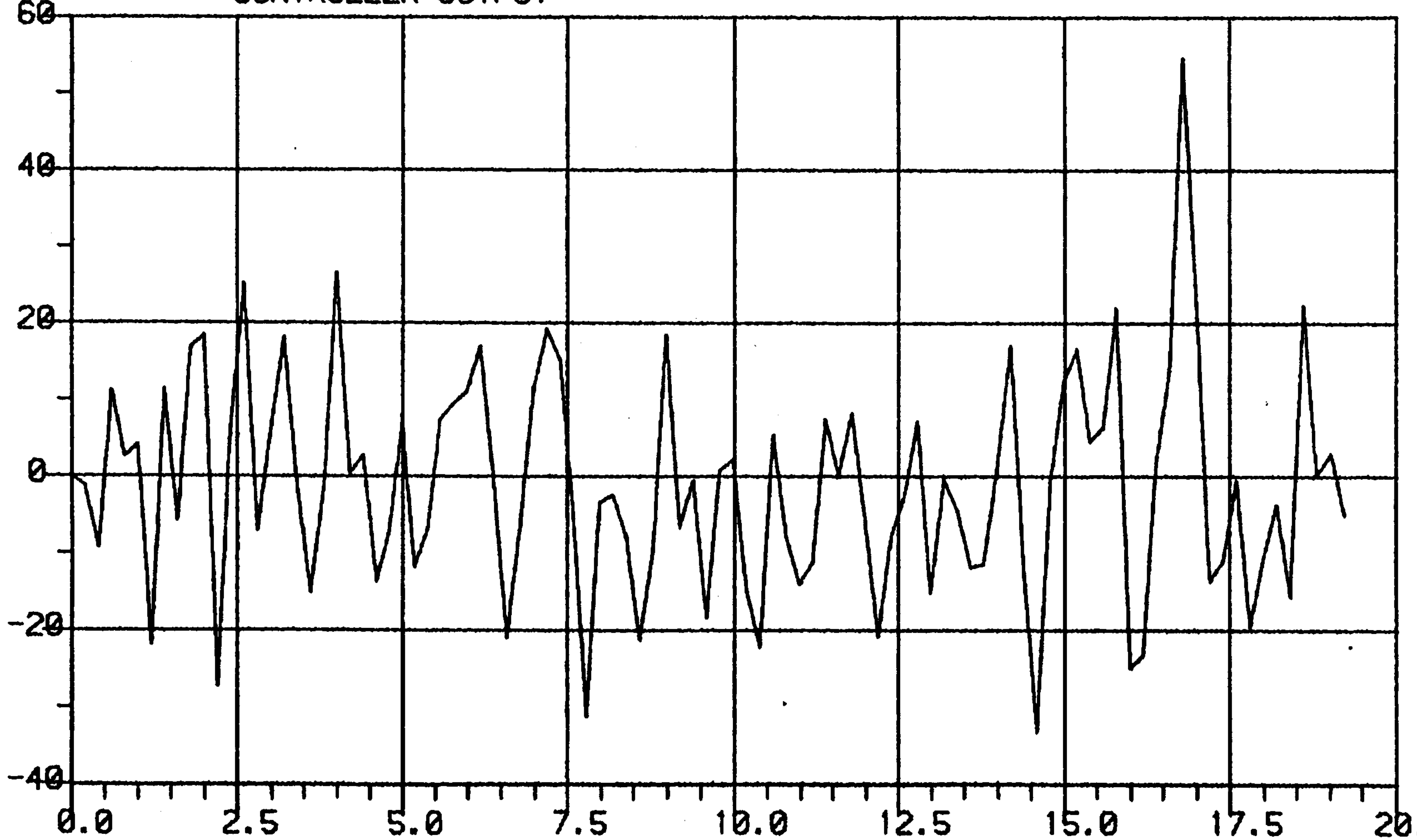
SPACE STATION PITCH JET DISTURBANCE

POF CROSS ELEVATION AXIS



SPACE STATION LOW LEVEL CREW MOTION FORCE

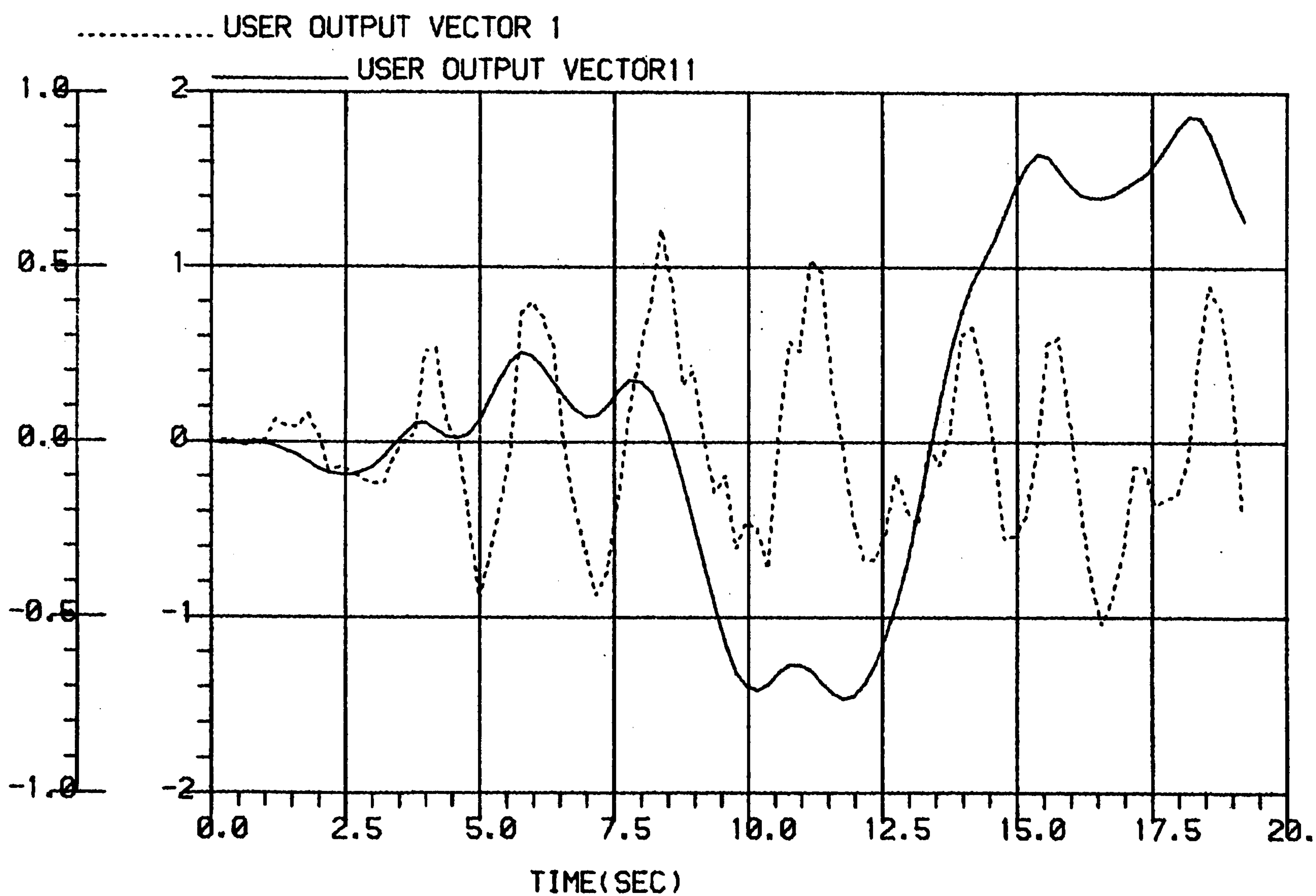
CREW MOTION FORCE



0

SPACE STATION LOW LEVEL CREW MOTION FORCE

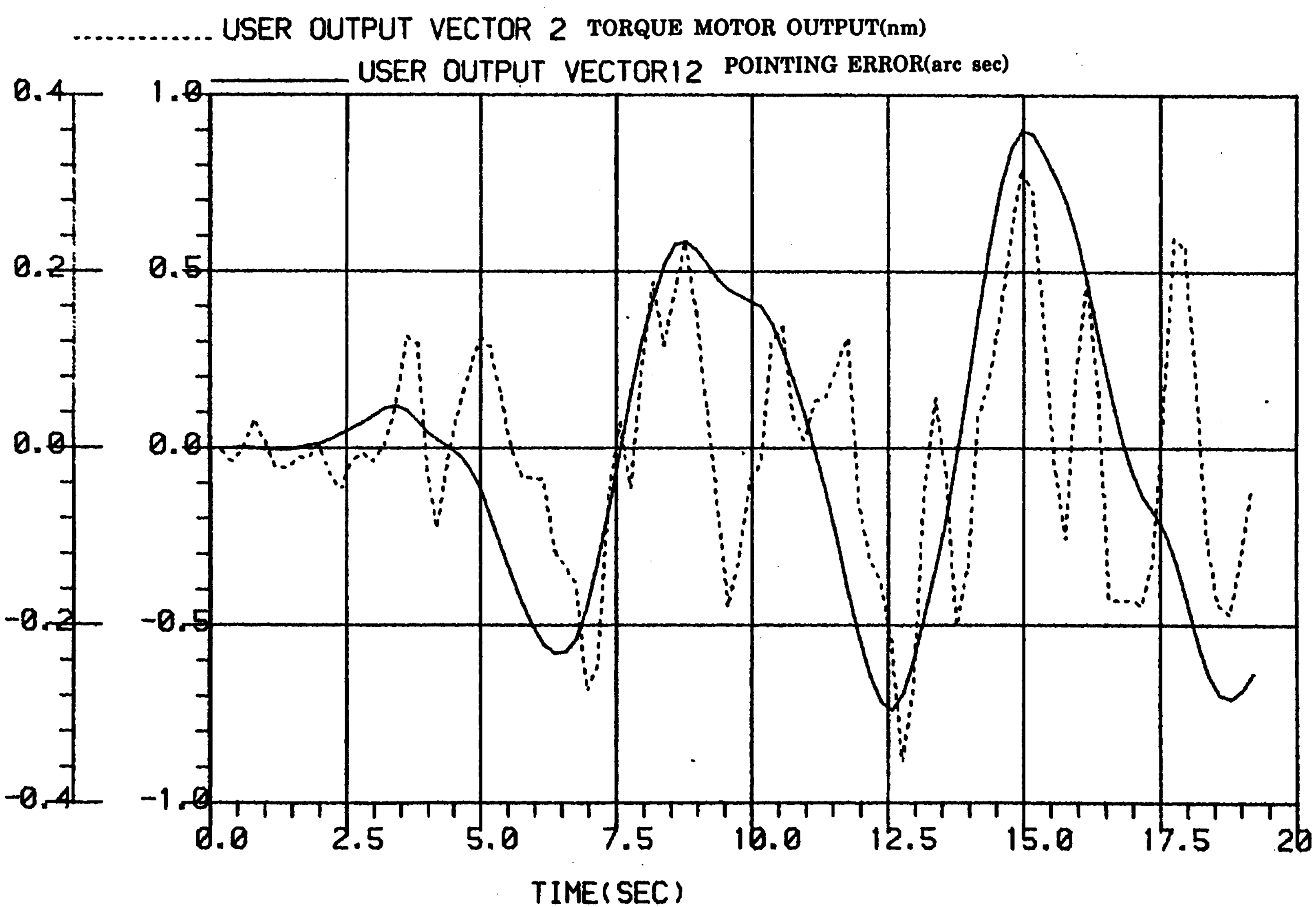
POF ELEVATION AXIS



0

SPACE STATION LOW LEVEL CREW MOTION FORCE

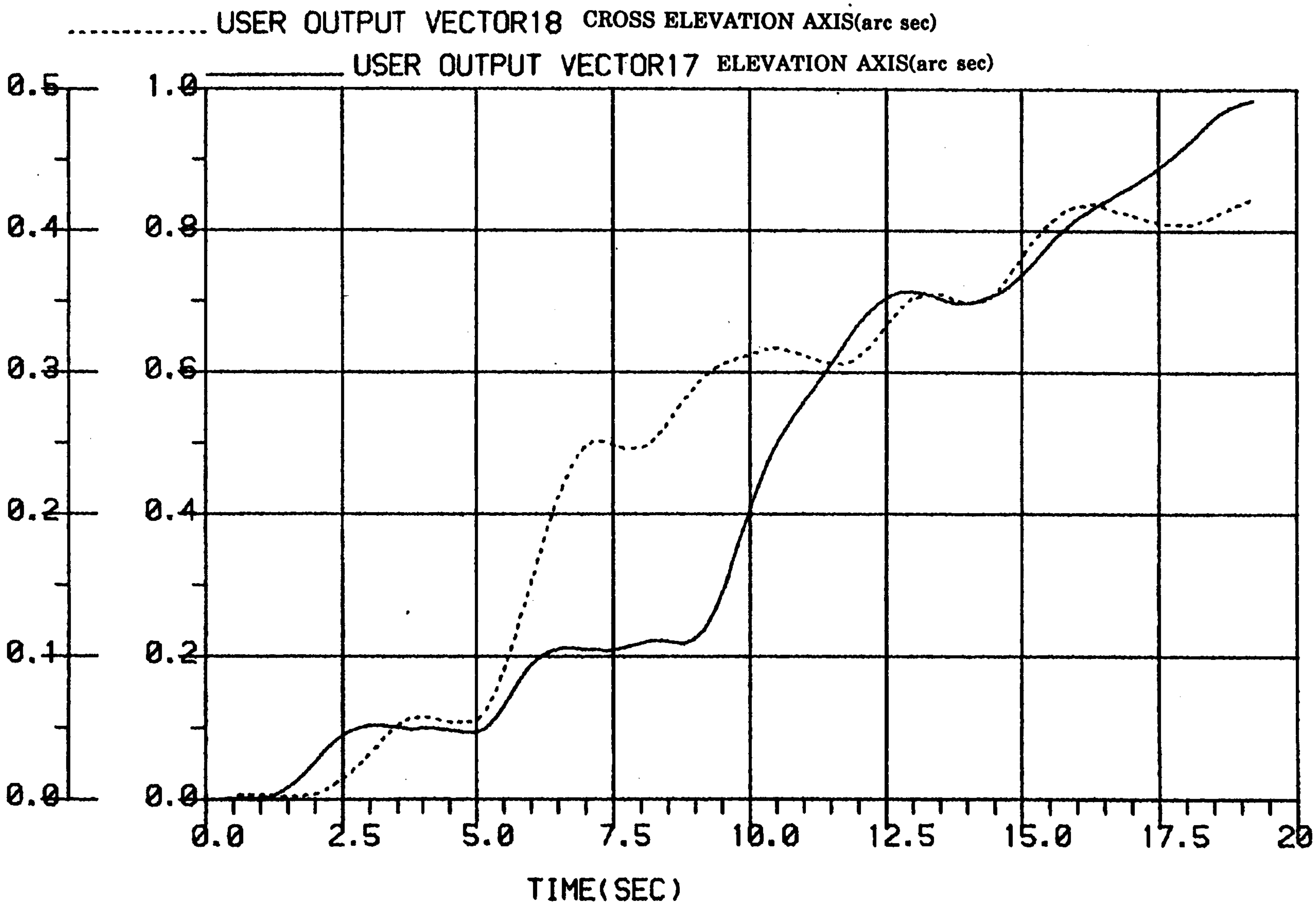
POF CROSS ELEVATION AXIS



0

SPACE STATION LOW LEVEL CREW MOTION FORCE

POINTING STANDARD DEVIATION



0

TREETOPS FILES

File Name	Description
POF3.INT	Orbiter/POF Interactive File
GSSMP3.INT	Space Station/POF Interactive File
DPOF3CE.FOR	Orbiter/POF User Controller
SSPOF3.FOR	Space Station/POF User Controller
DIST3.FOR	User Disturbance
DPOF3CE.DAT	Sensor and Actuator Error Source Parameters

POF3.INT

```
 1.000  SIM CONTROL #  0 Title header of user problem (< 40 char)           = POF3      :
 2.000  SIM CONTROL #  0 Simulation stop time(sec)                          =   5.0000
 3.000  SIM CONTROL #  0 Data output delta, Initial, Final time (sec)       =  0.10000     0.00000E+00   5.0000
 4.000  SIM CONTROL #  0 Integration(R=rkuta,S=sandia,U=user),Dt(sec),Fname= R            0.50000E-02
 5.000  SIM CONTROL #  0 Linearization option (N=none,L=linear), Time(sec) = N
 6.000  SIM CONTROL #  0 Restart option (N=none,R=restart), Restart fname   = N            0.00000E+00
 7.000  SPEEDUP     #  0 Small angle computation(All,Bypass,First,Nth) pass= FIRST
 8.000  SPEEDUP     #  0 Mass matrix computation(All,Bypass,First,Nth) pass= BYPASS
 9.000  SPEEDUP     #  0 Non-linear computation(All,Bypass,First,Nth) pass = BYPASS
10.000  CONTROLLER  C  2 Type(C=contin,D=discrete,U=user), Dt(sec), Fname  = CONTINUOUS
11.000  CONTROLLER  C  2 Number of inputs, Number of outputs               =   16.000     16.000
12.000  CONTROLLER  C  2 OUTPUT  1 type(I=incont,T=trans,J=junct), ID#,Gain= T            1.0000       1.0000
13.000  CONTROLLER  C  2 OUTPUT  2 Type(I=incont,T=trans,J=junct), ID#,Gain= T            2.0000       1.0000
14.000  CONTROLLER  C  2 OUTPUT  3 Type(I=incont,T=trans,J=junct), ID#,Gain= T            3.0000       1.0000
15.000  CONTROLLER  C  2 OUTPUT  4 Type(I=incont,T=trans,J=junct), ID#,Gain= T            4.0000       1.0000
16.000  CONTROLLER  C  2 OUTPUT  5 Type(I=incont,T=trans,J=junct), ID#,Gain= T            5.0000       1.0000
17.000  CONTROLLER  C  2 OUTPUT  6 Type(I=incont,T=trans,J=junct), ID#,Gain= T            6.0000       1.0000
18.000  CONTROLLER  C  2 OUTPUT  7 Type(I=incont,T=trans,J=junct), ID#,Gain= T            7.0000       1.0000
19.000  CONTROLLER  C  2 OUTPUT  8 Type(I=incont,T=trans,J=junct), ID#,Gain= T            11.000       1.0000
20.000  CONTROLLER  C  2 OUTPUT  9 Type(I=incont,T=trans,J=junct), ID#,Gain= T            12.000       1.0000
21.000  CONTROLLER  C  2 OUTPUT 10 Type(I=incont,T=trans,J=junct), ID#,Gain= T            13.000       1.0000
22.000  CONTROLLER  C  2 OUTPUT 11 Type(I=incont,T=trans,J=junct), ID#,Gain= T            17.000       1.0000
23.000  CONTROLLER  C  2 OUTPUT 12 Type(I=incont,T=trans,J=junct), ID#,Gain= T            18.000       1.0000
24.000  CONTROLLER  C  2 OUTPUT 13 Type(I=incont,T=trans,J=junct), ID#,Gain= T            19.000       1.0000
25.000  CONTROLLER  C  2 OUTPUT 14 Type(I=incont,T=trans,J=junct), ID#,Gain= T            20.000       0.00000E+00
26.000  CONTROLLER  C  2 OUTPUT 15 Type(I=incont,T=trans,J=junct), ID#,Gain= T            21.000       0.00000E+00
27.000  CONTROLLER  C  2 OUTPUT 16 Type(I=incont,T=trans,J=junct), ID#,Gain= T            22.000       0.00000E+00
28.000  TRANS FUNCT C  1 Input type(I=incont,T=trans,J=junct), Input ID#   = I            1.0000       CONTINUOUS
29.000  TRANS FUNCT C  1 Order of numerator, Order of denominator           =  0.00000E+00 1.0000
30.000  TRANS FUNCT C  1 Numerator coeff (ascending order 1-4 per line)     =   1.0000
31.000  TRANS FUNCT C  1 Denominator coeff (ascending order 1-4 per line)   =   1.0000     0.16000E-01
32.000  TRANS FUNCT C  2 Input type(I=incont,T=trans,J=junct), Input ID#   = I            2.0000       CONTINUOUS
33.000  TRANS FUNCT C  2 Order of numerator, Order of denominator           =  0.00000E+00 1.0000
34.000  TRANS FUNCT C  2 Numerator coeff (ascending order 1-4 per line)     =   1.0000
35.000  TRANS FUNCT C  2 Denominator coeff (ascending order 1-4 per line)   =   1.0000     0.16000E-01
36.000  TRANS FUNCT C  3 Input type(I=incont,T=trans,J=junct), Input ID#   = I            3.0000       CONTINUOUS
37.000  TRANS FUNCT C  3 Order of numerator, Order of denominator           =  0.00000E+00 1.0000
38.000  TRANS FUNCT C  3 Numerator coeff (ascending order 1-4 per line)     =   1.0000
39.000  TRANS FUNCT C  3 Denominator coeff (ascending order 1-4 per line)   =   1.0000     0.16000E-01
40.000  TRANS FUNCT C  4 Input type(I=incont,T=trans,J=junct), Input ID#   = I            4.0000       CONTINUOUS
41.000  TRANS FUNCT C  4 Order of numerator, Order of denominator           =  0.00000E+00 1.0000
42.000  TRANS FUNCT C  4 Numerator coeff (ascending order 1-4 per line)     =   1.0000
43.000  TRANS FUNCT C  4 Denominator coeff (ascending order 1-4 per line)   =   1.0000     0.16000E-01
44.000  TRANS FUNCT C  5 Input type(I=incont,T=trans,J=junct), Input ID#   = I            5.0000       CONTINUOUS
45.000  TRANS FUNCT C  5 Order of numerator, Order of denominator           =  0.00000E+00 2.0000
46.000  TRANS FUNCT C  5 Numerator coeff (ascending order 1-4 per line)     =   15790.
47.000  TRANS FUNCT C  5 Denominator coeff (ascending order 1-4 per line)   =   15790.     175.92       1.0000
48.000  TRANS FUNCT C  6 Input type(I=incont,T=trans,J=junct), Input ID#   = I            6.0000       CONTINUOUS
49.000  TRANS FUNCT C  6 Order of numerator, Order of denominator           =  0.00000E+00 2.0000
50.000  TRANS FUNCT C  6 Numerator coeff (ascending order 1-4 per line)     =   15790.
51.000  TRANS FUNCT C  6 Denominator coeff (ascending order 1-4 per line)   =   15790.     175.92       1.0000
52.000  TRANS FUNCT C  7 Input type(I=incont,T=trans,J=junct), Input ID#   = I            7.0000       CONTINUOUS
53.000  TRANS FUNCT C  7 Order of numerator, Order of denominator           =  0.00000E+00 2.0000
54.000  TRANS FUNCT C  7 Numerator coeff (ascending order 1-4 per line)     =   15790.
55.000  TRANS FUNCT C  7 Denominator coeff (ascending order 1-4 per line)   =   15790.     175.92       1.0000
56.000  TRANS FUNCT C  8 Input type(I=incont,T=trans,J=junct), Input ID#   = I            8.0000       CONTINUOUS
57.000  TRANS FUNCT C  8 Order of numerator, Order of denominator           =   1.0000     2.0000
58.000  TRANS FUNCT C  8 Numerator coeff (ascending order 1-4 per line)     =   20884.     1384.9
59.000  TRANS FUNCT C  8 Denominator coeff (ascending order 1-4 per line)   =   20884.     204.30       1.0000
60.000  TRANS FUNCT C  9 Input type(I=incont,T=trans,J=junct), Input ID#   = I            9.0000       CONTINUOUS
61.000  TRANS FUNCT C  9 Order of numerator, Order of denominator           =   1.0000     2.0000
```

```
 62.000  TRANS FUNCT C    9 Numerator coeff (ascending order 1-4 per line)    =   20884.        1384.9
 63.000  TRANS FUNCT C    9 Denominator coeff (ascending order 1-4 per line)  =   20884.        204.30        1.0000
 64.000  TRANS FUNCT C   10 Input type(I=incont,T=trans,J=junct), Input ID#   =  I              10.000       CONTINUOUS
 65.000  TRANS FUNCT C   10 Order of numerator, Order of denominator          =   1.0000        2.0000
 66.000  TRANS FUNCT C   10 Numerator coeff (ascending order 1-4 per line)    =   20884.        1384.9
 67.000  TRANS FUNCT C   10 Denominator coeff (ascending order 1-4 per line)  =   20884.        204.30        1.0000
 68.000  TRANS FUNCT C   11 Input type(I=incont,T=trans,J=junct), Input ID#   =  T              8.0000       CONTINUOUS
 69.000  TRANS FUNCT C   11 Order of numerator, Order of denominator          =  0.00000E+00    2.0000
 70.000  TRANS FUNCT C   11 Numerator coeff (ascending order 1-4 per line)    =   83521.
 71.000  TRANS FUNCT C   11 Denominator coeff (ascending order 1-4 per line)  =   83521.        202.30        1.0000
 72.000  TRANS FUNCT C   12 Input type(I=incont,T=trans,J=junct), Input ID#   =  T              9.0000       CONTINUOUS
 73.000  TRANS FUNCT C   12 Order of numerator, Order of denominator          =  0.00000E+00    2.0000
 74.000  TRANS FUNCT C   12 Numerator coeff (ascending order 1-4 per line)    =   83521.
 75.000  TRANS FUNCT C   12 Denominator coeff (ascending order 1-4 per line)  =   83521.        202.30        1.0000
 76.000  TRANS FUNCT C   13 Input type(I=incont,T=trans,J=junct), Input ID#   =  T              10.000       CONTINUOUS
 77.000  TRANS FUNCT C   13 Order of numerator, Order of denominator          =  0.00000E+00    2.0000
 78.000  TRANS FUNCT C   13 Numerator coeff (ascending order 1-4 per line)    =   83521.
 79.000  TRANS FUNCT C   13 Denominator coeff (ascending order 1-4 per line)  =   83521.        202.30        1.0000
 80.000  TRANS FUNCT C   14 Input type(I=incont,T=trans,J=junct), Input ID#   =  I              11.000       CONTINUOUS
 81.000  TRANS FUNCT C   14 Order of numerator, Order of denominator          =  0.00000E+00    1.0000
 82.000  TRANS FUNCT C   14 Numerator coeff (ascending order 1-4 per line)    =   1.0000
 83.000  TRANS FUNCT C   14 Denominator coeff (ascending order 1-4 per line)  =   1.0000       0.44200E-02
 84.000  TRANS FUNCT C   15 Input type(I=incont,T=trans,J=junct), Input ID#   =  I              12.000       CONTINUOUS
 85.000  TRANS FUNCT C   15 Order of numerator, Order of denominator          =  0.00000E+00    1.0000
 86.000  TRANS FUNCT C   15 Numerator coeff (ascending order 1-4 per line)    =   1.0000
 87.000  TRANS FUNCT C   15 Denominator coeff (ascending order 1-4 per line)  =   1.0000       0.44200E-02
 88.000  TRANS FUNCT C   16 Input type(I=incont,T=trans,J=junct), Input ID#   =  I              13.000       CONTINUOUS
 89.000  TRANS FUNCT C   16 Order of numerator, Order of denominator          =  0.00000E+00    1.0000
 90.000  TRANS FUNCT C   16 Numerator coeff (ascending order 1-4 per line)    =   1.0000
 91.000  TRANS FUNCT C   16 Denominator coeff (ascending order 1-4 per line)  =   1.0000       0.44200E-02
 92.000  TRANS FUNCT C   17 Input type(I=incont,T=trans,J=junct), Input ID#   =  T              14.000       CONTINUOUS
 93.000  TRANS FUNCT C   17 Order of numerator, Order of denominator          =  0.00000E+00    2.0000
 94.000  TRANS FUNCT C   17 Numerator coeff (ascending order 1-4 per line)    =   90958.
 95.000  TRANS FUNCT C   17 Denominator coeff (ascending order 1-4 per line)  =   90958.        390.27        1.0000
 96.000  TRANS FUNCT C   18 Input type(I=incont,T=trans,J=junct), Input ID#   =  T              15.000       CONTINUOUS
 97.000  TRANS FUNCT C   18 Order of numerator, Order of denominator          =  0.00000E+00    2.0000
 98.000  TRANS FUNCT C   18 Numerator coeff (ascending order 1-4 per line)    =   90958.
 99.000  TRANS FUNCT C   18 Denominator coeff (ascending order 1-4 per line)  =   90958.        390.27        1.0000
100.000  TRANS FUNCT C   19 Input type(I=incont,T=trans,J=junct), Input ID#   =  T              16.000       CONTINUOUS
101.000  TRANS FUNCT C   19 Order of numerator, Order of denominator          =  0.00000E+00    2.0000
102.000  TRANS FUNCT C   19 Numerator coeff (ascending order 1-4 per line)    =   90958.
103.000  TRANS FUNCT C   19 Denominator coeff (ascending order 1-4 per line)  =   90958.        390.27        1.0000
104.000  TRANS FUNCT C   20 Input type(I=incont,T=trans,J=junct), Input ID#   =  I              14.000       CONTINUOUS
105.000  TRANS FUNCT C   20 Order of numerator, Order of denominator          =   1.0000        2.0000
106.000  TRANS FUNCT C   20 Numerator coeff (ascending order 1-4 per line)    =  0.00000E+00    343.24
107.000  TRANS FUNCT C   20 Denominator coeff (ascending order 1-4 per line)  =   30.030        12.930        1.0000
108.000  TRANS FUNCT C   21 Input type(I=incont,T=trans,J=junct), Input ID#   =  I              15.000       CONTINUOUS
109.000  TRANS FUNCT C   21 Order of numerator, Order of denominator          =   1.0000        2.0000
110.000  TRANS FUNCT C   21 Numerator coeff (ascending order 1-4 per line)    =  0.00000E+00    343.24
111.000  TRANS FUNCT C   21 Denominator coeff (ascending order 1-4 per line)  =   30.030        12.930        1.0000
112.000  TRANS FUNCT C   22 Input type(I=incont,T=trans,J=junct), Input ID#   =  I              16.000       CONTINUOUS
113.000  TRANS FUNCT C   22 Order of numerator, Order of denominator          =   1.0000        2.0000
114.000  TRANS FUNCT C   22 Numerator coeff (ascending order 1-4 per line)    =  0.00000E+00    343.24
115.000  TRANS FUNCT C   22 Denominator coeff (ascending order 1-4 per line)  =   30.030        12.930        1.0000
116.000  TRANS FUNCT C   22 >>>>>***** END OF DATA *****<<<<<                 =
117.000  SUM JUNCT   C   19 >>>>>***** END OF DATA *****<<<<<                 =
118.000  CONTROLLER  U    1 Type(C=contin,D=discrete,U=user), Dt(sec), Fname  =  USER          0.25000E-01  DPOF3CE
119.000  CONTROLLER  U    1 Number of inputs, Number of outputs               =   20.000        30.000
120.000  FUNCT GENER #    1 type(step,ramp,pulse,sawtooth,sine,user)          =  STEP
121.000  FUNCT GENER #    1 step amplitude, start time(sec)                   =  -0.20000      0.00000E+00
122.000  FUNCT GENER #    1 >>>>>***** END OF DATA *****<<<<<                 =
```

```
123.000  FUNCT GENER *   2 type(step,ramp,pulse,sawtooth,sine,user)              =  STEP
124.000  FUNCT GENER *   2 step amplitude, start time(sec)                       =   0.20000       0.00000E+00
125.000  FUNCT GENER *   2 >>>>>***** END OF DATA *****<<<<<                     =
126.000  FUNCT GENER *   3 type(step,ramp,pulse,sawtooth,sine,user)              =  STEP
127.000  FUNCT GENER *   3 step amplitude, start time(sec)                       =  -0.20000       0.00000E+00
128.000  FUNCT GENER *   3 >>>>>***** END OF DATA *****<<<<<                     =
129.000  FUNCT GENER *   4 type(step,ramp,pulse,sawtooth,sine,user)              =  PULSE
130.000  FUNCT GENER *   4 pulse amplitude, start time, stop time(sec)           =   106.80        0.00000E+00  0.64000
131.000  FUNCT GENER *   4 >>>>>***** END OF DATA *****<<<<<                     =
132.000  FUNCT GENER *   5 type(step,ramp,pulse,sawtooth,sine,user)              =  USER
133.000  FUNCT GENER *   5 user fname(use same name for additional usergener)=  DIST3
134.000  FUNCT GENER *   5 >>>>>***** END OF DATA *****<<<<<                     =
135.000  FUNCT GENER *   6 type(step,ramp,pulse,sawtooth,sine,user)              =  USER
136.000  FUNCT GENER *   6 user fname(use same name for additional usergener)=  DIST3
137.000  FUNCT GENER *   6 >>>>>***** END OF DATA *****<<<<<                     =
138.000  FUNCT GENER *   7 type(step,ramp,pulse,sawtooth,sine,user)              =  USER
139.000  FUNCT GENER *   7 user fname(use same name for additional usergener)=  DIST3
140.000  FUNCT GENER *   7 >>>>>***** END OF DATA *****<<<<<                     =
141.000  SENSOR      *   1 Type Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos,Los,ID#=  G              1.0000
142.000  SENSOR      *   1 ID# of body, index of node location               =   5.0000         3.0000
143.000  SENSOR      *   1 Input axis unit vector (body coord) X,Y,Z         =   0.00000E+00   0.00000E+00   1.0000
144.000  SENSOR      *   1 >>>>>***** END OF DATA *****<<<<<                     =
145.000  SENSOR      *   2 Type Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos,Los,ID#=  G              2.0000
146.000  SENSOR      *   2 ID# of body, index of node location               =   5.0000         3.0000
147.000  SENSOR      *   2 Input axis unit vector (body coord) X,Y,Z         =   0.00000E+00    1.0000       0.00000E+00
148.000  SENSOR      *   2 >>>>>***** END OF DATA *****<<<<<                     =
149.000  SENSOR      *   3 Type Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos,Los,ID#=  G              3.0000
150.000  SENSOR      *   3 ID# of body, index of node location               =   5.0000         3.0000
151.000  SENSOR      *   3 Input axis unit vector (body coord) X,Y,Z         =   1.0000        0.00000E+00   0.00000E+00
152.000  SENSOR      *   3 >>>>>***** END OF DATA *****<<<<<                     =
153.000  SENSOR      *   4 Type Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos,Los,ID#=  A              4.0000
154.000  SENSOR      *   4 ID# of body, index of node location               =   2.0000         3.0000
155.000  SENSOR      *   4 Input axis unit vector (body coord) X,Y,Z         =   1.0000        0.00000E+00   0.00000E+00
156.000  SENSOR      *   4 >>>>>***** END OF DATA *****<<<<<                     =
157.000  SENSOR      *   5 Type Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos,Los,ID#=  A              5.0000
158.000  SENSOR      *   5 ID# of body, index of node location               =   2.0000         3.0000
159.000  SENSOR      *   5 Input axis unit vector (body coord) X,Y,Z         =   0.00000E+00    1.0000       0.00000E+00
160.000  SENSOR      *   5 >>>>>***** END OF DATA *****<<<<<                     =
161.000  SENSOR      *   6 Type Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos,Los,ID#=  A              6.0000
162.000  SENSOR      *   6 ID# of body, index of node location               =   2.0000         3.0000
163.000  SENSOR      *   6 Input axis unit vector (body coord) X,Y,Z         =   0.00000E+00   0.00000E+00   1.0000
164.000  SENSOR      *   6 >>>>>***** END OF DATA *****<<<<<                     =
165.000  SENSOR      *   7 Type Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos,Los,ID#=  G              7.0000
166.000  SENSOR      *   7 ID# of body, index of node location               =   3.0000         1.0000
167.000  SENSOR      *   7 Input axis unit vector (body coord) X,Y,Z         =   0.00000E+00    1.0000       0.00000E+00
168.000  SENSOR      *   7 >>>>>***** END OF DATA *****<<<<<                     =
169.000  SENSOR      *   8 Type Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos,Los,ID#=  G              8.0000
170.000  SENSOR      *   8 ID# of body, index of node location               =   4.0000         1.0000
171.000  SENSOR      *   8 Input axis unit vector (body coord) X,Y,Z         =   0.00000E+00   0.00000E+00   1.0000
172.000  SENSOR      *   8 >>>>>***** END OF DATA *****<<<<<                     =
173.000  SENSOR      *  99 Type Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos,Los,ID#=  LOS            99.000
174.000  SENSOR      *  99 ID# of body, index of node location               =   6.0000         2.0000
175.000  SENSOR      *  99 Pinhole target unit vector (inertial coord) X,Y,Z =   0.00000E+00   0.00000E+00  -1.0000
176.000  SENSOR      *  99 >>>>>***** END OF DATA *****<<<<<                     =
177.000  ACTUATOR    *   1 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  J              1.0000
178.000  ACTUATOR    *   1 ID# of body, index of node location               =   1.0000         3.0000
179.000  ACTUATOR    *   1 Output axis unit vector(body coord) X,Y,Z         =  -0.32650E-01  -0.69625      -0.71706
180.000  ACTUATOR    *   1 >>>>>***** END OF DATA *****<<<<<                     =
181.000  ACTUATOR    *   2 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  J              2.0000
182.000  ACTUATOR    *   2 ID# of body, index of node location               =   1.0000         4.0000
183.000  ACTUATOR    *   2 Output axis unit vector(body coord) X,Y,Z         =  -0.32650E-01   0.69625      -0.71706
```

```
184.000  ACTUATOR    #   2 >>>>>***** END OF DATA *****<<<<<                =
185.000  ACTUATOR    #   3 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  J              3.0000
186.000  ACTUATOR    #   3 ID# of body, index of node location              =   1.0000        5.0000
187.000  ACTUATOR    #   3 Output axis unit vector(body coord) X,Y,Z         =  0.00000E+00 -0.99967     -0.25820E-01
188.000  ACTUATOR    #   3 >>>>>***** END OF DATA *****<<<<<                =
189.000  ACTUATOR    #   4 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  J              4.0000
190.000  ACTUATOR    #   4 ID# of body, index of node location              =   1.0000        6.0000
191.000  ACTUATOR    #   4 Output axis unit vector(body coord) X,Y,Z         =  0.00000E+00  0.99967     -0.25820E-01
192.000  ACTUATOR    #   4 >>>>>***** END OF DATA *****<<<<<                =
193.000  ACTUATOR    #   5 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  J              5.0000
194.000  ACTUATOR    #   5 ID# of body, index of node location              =   1.0000        7.0000
195.000  ACTUATOR    #   5 Output axis unit vector(body coord) X,Y,Z         =  0.26560E-01 -0.28662     -0.95768
196.000  ACTUATOR    #   5 >>>>>***** END OF DATA *****<<<<<                =
197.000  ACTUATOR    #   6 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  J              6.0000
198.000  ACTUATOR    #   6 ID# of body, index of node location              =   1.0000        8.0000
199.000  ACTUATOR    #   6 Output axis unit vector(body coord) X,Y,Z         =  0.26560E-01  0.28662     -0.95768
200.000  ACTUATOR    #   6 >>>>>***** END OF DATA *****<<<<<                =
201.000  ACTUATOR    #   7 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  T              7.0000
202.000  ACTUATOR    #   7 ID# of hinge, index of rotation axis             =   5.0000        1.0000
203.000  ACTUATOR    #   7 >>>>>***** END OF DATA *****<<<<<                =
204.000  ACTUATOR    #   8 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  T              8.0000
205.000  ACTUATOR    #   8 ID# of hinge, index of rotation axis             =   3.0000        1.0000
206.000  ACTUATOR    #   8 >>>>>***** END OF DATA *****<<<<<                =
207.000  ACTUATOR    #   9 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  T              9.0000
208.000  ACTUATOR    #   9 ID# of hinge, index of rotation axis             =   4.0000        1.0000
209.000  ACTUATOR    #   9 >>>>>***** END OF DATA *****<<<<<                =
210.000  ACTUATOR    #  10 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  J              10.000
211.000  ACTUATOR    #  10 ID# of body, index of node location              =   1.0000        9.0000
212.000  ACTUATOR    #  10 Output axis unit vector(body coord) X,Y,Z         =   1.0000        0.00000E+00  0.00000E+00
213.000  ACTUATOR    #  10 >>>>>***** END OF DATA *****<<<<<                =
214.000  ACTUATOR    #  11 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  J              11.000
215.000  ACTUATOR    #  11 ID# of body, index of node location              =   1.0000        9.0000
216.000  ACTUATOR    #  11 Output axis unit vector(body coord) X,Y,Z         =  0.00000E+00  1.0000       0.00000E+00
217.000  ACTUATOR    #  11 >>>>>***** END OF DATA *****<<<<<                =
218.000  ACTUATOR    #  12 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  J              12.000
219.000  ACTUATOR    #  12 ID# of body, index of node location              =   1.0000        9.0000
220.000  ACTUATOR    #  12 Output axis unit vector(body coord) X,Y,Z         =  0.00000E+00  0.00000E+00  1.0000
221.000  ACTUATOR    #  12 >>>>>***** END OF DATA *****<<<<<                =
222.000  INT CONNECT #   1 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  U              5.0000
223.000  INT CONNECT #   1 Destination(C=contin,D=disc,U=user,A=actuator),ID#= A              1.0000
224.000  INT CONNECT #   1 >>>>>***** END OF DATA *****<<<<<                =
225.000  INT CONNECT #   2 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  U              6.0000
226.000  INT CONNECT #   2 Destination(C=contin,D=disc,U=user,A=actuator),ID#= A              2.0000
227.000  INT CONNECT #   2 >>>>>***** END OF DATA *****<<<<<                =
228.000  INT CONNECT #   3 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  U              7.0000
229.000  INT CONNECT #   3 Destination(C=contin,D=disc,U=user,A=actuator),ID#= A              3.0000
230.000  INT CONNECT #   3 >>>>>***** END OF DATA *****<<<<<                =
231.000  INT CONNECT #   4 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  U              8.0000
232.000  INT CONNECT #   4 Destination(C=contin,D=disc,U=user,A=actuator),ID#= A              4.0000
233.000  INT CONNECT #   4 >>>>>***** END OF DATA *****<<<<<                =
234.000  INT CONNECT #   5 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  U              9.0000
235.000  INT CONNECT #   5 Destination(C=contin,D=disc,U=user,A=actuator),ID#= A              5.0000
236.000  INT CONNECT #   5 >>>>>***** END OF DATA *****<<<<<                =
237.000  INT CONNECT #   6 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  U              10.000
238.000  INT CONNECT #   6 Destination(C=contin,D=disc,U=user,A=actuator),ID#= A              6.0000
239.000  INT CONNECT #   6 >>>>>***** END OF DATA *****<<<<<                =
240.000  INT CONNECT #   7 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  U              1.0000
241.000  INT CONNECT #   7 Destination(C=contin,D=disc,U=user,A=actuator),ID#= A              8.0000
242.000  INT CONNECT #   7 >>>>>***** END OF DATA *****<<<<<                =
243.000  INT CONNECT #   8 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  U              2.0000
244.000  INT CONNECT #   8 Destination(C=contin,D=disc,U=user,A=actuator),ID#= A              9.0000
```

```
245.000  INT CONNECT #   8 >>>>>***** END OF DATA *****<<<<<           =
246.000  INT CONNECT #   9 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#    =  U        3.0000
247.000  INT CONNECT #   9 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  A        7.0000
248.000  INT CONNECT #   9 >>>>>***** END OF DATA *****<<<<<           =
249.000  INT CONNECT #  10 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#    =  S        997.00
250.000  INT CONNECT #  10 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C        1.0000
251.000  INT CONNECT #  10 >>>>>***** END OF DATA *****<<<<<           =
252.000  INT CONNECT #  11 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#    =  S        996.00
253.000  INT CONNECT #  11 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C        2.0000
254.000  INT CONNECT #  11 >>>>>***** END OF DATA *****<<<<<           =
255.000  INT CONNECT #  12 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#    =  S        994.00
256.000  INT CONNECT #  12 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C        3.0000
257.000  INT CONNECT #  12 >>>>>***** END OF DATA *****<<<<<           =
258.000  INT CONNECT #  13 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#    =  S        995.00
259.000  INT CONNECT #  13 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C        4.0000
260.000  INT CONNECT #  13 >>>>>***** END OF DATA *****<<<<<           =
261.000  INT CONNECT #  14 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#    =  S        2.0000
262.000  INT CONNECT #  14 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C        5.0000
263.000  INT CONNECT #  14 >>>>>***** END OF DATA *****<<<<<           =
264.000  INT CONNECT #  15 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#    =  S        3.0000
265.000  INT CONNECT #  15 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C        6.0000
266.000  INT CONNECT #  15 >>>>>***** END OF DATA *****<<<<<           =
267.000  INT CONNECT #  16 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#    =  S        1.0000
268.000  INT CONNECT #  16 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C        7.0000
269.000  INT CONNECT #  16 >>>>>***** END OF DATA *****<<<<<           =
270.000  INT CONNECT #  17 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#    =  U        21.000
271.000  INT CONNECT #  17 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C        8.0000
272.000  INT CONNECT #  17 >>>>>***** END OF DATA *****<<<<<           =
273.000  INT CONNECT #  18 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#    =  U        22.000
274.000  INT CONNECT #  18 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C        9.0000
275.000  INT CONNECT #  18 >>>>>***** END OF DATA *****<<<<<           =
276.000  INT CONNECT #  19 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#    =  U        23.000
277.000  INT CONNECT #  19 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C        10.000
278.000  INT CONNECT #  19 >>>>>***** END OF DATA *****<<<<<           =
279.000  INT CONNECT #  20 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#    =  U        24.000
280.000  INT CONNECT #  20 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C        11.000
281.000  INT CONNECT #  20 >>>>>***** END OF DATA *****<<<<<           =
282.000  INT CONNECT #  21 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#    =  U        25.000
283.000  INT CONNECT #  21 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C        12.000
284.000  INT CONNECT #  21 >>>>>***** END OF DATA *****<<<<<           =
285.000  INT CONNECT #  22 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#    =  U        26.000
286.000  INT CONNECT #  22 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C        13.000
287.000  INT CONNECT #  22 >>>>>***** END OF DATA *****<<<<<           =
288.000  INT CONNECT #  23 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#    =  C        1.0000
289.000  INT CONNECT #  23 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U        4.0000
290.000  INT CONNECT #  23 >>>>>***** END OF DATA *****<<<<<           =
291.000  INT CONNECT #  24 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#    =  C        2.0000
292.000  INT CONNECT #  24 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U        8.0000
293.000  INT CONNECT #  24 >>>>>***** END OF DATA *****<<<<<           =
294.000  INT CONNECT #  25 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#    =  C        3.0000
295.000  INT CONNECT #  25 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U        3.0000
296.000  INT CONNECT #  25 >>>>>***** END OF DATA *****<<<<<           =
297.000  INT CONNECT #  26 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#    =  C        4.0000
298.000  INT CONNECT #  26 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U        7.0000
299.000  INT CONNECT #  26 >>>>>***** END OF DATA *****<<<<<           =
300.000  INT CONNECT #  27 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#    =  C        5.0000
301.000  INT CONNECT #  27 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U        1.0000
302.000  INT CONNECT #  27 >>>>>***** END OF DATA *****<<<<<           =
303.000  INT CONNECT #  28 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#    =  C        6.0000
304.000  INT CONNECT #  28 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U        5.0000
305.000  INT CONNECT #  28 >>>>>***** END OF DATA *****<<<<<           =
```

```
306.000  INT CONNECT #  29 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  C              7.0000
307.000  INT CONNECT #  29 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U              9.0000
308.000  INT CONNECT #  29 >>>>>***** END OF DATA *****<<<<<                 =
309.000  INT CONNECT #  30 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  C              8.0000
310.000  INT CONNECT #  30 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U              14.000
311.000  INT CONNECT #  30 >>>>>***** END OF DATA *****<<<<<                 =
312.000  INT CONNECT #  31 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  C              9.0000
313.000  INT CONNECT #  31 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U              15.000
314.000  INT CONNECT #  31 >>>>>***** END OF DATA *****<<<<<                 =
315.000  INT CONNECT #  32 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  C              10.000
316.000  INT CONNECT #  32 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U              16.000
317.000  INT CONNECT #  32 >>>>>***** END OF DATA *****<<<<<                 =
318.000  INT CONNECT #  33 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  C              11.000
319.000  INT CONNECT #  33 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U              17.000
320.000  INT CONNECT #  33 >>>>>***** END OF DATA *****<<<<<                 =
321.000  INT CONNECT #  34 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  C              12.000
322.000  INT CONNECT #  34 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U              18.000
323.000  INT CONNECT #  34 >>>>>***** END OF DATA *****<<<<<                 =
324.000  INT CONNECT #  35 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  C              13.000
325.000  INT CONNECT #  35 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U              19.000
326.000  INT CONNECT #  35 >>>>>***** END OF DATA *****<<<<<                 =
327.000  INT CONNECT #  36 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  S              992.00
328.000  INT CONNECT #  36 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U              2.0000
329.000  INT CONNECT #  36 >>>>>***** END OF DATA *****<<<<<                 =
330.000  INT CONNECT #  37 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  S              991.00
331.000  INT CONNECT #  37 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U              6.0000
332.000  INT CONNECT #  37 >>>>>***** END OF DATA *****<<<<<                 =
333.000  INT CONNECT #  38 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  S              993.00
334.000  INT CONNECT #  38 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U              10.000
335.000  INT CONNECT #  38 >>>>>***** END OF DATA *****<<<<<                 =
336.000  INT CONNECT #  39 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  C              14.000
337.000  INT CONNECT #  39 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  A              10.000
338.000  INT CONNECT #  39 >>>>>***** END OF DATA *****<<<<<                 =
339.000  INT CONNECT #  40 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  C              15.000
340.000  INT CONNECT #  40 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  A              11.000
341.000  INT CONNECT #  40 >>>>>***** END OF DATA *****<<<<<                 =
342.000  INT CONNECT #  41 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  C              16.000
343.000  INT CONNECT #  41 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  A              12.000
344.000  INT CONNECT #  41 >>>>>***** END OF DATA *****<<<<<                 =
345.000  INT CONNECT #  42 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  G              1.0000
346.000  INT CONNECT #  42 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U              11.000
347.000  INT CONNECT #  42 >>>>>***** END OF DATA *****<<<<<                 =
348.000  INT CONNECT #  43 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  G              2.0000
349.000  INT CONNECT #  43 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U              12.000
350.000  INT CONNECT #  43 >>>>>***** END OF DATA *****<<<<<                 =
351.000  INT CONNECT #  44 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  G              3.0000
352.000  INT CONNECT #  44 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U              13.000
353.000  INT CONNECT #  44 >>>>>***** END OF DATA *****<<<<<                 =
354.000  INT CONNECT #  45 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  G              5.0000
355.000  INT CONNECT #  45 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C              14.000
356.000  INT CONNECT #  45 >>>>>***** END OF DATA *****<<<<<                 =
357.000  INT CONNECT #  46 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  G              6.0000
358.000  INT CONNECT #  46 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C              15.000
359.000  INT CONNECT #  46 >>>>>***** END OF DATA *****<<<<<                 =
360.000  INT CONNECT #  47 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  G              7.0000
361.000  INT CONNECT #  47 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C              16.000
362.000  INT CONNECT #  47 >>>>>***** END OF DATA *****<<<<<                 =
363.000  HINGE       #   1 ID# of hinge, ID# of inbd body, ID# of outbd body =    1.0000   0.00000E+00   1.0000
364.000  HINGE       #   1 Node index of inboard body attach point           =
365.000  HINGE       #   1 Rotation degrees of freedom                       =    3.0000
366.000  HINGE       #   1 Base body(ID# 1)rotation option(F=free,G=gimbaled)=  G
```

```
367.000 HINGE  *  1 Rotation axis unit vector L1 inboard body  X,Y,Z  =   1.0000       0.00000E+00  0.00000E+00
368.000 HINGE  *  1 Rotation axis unit vector L1 outboard body X,Y,Z  =   1.0000       0.00000E+00  0.00000E+00
369.000 HINGE  *  1 Rotation axis unit vector L3 inboard body  X,Y,Z  =  0.00000E+00  0.00000E+00   1.0000
370.000 HINGE  *  1 Rotation axis unit vector L3 outboard body X,Y,Z  =  0.00000E+00  0.00000E+00   1.0000
371.000 HINGE  *  1 Rotation spring stiffness (N/R)    k1,k2,k3       =  0.00000E+00  0.00000E+00  0.00000E+00
372.000 HINGE  *  1 Rotation spring damping (N/R/S)    b1,b2,b3       =  0.00000E+00  0.00000E+00  0.00000E+00
373.000 HINGE  *  1 Initial rotation angle (DEG) theta1,theta2,theta3 =  0.00000E+00  0.00000E+00  0.00000E+00
374.000 HINGE  *  1 Null torque angle (DEG)      theta1,theta2,theta3 =  0.00000E+00  0.00000E+00  0.00000E+00
375.000 HINGE  *  1 Initial rotation rate (D/S)  thdot1,thdot2,thdot3 =  0.00000E+00  0.00000E+00  0.00000E+00
376.000 HINGE  *  1 Translation degrees of freedom                    =   3.0000
377.000 HINGE  *  1 Translation axis unit vector of 1ST DOF   X,Y,Z   =   1.0000       0.00000E+00  0.00000E+00
378.000 HINGE  *  1 Translation axis unit vector of 2ND DOF   X,Y,Z   =  0.00000E+00   1.0000       0.00000E+00
379.000 HINGE  *  1 Translation axis unit vector of 3RD DOF   X,Y,Z   =  0.00000E+00  0.00000E+00   1.0000
380.000 HINGE  *  1 Translation spring stiffnes (N/M)  k1,k2,k3       =  0.00000E+00  0.00000E+00  0.00000E+00
381.000 HINGE  *  1 Translation spring damping (N/M/S) b1,b2,b3       =  0.00000E+00  0.00000E+00  0.00000E+00
382.000 HINGE  *  1 Initial translation displacement (M)  y1,y2,y3    =  0.00000E+00  0.00000E+00  0.00000E+00
383.000 HINGE  *  1 Null force position (M)               y1,y2,y3    =  0.00000E+00  0.00000E+00  0.00000E+00
384.000 HINGE  *  1 Initial translation velocity (M/S  yd1,yd2,yd3    =  0.00000E+00  0.00000E+00  0.00000E+00
385.000 HINGE  *  1 >>>>>***** END OF DATA *****<<<<<                 =
386.000 HINGE  *  2 ID# of hinge, ID# of inbd body, ID# of outbd body =   2.0000        1.0000       2.0000
387.000 HINGE  *  2 Node index of inboard body attach point           =   2.0000
388.000 HINGE  *  2 Rotation degrees of freedom                       =  0.00000E+00
389.000 HINGE  *  2 Base body(ID# 1)rotation option(F=free,G=gimbaled)=
390.000 HINGE  *  2 Rotation axis unit vector L1 inboard body  X,Y,Z  =   1.0000       0.00000E+00  0.00000E+00
391.000 HINGE  *  2 Rotation axis unit vector L1 outboard body X,Y,Z  =   1.0000       0.00000E+00  0.00000E+00
392.000 HINGE  *  2 Rotation axis unit vector L3 inboard body  X,Y,Z  =  0.00000E+00  0.00000E+00   1.0000
393.000 HINGE  *  2 Rotation axis unit vector L3 outboard body X,Y,Z  =  0.00000E+00  0.00000E+00   1.0000
394.000 HINGE  *  2 Rotation spring stiffness (N/R)    k1,k2,k3       =
395.000 HINGE  *  2 Rotation spring damping (N/R/S)    b1,b2,b3       =
396.000 HINGE  *  2 Initial rotation angle (DEG) theta1,theta2,theta3 =  0.00000E+00  0.00000E+00  0.00000E+00
397.000 HINGE  *  2 Null torque angle (DEG)      theta1,theta2,theta3 =
398.000 HINGE  *  2 Initial rotation rate (D/S)  thdot1,thdot2,thdot3 =
399.000 HINGE  *  2 Translation degrees of freedom                    =  0.00000E+00
400.000 HINGE  *  2 Translation axis unit vector of 1ST DOF   X,Y,Z   =
401.000 HINGE  *  2 Translation axis unit vector of 2ND DOF   X,Y,Z   =
402.000 HINGE  *  2 Translation axis unit vector of 3RD DOF   X,Y,Z   =
403.000 HINGE  *  2 Translation spring stiffnes (N/M)  k1,k2,k3       =
404.000 HINGE  *  2 Translation spring damping (N/M/S) b1,b2,b3       =
405.000 HINGE  *  2 Initial translation displacement (M)  y1,y2,y3    =
406.000 HINGE  *  2 Null force position (M)               y1,y2,y3    =
407.000 HINGE  *  2 Initial translation velocity (M/S  yd1,yd2,yd3    =
408.000 HINGE  *  2 >>>>>***** END OF DATA *****<<<<<                 =
409.000 HINGE  *  3 ID# of hinge, ID# of inbd body, ID# of outbd body =   3.0000        2.0000       3.0000
410.000 HINGE  *  3 Node index of inboard body attach point           =   2.0000
411.000 HINGE  *  3 Rotation degrees of freedom                       =   1.0000
412.000 HINGE  *  3 Base body(ID# 1)rotation option(F=free,G=gimbaled)=
413.000 HINGE  *  3 Rotation axis unit vector L1 inboard body  X,Y,Z  =  0.00000E+00   1.0000       0.00000E+00
414.000 HINGE  *  3 Rotation axis unit vector L1 outboard body X,Y,Z  =  0.00000E+00   1.0000       0.00000E+00
415.000 HINGE  *  3 Rotation axis unit vector L3 inboard body  X,Y,Z  =  0.00000E+00  0.00000E+00   1.0000
416.000 HINGE  *  3 Rotation axis unit vector L3 outboard body X,Y,Z  =   1.0000       0.00000E+00  0.00000E+00
417.000 HINGE  *  3 Rotation spring stiffness (N/R)    k1,k2,k3       =  0.00000E+00  0.00000E+00  0.00000E+00
418.000 HINGE  *  3 Rotation spring damping (N/R/S)    b1,b2,b3       =  0.00000E+00  0.00000E+00  0.00000E+00
419.000 HINGE  *  3 Initial rotation angle (DEG) theta1,theta2,theta3 =  -90.01        0.00000E+00  0.00000E+00
420.000 HINGE  *  3 Null torque angle (DEG)      theta1,theta2,theta3 =  0.00000E+00  0.00000E+00  0.00000E+00
421.000 HINGE  *  3 Initial rotation rate (D/S)  thdot1,thdot2,thdot3 =  0.00000E+00  0.00000E+00  0.00000E+00
422.000 HINGE  *  3 Translation degrees of freedom                    =  0.00000E+00
423.000 HINGE  *  3 Translation axis unit vector of 1ST DOF   X,Y,Z   =
424.000 HINGE  *  3 Translation axis unit vector of 2ND DOF   X,Y,Z   =
425.000 HINGE  *  3 Translation axis unit vector of 3RD DOF   X,Y,Z   =
426.000 HINGE  *  3 Translation spring stiffnes (N/M)  k1,k2,k3       =
427.000 HINGE  *  3 Translation spring damping (N/M/S) b1,b2,b3       =
```

```
428.000  HINGE   #  3 Initial translation displacement (M)  y1,y2,y3       =
429.000  HINGE   #  3 Null force position (M)               y1,y2,y3       =
430.000  HINGE   #  3 Initial translation velocity (M/S  yd1,yd2,yd3       =
431.000  HINGE   #  3 >>>>>***** END OF DATA *****<<<<<                    =
432.000  HINGE   #  4 ID# of hinge, ID# of inbd body, ID# of outbd body    =   4.0000        3.0000        4.0000
433.000  HINGE   #  4 Node index of inboard body attach point              =   2.0000
434.000  HINGE   #  4 Rotation degrees of freedom                          =   1.0000
435.000  HINGE   #  4 Base body(ID# 1)rotation option(F=free,G=gimbaled)=
436.000  HINGE   #  4 Rotation axis unit vector L1 inboard body  X,Y,Z     =   1.0000        0.00000E+00   0.00000E+00
437.000  HINGE   #  4 Rotation axis unit vector L1 outboard body X,Y,Z     =   1.0000        0.00000E+00   0.00000E+00
438.000  HINGE   #  4 Rotation axis unit vector L3 inboard body  X,Y,Z     =   0.00000E+00   0.00000E+00   1.0000
439.000  HINGE   #  4 Rotation axis unit vector L3 outboard body X,Y,Z     =   0.00000E+00   0.00000E+00   1.0000
440.000  HINGE   #  4 Rotation spring stiffness (N/R)   k1,k2,k3           =   0.00000E+00   0.00000E+00   0.00000E+00
441.000  HINGE   #  4 Rotation spring damping (N/R/S)   b1,b2,b3           =   0.00000E+00   0.00000E+00   0.00000E+00
442.000  HINGE   #  4 Initial rotation angle (DEG) theta1,theta2,theta3    =   0.10000E-01   0.00000E+00   0.00000E+00
443.000  HINGE   #  4 Null torque angle (DEG)      theta1,theta2,theta3    =   0.00000E+00   0.00000E+00   0.00000E+00
444.000  HINGE   #  4 Initial rotation rate (D/S)  thdot1,thdot2,thdot3    =   0.00000E+00   0.00000E+00   0.00000E+00
445.000  HINGE   #  4 Translation degrees of freedom                       =   0.00000E+00
446.000  HINGE   #  4 Translation axis unit vector of 1ST DOF  X,Y,Z       =
447.000  HINGE   #  4 Translation axis unit vector of 2ND DOF  X,Y,Z       =
448.000  HINGE   #  4 Translation axis unit vector of 3RD DOF  X,Y,Z       =
449.000  HINGE   #  4 Translation spring stiffnes (N/M)  k1,k2,k3          =
450.000  HINGE   #  4 Translation spring damping (N/M/S) b1,b2,b3          =
451.000  HINGE   #  4 Initial translation displacement (M)  y1,y2,y3       =
452.000  HINGE   #  4 Null force position (M)               y1,y2,y3       =
453.000  HINGE   #  4 Initial translation velocity (M/S  yd1,yd2,yd3       =
454.000  HINGE   #  4 >>>>>***** END OF DATA *****<<<<<                    =
455.000  HINGE   #  5 ID# of hinge, ID# of inbd body, ID# of outbd body    =   5.0000        4.0000        5.0000
456.000  HINGE   #  5 Node index of inboard body attach point              =   2.0000
457.000  HINGE   #  5 Rotation degrees of freedom                          =   1.0000
458.000  HINGE   #  5 Base body(ID# 1)rotation option(F=free,G=gimbaled)=
459.000  HINGE   #  5 Rotation axis unit vector L1 inboard body  X,Y,Z     =   0.00000E+00   0.00000E+00   1.0000
460.000  HINGE   #  5 Rotation axis unit vector L1 outboard body X,Y,Z     =   0.00000E+00   0.00000E+00   1.0000
461.000  HINGE   #  5 Rotation axis unit vector L3 inboard body  X,Y,Z     =   0.00000E+00   1.0000        0.00000E+00
462.000  HINGE   #  5 Rotation axis unit vector L3 outboard body X,Y,Z     =   0.00000E+00   1.0000        0.00000E+00
463.000  HINGE   #  5 Rotation spring stiffness (N/R)   k1,k2,k3           =   0.00000E+00   0.00000E+00   0.00000E+00
464.000  HINGE   #  5 Rotation spring damping (N/R/S)   b1,b2,b3           =   0.00000E+00
465.000  HINGE   #  5 Initial rotation angle (DEG) theta1,theta2,theta3    =   0.10000E-01   0.00000E+00   0.00000E+00
466.000  HINGE   #  5 Null torque angle (DEG)      theta1,theta2,theta3    =   0.00000E+00
467.000  HINGE   #  5 Initial rotation rate (D/S)  thdot1,thdot2,thdot3    =   0.00000E+00
468.000  HINGE   #  5 Translation degrees of freedom                       =   0.00000E+00
469.000  HINGE   #  5 Translation axis unit vector of 1ST DOF  X,Y,Z       =
470.000  HINGE   #  5 Translation axis unit vector of 2ND DOF  X,Y,Z       =
471.000  HINGE   #  5 Translation axis unit vector of 3RD DOF  X,Y,Z       =
472.000  HINGE   #  5 Translation spring stiffnes (N/M)  k1,k2,k3          =
473.000  HINGE   #  5 Translation spring damping (N/M/S) b1,b2,b3          =
474.000  HINGE   #  5 Initial translation displacement (M)  y1,y2,y3       =
475.000  HINGE   #  5 Null force position (M)               y1,y2,y3       =
476.000  HINGE   #  5 Initial translation velocity (M/S  yd1,yd2,yd3       =
477.000  HINGE   #  5 >>>>>***** END OF DATA *****<<<<<                    =
478.000  HINGE   #  6 ID# of hinge, ID# of inbd body, ID# of outbd body    =   6.0000        5.0000        6.0000
479.000  HINGE   #  6 Node index of inboard body attach point              =   2.0000
480.000  HINGE   #  6 Rotation degrees of freedom                          =   0.00000E+00
481.000  HINGE   #  6 Base body(ID# 1)rotation option(F=free,G=gimbaled)=
482.000  HINGE   #  6 Rotation axis unit vector L1 inboard body  X,Y,Z     =   1.0000        0.00000E+00   0.00000E+00
483.000  HINGE   #  6 Rotation axis unit vector L1 outboard body X,Y,Z     =   1.0000        0.00000E+00   0.00000E+00
484.000  HINGE   #  6 Rotation axis unit vector L3 inboard body  X,Y,Z     =   0.00000E+00   0.00000E+00   1.0000
485.000  HINGE   #  6 Rotation axis unit vector L3 outboard body X,Y,Z     =   0.00000E+00   0.00000E+00   1.0000
486.000  HINGE   #  6 Rotation spring stiffness (N/R)   k1,k2,k3           =
487.000  HINGE   #  6 Rotation spring damping (N/R/S)   b1,b2,b3           =
488.000  HINGE   #  6 Initial rotation angle (DEG) theta1,theta2,theta3    =   0.00000E+00
```

```
489.000  HINGE   #  6 Null torque angle (DEG)       theta1,theta2,theta3 =
490.000  HINGE   #  6 Initial rotation rate (D/S)  thdot1,thdot2,thdot3 =
491.000  HINGE   #  6 Translation degrees of freedom                     =  0.00000E+00
492.000  HINGE   #  6 Translation axis unit vector of 1ST DOF  X,Y,Z     =
493.000  HINGE   #  6 Translation axis unit vector of 2ND DOF  X,Y,Z     =
494.000  HINGE   #  6 Translation axis unit vector of 3RD DOF  X,Y,Z     =
495.000  HINGE   #  6 Translation spring stiffnes (N/M)  k1,k2,k3        =
496.000  HINGE   #  6 Translation spring damping (N/M/S) b1,b2,b3        =
497.000  HINGE   #  6 Initial translation displacement (M)  y1,y2,y3     =
498.000  HINGE   #  6 Null force position (M)                y1,y2,y3    =
499.000  HINGE   #  6 Initial translation velocity (M/S  yd1,yd2,yd3     =
500.000  HINGE   #  6 >>>>>***** END OF DATA *****<<<<<                  =
501.000  BODY    #  1 Type of body (R=rigid, F=flexible),ID#             = R             1.0000
502.000  BODY    #  1 Mass of body (kg)                                  =  0.10344E+06
503.000  BODY    #  1 Inertia (kg-m2) Ixx,Iyy,Izz                        =  0.13951E+07  0.10690E+08  0.11064E+08
504.000  BODY    #  1 Inertia (kg-m2) Ixy,Ixz,Iyz                        =  -3177.1     -0.46457E+06   262.90
505.000  BODY    #  1 modal data option (T=tape,D=disc),fname and tape#  =
506.000  BODY    #  1 Modal inertia opt (N=none,I=inertia M,N,P dyadics) =
507.000  BODY    #  1 Modal coupling opt (N=none,C=coupling PHIXPHI vec) =
508.000  BODY    #  1 Number of flexible modes                           =
509.000  BODY    #  1 Attach point coordinates (m) x,y,z                 =  0.00000E+00  0.00000E+00  0.00000E+00
510.000  BODY    #  1 NODE    1 mass center coordinates (m) x,y,z        =  0.00000E+00  0.00000E+00  0.00000E+00
511.000  BODY    #  1 NODE    2 coordinates (m) x,y,z                    =   4.2090     -0.11350      0.20100
512.000  BODY    #  1 NODE    3 coordinates (m) x,y,z                    =   19.340       1.5200      0.61000
513.000  BODY    #  1 NODE    4 coordinates (m) x,y,z                    =   19.340      -1.5200      0.61000
514.000  BODY    #  1 NODE    5 coordinates (m) x,y,z                    =  -12.170       3.8100     -2.1600
515.000  BODY    #  1 NODE    6 coordinates (m) x,y,z                    =  -12.170      -3.8100     -2.1600
516.000  BODY    #  1 NODE    7 coordinates (m) x,y,z                    =  -12.170       3.0000     -2.0700
517.000  BODY    #  1 NODE    8 coordinates (m) x,y,z                    =  -12.170      -3.0000     -2.0700
518.000  BODY    #  1 NODE    9 coordinates (m) x,y,z                    =   13.996      0.00000E+00 -1.8580
519.000  BODY    #  1 >>>>>***** END OF DATA *****<<<<<                  =
520.000  BODY    #  1 >>>>>***** END OF DATA *****<<<<<                  =
521.000  BODY    #  2 Type of body (R=rigid, F=flexible),ID#             = R             2.0000
522.000  BODY    #  2 Mass of body (kg)                                  =   4358.0
523.000  BODY    #  2 Inertia (kg-m2) Ixx,Iyy,Izz                        =   7307.0       19805.       23648.
524.000  BODY    #  2 Inertia (kg-m2) Ixy,Ixz,Iyz                        =   645.20      -669.60      -146.40
525.000  BODY    #  2 modal data option (T=tape,D=disc),fname and tape#  =
526.000  BODY    #  2 Modal inertia opt (N=none,I=inertia M,N,P dyadics) =
527.000  BODY    #  2 Modal coupling opt (N=none,C=coupling PHIXPHI vec) =
528.000  BODY    #  2 Number of flexible modes                           =
529.000  BODY    #  2 Attach point coordinates (m) x,y,z                 =  0.00000E+00  0.00000E+00  0.00000E+00
530.000  BODY    #  2 NODE    1 mass center coordinates (m) x,y,z        =  0.00000E+00  0.00000E+00  0.00000E+00
531.000  BODY    #  2 NODE    2 coordinates (m) x,y,z                    =   1.1250      0.10460     -1.0750
532.000  BODY    #  2 NODE    3 coordinates (m) x,y,z                    =  0.85800      0.58460     -0.27000
533.000  BODY    #  2 >>>>>***** END OF DATA *****<<<<<                  =
534.000  BODY    #  2 >>>>>***** END OF DATA *****<<<<<                  =
535.000  BODY    #  3 Type of body (R=rigid, F=flexible),ID#             = R             3.0000
536.000  BODY    #  3 Mass of body (kg)                                  =   84.400
537.000  BODY    #  3 Inertia (kg-m2) Ixx,Iyy,Izz                        =   1.3554       5.4216       5.4216
538.000  BODY    #  3 Inertia (kg-m2) Ixy,Ixz,Iyz                        =  0.00000E+00  -8.1324      0.00000E+00
539.000  BODY    #  3 modal data option (T=tape,D=disc),fname and tape#  =
540.000  BODY    #  3 Modal inertia opt (N=none,I=inertia M,N,P dyadics) =
541.000  BODY    #  3 Modal coupling opt (N=none,C=coupling PHIXPHI vec) =
542.000  BODY    #  3 Number of flexible modes                           =
543.000  BODY    #  3 Attach point coordinates (m) x,y,z                 = -0.14700     -0.68600E-02 -0.30000E-02
544.000  BODY    #  3 NODE    1 mass center coordinates (m) x,y,z        =  0.00000E+00  0.00000E+00  0.00000E+00
545.000  BODY    #  3 NODE    2 coordinates (m) x,y,z                    =  0.33600     -0.68600E-02 -0.30000E-02
546.000  BODY    #  3 >>>>>***** END OF DATA *****<<<<<                  =
547.000  BODY    #  3 >>>>>***** END OF DATA *****<<<<<                  =
548.000  BODY    #  4 Type of body (R=rigid, F=flexible),ID#             = R             4.0000
549.000  BODY    #  4 Mass of body (kg)                                  =   105.20
```

```
550.000  BODY   *  4 Inertia (kg-m2) Ixx,Iyy,Izz                          =   9.4878        16.265         8.1324
551.000  BODY   *  4 Inertia (kg-m2) Ixy,Ixz,Iyz                          =   0.00000E+00  -28.463        -1.3554
552.000  BODY   *  4 modal data option (T=tape,D=disc),fname and tape#    =
553.000  BODY   *  4 Modal inertia opt (N=none,I=inertia M,N,P dyadics)=
554.000  BODY   *  4 Modal coupling opt (N=none,C=coupling PHIXPHI vec)=
555.000  BODY   *  4 Number of flexible modes                             =
556.000  BODY   *  4 Attach point coordinates (m) x,y,z                   =   0.19100       0.10200E-02  -0.31100
557.000  BODY   *  4 NODE    1 mass center coordinates (m) x,y,z          =   0.00000E+00   0.00000E+00   0.00000E+00
558.000  BODY   *  4 NODE    2 coordinates (m) x,y,z                      =  -0.29200       0.10200E-02   0.30000
559.000  BODY   *  4 >>>>>***** END OF DATA *****<<<<<                    =
560.000  BODY   *  4 >>>>>***** END OF DATA *****<<<<<                    =
561.000  BODY   *  5 Type of body (R=rigid, F=flexible),ID#               =  R              5.0000
562.000  BODY   *  5 Mass of body (kg)                                    =   1875.0
563.000  BODY   *  5 Inertia (kg-m2) Ixx,Iyy,Izz                          =   12691.        12354.        1903.0
564.000  BODY   *  5 Inertia (kg-m2) Ixy,Ixz,Iyz                          =   0.00000E+00   0.00000E+00   0.00000E+00
565.000  BODY   *  5 modal data option (T=tape,D=disc),fname and tape#    =
566.000  BODY   *  5 Modal inertia opt (N=none,I=inertia M,N,P dyadics)=
567.000  BODY   *  5 Modal coupling opt (N=none,C=coupling PHIXPHI vec)=
568.000  BODY   *  5 Number of flexible modes                             =
569.000  BODY   *  5 Attach point coordinates (m) x,y,z                   =   0.00000E+00   0.00000E+00   0.00000E+00
570.000  BODY   *  5 NODE    1 mass center coordinates (m) x,y,z          =   0.00000E+00   0.00000E+00   0.57610
571.000  BODY   *  5 NODE    2 coordinates (m) x,y,z                      =   0.00000E+00   0.00000E+00   4.3198
572.000  BODY   *  5 NODE    3 coordinates (m) x,y,z                      =   0.00000E+00   0.00000E+00   1.0000
573.000  BODY   *  5 >>>>>***** END OF DATA *****<<<<<                    =
574.000  BODY   *  5 >>>>>***** END OF DATA *****<<<<<                    =
575.000  BODY   *  6 Type of body (R=rigid, F=flexible),ID#               =  F              6.0000
576.000  BODY   *  6 Mass of body (kg)                                    =   69.870
577.000  BODY   *  6 Inertia (kg-m2) Ixx,Iyy,Izz                          =   65391.        65416.        81.000
578.000  BODY   *  6 Inertia (kg-m2) Ixy,Ixz,Iyz                          =   0.00000E+00   0.00000E+00   0.00000E+00
579.000  BODY   *  6 modal data option (T=tape,D=disc),fname and tape#    =  D             PINHOLE.FLX
580.000  BODY   *  6 Modal inertia opt (N=none,I=inertia M,N,P dyadics)=  N
581.000  BODY   *  6 Modal coupling opt (N=none,C=coupling PHIXPHI vec)=  N
582.000  BODY   *  6 Number of flexible modes                             =   8.0000
583.000  BODY   *  6 Attach point coordinates (m) x,y,z                   =   0.00000E+00   0.00000E+00   0.00000E+00
584.000  BODY   *  6 NODE    1 mass center coordinates (m) x,y,z          =   0.00000E+00   0.00000E+00   29.920
585.000  BODY   *  6 NODE    2 coordinates (m) x,y,z                      =   0.00000E+00   0.00000E+00   32.000
586.000  BODY   *  6 >>>>>***** END OF DATA *****<<<<<                    =
587.000  BODY   *  6 >>>>>***** END OF DATA *****<<<<<                    =
```

GSSMP3.INT

```
 1.000  SIM CONTROL  *  0 Title header of user problem (< 40 char)           = GSSMP3      :GENERIC SPACE STATION MODEL
 2.000  SIM CONTROL  *  0 Simulation stop time(sec),Job time(min),Core(kwds)=   20.000
 3.000  SIM CONTROL  *  0 Data output delta, Initial, Final time (sec)      =   0.20000     0.00000E+00   20.000
 4.000  SIM CONTROL  *  0 Integration(R=rkuta,S=sandia,U=user),Dt(sec),Fname= R             0.25000E-01
 5.000  SIM CONTROL  *  0 Linearization option (N=none,L=linear), Time(sec) = N
 6.000  SIM CONTROL  *  0 Restart option (N=none,R=restart), Restart fname   = N
 7.000  SPEEDUP      *  0 Small angle computation(All,Bypass,First,Nth) pass= FIRST
 8.000  SPEEDUP      *  0 Mass matrix computation(All,Bypass,First,Nth) pass= BYPASS
 9.000  SPEEDUP      *  0 Non-linear computation(All,Bypass,First,Nth) pass = BYPASS
10.000  CONTROLLER   C  3 Type(C=contin,D=discrete,U=user), Dt(sec), Fname  = CONTINUOUS
11.000  CONTROLLER   C  3 Number of inputs, Number of outputs               =   3.0000      3.0000
12.000  CONTROLLER   C  3 OUTPUT  1 type(I=incont,T=trans,J=junct), ID#,Gain= T             1.0000        10.954
13.000  CONTROLLER   C  3 OUTPUT  2 Type(I=incont,T=trans,J=junct), ID#,Gain= T             2.0000        10.954
14.000  CONTROLLER   C  3 OUTPUT  3 Type(I=incont,T=trans,J=junct), ID#,Gain= T             3.0000        10.954
15.000  TRANS FUNCT  C  1 Input type(I=incont,T=trans,J=junct), Input ID#   = I             1.0000        CONTINUOUS
16.000  TRANS FUNCT  C  1 Order of numerator, Order of denominator          =   1.0000      2.0000
17.000  TRANS FUNCT  C  1 Numerator coeff (ascending order 1-4 per line)    =  0.00000E+00  56.210
18.000  TRANS FUNCT  C  1 Denominator coeff (ascending order 1-4 per line)  =   5.7120      6.6400        1.0000
19.000  TRANS FUNCT  C  2 Input type(I=incont,T=trans,J=junct), Input ID#   = I             2.0000        CONTINUOUS
20.000  TRANS FUNCT  C  2 Order of numerator, Order of denominator          =   1.0000      2.0000
21.000  TRANS FUNCT  C  2 Numerator coeff (ascending order 1-4 per line)    =  0.00000E+00  56.210
22.000  TRANS FUNCT  C  2 Denominator coeff (ascending order 1-4 per line)  =   5.7120      6.6400        1.0000
23.000  TRANS FUNCT  C  3 Input type(I=incont,T=trans,J=junct), Input ID#   = I             3.0000        CONTINUOUS
24.000  TRANS FUNCT  C  3 Order of numerator, Order of denominator          =   1.0000      2.0000
25.000  TRANS FUNCT  C  3 Numerator coeff (ascending order 1-4 per line)    =  0.00000E+00  56.210
26.000  TRANS FUNCT  C  3 Denominator coeff (ascending order 1-4 per line)  =   5.7120      6.6400        1.0000
27.000  TRANS FUNCT  C  3 >>>>>***** END OF DATA *****<<<<<                 =
28.000  SUM JUNCT    C  3 >>>>>***** END OF DATA *****<<<<<                 =
29.000  CONTROLLER   U  2 Type(C=contin,D=discrete,U=user), Dt(sec), Fname  = USER          0.25000E-01 SSPOF3
30.000  CONTROLLER   U  2 Number of inputs, Number of outputs               =   13.000      20.000
31.000  FUNCT GENER  *  1 type(step,ramp,pulse,sawtooth,sine,user)          = PULSE
32.000  FUNCT GENER  *  1 pulse amplitude, start time, stop time(sec)       =   111.25      0.00000E+00   1.0000
33.000  FUNCT GENER  *  1 >>>>>***** END OF DATA *****<<<<<                 =
34.000  FUNCT GENER  *  2 type(step,ramp,pulse,sawtooth,sine,user)          = PULSE
35.000  FUNCT GENER  *  2 pulse amplitude, start time, stop time(sec)       =  -111.25      0.00000E+00   1.0000
36.000  FUNCT GENER  *  2 >>>>>***** END OF DATA *****<<<<<                 =
37.000  FUNCT GENER  *  3 type(step,ramp,pulse,sawtooth,sine,user)          = USER
38.000  FUNCT GENER  *  3 user fname(use same name for additional usergener)= DIST3
39.000  FUNCT GENER  *  3 >>>>>***** END OF DATA *****<<<<<                 =
40.000  FUNCT GENER  *  4 type(step,ramp,pulse,sawtooth,sine,user)          = USER
41.000  FUNCT GENER  *  4 user fname(use same name for additional usergener)= DIST3
42.000  FUNCT GENER  *  4 >>>>>***** END OF DATA *****<<<<<                 =
43.000  FUNCT GENER  *  5 type(step,ramp,pulse,sawtooth,sine,user)          = USER
44.000  FUNCT GENER  *  5 user fname(use same name for additional usergener)= DIST3
45.000  FUNCT GENER  *  5 >>>>>***** END OF DATA *****<<<<<                 =
46.000  SENSOR       *  3 Type(Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos), ID#  = GYRO          3.0000
47.000  SENSOR       *  3 ID# of body, index of node location               =   2.0000      3.0000
48.000  SENSOR       *  3 Input axis unit vector (body coord) X,Y,Z         =   1.0000      0.00000E+00  0.00000E+00
49.000  SENSOR       *  3 >>>>>***** END OF DATA *****<<<<<                 =
50.000  SENSOR       *  4 Type(Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos), ID#  = GYRO          4.0000
51.000  SENSOR       *  4 ID# of body, index of node location               =   2.0000      3.0000
52.000  SENSOR       *  4 Input axis unit vector (body coord) X,Y,Z         =  0.00000E+00  1.0000        0.00000E+00
53.000  SENSOR       *  4 >>>>>***** END OF DATA *****<<<<<                 =
54.000  SENSOR       *  5 Type(Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos), ID#  = GYRO          5.0000
55.000  SENSOR       *  5 ID# of body, index of node location               =   2.0000      3.0000
56.000  SENSOR       *  5 Input axis unit vector (body coord) X,Y,Z         =  0.00000E+00  0.00000E+00   1.0000
57.000  SENSOR       *  5 >>>>>***** END OF DATA *****<<<<<                 =
58.000  SENSOR       *  6 Type Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos,Los,ID#= ACCEL         6.0000
59.000  SENSOR       *  6 ID# of body, index of node location               =   1.0000      3.0000
60.000  SENSOR       *  6 Input axis unit vector (body coord) X,Y,Z         =   1.0000      0.00000E+00  0.00000E+00
61.000  SENSOR       *  6 >>>>>***** END OF DATA *****<<<<<                 =
```

```
 62.000  SENSOR      *  7 Type Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos,Los,ID#=  A              7.0000
 63.000  SENSOR      *  7 ID# of body, index of node location              =   1.0000       3.0000
 64.000  SENSOR      *  7 Input axis unit vector (body coord) X,Y,Z        =  0.00000E+00   1.0000       0.00000E+00
 65.000  SENSOR      *  7 >>>>>***** END OF DATA *****<<<<<                =
 66.000  SENSOR      *  8 Type Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos,Los,ID#=  A              8.0000
 67.000  SENSOR      *  8 ID# of body, index of node location              =   1.0000       3.0000
 68.000  SENSOR      *  8 Input axis unit vector (body coord) X,Y,Z        =  0.00000E+00   0.00000E+00   1.0000
 69.000  SENSOR      *  8 >>>>>***** END OF DATA *****<<<<<                =
 70.000  SENSOR      *  9 Type Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos,Los,ID#=  WDOT           9.0000
 71.000  SENSOR      *  9 ID# of body, index of node location              =   1.0000       3.0000
 72.000  SENSOR      *  9 Input axis unit vector (body coord) X,Y,Z        =   1.0000       0.00000E+00   0.00000E+00
 73.000  SENSOR      *  9 >>>>>***** END OF DATA *****<<<<<                =
 74.000  SENSOR      * 10 Type Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos,Los,ID#=  W              10.000
 75.000  SENSOR      * 10 ID# of body, index of node location              =   1.0000       3.0000
 76.000  SENSOR      * 10 Input axis unit vector (body coord) X,Y,Z        =  0.00000E+00   1.0000       0.00000E+00
 77.000  SENSOR      * 10 >>>>>***** END OF DATA *****<<<<<                =
 78.000  SENSOR      * 11 Type Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos,Los,ID#=  W              11.000
 79.000  SENSOR      * 11 ID# of body, index of node location              =   1.0000       3.0000
 80.000  SENSOR      * 11 Input axis unit vector (body coord) X,Y,Z        =  0.00000E+00   0.00000E+00   1.0000
 81.000  SENSOR      * 11 >>>>>***** END OF DATA *****<<<<<                =
 82.000  SENSOR      * 99 Type Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos,Los,ID#=  LOS            99.000
 83.000  SENSOR      * 99 ID# of body, index of node location              =   3.0000       2.0000
 84.000  SENSOR      * 99 Pinhole target unit vector (inertial coord) X,Y,Z =  0.00000E+00   0.00000E+00   1.0000
 85.000  SENSOR      * 99 >>>>>***** END OF DATA *****<<<<<                =
 86.000  ACTUATOR    *  1 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  T              1.0000
 87.000  ACTUATOR    *  1 ID# of hinge, index of rotation axis             =   2.0000       3.0000
 88.000  ACTUATOR    *  1 >>>>>***** END OF DATA *****<<<<<                =
 89.000  ACTUATOR    *  2 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  T              2.0000
 90.000  ACTUATOR    *  2 ID# of hinge, index of rotation axis             =   2.0000       2.0000
 91.000  ACTUATOR    *  2 >>>>>***** END OF DATA *****<<<<<                =
 92.000  ACTUATOR    *  3 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  T              3.0000
 93.000  ACTUATOR    *  3 ID# of hinge, index of rotation axis             =   2.0000       1.0000
 94.000  ACTUATOR    *  3 >>>>>***** END OF DATA *****<<<<<                =
 95.000  ACTUATOR    *  4 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  J              4.0000
 96.000  ACTUATOR    *  4 ID# of body, index of node location              =   1.0000       3.0000
 97.000  ACTUATOR    *  4 Output axis unit vector(body coord) X,Y,Z        =   1.0000       0.00000E+00   0.00000E+00
 98.000  ACTUATOR    *  4 >>>>>***** END OF DATA *****<<<<<                =
 99.000  ACTUATOR    * 10 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  J              10.000
100.000  ACTUATOR    * 10 ID# of body, index of node location              =   1.0000       2.0000
101.000  ACTUATOR    * 10 Output axis unit vector(body coord) X,Y,Z        =   1.0000       0.00000E+00   0.00000E+00
102.000  ACTUATOR    * 10 >>>>>***** END OF DATA *****<<<<<                =
103.000  ACTUATOR    * 11 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  J              11.000
104.000  ACTUATOR    * 11 ID# of body, index of node location              =   1.0000       4.0000
105.000  ACTUATOR    * 11 Output axis unit vector(body coord) X,Y,Z        =   1.0000       0.00000E+00   0.00000E+00
106.000  ACTUATOR    * 11 >>>>>***** END OF DATA *****<<<<<                =
107.000  ACTUATOR    * 12 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  J              12.000
108.000  ACTUATOR    * 12 ID# of body, index of node location              =   1.0000       5.0000
109.000  ACTUATOR    * 12 Output axis unit vector(body coord) X,Y,Z        =   1.0000       0.00000E+00   0.00000E+00
110.000  ACTUATOR    * 12 >>>>>***** END OF DATA *****<<<<<                =
111.000  ACTUATOR    * 13 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  J              13.000
112.000  ACTUATOR    * 13 ID# of body, index of node location              =   1.0000       6.0000
113.000  ACTUATOR    * 13 Output axis unit vector(body coord) X,Y,Z        =   1.0000       0.00000E+00   0.00000E+00
114.000  ACTUATOR    * 13 >>>>>***** END OF DATA *****<<<<<                =
115.000  ACTUATOR    * 14 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  J              14.000
116.000  ACTUATOR    * 14 ID# of body, index of node location              =   1.0000       7.0000
117.000  ACTUATOR    * 14 Output axis unit vector(body coord) X,Y,Z        =   1.0000       0.00000E+00   0.00000E+00
118.000  ACTUATOR    * 14 >>>>>***** END OF DATA *****<<<<<                =
119.000  INT CONNECT *  1 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#  =  U              1.0000
120.000  INT CONNECT *  1 Destination(C=contin,D=disc,U=user,A=actuator),ID#= A              2.0000
121.000  INT CONNECT *  1 >>>>>***** END OF DATA *****<<<<<                =
122.000  INT CONNECT *  2 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#  =  U              2.0000
```

```
123.000  INT CONNECT *   2 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  A          3.0000
124.000  INT CONNECT *   2 >>>>>***** END OF DATA *****<<<<<                 =
125.000  INT CONNECT *   3 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  U          3.0000
126.000  INT CONNECT *   3 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  A          1.0000
127.000  INT CONNECT *   3 >>>>>***** END OF DATA *****<<<<<                 =
128.000  INT CONNECT *   4 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  S          4.0000
129.000  INT CONNECT *   4 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U          1.0000
130.000  INT CONNECT *   4 >>>>>***** END OF DATA *****<<<<<                 =
131.000  INT CONNECT *   5 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  S          992.00
132.000  INT CONNECT *   5 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U          2.0000
133.000  INT CONNECT *   5 >>>>>***** END OF DATA *****<<<<<                 =
134.000  INT CONNECT *   6 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  S          994.00
135.000  INT CONNECT *   6 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U          3.0000
136.000  INT CONNECT *   6 >>>>>***** END OF DATA *****<<<<<                 =
137.000  INT CONNECT *   7 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  S          997.00
138.000  INT CONNECT *   7 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U          4.0000
139.000  INT CONNECT *   7 >>>>>***** END OF DATA *****<<<<<                 =
140.000  INT CONNECT *   8 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  S          3.0000
141.000  INT CONNECT *   8 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U          5.0000
142.000  INT CONNECT *   8 >>>>>***** END OF DATA *****<<<<<                 =
143.000  INT CONNECT *   9 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  S          991.00
144.000  INT CONNECT *   9 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U          6.0000
145.000  INT CONNECT *   9 >>>>>***** END OF DATA *****<<<<<                 =
146.000  INT CONNECT *  10 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  S          995.00
147.000  INT CONNECT *  10 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U          7.0000
148.000  INT CONNECT *  10 >>>>>***** END OF DATA *****<<<<<                 =
149.000  INT CONNECT *  11 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  S          996.00
150.000  INT CONNECT *  11 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U          8.0000
151.000  INT CONNECT *  11 >>>>>***** END OF DATA *****<<<<<                 =
152.000  INT CONNECT *  12 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  S          5.0000
153.000  INT CONNECT *  12 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U          9.0000
154.000  INT CONNECT *  12 >>>>>***** END OF DATA *****<<<<<                 =
155.000  INT CONNECT *  13 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  S          993.00
156.000  INT CONNECT *  13 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U          10.000
157.000  INT CONNECT *  13 >>>>>***** END OF DATA *****<<<<<                 =
158.000  INT CONNECT *  14 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  G          3.0000
159.000  INT CONNECT *  14 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C          1.0000
160.000  INT CONNECT *  14 >>>>>***** END OF DATA *****<<<<<                 =
161.000  INT CONNECT *  15 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  G          4.0000
162.000  INT CONNECT *  15 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C          2.0000
163.000  INT CONNECT *  15 >>>>>***** END OF DATA *****<<<<<                 =
164.000  INT CONNECT *  16 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  G          5.0000
165.000  INT CONNECT *  16 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C          3.0000
166.000  INT CONNECT *  16 >>>>>***** END OF DATA *****<<<<<                 =
167.000  INT CONNECT *  17 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  C          1.0000
168.000  INT CONNECT *  17 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  A          10.000
169.000  INT CONNECT *  17 >>>>>***** END OF DATA *****<<<<<                 =
170.000  INT CONNECT *  18 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  C          2.0000
171.000  INT CONNECT *  18 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  A          11.000
172.000  INT CONNECT *  18 >>>>>***** END OF DATA *****<<<<<                 =
173.000  INT CONNECT *  19 Source(C=contin,D=disc,U=user,S=sen,G=gen),  ID#   =  C          3.0000
174.000  INT CONNECT *  19 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  A          12.000
175.000  INT CONNECT *  19 >>>>>***** END OF DATA *****<<<<<                 =
176.000  HINGE       *   1 ID# of hinge, ID# of inbd body, ID# of outbd body =   1.0000       0.00000E+00   1.0000
177.000  HINGE       *   1 Node index of inboard body attach point            =
178.000  HINGE       *   1 Rotation degrees of freedom                        =   3.0000
179.000  HINGE       *   1 Base body(ID# 1)rotation option(F=free,G=gimbaled)=  G
180.000  HINGE       *   1 Rotation axis unit vector L1 inboard body  X,Y,Z  =   1.0000       0.00000E+00   0.00000E+00
181.000  HINGE       *   1 Rotation axis unit vector L1 outboard body X,Y,Z  =   1.0000       0.00000E+00   0.00000E+00
182.000  HINGE       *   1 Rotation axis unit vector L3 inboard body  X,Y,Z  =  0.00000E+00   0.00000E+00   1.0000
183.000  HINGE       *   1 Rotation axis unit vector L3 outboard body X,Y,Z  =  0.00000E+00   0.00000E+00  -1.0000
```

```
184.000  HINGE    *  1 Rotation spring stiffness (N/R)   k1,k2,k3           =  0.00000E+00  0.00000E+00  0.00000E+00
185.000  HINGE    *  1 Rotation spring damping (N/R/S)   b1,b2,b3           =  0.00000E+00  0.00000E+00  0.00000E+00
186.000  HINGE    *  1 Initial rotation angle (DEG) theta1,theta2,theta3   =  0.00000E+00  0.00000E+00  0.00000E+00
187.000  HINGE    *  1 Null torque angle (DEG)      theta1,theta2,theta3   =  0.00000E+00  0.00000E+00  0.00000E+00
188.000  HINGE    *  1 Initial rotation rate (D/S)  thdot1,thdot2,thdot3   =  0.00000E+00  0.00000E+00  0.00000E+00
189.000  HINGE    *  1 Translation degrees of freedom                      =   3.0000
190.000  HINGE    *  1 Translation axis unit vector of 1ST DOF  X,Y,Z      =   1.0000      0.00000E+00  0.00000E+00
191.000  HINGE    *  1 Translation axis unit vector of 2ND DOF  X,Y,Z      =  0.00000E+00  -1.0000      0.00000E+00
192.000  HINGE    *  1 Translation axis unit vector of 3RD DOF  X,Y,Z      =  0.00000E+00  0.00000E+00  -1.0000
193.000  HINGE    *  1 Translation spring stiffnes (N/M)  k1,k2,k3        =  0.00000E+00  0.00000E+00  0.00000E+00
194.000  HINGE    *  1 Translation spring damping (N/M/S) b1,b2,b3        =  0.00000E+00  0.00000E+00  0.00000E+00
195.000  HINGE    *  1 Initial translation displacement (M)  y1,y2,y3     =  0.00000E+00  0.00000E+00  0.00000E+00
196.000  HINGE    *  1 Null force position (M)               y1,y2,y3     =  0.00000E+00  0.00000E+00  0.00000E+00
197.000  HINGE    *  1 Initial translation velocity (M/S  yd1,yd2,yd3     =  0.00000E+00  0.00000E+00  0.00000E+00
198.000  HINGE    *  1 >>>>>***** END OF DATA *****<<<<<                  =
199.000  HINGE    *  2 ID# of hinge, ID# of inbd body, ID# of outbd body  =   2.0000       1.0000       2.0000
200.000  HINGE    *  2 Node index of inboard body attach point            =   3.0000
201.000  HINGE    *  2 Rotation degrees of freedom                        =   3.0000
202.000  HINGE    *  2 Base body(ID# 1)rotation option(F=free,G=gimbaled)=
203.000  HINGE    *  2 Rotation axis unit vector L1 inboard body  X,Y,Z   =   1.0000      0.00000E+00  0.00000E+00
204.000  HINGE    *  2 Rotation axis unit vector L1 outboard body X,Y,Z   =   1.0000      0.00000E+00  0.00000E+00
205.000  HINGE    *  2 Rotation axis unit vector L3 inboard body  X,Y,Z   =  0.00000E+00  0.00000E+00  -1.0000
206.000  HINGE    *  2 Rotation axis unit vector L3 outboard body X,Y,Z   =  0.00000E+00  0.00000E+00   1.0000
207.000  HINGE    *  2 Rotation spring stiffness (N/R)   k1,k2,k3          =  0.00000E+00  0.00000E+00  0.00000E+00
208.000  HINGE    *  2 Rotation spring damping (N/R/S)   b1,b2,b3          =  0.00000E+00  0.00000E+00  0.00000E+00
209.000  HINGE    *  2 Initial rotation angle (DEG) theta1,theta2,theta3  =  0.00000E+00  0.00000E+00  0.00000E+00
210.000  HINGE    *  2 Null torque angle (DEG)      theta1,theta2,theta3  =  0.00000E+00  0.00000E+00  0.00000E+00
211.000  HINGE    *  2 Initial rotation rate (D/S)  thdot1,thdot2,thdot3  =  0.00000E+00  0.00000E+00  0.00000E+00
212.000  HINGE    *  2 Translation degrees of freedom                     =  0.00000E+00
213.000  HINGE    *  2 Translation axis unit vector of 1ST DOF  X,Y,Z     =
214.000  HINGE    *  2 Translation axis unit vector of 2ND DOF  X,Y,Z     =
215.000  HINGE    *  2 Translation axis unit vector of 3RD DOF  X,Y,Z     =
216.000  HINGE    *  2 Translation spring stiffnes (N/M)  k1,k2,k3       =
217.000  HINGE    *  2 Translation spring damping (N/M/S) b1,b2,b3       =
218.000  HINGE    *  2 Initial translation displacement (M)  y1,y2,y3    =
219.000  HINGE    *  2 Null force position (M)               y1,y2,y3    =
220.000  HINGE    *  2 Initial translation velocity (M/S  yd1,yd2,yd3    =
221.000  HINGE    *  2 >>>>>***** END OF DATA *****<<<<<                 =
222.000  HINGE    *  3 ID# of hinge, ID# of inbd body, ID# of outbd body =   3.0000       2.0000       3.0000
223.000  HINGE    *  3 Node index of inboard body attach point           =   2.0000
224.000  HINGE    *  3 Rotation degrees of freedom                       =  0.00000E+00
225.000  HINGE    *  3 Base body(ID# 1)rotation option(F=free,G=gimbaled)=
226.000  HINGE    *  3 Rotation axis unit vector L1 inboard body  X,Y,Z  =   1.0000      0.00000E+00  0.00000E+00
227.000  HINGE    *  3 Rotation axis unit vector L1 outboard body X,Y,Z  =   1.0000      0.00000E+00  0.00000E+00
228.000  HINGE    *  3 Rotation axis unit vector L3 inboard body  X,Y,Z  =  0.00000E+00  0.00000E+00   1.0000
229.000  HINGE    *  3 Rotation axis unit vector L3 outboard body X,Y,Z  =  0.00000E+00  0.00000E+00   1.0000
230.000  HINGE    *  3 Rotation spring stiffness (N/R)   k1,k2,k3         =
231.000  HINGE    *  3 Rotation spring damping (N/R/S)   b1,b2,b3         =
232.000  HINGE    *  3 Initial rotation angle (DEG) theta1,theta2,theta3 =  0.00000E+00  0.00000E+00  0.00000E+00
233.000  HINGE    *  3 Null torque angle (DEG)      theta1,theta2,theta3 =
234.000  HINGE    *  3 Initial rotation rate (D/S)  thdot1,thdot2,thdot3 =
235.000  HINGE    *  3 Translation degrees of freedom                    =  0.00000E+00
236.000  HINGE    *  3 Translation axis unit vector of 1ST DOF  X,Y,Z    =
237.000  HINGE    *  3 Translation axis unit vector of 2ND DOF  X,Y,Z    =
238.000  HINGE    *  3 Translation axis unit vector of 3RD DOF  X,Y,Z    =
239.000  HINGE    *  3 Translation spring stiffnes (N/M)  k1,k2,k3      =
240.000  HINGE    *  3 Translation spring damping (N/M/S) b1,b2,b3      =
241.000  HINGE    *  3 Initial translation displacement (M)  y1,y2,y3   =
242.000  HINGE    *  3 Null force position (M)               y1,y2,y3   =
243.000  HINGE    *  3 Initial translation velocity (M/S  yd1,yd2,yd3   =
244.000  HINGE    *  3 >>>>>***** END OF DATA *****<<<<<                =
```

```
245.000  BODY   #  1 Type of body (R=rigid, F=flexible),ID#              =  F                 1.0000
246.000  BODY   #  1 Mass of body (kg)                                   =  0.30710E+06
247.000  BODY   #  1 Inertia (kg-m2) Ixx,Iyy,Izz                         =  0.29855E+09   0.23200E+09   0.94365E+08
248.000  BODY   #  1 Inertia (kg-m2) Ixy,Ixz,Iyz                         = -0.44970E+06   0.71475E+07   0.83242E+07
249.000  BODY   #  1 modal data option (T=tape,D=disc),fname and tape#   =  D                GSSMP2.FLX
250.000  BODY   #  1 Modal inertia opt (N=none,I=inertia M,N,P dyadics)= N
251.000  BODY   #  1 Modal coupling opt (N=none,C=coupling PHIXPHI vec)= N
252.000  BODY   #  1 Number of flexible modes                            =  94.000
253.000  BODY   #  1 Attach point coordinates (m) x,y,z                  = -0.44440       0.40549      -5.6515
254.000  BODY   #  1 NODE    1 mass center coordinates (m) x,y,z         = -0.44440       0.40549      -5.6515
255.000  BODY   #  1 NODE    2 coordinates (m) x,y,z                     = -21.275       -3.5560        4.1148
256.000  BODY   #  1 NODE    3 coordinates (m) x,y,z                     =  0.00000E+00  -17.831       -38.405
257.000  BODY   #  1 NODE    4 coordinates (m) x,y,z                     =  0.00000E+00  -17.831       -21.946
258.000  BODY   #  1 NODE    5 coordinates (m) x,y,z                     =  0.00000E+00   17.831       -21.946
259.000  BODY   #  1 NODE    6 coordinates (m) x,y,z                     =  0.00000E+00  -17.831        21.946
260.000  BODY   #  1 NODE    7 coordinates (m) x,y,z                     =  0.00000E+00   17.831        21.946
261.000  BODY   #  1 >>>>>***** END OF DATA *****<<<<<                   =
262.000  BODY   #  1 >>>>>***** END OF DATA *****<<<<<                   =
263.000  BODY   #  2 Type of body (R=rigid, F=flexible),ID#              =  R                 2.0000
264.000  BODY   #  2 Mass of body (kg)                                   =  1875.0
265.000  BODY   #  2 Inertia (kg-m2) Ixx,Iyy,Izz                         =  12691.         12354.         1903.0
266.000  BODY   #  2 Inertia (kg-m2) Ixy,Ixz,Iyz                         =  0.00000E+00   0.00000E+00   0.00000E+00
267.000  BODY   #  2 modal data option (T=tape,D=disc),fname and tape#   =
268.000  BODY   #  2 Modal inertia opt (N=none,I=inertia M,N,P dyadics)=
269.000  BODY   #  2 Modal coupling opt (N=none,C=coupling PHIXPHI vec)=
270.000  BODY   #  2 Number of flexible modes                            =  0.00000E+00
271.000  BODY   #  2 Attach point coordinates (m) x,y,z                  =  0.00000E+00   0.00000E+00   0.00000E+00
272.000  BODY   #  2 NODE    1 mass center coordinates (m) x,y,z         =  0.00000E+00   0.00000E+00   0.57610
273.000  BODY   #  2 NODE    2 coordinates (m) x,y,z                     =  0.00000E+00   0.00000E+00   4.3198
274.000  BODY   #  2 NODE    3 coordinates (m) x,y,z                     =  0.00000E+00   0.00000E+00   1.0000
275.000  BODY   #  2 >>>>>***** END OF DATA *****<<<<<                   =
276.000  BODY   #  2 >>>>>***** END OF DATA *****<<<<<                   =
277.000  BODY   #  3 Type of body (R=rigid, F=flexible),ID#              =  F                 3.0000
278.000  BODY   #  3 Mass of body (kg)                                   =  69.870
279.000  BODY   #  3 Inertia (kg-m2) Ixx,Iyy,Izz                         =  65391.         65416.         81.000
280.000  BODY   #  3 Inertia (kg-m2) Ixy,Ixz,Iyz                         =  0.00000E+00   0.00000E+00   0.00000E+00
281.000  BODY   #  3 modal data option (T=tape,D=disc),fname and tape#   =  D                GSSMP2.FLX
282.000  BODY   #  3 Modal inertia opt (N=none,I=inertia M,N,P dyadics)= N
283.000  BODY   #  3 Modal coupling opt (N=none,C=coupling PHIXPHI vec)= N
284.000  BODY   #  3 Number of flexible modes                            =  8.0000
285.000  BODY   #  3 Attach point coordinates (m) x,y,z                  =  0.00000E+00   0.00000E+00   0.00000E+00
286.000  BODY   #  3 NODE    1 mass center coordinates (m) x,y,z         =  0.00000E+00   0.00000E+00   29.920
287.000  BODY   #  3 NODE    2 coordinates (m) x,y,z                     =  0.00000E+00   0.00000E+00   32.000
288.000  BODY   #  3 >>>>>***** END OF DATA *****<<<<<                   =
289.000  BODY   #  3 >>>>>***** END OF DATA *****<<<<<                   =
```

DPOF3CE.FOR

```
      SUBROUTINE UCONTROL(TIME,U,R)
C
C       REVISION 12/4/86
C       CHANGE SIGN ON TORQUE CMD OUTPUTS FROM FSCONT
C
        INCLUDE 'DBP.FOR'
        INCLUDE 'USER1:[TREETOPS]DBSV.FOR'
        COMMON/SETUP/DT,ISEED,JSEED
        COMMON/S1/SFLOS(3),BLOS(3),SIGLOS,QLOS
        COMMON/S2/SFINT(3),BINT(3),SIGINT,QINT
        COMMON/S3/SFRG(3),BRG(3),SIGRG,QRG
        COMMON/A1/SFTM(3),QTM,SIGTM,CCOG,CRIP,C1,C2,ALPHA,
     *            TO1,TO2,WB,FCOG,FRIP,TWOPI,CONV,TDUMAX(3)
        COMMON/S1USER/SFLOSU,BLOSU,SGLOSU,QLOSU
        COMMON/S2USER/SFINTU,BINTU,SGINTU,QINTU
        COMMON/S3USER/SFRGU,BRGU,SIGRGU,QRGU
        COMMON/A1USER/SFTMU,QTMU,SIGTMU,CCOGU,CRIPU,C1U,C2U
        COMMON/S1NOM/SFLOSN,BLOSN,SGLOSN,QLOSN
        COMMON/S2NOM/SFINTN,BINTN,SGINTN,QINTN
        COMMON/S3NOM/SFRGN,BRGN,SIGRGN,QRGN
        COMMON/A1NOM/SFTMN,QTMN,SIGTMN,CCOGN,CRIPN,C1N,C2N
        COMMON/FLAGS/LROLL,LELEV,LXELEV
        NAMELIST/PAR/LMCRLO,LROLL,LELEV,LXELEV,
     *                 SFLOSU,BLOSU,SGLOSU,QLOSU,
     *                 SFINTU,BINTU,SGINTU,QINTU,
     *                 SFRGU,BRGU,SIGRGU,QRGU,
     *                 SFTMU,QTMU,SIGTMU,CCOGU,CRIPU,C1U,C2U,
     *                 SFLOSN,BLOSN,SGLOSN,QLOSN,
     *                 SFINTN,BINTN,SGINTN,QINTN,
     *                 SFRGN,BRGN,SIGRGN,QRGN,
     *                 SFTMN,QTMN,SIGTMN,CCOGN,CRIPN,C1N,C2N,
     *                 TO1,TO2,WB,FCOG,FRIP,TDUMAX,ALPHA
        LOGICAL LROLL,LELEV,LXELEV
        COMMON/VEL/X1(6),XC1(6),Y1(4),R1
        COMMON/VXEL/X2(6),XC2(6),Y2(4),R2
        COMMON/VROLL/X3(4),XC3(4),Y3(2),R3
      DIMENSION DESATT(3),U(1),R(1),JETON(6),IPP(3),UERROR(20),RD(3)
        DATA TWOPI/6.283185/,CONV/206264./,IFIRST /0/
C
      IF(IFIRST.EQ.1)GO TO 100
C     INITIALIZE THE VERNIER RCS CONTROLLER
C  AND THE FULL STATE GIMBAL CONTROLLER.
        CALL ASSIGN(8,'DPOF3CE.DAT')
        READ(8,NML=PAR)
        CALL PARSET
        IPP(1)=1          !SET 1ST PASS FLAGS FOR POST PROCESSOR
        IPP(2)=1
        IPP(3)=1
      ACGAIN=0
      INIT=-1
      CALL FCS(INIT,DESATT,JETON)
      INIT=0
      CALL FCS(INIT,DESATT,JETON)
        DT=.025
        CALL FSCONT(0)
      I125=3
        I100=4
      INIT=1
      IFIRST=1
  100 CONTINUE
C
```

```
C       3 AXIS GIMBAL CONTROL LAW
C
        CALL SENERR(U,UERROR,R)          !INCORPORATE SENSOR ERROR SOURCES
        Y1(1)=UERROR(1)
        Y1(2)=U(2)
        Y1(3)=UERROR(3)
        Y1(4)=-UERROR(4)
        Y2(1)=UERROR(5)
        Y2(2)=U(6)
        Y2(3)=UERROR(7)
        Y2(4)=-UERROR(8)
        Y3(1)=UERROR(9)
        Y3(2)=U(10)+THETA(1,5)
        CALL FSCONT(1)
        CALL ACTERR(R1,R2,R3,U,R,RD)     !INCORPORATE T.M. ERROR SOURCES
        I100=I100+1
        IF(I100.LT.4)GO TO 101
        I100=0
        R(1)=R1+RD(1)
        R(2)=R2+RD(2)
        R(3)=R3+RD(3)
101     R(11)=206264.*U(4)
        R(12)=206264.*U(8)
        R(13)=206264.*THETA(1,5)
        CALL POSTP(TIME,1,IPP(1),R(11),R(14),R(17))
        CALL POSTP(TIME,2,IPP(2),R(12),R(15),R(18))
        CALL POSTP(TIME,3,IPP(3),R(13),R(16),R(19))
C
C  ********** ORBITER VERNIER RCS CONTROL LAW *****************
      I125=I125+1
      IF(I125.LT.3)GO TO 102
      I125=0
C      DESATT(1)=-U(11)
C      DESATT(2)= U(12)
C      DESATT(3)=-U(13)
      CALL FCS(1,DESATT,JETON)
C      R(5)=JETON(2)*109.
C      R(6)=JETON(1)*109.
C      R(7)=JETON(4)*106.8
C      R(8)=JETON(3)*106.8
C      R(9)=JETON(6)*64.68
C      R(10)=JETON(5)*64.68
  102 CONTINUE
      RETURN
      END
      SUBROUTINE FCS(INIT,DESATT,JETON)
      DIMENSION DWRCS(3),RJCMD(3),UDACC(3),JETON(6)
      DIMENSION RATEST(3),ATTITUDE(3),AE(3),WE(3),DESATT(3)
C
      IF(INIT)100,200,300
  100 CONTINUE
C
C  SET PARAMETER DEFAULT VALUES
C
      CALL STATEST(INIT,DWRCS,ATTITUDE,RATEST,UDACC)
      CALL OPHPL(INIT,AE,WE,UDACC,RJCMD)
      CALL JET SELECT(INIT,RJCMD,JETON,DWRCS)
      RETURN
C
C  * SET INITIAL VALUES FOR FLIGHT CONTROL MODULES
C
```

```
  200 CONTINUE
      CALL STATEST(INIT,DWRCS,ATTITUDE,RATEST,UDACC)
      CALL JET SELECT(INIT,RJCMD,JETON,DWRCS)
      RETURN
C
C  * TIME HISTORY COMPUTATIONS
C
  300 CONTINUE
      CALL STATEST(INIT,DWRCS,ATTITUDE,RATEST,UDACC)
      DO 3 IAX=1,3
      AE(IAX)=ATTITUDE(IAX)-DESATT(IAX)
    3 WE(IAX)=RATEST(IAX)
      CALL OPHPL(INIT,AE,WE,UDACC,RJCMD)
      CALL JET SELECT(INIT,RJCMD,JETON,DWRCS)
C
      RETURN
      END
      SUBROUTINE ATTPROC(INIT,DELATT)
C
      DIMENSION DELATT(3),CT(3,3),C(3,3)
      INCLUDE 'DBP.FOR'
      INCLUDE 'DBB.FOR'
      IF(INIT)100,200,300
  100 CONTINUE
      RETURN
  200 CONTINUE
      DO 201 I=1,3
      DO 202 J=1,3
  202 CT(I,J)=0.0
  201 CT(I,I)=1.0
      RETURN
  300 CONTINUE
      CALL MXM(CT,CTRANS(1,1,1),C,3,3,3,3)
      DELATT(1)=(C(2,3)-C(3,2))*28.6479
      DELATT(2)=(C(1,3)-C(3,1))*28.6479
      DELATT(3)=(C(1,2)-C(2,1))*28.6479
      DO 301 I=1,3
      DO 301 J=1,3
  301 CT(I,J)=CTRANS(J,I,1)
      RETURN
      END
      SUBROUTINE STATEST(INIT,DWRCS,ATTITUDE,OMEGA1,ALPHA2)
      DIMENSION DWRCS(3),ATTITUDE(3),DELATT(3),THETAM(3),
     *THETA1(3),THETA2(3),DELY1(3),DELY2(3),
     *OMEGA1(3),OMEGA2(3),ALPHA2(3)
C
      IF(INIT)100,200,300
  100 CONTINUE
C
C  ***** SET DEFAULT VALUES *****
      TMEAS=.16
      TDAP=.08
      TDAP2=TDAP/2.
      ATGAIN1=.064
      ATGAIN2=1.0
      RGAIN1=.0016/TMEAS
      RGAIN2=.013/TMEAS
      ACCGAIN=6.4E-5/(TMEAS**2)
      CALL ATTPROC(INIT,DELATT)
       RETURN
  200 CONTINUE
```

```
C
C  ***** SET INITIAL CONDITIONS *****
      DO 201 I=1,3
      DWRCS(I)=0.
      ATTITUDE(I)=0.
      THETAM(I)=0.
      THETA1(I)=0.
      THETA2(I)=0.
      OMEGA1(I)=0.
      OMEGA2(I)=0.
      ALPHA2(I)=0.
  201 CONTINUE
      I625=0.
      RETURN
  300 CONTINUE
C
C  ***** EXTRAPOLATE THE STATE OF ALL THREE FILTERRS *****
      DO 301 I=1,3
      ATTITUDE(I)=ATTITUDE(I)+TDAP*(ALPHA2(I)*TDAP2+OMEGA1(I)+DWRCS(I))
C
      THETA1(I)=THETA1(I)+TDAP*(ALPHA2(I)*TDAP2+OMEGA1(I)+DWRCS(I))
      OMEGA1(I)=OMEGA1(I)+TDAP*ALPHA2(I)+DWRCS(I)
C
      THETA2(I)=THETA2(I)+TDAP*(ALPHA2(I)*TDAP2+OMEGA2(I)+DWRCS(I))
      OMEGA2(I)=OMEGA2(I)+TDAP*ALPHA2(I)+DWRCS(I)
C
  301 CONTINUE
C  *****UPDATE THE STATES AT 6.25 HZ RATE *****
      I625=I625+1
      IF(I625.EQ.1)GO TO 2
      I625=0
      CALL ATTPROC(INIT,DELATT)
      DO 302 I=1,3
C
      THETAM(I)=THETAM(I)+DELATT(I)
C
      ATTITUDE(I)=THETAM(I)
C
      DELY1(I)=THETAM(I)-THETA1(I)
      THETA1(I)=THETA1(I)+ATGAIN1*DELY1(I)
      OMEGA1(I)=OMEGA1(I)+ RGAIN1*DELY1(I)
C
      DELY2(I)=THETAM(I)-THETA2(I)
      THETA2(I)=THETA2(I)+ATGAIN2*DELY2(I)
      OMEGA2(I)=OMEGA2(I)+RGAIN2 *DELY2(I)
      ALPHA2(I)=ALPHA2(I)+ACCGAIN*DELY2(I)
C
  302 CONTINUE
    2 CONTINUE
      RETURN
      END
      SUBROUTINE JET SELECT(INIT,RJCMD,JETON,DWRCS)
      DIMENSION JFW(6),RJCMD(3),JETON(6)
      DIMENSION HFAIL(6),
     *OJETON(6),DWRCS(3),ORJCMD(3),ANGINC(6,3),
     *VJCMD(3),ROTVAL(3)
      INTEGER I,J,FAILC,JETC,JETON,OJETON,
     *INIT,M,K,Q,R,S
      REAL FSFLAG,RJCMD,JFW,X,VFAILD,
     *DWRCS,ORJCMD,VJCMD,ROTVAL,
     * MREPEAT,THRESH2,THRESH3,
```

```
     *ANGINC ,IJSL,KJSL,
     *VAL1,VALX,VALY,VALZ
      LOGICAL HFAIL,JFCF,VFAIL
      DATA((ANGINC(I,J),J=1,3),I=1,6)/
     *-.4401E-03 , 0.7197E-03 , -.6793E-03 ,
     *0.4401E-03 , 0.7197E-03 , 0.6793E-03 ,
     *-.7501E-03 , -.1581E-04 , 0.5561E-03 ,
     *0.7500E-03 , -.1553E-04 , -.5561E-03 ,
     *-.7800E-03 , -.3407E-03 , 0.7697E-04 ,
     *0.7800E-03 , -.3405E-03 , -.7699E-04 /
C
C*****************************************************************************
C     INPUTS: RJCMD(3)    = ROTATION COMMAND (1,0,-1)(R,P,Y)
C
C     OUTPUTS: JETON(6) = RCS JET COMMAND (1=ON,0= OFF)
C
C*********************************************************************:
      IF(INIT) 1,2,3
    1 CONTINUE
      MREPEAT = 5
      THRESH2 = .50
      THRESH3 = .4
      DO 8 I = 1,6
    8 OJETON(I) = 0
      DO 9 I = 1,3
    9 ORJCMD(I) = 0
      KJSL = 0
    2 RETURN
C     THIS PROGRAM IS THE ON=ORBIT DAP JET SELECT LOGIC
C       FOR THE VERIER JETS
C
C*****************************************************************
C
C               RJCMD(3)       ROTATIONAL CMDS
C
C     CONSTANTS: ANGINC  = ANGULAR RATES
C                 MREPEAT    = NUMBER OF PASSES WITHOUT CMD CHANGE (5)
C                 THRESH2   = THRESHOLD FOR 2ND VERNIER JET(.50)
C                 THRES3 = THRESHOLD FOR VERNIER JET 3(.4)
C
C     OUTPUTS:  JETON(6)      = RCS JET COMMANDS
C               DWRCS         =  DELTA OMEGA RCS
C
C*******************************************************************
    3 CONTINUE
      DO 5 I=1,6
    5 JETON(I)=0
      VALX = 0
      VALY = 0
      VALZ = 0
      Q = 0
      R = 0
      S = 0
C**************************************************************************
C      CHECK IF VERNIER CMDS ARE DIFFERENT FROM LAST PASS
C           OR MAX REPEAT IS EXCEEDED
C
      M=0
      DO 40 I = 1,3
      IF((INT(RJCMD(I)).NE.ORJCMD(I)).OR.(KJSL.EQ.MREPEAT)) M=1
   40 CONTINUE
```

```
      IF(M .NE.1) GO TO 300
C*****************************************************
C      CHECK IF ABS(VERNIER CMD)  = 1.0
C
      IF((ABS(RJCMD(1)).EQ.1.0).OR.(ABS(RJCMD(2)).EQ.1.0).OR.
     *(ABS(RJCMD(3)).EQ.1.0)) GO TO 50
      GO TO 200
C
   50 CONTINUE
C
C**************************************************************
C      CONDUCT TESTS 1-3 PER FIG. 4.2.2.2.1-20
C          TO SELECT VERNIER JET CMDS (JETON(1-6)
C
C  TEST 1 FIND MAX OF ANG INC * VECTOR(VJCMD)
      DO 70 I = 1,6
      VAL1 = 0
      DO 80 K = 1,3
      VAL1 = VAL1 + RJCMD(K)*ANGINC(I,K)
   80 CONTINUE
      IF(VAL1.GT.VALX) GO TO 75
      GO TO 70
   75 VALX = VAL1
      Q = I
   70 CONTINUE
C************************************************************
C  TEST 2 FIND 2ND MAX OF ANGINC*VECTOR(VJCMD)>THRESH2*VALX
      DO 90 I = 1,6
      VAL1 = 0
      IF(I.EQ.Q) GO TO 90
      DO 100 K = 1,3
      VAL1 = VAL1 + RJCMD(K)*ANGINC(I,K)
  100 CONTINUE
      IF((VAL1.GT.(THRESH2*VALX)).AND.(VAL1.GT.VALY)) GO TO 95
      GO TO 90
   95 VALY = VAL1
      R = I
   90 CONTINUE
C************************************************************
C  TEST 3 FIND 3RD MAX OF ANGINC*VECTOR(VJCMD)>THRESH3*VALX
      IF(R.EQ.0) GO TO 120
      DO 120 I = 1,6
      VAL1 = 0
      IF((I.NE.Q).AND.(I.NE.R)) GO TO 105
      GO TO 120
  105 DO 110 K = 1,3
      VAL1 = VAL1 + RJCMD(K)*ANGINC(I,K)
  110 CONTINUE
      IF((VAL1.GT.(THRESH3*VALX)).AND.(VAL1.GT.VALZ)) GO TO 115
      GO TO 120
  115 VALZ = VAL1
      S = I
  120 CONTINUE
C***********************************************************
C      SET JETON(39-44) PER RESULTS OF TESTS 1-3 (Q,R,S)
      JETON(Q)=1
      IF(R.NE.0) JETON(R) = 1
      IF(S.NE.0) JETON(S) = 1
  200 CONTINUE
C  COMPUTE DELTA OMEGA RCS
      DO 220 K = 1,3
```

```
      DWRCS(K)=0.0
      DO 220 I = 1,6
      IF(JETON(I).EQ.1)DWRCS(K)=DWRCS(K)+ANGINC(I,K)
  220 CONTINUE
      KJSL = 0
      DO 260 I = 1,3
      ORJCMD(I)  = INT(RJCMD(I))
  260 CONTINUE
      DO 270 I = 1,6
      OJETON(I)     = JETON(I)
  270 CONTINUE
      GO TO 310
  300 KJSL = KJSL + 1
      DO 310  I = 1,6
      JETON(I) = OJETON(I)
  310 CONTINUE
      RETURN
      END
C
      SUBROUTINE OPHPL(INIT,AE,WE,UDACC,RJCMD)
C
C  ----- ON-ORBIT PHASE PLANE -----
C
      DIMENSION AE(3),WE(3),DB1(3),RLIMIT(3),ACC(3),
     *          UDACC(3),WMIN(3),RJCMD(3),BYPASS(3),WFRATE(3),
     *          X1(3),Y1(3),X2(3),Y2(3)
C
      IF(INIT)100,200,300
  100 CONTINUE
C SET DEFAULT VALUES
      DO 101 I=1,3
      WFRATE(I)=0.8
      DB1(I)=0.1
      RLIMIT(I)=.02

  101 CONTINUE
      ACC(1)=.01872
      ACC(2)=.01096
      ACC(3)=.01264
      WMIN(1)=.001872
      WMIN(2)=.001096
      WMIN(3)=.001264
      RETURN
  200 CONTINUE
      RETURN
  300 CONTINUE
C
C  DEFINE LOCAL VARIABLES
C
      DO 301 IAX=1,3
      X1(IAX) = SIGN(1.,WE(IAX))*AE(IAX)
      X2(IAX) = ABS(WE(IAX))
      Y1(IAX) = SIGN(1.0,UDACC(IAX))*AE(IAX)
      Y2(IAX) = SIGN(1.0,UDACC(IAX))*WE(IAX)
      C  = 1.0
      IF(ABS(RJCMD(IAX)) .NE. 1.0) C= 1.25
C
C  DEFINE SWITCHING LINES
C
C     REV 1B DELETED ABS UDACC PER CR12710
```

```
      UCP = ACC(IAX)-SIGN(1.,WE(IAX))*UDACC(IAX)
        S = 0.0
        SY = 0.0
        IF(UCP .EQ. 0.0) GO TO 105
      S    = -(X2(IAX)*X2(IAX))/(2.0*UCP)
      SY   = -(Y2(IAX)*Y2(IAX))/(2.0*UCP)
 105    CONTINUE
C
      S1   = S + DB1(IAX)
      S1Y  = SY + DB1(IAX)
      S2   = S*C - 1.2*DB1(IAX)
      S2Y  = SY*C - 1.2*DB1(IAX)
      S3   = RLIMIT(IAX)
      S4 = 0.8*RLIMIT(IAX)
      S5 = 0.6*RLIMIT(IAX)
      S7 = (-1)*SIGN(1.,Y2(IAX))*(-1.)*SY-DB1(IAX)
      S8   = -RLIMIT(IAX)
      S10 = (-1.)*C*SY + 1.2*DB1(IAX)
      IF((Y1(IAX).LT.(-.5)*DB1(IAX)).AND.(Y1(IAX).GE.(-1.2)*DB1(IAX)
     *))
     * S11 = 0.0
      IF((Y1(IAX).GE.-0.5*DB1(IAX)).AND.(Y1(IAX).LE.S10)
     *) S11 = -SQRT(
     *   2.0*ABS(UDACC(IAX))*(Y1(IAX)+0.5*DB1(IAX)))+WMIN(IAX)
      IF(S11 .GT. 0.0) S11 = 0.0
      IF(S11.LT.(-RLIMIT(IAX)+WMIN(IAX)))S11=-RLIMIT(IAX)+WMIN(IAX)
      S14 = -SY + DB1(IAX)
C
C  PHASE PLANE CONTROL LOGIC
C
C
C     REGION 1 CONTROL
C
      IF((X1(IAX).GT.S1).OR.(X2(IAX).GT.S3)) GO TO 10
C
C     REGION 2 CONTROL
C   **DISTURBANCE HYSTERESIS REGION**
C
      IF((X1(IAX).GE.S2).AND.(X1(IAX).LE.S1).AND.(X2(IAX).LE.S3))
     * GO TO 15
C
C     REGION 3 CONTROL
C
      IF((X1(IAX).LT.S2).AND.(X2(IAX).LT.S5)) GO TO 25
C
C     REGION 4 CONTROL
C
      IF((X1(IAX).LT.S2).AND.(S4.LE.X2(IAX)).AND.(X2(IAX).LE.S3))
     * GO TO 35
C    REGION 5 CONTROL
      IF((X1(IAX).LT.S2).AND.(S5.LE.X2(IAX)).AND.(X2(IAX).LT.S4))
     * GO TO 40
      WRITE(6,1000)
 1000 FORMAT( 'PHASE PLANE ERROR - FALLS THRU REGION 5 LOGIC')
      GO TO 80
C
C
C  PHASE PLANE ACTION
C
C     REGION 1 ACTION
C
```

```
C
   10 CONTINUE
      RJCMD(IAX) = -SIGN(1.,WE(IAX))
      GO TO 80
C
C     REGION 2 ACTION
C    **DISTURBANCE HYSTERESIS REGION LOGIC
C
C     REGION 2 ACTION, CS REGION CONTROL
C
C
   15 CONTINUE
      IF(((S2Y.LE.Y1(IAX)).AND.(Y1(IAX).LT.S7).AND.(Y2(IAX).GE.0.0).AND.
     *    (Y2(IAX).LE.S3)).OR.((S14.LT.Y1(IAX)).AND.(Y1(IAX).LE.S10)
     *     .AND.(S8.LE.Y2(IAX)).AND.(Y2(IAX).LT.S11))) GO TO 20
C
C     DISTURBANCE HYSTERESIS REGION 1
C
C
      IF(((S7.LE.Y1(IAX)).AND.(Y1(IAX).LE.S1Y).AND.(Y2(IAX).GE.0.0).AND.
     *    (Y2(IAX).LE.S3)).OR.((S7.LE.Y1(IAX)).AND.(Y1(IAX).LE.S10).AND.
     *    (S11.LE.Y2(IAX)).AND.(Y2(IAX).LT.0.0))) GO TO 21
C
C     DISTURBANCE HYSTERESIS REGION 2
C      DEFAULT DISTURBANCE HYSTERESIS
C
C
      GO TO 23
C
C
C
C     CS REGION ACTION
C
   20 CONTINUE
      RJCMD(IAX) = SIGN(1.0,UDACC(IAX))*WFRATE(IAX)*
     *((S11-Y2(IAX))/(RLIMIT(IAX)+S11))
      GO TO 80
C
C     HYSTERESIS REGION 1 ACTION
C
   21 CONTINUE
      IF(RJCMD(IAX).EQ.-SIGN(1.0,UDACC(IAX))) GO TO 80
   22 CONTINUE
      RJCMD(IAX) = (-1.)*SIGN(1.0,UDACC(IAX))*WFRATE(IAX)*
     *              ((Y2(IAX)-S11)/(RLIMIT(IAX)-S11))
      GO TO 80
C
C     HYSTERESIS REGION 2 ACTION
C      DEFAULT ACTION
C
   23 CONTINUE
      IF(RJCMD(IAX).EQ.SIGN(1.0,UDACC(IAX))) GO TO 80
      RJCMD(IAX) = SIGN(1.0,UDACC(IAX))*WFRATE(IAX)*
     *              ((S11-Y2(IAX))/(RLIMIT(IAX)+S11))
      GO TO 80
C
C     REGION 3 ACTION
C
   25 CONTINUE
      RJCMD(IAX) = SIGN(1.,WE(IAX))
      GO TO 80
```

```
C
C     REGION 4 ACTION
C
   35 IF(RJCMD(IAX).EQ.-SIGN(1.0,WE(IAX))) GO TO 80
      RJCMD(IAX)=-SIGN(1.0,WE(IAX))*WFRATE(IAX)*((0.8*RLIMIT(IAX)
     *-X2(IAX))/(0.2*RLIMIT(IAX)))
      GO TO 80
   40 IF(RJCMD(IAX).EQ.SIGN(1.0,WE(IAX))) GO TO 80
      RJCMD(IAX)=SIGN(1.0,WE(IAX))*WFRATE(IAX)*
     *((0.8*RLIMIT(IAX)-X2(IAX))/(0.2*RLIMIT(IAX)))
      GO TO 80
C
C
   80 CONTINUE
   81 CONTINUE
  301 CONTINUE
      RETURN
C
      END
        SUBROUTINE FSCONT(INIT)
        COMMON/VEL/X1(6),XC1(6),Y1(4),R1
        COMMON/VXEL/X2(6),XC2(6),Y2(4),R2
        COMMON/VROLL/X3(4),XC3(4),Y3(2),R3
        DIMENSION EFGH1(16,16),EFGH2(16,16),EFGH3(10,10)
        DIMENSION XTEMP1(16),YTEMP1(7),XTEMP2(16),YTEMP2(7)
        DIMENSION XTEMP3(10),YTEMP3(5)
        COMMON/MAIN1/NDIM
        EQUIVALENCE (X1(1),XTEMP1(1)),(XC1(1),XTEMP1(7)),
     1              (Y1(1),XTEMP1(13))
        EQUIVALENCE (X2(1),XTEMP2(1)),(XC2(1),XTEMP2(7)),
     1              (Y2(1),XTEMP2(13))
        EQUIVALENCE (X3(1),XTEMP3(1)),(XC3(1),XTEMP3(5)),
     1              (Y3(1),XTEMP3(9))
        IF(INIT.EQ.0)CALL ASSIGN(7,'FSCONT.DAT')
        IF(INIT.GT.0)GO TO 100
C
C       ZERO REDUCED STATE VECTORS X1,X2,X3 AND INPUT VECTOR XC1,XC2,XC3
C
        DO 10 I=1,6
        X1(I)=0.
        XC1(I)=0.
        X2(I)=0.
        XC2(I)=0.
        IF(I.GT.4)GO TO 10
        X3(I)=0.
        XC3(I)=0.
10      CONTINUE
C
C       READ EFGH MATRIX QUADS
C
        READ(7)
        READ(7)IV,((EFGH1(I,J),I=1,7),J=1,16)   !ELEVATION AXIS
        READ(7)
        READ(7)IV,((EFGH2(I,J),I=1,7),J=1,16)   !CROSS ELEVATION AXIS
        READ(7)
        READ(7)IV,((EFGH3(I,J),I=1,5),J=1,10)   !ROLL AXIS
        RETURN
100     CONTINUE
C
C       CALCULATE X(K+1)=AX(K)+BU(K) , Y(K)=CX(K)+DU(K)
C
```

```
      NDIM=16
      CALL VMMUL(7,16,EFGH1,XTEMP1,YTEMP1) .   !ELEVATION AXIS
      CALL VECTEQ(6,YTEMP1,X1)
      R1=YTEMP1(7)
      CALL VMMUL(7,16,EFGH2,XTEMP2,YTEMP2)     !CROSS ELEVATION AXIS
      CALL VECTEQ(6,YTEMP2,X2)
      R2=YTEMP2(7)
      NDIM=10
      CALL VMMUL(5,10,EFGH3,XTEMP3,YTEMP3)     !ROLL AXIS
      CALL VECTEQ(4,YTEMP3,X3)
      R3=YTEMP3(5)
      RETURN
      END
      SUBROUTINE VMMUL(N1,N2,X,Y,Z)
      DIMENSION X(1),Y(1),Z(1)
      COMMON/MAIN1/NDIM
      DO 10 I=1,N1
      Z(I)=0.
      DO 10 J=1,N2
10    Z(I)=Z(I)+X(I+(J-1)*NDIM)*Y(J)
      RETURN
      END
      SUBROUTINE VECTEQ(N,X,Y)
      DIMENSION X(1),Y(1)
      DO 10 I=1,N
10    Y(I)=X(I)
      RETURN
      END
      SUBROUTINE POSTP(TIME,J,I,XIN,XMEAN,XRMS)
C
C     THIS SUBROUTINE IS A POST PROCESSOR FOR DETERMINING THE
C     MEAN AND MEAN SQUARE ERROR STATISTICS FOR A GIVEN RANDOM
C     VARIABLE.
C
C     INPUTS: 1)I - FIRST PASS FLAG
C             2)XIN - RANDOM VARIABLE
C             3)J - RV IDENTIFIER NUMBER
C             4)TIME - TIME
C
C     OUTPUTS:1)XMEAN - RANDOM VARIABLE MEAN
C             2)XRMS - RANDOM VARIABLE RMS
C
      DIMENSION I(1),SUM(10),SUM1(10),K(10)
      IF(I(J).EQ.0)GO TO 10
      K(J)=0
      I(J)=0
      SUM(J)=0.
      SUM1(J)=0.
10    IF(TIME.LT.0.)GO TO 20
      K(J)=K(J)+1
      SUM(J)=SUM(J)+XIN
      XMEAN=SUM(J)/K(J)                  !COMPUTE MEAN
      SUM1(J)=SUM1(J)+XIN*XIN
      XRMS=SQRT(SUM1(J)/K(J))            !COMPUTE STD DEV
20    CONTINUE
      RETURN
      END
      SUBROUTINE PARSET
      COMMON/SETUP/DT,ISEED,JSEED
      COMMON/S1/SFLOS(3),BLOS(3),SIGLOS,QLOS
      COMMON/S2/SFINT(3),BINT(3),SIGINT,QINT
```

```
      COMMON/S3/SFRG(3),BRG(3),SIGRG,QRG
      COMMON/A1/SFTM(3),QTM,SIGTM,CCOG,CRIP,C1,C2,ALPHA,
     *          TO1,TO2,WB,FCOG,FRIP,TWOPI,CONV,TDUMAX(3)
      COMMON/S1USER/SFLOSU,BLOSU,SGLOSU,QLOSU
      COMMON/S2USER/SFINTU,BINTU,SGINTU,QINTU
      COMMON/S3USER/SFRGU,BRGU,SIGRGU,QRGU
      COMMON/A1USER/SFTMU,QTMU,SIGTMU,CCOGU,CRIPU,C1U,C2U
      COMMON/S1NOM/SFLOSN,BLOSN,SGLOSN,QLOSN
      COMMON/S2NOM/SFINTN,BINTN,SGINTN,QINTN
      COMMON/S3NOM/SFRGN,BRGN,SIGRGN,QRGN
      COMMON/A1NOM/SFTMN,QTMN,SIGTMN,CCOGN,CRIPN,C1N,C2N
      COMMON/FLAGS/LMCRLO,LROLL,LELEV,LXELEV
      LOGICAL LMCRLO,LELEV,LXELEV,LROLL
      REAL NOISE
      SEC=SECNDS(0.)
      ISEC=IFIX(SEC)
      ISEED=ISECS/2+1
      JSEED=ISECS/4+1
      IF(LMCRLO)GO TO 15
      SIGLOS=SGLOSU/CONV                        !SENSITIVITY RUN PAR. SETUP
      SIGINT=SGINTU
      SIGRG=SIGRGU/CONV
      QLOS=QLOSU/CONV
      QINT=QINTU
      QRG=QRGU/CONV
      QTM=QTMU
      SIGTM=SIGTMU
      CCOG=CCOGU
      CRIP=CRIPU
      C1=C1U
      C2=C2U
      DO 10 I=1,3
      IF(I.EQ.1.AND.LELEV)GO TO 5
      IF(I.EQ.2.AND.LXELEV)GO TO 5
      IF(I.EQ.3.AND.LROLL)GO TO 5
      GO TO 10
5     SFLOS(I)=SFLOSU/CONV
      SFINT(I)=SFINTU
      SFRG(I)=SFRGU/CONV
      SFTM(I)=SFTMU
      BLOS(I)=BLOSU/CONV
      BINT(I)=BINTU
      BRG(I)=BRGU/CONV
10    CONTINUE
      RETURN
15    SIGLOS=SGLOSN/CONV                        !MONTE CARLO RUN PAR. SETUP
      SIGINT=SGINTN
      SIGRG=SIGRGN/CONV
      QLOS=QLOSN/CONV
      QINT=QINTN
      QRG=QRGN/CONV
      QTM=QTMU
      SIGTM=SIGTMN
      CCOG=CCOGN
      CRIP=CRIPN
      C1=C1N
      C2=C2N
      DO 25 I=1,3
      IF(I.EQ.1.AND.LELEV)GO TO 20
      IF(I.EQ.2.AND.LXELEV)GO TO 20
      IF(I.EQ.3.AND.LROLL)GO TO 20
```

```
20    SFLOS(I)=SFLOSN*NOISE(ISEED)
      SFINT(I)=SFINTN*NOISE(ISEED)
      SFRG(I)=SFRGN*NOISE(ISEED)
      SFTM(I)=SFTMN*NOISE(JSEED)
      BLOS(I)=BLOSN*NOISE(ISEED)
      BINT(I)=BINTN*NOISE(ISEED)
      BRG(I)=BRGN*NOISE(ISEED)
25    CONTINUE
      RETURN
      END
      SUBROUTINE SENERR(U,UERROR,R)
C
C     THIS SUBROUTINE ADDS ERROR SOURCES TO THE SENSOR OUTPUTS
C
C     INPUTS:1)LROLL - ROLL AXIS ERROR SOURCE ENABLE DISCRETE
C            2)LELEV - ELEVATION AXIS ERROR SOURCE ENABLE DISCRETE
C            3)LXELEV - CROSS ELEVATION AXIS ERROR SOURCE ENABLE DISCRETE
C            4)U(1) - ELEVATION AXIS RATE GYRO OUTPUT
C            5)U(3) - X1:BOOM TIP DEFLECTION
C            6)U(4) - d2:LOS ANGLE
C            7)U(5) - CROSS ELEVATION AXIS RATE GYRO OUTPUT
C            8)U(7) - X2:BOOM TIP DEFLECTION
C            9)U(8) - d1:LOS ANGLE
C           10)U(9) - ROLL AXIS RATE GYRO
C           11)U(14) - ELEVATION AXIS SHAPING FILTER OUTPUT
C           12)U(15) - CROSS ELEVATION AXIS SHAPING FILTER OUTPUT
C           13)U(16) - ROLL AXIS SHAPING FILTER OUTPUT
C
C     OUTPUTS:1)UERROR(1) - U(1) WITH ERROR SOURCES
C             2)UERROR(3) - U(3) WITH ERROR SOURCES
C             3)UERROR(4) - U(4) WITH ERROR SOURCES
C             4)UERROR(5) - U(5) WITH ERROR SOURCES
C             5)UERROR(7) - U(7) WITH ERROR SOURCES
C             6)UERROR(8) - U(8) WITH ERROR SOURCES
C             7)UERROR(9) - U(9) WITH ERROR SOURCES
C             8)R(21) - NOISE INPUT TO SHAPING FILTER(EL AXIS)
C             9)R(22) - NOISE INPUT TO SHAPING FILTER(X EL AXIS)
C            10)R(23) - NOISE INPUT TO SHAPING FILTER(ROLL AXIS)
C
      DIMENSION U(1),UERROR(1),R(1)
      COMMON/FLAGS/LMCRLO,LROLL,LELEV,LXELEV
      COMMON/SETUP/DT,ISEED,JSEED
      LOGICAL LROLL,LELEV,LXELEV
      COMMON/S1/SFLOS(3),BLOS(3),SIGLOS,QLOS
      COMMON/S2/SFINT(3),BINT(3),SIGINT,QINT
      COMMON/S3/SFRG(3),BRG(3),SIGRG,QRG
      COMMON/A1/SFTM(3),QTM,SIGTM,CCOG,CRIP,C1,C2,ALPHA,
     *        TO1,TO2,WB,FCOG,FRIP,TWOPI,CONV,TDUMAX(3)
      CALL LOS(1,LELEV,U(4),UERROR(4))        !ADD ELEVATION AXIS SENSOR ERRORS
      CALL LASER(1,LELEV,U(3),UERROR(3))
      CALL GYRO(1,LELEV,U(1),U(14),R(21),UERROR(1))
      CALL LOS(2,LXELEV,U(8),UERROR(8))       !ADD CROSS EL AXIS SENSOR ERRORS
      CALL LASER(2,LXELEV,U(7),UERROR(7))
      CALL GYRO(2,LXELEV,U(5),U(15),R(22),UERROR(5))
      CALL GYRO(3,LROLL,U(9),U(16),R(23),UERROR(9))    !ADD ROLL AXIS ERRORS
      RETURN
      END
      SUBROUTINE ACTERR(TCX,TCY,TCZ,U,R,RD)
C
C     THIS SUBROUTINE ADDS ERROR SOURCES TO THE 3 GIMBAL MOUNTED
C     TORQUE ACTUATORS
```

```
C
C      INPUTS:1)LROLL - ROLL AXIS ERROR SOURCE ENABLE DISCREET
C              2)LELEV - ELEVATION AXIS ERROR SOURCE ENABLE DISCREET
C              3)LXELEV - CROSS EL AXIS ERROR SOURCE ENABLE DISCREET
C              4)THETA(1,3) - ELEVATION AXIS GIMBAL ANGLE
C              5)THETA(1,4) - CROSS ELEVATION AXIS GIMBAL ANGLE
C              6)THETA(1,5) - ROLL AXIS GIMBAL ANGLE
C              7)THDOT(1,3) - ELEVATION AXIS GIMBAL RATE
C              8)THDOT(1,4) - CROSS ELEVATION AXIS GIMBAL RATE
C              9)THDOT(1,5) - ROLL AXIS GIMBAL RATE
C             10)TCX - UNQUANTIZED ELEVATION AXIS TORQUE COMMAND
C             11)TCY - UNQUANTIZED CROSS EL AXIS TORQUE COMMAND
C             12)TCZ - UNQUANTIZED ROLL AXIS TORQUE COMMAND
C             13)U(17) - ELEVATION AXIS TORQUER OUTPUT
C             14)U(18) - CROSS ELEVATION AXIS TORQUER OUTPUT
C             15)U(19) - ROLL AXIS TORQUER OUTPUT
C
C      OUTPUTS:1)R(24) - QUANTIZED ELEVATION AXIS TORQUE COMMAND
C               2)RD(1) - ELEVATION AXIS DISTURBANCE TORQUE
C               3)R(25) - QUANTIZED CROSS EL AXIS TORQUE COMMAND
C               4)RD(2) - CROSS ELEVATION AXIS DISTURBANCE TORQUE
C               5)R(26) - QUANTIZED ROLL AXIS TORQUE COMMAND
C               6)RD(3) - ROLL AXIS DISTURBANCE TORQUE
C
      COMMON/FLAGS/LMCRLO,LROLL,LELEV,LXELEV
      COMMON/A1/SFTM(3),QTM,SIGTM,CCOG,CRIP,C1,C2,ALPHA,
     *       TO1,TO2,WB,FCOG,FRIP,TWOPI,CONV,TDUMAX(3)
      COMMON/SETUP/DT,ISEED,JSEED
      LOGICAL LMCRLO,LROLL,LELEV,LXELEV
      INCLUDE 'DBP.FOR'
      INCLUDE 'USER1:[TREETOPS]DBSV.FOR'
      DIMENSION U(1),R(1),RD(1)
      CALL MOTOR(LELEV,1,THDOT(1,3),THETA(1,3),TCY,U(17),R(24),RD(1))
      CALL MOTOR(LXELEV,2,THDOT(1,4),THETA(1,4),TCZ,U(18),R(25),RD(2))
      CALL MOTOR(LROLL,3,THDOT(1,5),THETA(1,5),TCX,U(19),R(26),RD(3))
      RETURN
      END
      SUBROUTINE LOS(I,ENABLE,X,Y)
C
C      THIS SUBROUTINE ADDS LOS SENSOR ERROR SOURCES - SCALE FACTOR,
C      BIAS, NOISE AND QUANTIZATION TO THE TREETOPS SENSOR OUTPUT.
C
C      INPUTS:1)SFLOS - SCALE FACTOR ERROR(%)
C              2)BLOS - BIAS ERROR(ARCSEC)
C              3)SIGLOS - STD DEV OF NOISE(ARCSEC)
C              4)QLOS - QUANTIZATION ERROR(ARCSEC)
C              5)X - SENSOR OUTPUT FROM TREETOPS
C              6)TIME - TIME(sec)
C              7)ENABLE - ENABLE ERROR SOURCE CALCULATION FLAG
C              8)I - AXIS INDICATOR
C
C      OUTPUTS:1)Y - SENSOR OUTPUT + ERROR SOURCES
C
      COMMON/S1/SFLOS(3),BLOS(3),SIGLOS,QLOS
      COMMON/SETUP/DT,ISEED,JSEED
      REAL NOISE
      LOGICAL ENABLE
      IF(ENABLE)GO TO 10
      Y=X
      RETURN
10    Y=(1.+SFLOS(I))*X+BLOS(I)+SIGLOS*NOISE(ISEED)
```

```
      IF(QLOS.NE.O.)Y=QLOS*NINT(Y/QLOS)
      RETURN
      END
      SUBROUTINE LASER(I,ENABLE,X,Y)
C
C     THIS SUBROUTINE ADDS LASER INTERFEROMETER SENSOR ERROR SOURCES
C     - SCALE FACTOR, BIAS, NOISE AND QUANTIZATION TO THE TREETOPS
C     SENSOR OUTPUT.
C
C     INPUTS:1)SFINT - SCALE FACTOR ERROR(%)
C            2)BINT - BIAS ERROR(METERS)
C            3)SIGINT - STD DEV OF NOISE(METERS)
C            4)QINT - QUANTIZATION ERROR(METERS)
C            5)X - SENSOR OUTPUT FROM TREETOPS(METERS)
C            6)TIME - TIME(sec)
C            7)ENABLE - ENABLE ERROR SOURCE CALCULATION FLAG
C            8)I - AXIS INDICATOR
C
C     OUTPUTS:1)Y - SENSOR OUTPUT + ERROR SOURCES
C
      REAL NOISE
      LOGICAL ENABLE
      COMMON/S2/SFINT(3),BINT(3),SIGINT,QINT
      COMMON/SETUP/DT,ISEED,JSEED
      IF(ENABLE)GO TO 10
      Y=X
      RETURN
10    Y=(1.+SFINT(3))*X+BINT(3)+SIGINT*NOISE(ISEED)
      IF(QINT.NE.O.)Y=QINT*NINT(Y/QINT)
      RETURN
      END
      SUBROUTINE GYRO(I,ENABLE,X1,X2,Y1,Y2)
C
C     THIS SUBROUTINE ADDS RATE GYRO ERROR SOURCES - SCALE FACTOR,
C     BIAS, NOISE AND QUANTIZATION TO THE TREETOPS SENSOR OUTPUT
C
C     INPUTS:1)SFRG - SCALE FACTOR ERROR(%)
C            2)BRG - BIAS ERROR
C            3)SIGRG - STD DEV OF NOISE
C            4)QRG - QUANTIZATION ERROR
C            5)X1 - GYRO DYNAMICS OUTPUT
C            6)X2 - SHAPING FILTER OUTPUT
C            7)ENABLE - ENABLE ERROR SOURCE CALCULATION FLAG
C            8)I - AXIS INDICATOR
C
C     OUTPUTS:1)Y1 - INPUT TO SHAPING FILTER
C             2)Y2 - SENSOR OUTPUT + ERROR SOURCES
C
      COMMON/S3/SFRG(3),BRG(3),SIGRG,QRG
      COMMON/SETUP/DT,ISEED,JSEED
      LOGICAL ENABLE
      REAL NOISE
      IF(ENABLE)GO TO 10
      Y1=O.
      Y2=X1
      RETURN
10    Y1=SIGRG*NOISE(ISEED)
      Y2=(1.+SFRG(I))*X1+BRG(I)+X2
      IF(QRG.NE.O.)Y2=QRG*NINT(Y2/QRG)
      RETURN
      END
```

```
      SUBROUTINE MOTOR(ENABLE,IAXIS,RATE,ANGLE,X1,X2,Y1,Y2)
C
C     THIS SUBROUTINE ADDS TORQUE MOTOR ERROR SOURCES TO THE CONTROLLER
C     GENERATOR TORQUE MOTOR COMMAND.  THESE ERROR SOURCES INCLUDE
C     QUANTIZATION, SCALE FACTOR, COGGING & RIPPLE, CROSS COUPLING AND
C     DAHL FRICTION.
C
C     INPUTS:1)SFTM - SCALE FACTOR ERROR(%)
C            2)QTM - QUANTIZATION ERROR(nm)
C            3)SIGTM - ROUGHNESS TORQUE NOISE STD DEV(nm)
C            4)RATE - GIMBAL RATE(rad/sec)
C            5)ANGLE - GIMBAL ANGLE(rad)
C            6)X1 - COMMANDED TORQUE INPUT(nm)
C            7)X2 - TORQUER OUTPUT(nm)
C            8)ENABLE - ENABLE ERROR SOURCE CALCULATION FLAG
C            9)IAXIS - 1 = ELEVATION AXIS
C                      2 = CROSS ELEVATION AXIS
C                      3 = ROLL AXIS
C
C     OUTPUTS:1)Y2 - DISTURBANCE TORQUE
C             2)Y1 - TORQUE OUTPUT TO PEU & TORQUER DYNAMICS
C
      COMMON/SETUP/DT,ISEED,JSEED
      COMMON/A1/SFTM(3),QTM,SIGTM,CCOG,CRIP,C1,C2,ALPHA,
     *          TO1,TO2,WB,FCOG,FRIP,TWOPI,CONV,TDUMAX(3)
      REAL NOISE
      LOGICAL ENABLE
C
C     LIMIT DRIVE UNIT MAXIMUM OUTPUT TORQUE
C
      IF(IAXIS.EQ.3.AND.ABS(X2).GT.TDUMAX(1))X2=SIGN(1.,X2)*TDUMAX(1)
      IF(IAXIS.NE.3.AND.ABS(X2).GT.TDUMAX(2))X2=SIGN(1.,X2)*TDUMAX(2)
      IF(ENABLE)GO TO 3
      Y2=0.
      Y1=X1
      RETURN
3     CONTINUE
C
C     COMPUTE INPUT TO TORQUER DYNAMICS
C
      Y=X1
      IF(QTM.NE.0.)Y=QTM*NINT(Y/QTM)
      Y1=(1.+SFTM(IAXIS))*Y
C
C     COMPUTE FRICTION TORQUES(DAHL)
C
      IF(TIME.NE.0.)GO TO 5
      G1=C1/(4.*TO1**2)
      G2=C2/(4.*TO2**2)
5     CALL DAHL(G1,TO1,RATE,IF1,TF1)
      CALL DAHL(G2,TO2,RATE,IF2,TF2)
C
C     COMPUTE ROUGHNESS TORQUE
C
      IF(ABS(RATE).EQ.0.)GO TO 10
      A=EXP(-WB*ABS(RATE)*DT)
      TR=A*TR+(1.-A)*SQRT((1.+A)/(1.-A))*SIGTM*NOISE(JSEED)
10    CONTINUE
C
C     COMPUTE COGGING TORQUE
C
```

```
      TCOG=2.*CCOG*SIN(FCOG*ANGLE)
C
C     COMPUTE RIPPLE TORQUE
C
      TRIP=X2*CRIP*SIN(FRIP*ANGLE)
      Y2=TF1+TF2+TR+TCOG+TRIP
C
C     COMPUTE TORQUE CROSS COUPLING
C
      IF(IAXIS.EQ.2)DTXDU=X2-Y2
      IF(IAXIS.EQ.3)DTRDU=X2-Y2
      IF(IAXIS.NE.1)GO TO 20
      TCEDU=-2.*DTRDU*COS(ANGLE+ALPHA/2.)*SIN(ALPHA/2.)-
     *       DTXDU*SIN(ALPHA)
      Y2=Y2+TCEDU
      RETURN
20    IF(IAXIS.NE.2)RETURN
      TCXDU=-DTRDU*SIN(ALPHA)
      Y2=Y2+TCXDU
      RETURN
      END
      SUBROUTINE DAHL(K1,K2,RATE,IFLAG,TF)
      REAL K1,K2
      COMMON/SETUP/DT,ISEED,JSEED
     IF(TF*SIGN(1.,RATE).LE.K2)TFDOT=RATE*K1*(TF*SIGN(1.,RATE)-K2)**2
     IF(TF*SIGN(1.,RATE).GT.K2)TFDOT=-RATE*K1*(TF*SIGN(1.,RATE)-K2)**2
      TF=TF+TFDOT*DT
      IF(ABS(TF).GT.K2)TF=SIGN(1.,RATE)*K2
      RETURN
      END
      REAL FUNCTION NOISE(ISEED)
      SUM=0.
      DO 4 J=1,6
      SUM=SUM+RAN(ISEED)
4     NOISE=SQRT(2.)*(SUM-3.)
      RETURN
      END
```

SSPOF3.FOR

```
      SUBROUTINE UCONTROL(TIME,U,R)
C
C       REVISION 12/4/86
C       CHANGE SIGN ON TORQUE CMD OUTPUTS FROM FSCONT
C
        INCLUDE 'DBP.FOR'
        INCLUDE 'USER1:[TREETOPS]DBSV.FOR'
        COMMON/VEL/X1(6),XC1(6),Y1(4),R1
        COMMON/VXEL/X2(6),XC2(6),Y2(4),R2
        COMMON/VROLL/X3(4),XC3(4),Y3(2),R3
      DIMENSION DESATT(3),U(1),R(1),JETON(6),IPP(3)
      DATA IFIRST /0/
C
      IF(IFIRST.EQ.1)GO TO 100
C     INITIALIZE THE FULL STATE GIMBAL CONTROLLER.
        IPP(1)=1          !SET 1ST PASS FLAGS FOR POST PROCESSOR
        IPP(2)=1
        IPP(3)=1
        DT=.025
        CALL FSCONT(0)
      IFIRST=1
  100 CONTINUE
C
C       3 AXIS GIMBAL CONTROL LAW
C
        Y1(1)=U(1)
        Y1(2)=U(2)
        Y1(3)=U(3)
        Y1(4)=-U(4)
        Y2(1)=U(5)
        Y2(2)=U(6)
        Y2(3)=U(7)
        Y2(4)=-U(8)
        Y3(1)=U(9)
        Y3(2)=U(10)+THETA(1,5)
        CALL FSCONT(1)
        IF(ABS(R1).GT.26.37)R1=(R1/ABS(R1))*26.37       !REV 12/4
        IF(ABS(R2).GT.26.37)R2=(R2/ABS(R2))*26.37       !REV 12/4
        IF(ABS(R3).GT.26.37)R3=(R3/ABS(R3))*9.88        !REV 12/4
        R(1)=R1                                         !REV 12/4
        R(2)=R2                                         !REV 12/4
        R(3)=R3                                         !REV 12/4
        R(11)=206264.*U(4)
        R(12)=206264.*U(8)
        R(13)=206264.*THETA(1,5)
        R(14)=X1(1)
        R(15)=X1(2)
        R(16)=X1(3)
        R(17)=X1(4)
        R(18)=X1(5)
        R(19)=X1(6)
C       CALL POSTP(TIME,1,IPP(1),R(11),R(14),R(17))
C       CALL POSTP(TIME,2,IPP(2),R(12),R(15),R(18))
C       CALL POSTP(TIME,3,IPP(3),R(13),R(16),R(19))
      RETURN
      END
        SUBROUTINE FSCONT(INIT)
        COMMON/VEL/X1(6),XC1(6),Y1(4),R1
        COMMON/VXEL/X2(6),XC2(6),Y2(4),R2
        COMMON/VROLL/X3(4),XC3(4),Y3(2),R3
        DIMENSION EFGH1(16,16),EFGH2(16,16),EFGH3(10,10)
```

```
      DIMENSION XTEMP1(16),YTEMP1(7),XTEMP2(16),YTEMP2(7)
      DIMENSION XTEMP3(10),YTEMP3(5)
      COMMON/MAIN1/NDIM
      EQUIVALENCE (X1(1),XTEMP1(1)),(XC1(1),XTEMP1(7)),
     1            (Y1(1),XTEMP1(13))
      EQUIVALENCE (X2(1),XTEMP2(1)),(XC2(1),XTEMP2(7)),
     1            (Y2(1),XTEMP2(13))
      EQUIVALENCE (X3(1),XTEMP3(1)),(XC3(1),XTEMP3(5)),
     1            (Y3(1),XTEMP3(9))
      IF(INIT.EQ.0)CALL ASSIGN(7,'FSCONT.DAT')
      IF(INIT.GT.0)GO TO 100
C
C     ZERO REDUCED STATE VECTORS X1,X2,X3 AND INPUT VECTOR XC1,XC2,XC3
C
      DO 10 I=1,6
      X1(I)=0.
      XC1(I)=0.
      X2(I)=0.
      XC2(I)=0.
      IF(I.GT.4)GO TO 10
      X3(I)=0.
      XC3(I)=0.
10    CONTINUE
C
C     READ EFGH MATRIX QUADS
C
      READ(7)
      READ(7)IV,((EFGH1(I,J),I=1,7),J=1,16)   !ELEVATION AXIS
      READ(7)
      READ(7)IV,((EFGH2(I,J),I=1,7),J=1,16)   !CROSS ELEVATION AXIS
      READ(7)
      READ(7)IV,((EFGH3(I,J),I=1,5),J=1,10)   !ROLL AXIS
      RETURN
100   CONTINUE
C
C     CALCULATE X(K+1)=AX(K)+BU(K) , Y(K)=CX(K)+DU(K)
C
      NDIM=16
      CALL VMMUL(7,16,EFGH1,XTEMP1,YTEMP1)    !ELEVATION AXIS
      CALL VECTEQ(6,YTEMP1,X1)
      R1=YTEMP1(7)
      CALL VMMUL(7,16,EFGH2,XTEMP2,YTEMP2)    !CROSS ELEVATION AXIS
      CALL VECTEQ(6,YTEMP2,X2)
      R2=YTEMP2(7)
      NDIM=10
      CALL VMMUL(5,10,EFGH3,XTEMP3,YTEMP3)    !ROLL AXIS
      CALL VECTEQ(4,YTEMP3,X3)
      R3=YTEMP3(5)
      RETURN
      END
      SUBROUTINE VMMUL(N1,N2,X,Y,Z)
      DIMENSION X(1),Y(1),Z(1)
      COMMON/MAIN1/NDIM
      DO 10 I=1,N1
      Z(I)=0.
      DO 10 J=1,N2
10    Z(I)=Z(I)+X(I+(J-1)*NDIM)*Y(J)
      RETURN
      END
      SUBROUTINE VECTEQ(N,X,Y)
      DIMENSION X(1),Y(1)
```

```
      DO 10 I=1,N
10    Y(I)=X(I)
      RETURN
      END
      SUBROUTINE POSTP(TIME,J,I,XIN,XMEAN,XRMS)
C
C     THIS SUBROUTINE IS A POST PROCESSOR FOR DETERMINING THE
C     MEAN AND MEAN SQUARE ERROR STATISTICS FOR A GIVEN RANDOM
C     VARIABLE.
C
C     INPUTS: 1)I - FIRST PASS FLAG
C             2)XIN - RANDOM VARIABLE
C             3)J - RV IDENTIFIER NUMBER
C             4)TIME - TIME
C
C     OUTPUTS:1)XMEAN - RANDOM VARIABLE MEAN
C             2)XRMS - RANDOM VARIABLE RMS
C
      DIMENSION I(1),SUM(10),SUM1(10),K(10)
      IF(I(J).EQ.0)GO TO 10
      K(J)=0
      I(J)=0
      SUM(J)=0.
      SUM1(J)=0.
10    IF(TIME.LT.0.)GO TO 20
      K(J)=K(J)+1
      SUM(J)=SUM(J)+XIN
      XMEAN=SUM(J)/K(J)                !COMPUTE MEAN
      SUM1(J)=SUM1(J)+XIN*XIN
      XRMS=SQRT(SUM1(J)/K(J))          !COMPUTE STD DEV
20    CONTINUE
      RETURN
      END
```

DIST3.FOR

```
      SUBROUTINE DIST(TIME,FDIST)
      INCLUDE 'DBP.FOR'
      DIMENSION FDIST(PNUFG),SUM(PNUFG)
      INTEGER*4 ISEED(PNUFG),ISECS,ITSEED
      NN=3
      IF(TIME.NE.0.)GO TO 10
      Y=SECNDS(0.)
      ISECS=IFIX(Y)
      ITSEED=ISECS/2+1
C
C     INITIALIZE RANDOM # GENERATOR SEEDS TO LARGE ODD INTEGER VALUE
C
      DO 10 J=1,NN
      ISEED(J)=ITSEED+2*J
10    CONTINUE
C
C     GAUSSIAN WHITE NOISE SEQUENCE GENERATION -DISTUBANCES 1,2,3,...,NUFG
C
      DO 30 J=1,NN
      SUM(J)=0.
      DO 20 I=1,6
20    SUM(J)=SUM(J)+RAN(ISEED(J))
      FDIST(J)=SQRT(2.)*(SUM(J)-3.)
30    CONTINUE
      RETURN
      END
```

DPOF3CE.DAT

```
$PAR
LMCRLO=.FALSE.          !ENABLE MONTE CARLO ANALYSIS/SENSITIVITY ANALYSIS
LROLL=.FALSE.           !ENABLE AXES: ROLL
LELEV=.FALSE.           !             ELEVATION
LXELEV=.FALSE.          !             CROSS ELEVATION
SFLOSU=.1               !ENABLE ERROR SOURCE PAR.: LOS SCALE FACTOR
BLOSU=20626.4           !                              BIAS
SGLOSU=206264.          !                              NOISE STD DEV
QLOSU=20626.4           !                              QUANTIZATION
SFINTU=.1               !                        : LASER INT. SCALE FACTOR
BINTU=.1                !                                     BIAS
SGINTU=1.               !                                     NOISE STD DEV
QINTU=.1                !                                     QUANTIZATION
SFRGU=.1                !                        : RATE GYRO SCALE FACTOR
BRGU=20626.4            !                                    BIAS
SIGRGU=206624.          !                                    NOISE STD DEV
QRGU=20662.4            !                                    QUANTIZATION
SFTMU=.0                !                        : TORQUE MOTOR SCALE FACTOR
QTMU=.0                 !                                       QUANTIZATION
SIGTMU=0.               !                                       NOISE STD DEV
CCOGU=0.                !                                       COGGING GAIN
CRIPU=0.                !                                       RIPPLE GAIN
C1U=0.                  !                                       SLOPE(DAHL 1)
C2U=0.                  !                                       SLOPE(DAHL 2)
SFLOSN=.01              !NOMINAL ERROR SOURCE    : LOS SCALE FACTOR
BLOSN=.25               !PARAMETERS                         BIAS
SGLOSN=.025             !                                   NOISE STD DEV
QLOSN=.14               !                                   QUANTIZATION
SFINTN=.01              !                      : LASER INT. SCALE FACTOR
BINTN=10.67E-6          !                                   BIAS
SGINTN=5.E-6            !                                   NOISE STD DEV
QINTN=.01E-6            !                                   QUANTIZATION
SFRGN=.000478           !                      : RATE GYRO SCALE FACTOR
BRGN=1.                 !                                  BIAS
SIGRGN=.0148            !                                  NOISE STD DEV
QRGN=.00167             !                                  QUANTIZATION
SFTMN=.043              !                      : TORQUE MOTOR SCALE FACTOR
QTMN=.0618              !                                     QUANTIZATION
SIGTMN=.025             !                                     NOISE STD DEV
CCOGN=.1                !                                     COGGING GAIN
CRIPN=.0275             !                                     RIPPLE GAIN
C1N=2500.               !                                     SLOPE(DAHL 1)
C2N=150.                !                                     SLOPE(DAHL 2)
TO1=.6                  !                                     SATURATION TORQUE(DAHL 1)
TO2=2.                  !                                     SATURATION TORQUE(DAHL 2)
WB=75.                  !                                     ROUGHNESS TORQUE WORST CASE BW
FCOG=96.                !                                     COGGING FREQ.
FRIP=48.                !                                     RIPPLE FREQ.
TDUMAX=9.88,26.37,26.37     !                                 MAX TORQUE OUTPUTS
ALPHA=.008726           !                                     GIMBAL MISALIGNMENT
$END
```

CREW MOTION SUPPLEMENTAL DATA

ORIGINAL PAGE IS
OF POOR QUALITY

Date: February 24, 1986

To: J.F.L. Lee

From: M.J. Kaindl

cc: H. Gelderloos
J. Yeichner
R. Matulenko
W. McGuire

Subject: Stochastic Crew Motion Models For Flexible Space Station Response Analysis

Reference: 1. "Handbook on Astronaut Crew Motion Disturbances for Control System Design", NASA Reference Publication 1025

2. CEL from W. McGuire to MDAC (attn. John Kelly and Dave Barrows) dated 1 October 1985.

Summary:

Two stochastic crew motion disturbance models were generated for use in flexible Space Station response analyses. The models are conservative, simple, and representative, based on data presented in ref. 1. The first model represents a high level crew motion which encompasses all crew activities. The data used to characterize the activities to be recognized as high level is given in ref. 1 under

Bowing Motion
Arm Flapping
Crouch and Pushoff
Soaring
- One-man Normal
- One-man Forceful
- Two-man Normal
- Two-man Forceful

The second model represents crew motion which is restricted to include only low level activities. The activities selected from those analyzed in ref. 1 to characterize this low level are

Console Operations
Respiration, Coughing, Sneezing
Arm Motion
Leg Motion
Swaying Motion

The resulting models are implemented by running the output of a random number generator, with uniform distribution between +/- 1 Newton (+/- .2248 pounds force), through shaping transfer functions, G(s), as shown in figures 1 and 2.

The transfer function parameters for each shaping filter are given in table 1. Notes on implementing these shaping filters in time simulations are given in appendix A, excerpted from chapter 5 of ref. 1.

Assumptions:

1. The force disturbances along all three coordinate axes are similar but uncorrelated.

2. The effects of the moment disturbances resulting from crew motion applied at the habitat module of the Dual Keel I.O.C. are neglegible compared to the effects of the force disturbances from crew motion.

Model Generation:

A. The following criteria were used in selecting the crew motion models .

1. The model must be simple, limited to the form

$$G(s) = \frac{(K)*(2*zeta*omega)*s}{s^2 + (2*zeta*omega)*s + omega^2}$$

where s is the LaPlace variable.

2. The model must be conservative, with the model's PSD at least as strong as the PSD being approximated, at all frequencies.

3. The model must be representative, with a PSD as close to the PSD being approximated as possible while still satisfying criteria 1 and 2.

4. The time histories of the model must have max peaks similar to the max peaks of the data presented in ref. 1.

B. Method of Selecting a Power Spectral Density to Model

The PSD to be modelled in each case was chosen by taking the maximum PSD value over all three axes, for all activities in the set, at each frequency.

C. Method of Selecting a Simple, Conservative, Representative Model for the Power Spectrum

The magnitude of the frequency response of a transfer function of the form in criterion 1 above is shown in figure 3 for various values of damping. From the figure it can be seen that the ratio of "hump max" to "hump width" is a function of damping ratio. "Hump max" is also a function of a constant gain factor. G(s) for a model was determined by

1. Selecting a natural frequency to position the center of the "hump".
2. Selecting a damping ratio to adjust height over width.
3. Selecting a constant gain to adjust the amplitude of the function.

The final values were determined after a few iterations, and are given in table 1.

Table 1. Shaping Filter Transfer Functions

ORIGINAL PAGE IS OF POOR QUALITY

$$G(s) = \frac{(K)*(2*zeta*omega)*s}{s^2 + (2*zeta*omega)*s + omega^2}$$

ORIGINAL PAGE IS
OF POOR QUALITY

	K	zeta	omega
High Level Crew Motion	26.54	1.18	5.48
Low Level Crew Motion	8.46	1.39	2.34

Results:

Figures 1 and 2 compare the modeled PSD's with the actual PSD's. Figures 4 and 5 show time histories of random number generator outputs run through the shaping filters. Figure 6 shows the maximum and average force outputs for the activities studied in ref. 1.

The maximum peaks of figures 4 and 5 are somewhat smaller than the data presented in figure 6. This is partly because figure 6 represents total force and figures 4 and 5 are single axis components. Also, the max peaks from the experiment data sited in figure 6 look to be of high frequency content, and may not be included in the PSD data presented in ref. 1 (0 to 20 rad./sec.).

Recommendation:

Due to the stochastic nature of crew disturbances, the shaping filters given above should be used as a conservative representation in time domain and frequency domain crew disturbance response analyses.

Michael J. Kaindl
M.J. Kaindl
D. and D. Engineer

FIGURE 1

HIGH LEVEL CREW MOTION DISTURBANCE PSD's

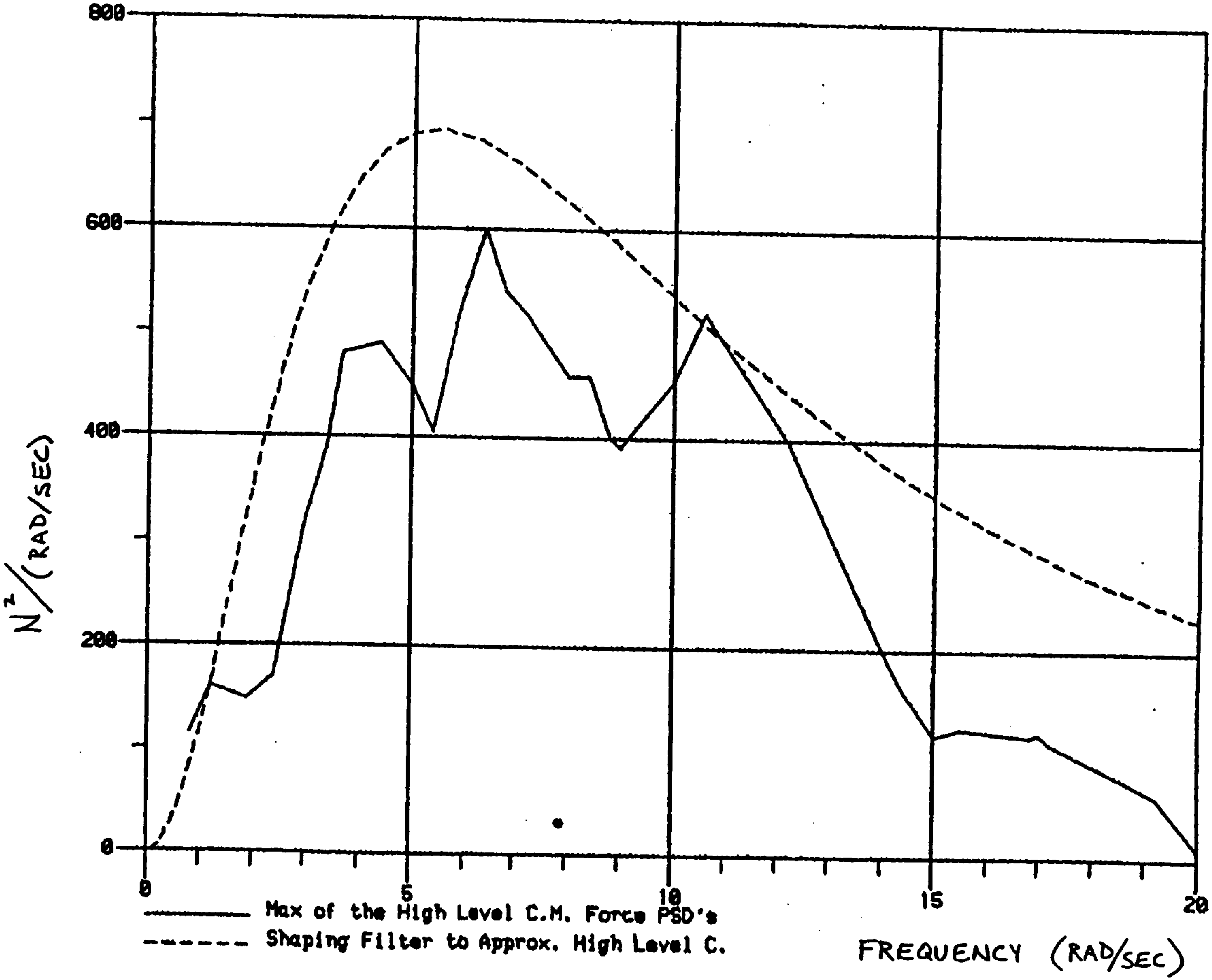


FIGURE 2

LOW LEVEL CREW MOTION DISTURBANCE PSD's

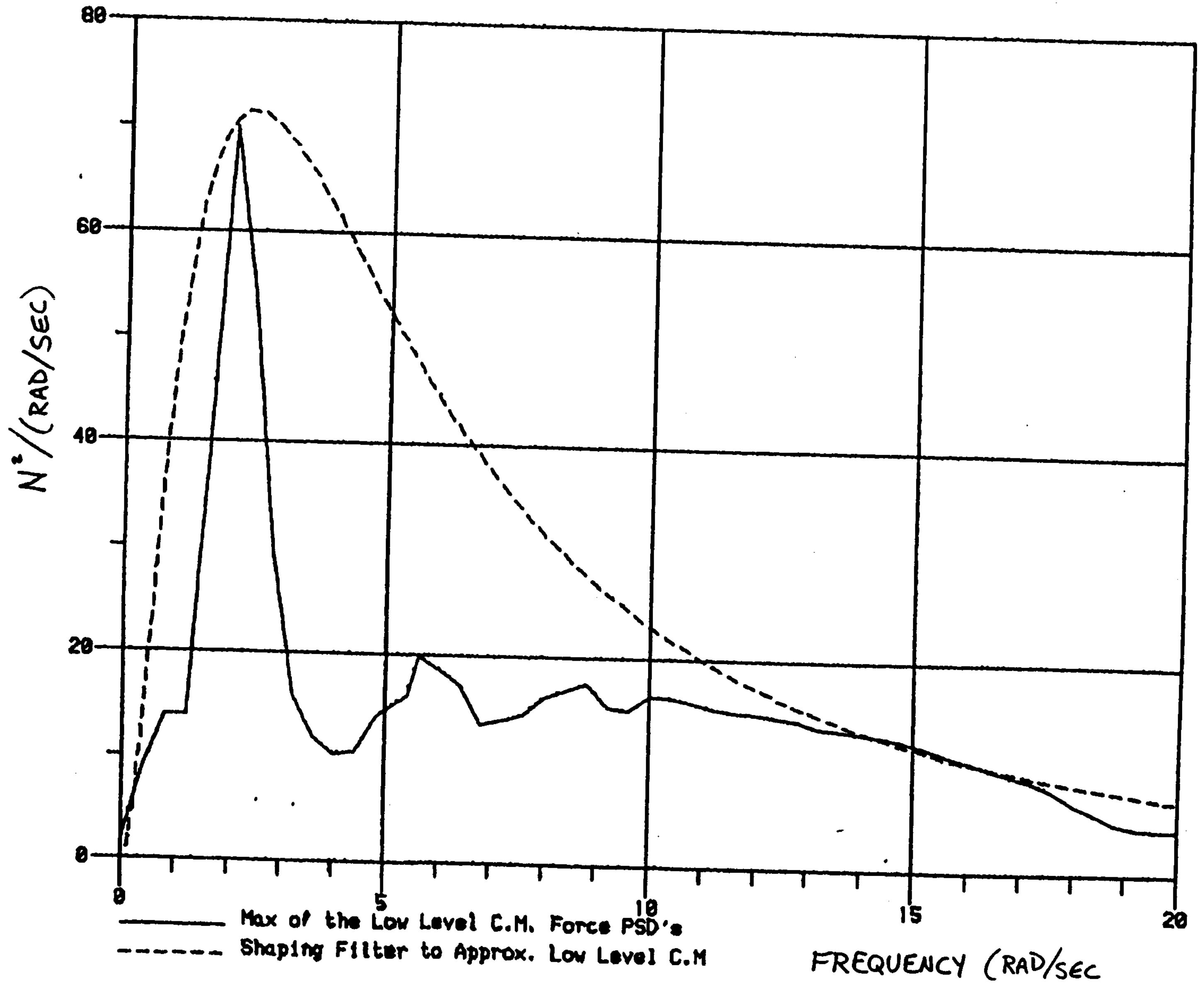
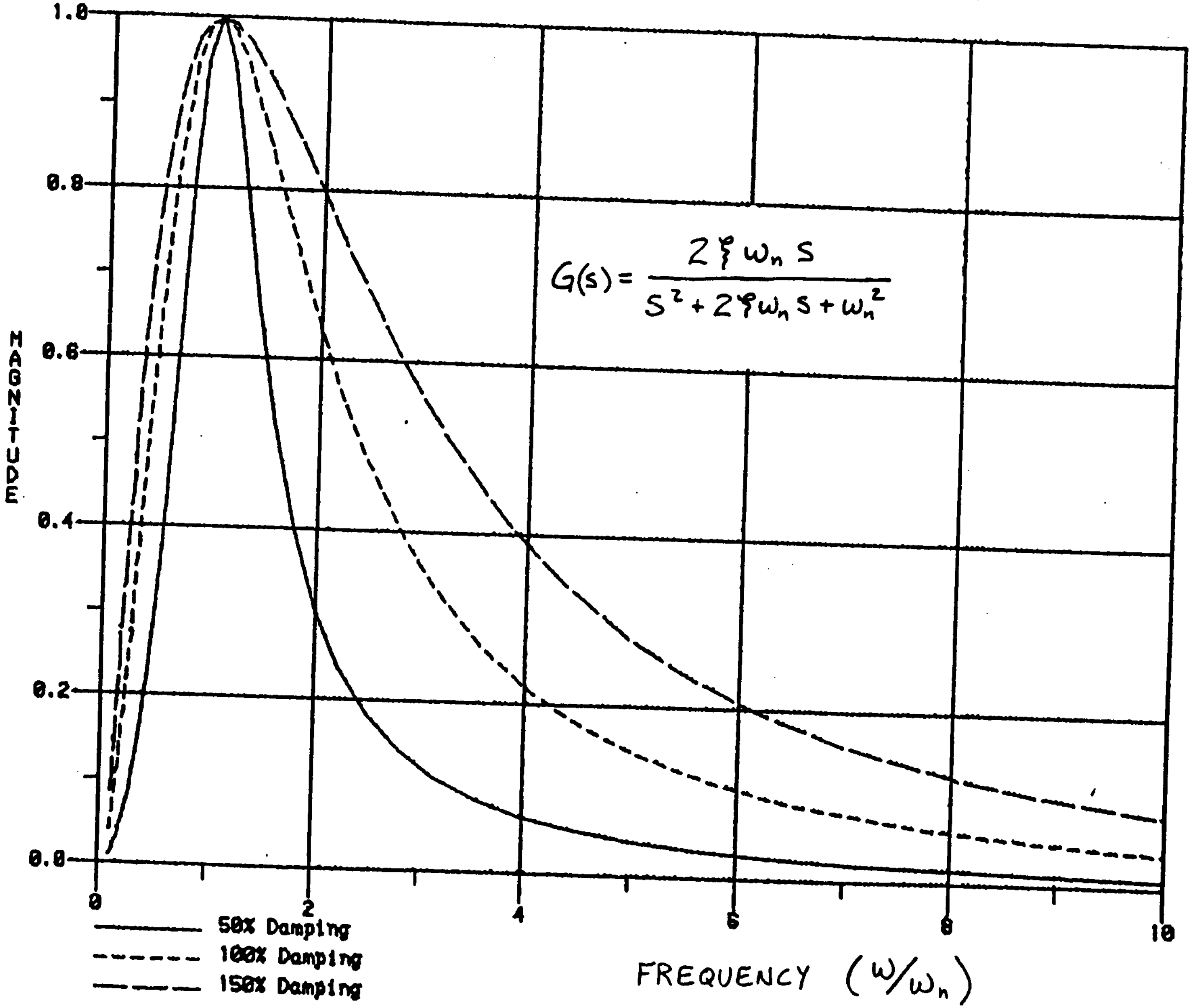


FIGURE 3

2nd ORDER FILTER W/ DIFFERENTIATOR PSD's



$$G(s) = \frac{2\xi\omega_n s}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

FIGURE 4

HIGH LEVEL CREW MOTION DISTURBANCE

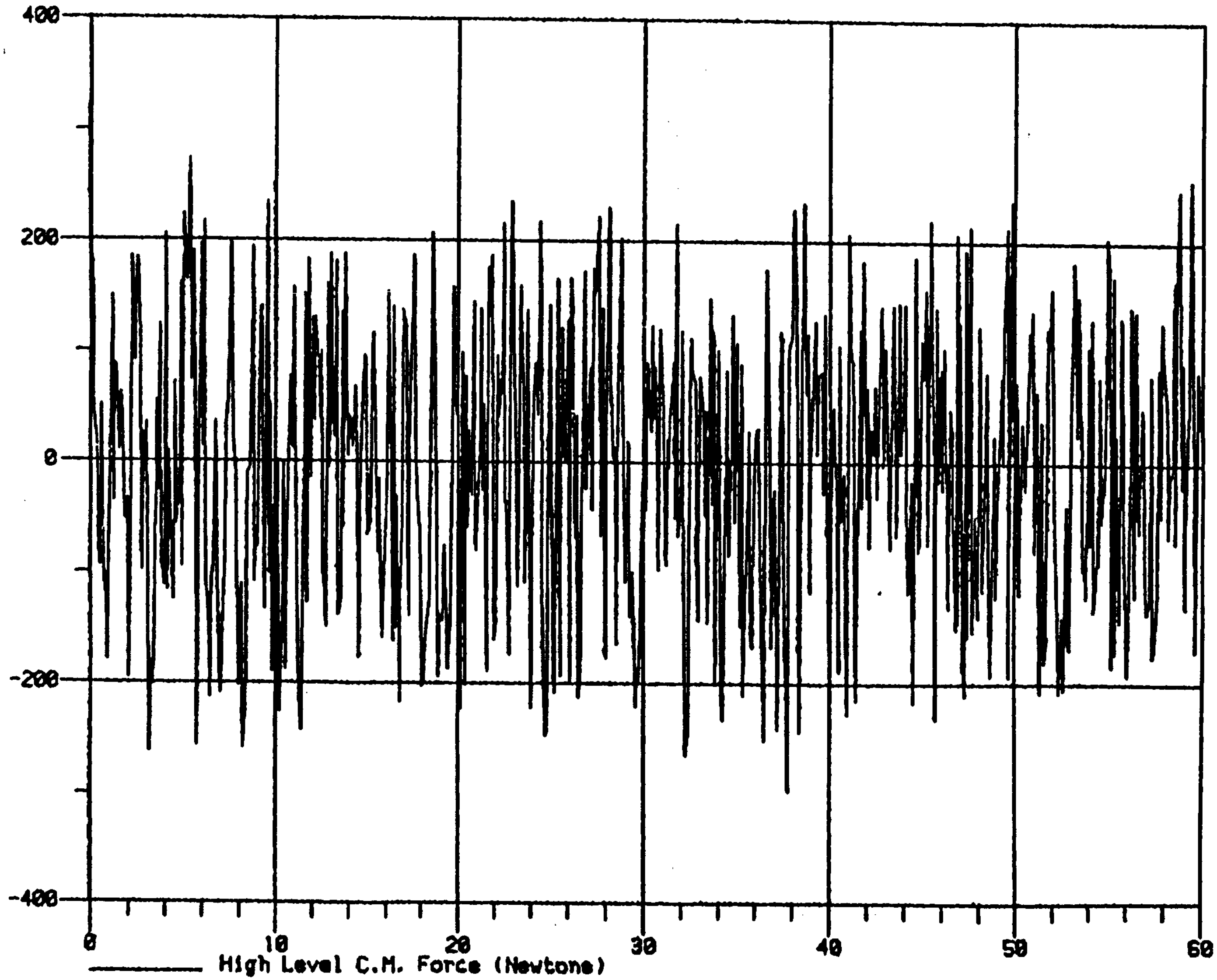
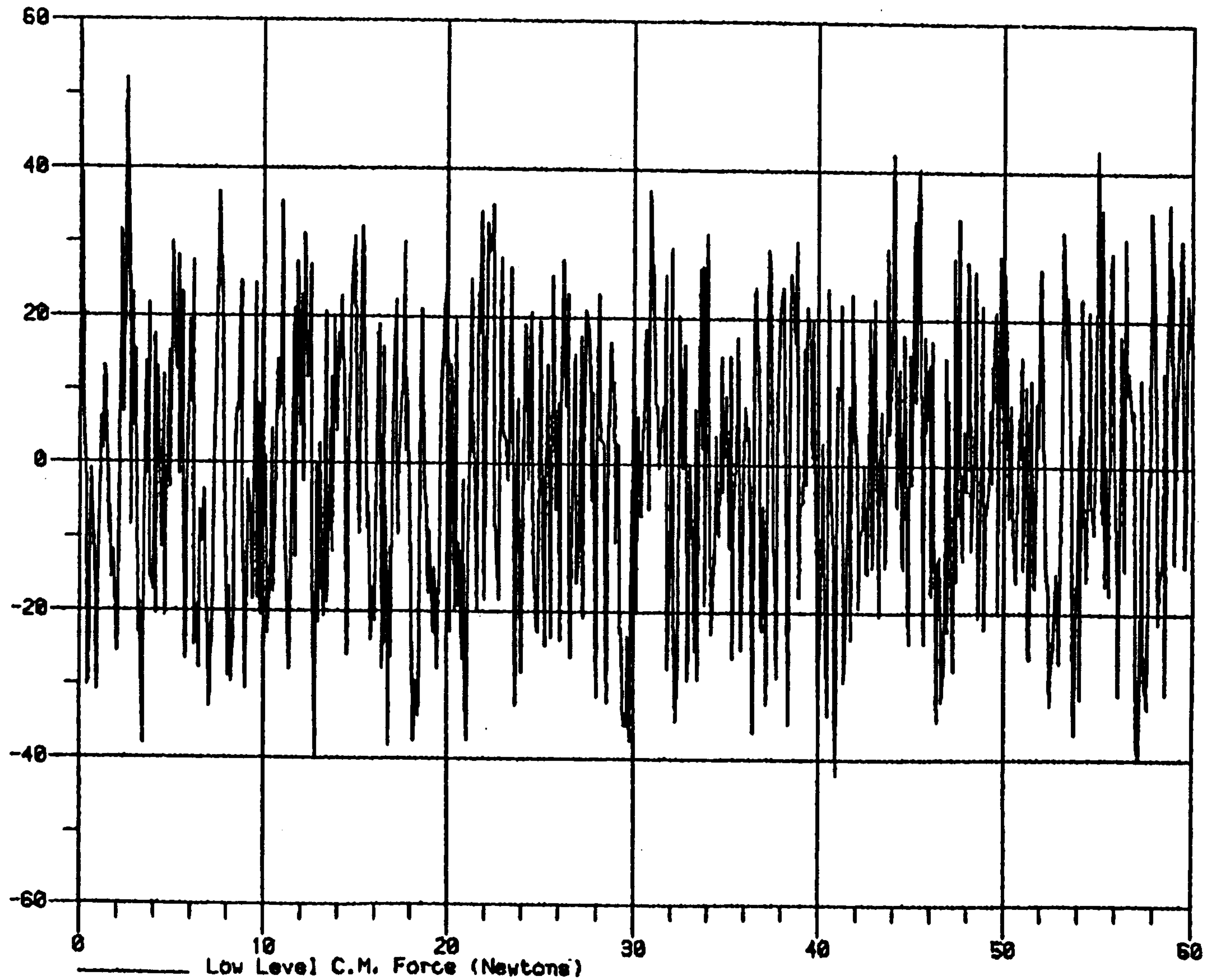


FIGURE 5

LOW LEVEL CREW MOTION DISTURBANCE



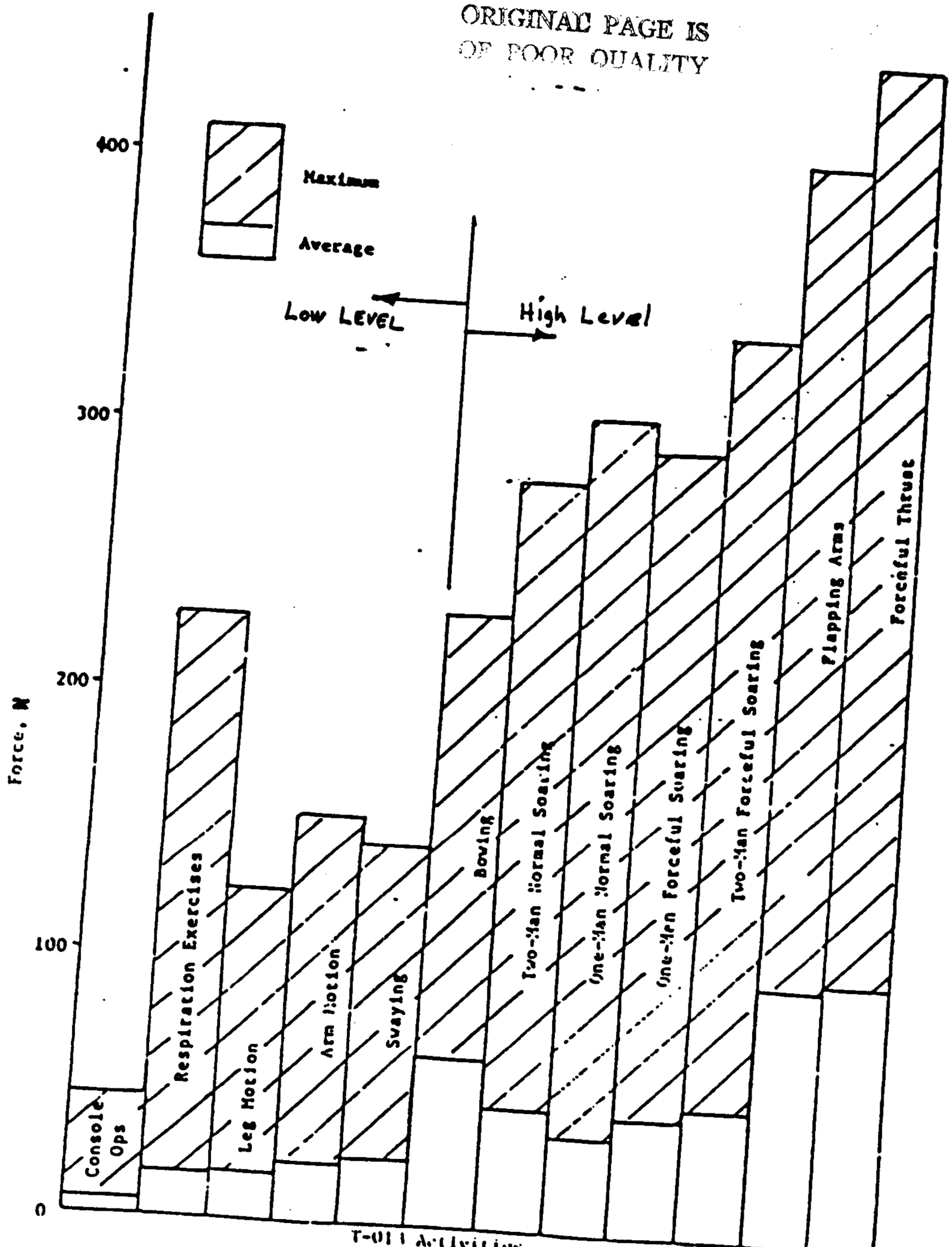


FIG. 6 RESPONSE DATA FROM REF. 1

HONEY-X COMMAND FILES

```
$ TYPE DESIGN.COM
$!*********************************************************
$!                                                        *
$!      PINHOLE "OPTIMAL" CONTROLLER DESIGN COMMAND FILE  *
$!                                                        *
$!      INPUTS:                                           *
$!                                                        *
$!      [KP  KP*KR] ----> IPS.D1   KP PORPORTIONAL IPS GAIN  *
$!      [KP]        ----> IPS.D2   KR IPS RATE GAIN       *
$!      ALPHA       ----> LOS PENALTY FUNCTION VALUE      *
$!      R           ----> CONTROL EFFORT PENALTY VALUE    *
$!      RHO         ----> DISTURBANCE/NOISE WEIGHTING FUNC.  *
$!                                                        *
$!*********************************************************
$!
$!   GENERATING THE REDUCED ORDER PLANT EQUATION WHICH INCLUDES
$!   THE CLOSED LOOP IPS CONTROLLER
$!
$MPY POF3Y_R.B IPS.D1 TEMP
$MPY TEMP POF3Y_R.C1 TEMP
$SUBT POF3Y_R.A TEMP PLT_IPS_R.A
$MPY POF3Y_R.B IPS.D2 PLT_IPS_R.B
$!
$!   GENERATING THE PLANT EQUATION WHICH INCLUDES
$!   THE CLOSED LOOP IPS CONTROLLER
$!
$MPY POF3Y.B IPS.D1 TEMP
$MPY TEMP POF3Y.C1 TEMP
$SUBT POF3Y.A TEMP PLT_IPS.A
$MPY POF3Y.B IPS.D2 PLT_IPS.B
$!
$!    GENERATING THE REGULATOR GAIN MATRIX Kc
$!
$!    WHERE THE COST FUNCTION IS
$!
$!    J = INT 0->00 { (X" BETA" * ALPHA * BETA X)+(X"*ALPHA*BETA*U)+(U"*R*U) dt
$!
$!    INPUTS: (1) SYS_NAME.A
$!
$!            (2) SYS_NAME.B
$!
$!            (3) SYS_NAME.HC   ---> Q MATRIX : STATE "x" WEIGHTING MATRIX
$!                                   Q= ALPHA*BETA
$!
$!            (4) SYS_NAME.HD   ---> R MATRIX : CONTROL "u" WEIGHTING MATRIX
$!                                   R = r
$MPY ALPHA BETA PLT_IPS_R.HC
$COPY R.DAT PLT_IPS_R.HD
$LOGN PLT_IPS_R
$MPY PLT_IPS_R.B PLT_IPS_R.KC TEMP
$SUBT PLT_IPS_R.A TEMP TEMP
$EIG TEMP
$RENAME TEMP.E REGULATOR.E
$RIFD REGULATOR.E
$!
$!    GENERATING THE KALMAN FILTER GAIN MATRIX Ko
$!
$!    INPUTS: (1)SYS_NAME.A
$!
$!            (2)SYS_NAME.C
```

```
$!
$!          (3) Q' = SYS_NAME.V1   "DISTURBANCCE PENALTY FUNCTION"
$!              Q' = DIAG { RHO G}
$!
$!          (4) R = SYS_NAME.V2  "MEASUREMENT NOISE PENALTY FUNCTION"
$!
$!
$MPY GYK RHO TEMP
$DIA TEMP PLT_IPS_R.V1
$KFILT PLT_IPS_R
$MPY PLT_IPS_R.KO PLT_IPS_R.C TEMP
$SUBT PLT_IPS_R.A TEMP TEMP
$EIG TEMP
$RENAME TEMP.E KALMAN.E
$RIFD KALMAN.E
$!
$!   GENERATING THE CLOSED LOOP SYSTEM "Acl" MATRIX
$!   AND CHECKING FOR STABILITY BY FINDING ROOTS
$!   OF THE CLOSED LOOP "Acl" MATRIX
$!
$SCL PLT_IPS.B TEMP -1.0
$MPY TEMP PLT_IPS_R.KC TEMP12
$MPY PLT_IPS_R.KO PLT_IPS.C TEMP21
$MPY PLT_IPS_R.B PLT_IPS_R.KC TEMP221
$MPY PLT_IPS_R.KO PLT_IPS_R.C TEMP222
$SUBT PLT_IPS_R.A TEMP221 TEMP
$SUBT TEMP TEMP222 TEMP22
$!
$! SAVING THE Y-AXIS CONTROLLER
$!
$COPY TEMP22.DAT CONTROLER_Y.DAT
$EIG TEMP22
$RIFD TEMP22.E
$AUG PLT_IPS.A TEMP21 TEMP1 0,1
$AUG TEMP12 TEMP22 TEMP2 0,1
$AUG TEMP1 TEMP2 ACL.A 1,0
$EIG ACL.A
$RIFD ACL.E
$DEL TEMP*.*.*
$!
$!   GENERATING THE OPEN LOOP LOS SYSTEM MATRIX "AOL_LOS"
$!
$!
$SCL PLT_IPS.B TEMP -1.0
$MPY TEMP PLT_IPS_R.KC TEMP12
$MPY PLT_IPS_R.KO C_OL.C TEMP21
$MPY PLT_IPS_R.B PLT_IPS_R.KC TEMP221
$MPY PLT_IPS_R.KO PLT_IPS_R.C TEMP222
$SUBT PLT_IPS_R.A TEMP221 TEMP
$SUBT TEMP TEMP222 TEMP22
$AUG PLT_IPS.A TEMP21 TEMP1 0,1
$AUG TEMP12 TEMP22 TEMP2 0,1
$AUG TEMP1 TEMP2 AOL_LOS.A 1,0
$CON TEMP1 10,1,0
$MPY PLT_IPS_R.KO BDELTA TEMP2
$SEL TEMP2 TEMP21 A 4/P
$SEL TEMP2 TEMP22 A 5
$SEL TEMP2 TEMP23 A 6
$ADD TEMP21 TEMP22 TEMP2
$ADD TEMP2 TEMP23 TEMP2/P
```

```
$AUG TEMP1 TEMP2 AOL_LOS.B 0,1
$SEL PLT_IPS.C TEMP1 4 A
$SEL PLT_IPS_R.C TEMP2 4 A
$SCL TEMP2 TEMP2 0.0
$AUG TEMP1 TEMP2 AOL_LOS.C 1,0
$DEL TEMP*.*.*
$
```

```
$ TYPE ANALYSIS.COM
$!**************************************************************
$!*                                                            *
$!*       PINHOLE ANALYSIS COMMAND FILE                        *
$!*                                                            *
$!*         FUNCTIONS:                                         *
$!*                                                            *
$!*           (1) CREATES CLOSED LOOP STOCHASTIC TIME          *
$!*               RESPONSE DATA                                *
$!*                                                            *
$!*           (2) CALCULATES Mu FOR UNCERTAINTY IN FREQUENCY   *
$!*                                                            *
$!*           (3) CALCULATE THE OPEN LOOP FREQUENCY RESPONSE   *
$!*               FROM LOS command ---> LOS sensed             *
$!*                                                            *
$!*                                                            *
$!*                                                            *
$!*        INPUTS:                                             *
$!*                                                            *
$!*           (1) K, ZETA, OMEGA ---> CREW MOTION SHAPPING     *
$!*                                   FILTER COIFICIENTS       *
$!*                                                            *
$!*           (2) AMPLITUDE ---> CREW MOTION DIST. AMPLITUDE   *
$!*                                                            *
$!*           (3) ACL.A,B,C ---> CLOSED LOOP SYSTEM MATRIX     *
$!*                                                            *
$!*                                                            *
$!*                                                            *
$!*                                                            *
$!*                                                            *
$!*                                                            *
$!**************************************************************
$!
$!    GENERATING THE FREQUENCY RESPONSE FOR OPEN LOOP LOS
$!
$COPY [PINHOLEA.HBW.D]AOL_LOS.* *.*
$!SCL AOL_LOS.C AOL_LOS.C -1.0
$PACK AOL_LOS AOL_LOS
$FRESP AOL_LOS 0.001,100,50
$!
$!   GENERATING THE CLOSED LOOP TIME RESPONSE TO STOCASTIC
$!   CREW MOTION
$!
$COPY [PINHOLEA.HBW.D]ACL.A *.*
$COPY [PINHOLEA.HBW.D]PLT_IPS.* *.*
$CON TEMP31 2,8,0
$CON TEMP32 2,6,0
$COPY [PINHOLEA.HBW.D]GZ.DAT *.*
$MPY GZ DIST.C TEMP13
$CON TEMP23 6,2,0
$AUG TEMP31 TEMP32 TEMP2 1,0
$AUG ACL.A TEMP2 TEMP1 0,1
$AUG TEMP13 TEMP23 TEMP2 0,1
$AUG TEMP2 DIST.A TEMP2 0,1
$AUG TEMP1 TEMP2 SYS_D.A 1,0
$CON TEMP 14,1,0
$AUG TEMP DIST.B SYS_D.B 0,1
$CON TEMP 4,8,0
$AUG PLT_IPS.C TEMP SYS_D.C 1,0
$PACK SYS_D SYS_D
```

```
$THIST SYS_D 40,0.01,0.01
$!
$!    GENERATING THE CLOSED LOOP TIME RESPONSE TO STOCASTIC
$!    CREW MOTION "HIGH LEVAL" ----> DISTH
$!
$COPY [PINHOLEA.HBW.D]ACL.A *.*
$COPY [PINHOLEA.HBW.D]PLT_IPS.* *.*
$CON TEMP31 2,8,0
$CON TEMP32 2,6,0
$COPY [PINHOLEA.HBW.D]GZ.DAT *.*
$MPY GZ DISTH.C TEMP13
$CON TEMP23 6,2,0
$AUG TEMP31 TEMP32 TEMP2 1,0
$AUG ACL.A TEMP2 TEMP1 0,1
$AUG TEMP13 TEMP23 TEMP2 0,1
$AUG TEMP2 DISTH.A TEMP2 0,1
$AUG TEMP1 TEMP2 SYS_DH.A 1,0
$CON TEMP 14,1,0
$AUG TEMP DISTH.B SYS_DH.B 0,1
$CON TEMP 4,8,0
$AUG PLT_IPS.C TEMP SYS_DH.C 1,0
$PACK SYS_DH SYS_DH
$THIST SYS_DH 40,0.01,0.01
$!
$!     GENERATING THE CLOSED LOOP STEP RESPONSE TO FOR THETAips =.5 degrees
$!     and n1 = 2 degrees
$!
$COPY SYS_DH.DAT SYS_X0.DAT
$THIST SYS_X0 25,0.05,0.01
$DEL *.X.*
$DEL *.XD.*
$!
$!    GENERATING Mu FOR FREQUENCY UNCCERTAINTY
$!
$COPY ACL.A DELTAZ.A
$PACK DELTAZ SYS_MU
$FRESP SYS_MU 0.1,100,50
$DET SYS_MU.G
$DEL TEMP*.*.*
$
```