(NASA-CR-178875) MODELING AND ANALYSIS OF PINHOLE OCCULTER EXPERIMENT Final Report (Honeywell, Inc.) 84 p CSCL 14B

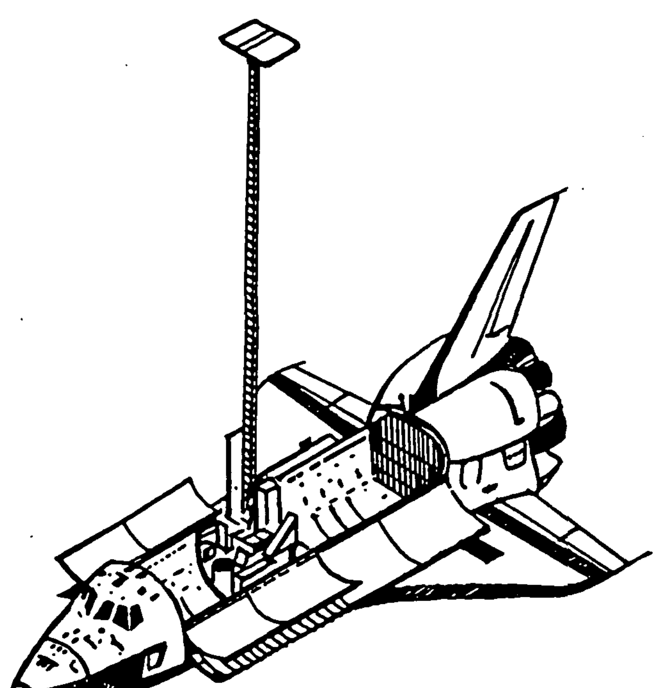
N86-30807

Unclas
G3/19 43385

Modeling and Analysis
of
PINHOLE OCCULTER EXPERIMENT

Contract No. NAS8-36101

FINAL REPORT



Honeywell

SPACE & STRATEGIC
AVIONICS DIVISION
CLEARWATER, FLORIDA

MODELING AND ANALYSIS

of

PINHOLE OCCULTER EXPERIMENT

Contract No. NAS8-36101

FINAL REPORT

Prepared by: J. R. RING

Date: JUNE 20,1986

TABLE OF CONTENTS

1.0 INTRODUCTION

CONTRACT OBJECTIVES:

- IMPROVED POINTING CONTROL SYSTEM IMPLEMENTATION- Convert the dynamic compensator from a continuous domain representation to a discrete one

- DETERMINE POINTING STABILITY SENSITIVITIES TO SENSOR AND ACTUATOR ERRORS- Add sensor and actuator error models to TREETOPS. Develop an error budget for meeting pointing stability requirements

- DETERMINE POINTING PERFORMANCE FOR ALTERNATE MOUNTING BASES- Space Station for example

2.0 CONCLUSIONS

- 25 Hz sample rate is recommended for the P/OF dynamic compensator algorithm processing
- Pointing accuracy varies linearly with error source magnitude
- Fixed gyro drift is the largest contributor to pointing inaccuracy
- A root sum square(rss) LOS error of 1 arc sec can be met provided the IPS fixed gyro drift is reduced by 34%
- The P/OF doesn't perform well with the Space Station as the mounting base and with the current controller.

3.0 RECOMMENDATIONS

- Improve IPS model fidelity - Model IPS as 3 single DOF hinges separated by two bodies. Model dynamics of IPS operational controller. Implement both models in TREETOPS.

- Controller Design - Specify IPS operational controller software changes and any external processing necessary to accommodate the P/OF payload.

- Performance Assessment - Specify transient, steady state and robustness characteristics of baseline(orbiter) pointing system. Include an assessment of the P/OF performance utilizing an alternate mounting base.

4.0 DISCUSSION

4.1 REFERENCE DIGITAL CONTROLLER STRUCTURE

- TASK: Convert the Linear Continuous Time Compensator shown on page 6 to an equivalent Linear Discrete Time Dynamic Compensator. Note that the discrete-time dynamic compensator is represented by a set of first-order difference equations instead of a set of first-order differential equations, as in the continuous-time dynamic compensator. Also, in comparison with the continuous-time systems, discrete-time systems utilize summations instead of integrations.

- OBJECTIVE: Reduced computer resources required for implementation

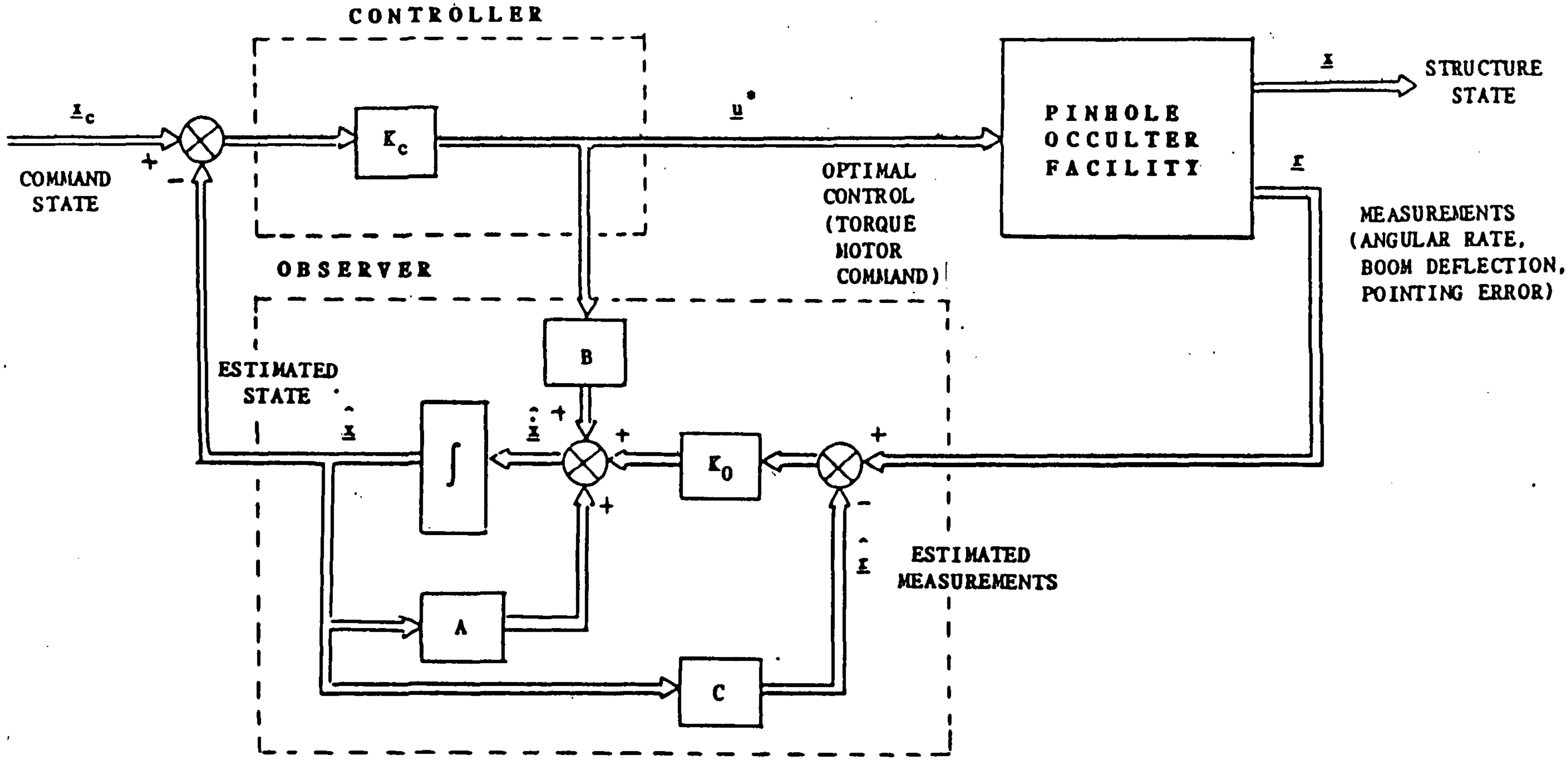
- PROCEDURE:

Step 1) Express compensator in the form

$$\dot{\hat{X}} = A'\hat{X} + B'U$$

$$Y = C'\hat{X} + D'U$$ (see page 7)

Step 2) Use the control system analysis tool capabilities in the digital computer program DIGIKON to obtain the transition matrices F,G,H,E that describe the discrete equivalent of the continuous system *described in step 1.*

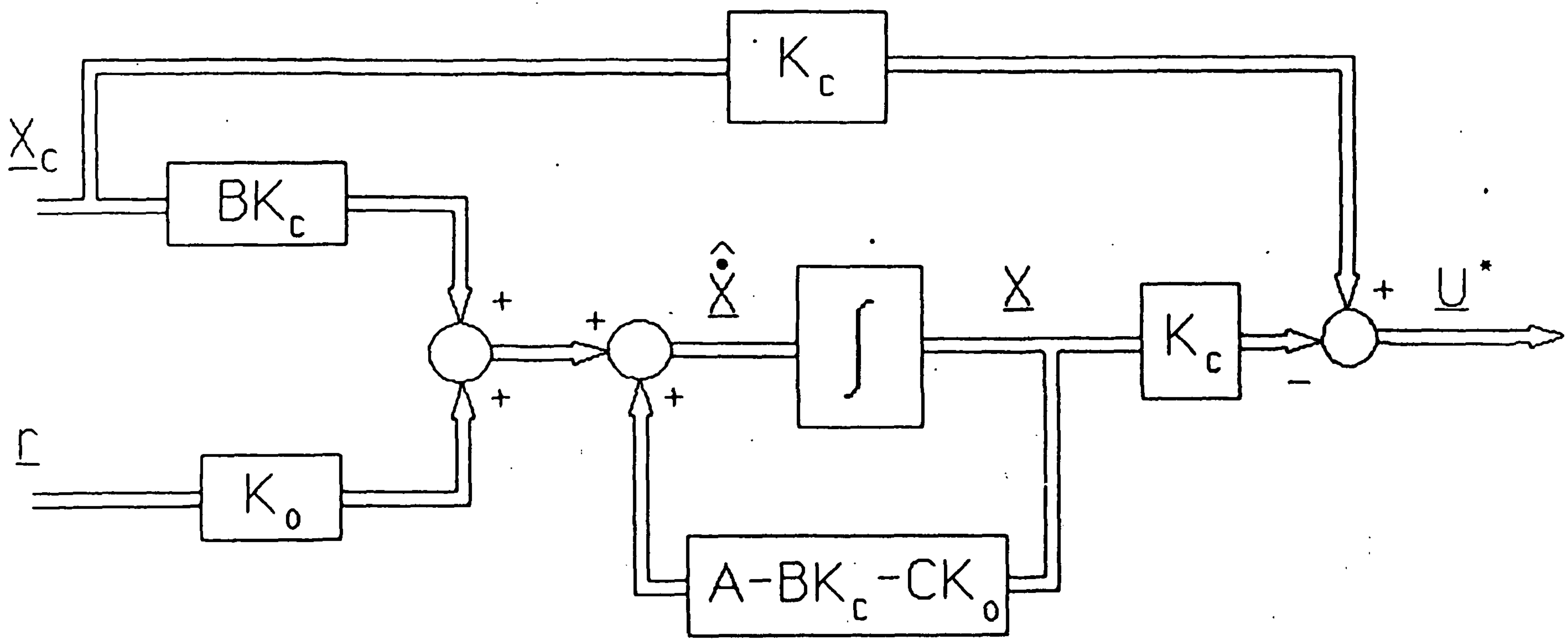


FULL STATE CONTROLLER

for

P/OF LINE-OF-SIGHT POINTING

P/OF FSC STATE SPACE MODEL EQUIVALENT



$A' = (A - BK_c - K_0)$

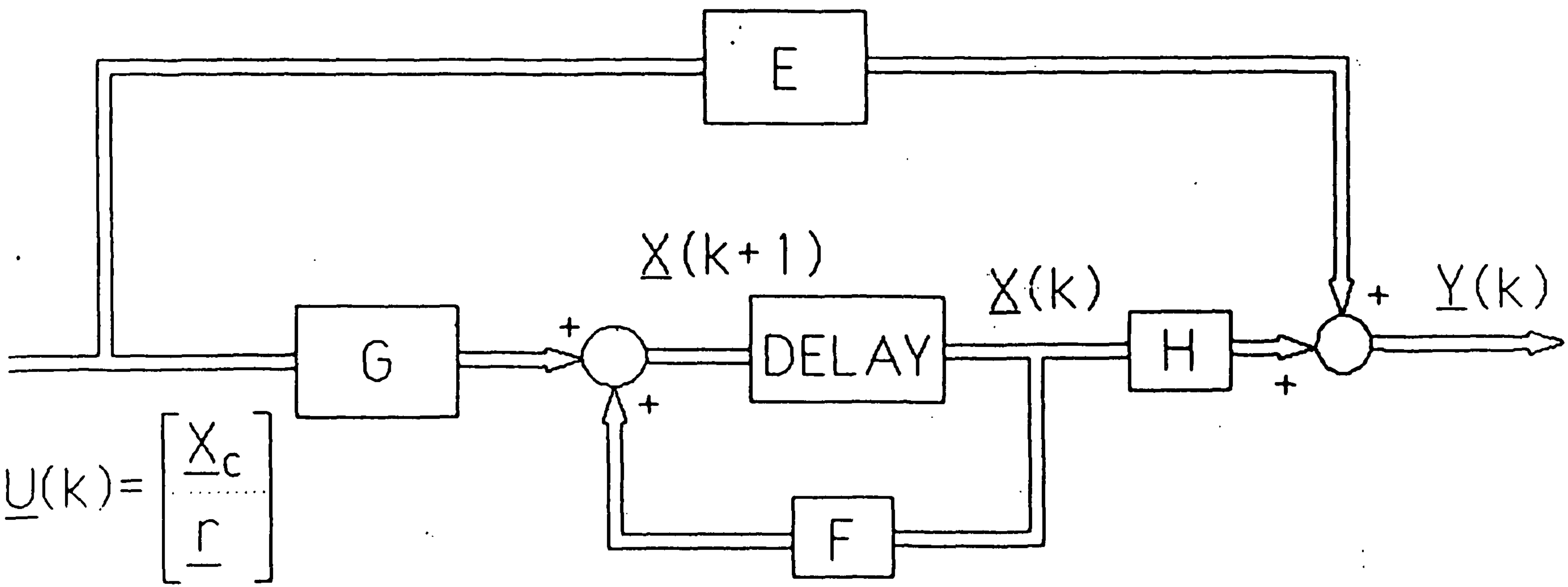
$B' = (BK_c \vdots K_0)$

$C' = -K_c$

$D' = (K_c \vdots 0)$

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

P/OF FSC DISCRETE STATE SPACE MODEL EQUIVALENT



REFERENCE DIGITAL CONTROLLER SAMPLE RATE

- TASK: Select a sample rate for the digital controller

- OBJECTIVE: Negligible degradation in pointing performance due to digital implementation

- PROCEDURE: Determine pointing performance sensitivity to controller sample rate by performing the following steps

  Step 1) Obtain the F,G,H,E transition matrices that describe the discrete controller for sample rates of 25, 12.5, 6.25, and 3.125 Hz (DIGIKON-EXPO)

  Step 2) Obtain time history of digital controllers defined in step 1 for a 1 deg line of sight step input command. (DIGIKON-TRANS) Refer to pages 11 and 12. The DIGIKON-TRANS) program was used instead of TREETOPS for this time history study primarily to reduce costs. Refer to pages 11 and 12. The P/OF with the 3.125 Hz digital controller was unstable.

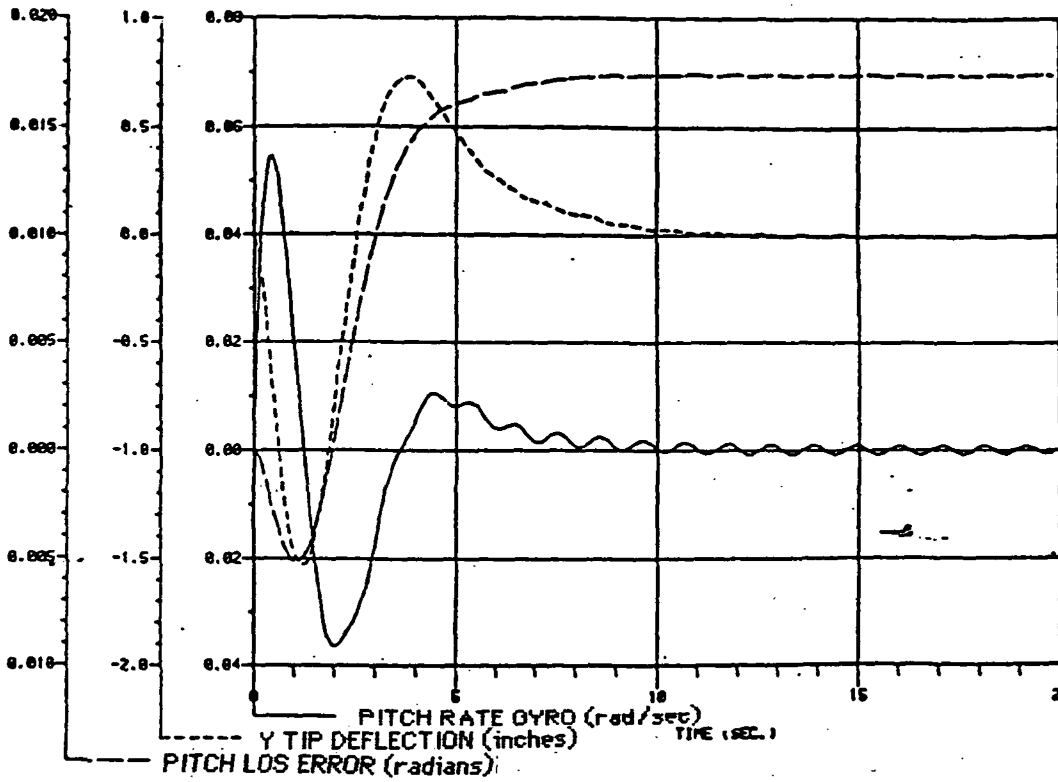
  Step 3) Select a sample rate from the plots obtained in step 2. The 25 Hz controller was selected because its performance is nearly identical with the continuous time controller. The IPS processor also runs at 25Hz.

Step 4) Replace the continuous controller modeled in the TREETOPS "user controller" with the 25 Hz discrete time controller.

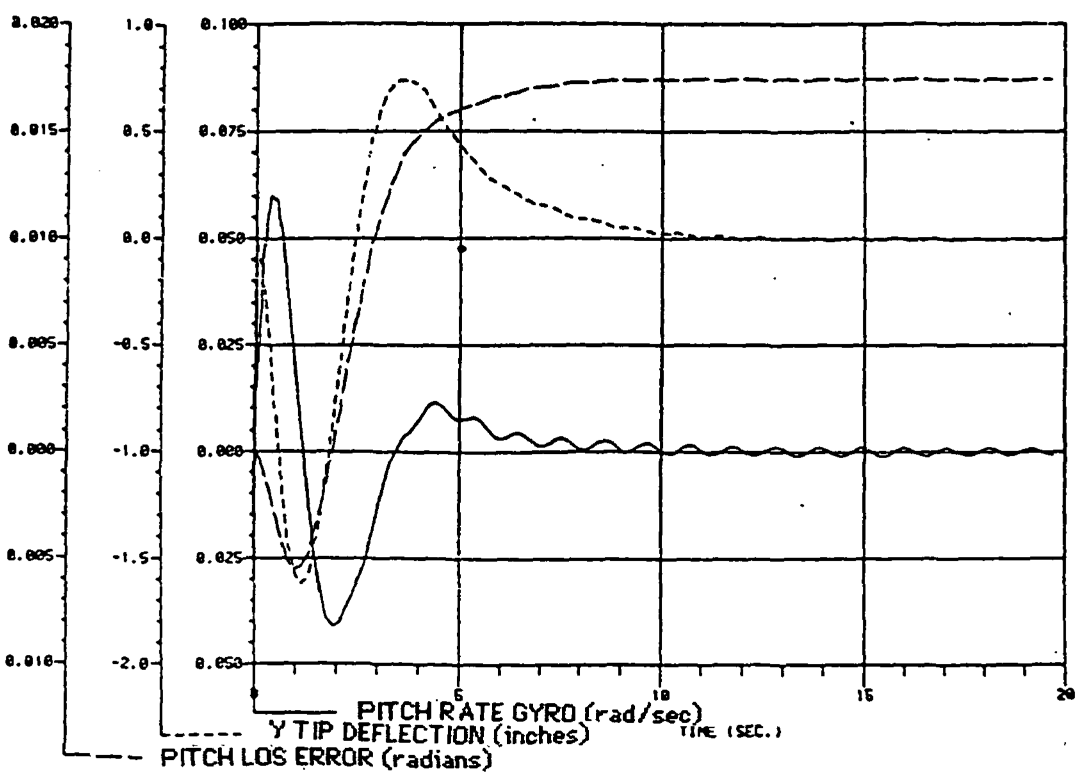
Step 5) Compare the 25 Hz discrete controller pointing performance with the continuous controller performance for a worse case Shuttle attitude command disturbance. Generate TREETOPS time histories for both controllers. See pages 13 through 16. The pitch torque motor output on pages 14 and 16 reveals a very slightly divergent mode. However, these results were obtained assuming perfect actuators. When torque motor non-linearities(quantization, friction, etc.) and dynamics are added, this mode doesn't diverge.

DIGIKON TIME HISTORIES

POF AND CONTINUOUS CONTROLLER

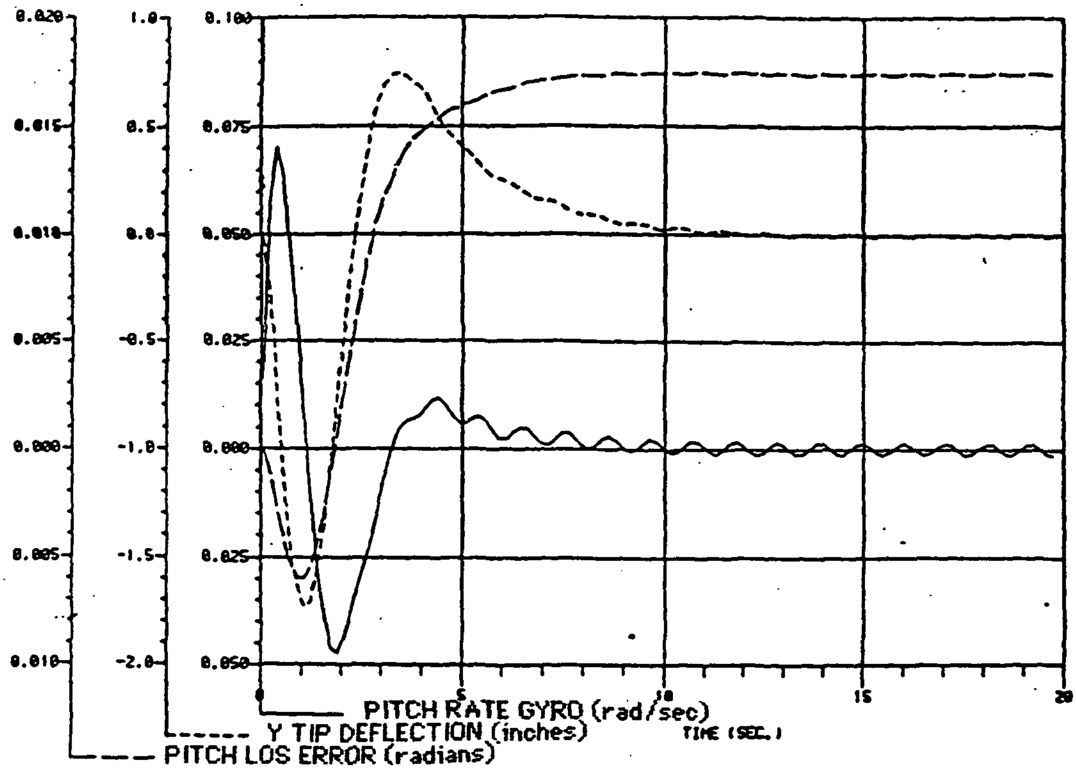


POF AND 25 H_Z CONTROLLER

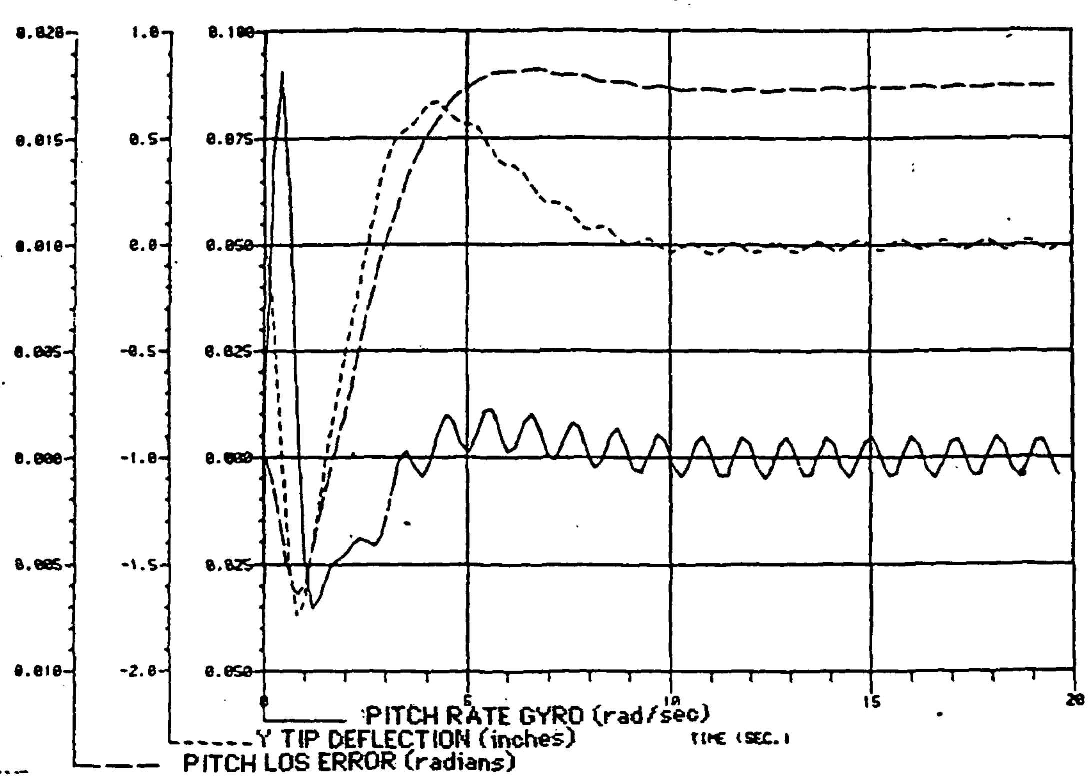


DIGIKON TIME HISTORIES

POF AND 12.5 Hz CONTROLLER

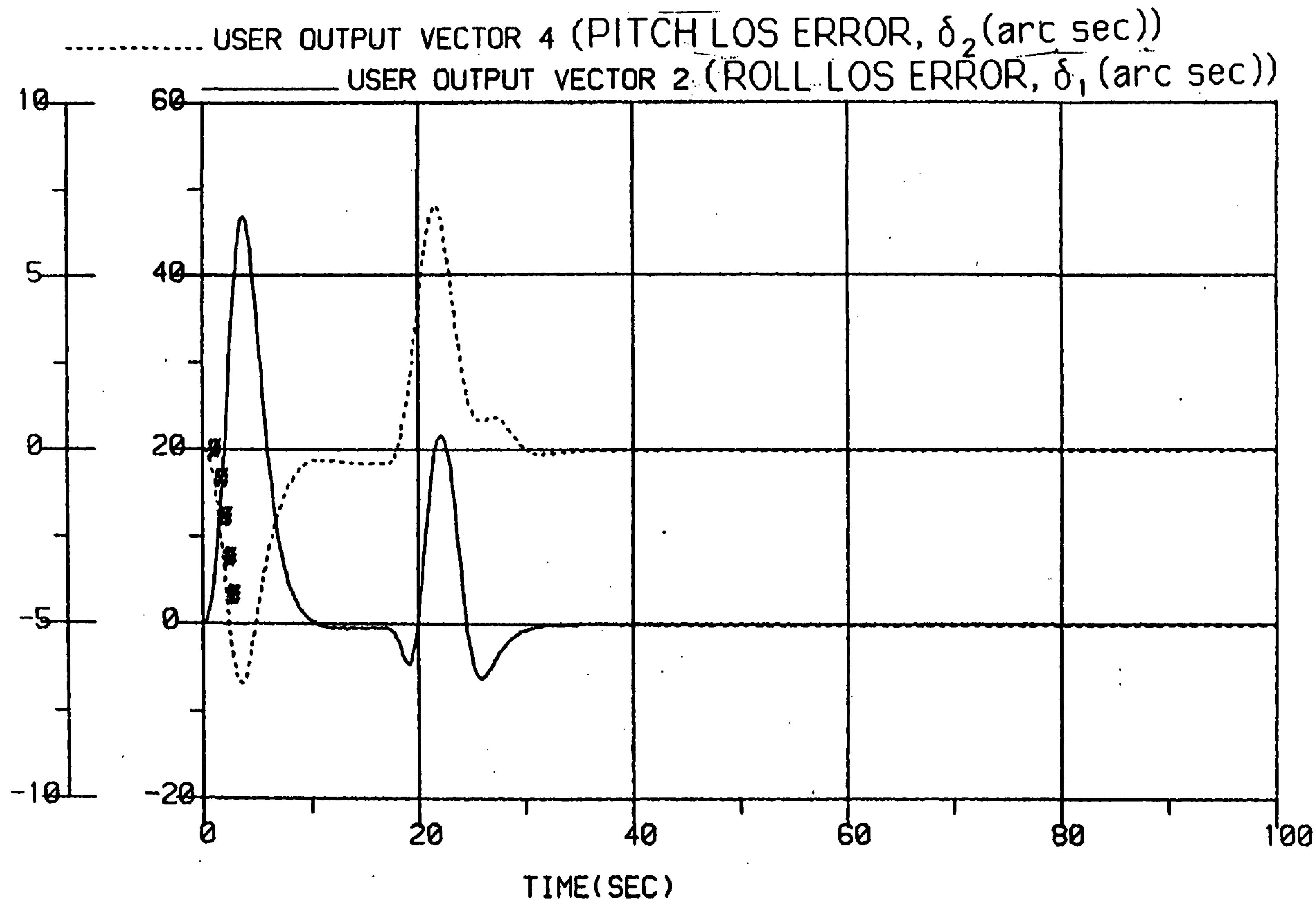


POF AND 6.25 Hz CONTROLLER



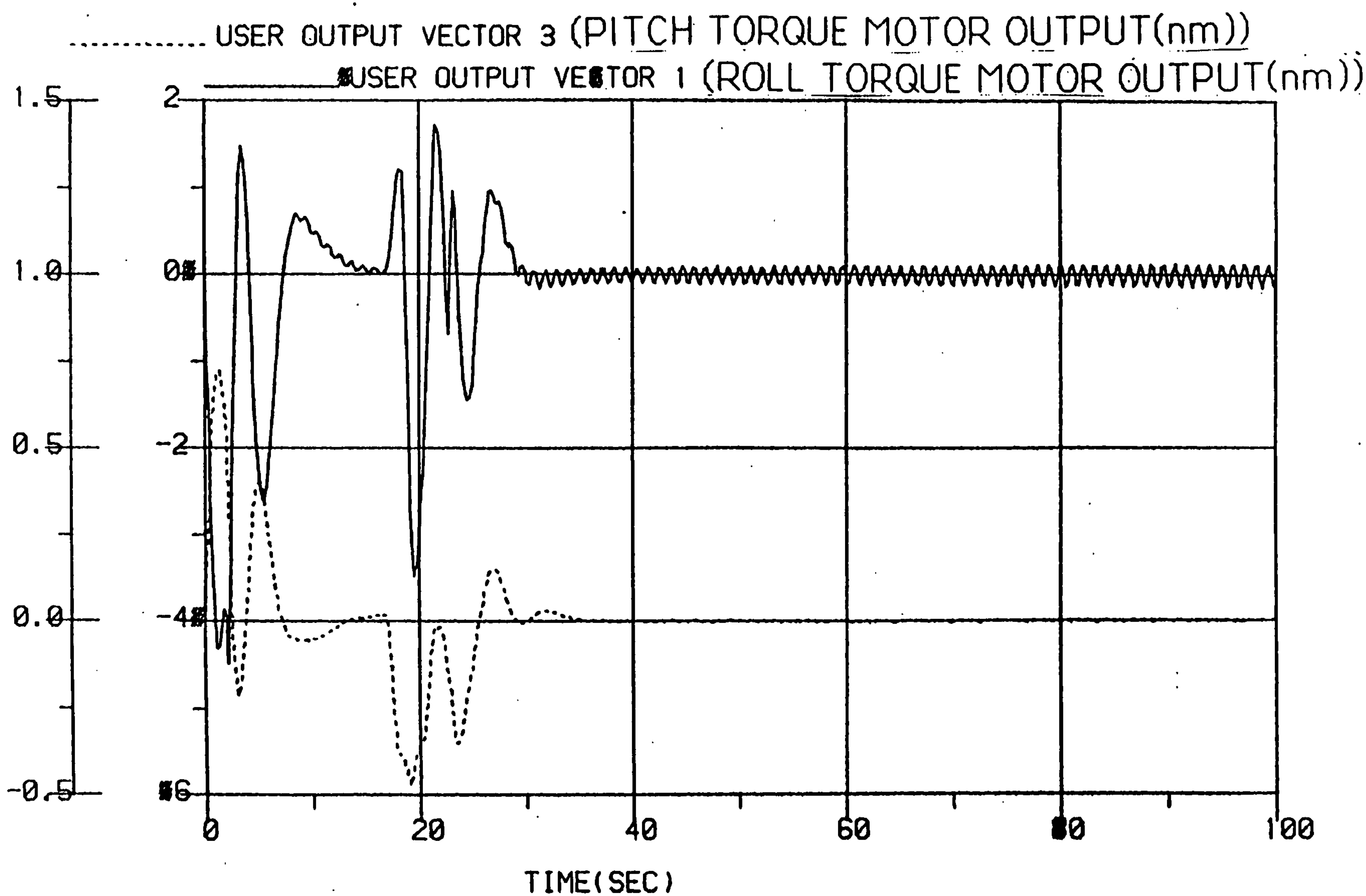
TREETOPS TIME HISTORY

P/OF WITH CONTINUOUS CONTROLLER



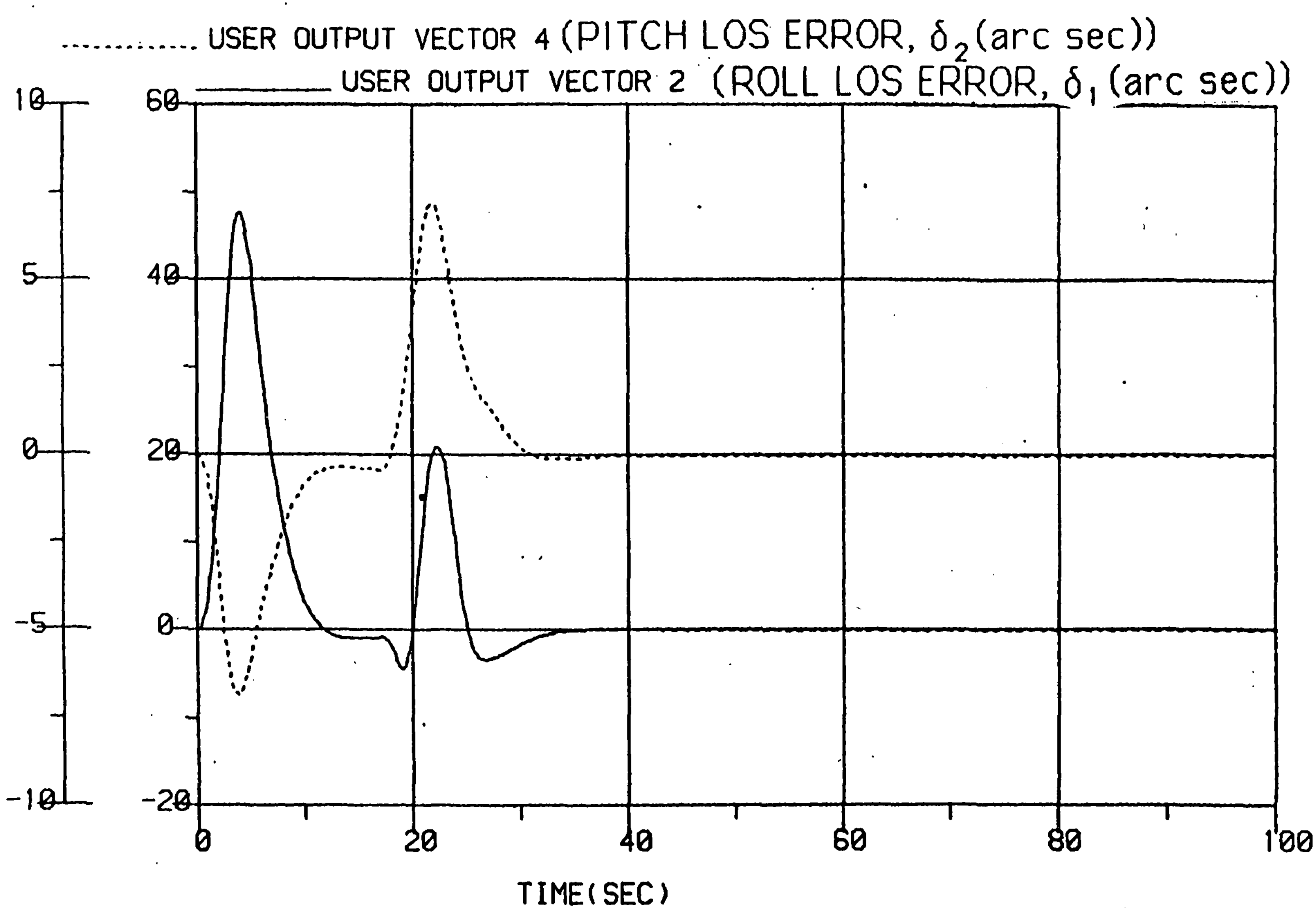
TREETOPS TIME HISTORY

P/OF WITH CONTINUOUS CONTROLLER



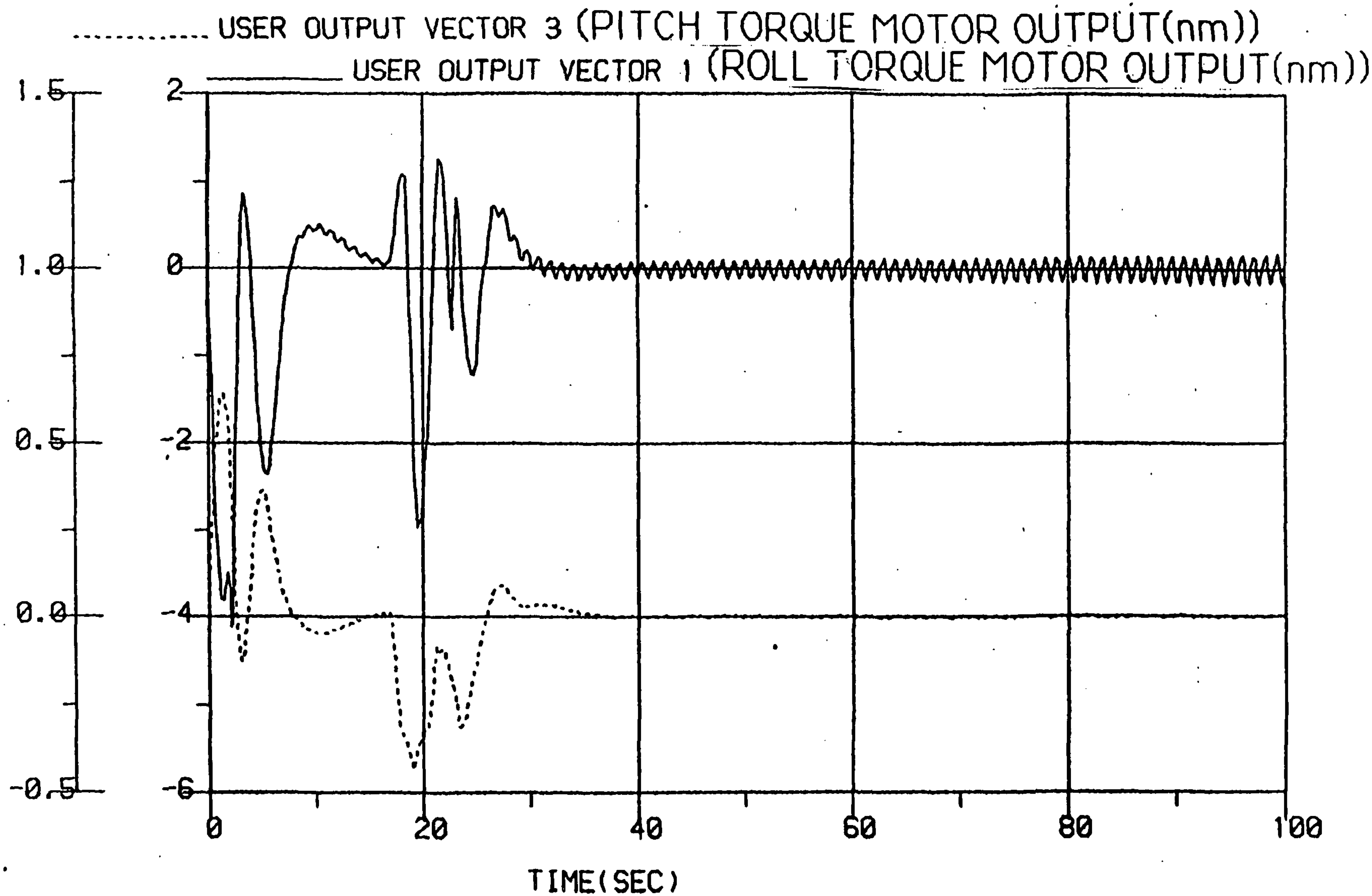
(15) TREETOPS TIME HISTORY

P/OF WITH 40 msec DIGITAL CONTROLLER



(16) TREETOPS TIME HISTORY

P/OF WITH 40 msec DIGITAL CONTROLLER



4.2 ERROR MODELS

- TASK - Define sensor and actuator error models to be used for parameter sensitivity studies

- OBJECTIVE - Math models that can be implemented into TREETOPS

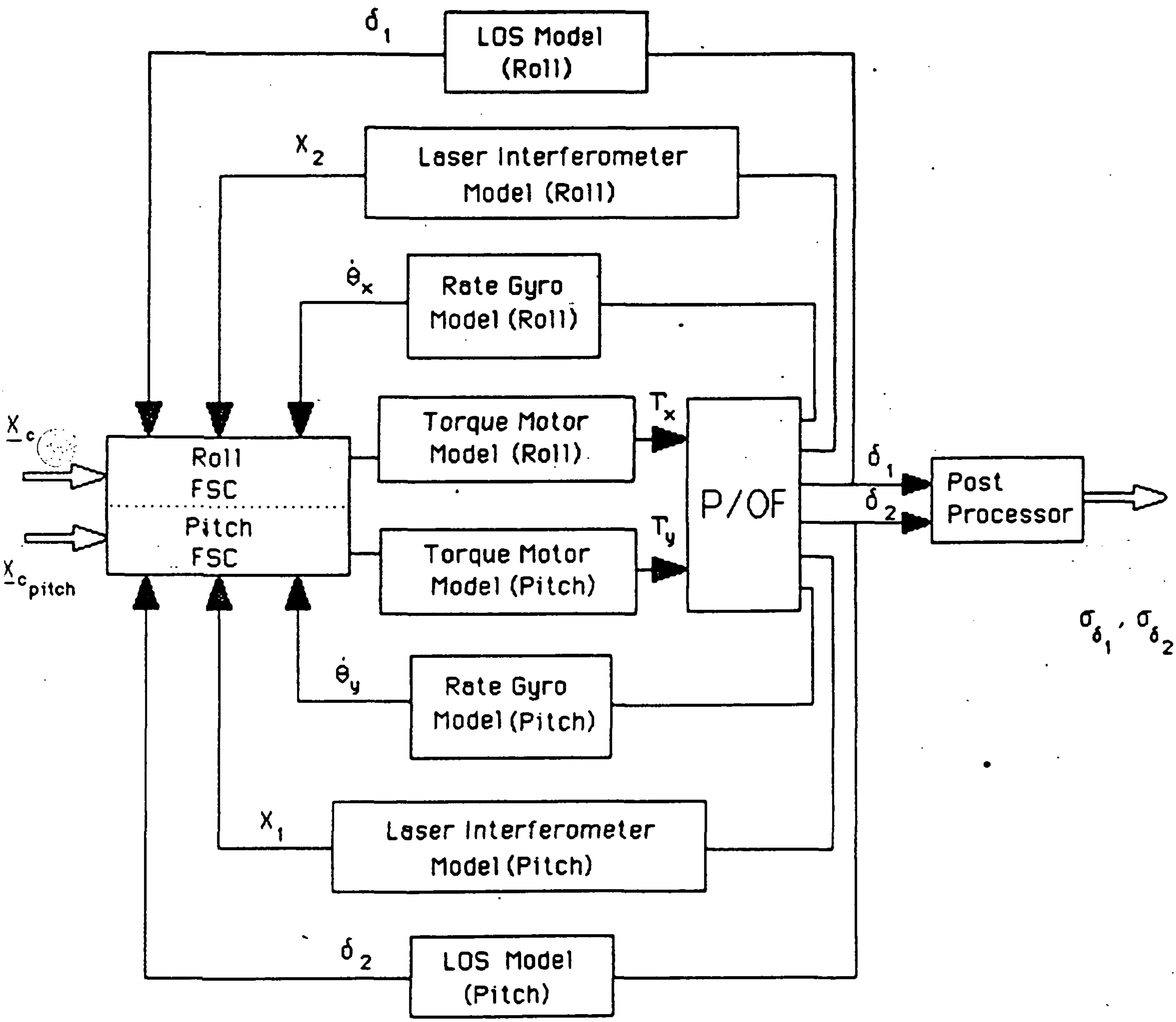
- COMMENTS -

  1. The P/OF full state controller utilizes measurements from 3 sensors. The LOS sensor measures line of sight error between a target vector and the vector which connects the base of the boom to its tip. The Laser Interferometer measures the relative displacement of the mask wrt the detector due to elastic deformation. The rate gyro measures angular rates at the instrument mounting plate. Control actuation is provided by torque motors. Refer to page 18.

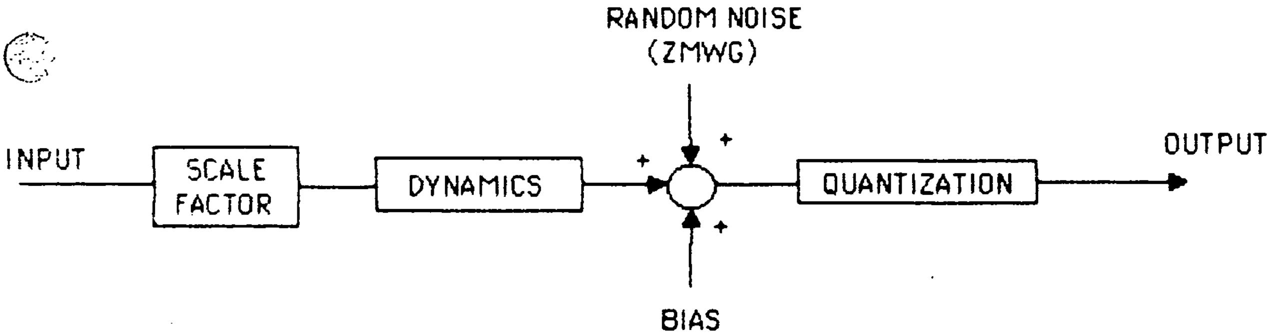
  2. A functional block diagram of the error sources considered for each Sensor/Actuator error model is shown on page 19.

  3. Pages 20 through 24 contain the details on the Sensor/ Actuator math models that are implemented in the TREETOPS simulation. Nominal values for the error model parameters are supplied along with the sources of these values.

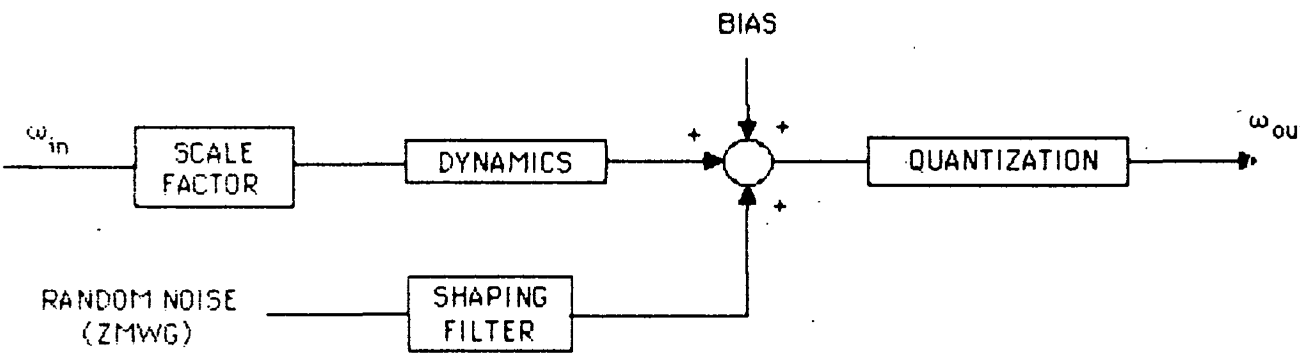
P/OF FULL STATE CONTROLLER WITH SENSOR/ACTUATOR ERROR MODELS



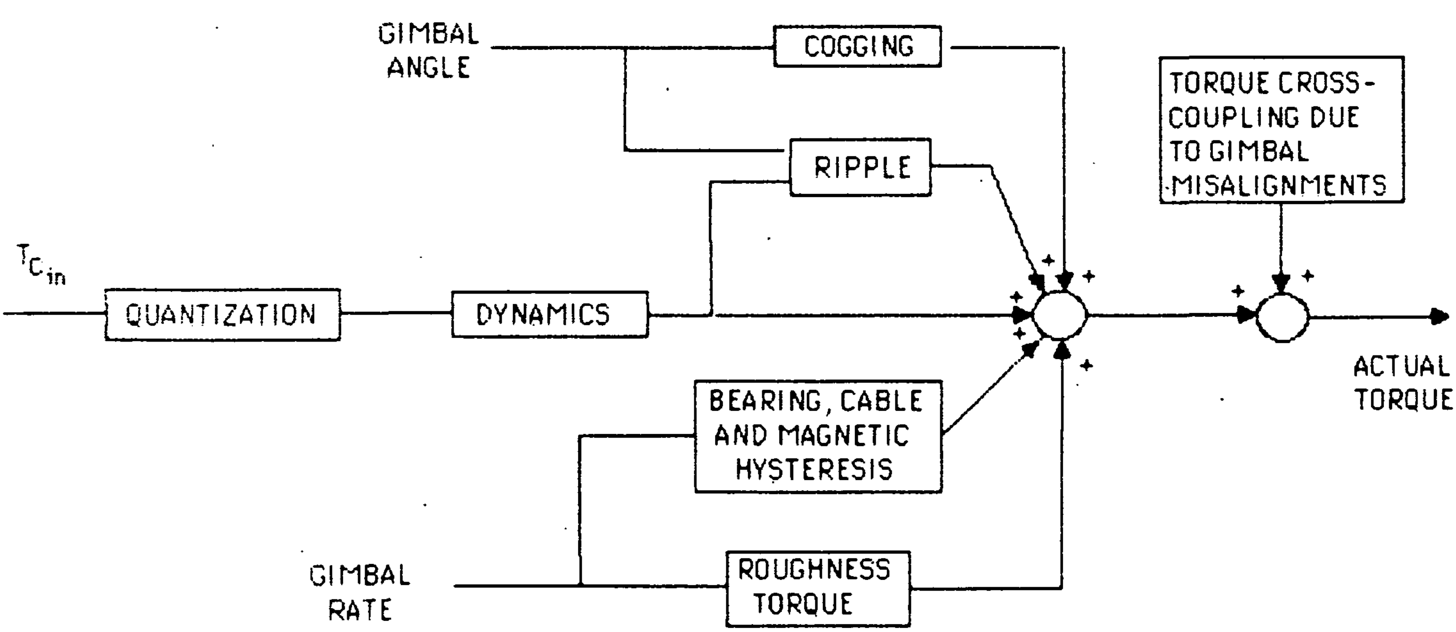
SENSOR/ACTUATOR ERROR MODEL STRUCTURE



LOS & LASER INTERFEROMETER ERROR SOURCE MODEL



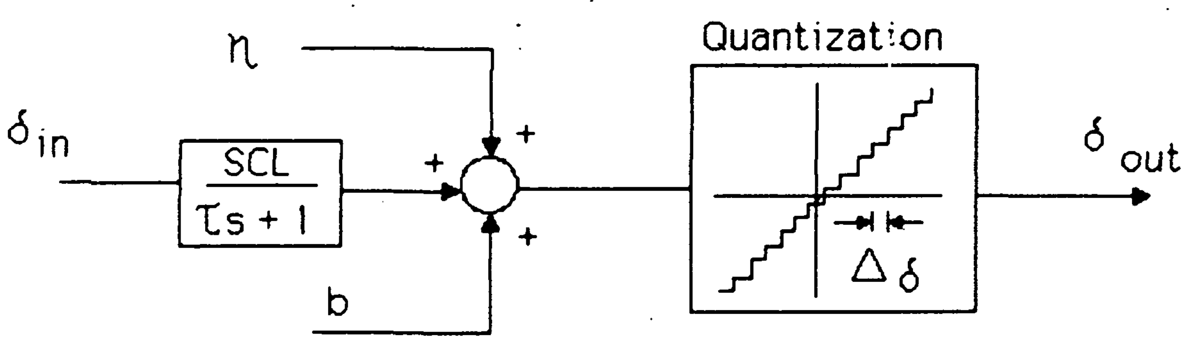
RATE GYRO ERROR SOURCE MODEL



TORQUE MOTOR ERROR SOURCE MODEL

(20)

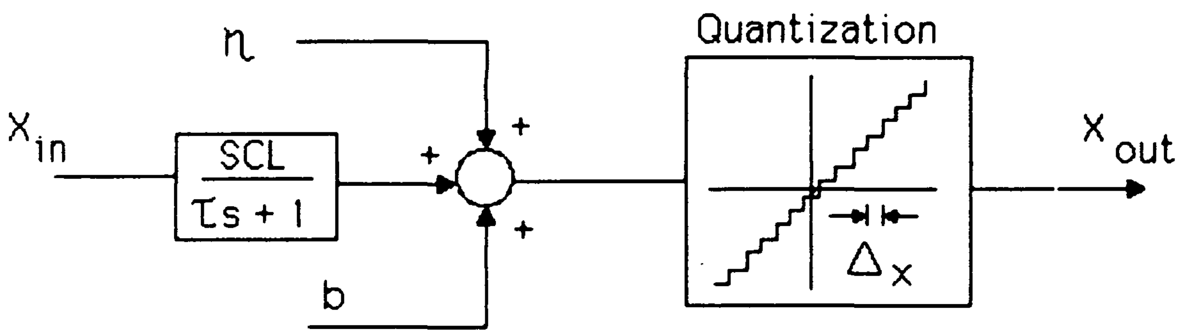
LOS MODEL



PARAMETER	NOMINAL VALUE	DESCRIPTION	SOURCE
σ^2	$(.025 \text{ arc sec})^2$	Intensity of zero mean Gaussian White Noise η	PINHOLE/CORONOGRAPH POINTING CONTROL SYSTEM INTEGRATION & NOISE REDUCTION ANALYSIS, "Michael Greene" Nasa contract # NAS8-34529 Sept 1981, page 27.
SCL	1.0	Scale factor	Nominal deviation from 1. is arbitrarily chosen to be 1%.
b	.25 arc sec	Bias	IPS Simulation Parameters for Performance Analyses. Dornier, IPS-DS-TN-0184. Page 42.
Δ_δ	.14 arc sec	Quantization error	IPS Simulation Parameters for Performance Analyses. Dornier, IPS-DS-TN-0184. Page 42.
τ	.016 sec	Time constant	Selected from in-house Library report.

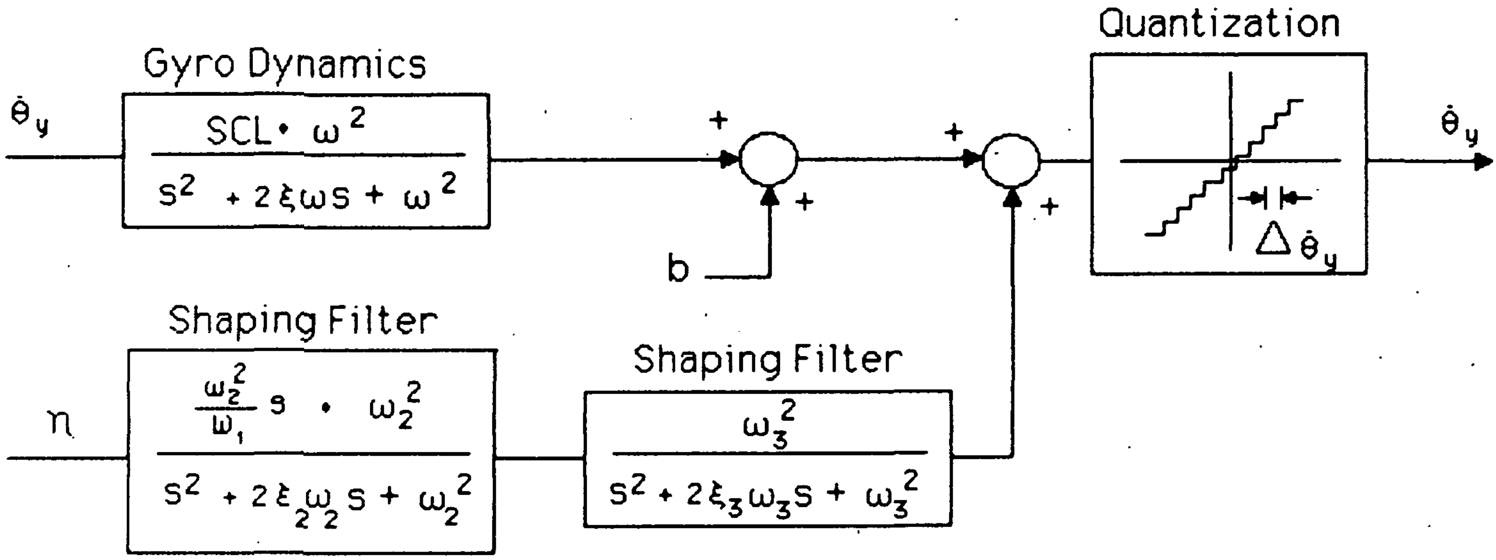
(21)

LASER INTERFEROMETER MODEL



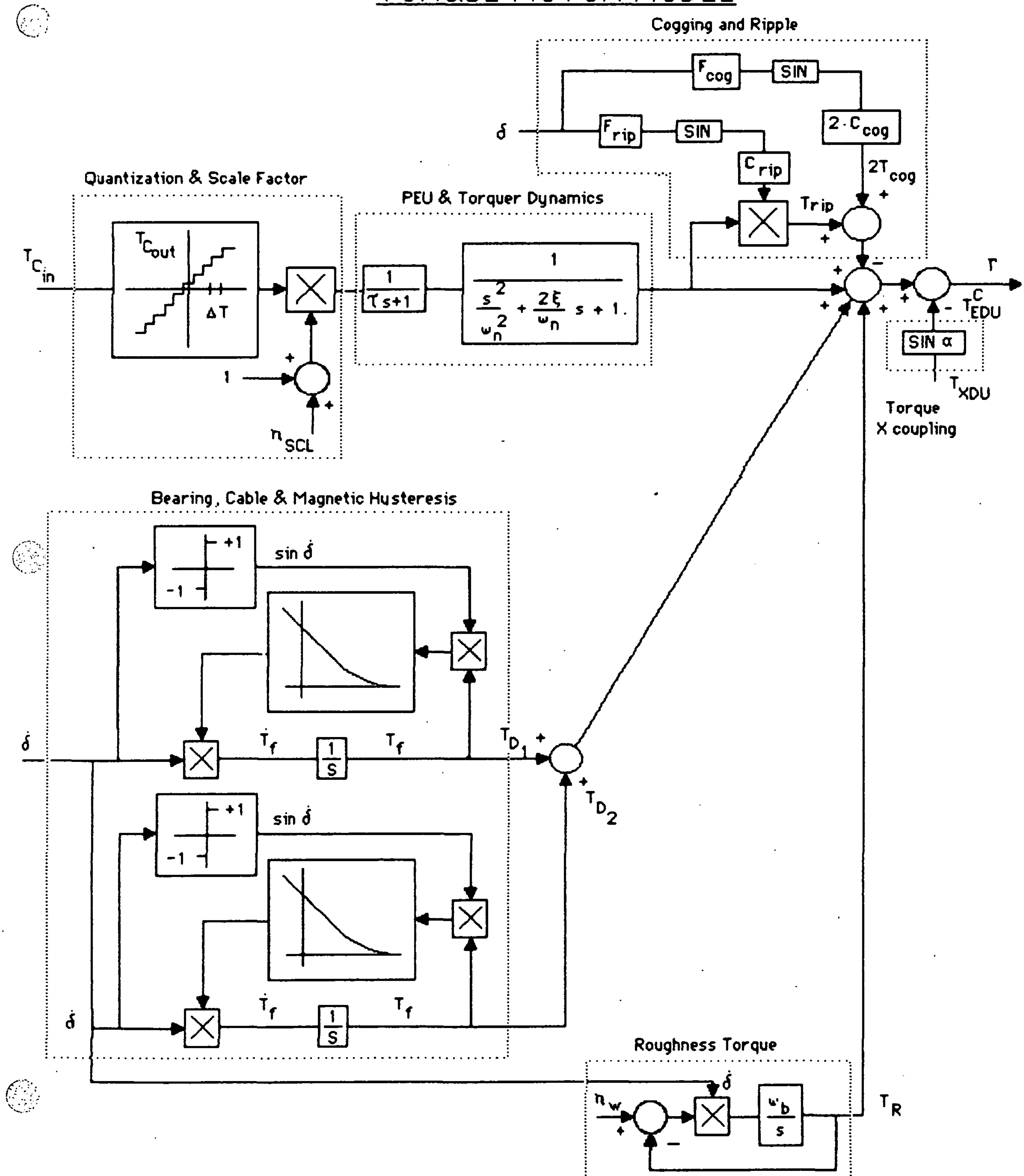
PARAMETER	NOMINAL VALUE	DESCRIPTION	SOURCE
σ^2	$(5 \times 10^{-6}\, m)^2$	Intensity of zero mean Gaussian White Noise η	PINHOLE/CORONOGRAPH POINTING CONTROL SYSTEM INTEGRATION & NOISE REDUCTION ANALYSIS, "Michael Greene" Nasa contract # NAS8-34529 Sept 1981, page 27.
SCL	1.0	Scale factor	Nominal deviation from 1. is arbitrarily chosen to be 1%.
b	$10.67 \times 10^{-6}\, m$	Bias	"Progress in absolute distance interferometry", C. W. Gillard, N.E. Buholz, Optical Engineering, May/June 1983, Page 351
Δx	$.01 \times 10^{-6}\, m$	Quantization error	"Progress in absolute distance interferometry", C. W. Gillard, N.E. Buholz, Optical Engineering, May/June 1983, Page 351
τ	.016 sec	Time constant	PINHOLE/CORONOGRAPH POINTING CONTROL SYSTEM INTEGRATION & NOISE REDUCTION ANALYSIS, "Michael Greene" Nasa contract # NAS8-34529 Sept 1981, page 28.

RATE GYRO MODEL



PARAMETER	NOMINAL VALUE	DESCRIPTION	SOURCE
σ^2	$(.0148 \frac{\text{arc sec}}{\text{sec}})^2$	Zero mean white noise Intensity	IPS Simulation Parameters for Performance Analyses. Dornier, IPS-DS-TN-0184, page 38.
SCL	1.0 1 σ deviation from nominal value is .000478	Scale Factor	IPS Simulation Model, Dornier IPS-DS-TN-183, page 61.
$\Delta\dot{\theta}_y$	.00167 $\frac{\text{arc sec}}{\text{sec}}$	Quantization error	IPS Simulation Parameters for Performance Analyses. Dornier, IPS-DS-TN-0184, page 39.
b	1 $\frac{\text{arc sec}}{\text{sec}}$	1 σ value of fixed drift	IBID, page 39
		Gyro Dynamics:	IBID, page 39
ω	125.66 r/s	Natural Frequency	
ξ	.7	Damping Ratio	
		Shaping Filters:	IBID, page 38
ω_1	$2\pi \cdot 2.4$ r/s	Natural Frequency	
ω_2	$2\pi \cdot 23$ r/s	Natural Frequency	
ω_3	$2\pi \cdot 46$ r/s	Natural Frequency	
ξ_1	.707	Damping Ratio	
ξ_2	.35	Damping Ratio	

(23)
TORQUE MOTOR MODEL



TORQUE MOTOR MODEL PARAMETER DEFINITIONS

PARAMETER	NOMINAL VALUE	DESCRIPTION	SOURCE
ΔT	.0618 nm	Torque quantization	IPS Simulation Parameters for Performance Analyses. Dornier, IPS-DS-TN-0184, page 33.
σ_{SCL}	.043 nm	1 σ value of scale factor error η_{SCL}	IBID, page 33
τ	.00442 sec	Drive unit time constant	IBID, page 34
ω_n	48 Hz	PEU break frequency	IBID, page 34
ξ	.647	PEU damping ratio	IBID, page 34
F_{cog}	96 cycles/REV	Cogging frequency	IBID, page 33
C_{cog}	.1 nm	Cogging gain	IBID, page 33
F_{rip}	48 cycles/REV	Ripple frequency	IBID, page 34
C_{rip}	.0275	Ripple gain	IBID, page 34
T_{0_1}	.6 nm	Saturation torque Dahl Model #1	IBID, page 36
T_{0_2}	2 nm	Saturation torque Dahl Model #2	IBID, page 36
γ_1	1736	Slope Parameter Dahl Model #1	IBID, page 36
γ_2	9.375	Slope Parameter Dahl Model #2	IBID, page 36
σ_w^2	$(.025\text{nm})^2$	Intensity of roughness torque ZMWG noise η_w	IBID, page 37
ω_b	75 cycles/deg	Roughness torque bandwidth	IBID, page 37
α	.5 deg	Gimbal misalignment	IBID, page 37

4.3 PARAMETER SENSITIVITIES

- TASK: Determine the effect each error source has on pointing performance.

- Objective: To be able to specify a sensor/actuator error budget that will satisfy a 1 arc second steady state pointing error requirement.

- PROCEDURE:

  STEP 1) Introduce the particular error source into TREETOPS that defines the particular parameter under study.

  STEP 2) Place an initial condition on the LOS error(10 arc sec). Once the system has settled out(25 sec), begin recording the rms value of the LOS error.

  STEP 3) Record the stabilized rms value of the LOS error.

  STEP 4) Repeat steps 1-3 with the parameter set at X10 its nominal value.

  STEP 5) Remove the previous error source modeled in TREETOPS and then insert the next error source defining the new parameter to be studied. Repeat steps 1 through 5 until all parameter sensitivities have been obtained.

COMMENTS/RESULTS

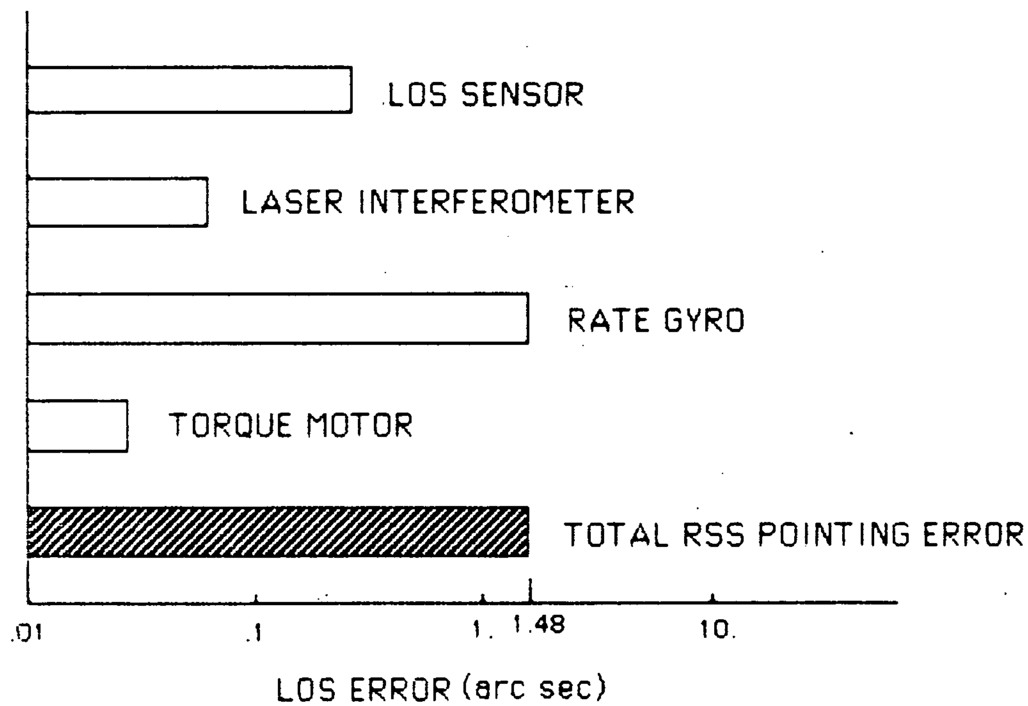
- Page 27 summarizes the parameter sensitivities studied

- Page 28 shows the contribution of each sensor/actuator to the total LOS RSS pointing error.
- Pages 29 and 30 show the contribution of each individual error source to the RSS instrument error.
- The LOS error showed a linear relationship to each individual parameter variation.
- The gyro fixed drift is the largest contributer to the RSS LOS pointing error.

PARAMETER SENSITIVITIES STUDIED

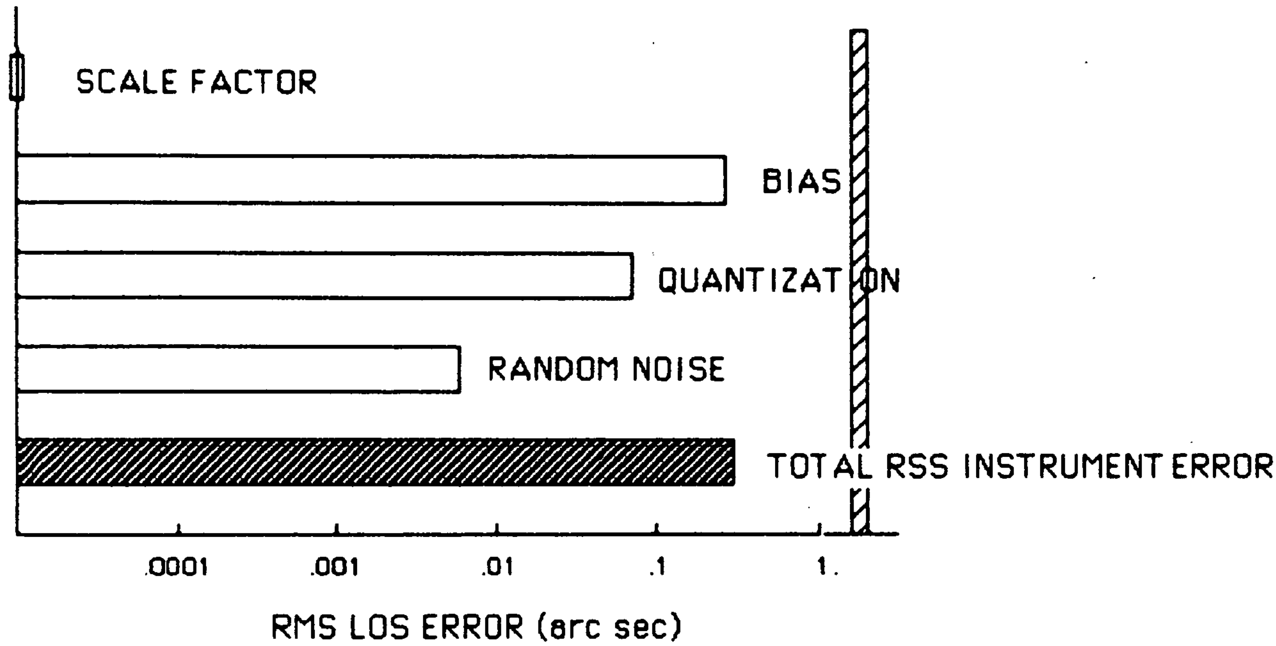
SYSTEMATIC ERRORS:	NON-SYSTEMATIC ERRORS:
LOS MODEL	
• SCALE FACTOR	• RANDOM ZMWG NOISE
• BIAS	• QUANTIZATION
LASER INTERFEROMETER	
• SCALE FACTOR	• RANDOM ZMWG NOISE
• BIAS	• QUANTIZATION
RATE GYRO	
• SCALE FACTOR	• RANDOM COLORED NOISE
• BIAS	• QUANTIZATION
TORQUE MOTOR	
• SCALE FACTOR	• RIPPLE TORQUE
• CROSS COUPLING DUE TO GIMBAL MISALIGNMENTS	• COGGING TORQUE
	• ROUGHNESS TORQUE
	• QUANTIZATION
	• BEARING, CABLE & HYSTERESIS TORQUE-# 1
	• SATURATION TORQUE
	• SLOPE PARAMETER
	• BEARING, CABLE & HYSTERESIS TORQUE-# 2
	• SATURATION TORQUE
	• SLOPE PARAMETER

STEADY STATE POINTING PERFORMANCE STUDY

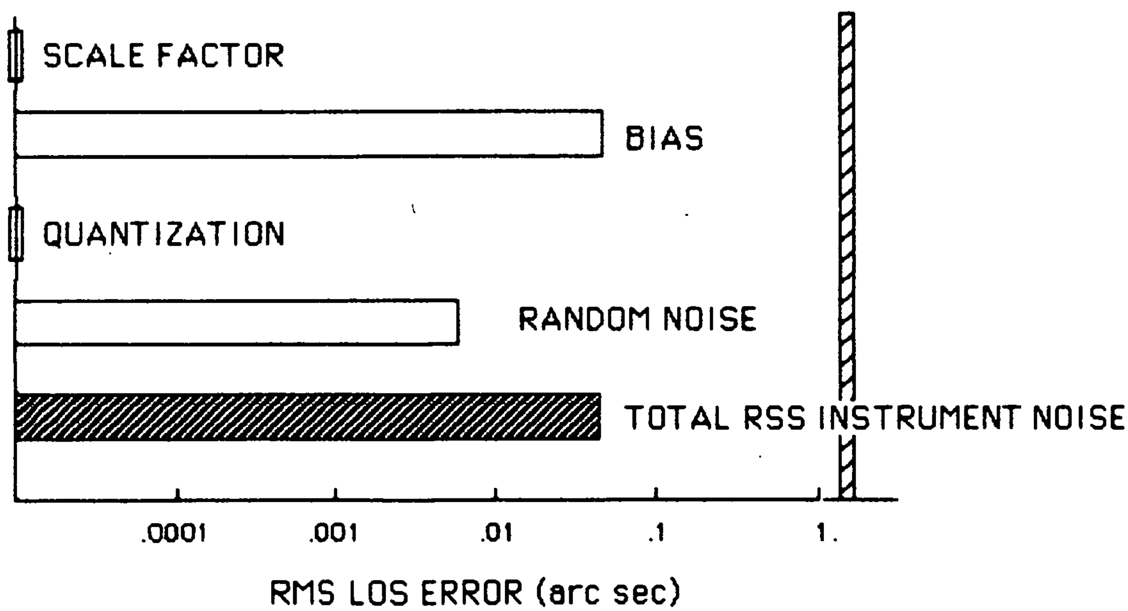


CONTRIBUTION OF EACH SENSOR/ACTUATOR TOWARD TOTAL L.O.S. POINTING ERROR (1.48 arc sec)

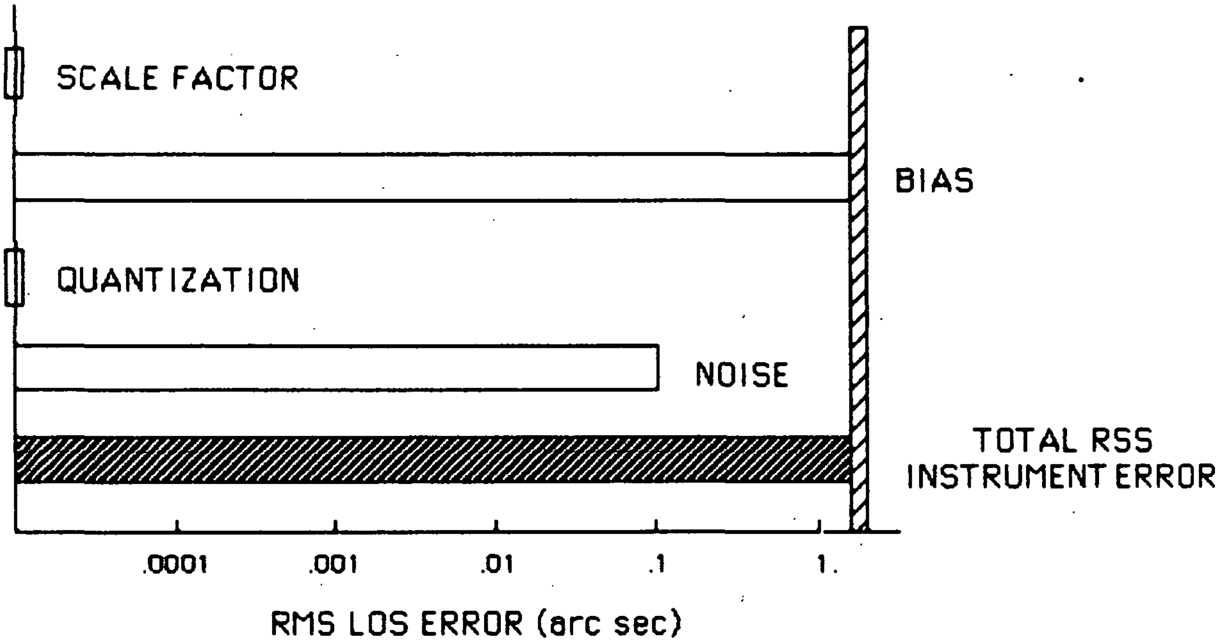
CONTRIBUTION OF VARIOUS L.O.S. SENSOR ERROR TERMS TO TOTAL POINTING ERROR(1 σ = 1.48 arc sec)



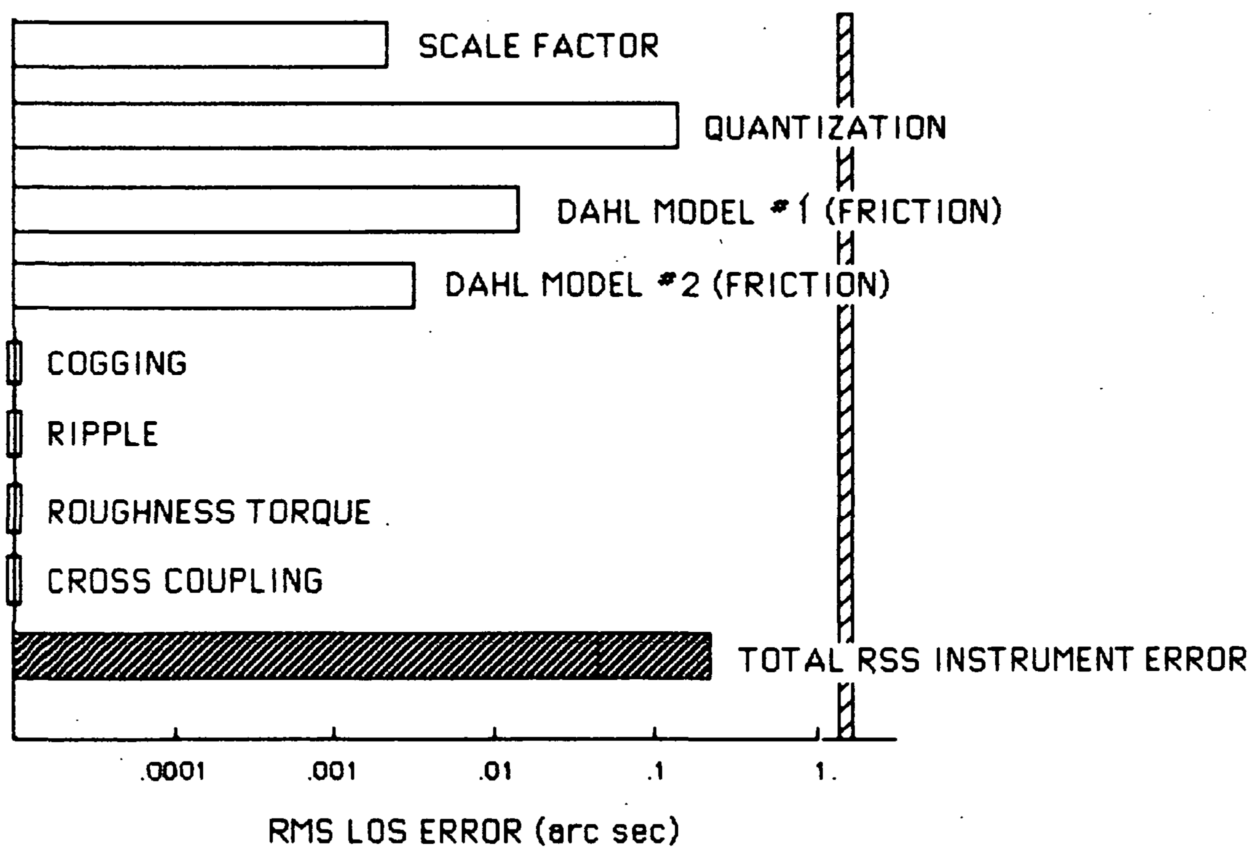
CONTRIBUTION OF VARIOUS LASER INTERFEROMETER ERROR TERMS TO TOTAL POINTING ERROR (1 σ = 1.48 arc sec)



CONTRIBUTION OF VARIOUS RATE GYRO ERROR TERMS TO TOTAL POINTING ERROR(1 σ = 1.48 arc sec)



CONTRIBUTION OF VARIOUS TORQUE MOTOR ERROR TERMS TO TOTAL POINTING ERROR (1 σ = 1.48 arc sec)



4.4 ERROR BUDGET ALLOCATTION

- TASK: Specify nominal parameter values for all error sources to insure that the RSS LOS pointing error is less then 1 arc sec.

- OBJECTIVE: To make reasonable judgements on instrument fidelity required to meet P/OF mission goals.

- PROCEDURE: Given the RMS LOS pointing errors that correspond to each error source nominal parameter value, determine which parameters need to be reduced in order to bring the RSS LOS error under 1 arc sec. Refer to page 32.

ERROR BUDGET ALLOCATION

ERROR SOURCE	NOMINAL VALUE	RMS ERROR(arc sec)	
L.O.S. Sensor			
Scale Factor error	.01 (1σ)	0	
Bias	.25 arc sec (1σ)	.25	
Quantization	.14 arc sec	.09	
Random Noise	.025 arc sec (1σ)	.007	
Laser Interferometer			
Scale Factor	.01 (σ)	0	
Bias	10.67×10^{-6}m(1σ)	.054	
Quantization	$.01\times10^{-6}$ m	0	
Random noise	$5.\times10^{-6}$ m (1σ)	.007	
Rate Gyro			
Scale Factor	.000478 (1σ)	0	
Bias(nominal)	1 (arc sec)/sec	1.46	
(reduced nominal)	.66 (arc sec)/sec(1σ)		.96
Quantization	.00167 arc sec/sec	0	
Random Noise	$.0148\ \left(\frac{\text{arc sec}}{\text{sec}}\right)^2/\text{Hz}\ (1\sigma)$	.1	
Torque Motor			
Scale Factor	.043 (1σ)	.00225	
Quantization	.0618 nm	.013	
Dahl Friction model 1	(T_0 =.6nm, ψ=1736)	.019	
Dahl Friction model 2	(T_0 =2nm, ψ=9.375)	.003	
Cogging	.1 nm	0	
Ripple	.0275 nm	0	
Roughness Torque	.025 nm (1σ)	0	
Cross Coupling	.5 deg	0	
	R.S.S.	1.49 [1]	1.0 [2]

Notes:

1) RSS LOS error for nominal parameter values
2) RSS LOS error with rate gyro fixed drift bias reduced to .66 arc sec/sec

4.5 ALTERNATE MOUNTING BASE COMPARISON

- TASK: Evaluate P/OF pointing performance using the Space Station as an alternate mounting base.

- OBJECTIVE: Enhanced viability of the P/OF concept

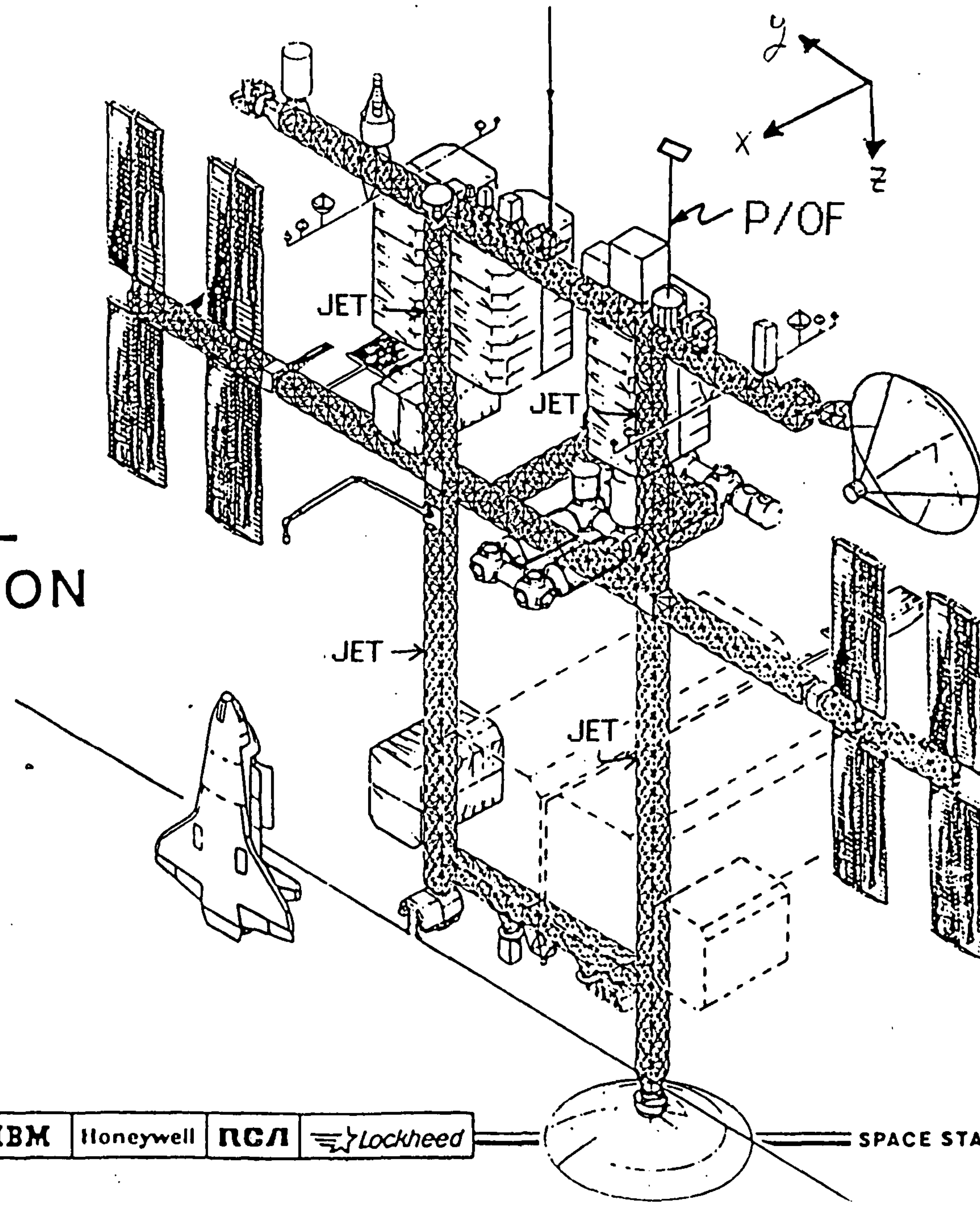
- PROCEDURE :

  Step 1) Merge TREETOPS simulations of the P/OF and the Dual Keel configured Space Station.

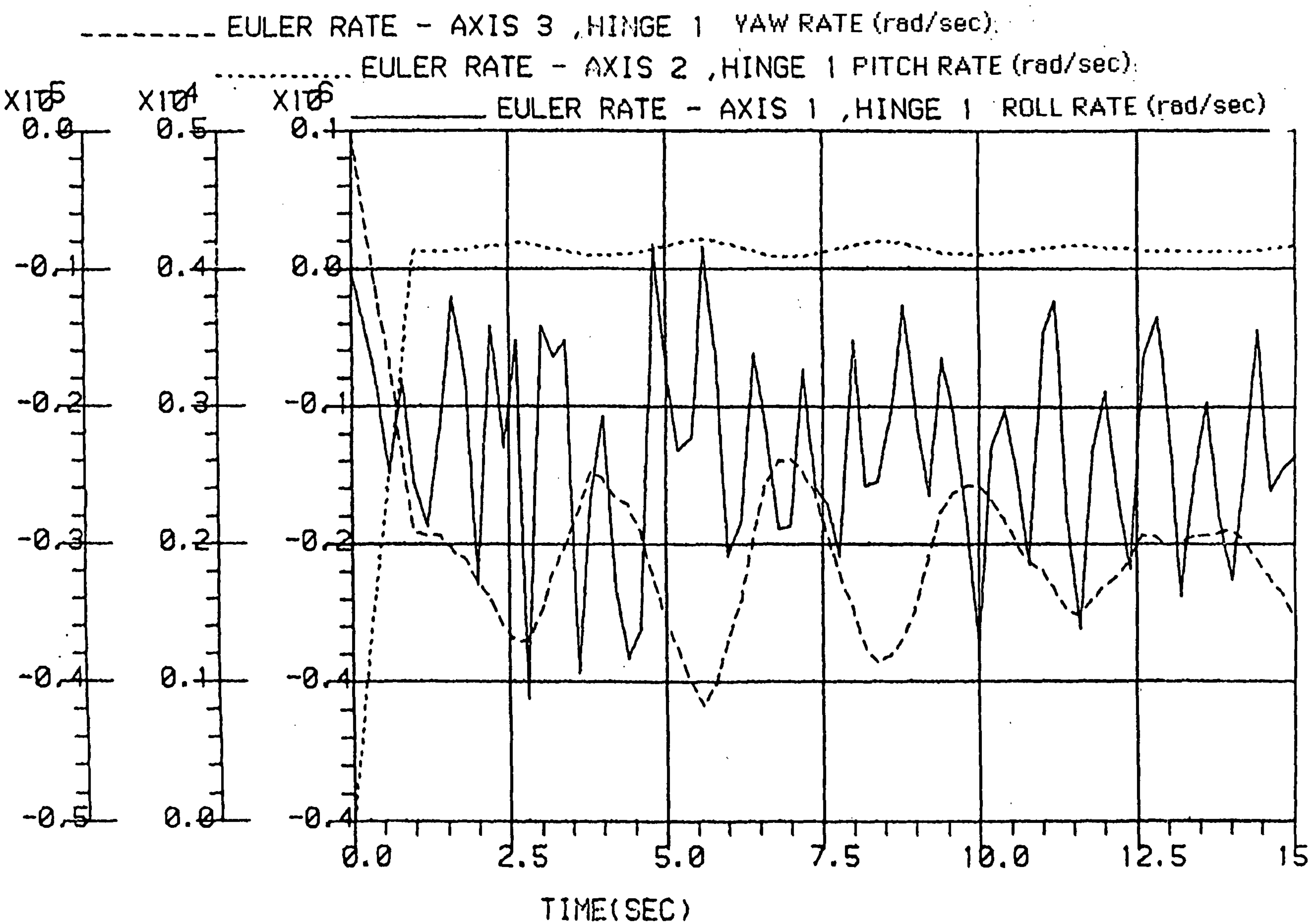
  Step 2) Run TREETOPS simulation to obtain time histories of the P/OF pointing performance for a worse case attitude control system disturbance. The 4 jets as shown on page 34 were fired for 1 sec with each having a magnitude of 25 lbs to generate a pitching moment. Pages 35 through 41 show the resulting Space Station euler rates, euler angles, LOS errors and torque motor outputs and base accelerations.

VHK093

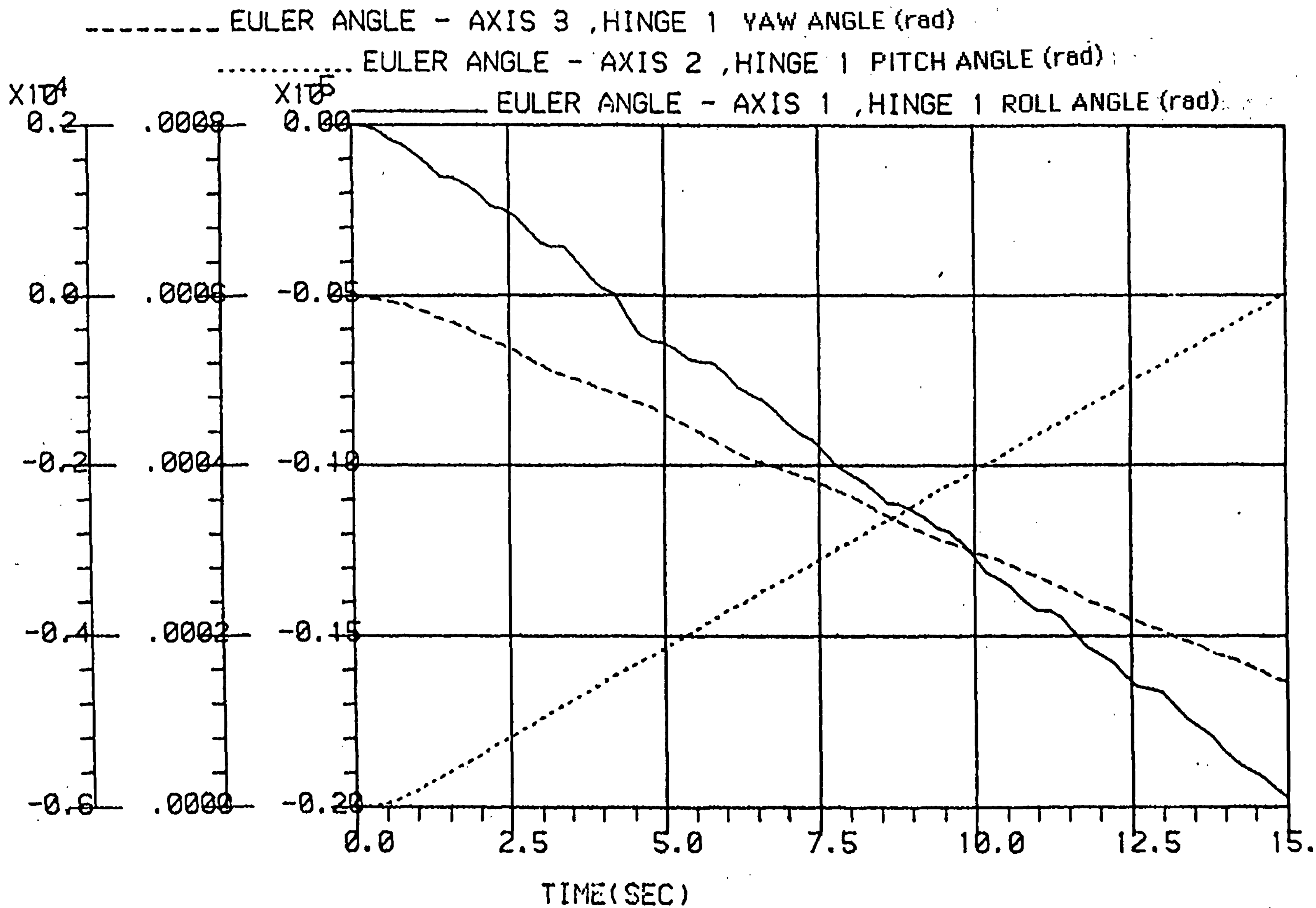
DUAL KEEL SPACE STATION



SPACE STATION ATTITUDE CONTROL – PITCH JET FIRING
4 JETS; 2 +X, 2 -Y, Magnitude - 25# , Duration - 1 sec



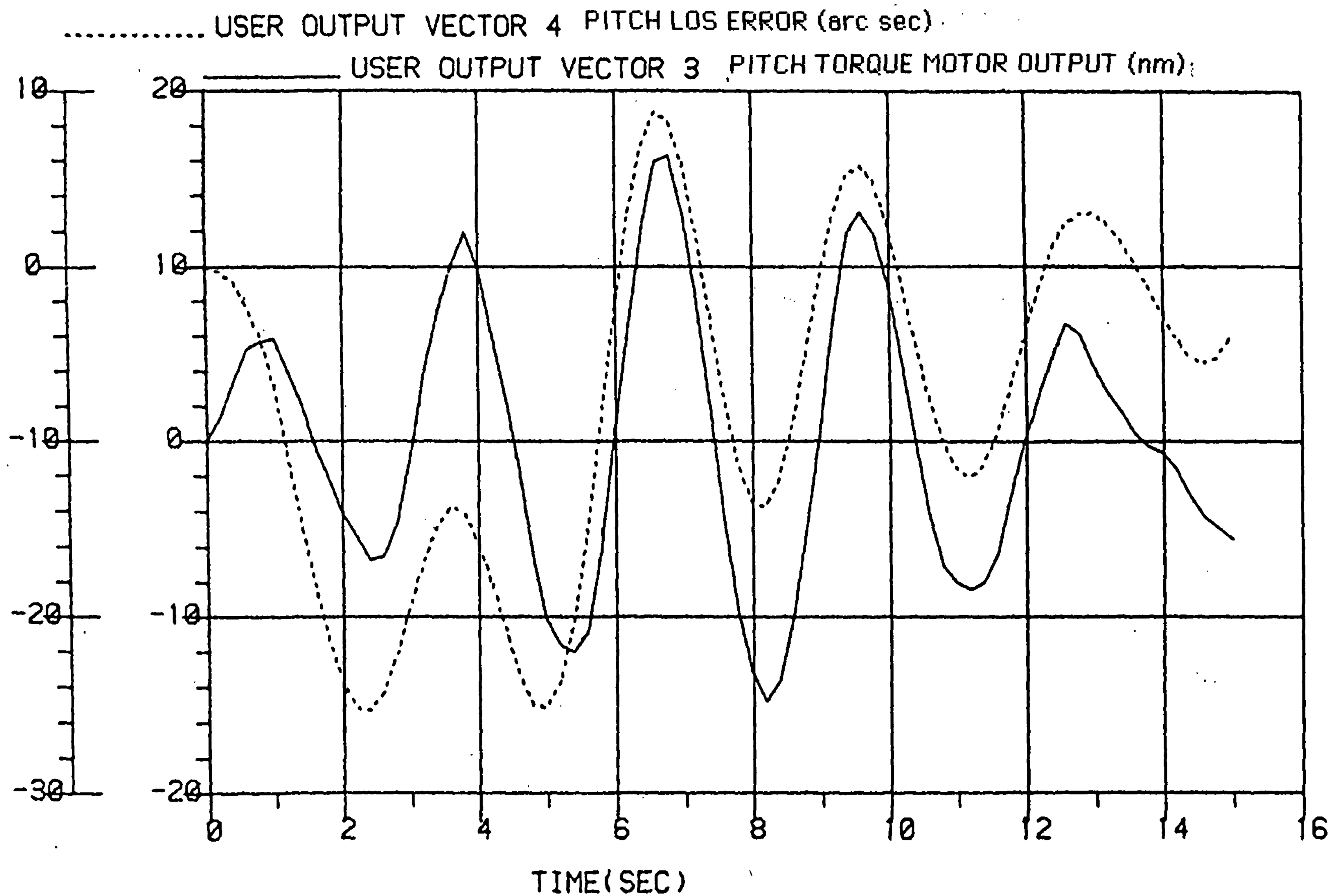
0



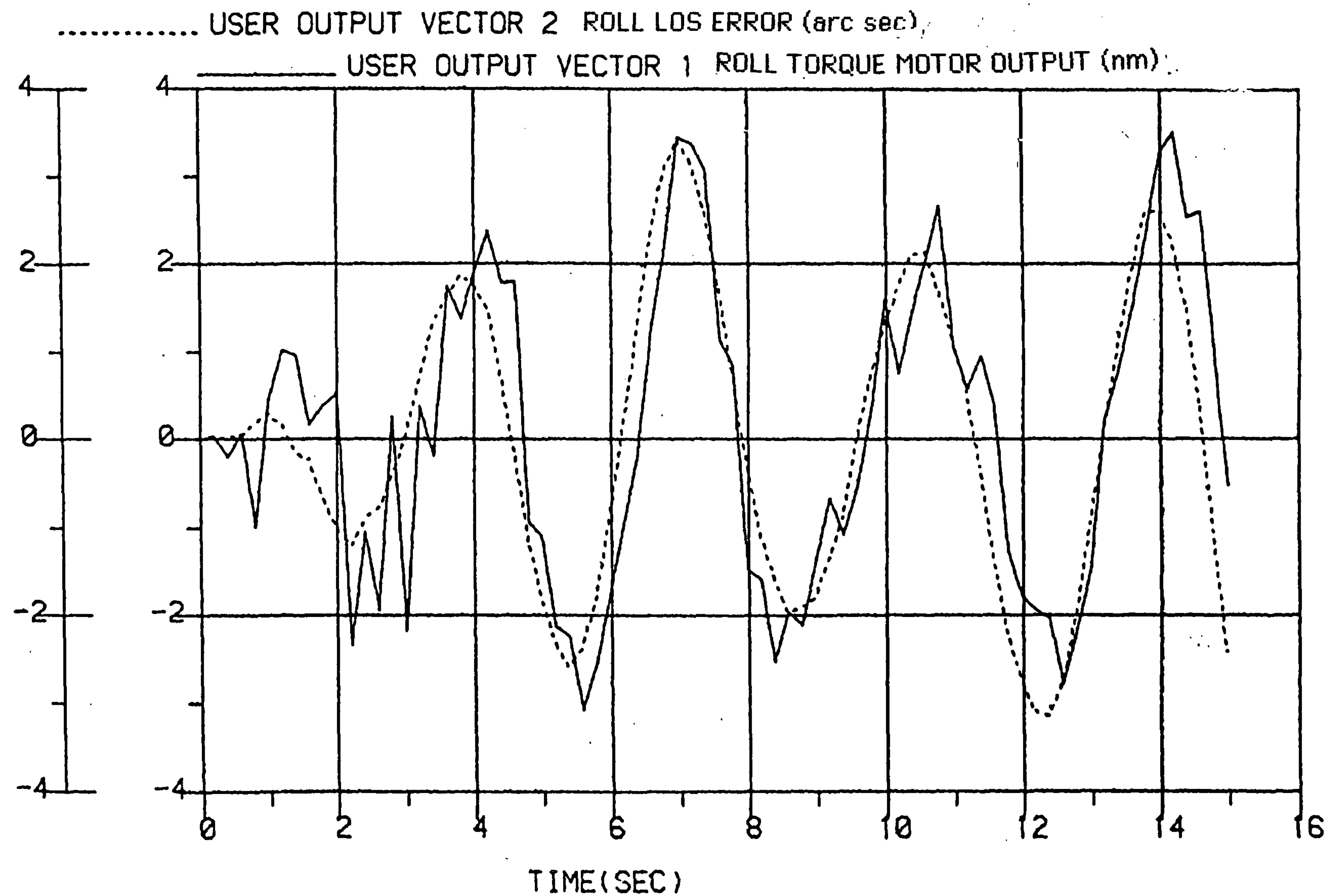
EULER ANGLE - AXIS 3 ,HINGE 1 YAW ANGLE (rad)
EULER ANGLE - AXIS 2 ,HINGE 1 PITCH ANGLE (rad)
EULER ANGLE - AXIS 1 ,HINGE 1 ROLL ANGLE (rad)
X10^-4
X10^-5
0.2
0.0
-0.2
-0.4
-0.6
.0008
.0006
.0004
.0002
.0000
0.00
-0.05
-0.10
-0.15
-0.20
0.0
2.5
5.0
7.5
10.0
12.5
15.
TIME(SEC)

0

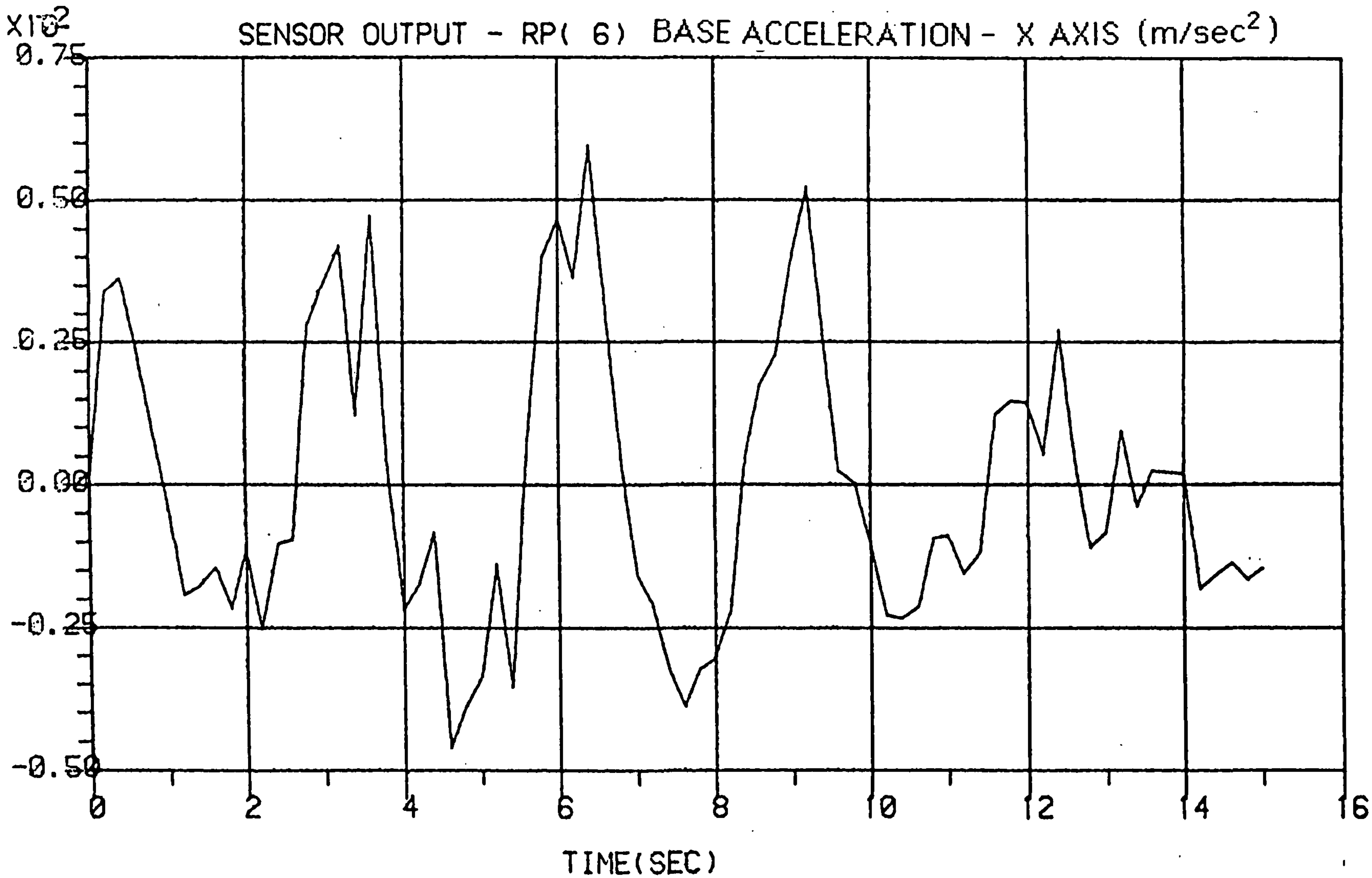
KEYWORD,NAME(1)?



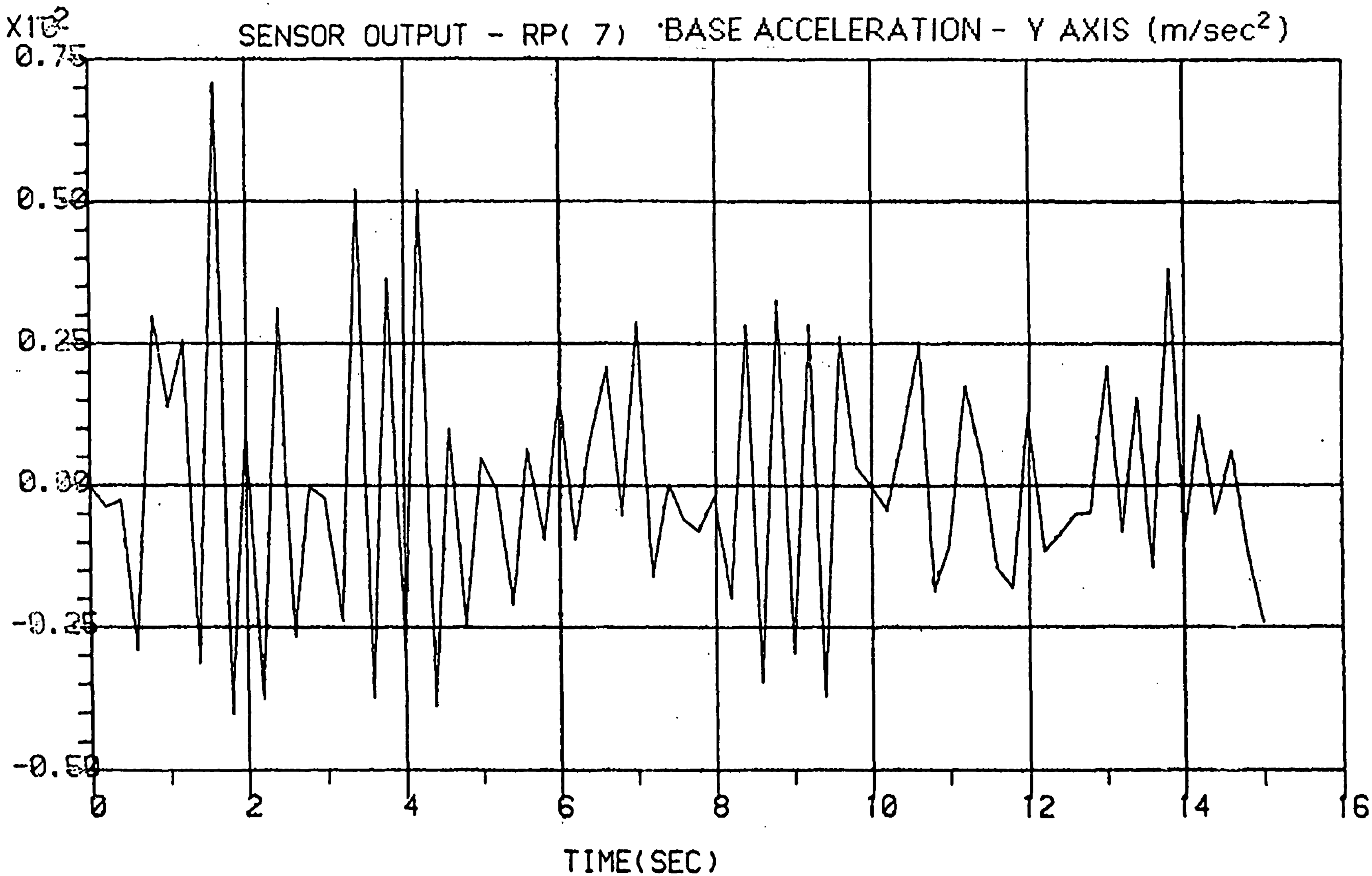
KEYWORD, NAME(1)?



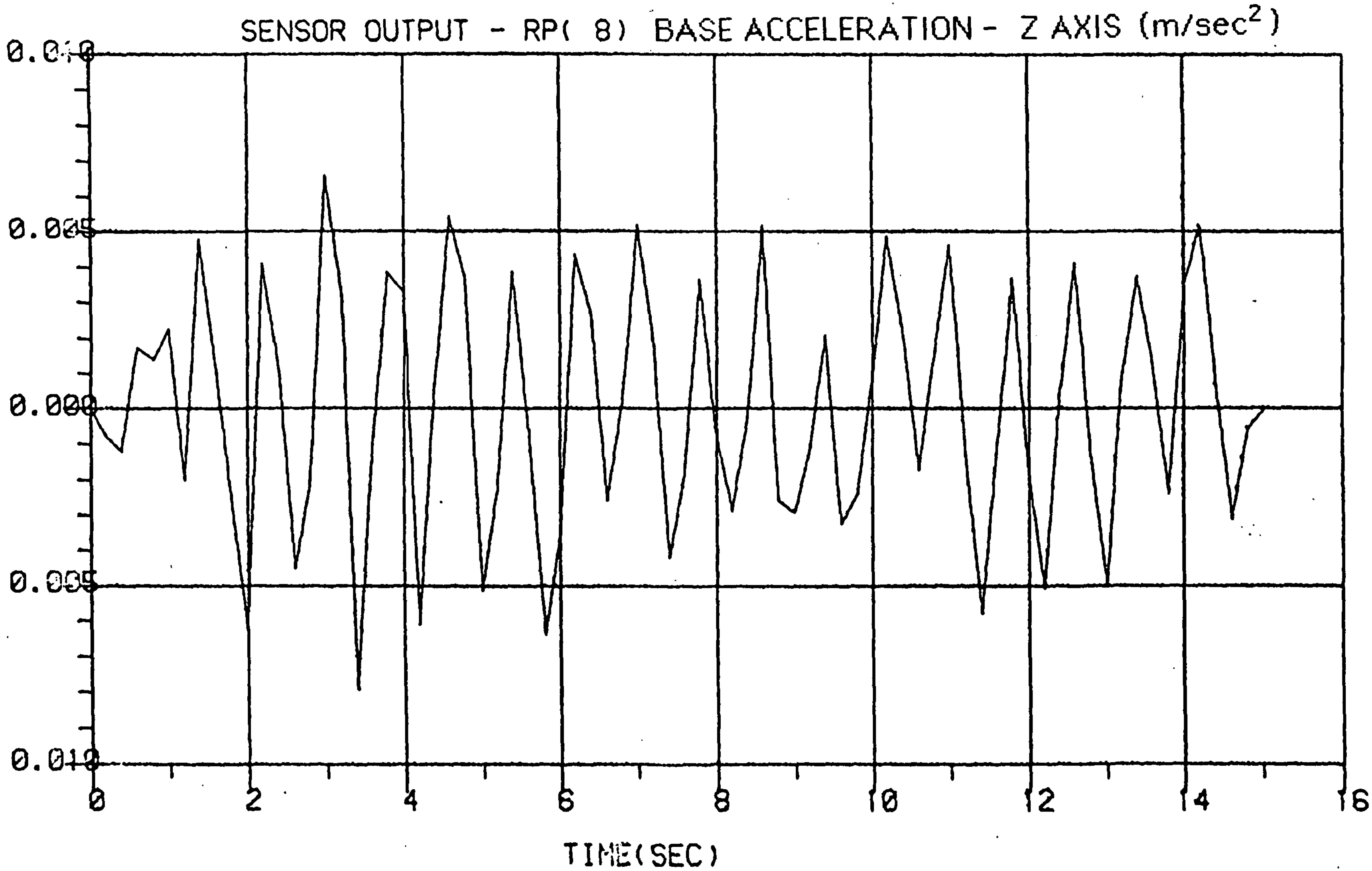
KEYWORD,NAME(1)?



KEYWORD,NAME(1)?



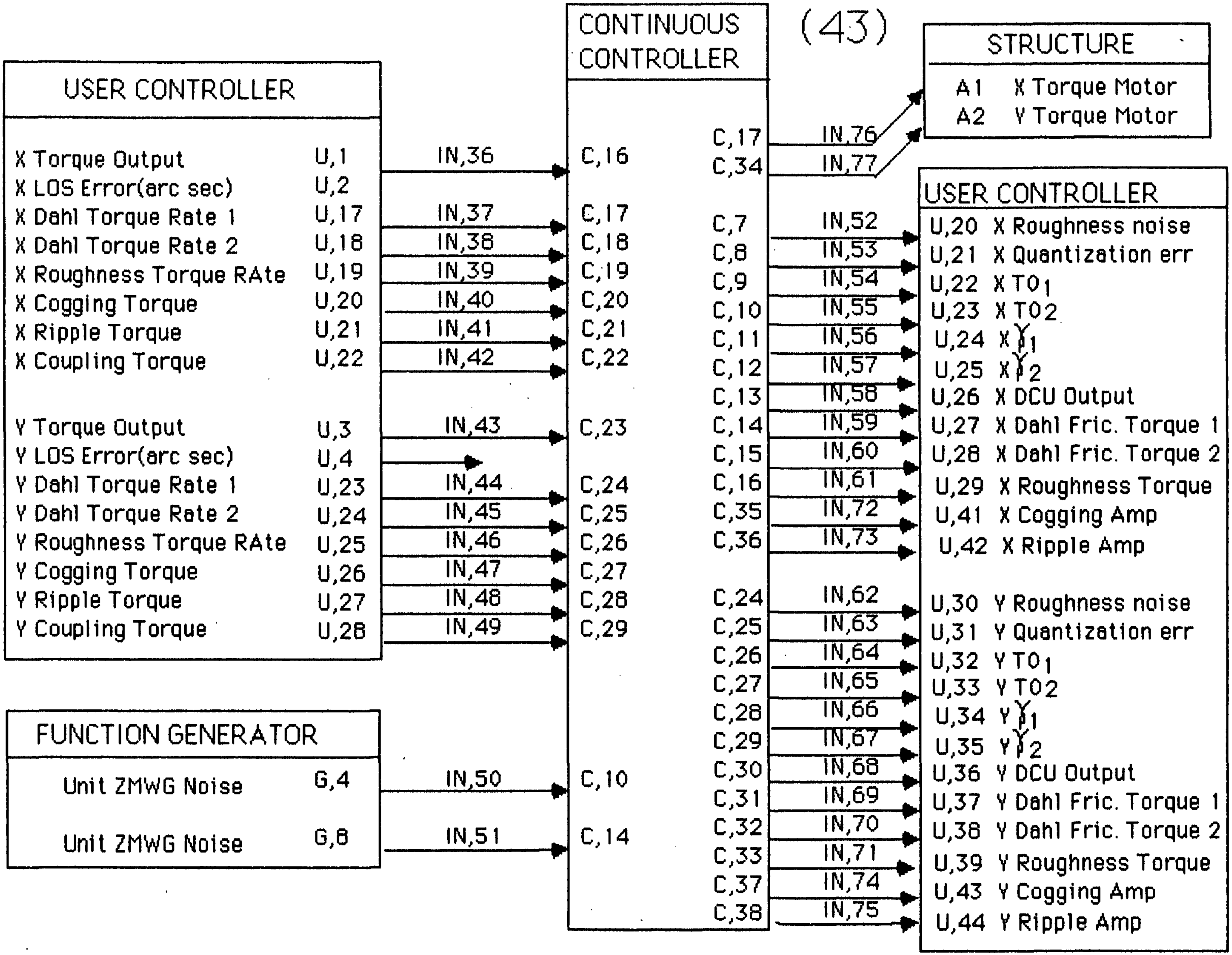
KEYWORD,NAME(1)?



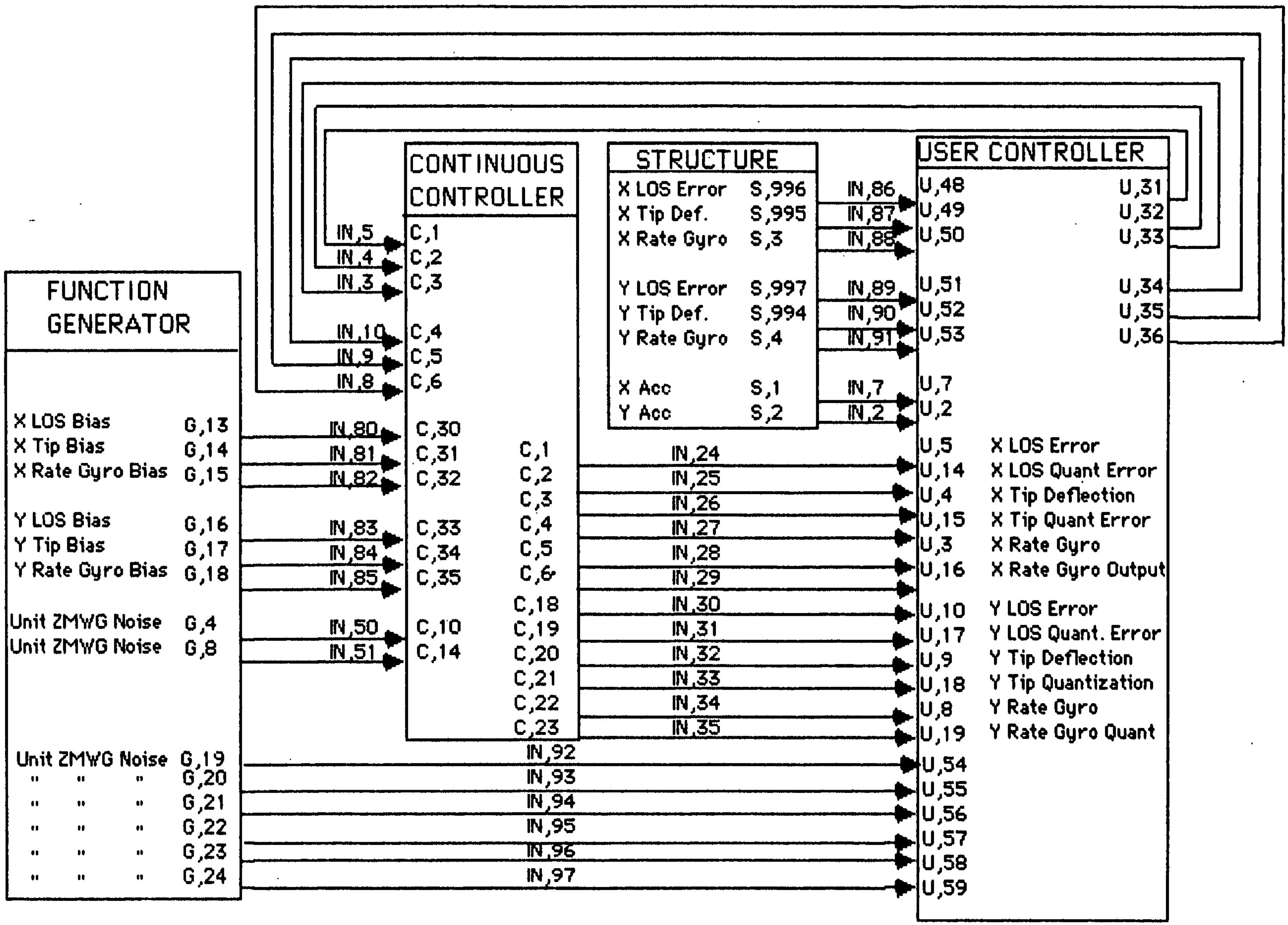
5.0 BASELINE TREETOPS SIMULATION STRUCTURE

The TREETOPS simulation structure for the P/OF is presented in this section. Included are the interconnect diagrams (pages 43 through 46) which show how the various components of the TREETOPS simulation communicate with each other. The continuous controller which was used primarily to model the sensor and torque motor dynamics are presented in block diagram form on pages 47 through 52. The user defined function generator was used to define the random noise and random constants used in the simulation. The user controller was modified to define the discrete implementation of the P/OF full state controller. Other additions to the user controller were made to model sensor/actuator nonlinearities.

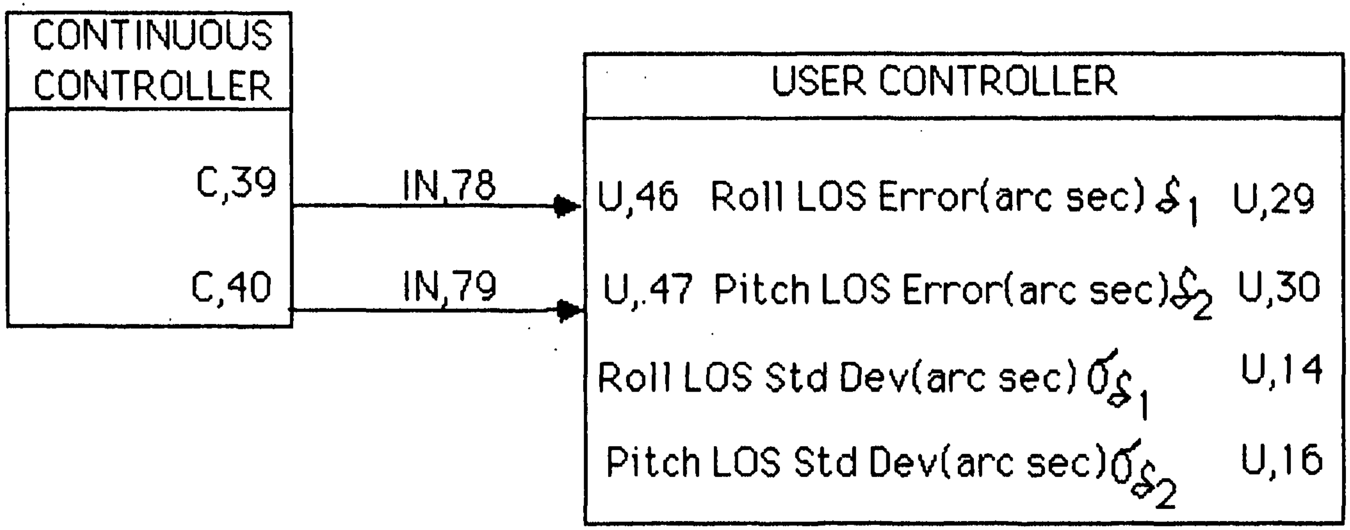
TORQUE MOTOR INTERCONNECT DIAGRAM



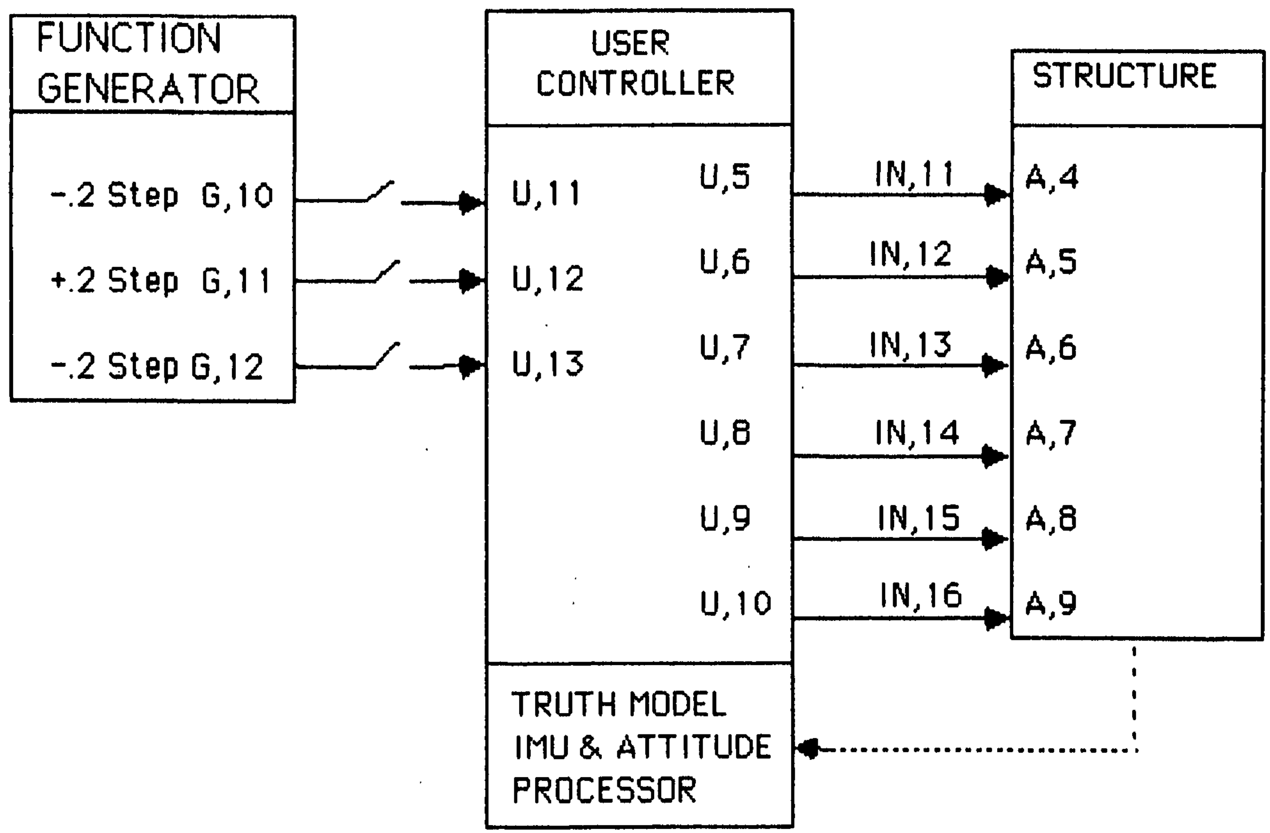
SENSOR INTERCONNECT DIAGRAM



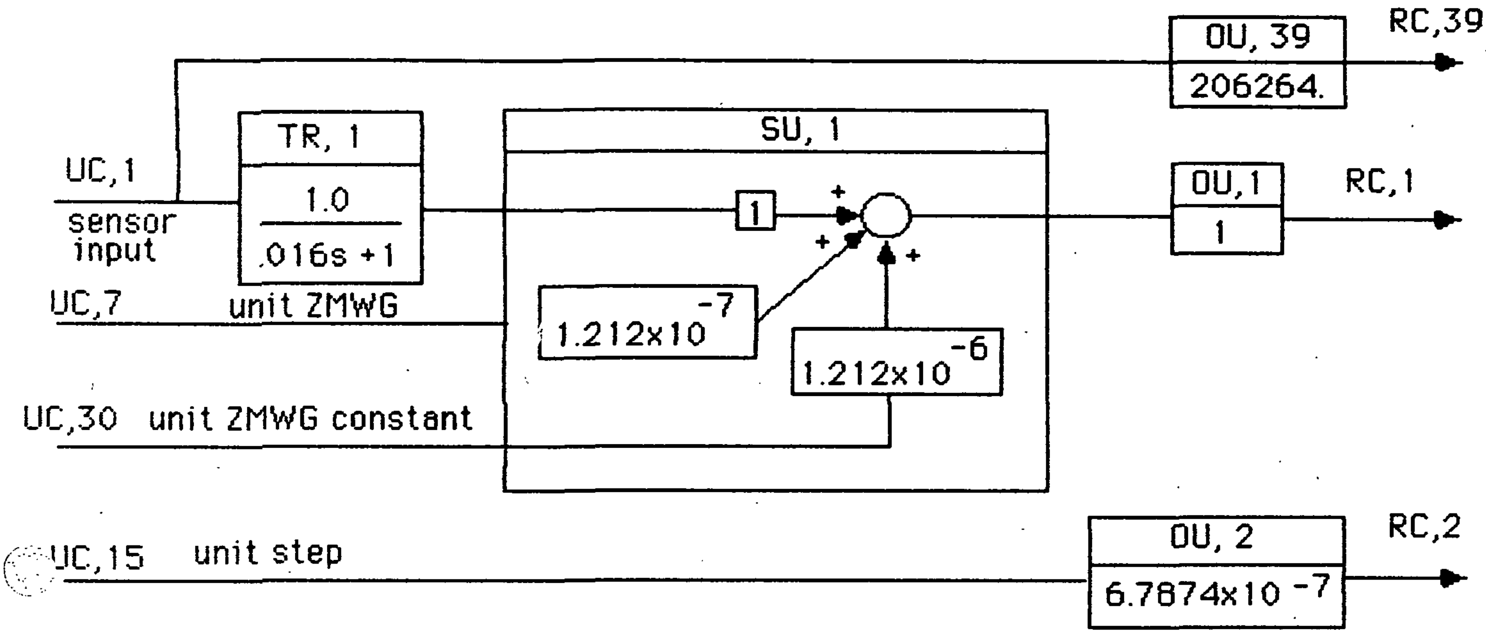
POST PROCESSOR INTERCONNECT DIAGRAM



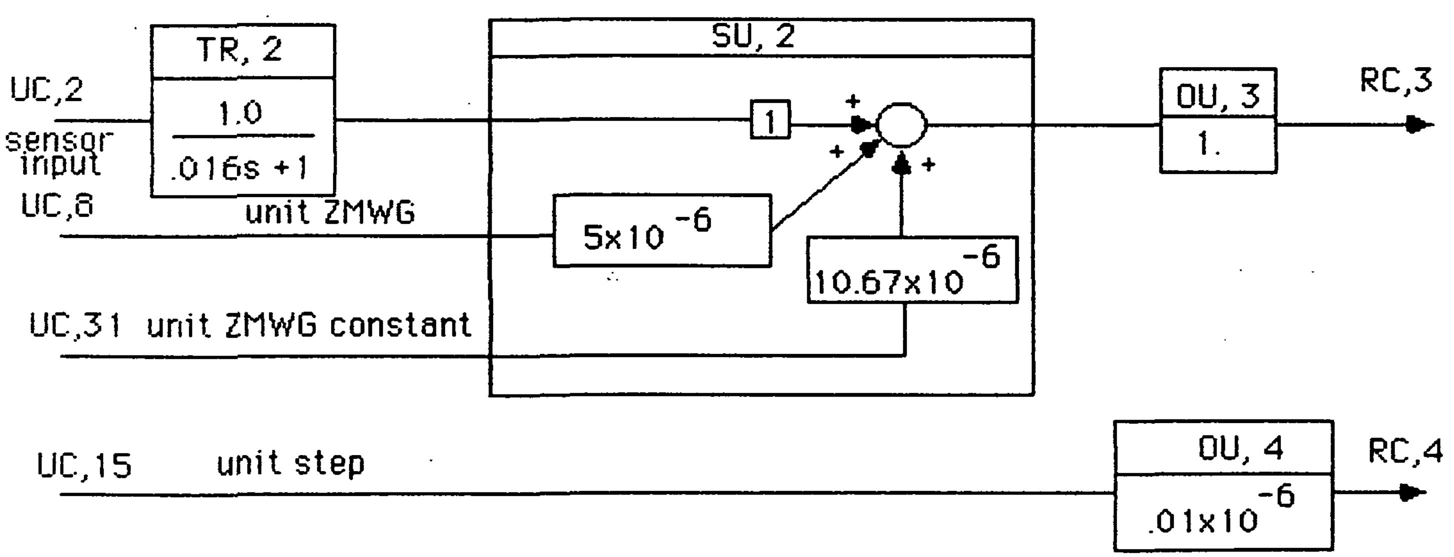
VERNIER RCS ORBITER ATTITUDE CONTROL INTERCONNECT



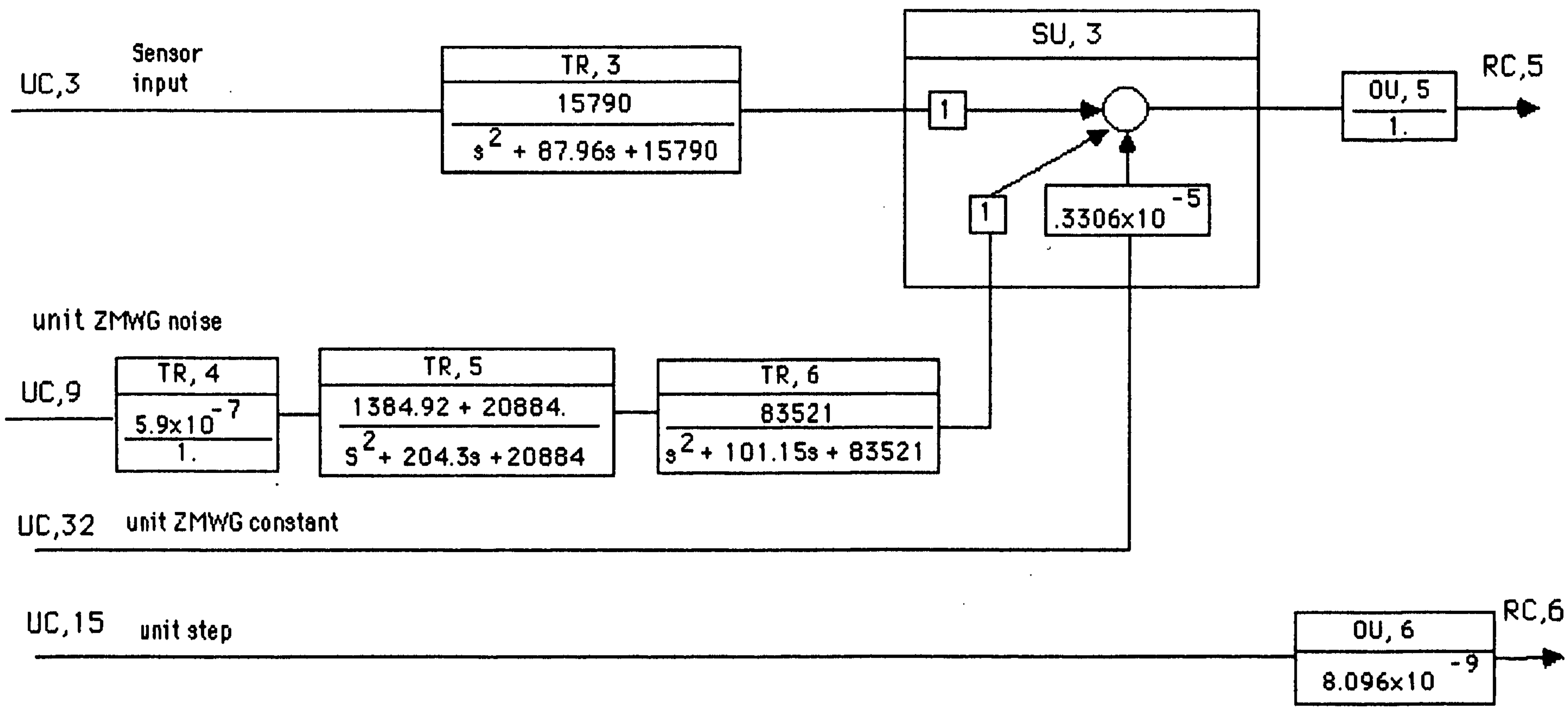
TREETOPS CONTINUOUS CONTROLLER - LOS SENSOR (ROLL)



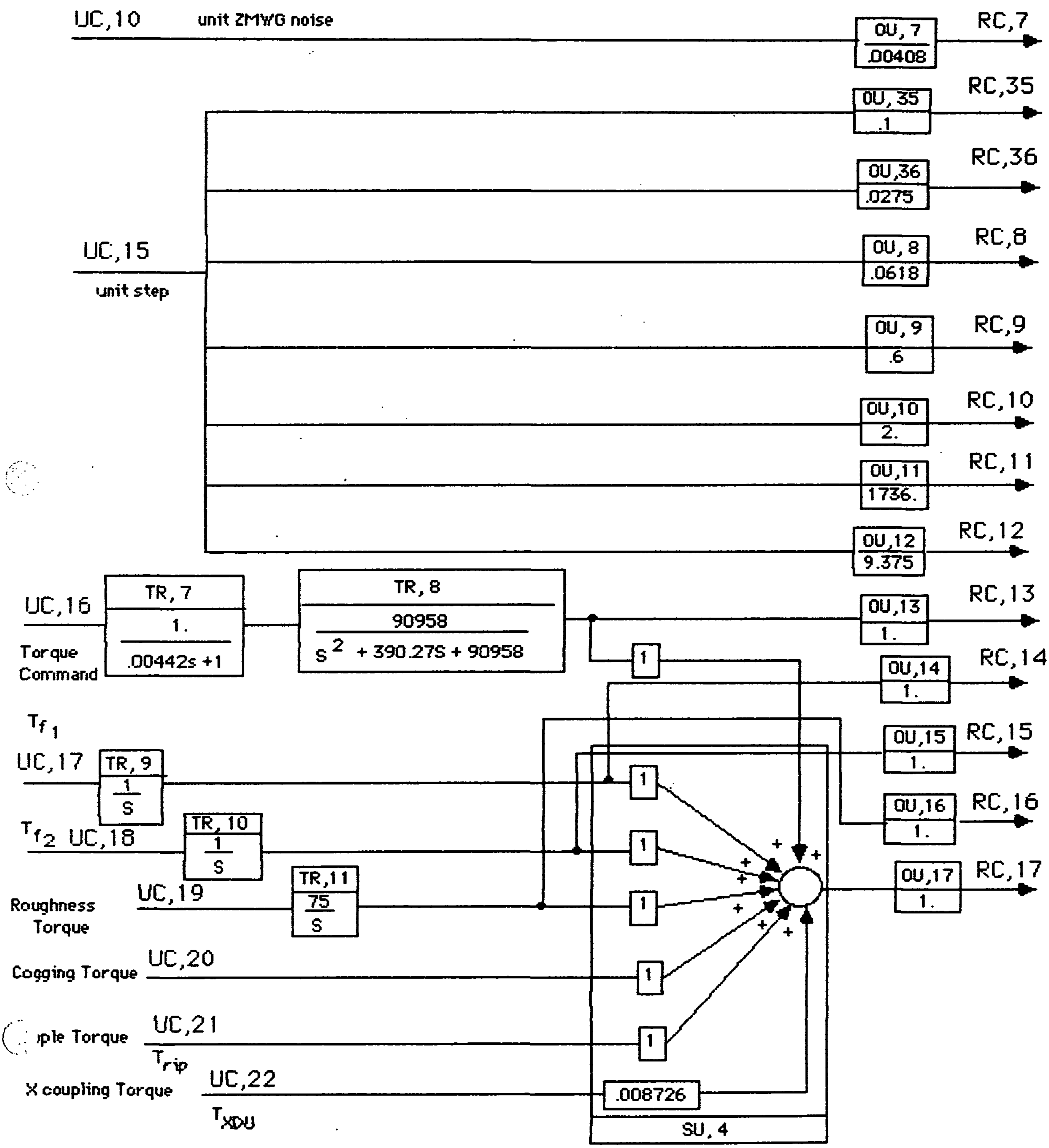
TREETOPS CONTINUOUS CONTROLLER - LASER INTERFERFEROMETER(ROLL)



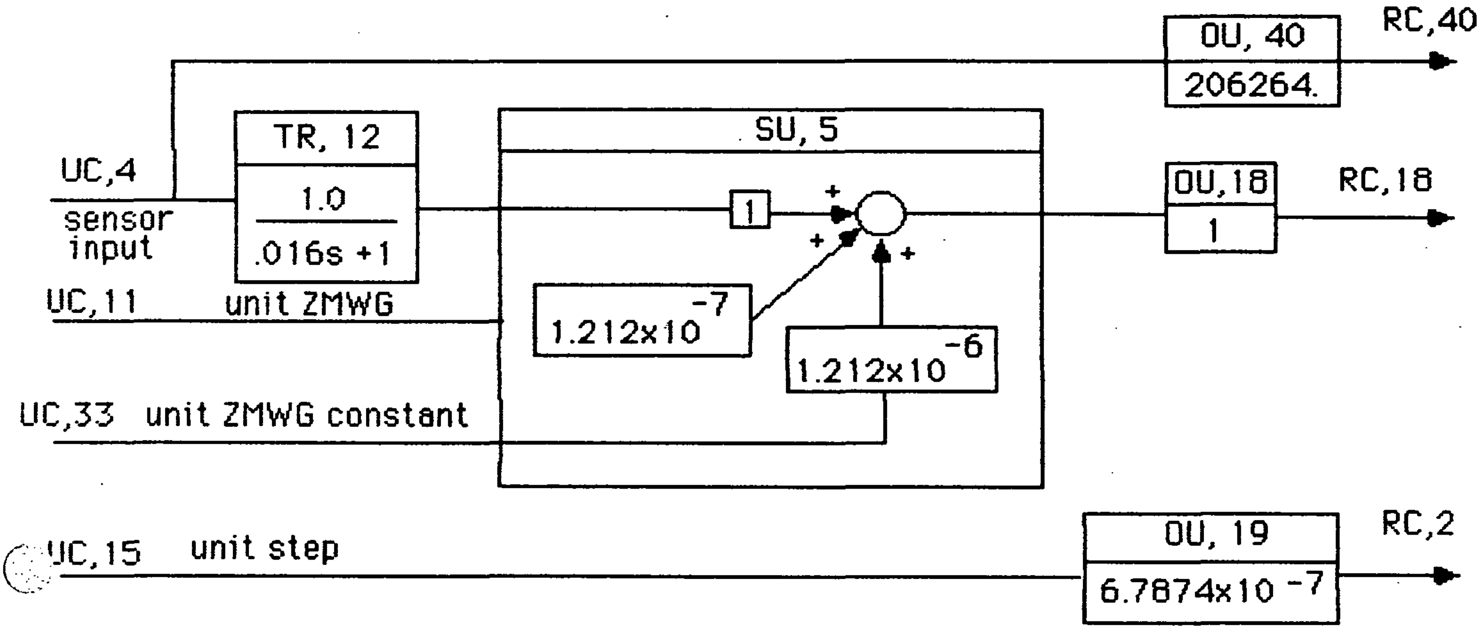
TREETOPS CONTINUOUS CONTROLLER - RATE GYRO SENSOR(ROLL)



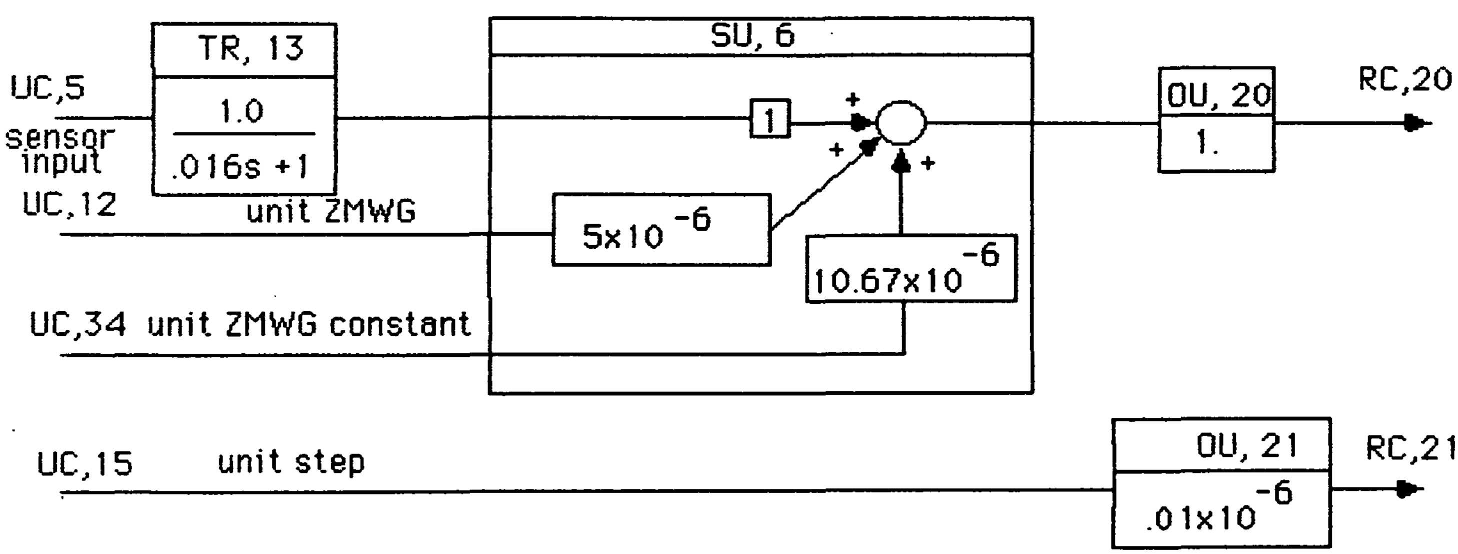
TREETOPS CONTINUOUS CONTROLLER – TORQUE MOTOR (ROLL)



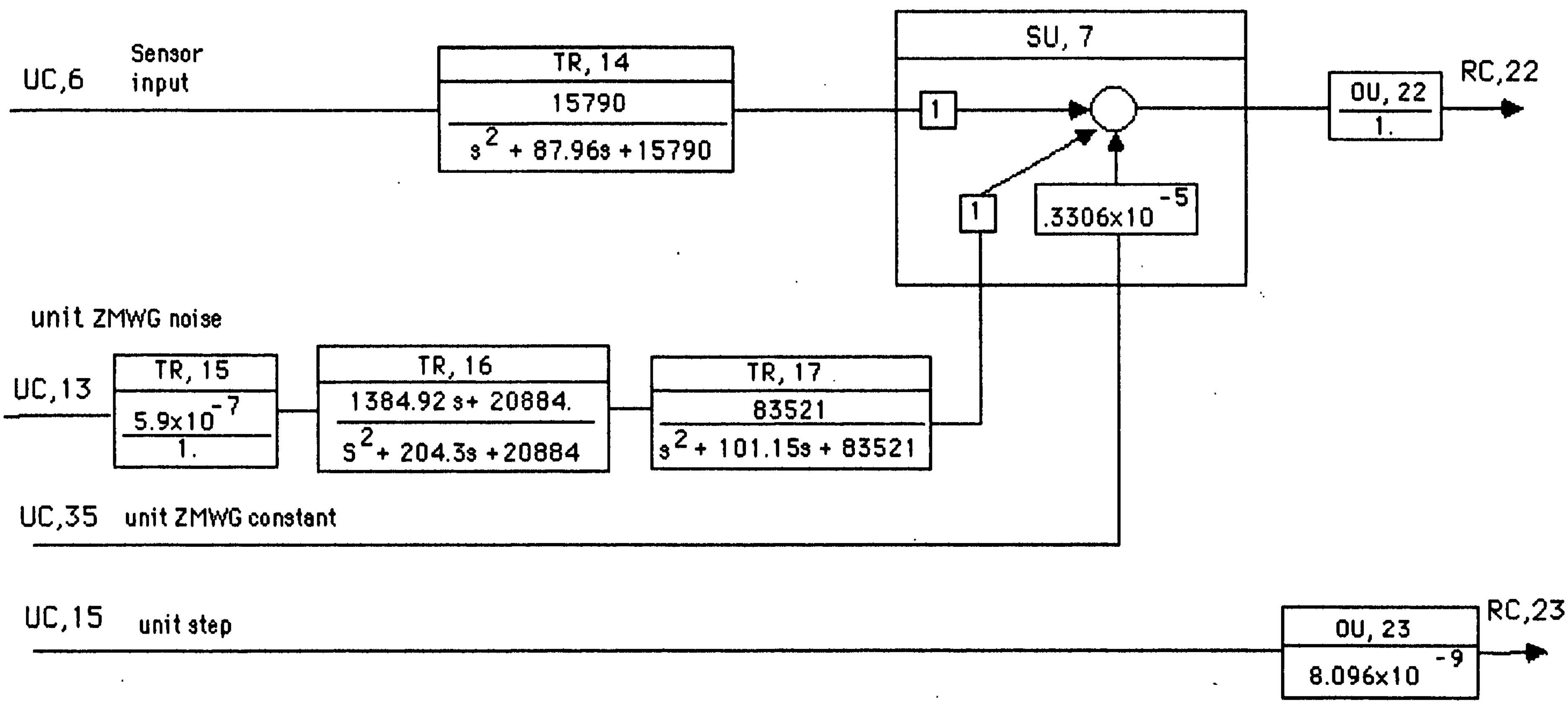
TREETOPS CONTINUOUS CONTROLLER - LOS SENSOR (PITCH)



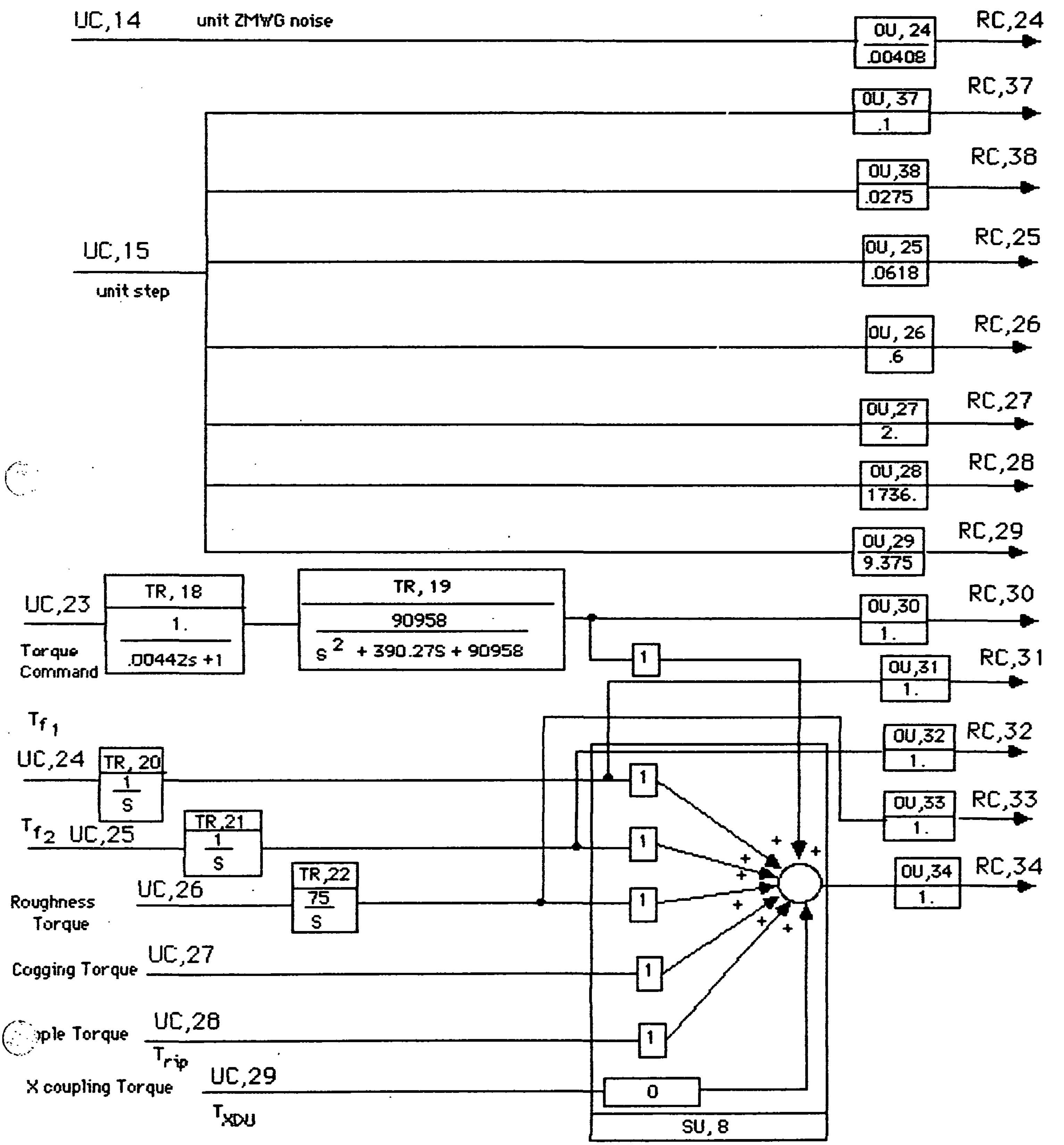
TREETOPS CONTINUOUS CONTROLLER - LASER INTERFERFEROMETER(PITCH)



TREETOPS CONTINUOUS CONTROLLER - RATE GYRO SENSOR(PITCH)



TREETOPS CONTINUOUS CONTROLLER - TORQUE MOTOR (PITCH)



6.0 COMPUTER LISTINGS

The interactive file (JRRPH2.INT), user controller source file (POFCONT.FOR) and user defined function generator source file(NOISE.FOR) required to simulate the P/OF pointing performance with corrupted sensor and actuator outputs are supplied in this section. A different sequence of random numbers will be produced for each TREETOPS run. The dynamics of the sensor/actuator models dictate that the simulation should be run with a step size of .004 sec. A 20 second run on the Honeywell VAX uses approximately 45 CPU minutes.

1.000	SIM CONTROL	#	0	Title header of user problem (< 40 char)	=	JRRPH2	:	
2.000	SIM CONTROL	#	0	Simulation stop time(sec),Job time(min);Core(kwds)	=	20.000	1.0000	1.0000
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	0.40000E-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	3	Type(C=contin,D=discrete,U=user), Dt(sec), Fname	=	CONTINUOUS		
11.000	CONTROLLER	C	3	Number of inputs, Number of outputs	=	35.000	40.000	
12.000	CONTROLLER	C	3	OUTPUT 1 type(I=incont,T=trans,J=junct), ID#,Gain	=	J	1.0000	1.0000
13.000	CONTROLLER	C	3	OUTPUT 2 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	15.000	0.67874E-06
14.000	CONTROLLER	C	3	OUTPUT 3 Type(I=incont,T=trans,J=junct), ID#,Gain	=	J	2.0000	1.0000
15.000	CONTROLLER	C	3	OUTPUT 4 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	15.000	0.10000E-07
16.000	CONTROLLER	C	3	OUTPUT 5 Type(I=incont,T=trans,J=junct), ID#,Gain	=	J	3.0000	1.0000
17.000	CONTROLLER	C	3	OUTPUT 6 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	15.000	0.80960E-08
18.000	CONTROLLER	C	3	OUTPUT 7 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	10.000	0.40800E-02
19.000	CONTROLLER	C	3	OUTPUT 8 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	15.000	0.61800E-01
20.000	CONTROLLER	C	3	OUTPUT 9 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	15.000	0.60000
21.000	CONTROLLER	C	3	OUTPUT 10 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	15.000	2.0000
22.000	CONTROLLER	C	3	OUTPUT 11 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	15.000	1736.0
23.000	CONTROLLER	C	3	OUTPUT 12 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	15.000	9.3750
24.000	CONTROLLER	C	3	OUTPUT 13 Type(I=incont,T=trans,J=junct), ID#,Gain	=	T	8.0000	1.0000
25.000	CONTROLLER	C	3	OUTPUT 14 Type(I=incont,T=trans,J=junct), ID#,Gain	=	T	9.0000	1.0000
26.000	CONTROLLER	C	3	OUTPUT 15 Type(I=incont,T=trans,J=junct), ID#,Gain	=	T	10.000	1.0000
27.000	CONTROLLER	C	3	OUTPUT 16 Type(I=incont,T=trans,J=junct), ID#,Gain	=	T	11.000	1.0000
28.000	CONTROLLER	C	3	OUTPUT 17 Type(I=incont,T=trans,J=junct), ID#,Gain	=	J	4.0000	1.0000
29.000	CONTROLLER	C	3	OUTPUT 18 Type(I=incont,T=trans,J=junct), ID#,Gain	=	J	5.0000	1.0000
30.000	CONTROLLER	C	3	OUTPUT 19 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	15.000	0.67874E-06
31.000	CONTROLLER	C	3	OUTPUT 20 Type(I=incont,T=trans,J=junct), ID#,Gain	=	J	6.0000	1.0000
32.000	CONTROLLER	C	3	OUTPUT 21 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	15.000	0.10000E-07
33.000	CONTROLLER	C	3	OUTPUT 22 Type(I=incont,T=trans,J=junct), ID#,Gain	=	J	7.0000	1.0000
34.000	CONTROLLER	C	3	OUTPUT 23 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	15.000	0.80960E-08
35.000	CONTROLLER	C	3	OUTPUT 24 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	14.000	0.40800E-02
36.000	CONTROLLER	C	3	OUTPUT 25 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	15.000	0.61800E-01
37.000	CONTROLLER	C	3	OUTPUT 26 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	15.000	0.60000
38.000	CONTROLLER	C	3	OUTPUT 27 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	15.000	2.0000
39.000	CONTROLLER	C	3	OUTPUT 28 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	15.000	1736.0
40.000	CONTROLLER	C	3	OUTPUT 29 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	15.000	9.3750
41.000	CONTROLLER	C	3	OUTPUT 30 Type(I=incont,T=trans,J=junct), ID#,Gain	=	T	19.000	1.0000
42.000	CONTROLLER	C	3	OUTPUT 31 Type(I=incont,T=trans,J=junct), ID#,Gain	=	T	20.000	1.0000
43.000	CONTROLLER	C	3	OUTPUT 32 Type(I=incont,T=trans,J=junct), ID#,Gain	=	T	21.000	1.0000
44.000	CONTROLLER	C	3	OUTPUT 33 Type(I=incont,T=trans,J=junct), ID#,Gain	=	T	22.000	1.0000
45.000	CONTROLLER	C	3	OUTPUT 34 Type(I=incont,T=trans,J=junct), ID#,Gain	=	J	8.0000	1.0000
46.000	CONTROLLER	C	3	OUTPUT 35 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	15.000	0.10000
47.000	CONTROLLER	C	3	OUTPUT 36 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	15.000	0.27500E-01
48.000	CONTROLLER	C	3	OUTPUT 37 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	15.000	0.10000
49.000	CONTROLLER	C	3	OUTPUT 38 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	15.000	0.27500E-01
50.000	CONTROLLER	C	3	OUTPUT 39 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	1.0000	0.20626E+06
51.000	CONTROLLER	C	3	OUTPUT 40 Type(I=incont,T=trans,J=junct), ID#,Gain	=	I	4.0000	0.20626E+06
52.000	TRANS FUNCT	C	1	Input type(I=incont,T=trans,J=junct), Input ID#	=	I	1.0000	CONTINUOUS
53.000	TRANS FUNCT	C	1	Order of numerator, Order of denominator	=	0.00000E+00	1.0000	
54.000	TRANS FUNCT	C	1	Numerator coeff (ascending order 1-4 per line)	=	1.0000		
55.000	TRANS FUNCT	C	1	Denominator coeff (ascending order 1-4 per line)	=	1.0000	0.16000E-01	
56.000	TRANS FUNCT	C	2	Input type(I=incont,T=trans,J=junct), Input ID#	=	I	2.0000	CONTINUOUS
57.000	TRANS FUNCT	C	2	Order of numerator, Order of denominator	=	0.00000E+00	1.0000	
58.000	TRANS FUNCT	C	2	Numerator coeff (ascending order 1-4 per line)	=	1.0000		
59.000	TRANS FUNCT	C	2	Denominator coeff (ascending order 1-4 per line)	=	1.0000	0.16000E-01	
60.000	TRANS FUNCT	C	3	Input type(I=incont,T=trans,J=junct), Input ID#	=	I	3.0000	CONTINUOUS
61.000	TRANS FUNCT	C	3	Order of numerator, Order of denominator	=	0.00000E+00	2.0000	

62.000	TRANS FUNCT C	3	Numerator coeff (ascending order 1-4 per line)	=	15790.		
63.000	TRANS FUNCT C	3	Denominator coeff (ascending order 1-4 per line)	=	15790.	87.960	1.0000
64.000	TRANS FUNCT C	4	Input type(I=incont,T=trans,J=junct), Input ID#	=	I	9.0000	CONTINUOUS
65.000	TRANS FUNCT C	4	Order of numerator, Order of denominator	=	0.00000E+00	0.00000E+00	
66.000	TRANS FUNCT C	4	Numerator coeff (ascending order 1-4 per line)	=	0.59000E-06		
67.000	TRANS FUNCT C	4	Denominator coeff (ascending order 1-4 per line)	=	1.0000		
68.000	TRANS FUNCT C	5	Input type(I=incont,T=trans,J=junct), Input ID#	=	T	4.0000	CONTINUOUS
69.000	TRANS FUNCT C	5	Order of numerator, Order of denominator	=	1.0000	2.0000	
70.000	TRANS FUNCT C	5	Numerator coeff (ascending order 1-4 per line)	=	20884.	1384.9	
71.000	TRANS FUNCT C	5	Denominator coeff (ascending order 1-4 per line)	=	20884.	204.30	1.0000
72.000	TRANS FUNCT C	6	Input type(I=incont,T=trans,J=junct), Input ID#	=	T	5.0000	CONTINUOUS
73.000	TRANS FUNCT C	6	Order of numerator, Order of denominator	=	0.00000E+00	2.0000	
74.000	TRANS FUNCT C	6	Numerator coeff (ascending order 1-4 per line)	=	83521.		
75.000	TRANS FUNCT C	6	Denominator coeff (ascending order 1-4 per line)	=	83521.	101.15	1.0000
76.000	TRANS FUNCT C	7	Input type(I=incont,T=trans,J=junct), Input ID#	=	I	16.000	CONTINUOUS
77.000	TRANS FUNCT C	7	Order of numerator, Order of denominator	=	0.00000E+00	1.0000	
78.000	TRANS FUNCT C	7	Numerator coeff (ascending order 1-4 per line)	=	1.0000		
79.000	TRANS FUNCT C	7	Denominator coeff (ascending order 1-4 per line)	=	1.0000	0.44200E-02	
80.000	TRANS FUNCT C	8	Input type(I=incont,T=trans,J=junct), Input ID#	=	T	7.0000	CONTINUOUS
81.000	TRANS FUNCT C	8	Order of numerator, Order of denominator	=	0.00000E+00	2.0000	
82.000	TRANS FUNCT C	8	Numerator coeff (ascending order 1-4 per line)	=	90958.		
83.000	TRANS FUNCT C	8	Denominator coeff (ascending order 1-4 per line)	=	90958.	390.27	1.0000
84.000	TRANS FUNCT C	9	Input type(I=incont,T=trans,J=junct), Input ID#	=	I	17.000	CONTINUOUS
85.000	TRANS FUNCT C	9	Order of numerator, Order of denominator	=	0.00000E+00	1.0000	
86.000	TRANS FUNCT C	9	Numerator coeff (ascending order 1-4 per line)	=	1.0000		
87.000	TRANS FUNCT C	9	Denominator coeff (ascending order 1-4 per line)	=	0.00000E+00	1.0000	
88.000	TRANS FUNCT C	10	Input type(I=incont,T=trans,J=junct), Input ID#	=	I	18.000	CONTINUOUS
89.000	TRANS FUNCT C	10	Order of numerator, Order of denominator	=	0.00000E+00	1.0000	
90.000	TRANS FUNCT C	10	Numerator coeff (ascending order 1-4 per line)	=	1.0000		
91.000	TRANS FUNCT C	10	Denominator coeff (ascending order 1-4 per line)	=	0.00000E+00	1.0000	
92.000	TRANS FUNCT C	11	Input type(I=incont,T=trans,J=junct), Input ID#	=	I	19.000	CONTINUOUS
93.000	TRANS FUNCT C	11	Order of numerator, Order of denominator	=	0.00000E+00	1.0000	
94.000	TRANS FUNCT C	11	Numerator coeff (ascending order 1-4 per line)	=	75.000		
95.000	TRANS FUNCT C	11	Denominator coeff (ascending order 1-4 per line)	=	0.00000E+00	1.0000	
96.000	TRANS FUNCT C	12	Input type(I=incont,T=trans,J=junct), Input ID#	=	I	4.0000	CONTINUOUS
97.000	TRANS FUNCT C	12	Order of numerator, Order of denominator	=	0.00000E+00	1.0000	
98.000	TRANS FUNCT C	12	Numerator coeff (ascending order 1-4 per line)	=	1.0000		
99.000	TRANS FUNCT C	12	Denominator coeff (ascending order 1-4 per line)	=	1.0000	0.16000E-01	
100.000	TRANS FUNCT C	13	Input type(I=incont,T=trans,J=junct), Input ID#	=	I	5.0000	CONTINUOUS
101.000	TRANS FUNCT C	13	Order of numerator, Order of denominator	=	0.00000E+00	1.0000	
102.000	TRANS FUNCT C	13	Numerator coeff (ascending order 1-4 per line)	=	1.0000		
103.000	TRANS FUNCT C	13	Denominator coeff (ascending order 1-4 per line)	=	1.0000	0.16000E-01	
104.000	TRANS FUNCT C	14	Input type(I=incont,T=trans,J=junct), Input ID#	=	I	6.0000	CONTINUOUS
105.000	TRANS FUNCT C	14	Order of numerator, Order of denominator	=	0.00000E+00	2.0000	
106.000	TRANS FUNCT C	14	Numerator coeff (ascending order 1-4 per line)	=	15790.		
107.000	TRANS FUNCT C	14	Denominator coeff (ascending order 1-4 per line)	=	15790.	87.960	1.0000
108.000	TRANS FUNCT C	15	Input type(I=incont,T=trans,J=junct), Input ID#	=	I	13.000	CONTINUOUS
109.000	TRANS FUNCT C	15	Order of numerator, Order of denominator	=	0.00000E+00	0.00000E+00	
110.000	TRANS FUNCT C	15	Numerator coeff (ascending order 1-4 per line)	=	0.59000E-06		
111.000	TRANS FUNCT C	15	Denominator coeff (ascending order 1-4 per line)	=	1.0000		
112.000	TRANS FUNCT C	16	Input type(I=incont,T=trans,J=junct), Input ID#	=	T	15.000	CONTINUOUS
113.000	TRANS FUNCT C	16	Order of numerator, Order of denominator	=	1.0000	2.0000	
114.000	TRANS FUNCT C	16	Numerator coeff (ascending order 1-4 per line)	=	20884.	1384.9	
115.000	TRANS FUNCT C	16	Denominator coeff (ascending order 1-4 per line)	=	20884.	204.30	1.0000
116.000	TRANS FUNCT C	17	Input type(I=incont,T=trans,J=junct), Input ID#	=	T	16.000	CONTINUOUS
117.000	TRANS FUNCT C	17	Order of numerator, Order of denominator	=	0.00000E+00	2.0000	
118.000	TRANS FUNCT C	17	Numerator coeff (ascending order 1-4 per line)	=	83521.		
119.000	TRANS FUNCT C	17	Denominator coeff (ascending order 1-4 per line)	=	83521.	101.15	1.0000
120.000	TRANS FUNCT C	18	Input type(I=incont,T=trans,J=junct), Input ID#	=	I	23.000	CONTINUOUS
121.000	TRANS FUNCT C	18	Order of numerator, Order of denominator	=	0.00000E+00	1.0000	
122.000	TRANS FUNCT C	18	Numerator coeff (ascending order 1-4 per line)	=	1.0000		

123.000	TRANS FUNCT	C	18	Denominator coeff (ascending order 1-4 per line)	=	1.0000	0.44200E-02	
124.000	TRANS FUNCT	C	19	Input type(I=incont,T=trans,J=junct), Input ID#	=	T	18.000	CONTINUOUS
125.000	TRANS FUNCT	C	19	Order of numerator, Order of denominator	=	0.00000E+00	2.0000	
126.000	TRANS FUNCT	C	19	Numerator coeff (ascending order 1-4 per line)	=	90958.		
127.000	TRANS FUNCT	C	19	Denominator coeff (ascending order 1-4 per line)	=	90958.	390.27	1.0000
128.000	TRANS FUNCT	C	20	Input type(I=incont,T=trans,J=junct), Input ID#	=	I	24.000	CONTINUOUS
129.000	TRANS FUNCT	C	20	Order of numerator, Order of denominator	=	0.00000E+00	1.0000	
130.000	TRANS FUNCT	C	20	Numerator coeff (ascending order 1-4 per line)	=	1.0000		
131.000	TRANS FUNCT	C	20	Denominator coeff (ascending order 1-4 per line)	=	0.00000E+00	1.0000	
132.000	TRANS FUNCT	C	21	Input type(I=incont,T=trans,J=junct), Input ID#	=	I	25.000	CONTINUOUS
133.000	TRANS FUNCT	C	21	Order of numerator, Order of denominator	=	0.00000E+00	1.0000	
134.000	TRANS FUNCT	C	21	Numerator coeff (ascending order 1-4 per line)	=	1.0000		
135.000	TRANS FUNCT	C	21	Denominator coeff (ascending order 1-4 per line)	=	0.00000E+00	1.0000	
136.000	TRANS FUNCT	C	22	Input type(I=incont,T=trans,J=junct), Input ID#	=	I	26.000	CONTINUOUS
137.000	TRANS FUNCT	C	22	Order of numerator, Order of denominator	=	0.00000E+00	1.0000	
138.000	TRANS FUNCT	C	22	Numerator coeff (ascending order 1-4 per line)	=	75.000		
139.000	TRANS FUNCT	C	22	Denominator coeff (ascending order 1-4 per line)	=	0.00000E+00	1.0000	
140.000	TRANS FUNCT	C	22	>>>>>***** END OF DATA *****<<<<<	=			
141.000	SUM JUNCT	C	1	Number of inputs to summing junction	=	3.0000	CONTINUOUS	
142.000	SUM JUNCT	C	1	INPUT 1 type(I=incont,T=trans,J=junct), ID#,Gain=		T	1.0000	1.0000
143.000	SUM JUNCT	C	1	INPUT 2 type(I=incont,T=trans,J=junct), ID#,Gain=		I	7.0000	0.12120E-06
144.000	SUM JUNCT	C	1	INPUT 3 type(I=incont,T=trans,J=junct), ID#,Gain=		I	30.000	0.12120E-05
145.000	SUM JUNCT	C	2	Number of inputs to summing junction	=	3.0000	CONTINUOUS	
146.000	SUM JUNCT	C	2	INPUT 1 type(I=incont,T=trans,J=junct), ID#,Gain=		T	2.0000	1.0000
147.000	SUM JUNCT	C	2	INPUT 2 type(I=incont,T=trans,J=junct), ID#,Gain=		I	8.0000	0.50000E-05
148.000	SUM JUNCT	C	2	INPUT 3 type(I=incont,T=trans,J=junct), ID#,Gain=		I	31.000	0.10670E-04
149.000	SUM JUNCT	C	3	Number of inputs to summing junction	=	3.0000	CONTINUOUS	
150.000	SUM JUNCT	C	3	INPUT 1 type(I=incont,T=trans,J=junct), ID#,Gain=		T	3.0000	1.0000
151.000	SUM JUNCT	C	3	INPUT 2 type(I=incont,T=trans,J=junct), ID#,Gain=		T	6.0000	1.0000
152.000	SUM JUNCT	C	3	INPUT 3 type(I=incont,T=trans,J=junct), ID#,Gain=		I	32.000	0.33060E-05
153.000	SUM JUNCT	C	4	Number of inputs to summing junction	=	7.0000	CONTINUOUS	
154.000	SUM JUNCT	C	4	INPUT 1 type(I=incont,T=trans,J=junct), ID#,Gain=		T	8.0000	1.0000
155.000	SUM JUNCT	C	4	INPUT 2 type(I=incont,T=trans,J=junct), ID#,Gain=		T	9.0000	1.0000
156.000	SUM JUNCT	C	4	INPUT 3 type(I=incont,T=trans,J=junct), ID#,Gain=		T	10.000	1.0000
157.000	SUM JUNCT	C	4	INPUT 4 type(I=incont,T=trans,J=junct), ID#,Gain=		T	11.000	1.0000
158.000	SUM JUNCT	C	4	INPUT 5 type(I=incont,T=trans,J=junct), ID#,Gain=		I	20.000	1.0000
159.000	SUM JUNCT	C	4	INPUT 6 type(I=incont,T=trans,J=junct), ID#,Gain=		I	21.000	1.0000
160.000	SUM JUNCT	C	4	INPUT 7 type(I=incont,T=trans,J=junct), ID#,Gain=		I	22.000	0.87260E-02
161.000	SUM JUNCT	C	5	Number of inputs to summing junction	=	3.0000	CONTINUOUS	
162.000	SUM JUNCT	C	5	INPUT 1 type(I=incont,T=trans,J=junct), ID#,Gain=		T	12.000	1.0000
163.000	SUM JUNCT	C	5	INPUT 2 type(I=incont,T=trans,J=junct), ID#,Gain=		I	11.000	0.12120E-06
164.000	SUM JUNCT	C	5	INPUT 3 type(I=incont,T=trans,J=junct), ID#,Gain=		I	33.000	0.12120E-05
165.000	SUM JUNCT	C	6	Number of inputs to summing junction	=	3.0000	CONTINUOUS	
166.000	SUM JUNCT	C	6	INPUT 1 type(I=incont,T=trans,J=junct), ID#,Gain=		T	13.000	1.0000
167.000	SUM JUNCT	C	6	INPUT 2 type(I=incont,T=trans,J=junct), ID#,Gain=		I	12.000	0.50000E-05
168.000	SUM JUNCT	C	6	INPUT 3 type(I=incont,T=trans,J=junct), ID#,Gain=		I	34.000	0.10670E-04
169.000	SUM JUNCT	C	7	Number of inputs to summing junction	=	3.0000	CONTINUOUS	
170.000	SUM JUNCT	C	7	INPUT 1 type(I=incont,T=trans,J=junct), ID#,Gain=		T	14.000	1.0000
171.000	SUM JUNCT	C	7	INPUT 2 type(I=incont,T=trans,J=junct), ID#,Gain=		T	17.000	1.0000
172.000	SUM JUNCT	C	7	INPUT 3 type(I=incont,T=trans,J=junct), ID#,Gain=		I	35.000	0.33060E-05
173.000	SUM JUNCT	C	8	Number of inputs to summing junction	=	7.0000	CONTINUOUS	
174.000	SUM JUNCT	C	8	INPUT 1 type(I=incont,T=trans,J=junct), ID#,Gain=		T	19.000	1.0000
175.000	SUM JUNCT	C	8	INPUT 2 type(I=incont,T=trans,J=junct), ID#,Gain=		T	20.000	1.0000
176.000	SUM JUNCT	C	8	INPUT 3 type(I=incont,T=trans,J=junct), ID#,Gain=		T	21.000	1.0000
177.000	SUM JUNCT	C	8	INPUT 4 type(I=incont,T=trans,J=junct), ID#,Gain=		T	22.000	1.0000
178.000	SUM JUNCT	C	8	INPUT 5 type(I=incont,T=trans,J=junct), ID#,Gain=		I	27.000	1.0000
179.000	SUM JUNCT	C	8	INPUT 6 type(I=incont,T=trans,J=junct), ID#,Gain=		I	28.000	1.0000
180.000	SUM JUNCT	C	8	INPUT 7 type(I=incont,T=trans,J=junct), ID#,Gain=		I	29.000	0.00000E+00
181.000	SUM JUNCT	C	8	>>>>>***** END OF DATA *****<<<<<	=			
182.000	CONTROLLER	U	2	Type(C=contin,D=discrete,U=user), Dt(sec), Fname	=	USER	0.40000E-01	POFCONT
183.000	CONTROLLER	U	2	Number of inputs, Number of outputs	=	59.000	36.000	

```
184.000  FUNCT GENER #   1 type(step,ramp,pulse,sawtooth,sine,user)           =  USER
185.000  FUNCT GENER #   1 user fname(use same name for additional usergener)=  NOISE
186.000  FUNCT GENER #   1 >>>>>***** END OF DATA *****<<<<<                  =
187.000  FUNCT GENER #   2 type(step,ramp,pulse,sawtooth,sine,user)           =  USER
188.000  FUNCT GENER #   2 user fname(use same name for additional usergener)=  NOISE
189.000  FUNCT GENER #   2 >>>>>***** END OF DATA *****<<<<<                  =
190.000  FUNCT GENER #   3 type(step,ramp,pulse,sawtooth,sine,user)           =  USER
191.000  FUNCT GENER #   3 user fname(use same name for additional usergener)=  NOISE
192.000  FUNCT GENER #   3 >>>>>***** END OF DATA *****<<<<<                  =
193.000  FUNCT GENER #   4 type(step,ramp,pulse,sawtooth,sine,user)           =  USER
194.000  FUNCT GENER #   4 user fname(use same name for additional usergener)=  NOISE
195.000  FUNCT GENER #   4 >>>>>***** END OF DATA *****<<<<<                  =
196.000  FUNCT GENER #   5 type(step,ramp,pulse,sawtooth,sine,user)           =  USER
197.000  FUNCT GENER #   5 user fname(use same name for additional usergener)=  NOISE
198.000  FUNCT GENER #   5 >>>>>***** END OF DATA *****<<<<<                  =
199.000  FUNCT GENER #   6 type(step,ramp,pulse,sawtooth,sine,user)           =  USER
200.000  FUNCT GENER #   6 user fname(use same name for additional usergener)=  NOISE
201.000  FUNCT GENER #   6 >>>>>***** END OF DATA *****<<<<<                  =
202.000  FUNCT GENER #   7 type(step,ramp,pulse,sawtooth,sine,user)           =  USER
203.000  FUNCT GENER #   7 user fname(use same name for additional usergener)=  NOISE
204.000  FUNCT GENER #   7 >>>>>***** END OF DATA *****<<<<<                  =
205.000  FUNCT GENER #   8 type(step,ramp,pulse,sawtooth,sine,user)           =  USER
206.000  FUNCT GENER #   8 user fname(use same name for additional usergener)=  NOISE
207.000  FUNCT GENER #   8 >>>>>***** END OF DATA *****<<<<<                  =
208.000  FUNCT GENER #   9 type(step,ramp,pulse,sawtooth,sine,user)           =  STEP
209.000  FUNCT GENER #   9 step amplitude, start time(sec)                    =    1.0000       0.00000E+00
210.000  FUNCT GENER #   9 >>>>>***** END OF DATA *****<<<<<                  =
211.000  FUNCT GENER #  10 type(step,ramp,pulse,sawtooth,sine,user)           =  STEP
212.000  FUNCT GENER #  10 step amplitude, start time(sec)                    =  -0.20000       0.00000E+00
213.000  FUNCT GENER #  10 >>>>>***** END OF DATA *****<<<<<                  =
214.000  FUNCT GENER #  11 type(step,ramp,pulse,sawtooth,sine,user)           =  STEP
215.000  FUNCT GENER #  11 step amplitude, start time(sec)                    =   0.20000       0.00000E+00
216.000  FUNCT GENER #  11 >>>>>***** END OF DATA *****<<<<<                  =
217.000  FUNCT GENER #  12 type(step,ramp,pulse,sawtooth,sine,user)           =  STEP
218.000  FUNCT GENER #  12 step amplitude, start time(sec)                    =  -0.20000       0.00000E+00
219.000  FUNCT GENER #  12 >>>>>***** END OF DATA *****<<<<<                  =
220.000  FUNCT GENER #  13 type(step,ramp,pulse,sawtooth,sine,user)           =  USER
221.000  FUNCT GENER #  13 user fname(use same name for additional usergener)=  NOISE
222.000  FUNCT GENER #  13 >>>>>***** END OF DATA *****<<<<<                  =
223.000  FUNCT GENER #  14 type(step,ramp,pulse,sawtooth,sine,user)           =  USER
224.000  FUNCT GENER #  14 user fname(use same name for additional usergener)=  NOISE
225.000  FUNCT GENER #  14 >>>>>***** END OF DATA *****<<<<<                  =
226.000  FUNCT GENER #  15 type(step,ramp,pulse,sawtooth,sine,user)           =  USER
227.000  FUNCT GENER #  15 user fname(use same name for additional usergener)=  NOISE
228.000  FUNCT GENER #  15 >>>>>***** END OF DATA *****<<<<<                  =
229.000  FUNCT GENER #  16 type(step,ramp,pulse,sawtooth,sine,user)           =  USER
230.000  FUNCT GENER #  16 user fname(use same name for additional usergener)=  NOISE
231.000  FUNCT GENER #  16 >>>>>***** END OF DATA *****<<<<<                  =
232.000  FUNCT GENER #  17 type(step,ramp,pulse,sawtooth,sine,user)           =  USER
233.000  FUNCT GENER #  17 user fname(use same name for additional usergener)=  NOISE
234.000  FUNCT GENER #  17 >>>>>***** END OF DATA *****<<<<<                  =
235.000  FUNCT GENER #  18 type(step,ramp,pulse,sawtooth,sine,user)           =  USER
236.000  FUNCT GENER #  18 user fname(use same name for additional usergener)=  NOISE
237.000  FUNCT GENER #  18 >>>>>***** END OF DATA *****<<<<<                  =
238.000  FUNCT GENER #  19 type(step,ramp,pulse,sawtooth,sine,user)           =  USER
239.000  FUNCT GENER #  19 user fname(use same name for additional usergener)=  NOISE
240.000  FUNCT GENER #  19 >>>>>***** END OF DATA *****<<<<<                  =
241.000  FUNCT GENER #  20 type(step,ramp,pulse,sawtooth,sine,user)           =  USER
242.000  FUNCT GENER #  20 user fname(use same name for additional usergener)=  NOISE
243.000  FUNCT GENER #  20 >>>>>***** END OF DATA *****<<<<<                  =
244.000  FUNCT GENER #  21 type(step,ramp,pulse,sawtooth,sine,user)           =  USER
```

```
245.000  FUNCT GENER #  21 user fname(use same name for additional usergener)=  NOISE
246.000  FUNCT GENER #  21 >>>>>***** END OF DATA *****<<<<<                 =
247.000  FUNCT GENER #  22 type(step,ramp,pulse,sawtooth,sine,user)           =  USER
248.000  FUNCT GENER #  22 user fname(use same name for additional usergener)=  NOISE
249.000  FUNCT GENER #  22 >>>>>***** END OF DATA *****<<<<<                 =
250.000  FUNCT GENER #  23 type(step,ramp,pulse,sawtooth,sine,user)           =  USER
251.000  FUNCT GENER #  23 user fname(use same name for additional usergener)=  NOISE
252.000  FUNCT GENER #  23 >>>>>***** END OF DATA *****<<<<<                 =
253.000  FUNCT GENER #  24 type(step,ramp,pulse,sawtooth,sine,user)           =  USER
254.000  FUNCT GENER #  24 user fname(use same name for additional usergener)=  NOISE
255.000  FUNCT GENER #  24 >>>>>***** END OF DATA *****<<<<<                 =
256.000  SENSOR      #   1 Type(Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos), ID#  =  ACCEL          1.0000
257.000  SENSOR      #   1 ID# of body, index of node location               =   1.0000        3.0000
258.000  SENSOR      #   1 Input axis unit vector (body coord) X,Y,Z         =   1.0000        0.00000E+00  0.00000E+00
259.000  SENSOR      #   1 >>>>>***** END OF DATA *****<<<<<                 =
260.000  SENSOR      #   2 Type(Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos), ID#  =  ACCEL          2.0000
261.000  SENSOR      #   2 ID# of body, index of node location               =   1.0000        3.0000
262.000  SENSOR      #   2 Input axis unit vector (body coord) X,Y,Z         =   0.00000E+00   1.0000       0.00000E+00
263.000  SENSOR      #   2 >>>>>***** END OF DATA *****<<<<<                 =
264.000  SENSOR      #   3 Type(Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos), ID#  =  GYRO           3.0000
265.000  SENSOR      #   3 ID# of body, index of node location               =   2.0000        3.0000
266.000  SENSOR      #   3 Input axis unit vector (body coord) X,Y,Z         =   1.0000        0.00000E+00  0.00000E+00
267.000  SENSOR      #   3 >>>>>***** END OF DATA *****<<<<<                 =
268.000  SENSOR      #   4 Type(Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos), ID#  =  GYRO           4.0000
269.000  SENSOR      #   4 ID# of body, index of node location               =   2.0000        3.0000
270.000  SENSOR      #   4 Input axis unit vector (body coord) X,Y,Z         =   0.00000E+00   1.0000       0.00000E+00
271.000  SENSOR      #   4 >>>>>***** END OF DATA *****<<<<<                 =
272.000  SENSOR      #   5 Type(Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos), ID#  =  GYRO           5.0000
273.000  SENSOR      #   5 ID# of body, index of node location               =   2.0000        3.0000
274.000  SENSOR      #   5 Input axis unit vector (body coord) X,Y,Z         =   0.00000E+00   0.00000E+00   1.0000
275.000  SENSOR      #   5 >>>>>***** END OF DATA *****<<<<<                 =
276.000  SENSOR      #  99 Type Resolver,Tach,Gyro,Wdot,Accel,Vel,Pos,Los,ID#=  LOS            99.000
277.000  SENSOR      #  99 ID# of body, index of node location               =   3.0000        2.0000
278.000  SENSOR      #  99 Pinhole target unit vector (inertial coord) X,Y,Z =   0.00000E+00   0.00000E+00   1.0000
279.000  SENSOR      #  99 >>>>>***** END OF DATA *****<<<<<                 =
280.000  ACTUATOR    #   1 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  T              1.0000
281.000  ACTUATOR    #   1 ID# of hinge, index of rotation axis              =   2.0000        1.0000
282.000  ACTUATOR    #   1 >>>>>***** END OF DATA *****<<<<<                 =
283.000  ACTUATOR    #   2 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  T              2.0000
284.000  ACTUATOR    #   2 ID# of hinge, index of rotation axis              =   2.0000        2.0000
285.000  ACTUATOR    #   2 >>>>>***** END OF DATA *****<<<<<                 =
286.000  ACTUATOR    #   3 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  T              3.0000
287.000  ACTUATOR    #   3 ID# of hinge, index of rotation axis              =   2.0000        3.0000
288.000  ACTUATOR    #   3 >>>>>***** END OF DATA *****<<<<<                 =
289.000  ACTUATOR    #   4 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  J              4.0000
290.000  ACTUATOR    #   4 ID# of body, index of node location               =   1.0000        4.0000
291.000  ACTUATOR    #   4 Output axis unit vector(body coord) X,Y,Z         =   0.32650E-01  -0.69625       0.71706
292.000  ACTUATOR    #   4 >>>>>***** END OF DATA *****<<<<<                 =
293.000  ACTUATOR    #   5 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  J              5.0000
294.000  ACTUATOR    #   5 ID# of body, index of node location               =   1.0000        5.0000
295.000  ACTUATOR    #   5 Output axis unit vector(body coord) X,Y,Z         =   0.32650E-01   0.69625       0.71706
296.000  ACTUATOR    #   5 >>>>>***** END OF DATA *****<<<<<                 =
297.000  ACTUATOR    #   6 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  J              6.0000
298.000  ACTUATOR    #   6 ID# of body, index of node location               =   1.0000        6.0000
299.000  ACTUATOR    #   6 Output axis unit vector(body coord) X,Y,Z         =   0.00000E+00  -0.99967       0.25820E-01
300.000  ACTUATOR    #   6 >>>>>***** END OF DATA *****<<<<<                 =
301.000  ACTUATOR    #   7 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  J              7.0000
302.000  ACTUATOR    #   7 ID# of body, index of node location               =   1.0000        7.0000
303.000  ACTUATOR    #   7 Output axis unit vector(body coord) X,Y,Z         =   0.00000E+00   0.99967       0.25820E-01
304.000  ACTUATOR    #   7 >>>>>***** END OF DATA *****<<<<<                 =
305.000  ACTUATOR    #   8 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  J              8.0000
```

```
306.000  ACTUATOR     #   8 ID# of body, index of node location             =    1.0000       8.0000
307.000  ACTUATOR     #   8 Output axis unit vector(body coord) X,Y,Z       =  -0.26560E-01 -0.28662        0.95768
308.000  ACTUATOR     #   8 >>>>>***** END OF DATA *****<<<<<               =
309.000  ACTUATOR     #   9 Type (J=jet,H=hydcyl,C=cmg,T=torque,B=brake), ID# =  J              9.0000
310.000  ACTUATOR     #   9 ID# of body, index of node location             =    1.0000       9.0000
311.000  ACTUATOR     #   9 Output axis unit vector(body coord) X,Y,Z       =  -0.26560E-01  0.28662        0.95768
312.000  ACTUATOR     #   9 >>>>>***** END OF DATA *****<<<<<               =
313.000  INT CONNECT #   2 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  S               2.0000
314.000  INT CONNECT #   2 Destination(C=contin,D=disc,U=user,A=actuator),ID#= U               2.0000
315.000  INT CONNECT #   2 >>>>>***** END OF DATA *****<<<<<               =
316.000  INT CONNECT #   3 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  U               33.000
317.000  INT CONNECT #   3 Destination(C=contin,D=disc,U=user,A=actuator),ID#= C               3.0000
318.000  INT CONNECT #   3 >>>>>***** END OF DATA *****<<<<<               =
319.000  INT CONNECT #   4 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  U               32.000
320.000  INT CONNECT #   4 Destination(C=contin,D=disc,U=user,A=actuator),ID#= C               2.0000
321.000  INT CONNECT #   4 >>>>>***** END OF DATA *****<<<<<               =
322.000  INT CONNECT #   5 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  U               31.000
323.000  INT CONNECT #   5 Destination(C=contin,D=disc,U=user,A=actuator),ID#= C               1.0000
324.000  INT CONNECT #   5 >>>>>***** END OF DATA *****<<<<<               =
325.000  INT CONNECT #   7 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  S               1.0000
326.000  INT CONNECT #   7 Destination(C=contin,D=disc,U=user,A=actuator),ID#= U               7.0000
327.000  INT CONNECT #   7 >>>>>***** END OF DATA *****<<<<<               =
328.000  INT CONNECT #   8 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  U               36.000
329.000  INT CONNECT #   8 Destination(C=contin,D=disc,U=user,A=actuator),ID#= C               6.0000
330.000  INT CONNECT #   8 >>>>>***** END OF DATA *****<<<<<               =
331.000  INT CONNECT #   9 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  U               35.000
332.000  INT CONNECT #   9 Destination(C=contin,D=disc,U=user,A=actuator),ID#= C               5.0000
333.000  INT CONNECT #   9 >>>>>***** END OF DATA *****<<<<<               =
334.000  INT CONNECT #  10 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  U               34.000
335.000  INT CONNECT #  10 Destination(C=contin,D=disc,U=user,A=actuator),ID#= C               4.0000
336.000  INT CONNECT #  10 >>>>>***** END OF DATA *****<<<<<               =
337.000  INT CONNECT #  11 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  U               5.0000
338.000  INT CONNECT #  11 Destination(C=contin,D=disc,U=user,A=actuator),ID#= A               4.0000
339.000  INT CONNECT #  11 >>>>>***** END OF DATA *****<<<<<               =
340.000  INT CONNECT #  12 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  U               6.0000
341.000  INT CONNECT #  12 Destination(C=contin,D=disc,U=user,A=actuator),ID#= A               5.0000
342.000  INT CONNECT #  12 >>>>>***** END OF DATA *****<<<<<               =
343.000  INT CONNECT #  13 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  U               7.0000
344.000  INT CONNECT #  13 Destination(C=contin,D=disc,U=user,A=actuator),ID#= A               6.0000
345.000  INT CONNECT #  13 >>>>>***** END OF DATA *****<<<<<               =
346.000  INT CONNECT #  14 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  U               8.0000
347.000  INT CONNECT #  14 Destination(C=contin,D=disc,U=user,A=actuator),ID#= A               7.0000
348.000  INT CONNECT #  14 >>>>>***** END OF DATA *****<<<<<               =
349.000  INT CONNECT #  15 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  U               9.0000
350.000  INT CONNECT #  15 Destination(C=contin,D=disc,U=user,A=actuator),ID#= A               8.0000
351.000  INT CONNECT #  15 >>>>>***** END OF DATA *****<<<<<               =
352.000  INT CONNECT #  16 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  U               10.000
353.000  INT CONNECT #  16 Destination(C=contin,D=disc,U=user,A=actuator),ID#= A               9.0000
354.000  INT CONNECT #  16 >>>>>***** END OF DATA *****<<<<<               =
355.000  INT CONNECT #  17 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  G               1.0000
356.000  INT CONNECT #  17 Destination(C=contin,D=disc,U=user,A=actuator),ID#= C               7.0000
357.000  INT CONNECT #  17 >>>>>***** END OF DATA *****<<<<<               =
358.000  INT CONNECT #  18 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  G               2.0000
359.000  INT CONNECT #  18 Destination(C=contin,D=disc,U=user,A=actuator),ID#= C               8.0000
360.000  INT CONNECT #  18 >>>>>***** END OF DATA *****<<<<<               =
361.000  INT CONNECT #  19 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  G               3.0000
362.000  INT CONNECT #  19 Destination(C=contin,D=disc,U=user,A=actuator),ID#= C               9.0000
363.000  INT CONNECT #  19 >>>>>***** END OF DATA *****<<<<<               =
364.000  INT CONNECT #  20 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  G               5.0000
365.000  INT CONNECT #  20 Destination(C=contin,D=disc,U=user,A=actuator),ID#= C               11.000
366.000  INT CONNECT #  20 >>>>>***** END OF DATA *****<<<<<               =
```

367.000	INT CONNECT #	21	Source(C=contin,D=disc,U=user,S=sen,G=gen), ID# =	G	6.0000
368.000	INT CONNECT #	21	Destination(C=contin,D=disc,U=user,A=actuator),ID#=	C	12.000
369.000	INT CONNECT #	21	>>>>>***** END OF DATA *****<<<<< =		
370.000	INT CONNECT #	22	Source(C=contin,D=disc,U=user,S=sen,G=gen), ID# =	G	7.0000
371.000	INT CONNECT #	22	Destination(C=contin,D=disc,U=user,A=actuator),ID#=	C	13.000
372.000	INT CONNECT #	22	>>>>>***** END OF DATA *****<<<<< =		
373.000	INT CONNECT #	23	Source(C=contin,D=disc,U=user,S=sen,G=gen), ID# =	G	9.0000
374.000	INT CONNECT #	23	Destination(C=contin,D=disc,U=user,A=actuator),ID#=	C	15.000
375.000	INT CONNECT #	23	>>>>>***** END OF DATA *****<<<<< =		
376.000	INT CONNECT #	24	Source(C=contin,D=disc,U=user,S=sen,G=gen), ID# =	C	1.0000
377.000	INT CONNECT #	24	Destination(C=contin,D=disc,U=user,A=actuator),ID#=	U	5.0000
378.000	INT CONNECT #	24	>>>>>***** END OF DATA *****<<<<< =		
379.000	INT CONNECT #	25	Source(C=contin,D=disc,U=user,S=sen,G=gen), ID# =	C	2.0000
380.000	INT CONNECT #	25	Destination(C=contin,D=disc,U=user,A=actuator),ID#=	U	14.000
381.000	INT CONNECT #	25	>>>>>***** END OF DATA *****<<<<< =		
382.000	INT CONNECT #	26	Source(C=contin,D=disc,U=user,S=sen,G=gen), ID# =	C	3.0000
383.000	INT CONNECT #	26	Destination(C=contin,D=disc,U=user,A=actuator),ID#=	U	4.0000
384.000	INT CONNECT #	26	>>>>>***** END OF DATA *****<<<<< =		
385.000	INT CONNECT #	27	Source(C=contin,D=disc,U=user,S=sen,G=gen), ID# =	C	4.0000
386.000	INT CONNECT #	27	Destination(C=contin,D=disc,U=user,A=actuator),ID#=	U	15.000
387.000	INT CONNECT #	27	>>>>>***** END OF DATA *****<<<<< =		
388.000	INT CONNECT #	28	Source(C=contin,D=disc,U=user,S=sen,G=gen), ID# =	C	5.0000
389.000	INT CONNECT #	28	Destination(C=contin,D=disc,U=user,A=actuator),ID#=	U	3.0000
390.000	INT CONNECT #	28	>>>>>***** END OF DATA *****<<<<< =		
391.000	INT CONNECT #	29	Source(C=contin,D=disc,U=user,S=sen,G=gen), ID# =	C	6.0000
392.000	INT CONNECT #	29	Destination(C=contin,D=disc,U=user,A=actuator),ID#=	U	16.000
393.000	INT CONNECT #	29	>>>>>***** END OF DATA *****<<<<< =		
394.000	INT CONNECT #	30	Source(C=contin,D=disc,U=user,S=sen,G=gen), ID# =	C	18.000
395.000	INT CONNECT #	30	Destination(C=contin,D=disc,U=user,A=actuator),ID#=	U	10.000
396.000	INT CONNECT #	30	>>>>>***** END OF DATA *****<<<<< =		
397.000	INT CONNECT #	31	Source(C=contin,D=disc,U=user,S=sen,G=gen), ID# =	C	19.000
398.000	INT CONNECT #	31	Destination(C=contin,D=disc,U=user,A=actuator),ID#=	U	17.000
399.000	INT CONNECT #	31	>>>>>***** END OF DATA *****<<<<< =		
400.000	INT CONNECT #	32	Source(C=contin,D=disc,U=user,S=sen,G=gen), ID# =	C	20.000
401.000	INT CONNECT #	32	Destination(C=contin,D=disc,U=user,A=actuator),ID#=	U	9.0000
402.000	INT CONNECT #	32	>>>>>***** END OF DATA *****<<<<< =		
403.000	INT CONNECT #	33	Source(C=contin,D=disc,U=user,S=sen,G=gen), ID# =	C	21.000
404.000	INT CONNECT #	33	Destination(C=contin,D=disc,U=user,A=actuator),ID#=	U	18.000
405.000	INT CONNECT #	33	>>>>>***** END OF DATA *****<<<<< =		
406.000	INT CONNECT #	34	Source(C=contin,D=disc,U=user,S=sen,G=gen), ID# =	C	22.000
407.000	INT CONNECT #	34	Destination(C=contin,D=disc,U=user,A=actuator),ID#=	U	8.0000
408.000	INT CONNECT #	34	>>>>>***** END OF DATA *****<<<<< =		
409.000	INT CONNECT #	35	Source(C=contin,D=disc,U=user,S=sen,G=gen), ID# =	C	23.000
410.000	INT CONNECT #	35	Destination(C=contin,D=disc,U=user,A=actuator),ID#=	U	19.000
411.000	INT CONNECT #	35	>>>>>***** END OF DATA *****<<<<< =		
412.000	INT CONNECT #	36	Source(C=contin,D=disc,U=user,S=sen,G=gen), ID# =	U	1.0000
413.000	INT CONNECT #	36	Destination(C=contin,D=disc,U=user,A=actuator),ID#=	C	16.000
414.000	INT CONNECT #	36	>>>>>***** END OF DATA *****<<<<< =		
415.000	INT CONNECT #	37	Source(C=contin,D=disc,U=user,S=sen,G=gen), ID# =	U	17.000
416.000	INT CONNECT #	37	Destination(C=contin,D=disc,U=user,A=actuator),ID#=	C	17.000
417.000	INT CONNECT #	37	>>>>>***** END OF DATA *****<<<<< =		
418.000	INT CONNECT #	38	Source(C=contin,D=disc,U=user,S=sen,G=gen), ID# =	U	18.000
419.000	INT CONNECT #	38	Destination(C=contin,D=disc,U=user,A=actuator),ID#=	C	18.000
420.000	INT CONNECT #	38	>>>>>***** END OF DATA *****<<<<< =		
421.000	INT CONNECT #	39	Source(C=contin,D=disc,U=user,S=sen,G=gen), ID# =	U	19.000
422.000	INT CONNECT #	39	Destination(C=contin,D=disc,U=user,A=actuator),ID#=	C	19.000
423.000	INT CONNECT #	39	>>>>>***** END OF DATA *****<<<<< =		
424.000	INT CONNECT #	40	Source(C=contin,D=disc,U=user,S=sen,G=gen), ID# =	U	20.000
425.000	INT CONNECT #	40	Destination(C=contin,D=disc,U=user,A=actuator),ID#=	C	20.000
426.000	INT CONNECT #	40	>>>>>***** END OF DATA *****<<<<< =		
427.000	INT CONNECT #	41	Source(C=contin,D=disc,U=user,S=sen,G=gen), ID# =	U	21.000

```
428.000  INT CONNECT #  41 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C     21.000
429.000  INT CONNECT #  41 >>>>>***** END OF DATA *****<<<<<              =
430.000  INT CONNECT #  42 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  U     22.000
431.000  INT CONNECT #  42 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C     22.000
432.000  INT CONNECT #  42 >>>>>***** END OF DATA *****<<<<<              =
433.000  INT CONNECT #  43 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  U     3.0000
434.000  INT CONNECT #  43 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C     23.000
435.000  INT CONNECT #  43 >>>>>***** END OF DATA *****<<<<<              =
436.000  INT CONNECT #  44 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  U     23.000
437.000  INT CONNECT #  44 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C     24.000
438.000  INT CONNECT #  44 >>>>>***** END OF DATA *****<<<<<              =
439.000  INT CONNECT #  45 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  U     24.000
440.000  INT CONNECT #  45 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C     25.000
441.000  INT CONNECT #  45 >>>>>***** END OF DATA *****<<<<<              =
442.000  INT CONNECT #  46 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  U     25.000
443.000  INT CONNECT #  46 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C     26.000
444.000  INT CONNECT #  46 >>>>>***** END OF DATA *****<<<<<              =
445.000  INT CONNECT #  47 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  U     26.000
446.000  INT CONNECT #  47 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C     27.000
447.000  INT CONNECT #  47 >>>>>***** END OF DATA *****<<<<<              =
448.000  INT CONNECT #  48 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  U     27.000
449.000  INT CONNECT #  48 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C     28.000
450.000  INT CONNECT #  48 >>>>>***** END OF DATA *****<<<<<              =
451.000  INT CONNECT #  49 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  U     28.000
452.000  INT CONNECT #  49 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C     29.000
453.000  INT CONNECT #  49 >>>>>***** END OF DATA *****<<<<<              =
454.000  INT CONNECT #  50 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  G     4.0000
455.000  INT CONNECT #  50 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C     10.000
456.000  INT CONNECT #  50 >>>>>***** END OF DATA *****<<<<<              =
457.000  INT CONNECT #  51 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  G     8.0000
458.000  INT CONNECT #  51 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C     14.000
459.000  INT CONNECT #  51 >>>>>***** END OF DATA *****<<<<<              =
460.000  INT CONNECT #  52 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  C     7.0000
461.000  INT CONNECT #  52 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U     20.000
462.000  INT CONNECT #  52 >>>>>***** END OF DATA *****<<<<<              =
463.000  INT CONNECT #  53 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  C     8.0000
464.000  INT CONNECT #  53 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U     21.000
465.000  INT CONNECT #  53 >>>>>***** END OF DATA *****<<<<<              =
466.000  INT CONNECT #  54 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  C     9.0000
467.000  INT CONNECT #  54 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U     22.000
468.000  INT CONNECT #  54 >>>>>***** END OF DATA *****<<<<<              =
469.000  INT CONNECT #  55 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  C     10.000
470.000  INT CONNECT #  55 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U     23.000
471.000  INT CONNECT #  55 >>>>>***** END OF DATA *****<<<<<              =
472.000  INT CONNECT #  56 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  C     11.000
473.000  INT CONNECT #  56 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U     24.000
474.000  INT CONNECT #  56 >>>>>***** END OF DATA *****<<<<<              =
475.000  INT CONNECT #  57 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  C     12.000
476.000  INT CONNECT #  57 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U     25.000
477.000  INT CONNECT #  57 >>>>>***** END OF DATA *****<<<<<              =
478.000  INT CONNECT #  58 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  C     13.000
479.000  INT CONNECT #  58 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U     26.000
480.000  INT CONNECT #  58 >>>>>***** END OF DATA *****<<<<<              =
481.000  INT CONNECT #  59 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  C     14.000
482.000  INT CONNECT #  59 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U     27.000
483.000  INT CONNECT #  59 >>>>>***** END OF DATA *****<<<<<              =
484.000  INT CONNECT #  60 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  C     15.000
485.000  INT CONNECT #  60 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U     28.000
486.000  INT CONNECT #  60 >>>>>***** END OF DATA *****<<<<<              =
487.000  INT CONNECT #  61 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#   =  C     16.000
488.000  INT CONNECT #  61 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U     29.000
```

489.000	INT CONNECT #	61 >>>>>***** END OF DATA *****<<<<<	=		
490.000	INT CONNECT #	62 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#	=	C	24.000
491.000	INT CONNECT #	62 Destination(C=contin,D=disc,U=user,A=actuator),ID#	=	U	30.000
492.000	INT CONNECT #	62 >>>>>***** END OF DATA *****<<<<<	=		
493.000	INT CONNECT #	63 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#	=	C	25.000
494.000	INT CONNECT #	63 Destination(C=contin,D=disc,U=user,A=actuator),ID#	=	U	31.000
495.000	INT CONNECT #	63 >>>>>***** END OF DATA *****<<<<<	=		
496.000	INT CONNECT #	64 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#	=	C	26.000
497.000	INT CONNECT #	64 Destination(C=contin,D=disc,U=user,A=actuator),ID#	=	U	32.000
498.000	INT CONNECT #	64 >>>>>***** END OF DATA *****<<<<<	=		
499.000	INT CONNECT #	65 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#	=	C	27.000
500.000	INT CONNECT #	65 Destination(C=contin,D=disc,U=user,A=actuator),ID#	=	U	33.000
501.000	INT CONNECT #	65 >>>>>***** END OF DATA *****<<<<<	=		
502.000	INT CONNECT #	66 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#	=	C	28.000
503.000	INT CONNECT #	66 Destination(C=contin,D=disc,U=user,A=actuator),ID#	=	U	34.000
504.000	INT CONNECT #	66 >>>>>***** END OF DATA *****<<<<<	=		
505.000	INT CONNECT #	67 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#	=	C	29.000
506.000	INT CONNECT #	67 Destination(C=contin,D=disc,U=user,A=actuator),ID#	=	U	35.000
507.000	INT CONNECT #	67 >>>>>***** END OF DATA *****<<<<<	=		
508.000	INT CONNECT #	68 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#	=	C	30.000
509.000	INT CONNECT #	68 Destination(C=contin,D=disc,U=user,A=actuator),ID#	=	U	36.000
510.000	INT CONNECT #	68 >>>>>***** END OF DATA *****<<<<<	=		
511.000	INT CONNECT #	69 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#	=	C	31.000
512.000	INT CONNECT #	69 Destination(C=contin,D=disc,U=user,A=actuator),ID#	=	U	37.000
513.000	INT CONNECT #	69 >>>>>***** END OF DATA *****<<<<<	=		
514.000	INT CONNECT #	70 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#	=	C	32.000
515.000	INT CONNECT #	70 Destination(C=contin,D=disc,U=user,A=actuator),ID#	=	U	38.000
516.000	INT CONNECT #	70 >>>>>***** END OF DATA *****<<<<<	=		
517.000	INT CONNECT #	71 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#	=	C	33.000
518.000	INT CONNECT #	71 Destination(C=contin,D=disc,U=user,A=actuator),ID#	=	U	39.000
519.000	INT CONNECT #	71 >>>>>***** END OF DATA *****<<<<<	=		
520.000	INT CONNECT #	72 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#	=	C	35.000
521.000	INT CONNECT #	72 Destination(C=contin,D=disc,U=user,A=actuator),ID#	=	U	41.000
522.000	INT CONNECT #	72 >>>>>***** END OF DATA *****<<<<<	=		
523.000	INT CONNECT #	73 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#	=	C	36.000
524.000	INT CONNECT #	73 Destination(C=contin,D=disc,U=user,A=actuator),ID#	=	U	42.000
525.000	INT CONNECT #	73 >>>>>***** END OF DATA *****<<<<<	=		
526.000	INT CONNECT #	74 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#	=	C	37.000
527.000	INT CONNECT #	74 Destination(C=contin,D=disc,U=user,A=actuator),ID#	=	U	43.000
528.000	INT CONNECT #	74 >>>>>***** END OF DATA *****<<<<<	=		
529.000	INT CONNECT #	75 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#	=	C	38.000
530.000	INT CONNECT #	75 Destination(C=contin,D=disc,U=user,A=actuator),ID#	=	U	44.000
531.000	INT CONNECT #	75 >>>>>***** END OF DATA *****<<<<<	=		
532.000	INT CONNECT #	76 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#	=	C	17.000
533.000	INT CONNECT #	76 Destination(C=contin,D=disc,U=user,A=actuator),ID#	=	A	1.0000
534.000	INT CONNECT #	76 >>>>>***** END OF DATA *****<<<<<	=		
535.000	INT CONNECT #	77 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#	=	C	34.000
536.000	INT CONNECT #	77 Destination(C=contin,D=disc,U=user,A=actuator),ID#	=	A	2.0000
537.000	INT CONNECT #	77 >>>>>***** END OF DATA *****<<<<<	=		
538.000	INT CONNECT #	78 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#	=	C	39.000
539.000	INT CONNECT #	78 Destination(C=contin,D=disc,U=user,A=actuator),ID#	=	U	46.000
540.000	INT CONNECT #	78 >>>>>***** END OF DATA *****<<<<<	=		
541.000	INT CONNECT #	79 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#	=	C	40.000
542.000	INT CONNECT #	79 Destination(C=contin,D=disc,U=user,A=actuator),ID#	=	U	47.000
543.000	INT CONNECT #	79 >>>>>***** END OF DATA *****<<<<<	=		
544.000	INT CONNECT #	80 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#	=	G	13.000
545.000	INT CONNECT #	80 Destination(C=contin,D=disc,U=user,A=actuator),ID#	=	C	30.000
546.000	INT CONNECT #	80 >>>>>***** END OF DATA *****<<<<<	=		
547.000	INT CONNECT #	81 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#	=	G	14.000
548.000	INT CONNECT #	81 Destination(C=contin,D=disc,U=user,A=actuator),ID#	=	C	31.000
549.000	INT CONNECT #	81 >>>>>***** END OF DATA *****<<<<<	=		

```
550.000  INT CONNECT #  82 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  G            15.000
551.000  INT CONNECT #  82 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C            32.000
552.000  INT CONNECT #  82 >>>>>***** END OF DATA *****<<<<<                  =
553.000  INT CONNECT #  83 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  G            16.000
554.000  INT CONNECT #  83 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C            33.000
555.000  INT CONNECT #  83 >>>>>***** END OF DATA *****<<<<<                  =
556.000  INT CONNECT #  84 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  G            17.000
557.000  INT CONNECT #  84 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C            34.000
558.000  INT CONNECT #  84 >>>>>***** END OF DATA *****<<<<<                  =
559.000  INT CONNECT #  85 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  G            18.000
560.000  INT CONNECT #  85 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  C            35.000
561.000  INT CONNECT #  85 >>>>>***** END OF DATA *****<<<<<                  =
562.000  INT CONNECT #  86 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  S            996.00
563.000  INT CONNECT #  86 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U            48.000
564.000  INT CONNECT #  86 >>>>>***** END OF DATA *****<<<<<                  =
565.000  INT CONNECT #  87 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  S            995.00
566.000  INT CONNECT #  87 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U            49.000
567.000  INT CONNECT #  87 >>>>>***** END OF DATA *****<<<<<                  =
568.000  INT CONNECT #  88 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  S            3.0000
569.000  INT CONNECT #  88 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U            50.000
570.000  INT CONNECT #  88 >>>>>***** END OF DATA *****<<<<<                  =
571.000  INT CONNECT #  89 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  S            997.00
572.000  INT CONNECT #  89 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U            51.000
573.000  INT CONNECT #  89 >>>>>***** END OF DATA *****<<<<<                  =
574.000  INT CONNECT #  90 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  S            994.00
575.000  INT CONNECT #  90 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U            52.000
576.000  INT CONNECT #  90 >>>>>***** END OF DATA *****<<<<<                  =
577.000  INT CONNECT #  91 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  S            4.0000
578.000  INT CONNECT #  91 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U            53.000
579.000  INT CONNECT #  91 >>>>>***** END OF DATA *****<<<<<                  =
580.000  INT CONNECT #  92 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  G            19.000
581.000  INT CONNECT #  92 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U            54.000
582.000  INT CONNECT #  92 >>>>>***** END OF DATA *****<<<<<                  =
583.000  INT CONNECT #  93 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  G            20.000
584.000  INT CONNECT #  93 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U            55.000
585.000  INT CONNECT #  93 >>>>>***** END OF DATA *****<<<<<                  =
586.000  INT CONNECT #  94 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  G            21.000
587.000  INT CONNECT #  94 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U            56.000
588.000  INT CONNECT #  94 >>>>>***** END OF DATA *****<<<<<                  =
589.000  INT CONNECT #  95 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  G            22.000
590.000  INT CONNECT #  95 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U            57.000
591.000  INT CONNECT #  95 >>>>>***** END OF DATA *****<<<<<                  =
592.000  INT CONNECT #  96 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  G            23.000
593.000  INT CONNECT #  96 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U            58.000
594.000  INT CONNECT #  96 >>>>>***** END OF DATA *****<<<<<                  =
595.000  INT CONNECT #  97 Source(C=contin,D=disc,U=user,S=sen,G=gen), ID#    =  G            24.000
596.000  INT CONNECT #  97 Destination(C=contin,D=disc,U=user,A=actuator),ID#=  U            59.000
597.000  INT CONNECT #  97 >>>>>***** END OF DATA *****<<<<<                  =
598.000  HINGE       #   1 ID# of hinge, ID# of inbd body, ID# of outbd body =   1.0000      0.00000E+00   1.0000
599.000  HINGE       #   1 Node index of inboard body attach point           =
600.000  HINGE       #   1 Rotation degrees of freedom                       =   3.0000
601.000  HINGE       #   1 Base body(ID# 1)rotation option(F=free,G=gimbaled)=  G
602.000  HINGE       #   1 Rotation axis unit vector L1 inboard body  X,Y,Z  =   1.0000      0.00000E+00  0.00000E+00
603.000  HINGE       #   1 Rotation axis unit vector L1 outboard body X,Y,Z  =   1.0000      0.00000E+00  0.00000E+00
604.000  HINGE       #   1 Rotation axis unit vector L3 inboard body  X,Y,Z  =  0.00000E+00  0.00000E+00   1.0000
605.000  HINGE       #   1 Rotation axis unit vector L3 outboard body X,Y,Z  =  0.00000E+00  0.00000E+00   1.0000
606.000  HINGE       #   1 Rotation spring stiffness (N/R)   k1,k2,k3        =  0.00000E+00  0.00000E+00  0.00000E+00
607.000  HINGE       #   1 Rotation spring damping (N/R/S)   b1,b2,b3        =  0.00000E+00  0.00000E+00  0.00000E+00
608.000  HINGE       #   1 Initial rotation angle (DEG) theta1,theta2,theta3 =  0.00000E+00  0.00000E+00  0.00000E+00
609.000  HINGE       #   1 Null torque angle (DEG)      theta1,theta2,theta3 =  0.00000E+00  0.00000E+00  0.00000E+00
610.000  HINGE       #   1 Translation degrees of freedom                    =   3.0000
```

```
611.000  HINGE   #  1 Translation axis unit vector of 1ST DOF  X,Y,Z    =   1.0000        0.00000E+00   0.00000E+00
612.000  HINGE   #  1 Translation axis unit vector of 2ND DOF  X,Y,Z    =   0.00000E+00   1.0000        0.00000E+00
613.000  HINGE   #  1 Translation axis unit vector of 3RD DOF  X,Y,Z    =   0.00000E+00   0.00000E+00   1.0000
614.000  HINGE   #  1 Translation spring stiffnes (N/M)  k1,k2,k3       =   0.00000E+00   0.00000E+00   0.00000E+00
615.000  HINGE   #  1 Translation spring damping (N/M/S) b1,b2,b3       =   0.00000E+00   0.00000E+00   0.00000E+00
616.000  HINGE   #  1 Initial translation displacement (M)  y1,y2,y3    =   0.00000E+00   0.00000E+00   0.00000E+00
617.000  HINGE   #  1 Null force position (M)               y1,y2,y3    =   0.00000E+00   0.00000E+00   0.00000E+00
618.000  HINGE   #  1 >>>>>***** END OF DATA *****<<<<<                 =
619.000  HINGE   #  2 ID# of hinge, ID# of inbd body, ID# of outbd body =   2.0000        1.0000        2.0000
620.000  HINGE   #  2 Node index of inboard body attach point           =   2.0000
621.000  HINGE   #  2 Rotation degrees of freedom                       =   3.0000
622.000  HINGE   #  2 Base body(ID# 1)rotation option(F=free,G=gimbaled)=
623.000  HINGE   #  2 Rotation axis unit vector L1 inboard body  X,Y,Z  =   1.0000        0.00000E+00   0.00000E+00
624.000  HINGE   #  2 Rotation axis unit vector L1 outboard body X,Y,Z  =   1.0000        0.00000E+00   0.00000E+00
625.000  HINGE   #  2 Rotation axis unit vector L3 inboard body  X,Y,Z  =   0.00000E+00   0.00000E+00   1.0000
626.000  HINGE   #  2 Rotation axis unit vector L3 outboard body X,Y,Z  =   0.00000E+00   0.00000E+00   1.0000
627.000  HINGE   #  2 Rotation spring stiffness (N/R)   k1,k2,k3        =   0.00000E+00   0.00000E+00   0.00000E+00
628.000  HINGE   #  2 Rotation spring damping (N/R/S)   b1,b2,b3        =   0.00000E+00   0.00000E+00   0.00000E+00
629.000  HINGE   #  2 Initial rotation angle (DEG) theta1,theta2,theta3 =   0.00000E+00   0.00000E+00   0.00000E+00
630.000  HINGE   #  2 Null torque angle (DEG)      theta1,theta2,theta3 =   0.00000E+00   0.00000E+00   0.00000E+00
631.000  HINGE   #  2 Translation degrees of freedom                    =   0.00000E+00
632.000  HINGE   #  2 Translation axis unit vector of 1ST DOF  X,Y,Z    =
633.000  HINGE   #  2 Translation axis unit vector of 2ND DOF  X,Y,Z    =
634.000  HINGE   #  2 Translation axis unit vector of 3RD DOF  X,Y,Z    =
635.000  HINGE   #  2 Translation spring stiffnes (N/M)  k1,k2,k3       =
636.000  HINGE   #  2 Translation spring damping (N/M/S) b1,b2,b3       =
637.000  HINGE   #  2 Initial translation displacement (M)  y1,y2,y3    =
638.000  HINGE   #  2 Null force position (M)               y1,y2,y3    =
639.000  HINGE   #  2 >>>>>***** END OF DATA *****<<<<<                 =
640.000  HINGE   #  3 ID# of hinge, ID# of inbd body, ID# of outbd body =   3.0000        2.0000        3.0000
641.000  HINGE   #  3 Node index of inboard body attach point           =   2.0000
642.000  HINGE   #  3 Rotation degrees of freedom                       =   0.00000E+00
643.000  HINGE   #  3 Base body(ID# 1)rotation option(F=free,G=gimbaled)=
644.000  HINGE   #  3 Rotation axis unit vector L1 inboard body  X,Y,Z  =   1.0000        0.00000E+00   0.00000E+00
645.000  HINGE   #  3 Rotation axis unit vector L1 outboard body X,Y,Z  =   1.0000        0.00000E+00   0.00000E+00
646.000  HINGE   #  3 Rotation axis unit vector L3 inboard body  X,Y,Z  =   0.00000E+00   0.00000E+00   1.0000
647.000  HINGE   #  3 Rotation axis unit vector L3 outboard body X,Y,Z  =   0.00000E+00   0.00000E+00   1.0000
648.000  HINGE   #  3 Rotation spring stiffness (N/R)   k1,k2,k3        =
649.000  HINGE   #  3 Rotation spring damping (N/R/S)   b1,b2,b3        =
650.000  HINGE   #  3 Initial rotation angle (DEG) theta1,theta2,theta3 =   0.00000E+00   0.00000E+00   0.00000E+00
651.000  HINGE   #  3 Null torque angle (DEG)      theta1,theta2,theta3 =
652.000  HINGE   #  3 Translation degrees of freedom                    =   0.00000E+00
653.000  HINGE   #  3 Translation axis unit vector of 1ST DOF  X,Y,Z    =
654.000  HINGE   #  3 Translation axis unit vector of 2ND DOF  X,Y,Z    =
655.000  HINGE   #  3 Translation axis unit vector of 3RD DOF  X,Y,Z    =
656.000  HINGE   #  3 Translation spring stiffnes (N/M)  k1,k2,k3       =
657.000  HINGE   #  3 Translation spring damping (N/M/S) b1,b2,b3       =
658.000  HINGE   #  3 Initial translation displacement (M)  y1,y2,y3    =
659.000  HINGE   #  3 Null force position (M)               y1,y2,y3    =
660.000  HINGE   #  3 >>>>>***** END OF DATA *****<<<<<                 =
661.000  BODY    #  1 Type of body (R=rigid, F=flexible),ID#            =  R              1.0000
662.000  BODY    #  1 Mass of body (kg)                                 =   95395.
663.000  BODY    #  1 Inertia (kg-m2) Ixx,Iyy,Izz                       =   0.12610E+07   0.96910E+07   0.10150E+08
664.000  BODY    #  1 Inertia (kg-m2) Ixy,Ixz,Iyz                       =  -1337.0       -0.31250E+06   676.60
665.000  BODY    #  1 modal data option (T=tape,D=disc),fname and tape# =
666.000  BODY    #  1 Modal inertia opt (N=none,I=inertia M,N,P dyadics)=
667.000  BODY    #  1 Modal coupling opt (N=none,C=coupling PHIXPHI vec)=
668.000  BODY    #  1 Number of flexible modes                          =   0.00000E+00
669.000  BODY    #  1 Attach point coordinates (m) x,y,z                =   27.736       -0.25000E-02   9.4690
670.000  BODY    #  1 NODE    1 mass center coordinates (m) x,y,z       =   27.736       -0.25000E-02   9.4690
671.000  BODY    #  1 NODE    2 coordinates (m) x,y,z                   =   25.527        0.00000E+00   10.566
```

672.000	BODY	#	1	NODE 3 coordinates (m) x,y,z	=	25.527	0.00000E+00	10.260
673.000	BODY	#	1	NODE 4 coordinates (m) x,y,z	=	8.2400	1.5200	8.8900
674.000	BODY	#	1	NODE 5 coordinates (m) x,y,z	=	8.2400	-1.5200	8.8900
675.000	BODY	#	1	NODE 6 coordinates (m) x,y,z	=	39.750	3.8100	11.660
676.000	BODY	#	1	NODE 7 coordinates (m) x,y,z	=	39.750	-3.8100	11.660
677.000	BODY	#	1	NODE 8 coordinates (m) x,y,z	=	39.750	3.0000	11.570
678.000	BODY	#	1	NODE 9 coordinates (m) x,y,z	=	39.750	-3.0000	11.570
679.000	BODY	#	1	>>>>>***** END OF DATA *****<<<<<	=			
680.000	BODY	#	1	>>>>>***** END OF DATA *****<<<<<	=			
681.000	BODY	#	2	Type of body (R=rigid, F=flexible),ID#	=	R	2.0000	
682.000	BODY	#	2	Mass of body (kg)	=	1875.0		
683.000	BODY	#	2	Inertia (kg-m2) Ixx,Iyy,Izz	=	12691.	12354.	1903.0
684.000	BODY	#	2	Inertia (kg-m2) Ixy,Ixz,Iyz	=	0.00000E+00	0.00000E+00	0.00000E+00
685.000	BODY	#	2	modal data option (T=tape,D=disc),fname and tape#	=			
686.000	BODY	#	2	Modal inertia opt (N=none,I=inertia M,N,P dyadics)	=			
687.000	BODY	#	2	Modal coupling opt (N=none,C=coupling PHIXPHI vec)	=			
688.000	BODY	#	2	Number of flexible modes	=	0.00000E+00		
689.000	BODY	#	2	Attach point coordinates (m) x,y,z	=	0.00000E+00	0.00000E+00	0.00000E+00
690.000	BODY	#	2	NODE 1 mass center coordinates (m) x,y,z	=	0.00000E+00	0.00000E+00	0.57610
691.000	BODY	#	2	NODE 2 coordinates (m) x,y,z	=	0.00000E+00	0.00000E+00	4.3198
692.000	BODY	#	2	NODE 3 coordinates (m) x,y,z	=	0.00000E+00	0.00000E+00	1.0000
693.000	BODY	#	2	>>>>>***** END OF DATA *****<<<<<	=			
694.000	BODY	#	2	>>>>>***** END OF DATA *****<<<<<	=			
695.000	BODY	#	3	Type of body (R=rigid, F=flexible),ID#	=	F	3.0000	
696.000	BODY	#	3	Mass of body (kg)	=	69.870		
697.000	BODY	#	3	Inertia (kg-m2) Ixx,Iyy,Izz	=	65391.	65416.	81.000
698.000	BODY	#	3	Inertia (kg-m2) Ixy,Ixz,Iyz	=	0.00000E+00	0.00000E+00	0.00000E+00
699.000	BODY	#	3	modal data option (T=tape,D=disc),fname and tape#	=	D	PINHOLE.FLX	
700.000	BODY	#	3	Modal inertia opt (N=none,I=inertia M,N,P dyadics)	=	N		
701.000	BODY	#	3	Modal coupling opt (N=none,C=coupling PHIXPHI vec)	=	N		
702.000	BODY	#	3	Number of flexible modes	=	8.0000		
703.000	BODY	#	3	Attach point coordinates (m) x,y,z	=	0.00000E+00	0.00000E+00	0.00000E+00
704.000	BODY	#	3	NODE 1 mass center coordinates (m) x,y,z	=	0.00000E+00	0.00000E+00	29.920
705.000	BODY	#	3	NODE 2 coordinates (m) x,y,z	=	0.00000E+00	0.00000E+00	32.000
706.000	BODY	#	3	>>>>>***** END OF DATA *****<<<<<	=			
707.000	BODY	#	3	>>>>>***** END OF DATA *****<<<<<	=			

```
      SUBROUTINE UCONTROL(TIME,U,R)
      DIMENSION DESATT(3),U(1),R(1),JETON(6),Y(3),X1(10),X2(10)
     *,XC(10),XVAR(3),XAVE(3),XSTD(3),IPP(3)
      DATA IFIRST /0/
C
      IF(IFIRST.EQ.1)GO TO 100
C     INITIALIZE THE VERNIER RCS CONTROLLER
C  AND THE FULL STATE GIMBAL CONTROLLER.
        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=.04
        CALL FSCONT(INIT,DT,X1,XC,Y,TEMP)
C       CALL FSCONT(INIT,DT,X2,XC,Y,TEMP)
      I125=2
      INIT=1
      IFIRST=1
  100 CONTINUE
C  ********** ROLL GIMBAL CONTROL LAW **************************
        CALL SENQNT(U(14),U(15),U(16),U(5),U(4),U(3))
      Y(1)=U(3)
      Y(2)=-U(4)
      Y(3)=U(5)
      CALL FSCONT(INIT,DT,X1,XC,Y,TEMP)
      R(1)=TEMP-ACGAIN*U(2)
      R(2)=206264.0*U(5)
        R(31)=(1.+.01*U(54))*U(48)            !INTRODUCE SCALE FACTOR ERRORS
        R(32)=(1.+.01*U(55))*U(49)
        R(33)=(1.+.000478*U(56))*U(50)
        CALL XTORQ(U,X1,R)
C
C  ********** PITCH GIMBAL CONTROL LAW *************************
        CALL SENQNT(U(17),U(18),U(19),U(10),U(9),U(8))
      Y(1)=U(8)
      Y(2)=U(9)
      Y(3)=U(10)
      CALL FSCONT(INIT,DT,X2,XC,Y,TEMP)
      R(3)=TEMP+ACGAIN*U(7)
      R(4)=206264.0*U(10)
        R(34)=(1.+.01*U(57))*U(51)            !INTRODUCE SCALE FACTOR ERRORS
        R(35)=(1.+.01*U(58))*U(52)
        R(36)=(1.+.000478*U(59))*U(53)
        CALL YTORQ(U,X2,R)
C
C  ********** ORBITER VERNIER RCS CONTROL LAW *****************
      I125=I125+1
      IF(I125.LT.2)GO TO 101
      I125=0
      DESATT(1)=-U(11)
      DESATT(2)= U(12)
      DESATT(3)=-U(13)
      CALL FCS(INIT,DESATT,JETON)
      R(5)=JETON(1)*109.
      R(6)=JETON(2)*109.
      R(7)=JETON(3)*106.8
      R(8)=JETON(4)*106.8
```

```
      R(9)=JETON(5)*64.68
      R(10)=JETON(6)*64.68
  101 CONTINUE
C
C       CALL POST PROCESSOR
C
        R(29)=U(46)
        R(30)=U(47)
        CALL POSTP(TIME,1,IPP(1),U(45),R(11),R(12))      !DRIVER STATISTICS
        CALL POSTP(TIME,2,IPP(2),R(29),R(13),R(14))      !ROLL LOS ERROR STATS
        CALL POSTP(TIME,3,IPP(3),R(30),R(15),R(16))      !PITCH LOS ERROR STATS
      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,DT,X,XC,Y,R)
        DIMENSION X(6),XC(6),Y(3),FX(6),G1Y(6),G2XC(6)
        DIMENSION F(6,6),G1(6,3),G2(6,6),H(6),E2(6)
        IF(INIT.GT.0)GO TO 100
C
C       ZERO STATE VECTOR X AND INPUT VECTOR XC
C
        DO 1 I=1,6
        X(I)=0.0
        XC(I)=0.0
1       CONTINUE
C
C       F MATRIX ELEMENTS
```

```
C
      F(1,1)=-8.1617E-01
      F(1,2)=-2.6777E-02
      F(1,3)=-3.1898E-04
      F(1,4)=-3.4064E-01
      F(1,5)=1.6334E-01
      F(1,6)=-2.5544E-03
      F(2,1)=6.6044E+01
      F(2,2)=1.9686E+00
      F(2,3)=1.1961E-02
      F(2,4)=1.1834E+01
      F(2,5)=-6.4053E+00
      F(2,6)=1.0555E-01
      F(3,1)=1.5244E+01
      F(3,2)=2.3370E-01
      F(3,3)=9.7291E-01
      F(3,4)=2.7773E+00
      F(3,5)=-1.1760E+00
      F(3,6)=-1.3462E+00
      F(4;1)=5.9056E-03
      F(4,2)=2.0417E-04
      F(4,3)=-7.7905E-06
      F(4,4)=9.6718E-01
      F(4,5)=1.0468E-02
      F(4,6)=-2.9981E-04
      F(5,1)=1.2091E+00
      F(5,2)=4.7770E-02
      F(5,3)=7.5288E-04
      F(5,4)=2.8363E-01
      F(5,5)=1.0475E-01
      F(5,6)=2.4144E-02
      F(6,1)=2.4434E-01
      F(6,2)=-3.1204E-03
      F(6,3)=3.9704E-02
      F(6,4)=3.8774E-03
      F(6,5)=-1.5652E-01
      F(6,6)=9.7712E-01
C
C     G1 MATRIX ELEMENTS
C
      G1(1,1)=5.6002E-01
      G1(1,2)=-1.6941E-01
      G1(1,3)=-1.2162E-02
      G1(2,1)=-2.0118E+01
      G1(2,2)=6.6046E+00
      G1(2,3)=1.0684E+00
      G1(3,1)=-4.4982E+00
      G1(3,2)=1.2237E+00
      G1(3,3)=2.4193E-01
      G1(4,1)=3.1338E-02
      G1(4,2)=-1.1486E-02
      G1(4,3)=3.2229E-02
      G1(5,1)=-1.1244E-01
      G1(5,2)=9.0007E-01
      G1(5,3)=2.5235E-02
      G1(6,1)=-2.1472E-01
      G1(6,2)=1.5654E-01
      G1(6,3)=4.4559E-03
C
C     G2 MATRIX ELEMENTS
C
```

```
      G2(1,1)=1.2474E+00
      G2(1,2)=3.2301E-02
      G2(1,3)=2.1575E-04
      G2(1,4)=3.5280E-01
      G2(1,5)=6.7913E-03
      G2(1,6)=3.4448E-04
      G2(2,1)=-4.5616E+01
      G2(2,2)=-1.1812E+00
      G2(2,3)=-7.8898E-03
      G2(2,4)=-1.2903E+01
      G2(2,5)=-2.4838E-01
      G2(2,6)=-1.2597E-02
      G2(3,1)=-1.0674E+01
      G2(3,2)=-2.7641E-01
      G2(3,3)=-1.8463E-03
      G2(3,4)=-3.0194E+00
      G2(3,5)=-5.8118E-02
      G2(3,6)=-2.9480E-03
      G2(4,1)=2.0926E-03
      G2(4,2)=5.4196E-05
      G2(4,3)=3.6194E-07
      G2(4,4)=5.9182E-04
      G2(4,5)=1.1392E-05
      G2(4,6)=5.7794E-07
      G2(5,1)=-1.0920E+00
      G2(5,2)=-2.8277E-02
      G2(5,3)=-1.8887E-04
      G2(5,4)=-3.0886E-01
      G2(5,5)=-5.9455E-03
      G2(5,6)=-3.0157E-04
      G2(6,1)=-2.9460E-02
      G2(6,2)=-7.6294E-04
      G2(6,3)=-5.0968E-06
      G2(6,4)=-8.3372E-03
      G2(6,5)=-1.6041E-04
      G2(6,6)=-8.1382E-06
C
C     H MATRIX ELEMENTS
C
      H(1)=-7.2887E+05
      H(2)=-1.8875E+04
      H(3)=-1.2607E+02
      H(4)=-2.0615E+05
      H(5)=-3.9682E+03
      H(6)=-2.0129E+02
C
C     E2 MATRIX ELEMENTS
C
      E2(1)=7.2887E+05
      E2(2)=1.8875E+04
      E2(3)=1.2607E+02.
      E2(4)=2.0615E+05
      E2(5)=3.9682E+03
      E2(6)=2.0129E+02
100   CONTINUE
C
C     UPDATE STATE EQUATION X(K+1)=F*X(K)+G1*Y(K)+G2*XC(K)
C
      CALL MXM(F,X,FX,6,6,1,6)               !F*X(K)
      CALL MXM(G1,Y,G1Y,6,3,1,3)             !G1*Y(K)
      CALL MXM(G2,XC,G2XC,6,6,1,6)           !G2*XC(K)
```

```
      DO 10 I=1,6
10    X(I)=FX(I)+G1Y(I)+G2XC(I)
C
C     COMPUTE CONTROLLER OUTPUT R(K)=H*X(K)+E2*XC(K)
C
      CALL MXM(H,X,HX,1,6,1,6)                  !H*X(K)
      CALL MXM(E2,XC,E2XC,1,6,1,6)              !E2*XC(K)
      R=HX+E2XC
      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 SENQNT(DQX,DQY,DQZ,QX,QY,QZ)
C
C     THIS SUBROUTINE CALCULATES THE OUTPUT OF THE LOS,
C     LASER INTERFEROMETER AND RATE GYRO QUANTIZAERS.
C
C     INPUTS:1)DQX-MAX QUANTIZING ERROR-LOS SENSOR
C            2)DQY-MAX QUANTIZING ERROR-LASER INTERFEROMETER
C            3)DQZ-MAX QUANTIZING ERROR-RATE GYRO
C            4)QX-QUANTIZED LOS SENSOR OUTPUT
C            5)QY-QUANTIZED LASER INTERFEROMETER OUTPUT
C            6)QZ-QUANTIZED RATE GYRO OUTPUT
C
C     OUTPUTS:1)QX-QUANTIZED LOS SENSOR OUTPUT
C             2)QY-QUANTIZED LASER INTERFEROMETER OUTPUT
C             3)QZ-QUANTIZED RATE GYRO OUTPUT
C
      IF(DQX.NE.0.)QX=2.*DQX*NINT(QX/(2.*DQX))
      IF(DQY.NE.0.)QY=2.*DQY*NINT(QY/(2.*DQY))
      IF(DQZ.NE.0.)QZ=2.*DQZ*NINT(QZ/(2.*DQZ))
      RETURN
      END
      SUBROUTINE XTORQ(U,X1,R)
```

```
C
C     THIS SUBROUTINE CALCULATES TORQUE MOTOR DISTURBANCES AND
C     A QUANTIZED TORQUE COMMAND.
C
C     INPUTS: 1)CCOG-COGGING GAIN
C             2)CRIP-RIPPLE GAIN
C             3)TDCU-DCU TORQUE OUTPUT
C             4)DTC-MAX TORQUE MOTOR QUANTIZATION ERROR
C             5)TCQ-ANALOG TORQUE COMMAND
C             6)TF1-FRICTION TORQUE(DAHL MODEL 1)
C             7)TO1-SATURATION TORQUE(DAHL MODEL 1)
C             8)GAMMA1-SLOPE PARAMETER(DAHL MODEL 1)
C             9)TF2-FRICTION TORQUE(DAHL MODEL 2)
C             10)TO2-SATURATION TORQUE(DAHL MODEL 2)
C             11)GAMMA2-SLOPE PARAMETER(DAHL MODEL 2)
C             12)DTHE22-SHAFT RATE
C             13)TR-ROUGHNESS TORQUE
C             14)STR-ROUGHNESS TORQUE WHITE NOISE
C
C     OUTPUTS:1)TCQ-QUANTIZED TORQUE COMMAND
C             2)DTF1-FRICTION TORQUE RATE(DAHL MODEL 1)
C             3)DTF2-FRICTION TORQUE RATE(DAHL MODEL 2)
C             4)DTR-ROUGHNESS TORQUE RATE
C             5)TCOG-COGGING TORQUE
C             6)TRIP-RIPPLE TORQUE
C
      DIMENSION U(1),X1(1),R(1)
      CCOG=U(41)
      CRIP=U(42)
      TDCU=U(26)
      DTC=U(21)
      TCQ=R(1)
      TF1=U(27)
      TO1=U(22)
      GAMMA1=U(24)
      TF2=U(28)
      TO2=U(23)
      GAMMA2=U(25)
      DTHE22=X1(1)
      TR=U(29)
      STR=U(20)
      THE22=X1(4)
C
C     QUANTIZED TORQUE COMMAND
C
      IF(DTC.NE.0.)TCQ=2.*DTC*NINT(TCQ/(2.*DTC))
C
C     DAHL MODELS FRICTION RATE DISTURBANCE
C
      DTF1=0.
     IF(ABS(TF1).LE.TO1)DTF1=DTHE22*GAMMA1*(TF1*SIGN(1.,DTHE22)-TO1)**2
      DTF2=0.
     IF(ABS(TF2).LE.TO2)DTF2=DTHE22*GAMMA2*(TF2*SIGN(1.,DTHE22)-TO2)**2
C
C     ROUGHNESS TORQUE RATE DISTURBANCE
C
      DTR=(STR-TR)*DTHE22
C
C     COGGING TORQUE DISTURBANCE
C
      TCOG=2.*CCOG*SIN(96.*THE22)
```

```
C
C     RIPPLE TORQUE DISTURBANCE
C
      TRIP=TDCU*CRIP*SIN(48.*THE22)
      R(1)=TCQ
      R(17)=DTF1
      R(18)=DTF2
      R(19)=DTR
      R(20)=TCOG
      R(21)=TRIP
      RETURN
      END
      SUBROUTINE YTORQ(U,X2,R)
C
C     THIS SUBROUTINE CALCULATES TORQUE MOTOR DISTURBANCES AND
C     A QUANTIZED TORQUE COMMAND.
C
C     INPUTS: 1)CCOG-COGGING GAIN
C             2)CRIP-RIPPLE GAIN
C             3)TDCU-DCU TORQUE OUTPUT
C             4)DTC-MAX TORQUE MOTOR QUANTIZATION ERROR
C             5)TCQ-ANALOG TORQUE COMMAND
C             6)TF1-FRICTION TORQUE(DAHL MODEL 1)
C             7)TO1-SATURATION TORQUE(DAHL MODEL 1)
C             8)GAMMA1-SLOPE PARAMETER(DAHL MODEL 1)
C             9)TF2-FRICTION TORQUE(DAHL MODEL 2)
C             10)TO2-SATURATION TORQUE(DAHL MODEL 2)
C             11)GAMMA2-SLOPE PARAMETER(DAHL MODEL 2)
C             12)DTHE22-SHAFT RATE
C             13)TR-ROUGHNESS TORQUE
C             14)STR-ROUGHNESS TORQUE WHITE NOISE
C
C     OUTPUTS:1)TCQ-QUANTIZED TORQUE COMMAND
C             2)DTF1-FRICTION TORQUE RATE(DAHL MODEL 1)
C             3)DTF2-FRICTION TORQUE RATE(DAHL MODEL 2)
C             4)DTR-ROUGHNESS TORQUE RATE
C             5)TCOG-COGGING TORQUE
C             6)TRIP-RIPPLE TORQUE
C
      DIMENSION U(1),X2(1),R(1)
      CCOG=U(43)
      CRIP=U(44)
      TDCU=U(36)
      DTC=U(31)
      TCQ=R(3)
      TF1=U(37)
      TO1=U(32)
      GAMMA1=U(34)
      TF2=U(38)
      TO2=U(33)
      GAMMA2=U(35)
      DTHE22=X2(1)
      TR=U(39)
      STR=U(30)
      THE22=X2(4)
C
C     QUANTIZED TORQUE COMMAND
C
      IF(DTC.NE.0.)TCQ=2.*DTC*NINT(TCQ/(2.*DTC))
C
C     DAHL MODELS FRICTION RATE DISTURBANCE
```

```
C
      DTF1=0.
   IF(ABS(TF1).LE.T01)DTF1=DTHE22*GAMMA1*(TF1*SIGN(1.,DTHE22)-T01)**2
      DTF2=0.
   IF(ABS(TF2).LE.T02)DTF2=DTHE22*GAMMA2*(TF2*SIGN(1.,DTHE22)-T02)**2
C
C     ROUGHNESS TORQUE RATE DISTURBANCE
C
      DTR=(STR-TR)*DTHE22
C
C     COGGING TORQUE DISTURBANCE
C
      TCOG=2.*CCOG*SIN(96.*THE22)
C
C     RIPPLE TORQUE DISTURBANCE
C
      TRIP=TDCU*CRIP*SIN(48.*THE22)
      R(3)=TCQ
      R(23)=DTF1
      R(24)=DTF2
      R(25)=DTR
      R(26)=TCOG
      R(27)=TRIP
      RETURN
      END
```

```
      SUBROUTINE DIST(TIME,FDIST)
      INCLUDE 'DBP.FOR'
      DIMENSION FDIST(PNUFG),SUM(PNUFG)
      INTEGER*4 ISEED(PNUFG),ISECS,ITSEED
      NN=8
      IF(TIME.NE.0.)GO TO 10
      NN=20
      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
```

7.0 VAX/TREETOPS USER NOTES

- To modify the interactive file JRRPH2.INT, type

  RUN HITIPG CR

  and follow the procedures outlined in the TREETOPS user manual

- To run the TREETOPS program, submit a batch job by typing

  SUBMIT JRRPH2.JCL CR

- To post process the output data first type

  @PPLOT CR

  then type

  RUN PPLOT CR

  Answer the questions as outlined in the TREETOPS user manual.

CR → carraige return