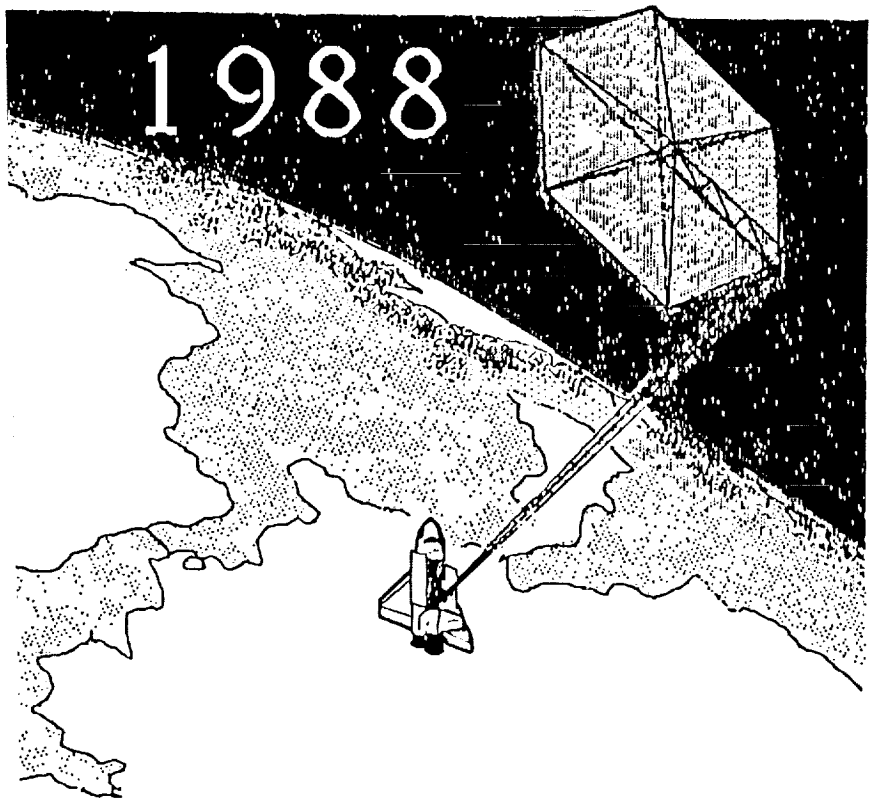
NASA Conference Publication 10057
Part 2

5th Annual NASA Spacecraft Control Laboratory Experiment (SCOLE) Workshop



Compiled by
Lawrence W. Taylor, Jr.
Langley Research Center
Hampton, Virginia

Proceedings of a workshop sponsored by the National Aeronautics and Space Administration and held at the Hilton Lodge Lake Arrowhead, California October 31, 1988

DECEMBER 1990

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

Introduction

SCOLE stands for the "Spacecraft Control Laboratory Experiment". The objective of the SCOLE Program is to provide an example configuration and control objectives which enables direct comparison of different techniques in modeling, systems identification and control. The "SCOLE Design Challenge" was formulated in 1983 by L. W. Taylor and A. V. Balakrishnan. The details of this challenge are reprinted at the end of this document.

Annual SCOLE Workshops have been held for specialists to share and compare their research results. This proceedings is a compilation of the material presented at the 5th Workshop held at Hilton Lodge at Lake Arrowhead, California on October 31, 1988.

Table of Contents - Part I*

*Published under separate cover.

Table of Contents, Continued

Table of Contents - Part II

Table of Contents, Continued

Quasi-Time Optimal Controllers Designed for the SCOLE Laboratory Model

by

Howard Kaufman

Rensselaer Polytechnic Institute

OUTLINE

INTRODUCTION

- Process model
- Objectives

PROCEDURES

- Model Following
- Time Optimal

Results

Present Activities

SPACECRAFT CONTROL LAB

EXPERIMENT (SCOLE)

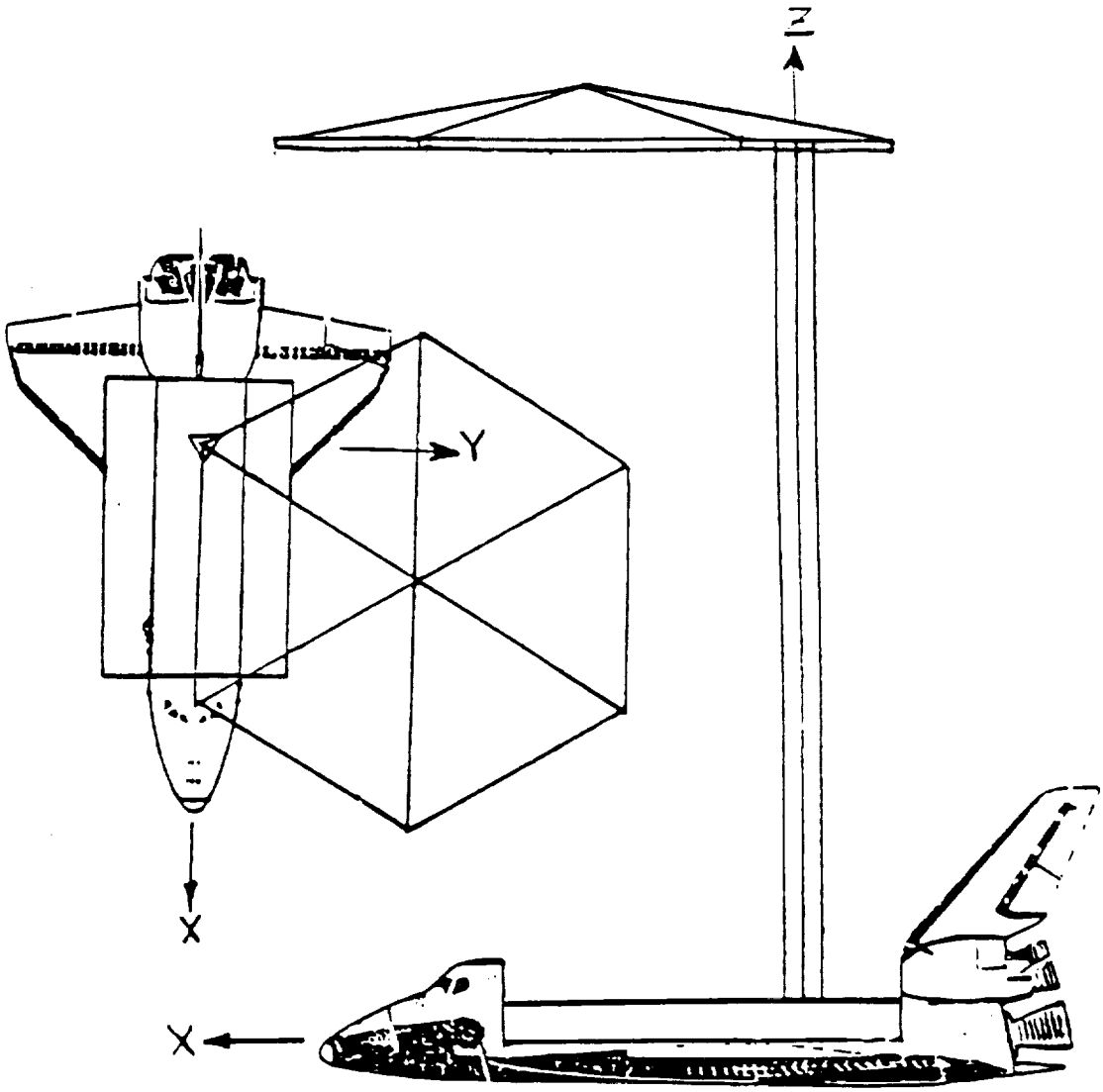


Figure A.1 Drawing of SCOLE Shuttle/Antenna Configuration

INTRODUCTION

Process Model: 6 mode flexible model
3 rate sensors
3 control torquers

Open Loop settling time ≈ 1400 - 1500 sec.

INTRODUCTION

Objective: Initial X-Y displacement = 1 foot
Min Time transfer to (0,0)
Control magnitude less than 100 oz-in.

Linear System Model

$$\dot{X}_p = A_p X_p + B_p U_p$$

$$Y_o = C_p X_p = (X \text{ \& } Y \text{ coordinates})$$

$$Y_s = H_p X_p =$$

$$(O_1 \quad O_2 \quad O_3)^T = (\text{rate sensor outputs})$$

$Mode \quad number$	$Damping \quad ratio$	$Frequency(Hz)$
1	0.001	0.442693
2	0.001	0.446678
3	0.001	1.504851
4	0.001	2.913638
5	0.001	4.345258
6	0.00158	6.821094
7	0.001250	10.182251
8	0.000603	13.575831
9	0.001340	14.744912
10	0.000841	22.255601

Table 1: PLANT MATRIX DATA

PROCEDURE

MODEL FOLLOWING:

Principle: Find fast reference model such that control constraints are satisfied.

Options: Model following of

- Sensor outputs
- Coordinate outputs

OUTPUT MODEL FOLLOWING

(CGT CONCEPT)

PROCESS

$$\dot{x}_p(t) = A_p x_p(t) + B_p u_p(t)$$

$$y_p(t) = C_p x_p(t)$$

REFERENCE MODEL

$$\dot{x}_m(t) = A_m x_m(t) + B_m u_m(t)$$

$$y_m(t) = C_m x_m(t)$$

MODEL FOLLOWING BEHAVIOR

$$Y_p(t_1) = Y_m(t_1) \Longrightarrow \quad Y_p(t) = Y_m(t)$$

$$t \geq t_1$$

$$Y_p(t_1) \neq Y_m(t_1) \Longrightarrow \quad Y_p(t) \rightarrow Y_m(t)$$

DEFINE IDEAL STATE AND CONTROL x_p^*, u_p^* SUCH THAT

$$C_p x_p^*(t) = C_m x_m(t)$$

AND

$$\dot{x}_p^*(t) = A_p x_p^*(t) + B_p u_p^*(t)$$

THEN ACTUAL CONTROL IS

$$u_p = u_p^* + K(Y_m - Y_p)$$

$$= u_p^* + K\, C_p(x_p^* - x_p)$$

$$\frac{d}{dt}(x_p^* - x_p) = \dot{e} = (A_p - B_p K_p C_p)e$$

THEREFORE K STABILIZES $(A_p - B_p K_p C_p)$

ASSUME THAT THE IDEAL PLANT RESPONSE SATISFIES:

$$\begin{bmatrix} x_p^*(t) \\ \\ u_p^*(t) \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} x_m \\ \\ u_m \end{bmatrix}$$

THEN

$$S_{11} = \Omega_{11}\, S_{11}\, A_m + \Omega_{12}\, C_m$$

$$S_{12} = \Omega_{11}\, S_{11}\, B_m$$

$$S_{21} = \Omega_{21}\, S_{11}\, A_m + \Omega_{22}\, C_m$$

$$S_{22} = \Omega_{21}\, S_{11}\, B_m$$

WITH

$$\Omega = \begin{bmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{bmatrix} = \begin{bmatrix} A_p & B_p \\ C_p & 0 \end{bmatrix}^{-1}$$

REQUIRE EIGENVALUES OF $\Omega_{11} \neq$ EIGENVALUES OF A_m^{-1}

SUMMARY OF CGT

$$u_p = S_{21}\, X_m + S_{22}\, u_m + K(Y_m - Y_p)$$

$$X_p^* - X_m \longrightarrow 0$$

$$Y_p \longrightarrow Y_m$$

X_p BOUNDED

ALTERNATIVE

If we wish to control Y_0 and we sense Y_s, stabilize plant first with feedback

$$U_p = K\ Y_s + U_f$$

Augmented plant is:

$$\dot{X}_p = (A_p + B_p\ K\ H_p)\ X_p + B_p\ U_f$$
$$= A + B_p\ U_p$$

Find U_f so that

$$Y_0 = C_p\ X_p \text{ follows } Y_m$$

MODEL DESIGNS

- Closed Loop Plant
- Low order model for each output

$$Y_{mi}(s)/U_{mi}(s) = G_m(s)$$

$$Yp_i \rightarrow Ym_i$$

- $U_m = 0$
- Gm first or second order

RESULTS

Model Following

- Feedback gain computed using EA
- Feedforward gains computed for
  - 1^{st} order model
  - 2^{nd} order model

BEST SOLUTION

DUAL MODEL REFERENCE CONTROL

First: Slow model, settling time ≈ 280 seconds

Second: Fast model, settling time ≈ 85 seconds

Switching time = 60 seconds

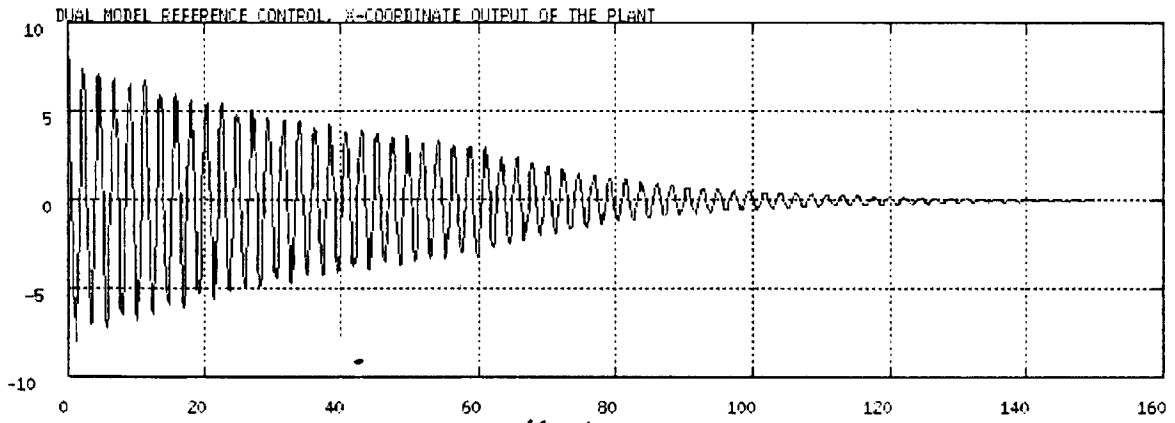

DUAL MODEL REFERENCE CONTROL, X-COORDINATE OUTPUT OF THE PLANT
10
5
0
-5
-10
0
20
40
60
80
100
120
140
160

fig. 1a

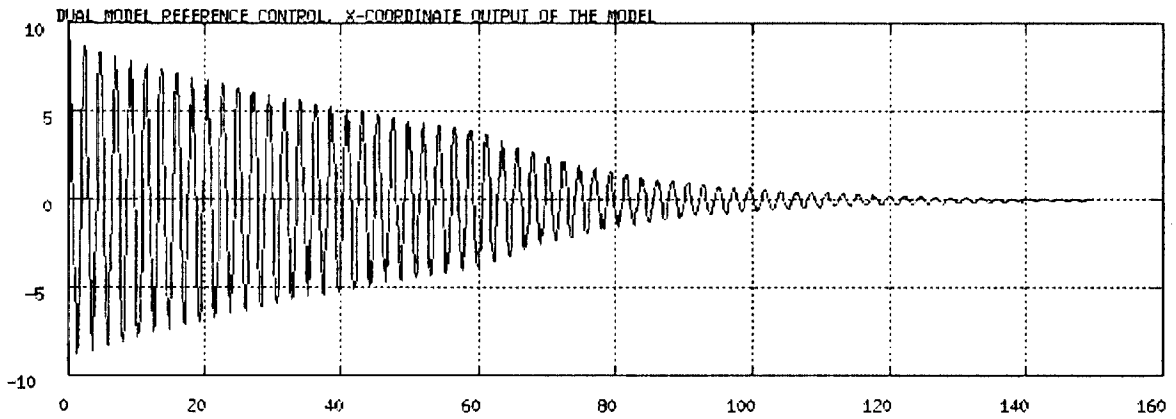

DUAL MODEL REFERENCE CONTROL, X-COORDINATE OUTPUT OF THE MODEL
10
5
0
-5
-10
0
20
40
60
80
100
120
140
160

fig. 1b

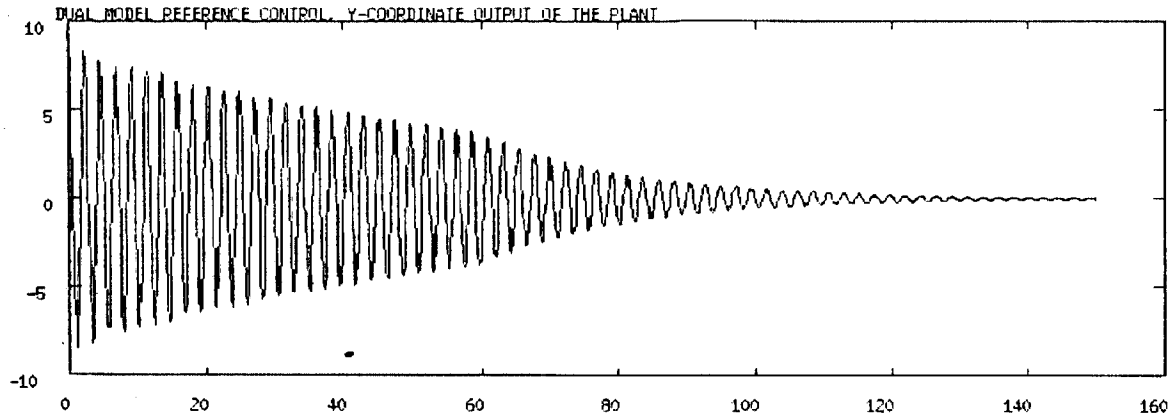


FIG. 2a

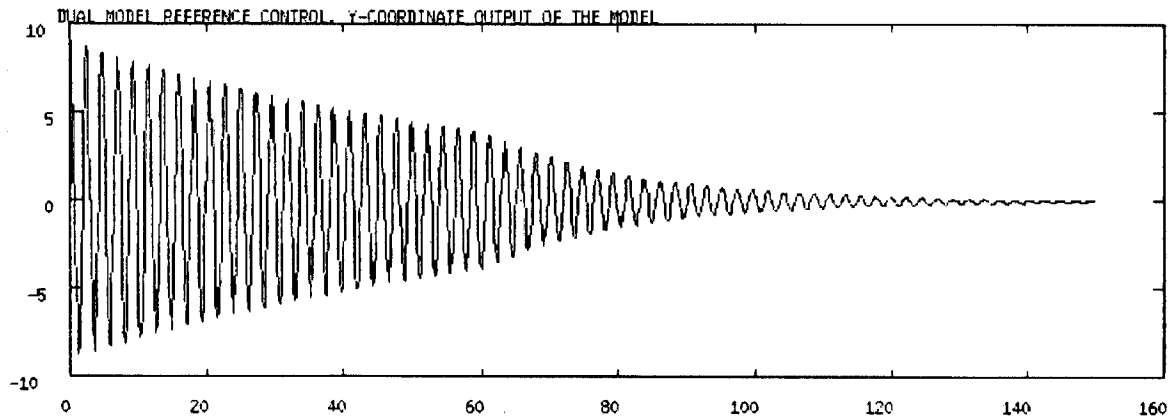


FIG. 2b

398

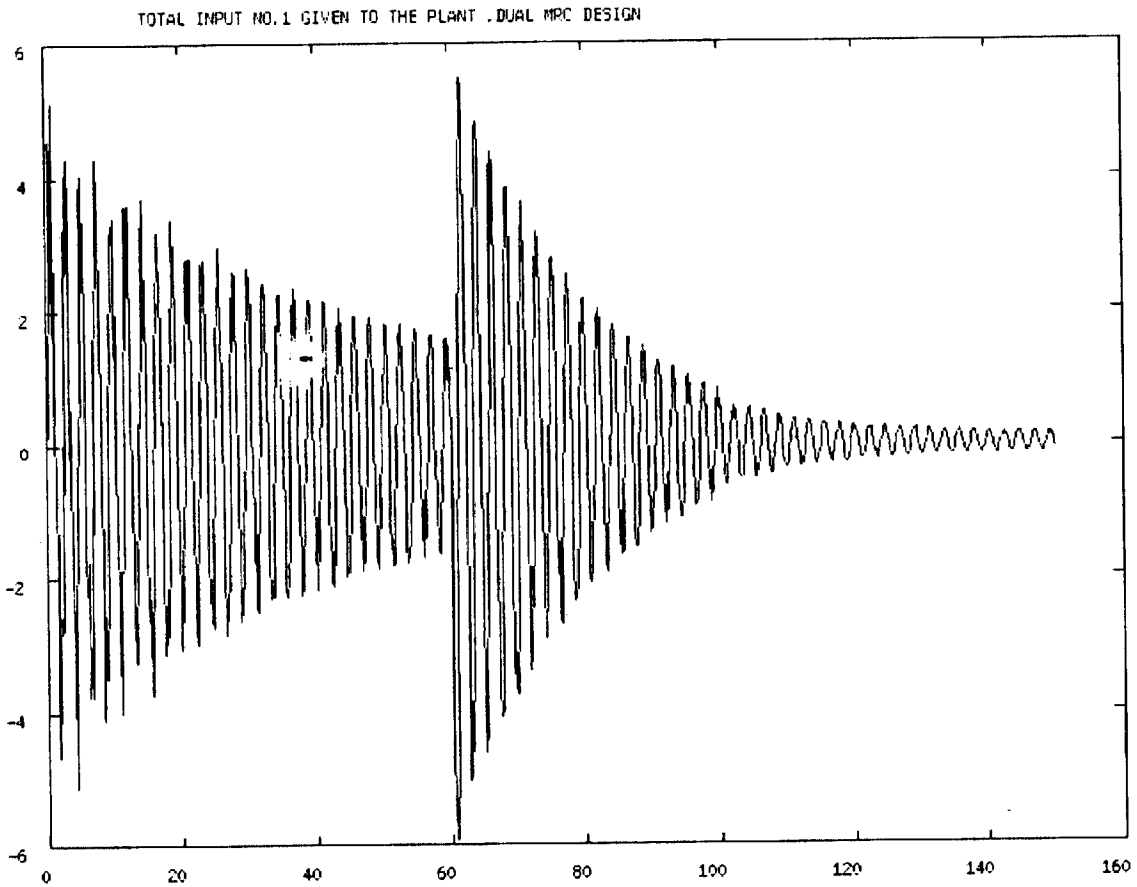

TOTAL INPUT NO.1 GIVEN TO THE PLANT .DUAL MPC DESIGN
6
4
2
0
-2
-4
-6
0
20
40
60
80
100
120
140
160

FIG. 3a

400

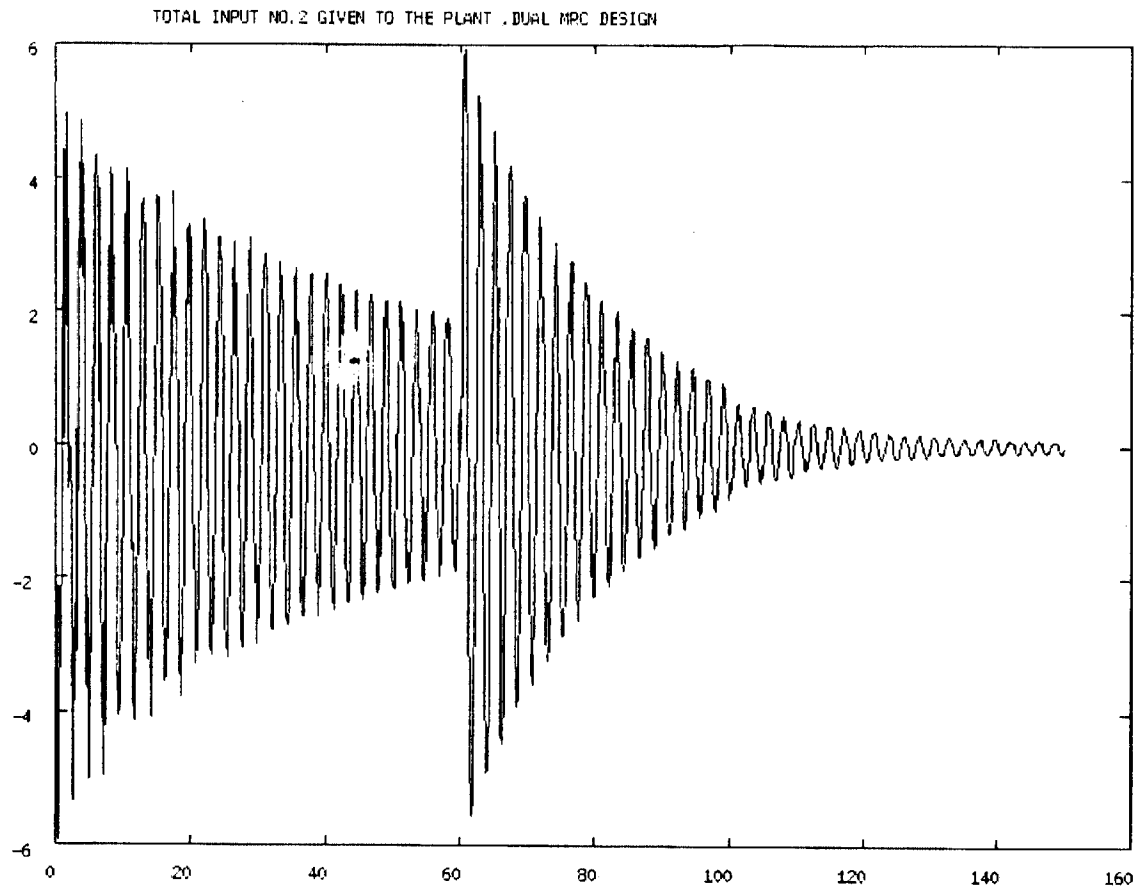
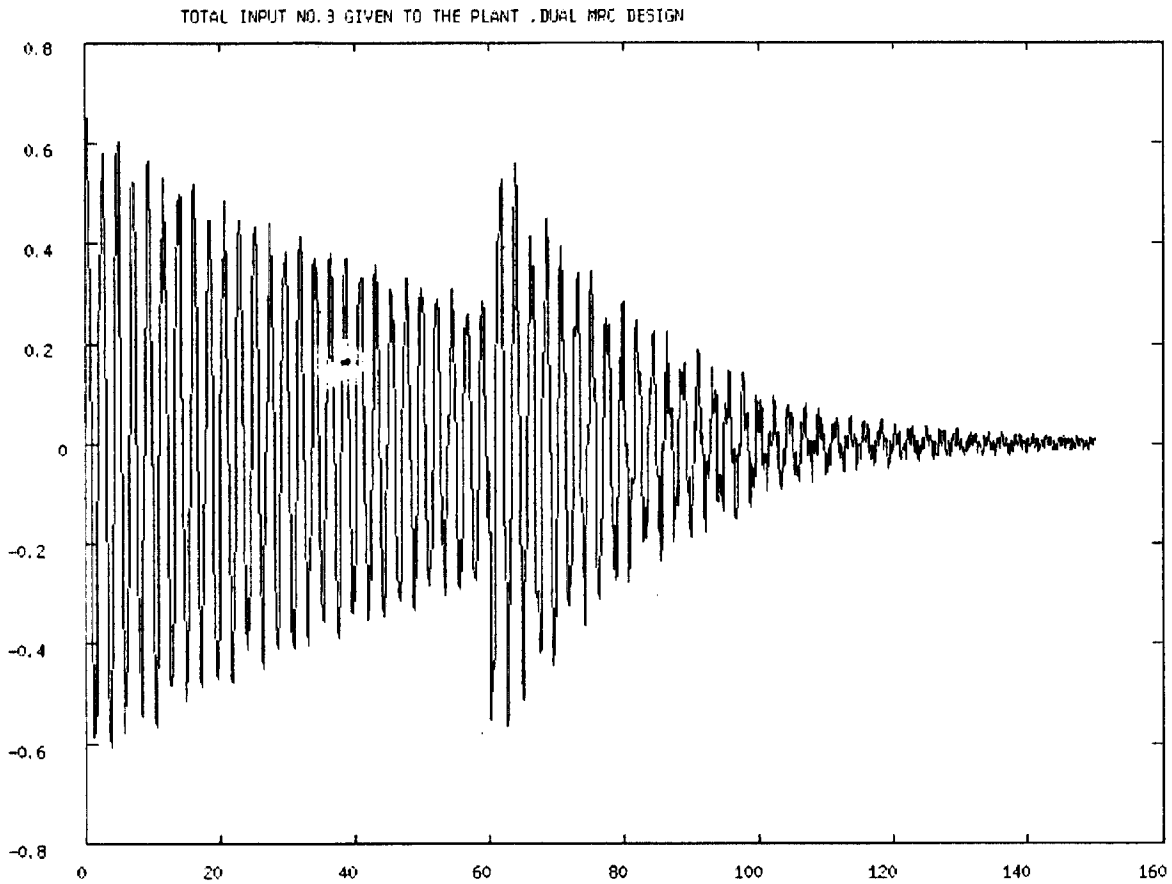

TOTAL INPUT NO.2 GIVEN TO THE PLANT .DUAL MPC DESIGN
6
4
2
0
-2
-4
-6
0
20
40
60
80
100
120
140
160

FIG. 3b


TOTAL INPUT NO. 3 GIVEN TO THE PLANT . DUAL MRC DESIGN
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
20
40
60
80
100
120
140
160

401

FIG. 3c

MIN TIME DESIGN PRINCIPLE

System:

$$\dot{X} = A\,X + Bu$$

$$X(T) = 1$$

$$\min J = T = \int_0^T dt$$

$$-M \leq U \leq +M$$

APPROACH

Define $Z(t) = T$

$Z = 0$

$\tau = t/T$

$$J = \int_0^1 Z(\tau)\, d\tau$$

$$\frac{dx}{d\tau} = A\ Z\ X + B\ Z\ U \qquad \begin{matrix} x(0) = x_0 \\ x(1) = 0 \end{matrix}$$

$$\frac{dZ}{d\tau} = 0$$

SOLUTION PROCEDURE

TPBVP

$$\dot{X} = A\,Z\,X - BZM\,\mathrm{sign}(B^T \lambda Z)$$

$$\dot{\lambda} = -A^T\,\lambda\,Z$$

$$\dot{Z} = 0$$

$$X(0) = X_0$$
$$X(1) = 0$$
$$1 + \lambda^T(1)\,Bu(1) = 0$$

Shoot from $X(0)$
$\lambda(0)$
$Z(0)$

PRESENT ACTIVITIES

- Sensor following
- Discrete Design
- Switching Logic
- Robustness

Model Reference Control of the Linearized SCOLE Model

Preliminary Report to NASA Langley Research Center

by

A. Musalem
H. Kaufman
Electrical, Computer, and Systems Engineering Department
Rensselaer Polytechnic Institute
Troy, New York 12180-3590

NASA Monitor: S. Joshi

ABSTRACT

A Perfect Model Following (PMF) controller for the linearized SCOLE project is described and evaluated. The plant includes both the flexible and rigid body modes comprising 8 modes and 16 states. The PMF controller is a special case of the output following or Command Generator Tracker concept (CGT) which is presented and discussed. Results obtained show that all state tracking causes some violation of the control moment constraints.

However a redesign for the tracking of only selected outputs should be feasible.

I. INTRODUCTION

The Spacecraft Control Laboratory Experiment (SCOLE) consists of a large antenna attached to the space shuttle orbiter by a flexible beam. The purpose of the experiment is to evaluate control laws for flexible spacecraft as proposed by Taylor and Balakrishman [1].

This report presents the results and the methodology used to evaluate a control law by the Perfect Model Following (PMF) method. PMF is a special case of the Command Generator Tracker concept (CGT) developed by Broussard [2]. In Section II of this report both concepts are outlined.

The SCOLE system used is a linearized model which consists of flexible and rigid body modes. The flexible model state variable vector consists of ten states representing 5 second order modes.

$$x^T = [q_1 \ \dot{q}_1 \ q_2 \ \dot{q}_2 \ \ldots\ldots q_5 \ \dot{q}_5] \tag{1}$$

The control is the 8 element vector

$$u^T = [T_{sx}, T_{sy}, T_{sz}, F_{rx}, F_{ry}, T_{rx}, T_{ry}, T_{rz}] \tag{2}$$

Where

T_{sx}, T_{sy}, T_{sz} = Moments applied at shuttle

T_{rx}, T_{ry}, T_{rz} = Moments applied at reflector

F_{rx}, F_{ry} = Forces applied at reflector

The output vector for the system is as follows:

$$y^T = [\phi_s, \theta_s, \psi_s, \dot{\phi}_s, \dot{\theta}_s, \dot{\psi}_s, \zeta_x, \zeta_y, \phi_r, \theta_r, \psi_r, \dot{\phi}_r, \dot{\theta}_r, \dot{\psi}_r] \tag{3}$$

where

ϕ_s, θ_s, ψ_s = Shuttle attitude.

$\dot{\phi}_s$, $\dot{\theta}_s$, $\dot{\psi}_s$ = Shuttle attitude rate.

ϕ_r, θ_r, ψ_r = Reflector attitude.

$\dot{\phi}_r$, $\dot{\phi}_r$, $\dot{\psi}_r$ = Reflector attitude rate

ζ_x, ζ_y = Deflection of tip with respect to the base.

The flexible system, then, is represented as follows:

$$\dot{x} = Ax + Bu \tag{4}$$

$$y = Cx \tag{5}$$

Where A is a (10 x 10) matrix, B is a (10 x 8) matrix, and C has dimensions (14 x 10). Matrices A, B and C are given in the appendix.

In order to define complete motion we have to add the rigid body equations. This rigid body model is represented as follows:

$$\begin{bmatrix} I_s \end{bmatrix} \begin{bmatrix} \ddot{\phi}_{RB} \\ \ddot{\theta}_{RB} \\ \ddot{\psi}_{RB} \end{bmatrix} = \underline{T}_s + \underline{T}_r + \begin{bmatrix} \theta & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \begin{bmatrix} F_{rx} \\ F_{ry} \\ 0 \end{bmatrix} \tag{6}$$

Where I_s is a (3 x 3) matrix representing the combined inertia motion given by

$$I_s \begin{bmatrix} 1.132508 \times 10^6 & 7555. & -115202. \\ 7555. & 7.007447 \times 10^6 & -52293. \\ -115202. & -52293 & >.113962 \times 10^6 \end{bmatrix}$$

Vector $\underline{T}_s$ and $\underline{T}_r$ and F_{rx}, F_{ry} are the controls from Eq. (2), and x, y, and z are 18.7136, -32.4363, and -129.621 respectively. The (3 x 3) matrix in the RHS of Eq. (6) corresponds to the moments due to the applied forces.

The total displacement y_{TD}, then, is expressed as the displacement of the flexible model plus the rigid body displacement. In equation form we have

$$y_{TD} = y + RBV \tag{6a}$$

where y is from Eq. (5) and RBV (Rigid Body Vector) is given by

$$(RBV)^T = [\phi_{RB}, \theta_{RB}, \psi_{RB}, \dot{\phi}_{RB}, \dot{\theta}_{RB}, \dot{\psi}_{RB}, 0, 0, \phi_{RB}, \theta_{RB}, \psi_{RB}, \dot{\phi}_{RB}, \dot{\theta}_{RB}, \dot{\psi}_{RB}] \tag{6b}$$

In order to couple the two models it is necessary to express Eq. (6) in state variable form, and then form an augmented system with Eqs. (4) and (5). Toward this end we define the rigid body state variables as follows:

$$(\dot{x}_1)_{RB} = (x_2)_{RB} = \dot{\phi}_{RB}$$

$$(\dot{x}_2)_{RB} = \ddot{\phi}_{RB}$$

$$(\dot{x}_3)_{RB} = (x_4)_{RB} = \dot{\theta}_{RB} \qquad (7)$$

$$(\dot{x}_4)_{RB} = \ddot{\theta}_{RB}$$

$$(\dot{x}_5)_{RB} = (\dot{x}_6)_{RB} = \dot{\psi}_{RB}$$

$$(\dot{x}_6)_{RB} = \ddot{\psi}_{RB}$$

Combining Eq. (6) and (7) results in the following system:

$$\dot{x}_{RB} = A_{RB}\, x_{RB} + B_{RB}\, u \qquad (8)$$

$$y_{RB} = C_{RB}\, x_{RB} \qquad (9)$$

Where A_{RB} is a (6 x 6) matrix, B_{RB} is (6 x 8) and C_{RB} is (14 x 6). These matrices are given below:

$$A_{RB} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (10)$$

$$B_{RB}^{\infty} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 8.844\text{x}10^{-7} & -8.46\text{x}10^{-10} & 1.431\text{x}10^{-8} & 5.741\text{x}10^{-7} \\ 0 & 0 & 0 & 0 \\ -8.467\text{x}10^{-10} & 1.427\text{x}10^{-7} & 1.035\text{x}10^{-9} & -1.846\text{x}10^{-5} \\ 0 & 0 & 0 & 0 \\ 1.431\text{x}10^{-8} & 1.035\text{x}10^{-9} & 1.408\text{x}10^{-7} & 4.433\text{x}10^{-6} \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1.149\text{x}10^{-4} & 8.84\text{x}10^{-7} & -8.467\text{x}10^{-10} & 1.431\text{x}10^{-8} \\ 0 & 0 & 0 & 0 \\ -9.037\text{x}10^{-8} & 8.467\text{x}10^{-10} & 1.427\text{x}10^{-7} & 1.035\text{x}10^{-9} \\ 0 & 0 & 0 & 0 \\ 4.491\text{x}10^{-6} & 1.431\text{x}10^{-8} & 1.035\text{x}10^{-8} & 1.408\text{x}10^{-7} \end{bmatrix} \quad (11)$$

$$C_{RB} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

The total system, which we call the plant is then given by:

$$\dot{x}_p = A_p x_p + B_p u_p \tag{13}$$

$$y_p = C_p x_p \tag{14}$$

Where the augmented matrices A_p, B_p and C_p are given by

$$A_p = \begin{bmatrix} A & 0 \\ \hline 0 & A_{RB} \end{bmatrix} \quad (16x16) \tag{15}$$

$$B_p = \begin{bmatrix} B \\ \\ B_{RB} \end{bmatrix}_{(16x8)} \tag{16}$$

$$C_p = [C \quad C_{RB}]_{(14x16)} \tag{17}$$

x_p is the (16x1) state vector given by

$$x_p^T = [x \mid x_{RB}]$$

Where x is the flexible model state vector of Eq. (1) and x_{RB} is the rigid body state vector of Eq. (7).

u_p is the plant control vector defined by Eq. (2), i.e.

$$u = u_p .$$

Finally the admissible controls must satisfy the inequality constraints

$$|F| < 800 \text{ lb} \tag{18}$$

$$|T| < 10{,}000 \text{ ft-lb} \tag{19}$$

Where F represents the forces applied at Reflector (F_{rx} and F_{ry}), and T the moments applied at Shuttle and Reflector (T_{sx}, T_{sy}, T_{sz} and T_{rx}, T_{ry}, T_{rz} respectively).

Our objective will be to find $u_p(t)$, within limits, in order to force ϕ_{RB} to decay from 20^o(0.349 rad) to 0^o in about 10 seconds.

II. METHOD

A. Command Generator Tracker (CGT).

In this section, we will outline the basic elements of the CGT and state the conditions for perfect model following.

Given the continuous linear system

$$\dot{x}_p(t) = A_p x_p(t) + B_p u_p(t) \tag{20}$$

$$y_p(t) = C_p x_p(t) \tag{21}$$

where $x_p(t)$ is the (nx1) plant state vector, $u_p(t)$ is the (mx1) control vector, $y_p(t)$ is the (qx1) plant output vector, and A_p, B_p are matrices with the appropriate dimensions. Our objective is to find the control $u_p(t)$ such that the plant output vector $y_p(t)$ approximates reasonably well the output of the following model:

$$\dot{x}_m(t) = A_m x_m(t) + B_m u_m(t) \tag{22}$$

$$y_m(t) = C_m x_m(t) \tag{23}$$

Where $x_m(t)$ is the (n_mx1) model state vector, $u_m(t)$ is the (mx1) model input or command, $y_m(t)$ is the (qx1) model output vector, and A_m, B_m are matrices with the appropriate dimensions. It is assumed that: the pair (A_p, B_p) is controllable and output stabilizable, the pair (A_p, C_p) is observable, and B_p has full rank.

When perfect output tracking occurs (i.e. when $y_p = y_m$ for $t \geq 0$) we define the corresponding state and control trajectories to be the ideal state x_p^*, and ideal control u_p^*, trajectories, respectively. By definition, the ideal plant satisfies the same dynamics as the real plant. Also, the output of the ideal plant is defined to be identically equal to the model output. Mathematically,

$$\dot{x}_p^*(t) = A_p x_p^* + B_p u_p^*, \text{ for all } t \geq 0 \tag{24}$$

$$y_p^*(t) = y_m(t) \Longrightarrow C_p x_p^* = C_m x_m \tag{25}$$

The basic assumption of the CGT theory is that the ideal trajectories are linear functions of the model state and model input, i.e.

$$\begin{bmatrix} x_p^*(t) \\ u_p^*(t) \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} x_m(t) \\ u_m \end{bmatrix} \tag{26}$$

In Eq. (26) we have restricted u_m to be a constant input; otherwise derivatives of the model input may be required. Matrices S_{ij}, $i, j = 1, 2$ are found as follows. Concatenate the ideal plant state equation with the ideal plant output equation described by equations (24) and (25) and get

$$\begin{bmatrix} \dot{x}_p^* \\ y_p^* \end{bmatrix} = \begin{bmatrix} A_p & B_p \\ C_p & 0 \end{bmatrix} \begin{bmatrix} x_p^* \\ u_p^* \end{bmatrix} \tag{27a}$$

Substitute Eq. (26) into Eq. (27a) and obtain

$$\begin{bmatrix} \dot{x}_p^* \\ y_p^* \end{bmatrix} = \begin{bmatrix} A_p & B_p \\ C_p & 0 \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} x_m \\ u_m \end{bmatrix} \tag{27b}$$

Now we differentiate the first equation in (26) to obtain

$$\dot{x}_p^* = S_{11}\, \dot{x}_m + S_{21}\, \dot{u}_m$$

and since u_m is a constant input, we have

$$\dot{x}_p^* = S_{11}\, \dot{x}_m \tag{27c}$$

Now we substitute the equation for the model dynamics into (27c) to obtain

$$\dot{x}_p^* = S_{11}\, A_m\, x_m + S_{11}\, B_m\, u_m \tag{27d}$$

We combine (27d) with (25) to obtain

$$\begin{bmatrix} x_p^* \\ y_p^* \end{bmatrix} = \begin{bmatrix} S_{11}\, A_m & B_{11}\, B_m \\ C_m & 0 \end{bmatrix} \begin{bmatrix} x_m \\ u_m \end{bmatrix} \tag{27e}$$

Now equate the right-hand sides of (27b) and (27e) and get

$$\begin{bmatrix} S_{11}A_m & S_{11}B_m \\ C_m & 0 \end{bmatrix} \begin{bmatrix} x_m \\ u_m \end{bmatrix} = \begin{bmatrix} A_p & B_p \\ C_p & 0 \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} x_m \\ u_m \end{bmatrix}$$

and noting that x_m and u_m are arbitrary we obtain

$$\begin{bmatrix} S_{11}A_m & S_{11}B_m \\ C_m & 0 \end{bmatrix} = \begin{bmatrix} A_p & B_p \\ C_p & 0 \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \tag{27f}$$

If we define

$$\begin{bmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{bmatrix} = \begin{bmatrix} A_p & B_p \\ C_p & 0 \end{bmatrix}^{-1} \tag{27g}$$

then the equations to be solved are

$$S_{11} = \Omega_{11} S_{11} A_m + \Omega_{12} C_m \tag{28a}$$

$$S_{12} = \Omega_{11} S_{11} B_m \tag{28b}$$

$$S_{21} = \Omega_{21} S_{11} A_m + \Omega_{22} C_m \tag{28c}$$

$$S_{22} = \Omega_{21} S_{11} B_m \tag{28d}$$

The existence of the inverse requires that the number of controls, m, equals the number of outputs, q. If $m > q$ a pseudo-inverse may be required, while the case $m < q$ might not have a solution.

For the SCOLE model, as seen in Section I, $m < q$. However, as will be seen later, this requirement is not necessary if the so-called conditions of perfect model following (PMF) can be satisfied.

Going back to the CGT controller, Eq. (26), the ideal control is given by

$$u_p^* = S_{21} x_m + S_{22} u_m \tag{29}$$

The existence of S_{ij} is assured provided (1) u_m is a constant, (2) $m > q$, and (3) the product of the i-th eigenvalue of Ω_{11} and the j-th eigenvalue of A_m does not equal unity for all i, j [2].

If at some time $x_p = x_p^*$ then $u_p = u_p^*$ will assure perfect tracking. However, if $y_p \neq y_m$ at $t = 0$, we may achieve asymptotic tracking provided a stabilizing output feedback gain is included in the control law. To see this, consider the error equation

$$\dot{e} = \dot{x}_p^* - \dot{x}_p = A_p x_p^* + B_p u_p^* - A_p x_p - B_p u_p = A_p e + B_p (u_p^* - u_p)$$

Now, with an output stabilizing gain, the control law is

$$u_p = u_p^* + K (y_m - y_p) \tag{30}$$

$$= u_p^* + K C_p e \tag{30a}$$

then the error equation becomes

$$\dot{e} = (A_p - B_p K C_p) e \tag{30b}$$

and the error will approach zero provided that K is an output stabilizing feedback gain.

Note that the time for y_p to approach y_m depends upon the eigenvalues of the matrix $(A_p - B_p K C_p)$.

B. Perfect Model Following (PMF)

The perfect model following conditions are a special case of the CGT when (1) the state vector is available, and (2) it is assumed that $x_p^*(t) = x_m(t)$, i.e. $C_p = C_m = I$.

Therefore from (26) the PMF conditions imply that

$$S_{11} = I$$

$$S_{12} = 0 \tag{31}$$

The ideal plant input, given by Eq. (26), is, as in the CGT controller,

$$u_p^* = S_{21} x_m + S_{22} u_m$$

To evaluate matrices S_{21} and S_{22} we proceed as follows. Substitute (31) into Eq. (27f) to obtain

$$\begin{bmatrix} A_m & B_m \\ I & 0 \end{bmatrix} = \begin{bmatrix} A_p & B_p \\ I & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ S_{21} & S_{22} \end{bmatrix} \tag{31a}$$

From Eq. (31a) we obtain the desired equations in S_{21} and S_{22}:

$$A_m = A_p + B_p S_{21}$$

$$B_m = B_p S_{22}$$

Therefore matrices S_{21} and S_{22} can be found if it is possible to solve the following equations

$$(A_m - A_p) = B_p S_{21} \tag{32a}$$

$$B_p S_{22} = B_m \tag{32b}$$

It should be noted that a solution of Eqs. (32a) and (32b) may exist even if the number of outputs, q, is larger than the number of controls, m.

We should point out, however, that the PMF has certain disadvantages which can make the design more difficult, namely

(i) We assume that the state vector is available. This is not a realistic assumption.

(ii) In a PMF controller all the plant states are forced to follow all the model states by definition. This fact shows that the dimension of A_m is equal to the dimension of A_p. If dim A_p is large, like in the SCOLE project, computation of S_{21} and S_{22} may be difficult.

If at some time $x_p = x_p^*$, then $u_p = u_p^*$ will assure perfect tracking. However if $x_p \neq x_m$ at $t = 0$ a state feedback stabilizing gain is required to achieve asymptotic tracking. Using a method similar to the one that led to Eq. (30b) it can be proved that the error equation for the PMF case is

$$\dot{e} = (A_p - B_p K)e$$

and the control law will be of the form

$$u_p = u_p^* + K\,(x_m - x_p) \tag{33}$$

where K is a state feedback stabilizing gain. Note that in this case the error is given by $e = x_p^* - x_p = x_m - x_p$ because by definition $x_p^* = x_m$ in the PMF theory.

A block diagram of the system (Plant, Model, and PMF) is shown in Figure 1.

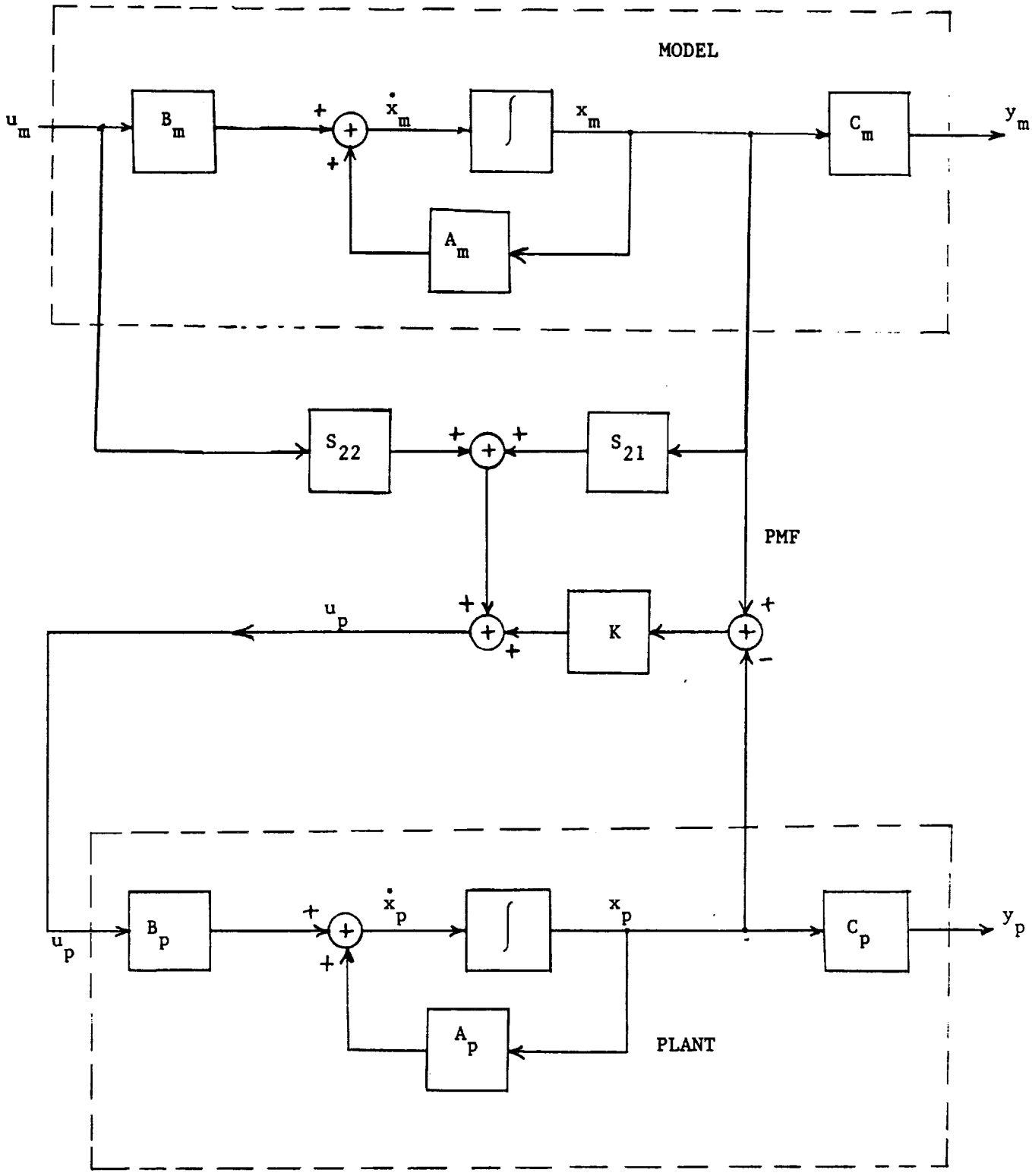


Figure 1: System block diagram.

III. DESIGN PROCEDURE

A. Approach: PMF vs. CGT.

As a preliminary controller design for the linearized SCOLE project we designed a PMF controller so that all states of the process follow all the states of the reference model. In this manner all 14 outputs defined in (3) should have acceptable responses.

For a CGT controller it is required that $m \geq q$ if equation (28a) to (28d) have a solution. It should be noted, however, that it is possible to design a CGT controller if appropriate changes are made in the dimension of the output vector or in the number of plant outputs that are to follow an equal number of model outputs. These considerations are discussed in Section IV.

Here we present the design criterion of a PMF controller. Specifically, in the subsections to follow, we present the design considerations and method of evaluation of matrices A_m, S_{21}, S_{22}, and K.

B. Plant:

The plant for the SCOLE problem is given by

$$\dot{x}_p = A_p x_p + B_p u_p \tag{34a}$$

$$y_p = C_p x_p \tag{34b}$$

where A_p is (16 x 16), B_p is (16 x 8) and C_p is (14 x 16). All these matrices are given in equations (15), (16) and (17). The controls u_p are constrained by (18) and (19), and the objective of the problem is to have $\phi_{RB} \to 0$ in about 10 seconds.

C. Model.

The model is given by

$$\dot{x}_m = A_m x_m + B_m u_m \tag{35a}$$

$$y_m = C_m x_m \tag{35b}$$

According to the theory developed in the last section for a PMF controller design we should choose A_m and B_m so that the system specifications and Eq. (32) are satisifed. Thus dim A_m = dim A_p, and dim B_m = dim B_p.

For the present design matrix B_m was selected to be equal to B_p. It should be noted that the set of equations (32) have a solution when the column vectors of the difference matrix $(A_m - A_p)$ and of the matrix B_m are linearly dependent on the column vectors of the matrix B_p. By choosing $B_m = B_p$ this condition is satisfied for Eq. (32b). For Eq. (32a) to have a solution A_m and A_p should have their odd numbered rows equal.

The design criteria for A_m was to have ϕ_{RB} 0 in about 10 seconds. Similar to the plant, the model will have 8 modes (16 states). Modes 6, 7, and 8 correspond to ϕ_{RB}, θ_{RB} and ψ_{RB} respectively. The first five modes are the flexible modes of the system. The model matrix is given by

$$A_m = \begin{bmatrix} A_{m1} & & & & & & & \\ & A_{m2} & & & & & & \\ & & A_{m3} & & 0 & & & \\ & & & A_{m4} & & & & \\ & & & & A_{m5} & & & \\ & & & & & A_{m6} & & \\ & 0 & & & & & A_{m7} & \\ & & & & & & & A_{m8} \end{bmatrix}$$

The A_{mi}'s represent second order systems of the form

$$A_{mi} = \begin{bmatrix} 0 & 1 \\ -\omega_{ni} & -2\,\omega_{ni} \end{bmatrix}$$

where ζ is the damping ratio, and ω_{ni} is the natural frequency of the i-th mode. In order to keep the overshoot at a reasonable level we will fix the damping ratio to $\zeta = .707$. Therefore, the design criteria will be the selection of the ω_{ni}.

ω_n will affect both the 2% settling time of each mode as well as the control law $u_p(t)$ through matrix S_{21} (see Eqs. (32a) and (29)). The effect of the $\omega_{ni's}$ (and thus of A_m) on the control $u_p(t)$ was tested by selecting appropriate natural frequencies (by the design procedure stated below) for A_m, running computer simulations, and observing the values obtained for the u_p's. Using this empirical method we found out that the larger the values of the $\omega_{ni's}$ the larger become the $u_{p's}$. Therefore we selected the smallest natural frequencies that would satisfy the specifications and the constraints on $u_p(t)$.

The 2% settling time criterion was applied as follows: For a second order system the 2% settling time, T_s, is given by

$$T_s = \frac{4}{\zeta \omega_n}$$

Each mode of matrix A_m will exhibit a settling time given by the last equation. Now, our objective in this report is to have the sixth mode (ϕ_{RB}) decay in about 10 seconds. Therefore, $T_s \leq 10$ sec. for this mode. Then, with $\zeta = 0.707$ the frequency of this mode should satisfy

$$\omega_n \geq \frac{4}{\zeta T_s}$$

or,

$$\omega_n \geq 0.567 \text{ rad/sec.} \qquad (36)$$

The other 7 natural frequencies were selected as explained before in order to satisfy, if possible, the constraints on the control.

The final A_m is shown in the appendix.

C. PMF Design

For a PMF we should find matrices S_{21} and S_{22} which should satisfy

$$(A_m - A_p) = B_p S_{21} \tag{37a}$$

$$B_p S_{22} = B_m \tag{37b}$$

C.1. Evaluation of S_{21}

Define

$$\Delta A \triangleq A_m - A_p \tag{38}$$

Then, Eq. (37a) becomes

$$B_p S = \Delta A \tag{39}$$

where ΔA is the (16 x 16) matrix given by Eq. (38). In Eq. (39) we have dropped the subscript of matrix S_{21} for clarity. Note also that matrices B_p and S have dimensions (16 x 8) and (8 x 16) respectively.

To solve for S in Eq. (39) we will make use of the structures of matrices B_p and ΔA. Both of these matrices in the SCOLE project have their odd rows equal to zero. Therefore Eq. (39) becomes

$$B_p S = \begin{bmatrix} 0 \\ BS_2 \\ 0 \\ BS_4 \\ 0 \\ . \\ . \\ . \\ . \\ 0 \\ BS_{16} \end{bmatrix} = \begin{bmatrix} 0 \\ \Delta A_2 \\ 0 \\ \Delta A_4 \\ 0 \\ . \\ . \\ . \\ . \\ 0 \\ \Delta A_{16} \end{bmatrix} = \Delta A \tag{40}$$

where BS_2, BS_4,.....,BS_{16} are the even rows of the product $B_p S$, and ΔA_2, ΔA_4,....ΔA_{16} the even rows of ΔA. Eq. (40) can be written as

$$\begin{bmatrix} BS_2 \\ BS_4 \\ . \\ . \\ . \\ . \\ BS_{16} \end{bmatrix} = \begin{bmatrix} \Delta A_2 \\ \Delta A_4 \\ . \\ . \\ . \\ . \\ \Delta A_{16} \end{bmatrix} \tag{41}$$

The left-hand side of Eq. (41) is the product of a reduced B_p matrix times the S_{21} matrix. This reduced B_p matrix is the original B_p matrix without its odd rows. We will call it B_{PR}. Then,

$$B_{PR} = \begin{bmatrix} B_2 \\ B_4 \\ B_6 \\ . \\ . \\ . \\ . \\ B_{16} \end{bmatrix}_{(8x8)} \tag{42}$$

where the $B_{i's}$, i = 2, 4,....,16 are the even rows of B_p.

The right-hand side of EQ. (41) is the corresponding reduced ΔA matrix. We call it ΔA_R. Therefore Eq. (41) can be written as

$$B_{PR}\, S = \Delta A_R \tag{43}$$

Note that the dimensions of B_{PR}, S, and ΔA_R are (8 x 8), (8 x 16) and (8 x 16) respectively. For the SCOLE project matrix B_{PR} is non-singular; therefore equation (43) has a unique solution given by

$$S_{21} = B_{PR}^{-1} \Delta A_R \tag{44}$$

In summary we have

(i) Matrix S_{21} is given by Eq. (44) and is unique.

ii) Matrix S_{21} depends on matrix A_m.

The computation of Eq. (44) was done by a computer program, and S_{21} is presented in the appendix.

C.2. Evaluation of S_{22}

S_{22} is the solution of Eq. (37b), i.e.

$$B_p S_{22} = B_m$$

Because in our model we selected $B_p = B_m$, we have

$$B_p S_{22} = B_p \tag{45}$$

The dimensions of B_p and S_{22} are (16 x 8) and (8 x 8) respectively. Using the method developed to evaluate S_{21} in section C1, we can express Eq. (45) as a set of 64 equations in 64 unknowns (the elements of S_{22}). The solution of this set of equations is unique and, therefore, matrix S_{22} is the (8 x 8) identity matrix, i.e. $S_{22} = I_8$.

D. State Feedback Stabilizing Gain, K.

A feedback stabilizing gain matrix, K, was designed using the pole assignment method.

The open-loop plant is given by

$$\dot{x}_p = A_p x_p + B_p u_p$$

and the feedback systems becomes

$$\dot{x}_p (A_p - B_p K) x_p$$

when K is the (8 x 16) feedback gain matrix.

The poles of the matrix $(A_p - B_p K)$ were located to the left of the poles of the model in order to achieve fast tracking in the event that the

initial conditions of the plant and model states are different, i.e. $x_p \neq x_m$ at $t = 0$.

The actual design of the time invariant matrix K was achieved by means of a computer program using the method given by Brogan [3].

Matrix K is given in the Appendix.

E. Computation

To simplify computations the plant state, x_p, the model state x_m, and the command u_m are concatenated into one system of 40 "new" states; the first 16 states are the plant states, the next 16 states are the model states, and the last 8 states are the command, u_m.

The commands are unit step functions and are modeled as follows:

$$\dot{u}_m = 0.0, \quad u_m(0) = 1.0$$

Therefore the augmented system becomes

$$\dot{x}_p = A_p x_p + B u_p \tag{46a}$$

$$\dot{x}_m = A_m x_m + B u_m \tag{46b}$$

$$\dot{u}_m = 0.0 \tag{46c}$$

and

$$u_p = S_{21} x_m + u_m + K (x_m - x_p) \tag{46d}$$

Substituting (46d) into 46a) gives

$$\dot{x}_p = (A_p - BK)x_p + B(S + K)x_m + Bu_m \tag{46e}$$

where $B \triangleq B_p = B_m$

The concatenated system then becomes:

$$\dot{x} = Ax \tag{47}$$

where

$$\dot{x} = \begin{bmatrix} \dot{x}_p \\ \dot{x}_m \\ \dot{u}_m \end{bmatrix} \quad (40x1) \tag{48}$$

and

$$A = \begin{bmatrix} (A_p - BK) & B(S+K) & B \\ 0 & A_m & B \\ 0 & 0 & 0 \end{bmatrix}_{(40x40)} \tag{49}$$

A is a (40 x 40) matrix.

The state and model output are given by

$$y_p = C\, x_p \tag{50a}$$

$$y_m = C\, x_m \tag{50b}$$

where $C = C_p = C_m$ and is given by equation (17). x_p and x_m are:

$$x_p^T = [x_1\ x_2 \ldots . x_{16}] \tag{51a}$$

$$x_m^T = [x_{17}\ x_{18} \ldots . x_{32}] \tag{51b}$$

where the $x_{i's}$ are given by Eq. (47). The rigid body states are:

$$(x_{RB}^T)_p = [x_{11}\ x_{12} \ldots . x_{16}] \tag{52a}$$

$$(x_{RB}^T)_m = [x_{27}\ x_{28} \ldots . x_{32}] \tag{52b}$$

The total displacement becomes

$$y_{TD} = y + y_{RB} \tag{53}$$

where y is the orginal output, and y_{RB} is the rigid body output.

The actual computation of Eqs. (47), (50a) and (50b) was carried out by a computer program developed by the authors using an integration subroutine. The integration step chosen is 0.01 seconds because the absolute value of the largest eigenvalue of matrix A, Eq. (49), is less than 10.

IV. RESULTS

The results of a simulation of the system are shown in this part of the report. The initial conditons and the command values used are the following:

Plant I.C's : $x_{pi}(0) = 10.00,\ i=1,2,\ldots,10$

Model I.C's : $x_{mi}(0) = 10.50,\ i=1,2,\ldots,10$

Rigid Body Plant and Model:

$\phi_{RB}(0) = 0.349$ rad (20^{o})

$\dot{\phi}_{RB}(0) = 0.0$

$\theta_{RB}(0) = 0.0$

$\dot{\theta}_{RB}(0) = 0.0$

$\psi_{RB}(0) = 0.0$

$\dot{\psi}_{RB}(0) = 0.0$

Command : $u_{mi}(t) = 0.0,\ t \geq 0,\ i=1,2,\ldots,8$

Several other runs with different sets of initial conditions were made which are not presented in this section. However, some comparisons with the results presented here are made.

Here we present four sets of plots (1) output plant and model of flexible modes, (2) output plant and model of rigid body, (3) total displacement, and (4) controls.

Figure 2: The eight plots presented here correspond to the original (flexible system) output vector of Eq. (3), not including the reflector and shuttle attitude rates. These plots represent, then: shuttle attitude (ϕ_s, θ_s, ψ_s), deflection of tip w.r.t. base (ζ_x, ζ_y), and the reflector attitude (ϕ_r, θ_r, ψ_r). In each plot we show the behavior of the plant and model together.

It can be seen that because the initial conditions of the plant and model states are different ($x_p \neq x_m$ at $t = 0$) the plant outputs achieve

asymptotic tracking. At about t = 10 seconds x_p approaches x_m, and the plant outputs achieve perfect tracking. Other runs (not presented here) with the same initial conditions show that perfect tracking occurs for all $t \geq 0$, as expected. Therefore, in both cases PMF is accomplished.

It should be noted that it is possible to decrease the time for y_p to approach y_m by making the model natural frequencies larger. However, this consideration must be taken carefully because a faster system requires more control effort. This, and other considerations of the model design were treated in Section III of this report.

Figure 3: This figure shows plots of the rigid-body attitude, ϕ_{RB}.

From this plot we see that our objective was achieved, namely, to have $\phi_{RB} \to 0$ in about 10 seconds. A computer print-out (not included here) shows that in about 10 seconds ϕ_{RB} decays to a value less than 2% of its initial condition value ($\phi_{RB}(10) = -0.675 \times 10^{-2}$). Also, it can be seen that the plant approaches the model assymptocially, as expected.

The other rigid body modes (θ_{RB} and ψ_{RB}) are not presented here because they stay at zero level for all $t \geq 0$. This is expected because their initial conditions were set to zero.

Figure 4: In this figure the plots of some elements of the total displacement vector are presented. The total displacement vector is given by Eq. (6a), i.e.

$$y_{TD} = y + RBV$$

where y is the original output vector (flexible modes) and RBV the rigid body vector. The six plots presented here correspond to the following elements of the above vector:

$$y_{TD1} = \phi_s + \phi_{RB}$$

$$y_{TD2} = \theta + \theta_{RB}$$

$$y_{TD3} = \psi_s + \psi_{RB}$$

$$y_{TD9} = \phi_r + \phi_{RB}$$

$$y_{TD10} = \theta_r + \theta_{RB}$$

$$y_{TD11} = \psi_r + \psi_{RB}$$

The first 3 are the shuttle attitude plus rigid body, and the last 3 the reflector attitude plus rigid body.

These plots show that the total displacement approaches zero in a short period of time; in all cases in about ten seconds. The following table shows the time, in seconds, to approach zero for each output presented in Fig. 3.

	t_{TD1}	y_{TD2}	y_{TD3}	y_{TD9}	y_{TD10}	y_{TD11}
Time	5.5	7.0	7.0	10	7.5	5.0

This data was taken from a computer print-out of the simulation.

The plots of the rate elements of the vector y_{TD} are not shown here. These are: rates of shuttle attitude plus rigid body ($\dot{\phi}_s + \dot{\phi}_{RB}$, $\dot{\theta}_s + \dot{\theta}_{RB}$, $\dot{\psi}_s + \psi_{RB}$), and rates of reflector plus rigid body ($\dot{\phi}_r + \dot{\phi}_{RB}$, $\dot{\theta}_r + \dot{\theta}_{RB}$, $\dot{\psi}_r + \dot{\psi}_{RB}$). The behavior of these elements are similar to the ones presented here.

The total displacement elements y_{TD7} and y_{TD8} are the deflection of tip w.r.t. base (ζ_x and ζ_y). Because the corresponding elements of the rigid body vector are zero (see Eq. (6b)), then $y_{TD7} = y_7$ amd $y_{TD8} = y_8$ where y_7 and y_8 are the original flexible modes presented in Fig. 2.

Figure 5: This figure shows the plots of the controls, $u_p(t)$. It is clear that not all the controls are within the permissible limits given by Eqs. (18) and (19). The worst case corresponds to the first control, T_{sx}, which at t = 0 attains a value of -186,000 ft-lb; as time passes T_{sx}

decreases very fast, and at about 7 seconds its value is less than the 10,000 ft-lb limit. For $t \geq 7$ seconds its value decreases steadily.

The second and third controls (T_{sy} and T_{sz}) at $t = 0$ have values of 15,000 ft-lb and 27,000 ft-lb respectively and for $t \geq 1$ sec remain within limits.

The sixth control (T_{rx}) at $t = 0$ has a value of 10,900 ft-lb. For $t \geq 0.5$ it remains within limits. Control numbers 4, 5, 7 and 8 (F_{rx}, F_{ry}, T_{rz} and T_{rz}) remain within limits for all $t \geq 0$.

As we see the first 3 controls present very large deviations from the minimum values desired. A solution to this problem would be a new design of matrices S_{21} and K as seen from the control equation given by

$$u_p = S_{21}\, x_m + K\,(x_m - x_p) + u_m \tag{54}$$

As we recall, S_{21} is the solution to

$$(A_m - A_p) = B_p\, S_{21} \tag{55}$$

and it was shown, due to the structures of matrices $(A_m - A_p)$ and B_p, that this solution is unique. Once matrix A_m is selected, S_{21} is unique. Thus, to redesign S_{21} requires a different A_m. However A_m controls the settling times of the states (see Section III of this report). Therefore, we see the difficulty of reconciling the limits on the controls and settling time (our objective with respect to ϕ_{RB}).

In relation to redesigning matrix K, the state feedback stabilizing gain, we see from Eq. (54) that when the initial conditions of the model and plant are equal, i.e. $x_p(0) = x_m(0)$, then the second term in Eq. (54) is zero and K has no effect on u_p. To see the effect of K on u_p when $x_p(0) \neq x_m(0)$ we did a simulation with $x_p(0) = x_m(0)$ and the result showed a decrease in the values of u_p, but not enough to meet the constraints. For example, the first control, at $t = 0$, reached a value of $u_{p1} = T_{sx} =$

- 130,000 ft-lb for the case $x_p(0) = x_m(0)$. This value is much better than the value obtained when $x_p(0) = 10.0$ and $x_m(0) = 10.5$; however, in both cases u_{p1} is far from being within the limits $|u_p| \leq 10{,}000$ ft-lb. Therefore, matrix K has little effect on u_p in general.

Another approach tried was to include a saturation element in the system. This element would limit u_p to its constrained value whenever u_p is off limits. If we call this saturation value u_p^s, we see that

$$u_p^s \neq u_p = S_{21}\, x_m + K\,(x_m - x_p) + u_m$$

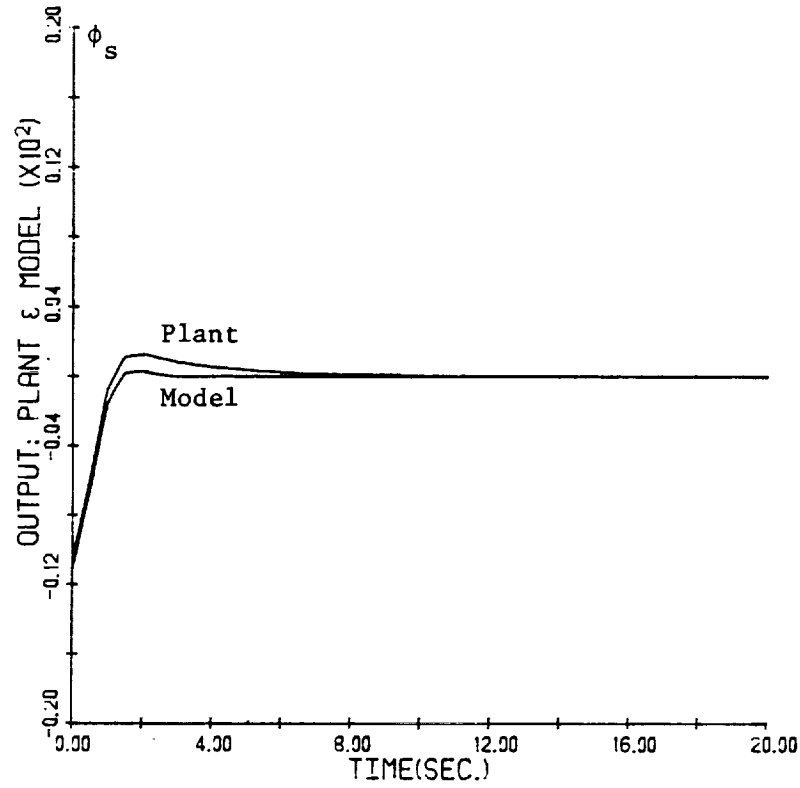
and the plant and model systems will be

$$\dot{x}_p^s = A_p\, x_p^s + B_p\, u_p^s$$

$$\dot{x}_m = A_m\, x_m + B_m\, u_m$$

where $x_p^s \equiv$ plant value when $u_p = u_p^s$. It is clear from above that $x_p \neq x_p^s$ and, because x_m is independent of u_p, it follows that the plant will not follow the model, thus destroying the purpose of the PMF controller. A simulation was done with the saturation element to see if at some time $t > 0$ the plant would approach the model, and the results in fact showed divergence. Therefore, this approach was disregarded.

As seen from the plots all $u_{pi} \to 0$ as $t \to \infty$. This should be the case because here we have $u_{mi}(t) = 0$. A simulation with the commands equal to the unit step ($u_{mi}(t) = 1(t)$) showed again that $x_p \to x_m$. We conclude, then, that because of the constraints on u_p it is very difficult, if not impossible, to design a PMF controller that will satisfy both the limits of u_p for all t and the objectives of the system. As explained in the next section, we believe that a CGT controller would be able to satisfy both.



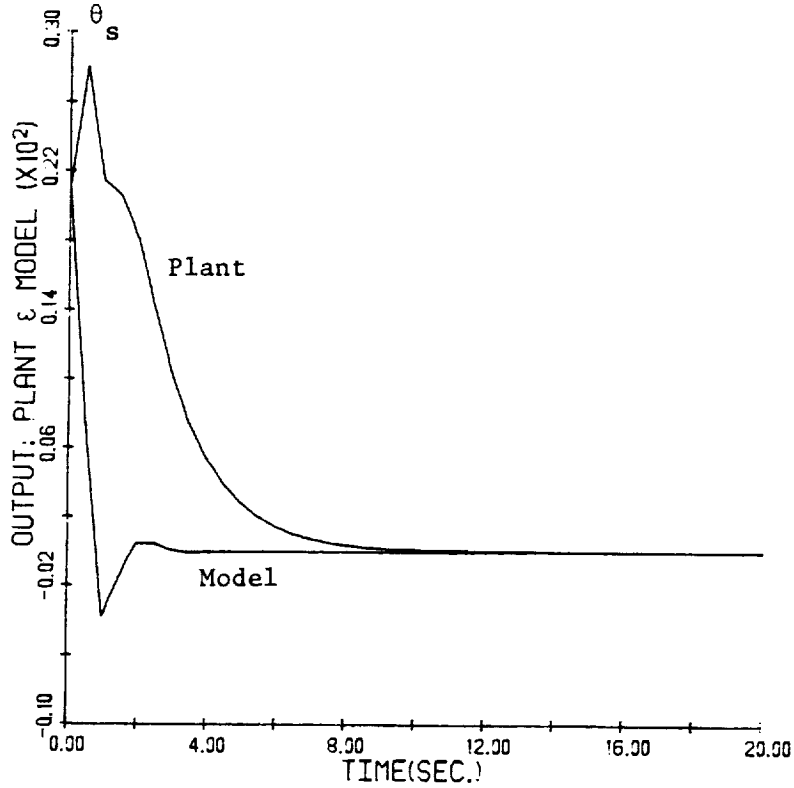
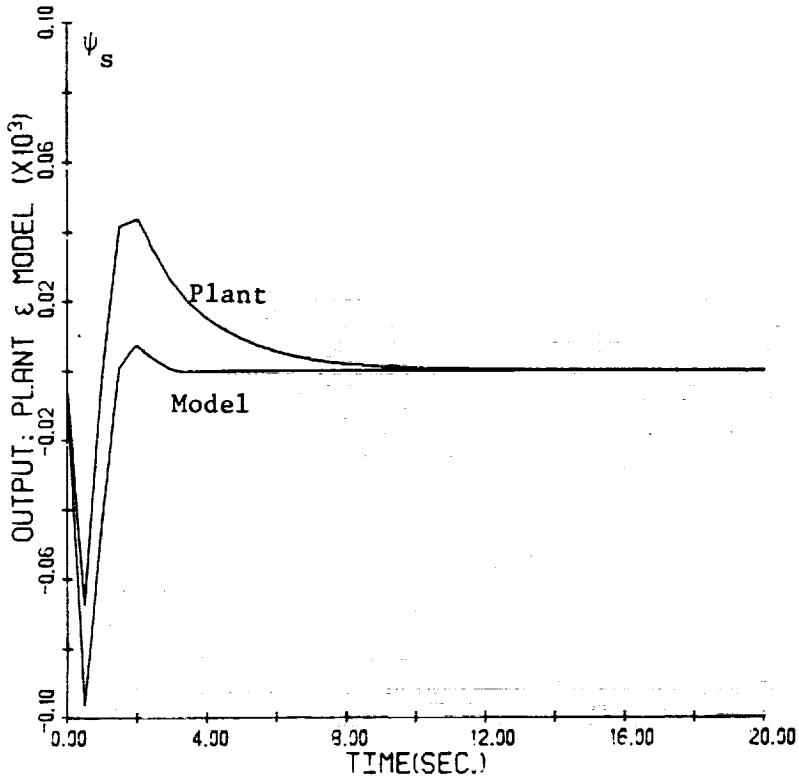


Figure 2: Selected outputs: Flexible modes



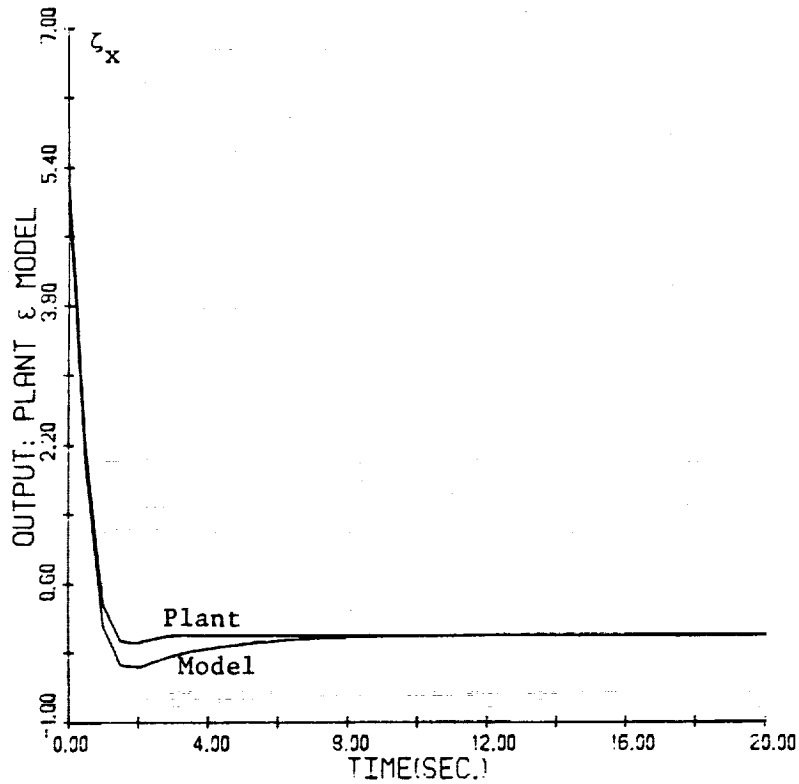
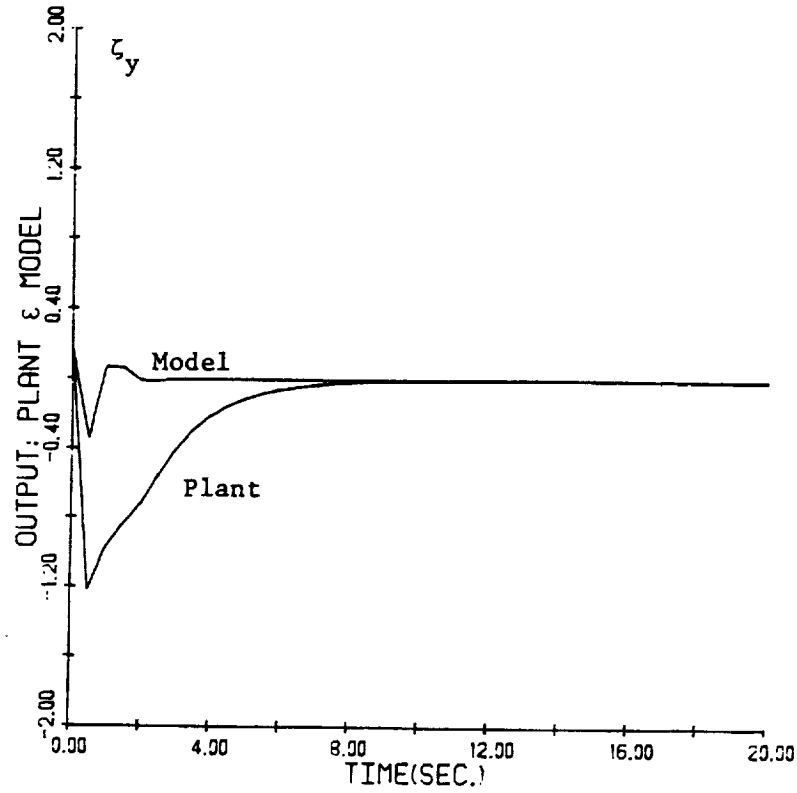


Figure 2: Selected outputs: Flexible modes, cont'd.



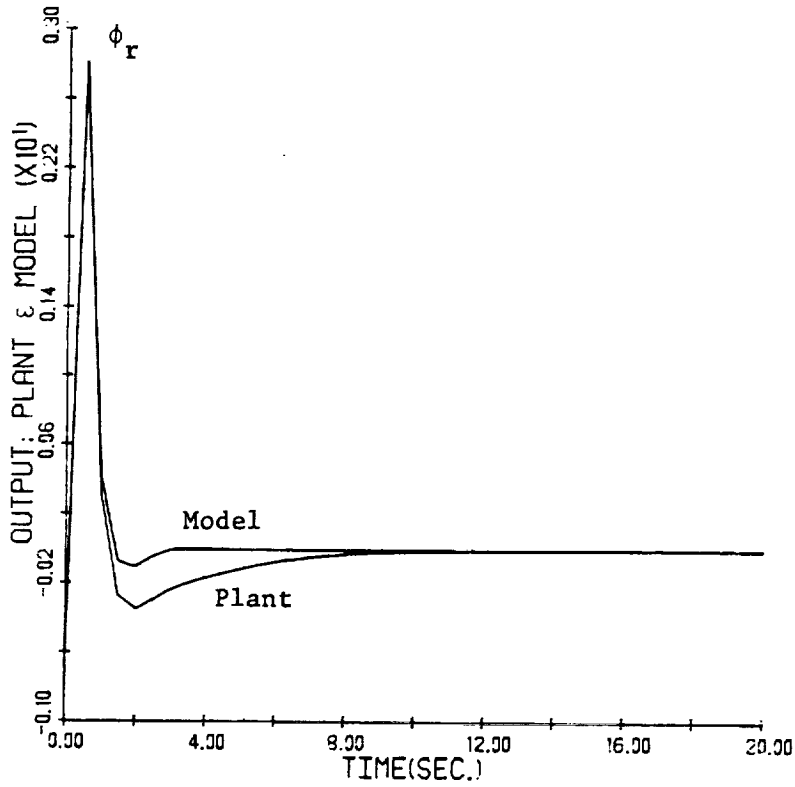
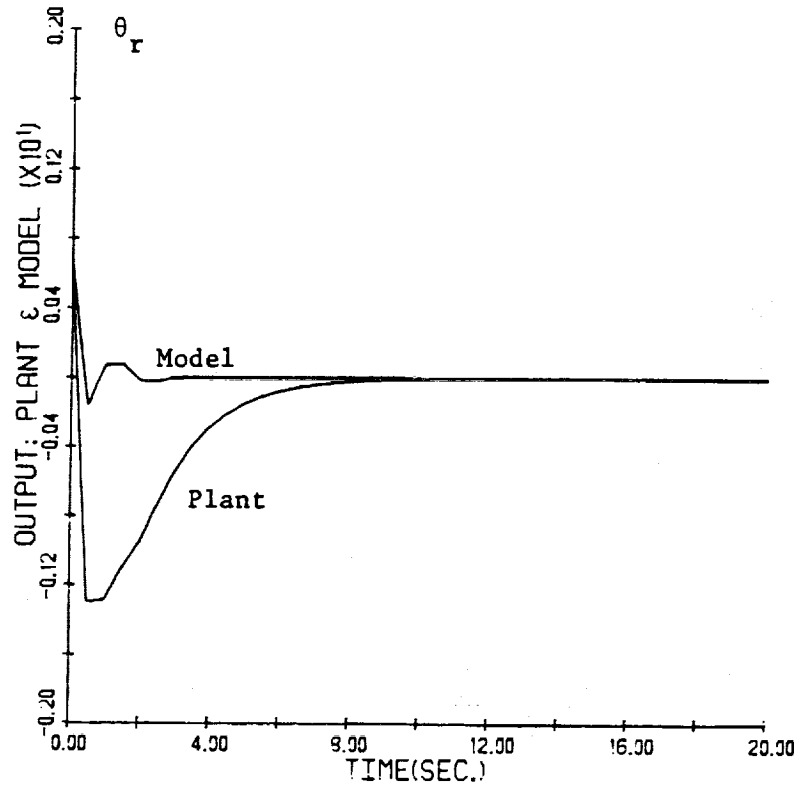


Figure 2: Selected outputs: Flexible modes, cont'd.



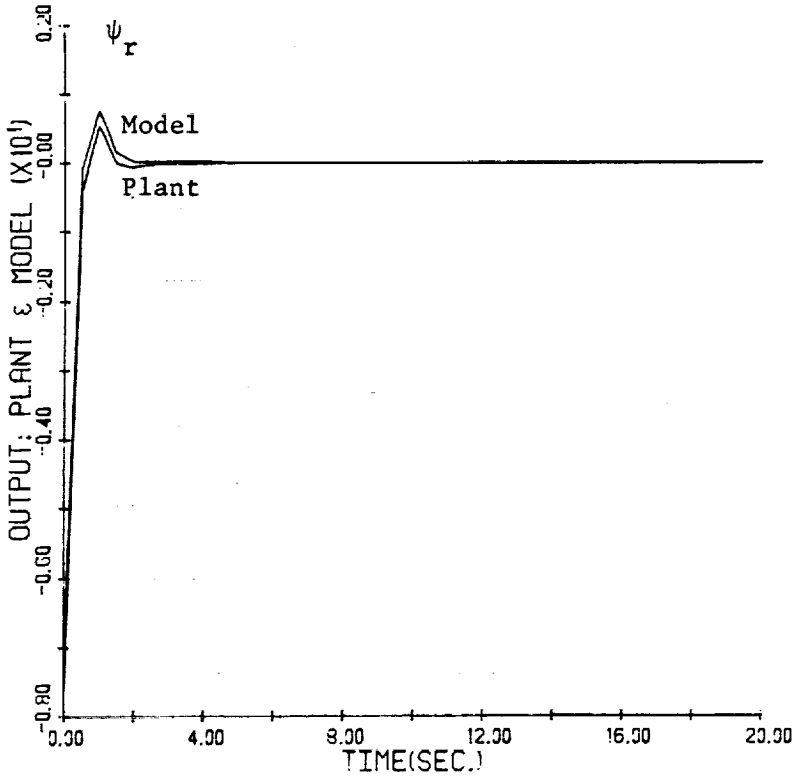


Figure 2: Selected outputs: Flexible modes, Cont'd.

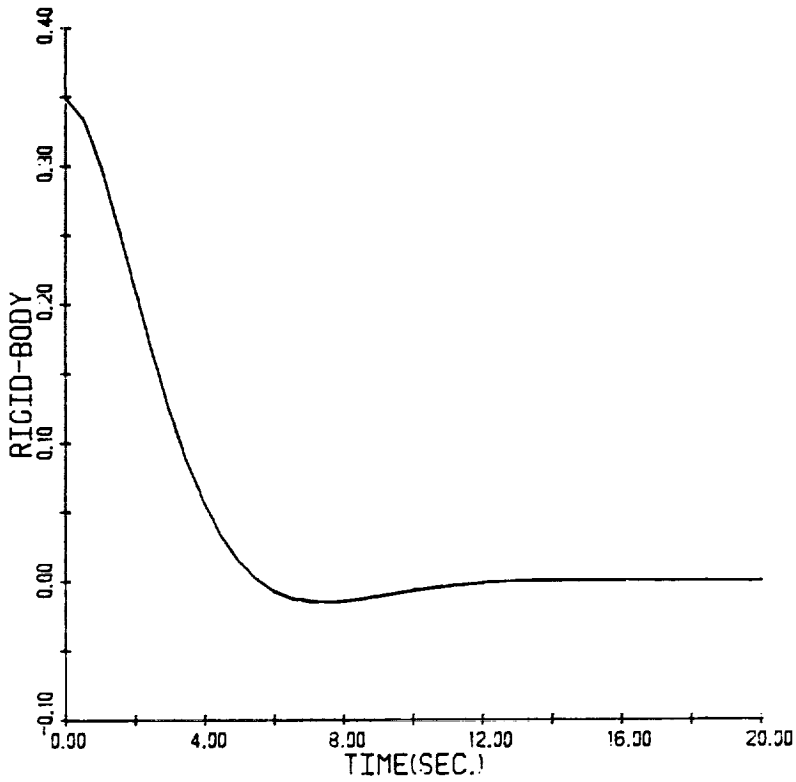
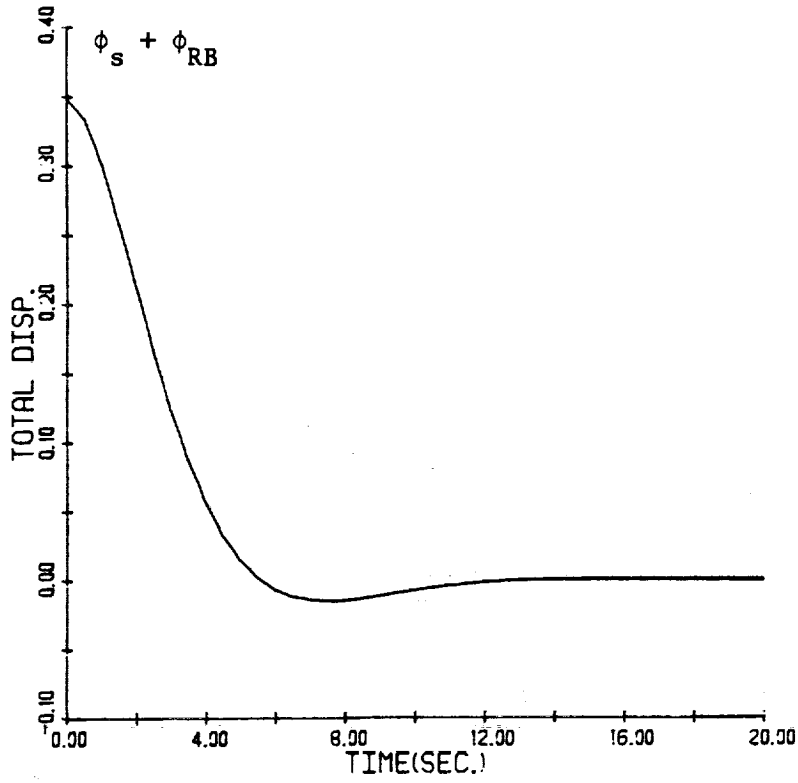


Figure 3: ϕ_{RB} vs. Time.



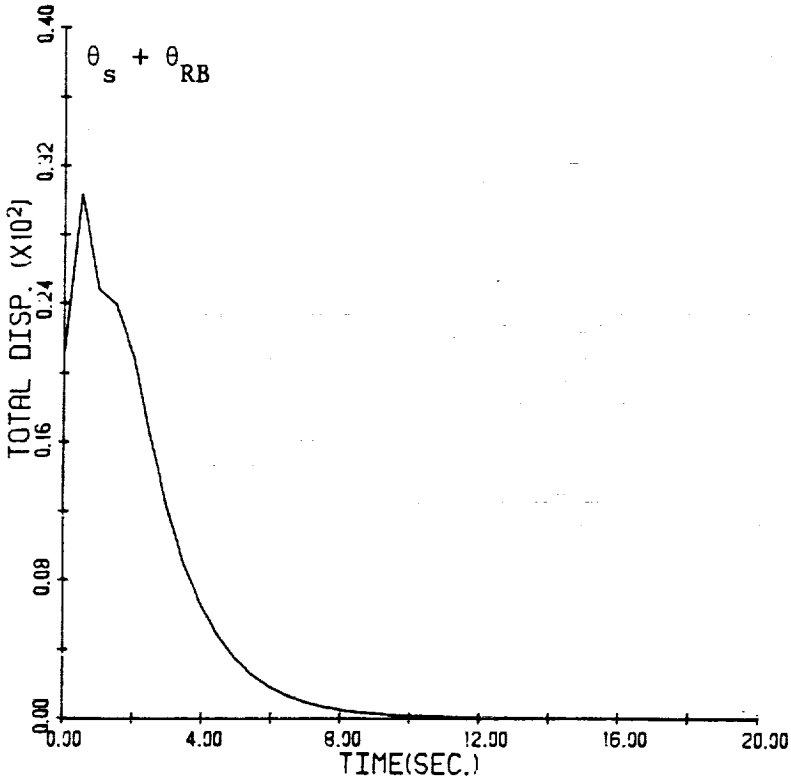
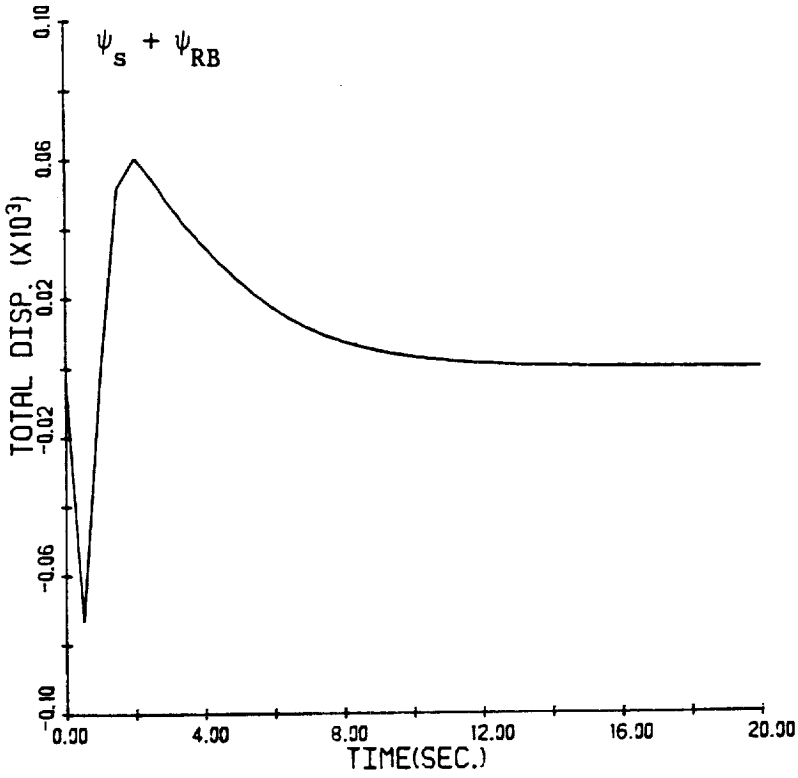


Figure 4: Selected total displacement outputs: Flexible plus rigis body modes.



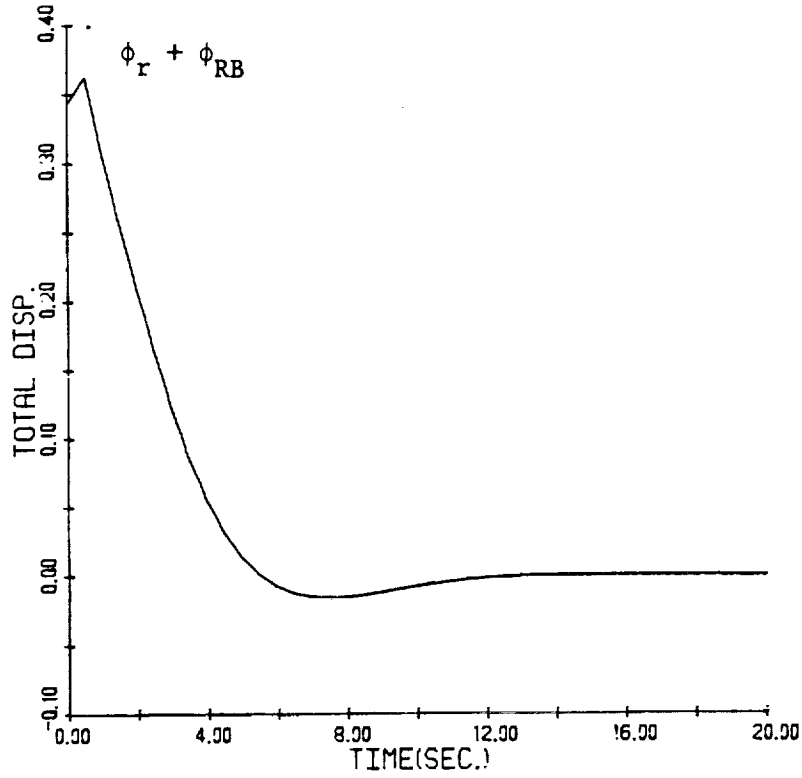
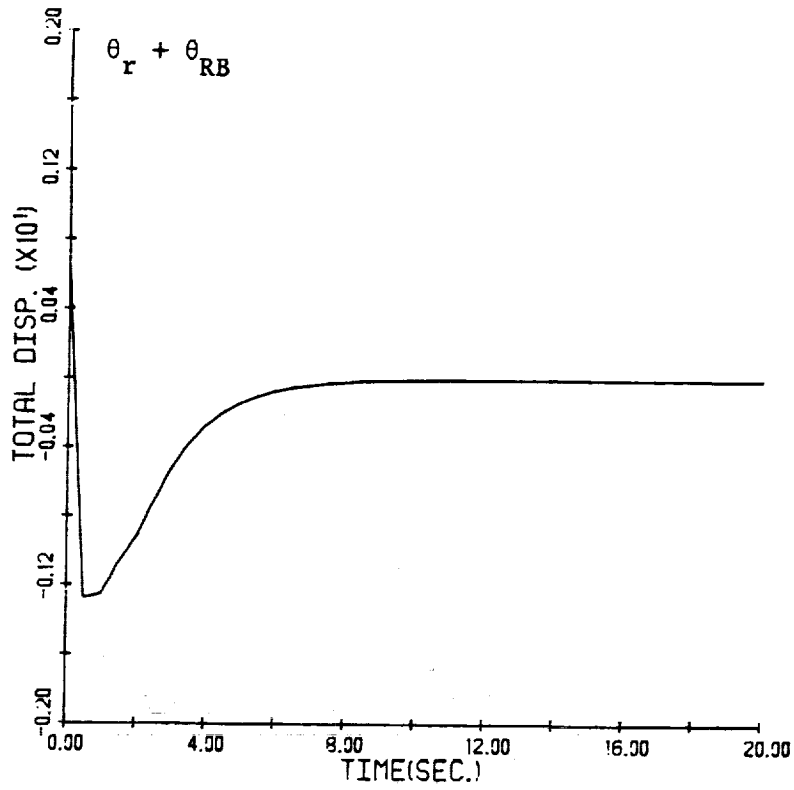


Figure 4: Selected total displacement outputs: Flexible plus rigid body modes, cont'd.



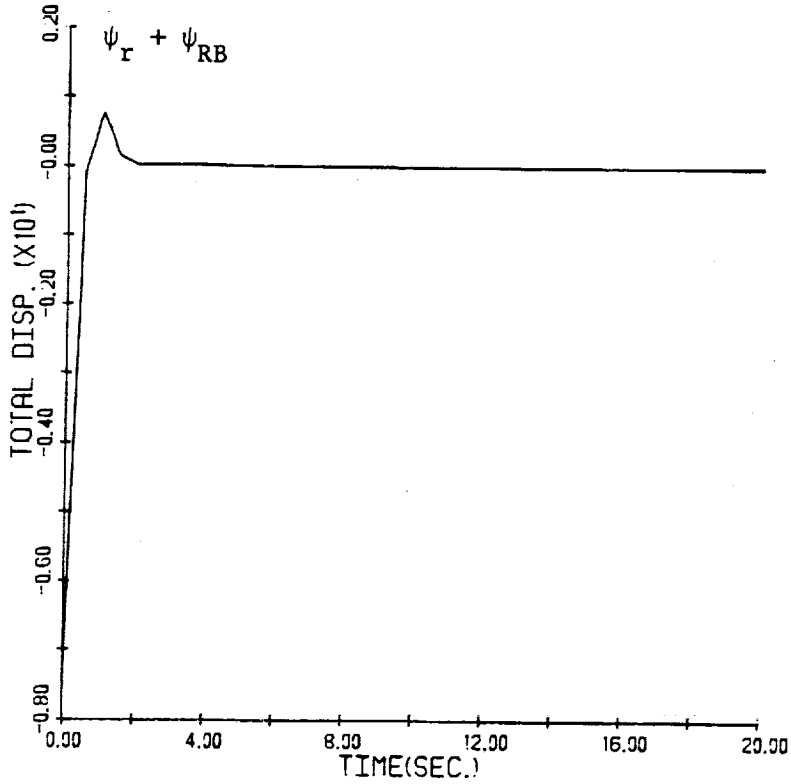
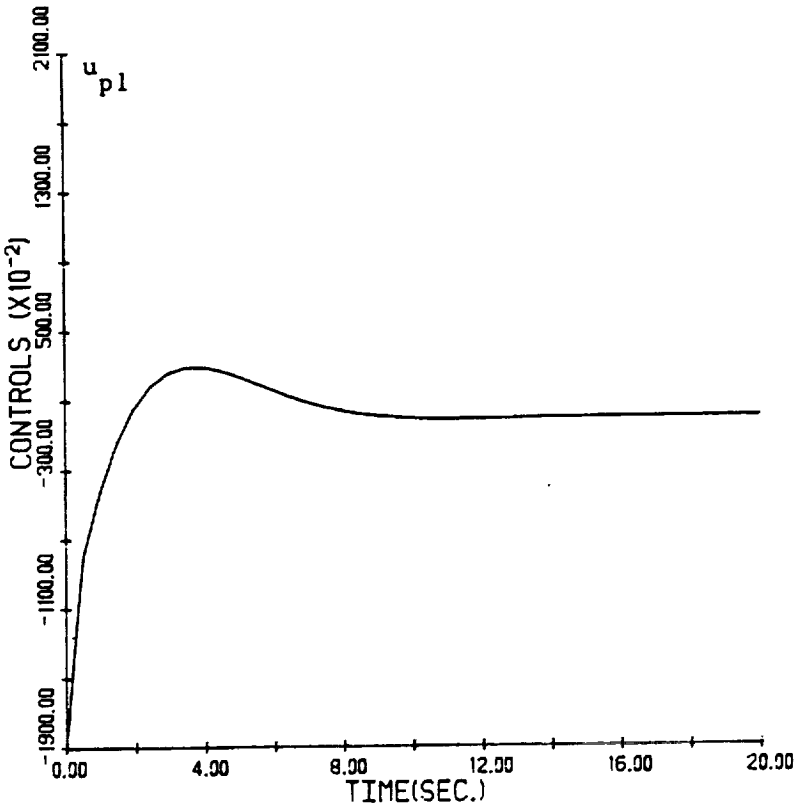


Figure 4: Selected total displacement outputs: Flexible plus rigid body modes, cont'd.



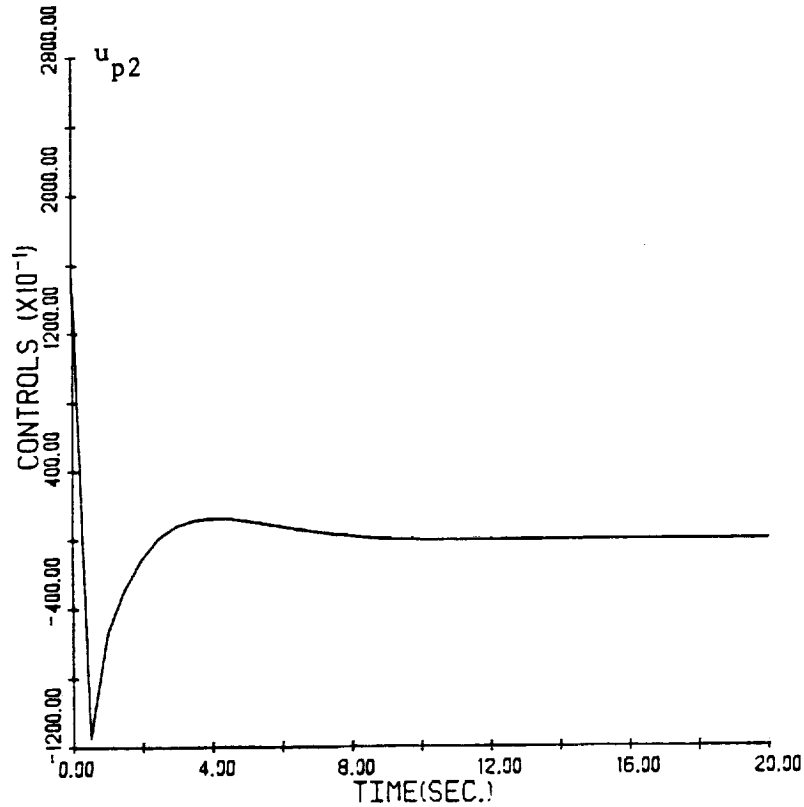
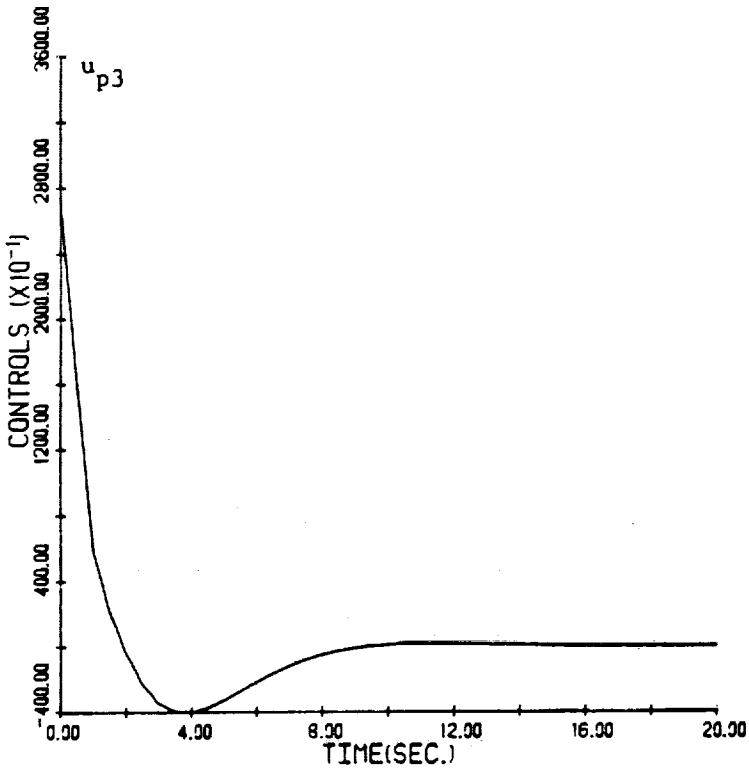


Figure 5: Controls vs. time.



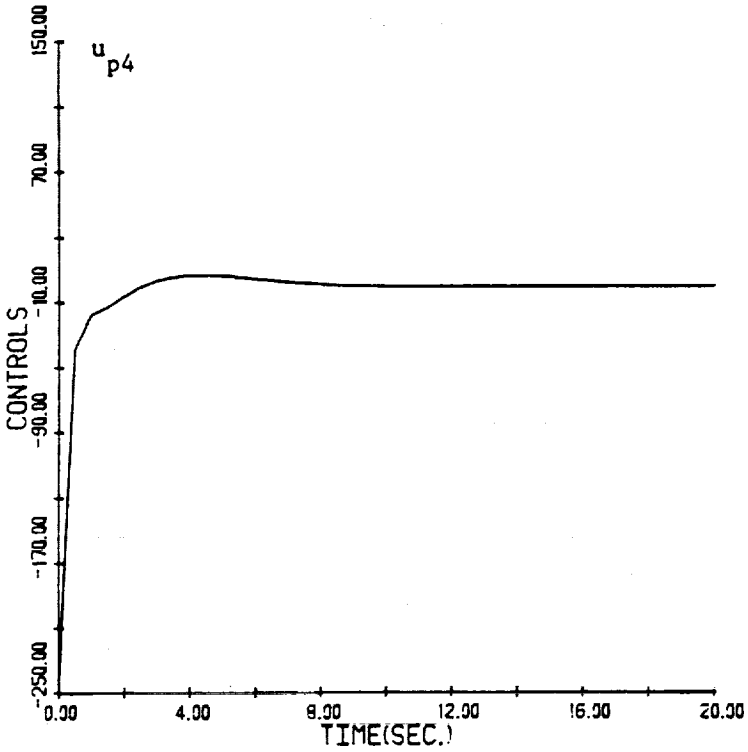
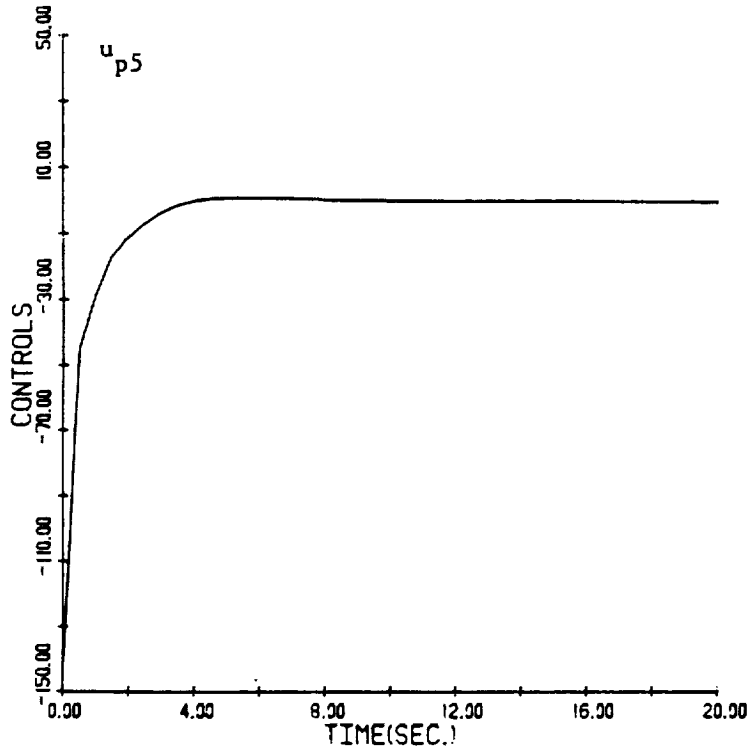


Figure 5: Controls vs. time, cont'd.



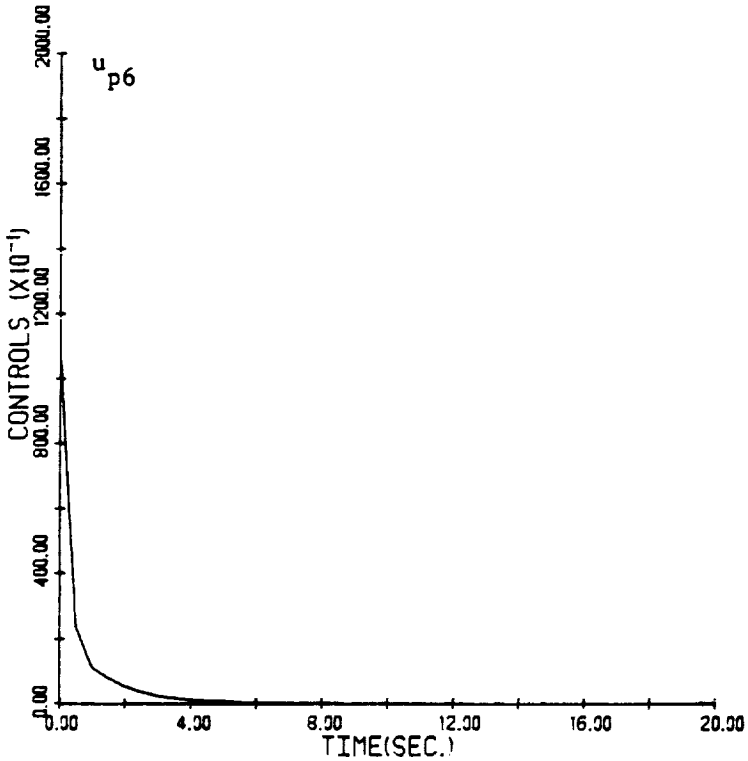
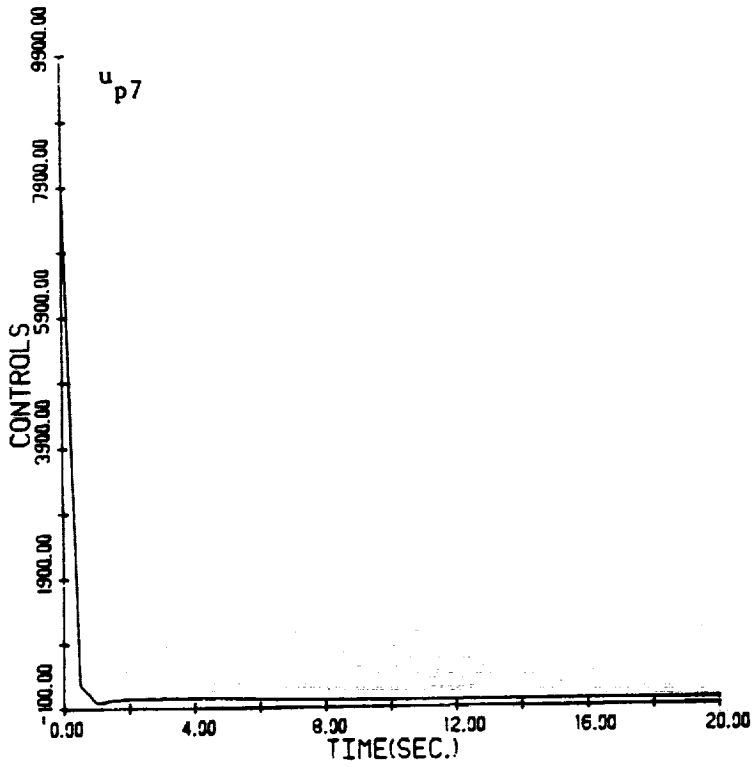


Figure 5: Controls vs. time, cont'd.



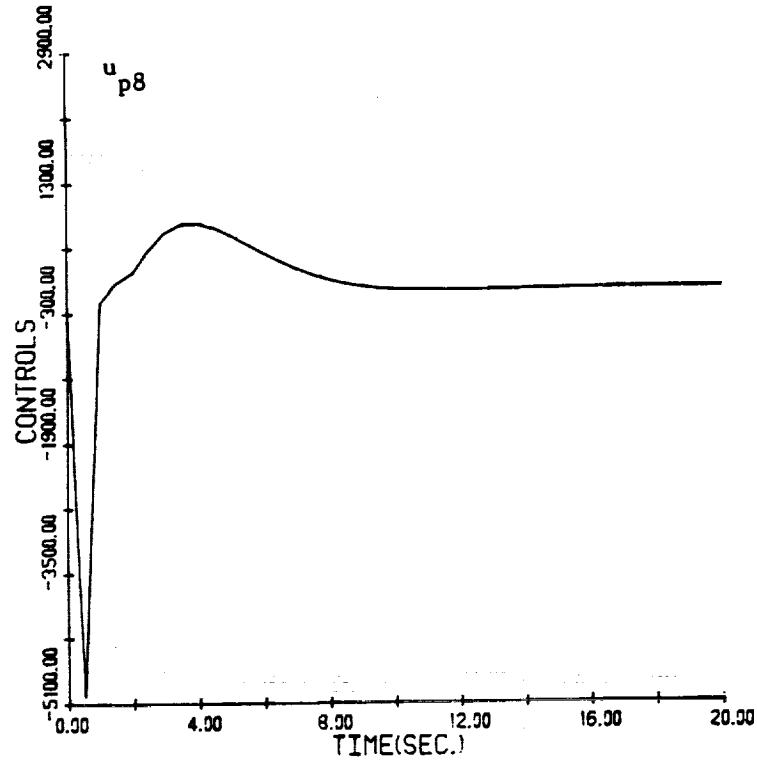


Figure 5: Controls vs. time, cont'd.

V. CONCLUSIONS AND RECOMMENDATIONS

A PMF controller design for the SCOLE project was presented here. The objective was to have $\phi_{EB} \rightarrow 0$ in about ten seconds, and to maintain the controls, $u_p(t)$ within specified limits (given by Eqs. (18) and (19)).

The first of the objectives was met as we can see from the results. The rigid body outputs, as well as the flexible mode outputs decay to zero in less than ten seconds.

However, it was not possible to meet all the constraints on the controls. The first three controls, the moments applied to the shuttle (T_{sx}, T_{sy}, T_{sz}), reach unacceptable values (-186,000, 15,000, and 27,000 ft-lb respectively) at t = 0. They decay, however, to values within the limits in 7 sec, 1.0 sec, and 1.0 sec respectively.

It was explained in the last section of this report the reasons behind the behavior of these controls: an attempt to improve the controls would require a slower model, making the response of the plant also slower (the model following action). A slower system would not meet the first objective. The main problem encountered in the PMF design, as explained, was the fact that matrix S_{21} is unique, and thus, we do not have too much freedom to select model parameters to meet both objectives.

This preliminary design, however, was attempted to prove that it is possible to design a direct model following controller for the linearized SCOLE project.

A more realistic approach is to design a CGT controller using the method presented in Section II. Unlike the PMF, the CGT does not require dim A_m = dim A_p. This is an advantage because we have the freedom to choose selected plant outputs to follow an equal number of model outputs. In a PMF we are forced to have all the plant states to follow the model states.

However, a solution to Eq. (28) of Section II requires the number of controls, m, to be equal to the number of outputs, q; or at least m > q. In this project we have m > q (m = 8 and q = 14). We could overcome this problem by re-defining the output vector of Eq. (3). Two recommendations could be explored.

(i) To have the elements of y to be the sum of position and rate as follows:

$$y^T = [\phi_s + \dot{\phi}_s,\ \theta_s + \dot{\theta}_s,\ \psi_s + \dot{\psi}_s,\ \zeta_x,\ \zeta_y,\ \phi_r + \dot{\phi}_r,\ \theta_r + \dot{\theta}_r,\ \psi_r + \dot{\psi}_r]_{(8x1)}$$

A corresponding C_p matrix should be defined accordingly. In this case we would have m = q = 8.

(ii) We could have fewer plant outputs follow an equal number of model outputs. In this case we have m > q. The selection of the plant outputs that are to follow the model outputs will depend on the objective of the design. In the SCOLE project an appropriate selection would be the inclusion of ϕ_{FB} in the output vector in order to meet the objective. Therefore the dimension of y might be as low as 1.

Because the CGT is an output follower, and because we do not assume that the state vector (or part of it) is available for feed-back, an output stabilizing gain should be designed to make the error of Eq. (30a) decay to zero in case the initial conditions of plant and model are different.

In conclusion, this preliminary approach has shown the feasibility of designing a model following type of controller for the SCOLE project. A more suitable controller should be possible using the CGT method.

References

1. Taylor, L. W. Jr., and Balakrishan, A. V., "A Mathematical Problem and a Spacecraft Control Laboratory Experiment (SCOLE) Used to Evaluate Control Laws for Flexible Spacecraft...NASA/IEEE Design Challenge" (unpublished).

2. Broussard, J. R. and O'Brien, M. J., "Feedforward Control to Track the Output of a Forced Model", 17th IEEE Conf. on Decision and Control1, 1149-1155, January 1979.

3. Brogan, W. L., "Modern Control Theory", Prentice-Hall, Inc./Quantum Publishers, Inc. 1982.

4. Sobel, K. M., "Model Reference Adaptive Control for Multi-Input Multi-Output Systems", Ph.D. Thesis, Rensselaer Polytechnic Institute, Troy, NY, 1980.

5. Bar-Kana, I. and Kaufman, H., "Some Applications of Direct Adaptive Control to Large Structural Systems", J. Guidance, Vol. 7, No. 6, Nov-Dec. 1984.

6. Hotz, A. F., Collins, E., and Skelton, R. E., "Linearized Dynamical Model for the NASA/IEEE SCOLE Configuration", NASA Contract Report 172394, September 1984.

APPENDIX

MATRIX AP

COLUMNS 1 TO 6

0.0	0.1000E+01	0.0	0.0	0.0	0.0
-.2965E+01	-.1033E-01	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.1000E+01	0.0	0.0
0.0	0.0	-.4116E+01	-.1217E-01	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.1000E+01
0.0	0.0	0.0	0.0	-.2217E+02	-.2825E-01
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0

COLUMNS 7 TO 10

0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.1000E+01	0.0	0.0
-.6111E+02	-.4690E-01	0.0	0.0
0.0	0.0	0.0	0.1000E+01
0.0	0.0	-.1662E+03	-.7735E-01

MATRIX BP

COLUMNS 1 TO 6

0.0	0.0	0.0	0.0	0.0	0.0
-.6865E-03	-.1810E-02	-.5437E-04	0.1859E+01	0.5008E+00	0.2222E-01
0.0	0.0	0.0	0.0	0.0	0.0
-.2504E-03	0.4373E-02	0.3749E-05	0.8932E+00	-.1769E+01	0.9991E-02
0.0	0.0	0.0	0.0	0.0	0.0
-.7555E-04	-.3069E-04	0.4506E-04	0.1861E+01	0.8247E+00	0.2661E-01
0.0	0.0	0.0	0.0	0.0	0.0
-.4918E-04	-.5426E-03	0.4261E-05	0.6651E+00	0.5973E+00	-.1207E-01
0.0	0.0	0.0	0.0	0.0	0.0
-.4234E-04	0.1912E-03	-.1052E-06	-.8471E-01	0.1277E-01	-.5279E-01

COLUMNS 7 TO 8

0.0	0.0
0.7367E-02	0.3658E-02
0.0	0.0
-.2349E-01	-.3516E-03
0.0	0.0
0.1858E-01	-.2295E-01
0.0	0.0
-.2641E-01	-.5986E-02
0.0	0.0
0.3063E-01	0.4014E-03

MATRIX CP

COLUMNS 1 TO 6

-.6560E-04	0.0	-.2393E-04	0.0	-.7220E-05	0.0
-.1730E-03	0.0	0.4179E-03	0.0	-.2933E-05	0.0
-.1580E-04	0.0	0.1090E-05	0.0	0.1310E-04	0.0
0.0	-.6560E-04	0.0	-.2393E-04	0.0	-.7220E-05
0.0	-.1730E-03	0.0	0.4179E-03	0.0	-.2933E-05
0.0	-.1580E-04	0.0	0.1090E-05	0.0	0.1310E-04
0.1781E+00	0.0	0.8558E-01	0.0	0.1779E+00	0.0
0.4801E-01	0.0	-.1695E+00	0.0	0.7876E-01	0.0
0.2123E-02	0.0	0.9547E-03	0.0	0.2543E-02	0.0
0.7040E-03	0.0	-.2245E-02	0.0	0.1775E-02	0.0
0.1063E-02	0.0	-.1022E-03	0.0	-.6671E-02	0.0
0.0	0.2123E-02	0.0	0.9547E-03	0.0	0.2543E-02
0.0	0.7040E-03	0.0	-.2245E-02	0.0	0.1775E-02
0.0	0.1063E-02	0.0	-.1022E-03	0.0	-.6671E-02

COLUMNS 7 TO 10

-.4699E-05	0.0	-.4046E-05	0.0
-.5185E-04	0.0	0.1828E-04	0.0
0.1239E-05	0.0	-.3059E-07	0.0
0.0	-.4699E-05	0.0	-.4046E-05
0.0	-.5185E-04	0.0	0.1828E-04
0.0	0.1239E-05	0.0	-.3059E-07
0.6366E-01	0.0	-.7978E-02	0.0
0.5724E-01	0.0	0.1150E-02	0.0
-.1154E-02	0.0	-.5045E-02	0.0
-.2523E-02	0.0	0.2927E-02	0.0
-.1740E-02	0.0	0.1167E-03	0.0
0.0	-.1154E-02	0.0	-.5045E-02
0.0	-.2523E-02	0.0	0.2927E-02
0.0	-.1740E-02	0.0	0.1167E-03

MATRIX AM

COLUMNS 1 TO 6

0.0	0.1000E+01	0.0	0.0	0.0	0.0
-.6250E+01	-.3536E+01	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.1000E+01	0.0	0.0
0.0	0.0	-.1225E+02	-.4950E+01	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.1000E+01
0.0	0.0	0.0	0.0	-.2500E+02	-.7071E+01
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0

COLUMNS 7 TO 12

0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.1000E+01	0.0	0.0	0.0	0.0
-.1000E+03	-.1414E+02	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.1000E+01	0.0	0.0
0.0	0.0	-.1690E+03	-.1838E+02	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.1000E+01
0.0	0.0	0.0	0.0	-.3600E+00	-.8485E+00
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0

MATRIX AM (CONT.)

COLUMNS 13 TO 16

0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.1000E+01	0.0	0.0
-.3600E+00	-.8485E+00	0.0	0.0
0.0	0.0	0.0	0.1000E+01
0.0	0.0	-.3600E+00	-.8485E+00

MATRIX S21

COLUMNS 1 TO 6

0.1174E+03	0.1260E+03	-.4184E+03	-.2540E+03	-.1024E+02	-.2547E+02
-.1472E+03	-.1579E+03	-.2239E+03	-.1359E+03	0.9040E+01	0.2248E+02
0.2018E+03	0.2166E+03	-.3323E+02	-.2017E+02	-.1004E+03	-.2498E+03
-.1222E+01	-.1311E+01	-.1548E+01	-.9395E+00	-.6065E-01	-.1508E+00
-.8556E+00	-.9182E+00	0.3101E+01	0.1882E+01	0.1475E+00	0.3668E+00
-.6475E+01	-.6949E+01	0.1644E+02	0.9981E+01	-.8876E+01	-.2207E+02
-.1118E+02	-.1200E+02	0.2325E+02	0.1411E+02	-.1690E+02	-.4203E+02
-.1462E+03	-.1569E+03	0.2541E+02	0.1542E+02	0.9964E+02	0.2478E+03

COLUMNS 7 TO 12

0.1458E+04	0.5283E+03	-.6665E+02	-.4358E+03	-.4020E+06	-.9474E+06
-.1530E+04	-.5544E+03	-.3980E+02	-.2602E+03	-.1885E+05	-.4443E+05
0.3076E+03	0.1115E+03	0.1719E+02	0.1124E+03	0.5757E+05	0.1357E+06
-.5783E+01	-.2096E+01	-.4910E+00	-.3210E+01	-.1291E+03	-.3043E+03
-.1474E+02	-.5342E+01	0.2063E+00	0.1349E+01	-.4443E+02	-.1047E+03
0.4528E+03	0.1641E+03	0.3991E+02	0.2610E+03	0.1432E+02	0.3375E+02
0.7800E+03	0.2827E+03	-.2384E+02	-.1559E+03	-.6064E+03	-.1429E+04
0.1558E+03	0.5647E+02	-.5120E+01	-.3348E+02	-.1108E+05	-.2612E+05

COLUMNS 13 TO 16

0.7904E+06	0.1863E+07	0.4052E+05	0.9550E+05
-.2470E+07	-.5823E+07	0.1418E+05	0.3341E+05
0.3496E+06	0.8240E+06	-.2553E+07	-.6016E+07
0.2798E+03	0.6594E+03	-.3475E+02	-.8191E+02
-.5945E+04	-.1401E+05	0.6525E+01	0.1538E+02
-.2248E+05	-.5299E+05	0.1085E+03	0.2557E+03
-.1598E+05	-.3767E+05	0.1452E+03	0.3423E+03
-.2286E+06	-.5388E+06	-.7505E+04	-.1769E+05

MATRIX K

COLUMNS 1 TO 6

-.5347E+04	-.3840E+04	-.6633E+03	-.5647E+03	0.1388E+04	0.6835E+03
0.3267E+04	0.2723E+04	0.2985E+04	0.1874E+04	-.4044E+03	-.4656E+03
0.2105E+03	-.8125E+02	-.4866E+01	0.1678E+02	-.1080E+04	-.9475E+02
-.8065E+00	0.7587E+00	-.6547E+00	0.3577E+00	-.5210E+00	0.2101E-01
0.1712E+01	0.2474E+01	0.2323E+01	-.5936E+00	0.9574E+00	-.3606E+00
0.2214E+01	0.9429E+01	0.4073E+01	-.8033E+01	-.5559E+02	0.1767E+02
-.2237E+02	-.3831E+01	-.1353E+02	-.2374E+02	-.9860E+02	0.4011E+02
0.4805E+02	0.2205E+03	0.4710E+02	-.1008E+02	0.5720E+03	-.2419E+03

CCLUMNS 7 TO 12

-.3662E+05	-.1450E+05	-.3636E+05	-.1888E+05	0.7742E+07	0.7997E+07
0.1318E+05	0.8723E+04	0.5146E+05	0.2324E+05	0.3851E+07	0.3162E+07
0.7638E+03	0.1588E+03	0.1684E+04	0.4098E+03	-.8839E+05	-.2866E+06
-.1090E+02	-.1214E+01	-.3401E+02	-.4163E+01	0.4075E+03	0.7896E+03
-.1821E+02	0.4200E+01	0.4578E+02	0.1417E+02	0.2419E+04	0.2130E+04
0.6753E+03	-.6012E+02	0.2398E+04	-.2347E+02	-.3281E+04	-.2845E+04
0.1011E+04	-.1907E+03	-.1619E+04	-.4930E+02	-.1952E+05	-.1347E+05
0.1830E+03	-.8022E+02	0.8091E+03	0.2631E+03	0.1380E+06	0.1536E+06

COLUMNS 13 TO 16

-.6811E+08	-.4936E+08	-.4502E+06	-.5725E+06
0.6411E+08	0.6324E+08	-.7270E+06	-.6220E+06
-.6235E+06	-.1878E+07	0.4961E+07	0.1347E+08
-.1684E+04	-.2183E+04	0.6416E+02	0.1847E+03
0.5073E+05	0.6779E+05	-.4317E+03	-.3581E+03
0.1447E+06	0.2136E+06	-.4024E+03	-.7419E+03
-.2233E+06	-.1043E+06	0.3105E+04	0.1749E+04
0.2271E+07	0.2980E+07	-.1716E+04	0.2939E+05

CRL CONTROL RESEARCH LABORATORY

Control and Identification Experiments for the NASA LaRC SCOLE Program

Stephen Yurkovich, Ümit Özgüner and Kathleen Ossman

CRL–1029–W88–R

The Ohio State University
Department of Electrical Engineering
2015 Neil Avenue, Columbus, OH 43210

459

Control and Identification Experiments for the NASA LaRC SCOLE Program

INTERIM REPORT

presented to

The National Aeronautics & Space Administration

February 27, 1988

submitted by

Stephen Yurkovich
Ümit Özgüner
Kathleen Ossman

Department of Electrical Engineering
2015 Neil Avenue
Columbus, Ohio 43210

Report Period:
1 Jan 1987—31 Dec 1987

460

Contents

1. INTRODUCTION

This report summarizes participation in the NASA LaRC Spacecraft Control Laboratory Experiment (SCOLE) under NASA Grant #NAG-1-720 in the study of various approaches in analysis and control of systems having characteristics of Large Space Structures (LSS). The work is in continuation of an ongoing project originated under Contract NCC1-108 entitled "Development of Decentralized Control Algorithms for Flexible Structures". Principal investigators for this continuing effort have been Professors Steve Yurkovich and Ümit Özgüner, and Prof. K. A. Ossman contributed to the effort as a Research Associate. Several graduate student research assistants, namely P. Cooper, A. Tzes, H. Fu, and M. Cheung, have assisted in the research program. M. Cheung, F. Khorrami and A. Iftar contributed to controller designs in COFS work.

The primary objective of this research has been twofold: first, to assess the performance of various control methods in simulation tests and actual experimentation, and second, to determine the effectiveness of various techniques for on-line and off-line system identification relative to control applications on the SCOLE apparatus. Justification for the objectives of this work lies in the inherent need for reliable and robust control designs in active vibration control, and in the necessity for accurate system models in most large space structure applications. Throughout our work for this program and others, we have recognized that traditional approaches, such as finite element techniques *alone*, may not in some instances be accurate enough for use in designing a vibration control system. For these reasons we have been pursuing approaches which realize a model *directly from experimental data*, considering techniques for system identification and parameter estimation. Despite recent efforts along these lines, little ground has been broken toward solution of the many problems posed, and the recognition of the immediate need for such research is apparent.

Indeed, these points are especially important for application of adaptive control techniques, a major emphasis in this work. Large space structures are excellent candidates for adaptive control because the modal data, in many cases, is not sufficiently accurate to design a fixed controller for the structure. An exact model is difficult to synthesize because the low resonant frequencies are closely packed, the damping is light and often unknown, the model parameters will change when the system is put into orbit, and the forces such as those due to gravity and temperature gradients will change continually throughout the orbit. Also, the system parameters will vary during a large angle slew. With the highly complex mechanical structures of the future and under current study, there is a need for robust adaptive controllers

which can meet the control objectives in spite of model uncertainty, nonlinearities, and spillover due to residual system interaction.

The overall group effort at Ohio State within the Department of Electrical Engineering for work at LaRC is summarized in Figure 1.1. While the focus of this report is on work related to SCOLE, we have begun preliminary studies on the MiniMast configuration at LaRC; details of these studies will appear in a later report for continuation of this project. In Figure 1.1 the dashed lines represent work in progress or projected studies, and the various acronyms which appear here and throughout this report represent, respectively, Variable Structure (VS) Control, Self-Tuning Adaptive Control (STAC), Maximum Entropy/Optimal Projection (MEOP) Control Design, and Model-Reference Adaptive Control (MRAC). Not represented in the figure, but also forming a portion of the research in this effort has been analysis and control design for the envisioned flight article in the Control of Flexible Structures (COFS) program. Because of the direction that program has recently taken, we have used this report as a vehicle for describing our findings and controller designs for the model of the COFS mast, an we will utilize the expertise gained on other configurations that may be proposed as a replacement.

The remainder of this report is organized as follows. In the next section we briefly formulate and review the basic problem as related to SCOLE. Following this is a description of the various controller design approaches taken in this research. We have grouped these design procedures into three main categories: Nonadaptive Designs, Direct Adaptive Designs, and Indirect Adaptive Designs. In each case extensive simulation studies have been undertaken in preparation for experimentation. After this is a section describing the actual experimentation completed at LaRC within the last twelve months. Following this is a brief summary of the work we completed relative to the COFS mast model. Finally, we conclude with comments on future directions in the research and in the laboratory experiments relative to SCOLE and the MiniMast.

2. SCOLE APPARATUS

The SCOLE is comprised of three basic structures, 83.8 by 54.0 inches shuttle platform, a 120 inches long mast and a hexagonal reflector panel. There are 15 sensors and 12 actuators distributed on the structure for the experiment, and our controller design studies have focused on different combinations of these actuators and sensors, for vibration suppresion in either the mast portion or the reflector

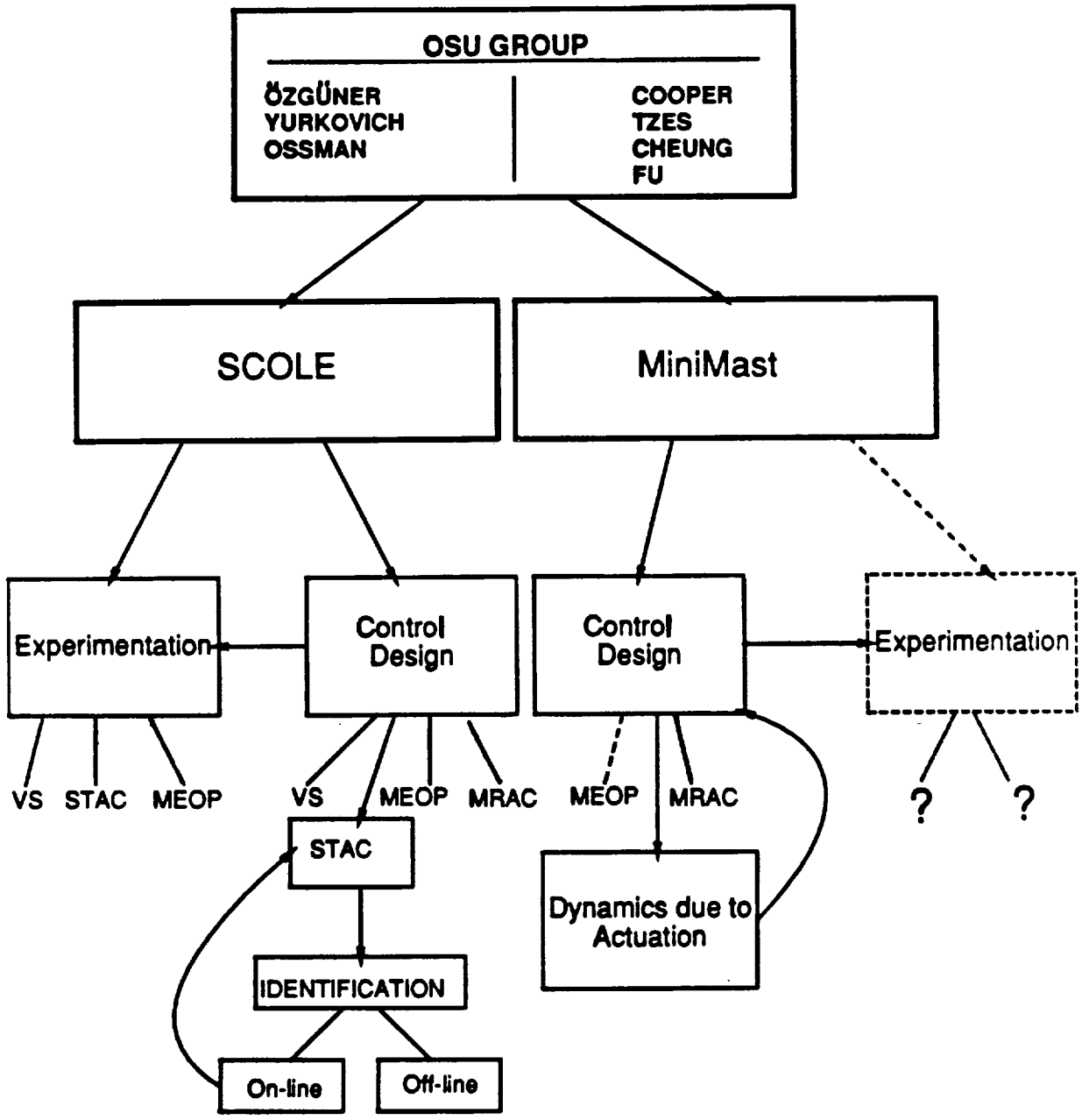


Figure 1.1: Summary of LaRC Effort

panel. The sensors include nine accelerometers and two, 3-axis rotational rate sensing units. Actuators include seven reaction wheels, two gas jets thrusters and three Control Moment Gyros (CMG), although the algorithms we currently consider do not employ the CMG's due to their availability status in the laboratory.

The model we employ in all of our studies for the SCOLE configuration is a ten mode finite element model with modal displacement equation

$$\frac{d}{dt}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -\Omega^2 & D \end{bmatrix}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0 \\ b \end{bmatrix} u \quad , \tag{2.1}$$

where

$$\Omega^2 = \begin{bmatrix} \omega_1^2 & & & \\ & \omega_2^2 & & \\ & & \ddots & \\ & & & \omega_{10}^2 \end{bmatrix} \quad D = \begin{bmatrix} -2\zeta\omega_1 & & & \\ & -2\zeta\omega_2 & & \\ & & \ddots & \\ & & & -2\zeta\omega_{10} \end{bmatrix} . \tag{2.2}$$

As noted above, inputs and outputs are selected as needed for various control algorithms and control objectives. Finally, the ten modal frequencies for this baseline model are given below in Table 2.1.

3. NONADAPTIVE CONTROL APPROACHES

Three fixed (nonadaptive) control schemes have been investigated for implementation on the SCOLE apparatus. The first two, simple rate feedback and an observer with pole placement design are relatively simple in nature, and we omit details of those algorithms here. In fact, our primary intention in considering these straightforward controller design schemes was to form a baseline comparison for more sophisticated designs to follow. In the section later in this report on experimentation we discuss the results of the rate feedback scheme and supply a few more details there.

The third nonadaptive scheme we have considered is the Maximum Entropy/Optimal Projection (MEOP) approach to designing optimal low-order controllers for high-order systems with parameter uncertainties, developed recently by Hyland and Bernstein [1], [2]. This technique presents a unique direct method to the design of

Table 2.1: Modal Frequencies

Frequency	Mode Type	Axis
.055	Pendulum	y axis
.088	Pendulum	x axis
.165	Pendulum	y axis
.251	Pendulum	x axis
.254	Pendulum	y axis
.565	Bending	
.638	Bending	
1.51	Torsional	z axis
2.94	Bending	
4.38	Bending	

optimal, robust, reduced-order compensators in that the compensator design and order reduction are performed simultaneously. This is in contrast to the conventional two–step approach to reduced order compensator design in which a compensator order reduction is preceded by the LQG design or a LQG design is preceded by plant order reduction.

The fundamental idea of the MEOP approach is to use stochastic modeling for the uncertainty in order to improve the system robustness with respect to parameter variations and higher order unmodeled dynamics. In [3] a study was conducted comparing this design philosophy with other competing asymptotic LQG–type design methods, where, for an example representative of lightly damped flexible structures, MEOP design fared favorably with regard to stability robustness to uncertainty in modal frequencies. The ME design portion for robustification is coupled with an oblique projection method to reduce the size of a compensator in some optimal fashion. Such a scheme is particularly appealing in the area of large flexible structure control where the size and robustness of the compensator are critical. Thus, the MEOP design methodology generalizes the highly sensitive classical LQG design to a robust, low-order dynamic compensator design.

3.1 MEOP Overview

As a brief introduction to the MEOP approach, consider an n^{th} order linear time invariant system

$$\dot{x} = Ax + Bu + w_1 \quad , \qquad y = Cx + w_2 \quad , \tag{3.1}$$

under the usual assumptions of controllability and observability, where w_1, w_2 are zero mean white noise processes with noise intensity matrices V_1 and V_2, respectively.

It is required to design an n_c^{th} order *robust*, zero set-point compensator

$$\dot{x_c} = A_c x + Fy \quad , \qquad u = -Kx_c \tag{3.2}$$

to minimize the cost functional

$$J = \lim_{\tau \to \infty} \frac{1}{\tau} E \int_0^{\tau} (x^T(t) R_1 x(t) + u^T(t) R_2 u(t)) dt \quad , \tag{3.3}$$

where $x \in \mathcal{R}^n$, $x_c \in \mathcal{R}^{n_c}$, $u \in \mathcal{R}^m$, $y \in \mathcal{R}^l$, $A \in \mathcal{R}^{n \times n}$, $B \in \mathcal{R}^{n \times m}$, $C \in \mathcal{R}^{l \times n}$, $A_c \in \mathcal{R}^{n_c \times n_c}$, $F \in \mathcal{R}^{n_c \times l}$, $K \in \mathcal{R}^{m \times n}$, $R_1 \in \mathcal{R}^{n \times n}$ (positive semi-definite), and $R_2 \in \mathcal{R}^{m \times m}$ (positive definite).

The first order necessary conditions for the quadratically optimal, steady state, robust, reduced-order dynamic compensation are the existence of non-negative definite matrices $P \in \mathcal{R}^{n \times n}, Q \in \mathcal{R}^{n \times n}, \hat{P} \in \mathcal{R}^{n \times n}, \hat{Q} \in \mathcal{R}^{n \times n}$ satisfying the following coupled Lyapunov and Riccati equations:

$$\begin{aligned} 0 &= PA_s + A_s^T P + \sum_{i=1}^{\mu} A_i^T P A_i - P_s^T R_{2_s}^{-1} P_s + R_1 \\ &\quad + \sum_{i=1}^{\mu} (A_i - Q_s V_{2_s}^{-1} C_i)^T \hat{P} (A_i - Q_s V_{2_s}^{-1} C_i) + \tau_\perp^T P B R_2^{-1} B^T P \tau_\perp \end{aligned} \tag{3.4}$$

$$\begin{aligned} 0 &= A_s Q + Q A_s^T + \sum_{i=1}^{\mu} A_i Q A_i^T - Q_s V_{2_s}^{-1} Q_S^T + V_1 \\ &\quad + \sum_{i=1}^{\mu} (A_i - B_i R_{2_s}^{-1} P_s) \hat{Q} (A_i - B_i R_{2_s}^{-1} P_s)^T + \tau_\perp Q C^T V_2^{-1} C Q \tau_\perp^T \end{aligned} \tag{3.5}$$

$$0 = \hat{P} A_{QS} + A_{QS}^T \hat{P} + P_s^T R_{2_s}^{-1} P_s - \tau_\perp^T P B R_2^{-1} B^T P \tau_\perp \tag{3.6}$$

$$0 = A_{PS} \hat{Q} + \hat{Q} A_{PS}^T + Q_s V_{2_s}^{-1} Q_s^T - \tau_\perp Q C^T V_2^{-1} C Q \tau_\perp^T \quad , \tag{3.7}$$

where the projection operator $\tau \in \mathcal{R}^{n \times n}$ is given by

$$\tau = \sum_{k=1}^{n_c} \prod_k [\hat{Q}\hat{P}]$$

and $\prod_k [\hat{Q}\hat{P}]$ represents the k^{th} eigenprojection of $\hat{Q}\hat{P}$, with $\tau_\perp = I_n - \tau$. In (3.4)–(3.7), $A_i \in \mathcal{R}^{n \times n}$, $B_i \in \mathcal{R}^{n \times m}$, $C_i \in \mathcal{R}^{l \times n}$ are the uncertainty matrices of the plant

[2], from which

$$A_s \equiv A + \tfrac{1}{2}\sum_{i=1}^{\mu} A_i^2 \quad , \quad B_s \equiv B + \tfrac{1}{2}\sum_{i=1}^{\mu} A_i B_i \quad , \quad C_s \equiv C + \tfrac{1}{2}\sum_{i=1}^{\mu} C_i A_i \quad ,$$

$$R_{2s} \equiv R_2 + \sum_{i=1}^{\mu} B_i^T (P + \hat{P}) B_i \quad , \quad V_{2s} \equiv V_2 + \sum_{i=1}^{\mu} C_i (Q + \hat{Q}) C_i^T \quad ,$$

$$P_S \equiv B_s^T P + \sum_{i=1}^{\mu} B_i^T (P + \hat{P}) A_i \quad , \quad Q_s \equiv Q C_s^T + \sum_{i=1}^{\mu} A_i (Q + \hat{Q}) C_i^T \quad ,$$

$$A_{QS} \equiv A_s - Q_s V_{2s}^{-1} C_s \quad , \quad A_{PS} \equiv A_s - B_s R_{2s}^{-1} P_s \quad ,$$

where μ is the number of sets of uncorrelated uncertainties.

In terms of the solution to the above MEOP equations, the compensator dynamics are specified according to

$$A_c = \Gamma(A_s - Q_s V_{2s}^{-1} C_s - B_s R_{2s}^{-1} P_s) G^T \quad , \tag{3.8}$$

$$F = \Gamma Q_s V_{2s}^{-1} \quad , \tag{3.9}$$

$$K = R_{2s}^{-1} P_s G^T \quad , \tag{3.10}$$

where the operators $G \in \mathcal{R}^{n_c \times n}, \Gamma \in \mathcal{R}^{n_c \times n}$ must satisfy $\Gamma G^T = I_{n_c}$, and $G^T \Gamma = \tau$. The complexity of the above expressions is apparent; indeed, this is a trade–off one experiences in designing with this method.

3.2 SCOLE Design

In this design and simulation study, we are interested in controlling the reflector panel, where the control objective was to damp vibrations. A full order system dynamics with three inputs and five outputs model was used for the designs to follow. The three inputs are the reaction wheel actuators located at the hub while the five outputs include three rotational rate sensors located at the hub and two accelerometers located at the center of the reflector.

The performance index to be considered for the SCOLE configuration is the infinite time state regulator problem given by:

$$J = \int_0^{\infty} [x^T(t) R_1 x(t) + u^T(t) R_2 u(t)] dt$$

where $x \in R^{20}$; $u \in R^3$; $R_1 \in R^{20 \times 20}$, non-negative definite; and $R_2 \in R^{3 \times 3}$, positive definite. The choice of state weighting matrix was made first by transforming the original system into its open loop internal balanced realization form. Then the state weighting matrix of the balanced system was taken simply equal to the diagonalized controllability or observability gramian, and the state weighting matrix for the

original system, R_1, was obtained by transforming the state weighting matrix from the balanced system. The input control weighting matrix was chosen to be diagonal and with equal weights.

A time invariant state estimator is considered here. The variance matrices at the input and output are assumed to be diagonal and stationary, and are chosen such that the poles of the estimator are faster than those in the controller.

The motivation of MEOP design is to improve the robustness of the OP design. We are interested in robustifying the modal frequencies in the model. The uncertainty matrix for the modal dynamics was taken as:

$$A_1 = \alpha \begin{bmatrix} 0 & 0 \\ -\Omega^2 & D \end{bmatrix} \quad ,$$

and the maximum entropy design parameter, α, was chosen to be 10%. The uncertainty matrices for both the input and the output were set to zero.

3.3 Results and Simulation

Both the MEOP and OP design methodologies were applied to the SCOLE configuration. Table 3.1 and Table 3.2 show the robustness and performance characteristics of the OP and MEOP designs. The second columns in each table, for the parameter $|\epsilon|$, use an improved robustness measure due to Yedavalli [4] for structured perturbations. The parameter, $|\epsilon|$, measured the guaranteed stable absolute upper bound for each component which is subjected to perturbation in the plant model dynamics. This robustness measure was applied to the closed loop dynamics, but only the submatrix, $-\Omega^2$, of the original, open loop system was assumed to be perturbed due to modal frequency variations. The corresponding perturbation on the submatrix, D, was assumed to be negligible due to the natural low damping on the structure. For instance, $|\epsilon|$ is equal to 0.0154 for the 8^{th} order MEOP design, implying that for each diagonal element in Ω^2 the maximum variation is bounded by an amount 0.0154 to ensure system stability. This corresponds to a -17 to $+15\%$ change in frequency for the first mode. The Yedavalli robustness measure is a fairly conservative one since it gives only a single absolute stability bound for all the components subjected to perturbation. However, this is used here as an indicator that the MEOP designs have improved the robustness of the OP designs.

The third columns show the stable range of the relative percentage change in all modal frequency related components in the open loop dynamics matrix from the closed loop realization. The significance of this information is in reflecting the actual robustness with respect to modal frequency variations. The final columns indicate

Table 3.1: Optimal Projection Design

Order	$\lvert\epsilon\rvert$	$\frac{d\omega}{\omega}$ (%)	Cost
20	0.0138	-20 to $+4$	0.229
12	0.0141	-30 to $+20$	0.231
10	0.0153	-45 to $+30$	0.231
8	0.0140	-9 to $+30$	0.235

Table 3.2: MEOP Designs

Order	$\lvert\epsilon\rvert$	$\frac{d\omega}{\omega}$ (%)	Cost
20	0.0148	-25 to $+40$	0.407
12	0.0156	-50 to $+50$	0.311
10	0.0154	-50 to $+50$	0.319
8	0.0154	-40 to $+40$	0.322

the steady state cost performance of each design. Clearly there is a trade off between the steady state performance and the order of the compensators. The inconsistent costs in Table 3.1 between the full order design and the reduced order designs may be accounted for by the possibility that the solutions have not yet converged in the design.

An initial condition was imposed on the SCOLE configuration for simulation purposes. Modes 1, 2, 6, 7 and 8 were initially displaced by 0.1 inch. Figure 3.1 shows the outputs for the open loop system, where y_1 and y_2 are the rate measurements from the center of the reflector while y_3 and y_4 rate measurements from the hub. The closed loop version is depicted in Figure 3.2, in which an 8^{th} order MEOP compensator design is simulated. In Figures 3.3, 3.4, and 3.5, u_1, u_2 and u_3 are the torque inputs at the hub on x, y, z axes, respectively. With the state space representation of the structure in modal coordinates, the vibrational energy of mode i is given by

$$E_i(t) = \omega_i^2 X_i^2(t) + \dot{\omega}_i^2(t) \quad , \tag{3.11}$$

where ω_i is the modal frequency of mode i and X_i is the general modal displacement associated with a given mode in a given direction.

Figure 3.6 shows the time profile of the vibration energy of the structure for LQG

and robustified LQG designs as compared with the open loop case. In Figure 3.7, the robustified reduced-order designs (10^{th} and 8^{th} order) are compared with the open loop case. For the designs indicated, all possess excellent vibration damping, since 90% of the initial vibrational energy was dissipated in the first second.

3.4 Discussion

The MEOP design approach is a very useful tool for control design, particularly for large space structures. The order of the compensator can be reduced significantly, as compared to the full order LQG design, to one implementable in typical hardware configurations without much loss in performance. The inherent robust characteristics in the compensator design are vital in applications where system parameters may vary when the structure is, for example, deployed in orbit. The issue of spillover or the robustness with respect to unmodeled dynamics is another important subject in large space structures control, and should therefore be investigated in this light.

In the aforementioned designs and simulation, all the reaction wheel actuators are assumed to have no physical constraints. From the simulations, the required control effort in the first few seconds is very near the limit of the actual hardware. Further simulations have shown, however, that when the actuator limit is observed and the control gains are constrained, vibration damping is achieved with only slight degradation in performance.

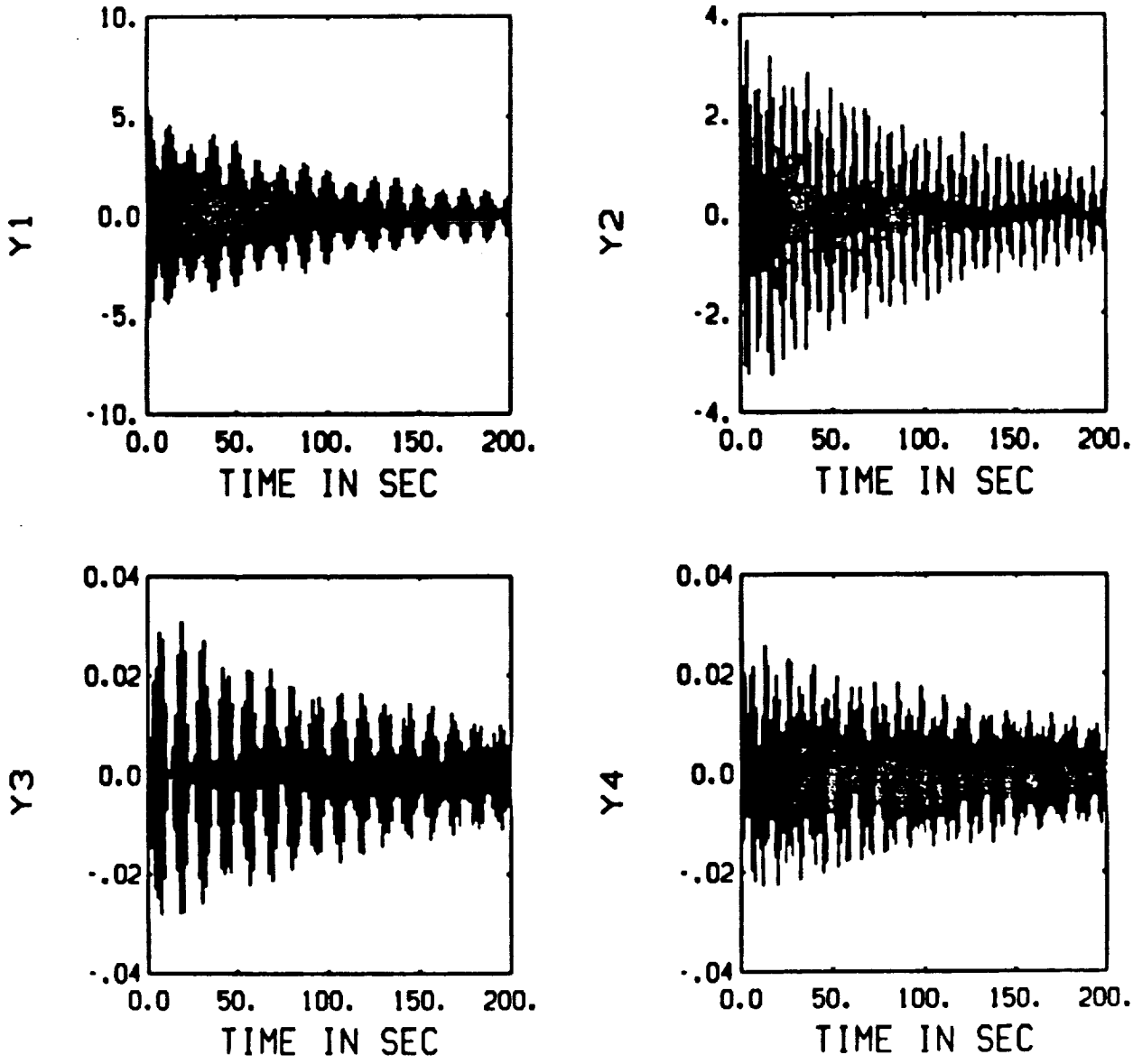


Figure 3.1: Open loop outputs

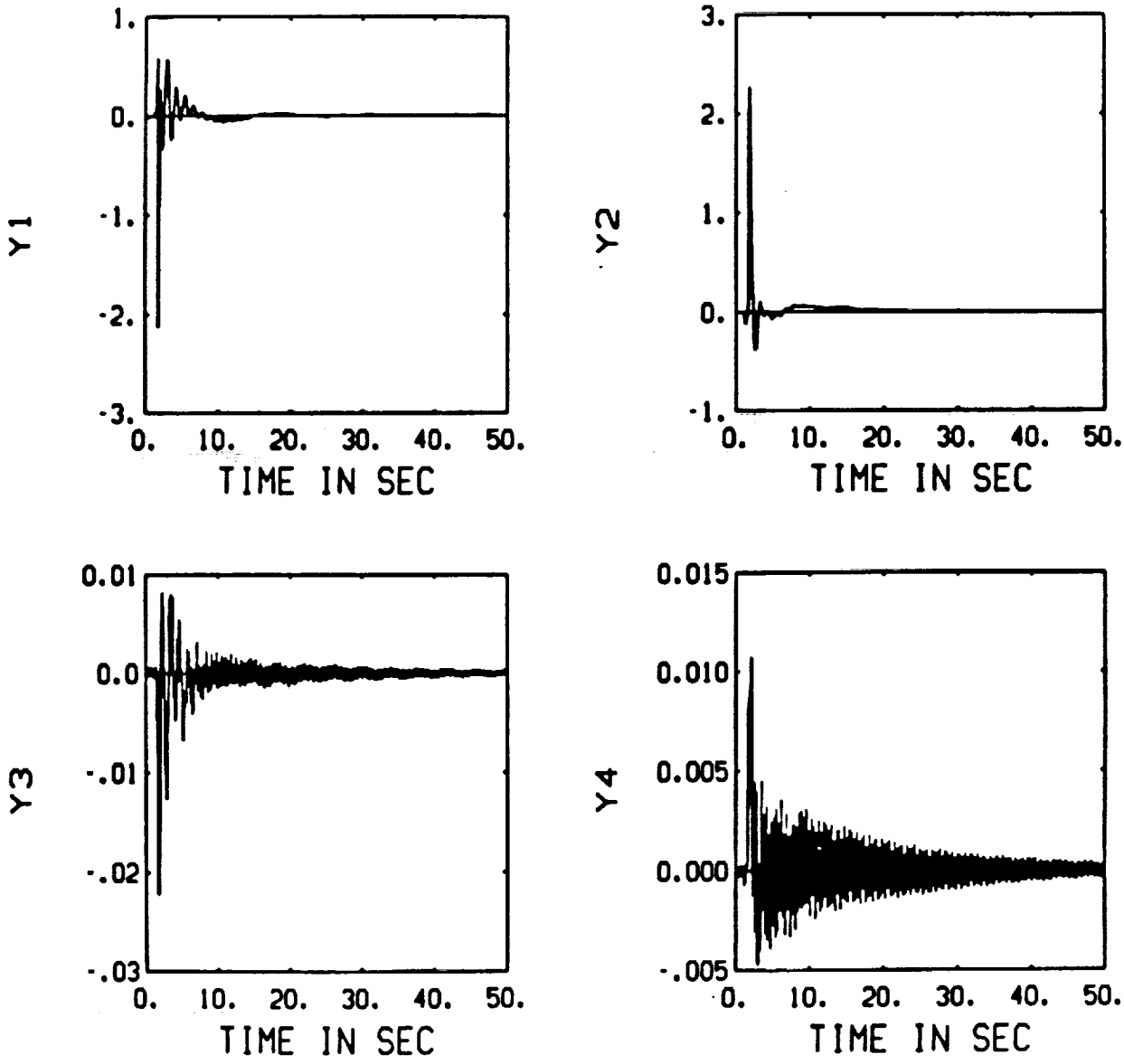


Figure 3.2: Closed loop outputs

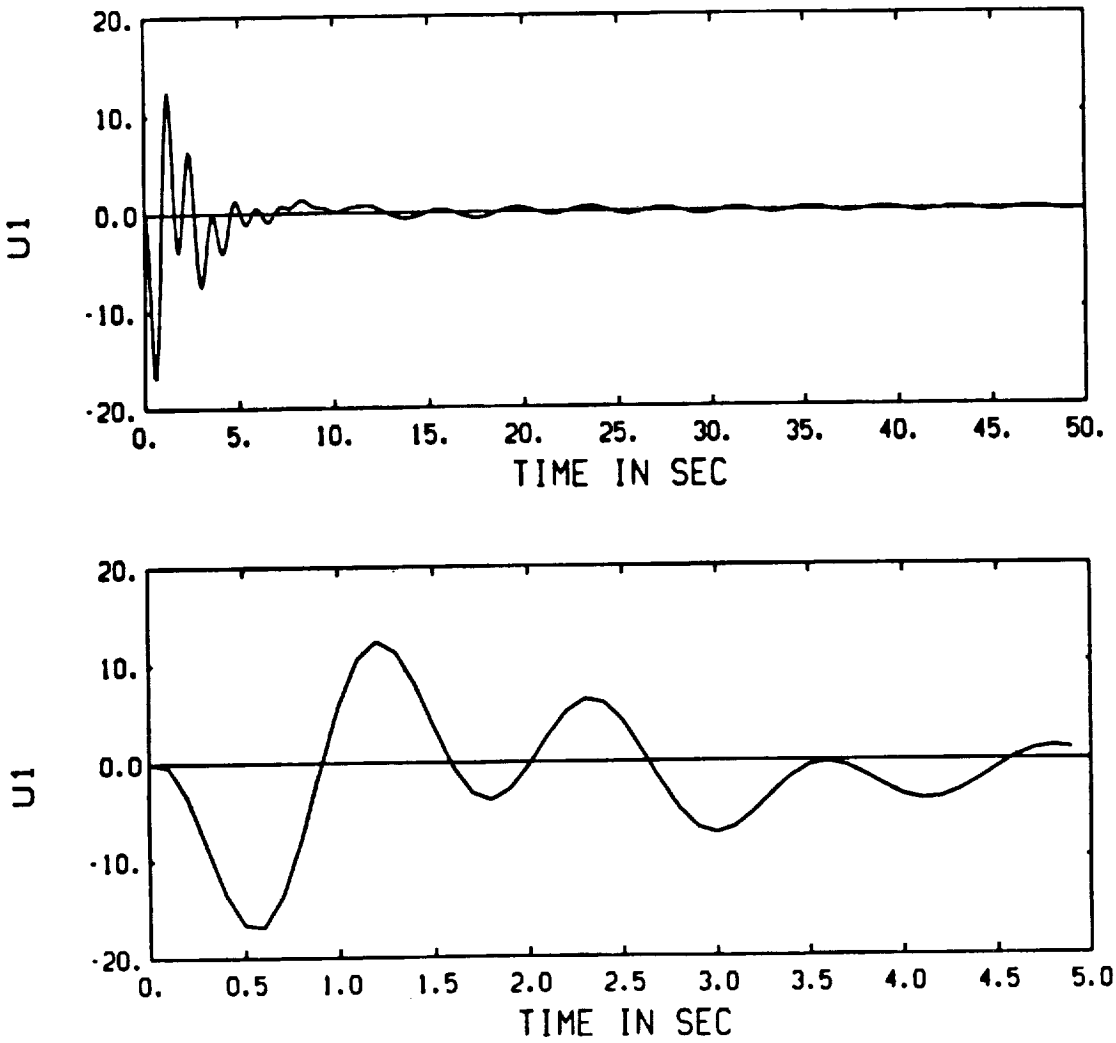


Figure 3.3: Torque input on x-axis at the hub

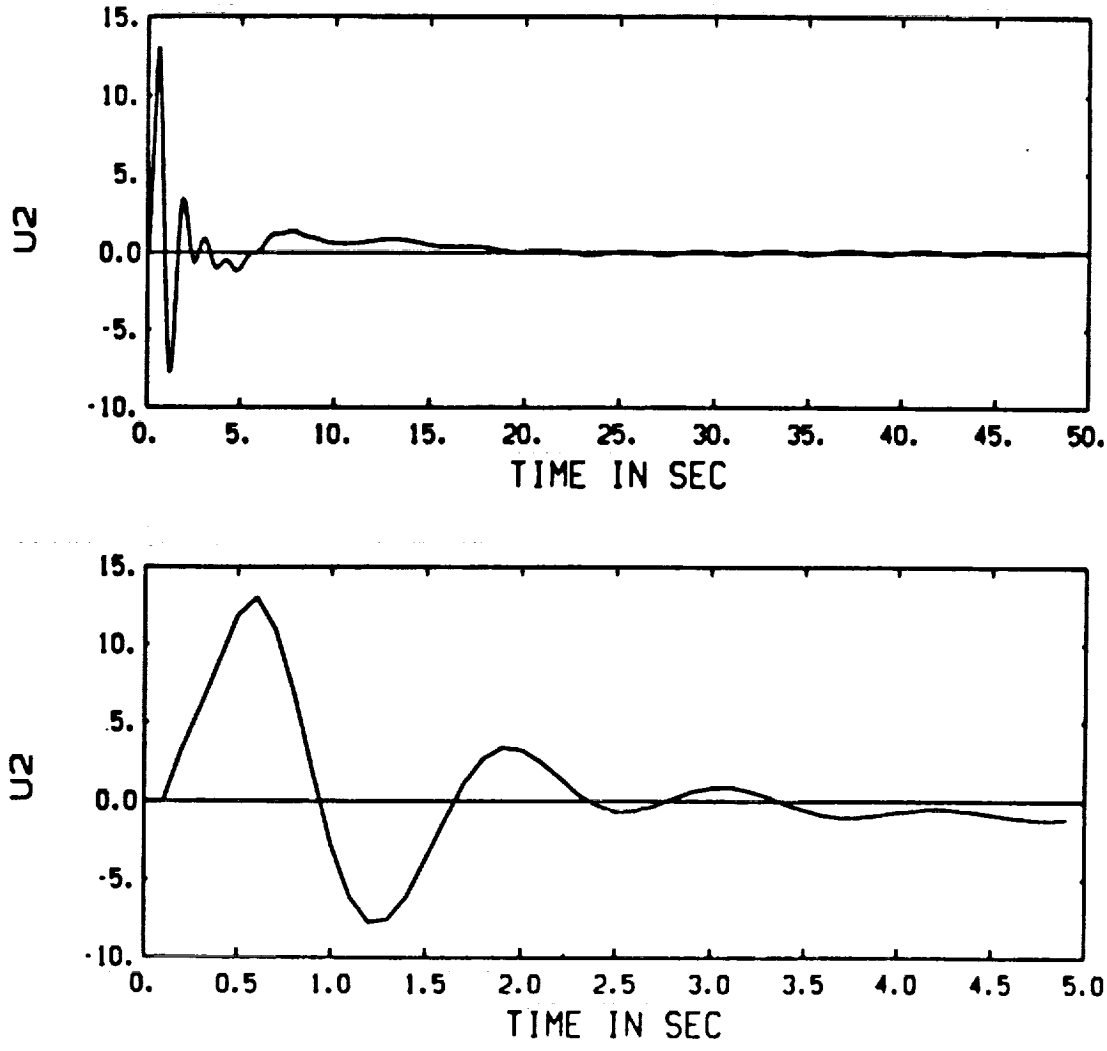


Figure 3.4: Torque input on y-axis at the hub

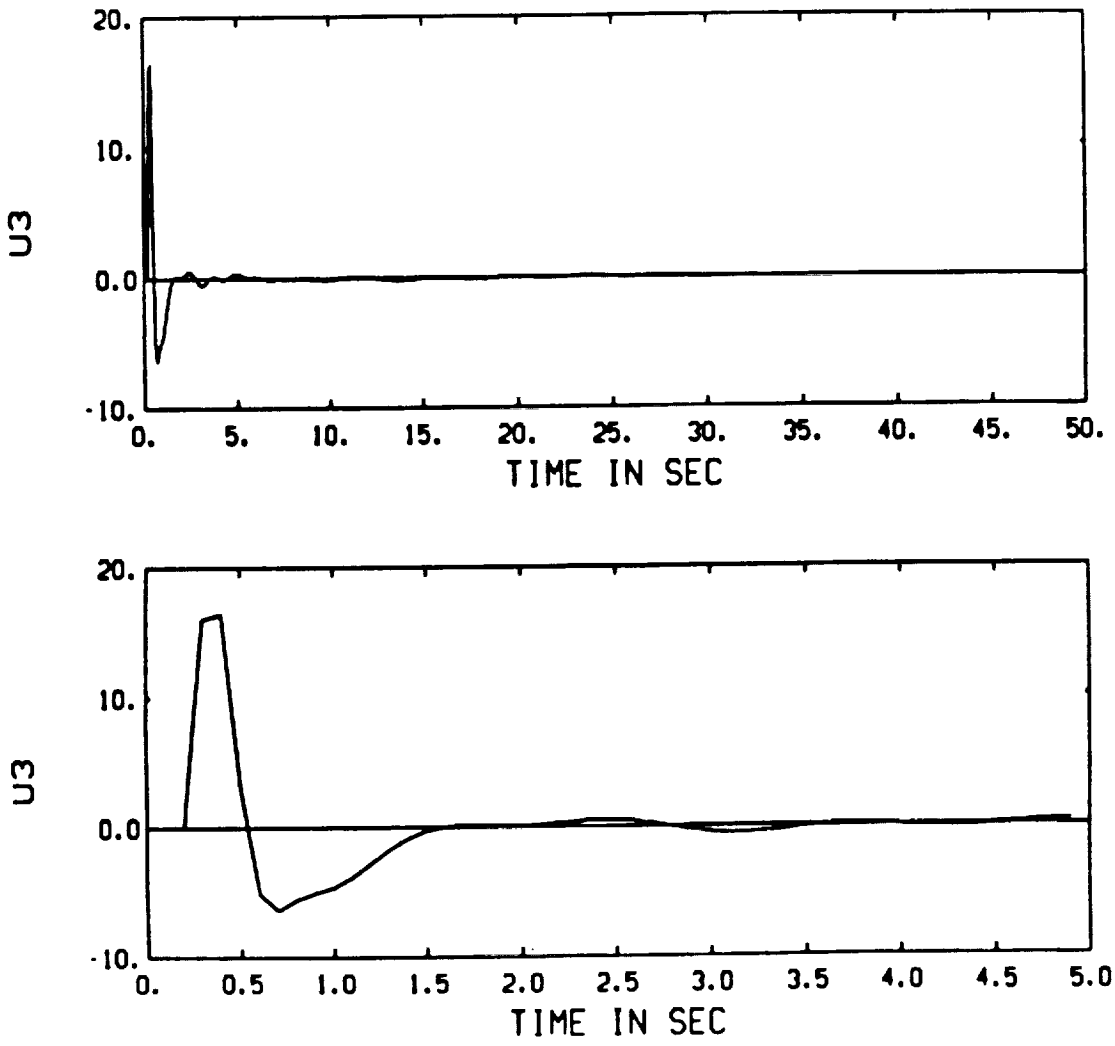


Figure 3.5: Torque input on z-axis at the hub

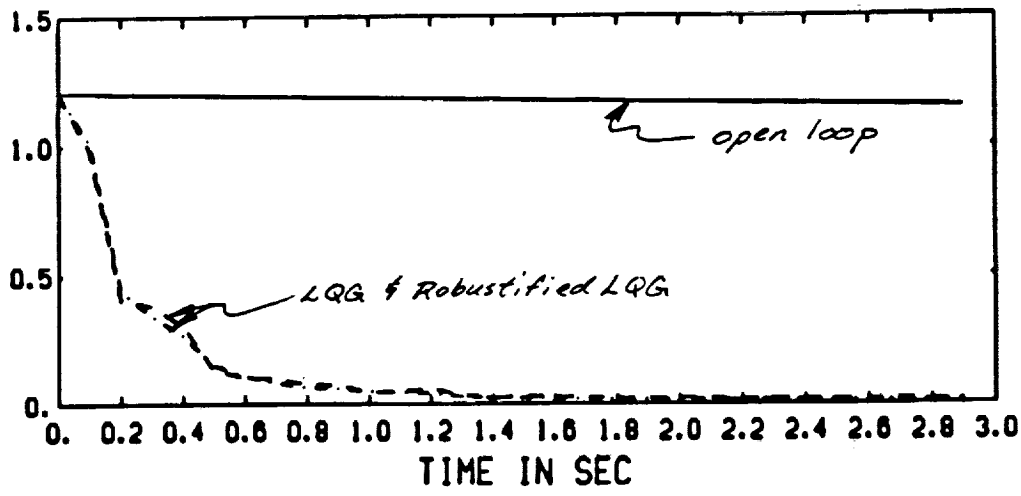


Figure 3.6: Vibrational energy profile (full–order designs)

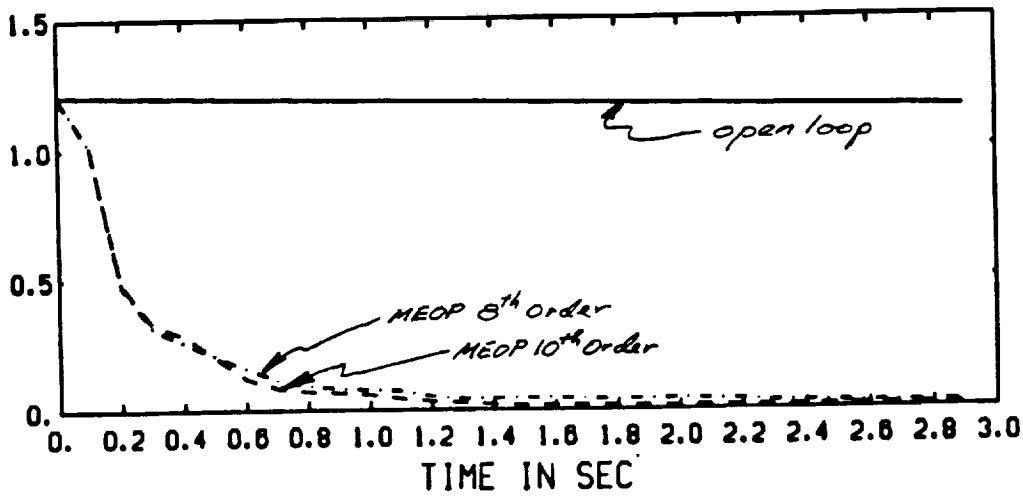


Figure 3.7: Vibrational energy profile (reduced-order designs)

4. DIRECT ADAPTIVE APPROACHES

Simply put, the model reference adaptive control approach considers a problem where specifications are given in terms of a reference model which describes how the process output should respond to a given command. Parameters of the regulator are adjusted in such a way that the error between the model output and the process output, say $y_m - y$, is made small; the difficulty arises in determining the adaptation rule. One approach is to use sensitivity–related feedback gains as a function of this error. For example, if $K(t)$ represents the vector of adjustable controller parameters, then one method for determining the adaptation is to choose the controller parameters according to the rule

$$\frac{dK}{dt} = -k(y_m - y)\nabla_K\{(y_m - y)\} \quad , \tag{4.1}$$

where k determines the rate of adaptation, and the elements of $\nabla_K\{\ \cdot\ \}$ are the sensitivity derivatives of the error with respect to $K(t)$. The justification for such a rule lies in the assumption that the adaptation parameters are slowly varying relative to other system quantities, so that the parameters are changed in the direction of the negative gradient of the error. Since the controller parameters are updated directly, such a scheme has come to be known as *direct adaptive control.*

Important issues involved in these investigations are:

1. The choice of k is not a trivial problem. If chosen small, such an adaptation scheme usually performs well, although instabilities may result with a naive selection for this adaptation parameter.
2. For complicated systems, the elements of ∇_K may not be determined exactly, thus requiring a certain degree of approximation.
3. Results are available for extension of the standard theory to nonlinear systems, multivariable systems, and nonminimum phase systems.

The presence of unmodeled dynamics becomes an important consideration in adaptive control schemes for flexible structure control. In simulation studies, therefore, we typically investigate spillover effects by designing control for reduced–order models and testing the design on full–order models with simulated disturbances and varying degrees of higher–order dynamical effects.

4.1 MRAC Schemes

To introduce the particular type of model reference adaptive schemes being pursued for this portion of the work, consider again the general model (2.1)–(2.2), rewritten in slightly different notation as

$$\dot{\tilde{x}}_p = A_p \tilde{x}_p + Bu \quad , \tag{4.2}$$

where

$$\tilde{x}_p = \begin{bmatrix} x_p \\ \dot{x}_p \end{bmatrix} \quad , \tag{4.3}$$

so that

$$\begin{bmatrix} \dot{x}_p \\ \ddot{x}_p \end{bmatrix} = \begin{bmatrix} 0 & I_{nxn} \\ -\Omega^2 & D \end{bmatrix} \begin{bmatrix} x_p \\ \dot{x}_p \end{bmatrix} + \begin{bmatrix} 0 \\ B_2 \end{bmatrix} u \tag{4.4}$$

$$y = \begin{bmatrix} 0 & C_2 \end{bmatrix} \tilde{x}_p \quad , \tag{4.5}$$

where $\tilde{x}_p \in \Re^{2n}$, $u \in \Re^3$, $y \in \Re^3$, and for our purposes here $n = 10$.

The basis of $\tilde{x}$ is chosen such that a decoupled set of second order systems or *modes* is formed. Ω^2 and D are diagonal submatrices containing the modal frequencies and damping respectively of each mode. The modal damping is chosen to be 1% for all modes.

The elements of the input–output vectors are as follows:

$$u = \begin{vmatrix} \text{y axis reaction wheel} \\ \text{x axis reaction wheel} \\ \text{z axis reaction wheel} \end{vmatrix}$$

$$y = \begin{vmatrix} \text{y axis rate gyro} \\ \text{x axis rate gyro} \\ \text{z axis rate gyro} \end{vmatrix}$$

The input - output devices are physically colocated on the structure itself, resulting in

$$C_2 = B_2^T \quad . \tag{4.6}$$

Table 4.1: Modal Frequencies Used for Control Law Design

Frequency	Mode Type	Axis
1.51	Torsional	z axis
2.94	Bending	
4.38	Bending	

Remarks

The control laws developed in the following subsections utilize only 3 modes of the full order model chosen via a balanced realization analysis, with one input per mode required. The modes upon which the designs are formulated appear in Table 4.1 The reduced order model is identical in structure to (4.4)–(4.5), except that notationally we shall replace x_p and $\tilde{x}_p$ by x and $\tilde{x}$, respectively, where now $\tilde{x} \in \Re^{2n}$, $u \in \Re^n$, $y \in \Re^n$, and $n = 3$, and where now Ω^2 and D are 3×3 matrices. Further, an observer is required to generate the state estimates for the state based MRAC law. The implementation details will be omited here for brevity; the equations developed in the following subsections assume that the states are directly measured.

4.1.1 Design

The MRAC law utilized in the designs requires a model for generation of a state error vector e. The model is as follows:

$$\dot{\tilde{x}}_m = A_m \tilde{x}_m + Bu \quad , \tag{4.7}$$

with

$$\tilde{x}_m = \begin{bmatrix} x_m \\ \dot{x}_m \end{bmatrix} \quad , \tag{4.8}$$

so that

$$\begin{bmatrix} \dot{x}_m \\ \ddot{x}_m \end{bmatrix} = \begin{bmatrix} 0 & I_{n\times n} \\ \Omega^2 & D_m \end{bmatrix} \begin{bmatrix} x_m \\ \dot{x}_m \end{bmatrix} + \begin{bmatrix} 0 \\ B_2 \end{bmatrix} u_r \quad , \tag{4.9}$$

where $\tilde{x}_m \in \Re^{2n}$, $u \in \Re^3$, and $n = 3$. The modal frequencies are those chosen for the reduced order plant given in table 4.1, and the damping for all modes is chosen to be 5%.

The basic assumptions for the adaptive control law implementation are:

- The influence matrix B_2 of the plant is not time varying. This allows the construction of an *adjustable* system or combination controller/plant where the parameters can be varied.
- Only vibration suppression control is considered. Therefore, the reference input u_r is always zero.

The error between the model states and the plant states is

$$\tilde{e} = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \tag{4.10}$$

$$\tilde{e} = \tilde{x}_m - \tilde{x} \tag{4.11}$$

The generic control law from Hyperstability theory for this formulation is given according to

$$u = \int \phi_1 \tilde{x} dt + \phi_2 \tilde{x} \quad . \tag{4.12}$$

The control elements ϕ can be chosen from the large class of hyperstable systems. For our implementation, ϕ is chosen for the the class of Proportional–Integral (PI) control laws and Relay–Integral (RI) control laws. Thus, for these classes ϕ is defined as

$$\phi_1 = F_a v[G_a \tilde{x}]^T \tag{4.13}$$

$$\phi_2 = F_a' v[G_a' \tilde{x}]^T \quad , \tag{4.14}$$

where, for $v \in \Re^6$,

$$v = W\tilde{e} \tag{4.15}$$

and where F_a,F_a',G_a, and G_a' are constant positive definite matrices.

Decoupling PI Implementation

For the decoupling PI design F_a and F_a' are chosen as

$$F = \begin{bmatrix} F_a & F_a' \end{bmatrix} \tag{4.16}$$

$$= \begin{bmatrix} 0_{n\times n} & B_2^{-1} \end{bmatrix} \quad , \tag{4.17}$$

and G_a, and $G_a^{'}$ are chosen such that

$$G_a = \alpha I_{2n\times 2n} \quad , \tag{4.18}$$
$$G_a^{'} = \alpha^{'} I_{2n\times 2n} \quad , \tag{4.19}$$

where α and $\alpha^{'}$ are the integral and proportional constants, respectively. Finally, the error transformation matrix W is chosen as $W = I_{2n\times 2n}$.

Given the above definitions, the resulting plant-controller system is *decoupled*, so that in fact

$$\ddot{x}_i = 2w_i(d_i)\dot{x}_i + w_i^2 x_i + \alpha \int_0^t \dot{e}_i \|\tilde{x}\|^2 dt + \alpha^{'}\dot{e}_i\|\tilde{x}\|^2 \quad , \tag{4.20}$$

where x_i and e_i are individual elements from the vectors $\tilde{x}$ and $\tilde{e}$, and w_i and d_i are the natural frequency and damping, respectively, for each mode taken from Ω^2 and D. We see that each mode of the system is dependent on the corresponding individual error vector and the Euclidian norm of the state vector.

Since we have achieved a decoupled modal control, it would be advantageous to apply greater control energy to the slow modes or those modes which are "most observable" in the output space. This can be accomplished by altering the choice of F_a and $F_a^{'}$ to

$$F = \left[\, 0_{n\times n} \quad B_2^{-1}\Gamma \,\right] \tag{4.21}$$

where Γ is a diagonal matrix of modal weighting elements. The resulting decoupled system with modal weighting is, therefore,

$$\ddot{x}_i = 2w_i(d_i)\dot{x}_i + w_i^2 x_i + \gamma_i\alpha \int_0^t \dot{e}_i \|\tilde{x}\|^2 dt + \gamma_i\alpha^{'}\dot{e}_i\|\tilde{x}\|^2 \quad , \tag{4.22}$$

where γ_i represents the approiate (diagonal) element from the weighting matrix Γ.

Simulation studies for this controller design consist of time plots for the input and output vectors. The simulations were performed by initializing all the modes with a value representing typical output vibrations. For purposes of comparison with the simulation results for controller designs in this section and those to follow (all MRAC approaches), Figure 4.1 gives the *free response* (no control) as simulated for the three rate gyro outputs. Figure 4.2 depicts the simulated rate gyro responses for the three axes with the MRAC PI *decoupled* algorithm, while Figure 4.3 depicts the corresponding input torques of the reaction wheel actuators. We note that the control was activated at time $t = 1.5$ seconds.

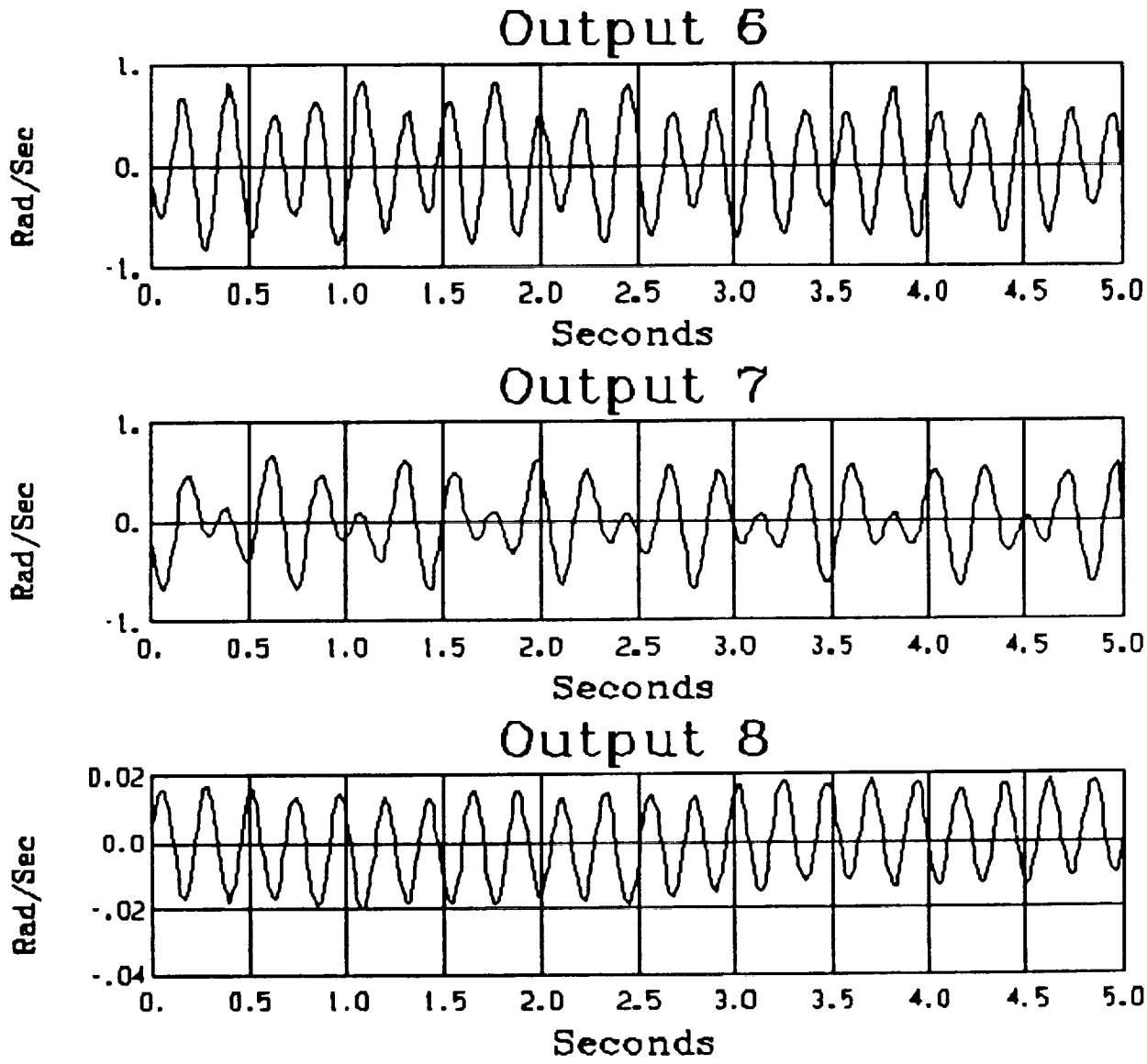


Figure 4.1: Rate gyro outputs: Free Response

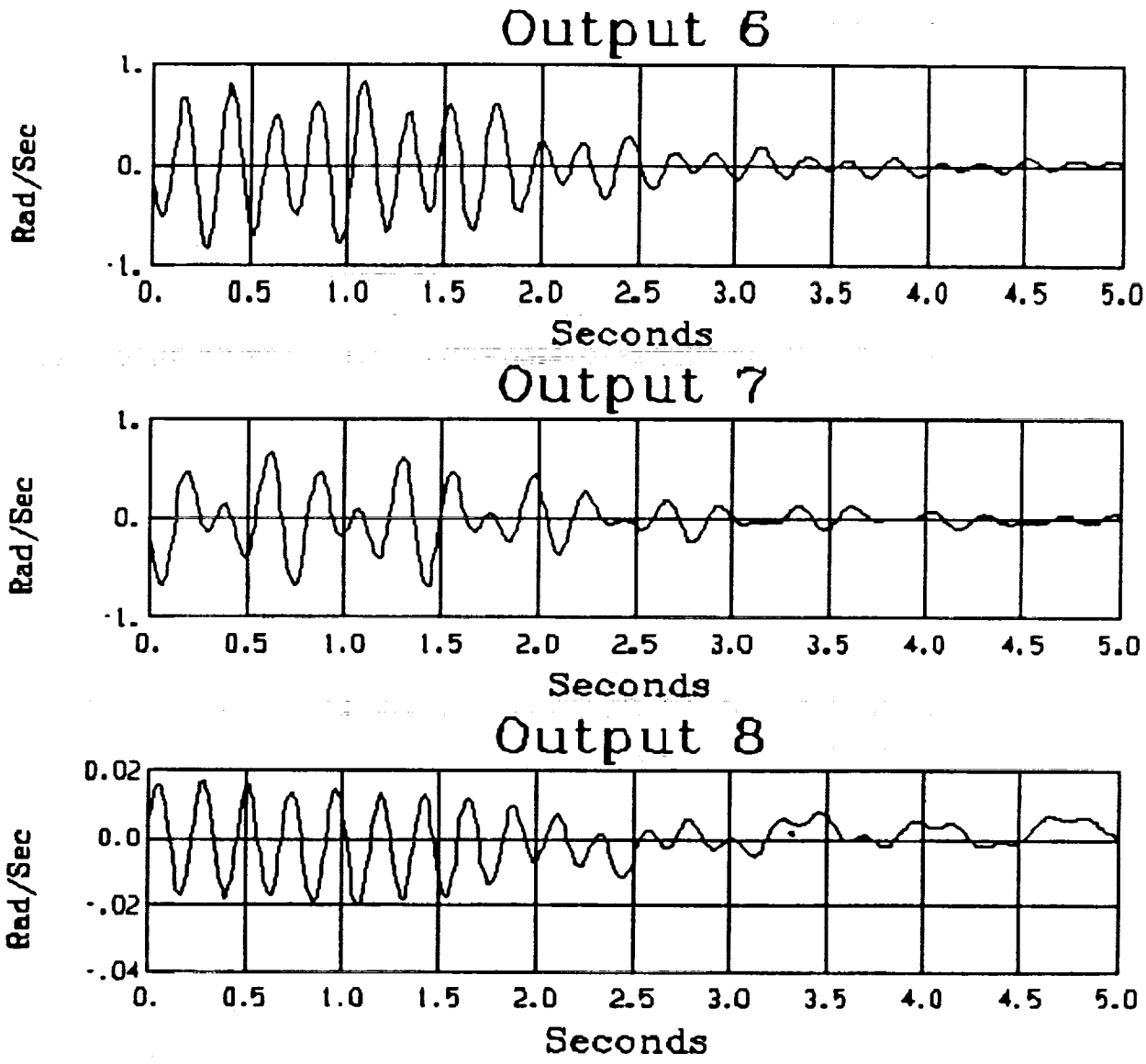


Figure 4.2: Rate gyro outputs: MRAC PI decoupled

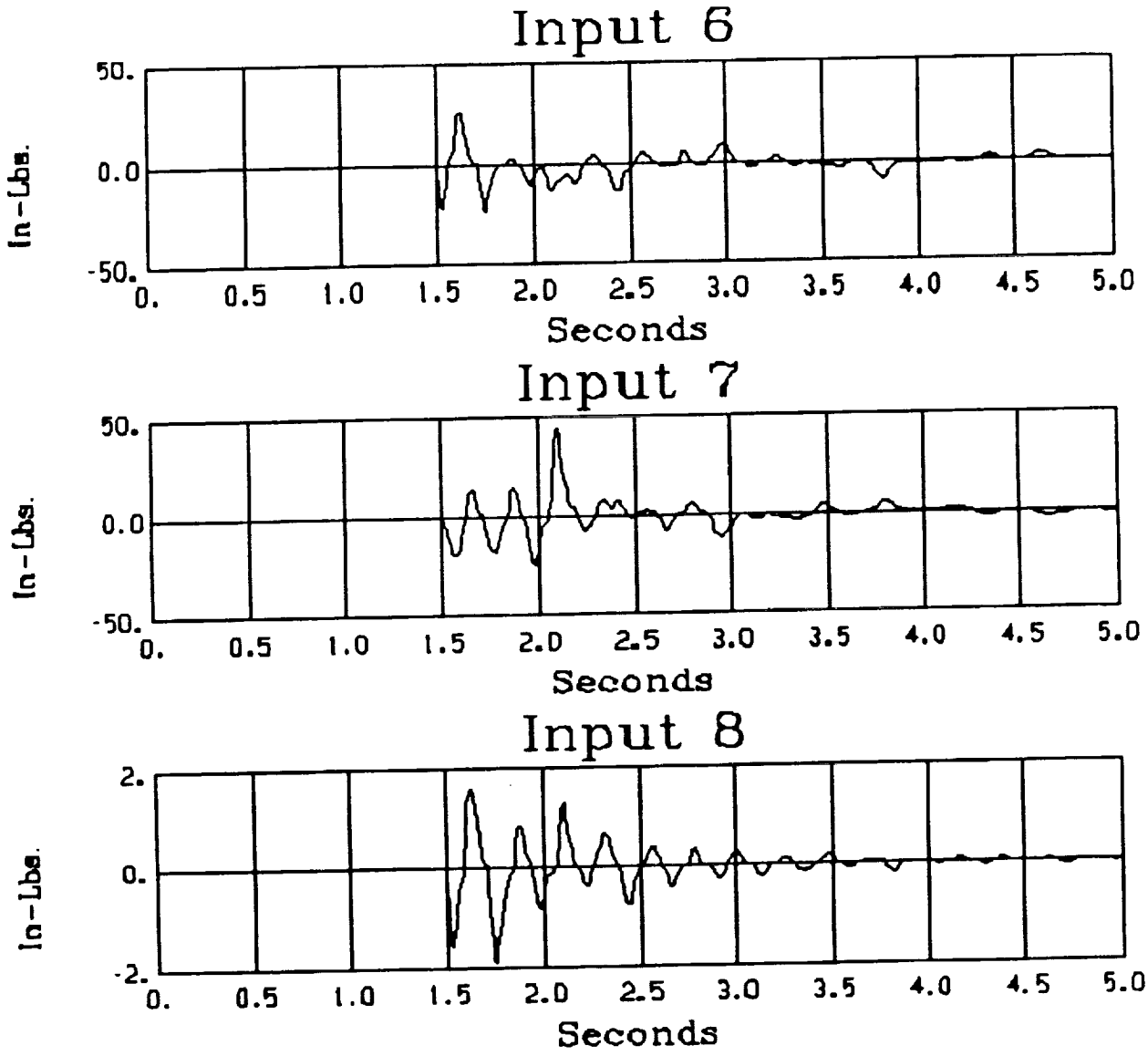


Figure 4.3: Reaction wheel inputs: MRAC PI decoupled

Coupled PI Implementation

The second variation on the MRAC design approach we have taken involves a proportional–integral type control law without the decoupling structure of the preceding case. In this case one choice of F_a and $F_a^{'}$ for an input–output coupled control is

$$F = \left[\begin{array}{cc} 0_{n\times n} & B_2^T \end{array} \right] \tag{4.23}$$

with G_a, $G_a^{'}$ and W chosen as in the previous design. The resulting control law is given by

$$u = F\alpha \int_0^t \tilde{e}\|\tilde{x}\|^2 dt \quad + \quad F\alpha^{'}\tilde{e}\|\tilde{x}\|^2 \tag{4.24}$$

Thus, because of the structure of F the controller–plant equation can be written as

$$\ddot{x} = W_{diag}x + Dp_{diag}\dot{x} + B_2B_2^T\alpha \int_0^t \dot{e}\|\tilde{x}\|^2 dt + B_2B_2^T\alpha^{'}\dot{e}\|\tilde{x}\|^2 \quad . \tag{4.25}$$

We have found that this non–decoupled implementation provides superior performance over the decoupled implementation for situations in which sensors/actuator pairs are not physically colocated. Figures 4.4 and 4.5 represent the output and input responses, respectively, for the MRAC PI non–decoupled algorithm.

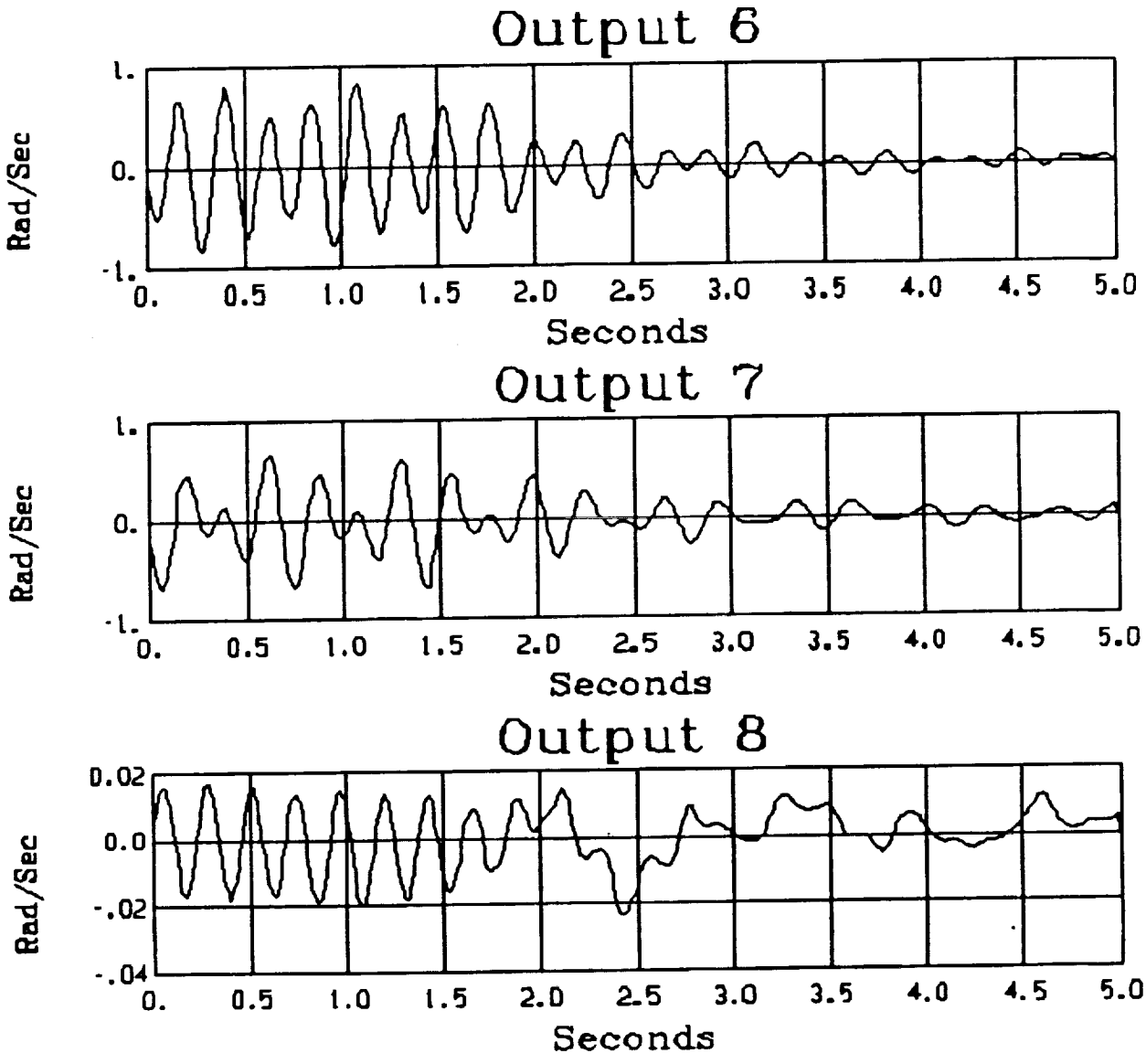


Figure 4.4: Rate gyro outputs: MRAC PI non-decoupled

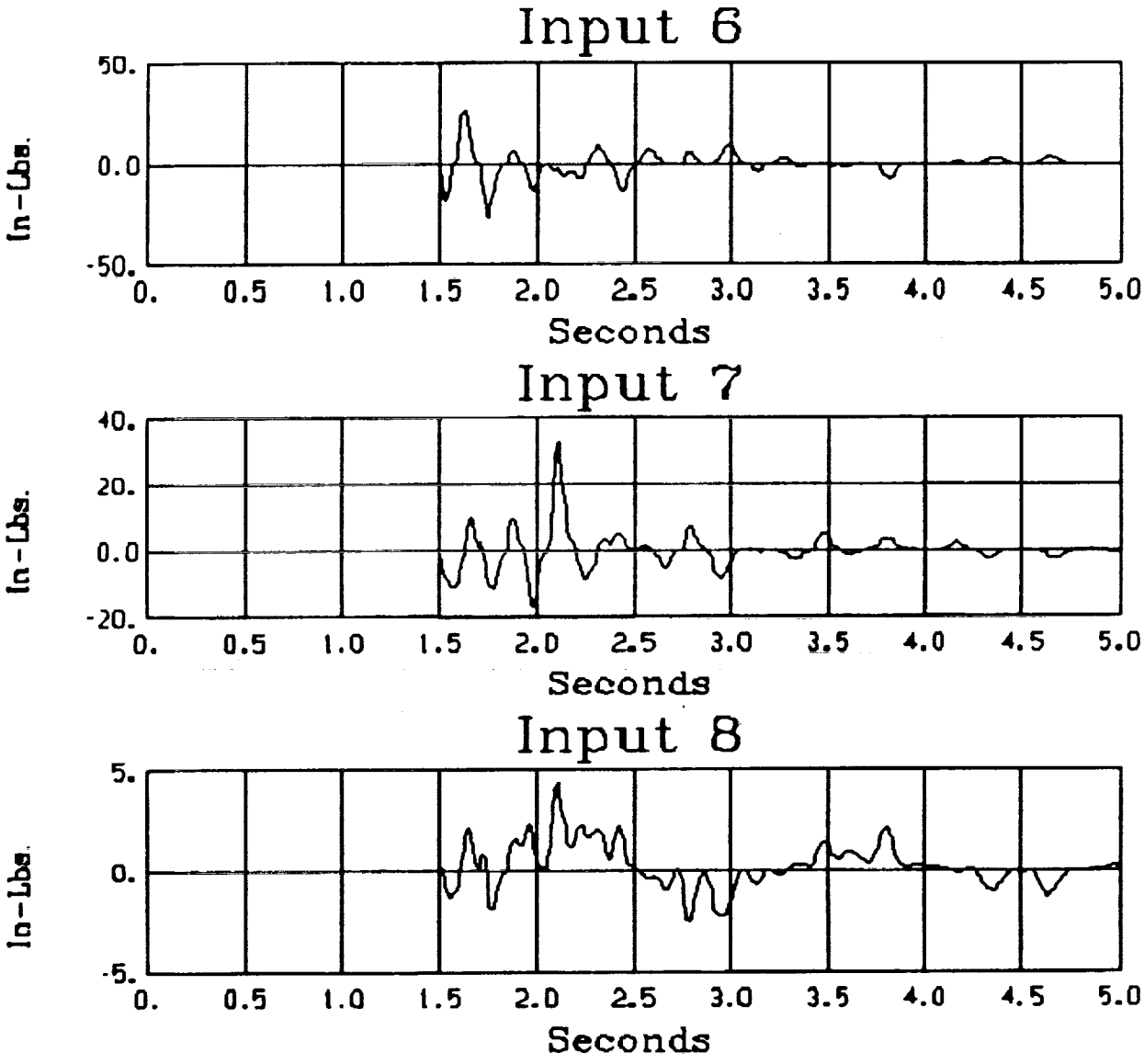


Figure 4.5: Reaction wheel inputs: MRAC PI non-decoupled

RI implementation

Development for the Relay–Integral control is essentially the same as described for the PI control scheme, with the difference being that the sign of the error vector is utilized rather than the error vector itself in the proportional part of the control law. Since the decoupled control is useful in the SCOLE context, we refer to the development of equations specified for the decoupled PI control for our implementation. The final decoupled RI control law, then, is

$$\ddot{x}_i = 2w_i(d_i)\,\dot{x}_i + w_i^2 x_i + \alpha \int_0^t \dot{e}_i \|\tilde{x}\|^2 dt + \alpha' sgn[\dot{e}_i]\|\tilde{x}\|^2 \quad . \tag{4.26}$$

Simulations were performed by initializing all the modes with a value representing typical output vibrations. Figures 4.6 and 4.7 are the input and output responses, respectively, with the MRAC RI algorithm.

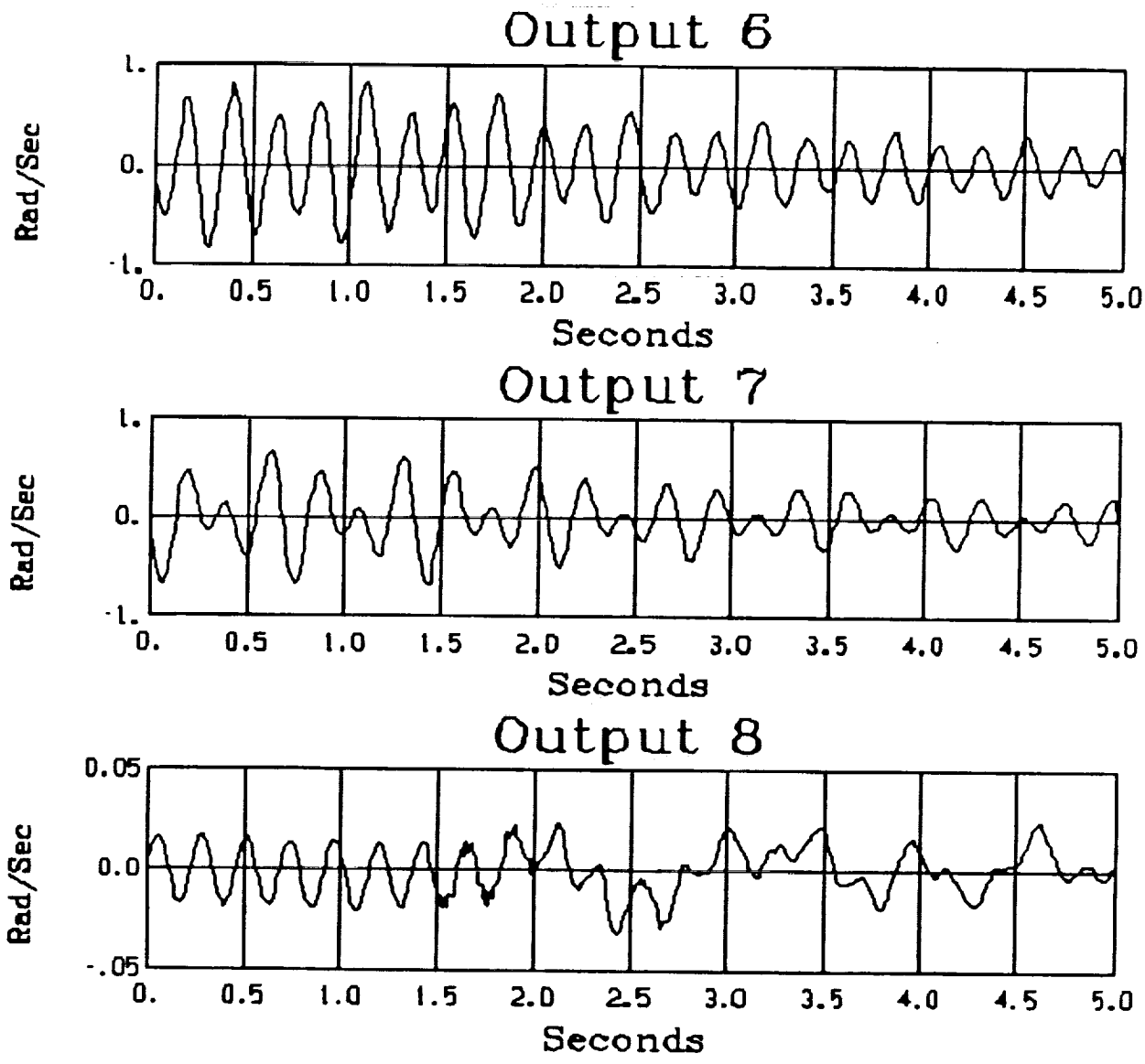


Figure 4.6: Rate gyro outputs: MRAC RI

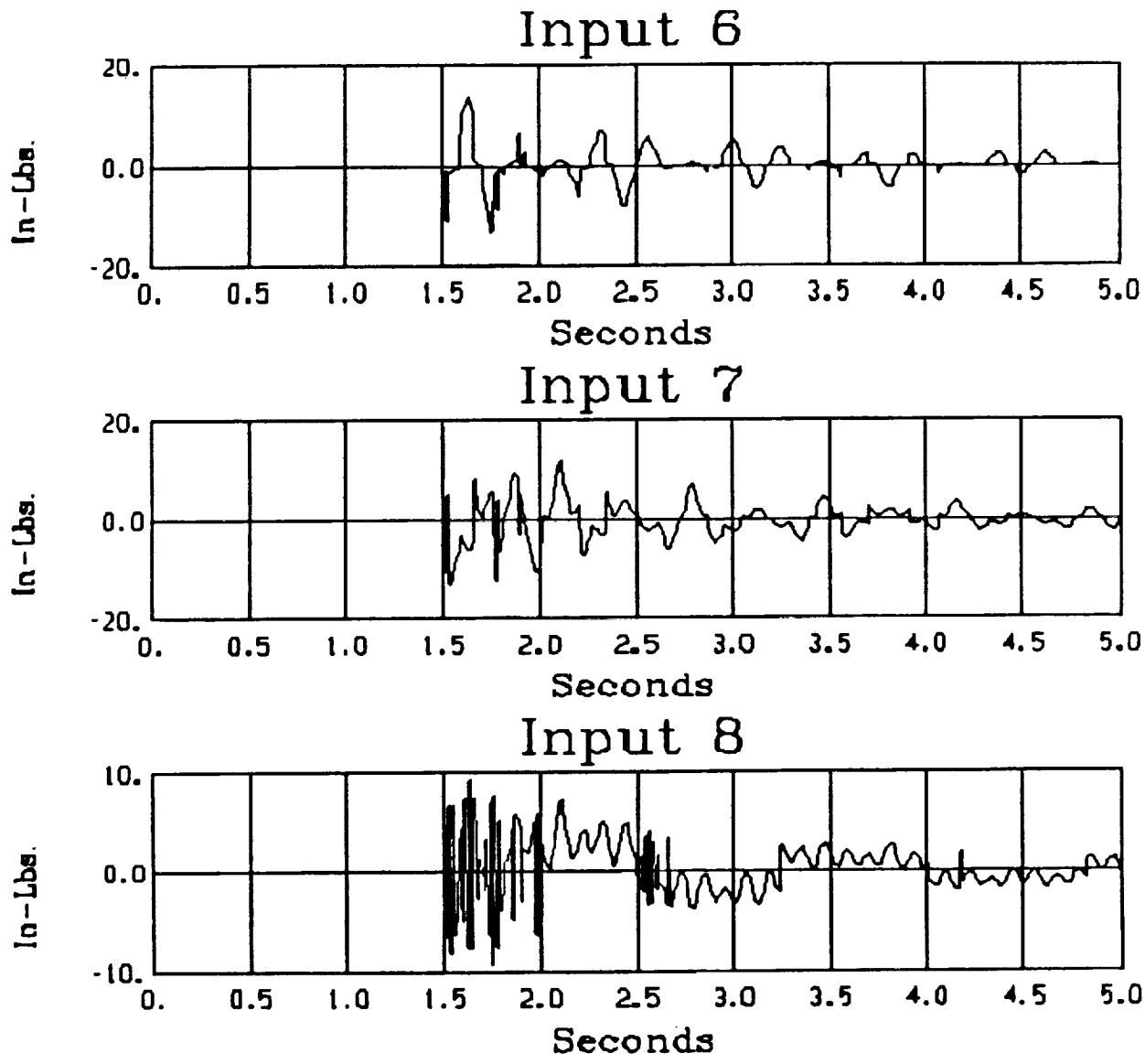


Figure 4.7: Reaction wheel inputs: MRAC RI

4.2 Adaptive Model Following VS Control

The next adaptive technique considered is adaptive model following in which the position and velocity outputs are forced via variable structure control to track given reference position and velocity paths. This technique for adaptive model following was first introduced by Young [5], and employed for flexible structure control by Özgüner and Yurkovich [6].

A variable structure controller drives the state trajectory to a chosen surface in the state space then forces the trajectory to slide along this surface to the origin. This is accomplished by switching the feedback gains between sets of feasible values whenever the state trajectory crosses the hypersurface in the state space. Variable structure control is robust to variations in parameters when in the sliding mode; however, the trajectories may "chatter" while sliding to the origin along the surface.

As mentioned above, in model reference control the objective is to force the plant output to track the output of some pre-specified reference model. One way to accomplish this is to switch the feedback gains to force the tracking error to converge to zero along a desired sliding surface. A block diagram of the model following variable structure controller is shown in Figure 4.2.

4.2.1 Design

Consider again the model for the SCOLE apparatus given in (2.1)–(2.2). For purposes in this design we assume velocity measurements (colocation) given by

$$\dot{y} = b^T \dot{q} \quad . \tag{4.27}$$

Note that y and $\dot{y}$ can be expressed as

$$\frac{d}{dt}\begin{bmatrix} y \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} y \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ -b^T\Omega^2 q \end{bmatrix} + \begin{bmatrix} 0 \\ b^T b \end{bmatrix} u \quad . \tag{4.28}$$

The desired position trajectory θ and velocity trajectory $\dot{\theta}$ are generated from the reference model

$$\frac{d}{dt}\begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ a_1 & a_2 \end{bmatrix}\begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \quad , \tag{4.29}$$

where a_1 and a_2 are design parameters. The tracking error vector is defined as

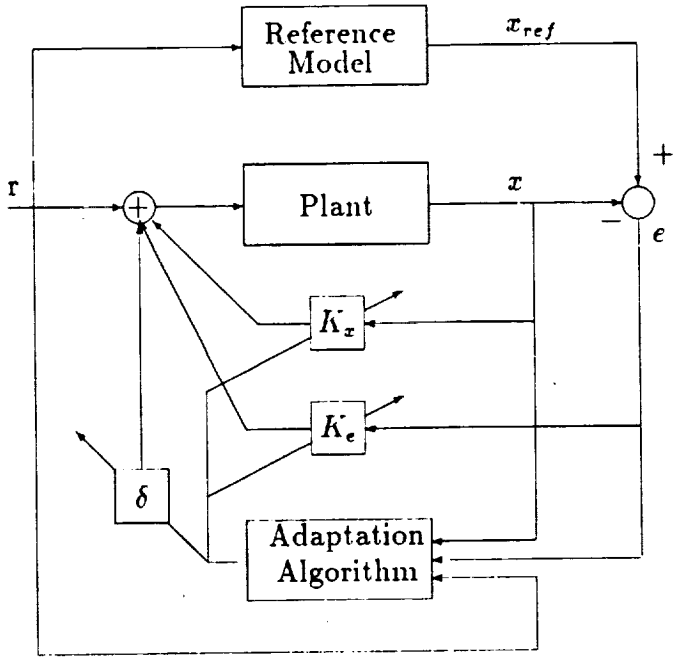


Figure 4.8: Model Following Variable Structure Control

$$e \triangleq \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} - \begin{bmatrix} y \\ \dot{y} \end{bmatrix}. \tag{4.30}$$

The feedback control law is given by

$$u = K_e e + K_1 \int \dot{y}\, dt + K_2 \dot{y} + \delta \quad , \tag{4.31}$$

where K_e, K_1, K_2, and δ are feedback switching gains chosen to force the position and velocity error to converge to zero along the switching line

$$\sigma = c^T e = c_1 e_1 + c_2 e_2 = 0 \quad . \tag{4.32}$$

It was shown in [6] that the tracking error would converge to zero along the given switching line if the feedback gains are chosen according to the following switching rules (assuming $c_2 > 0$):

$$e_1 \sigma > 0 \Longrightarrow K_e^1 > a_1/(b^T b) \quad ; \quad e_1 \sigma < 0 \Longrightarrow K_e^1 < a_1/(b^T b) \quad ;$$

$$e_2\sigma > 0 \Longrightarrow K_e^2 > (c_1 + c_2a_2)/c_2(b^Tb) \quad ; \quad e_2\sigma < 0 \Longrightarrow K_e^2 < (c_1 + c_2a_2)/c_2(b^Tb) \quad ;$$

$$y\sigma > 0 \Longrightarrow K_1 > a_1/(b^Tb) \quad ; \quad y\sigma < 0 \Longrightarrow K_1 < a_1/(b^Tb) \quad ;$$

$$\dot{y}\sigma > 0 \Longrightarrow K_2 > a_2/(b^Tb) \quad ; \quad \dot{y}\sigma < 0 \Longrightarrow K_2 < a_2/(b^Tb) \quad ;$$

$$\sigma > 0 \Longrightarrow \delta > (b^T\Omega^2q)/(b^Tb) \quad ; \quad \sigma < 0 \Longrightarrow \delta < (b^T\Omega^2q)/(b^Tb) \quad .$$

4.2.2 Simulation Results

The adaptive model following variable structure controller was applied to the reflector portion of the SCOLE configuration in simulation studies, with the control objective to damp vibrations. Using balanced realization model reduction techniques, a two mode (four state) model was developed for each sensor/actuator pair. There are five such pairs for the reflector: the accelerometers and cold gas jets (x, y directions) mounted in the center of the reflector panel and the rate sensor and reaction wheels mounted at the hub (x, y, z directions). The reference model was chosen to be second order with a natural frequency $\omega_n = 1$ rad/sec and a damping ratio $\zeta = 0.1$.

Figure 4.9 shows the control input (unconstrained) for the cold gas jet in the x–direction, the desired position and velocity trajectories generated from the reference model, and the true position and velocity trajectories. Figure 4.10 shows the control input for the cold gas jet and the true position and velocity trajectories when the physical constraints on the jet are taken into account. In both cases, "chattering" is apparent in the true velocity trajectory which is characteristic of variable structure control. As expected, the response is degraded somewhat when physical constraints are introduced.

Figure 4.11 shows the control input (unconstrained) for the reaction wheel in the y–direction, the reference velocity and position trajectories, and the measured position and velocity. Figur 4.12 shows the control input and the true position and velocity when the physical constraints on the reaction wheel are considered.

4.2.3 Discussion

These simulation studies have indicated several avenues to pursue in future studies for the application of switching–type control for flexible structures. First, we have seen here and in other studies ([6]) that this technique supplies the desired amount

of damping as specified by the reference model. The issue of control spillover remains an important question, however, since the chattering effect may introduce undesirable effects, depending on the type of actuation hardware used. To remedy this situation where needed, *time varying* sliding surfaces [7] or *sliding regions* may be utilized. Secondly, the actual equipment limitations, taken into account in the simulation studies here in terms of amplitude limits, will effect performance. Allowable switching speeds must also be considered; indeed, the reflector jets on SCOLE respond at frequencies up to about 40 or 50 Hz.

The simulation example of this report has indicated that the model following scheme for flexible structure vibration control shows promise and deserves further investigation. We believe, however, that the true power of the variable structure control approach lies in its application to nonlinear plants; many works to date have illustrated this point. For this reason, then, investigations focusing on the application to nonlinear slew maneuvers in, for example, configurations such as SCOLE are warranted.

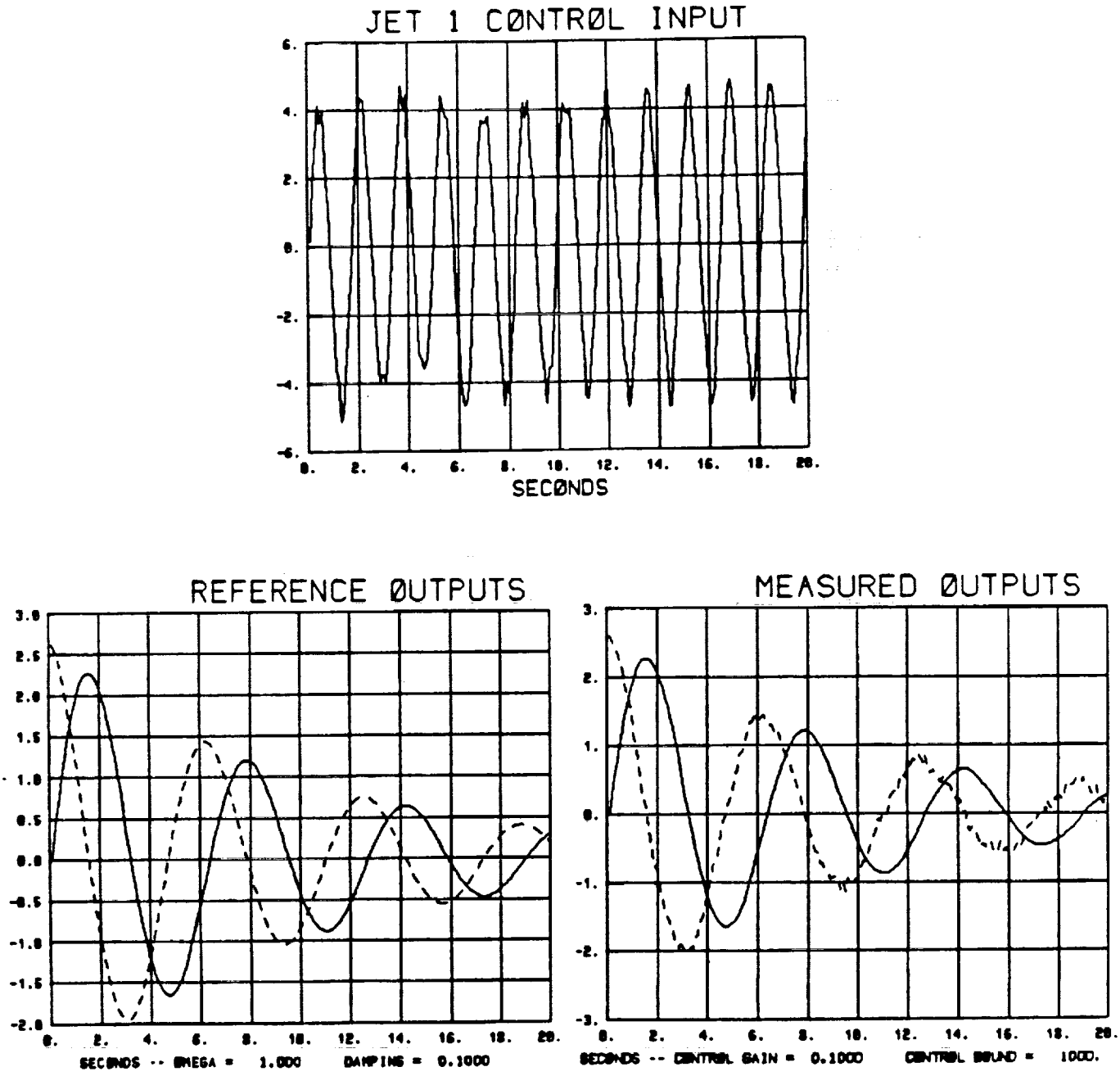


Figure 4.9: Jet input (unconstrained), reference and measured outputs

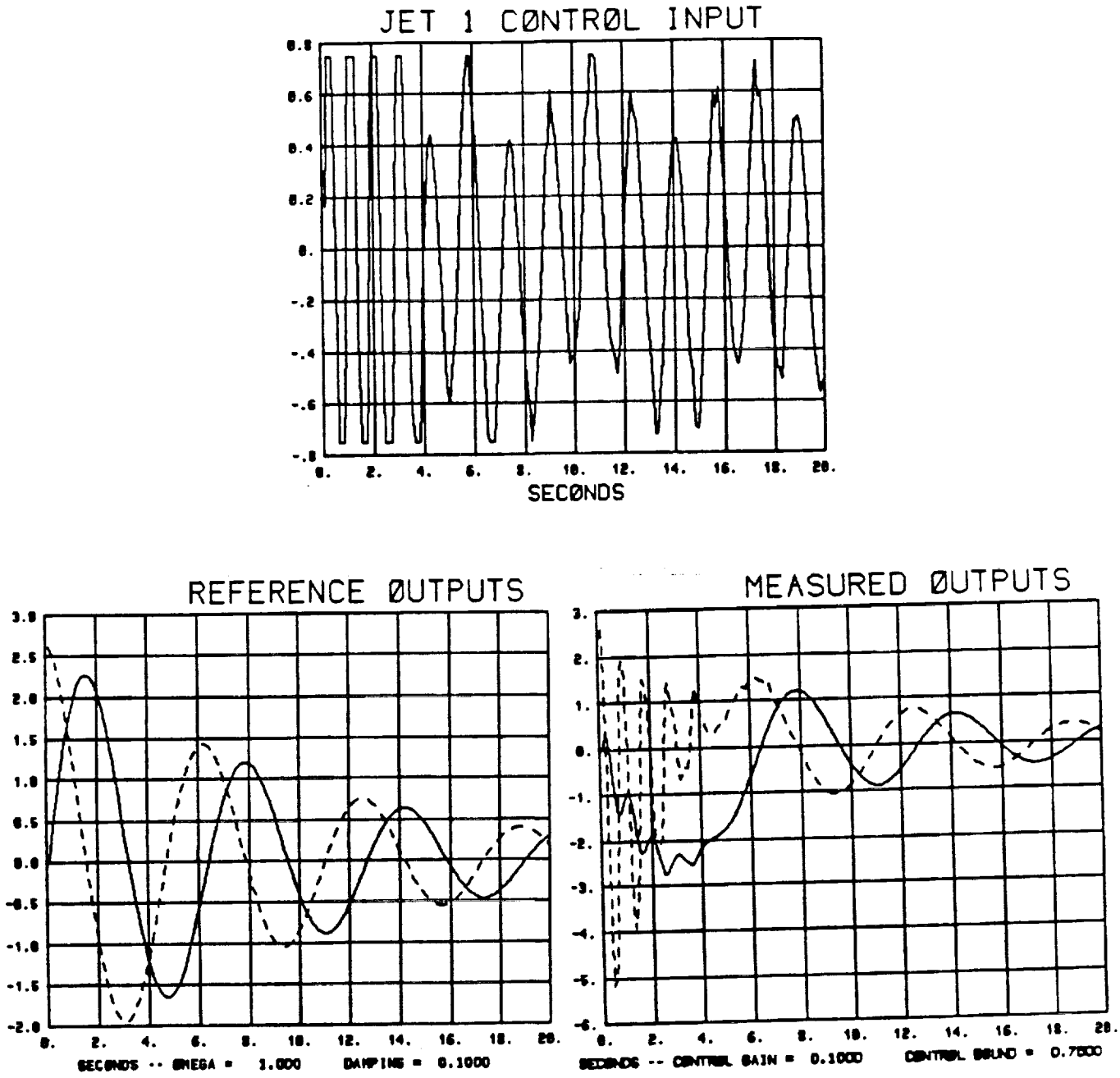


Figure 4.10: Jet input (constrained), reference and measured outputs

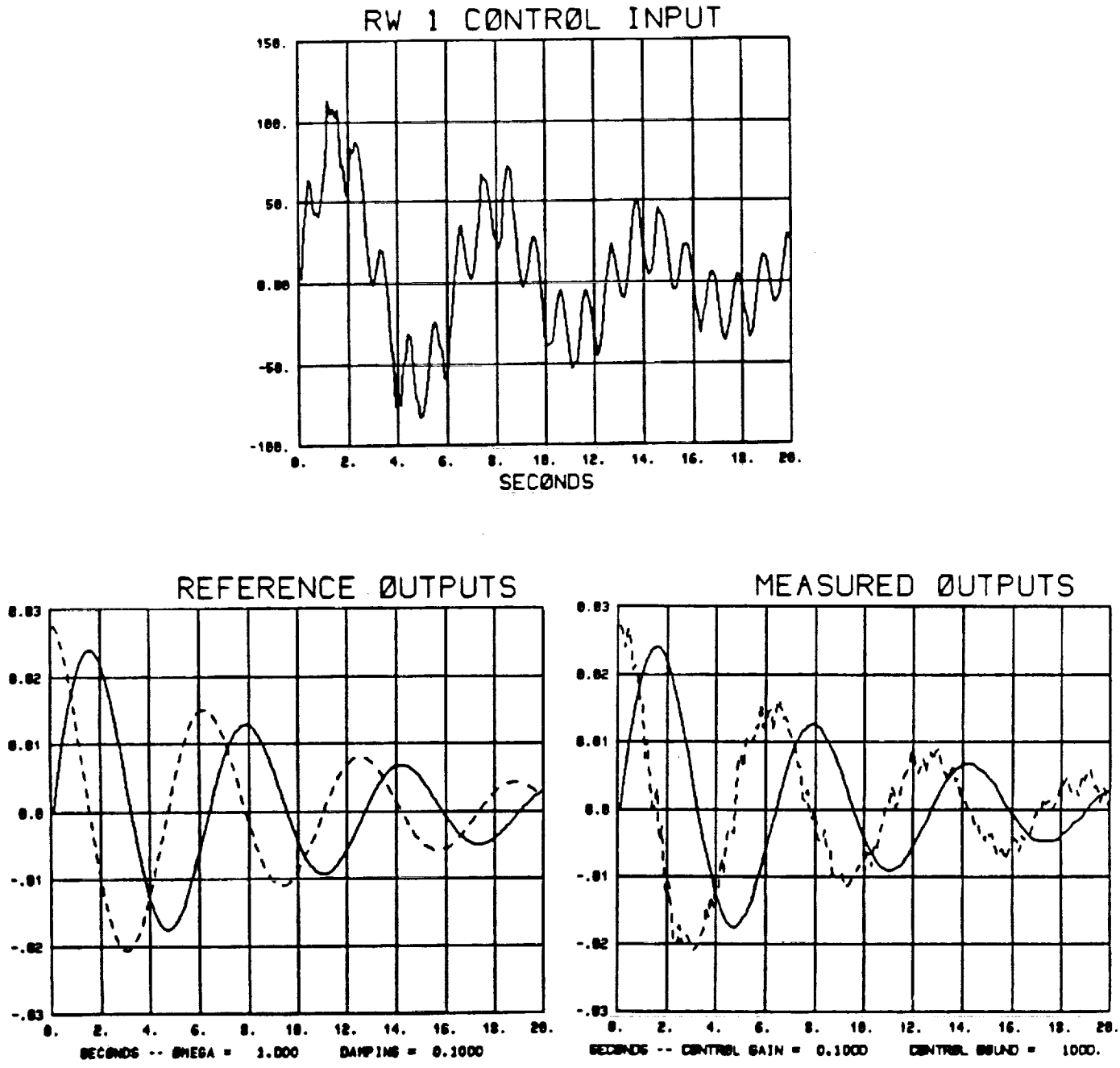


Figure 4.11: RW input (unconstrained), reference and measured outputs

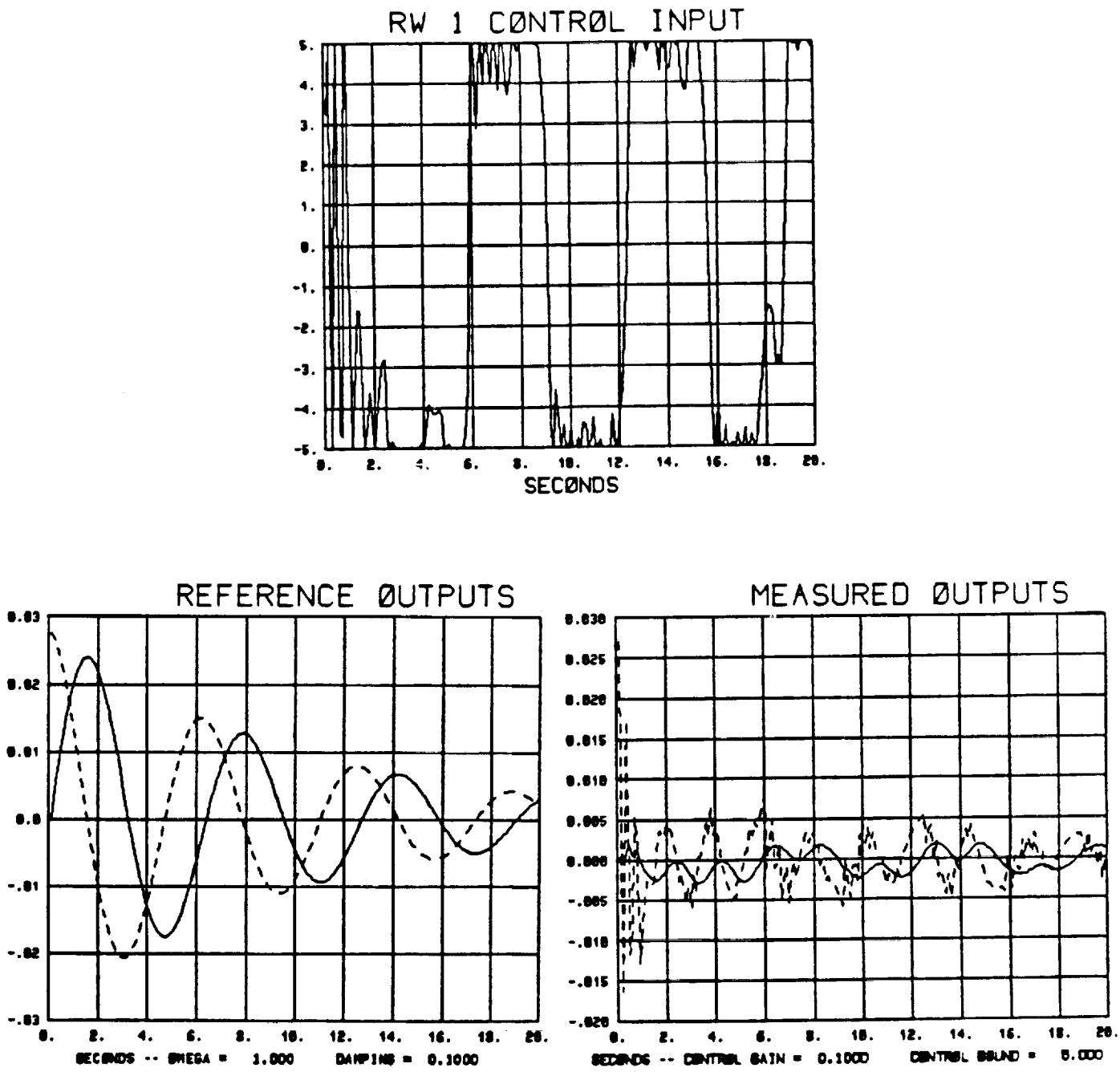


Figure 4.12: RW input (constrained), reference and measured outputs

5. INDIRECT ADAPTIVE APPROACHES

5.1 Design

In indirect adaptive control, the system parameters are estimated using some on-line identification technique such as recursive least-squares, then the control law is computed based on the most recent parameter estimates. For the SCOLE configuration, an indirect adaptive LQ controller was designed to quickly damp vibrations in the mast caused by movement of the shuttle body or reflector. This indirect LQ controller is discussed in [8] and will only be briefly described here.

There are four actuators (reaction wheels) and four sensors (accelerometers) mounted on the mast. The controller design was based on a reduced order version of the model (2.1)–(2.2) which included only the four bending modes. This model was discretized to give the state–variable formulation

$$x(k+1) = F_d x(k) + G_d u(k) \tag{5.1}$$

$$y_{acc}(k) = H_d x(k) + J u(k) \tag{5.2}$$

$$y_{pos}(k) = C x(k) \quad , \tag{5.3}$$

for appropriate dimension matrices F_d, G_d, H_d, J. It is assumed that the feedforward matrix J resulting from the acceleration measurements is known but F_d, G_d, and H_d are unknown. It is most likely that for actual testing on the SCOLE, the accelerometer data will be integrated for use in the identification process thus eliminating the need to know the feedforward matrix. For the purposes of parameter identification, the system was put in the input/output form

$$y_{acc}(k) = \theta^T \phi(k-1) + J u(k) \quad , \tag{5.4}$$

where

$$\theta^T = [-A_1 \quad -A_2 \quad (B_1 + A_1 J) \quad (B_2 + A_2 J)] \tag{5.5}$$

$$\phi^T(k-1) = [y_{acc}^T(k-1) \quad y_{acc}^T(k-1) \quad u^T(k-1) \quad u^T(k-2)] \quad . \tag{5.6}$$

The matrix θ contains the unknown system parameters and $\phi(k-1)$ is a regression vector of past input and output values available in real time. Denoting $\theta(k)$ as the estimate of θ at time k, the identification algorithm is given by

$$\theta(k) = \theta(k-1) - \alpha P(k-1) f(\theta(k-1)) \tag{5.7}$$

$$+ \frac{P(k-1)\phi(k-1)[y_{acc}(k) - \theta^T(k-1)\phi(k-1)]}{\eta_{k-1}^2 + \phi^T(k-1)P(k-1)\phi(k-1)} \tag{5.8}$$

$$P(k) = P(k-1) - \frac{P(k-1)\phi(k-1)\phi^T(k-1)P(k-1)}{\eta_{k-1}^2 + \phi^T(k-1)P(k-1)\phi(k-1)} \quad . \tag{5.9}$$

The parameter estimator is a recursive least-squares identifier with data normalization and an additional correction term which allows prior parameter information in the form of upper and lower bounds to be included.

The control law which is based on the most recent estimate of θ is a state feedback control law of the form

$$u(k) = -L(k)\hat{x}(k). \tag{5.10}$$

The state estimate is generated from an adaptive observer described by

$$\hat{x}(k+1) = F(k)\hat{x}(k) + G(k)u(k) + M(k)(y_{acc}(k) - H\hat{x}(k) - Ju(k)) \quad , \tag{5.11}$$

where $M^T(k) = [-A_1(k) \; -A_2(k)]$ and

$$F(k) = \begin{bmatrix} -A_1(k) & I \\ -A_2(k) & 0 \end{bmatrix} \quad G(k) = \begin{bmatrix} B_1(k) \\ B_2(k) \end{bmatrix} \quad H = [I \;\; 0 \ldots 0].$$

The feedback gain matrix $L(k)$ is computed from one interation of a Riccati difference equation in the manner

$$L(k) = [G^T(k)R_kG(k) + I]^{-1}G^T(k)R_kF(k) \tag{5.12}$$

$$R_{k+1} = Q + L^T(k)L(k) + (F(k) - G(k)L(k))^T R_k(F(k) - G(k)L(k)) \tag{5.13}$$

5.2 Simulation Results

With one exception to be discussed later, the output (position) was computed using the ten mode model in order to include the effects of the six unmodeled modes. Only one set of input/output pairs, that of the reaction wheel and accelerometer (x-direction) mounted toward the top of the mast, will be included in this report. For all cases, the initial state was chosen to be $x^T(0) = [0\ 0\ 0\ 0\ 0\ .1\ .1\ 0\ .1\ .1]$ (in) so that only the four modeled bending modes were initially excited.

For comparison purposes, a set of baseline plots were first generated. In Figure 5.1, the open-loop output position response is shown. In Figure 5.2, both the LQ control input without identification and the corresponding output position are shown. In this case the identification loop was bypassed and $\theta(k)$ was set equal to the actual θ for computing the control law. The effect of one of the unmodeled pendulum

modes on the output position is very apparent. Also, the control input greatly exceeds the physical constraints on the reaction wheel (1.25 in-lb.) for the SCOLE. In Figure 5.3, the LQ control input and the output position are shown when these physical constraints are taken into account.

Next, identification is introduced into the loop. All entries in $\theta(0)$, the initial guess for θ, are in error by varying values between zero and twenty percent from the true values. In the first run, unmodeled dynamics were not included, that is, the output position was computed using the four mode "control" model. The unconstrained and constrained control input along with the corresponding output position response are shown in Figures 5.4 and 5.5, respectively. In the absence of unmodeled modes, the adaptive controller performs well in vibration control of the mast.

Finally, in order to assess the effects of unmodeled dynamics, the adaptive controller was applied to the ten mode model of SCOLE. In Figure 5.6 the unconstrained control input and the output position response are shown. The unmodeled dynamics cause the closed-loop system to go unstable and the input and output grow without bound. In Figure 5.7 the constrained control input and the corresponding output position response are shown. The forced bounds on the control input lead to much better performance of the adaptive closed-loop system.

The results, thus far, indicate that a reasonable control strategy for vibration control in flexible structures would be to *shut off* the estimation process after the parameters are suitably identified and then use a fixed LQ controller. The estimation error would be continually monitored and identification would resume if this error exceeds some pre-specified bound indicating a change in system parameters has occured.

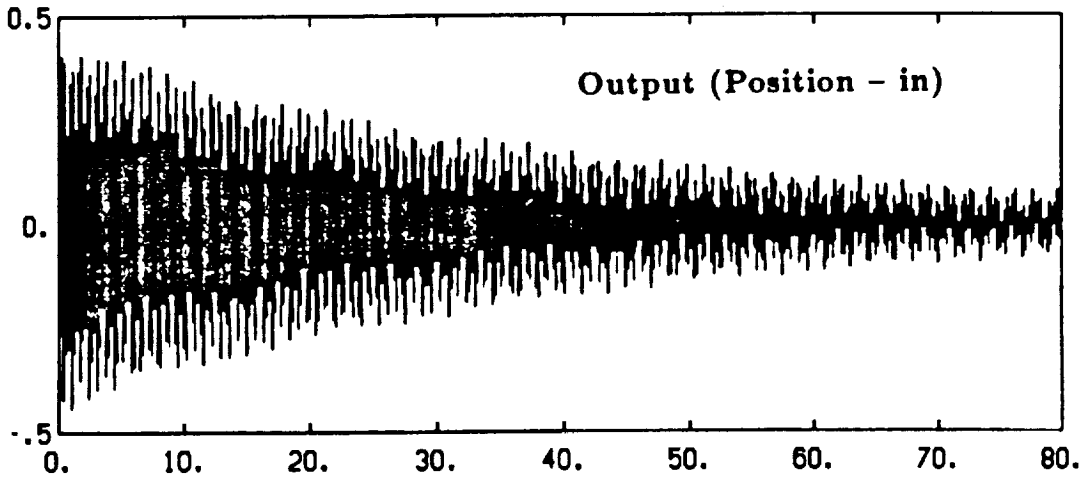
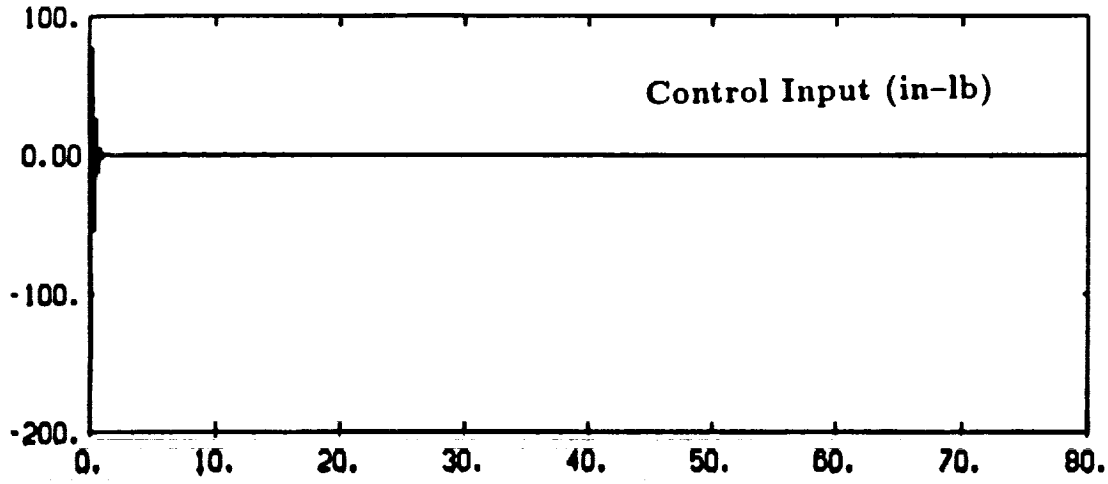


Figure 5.1: Open Loop Response



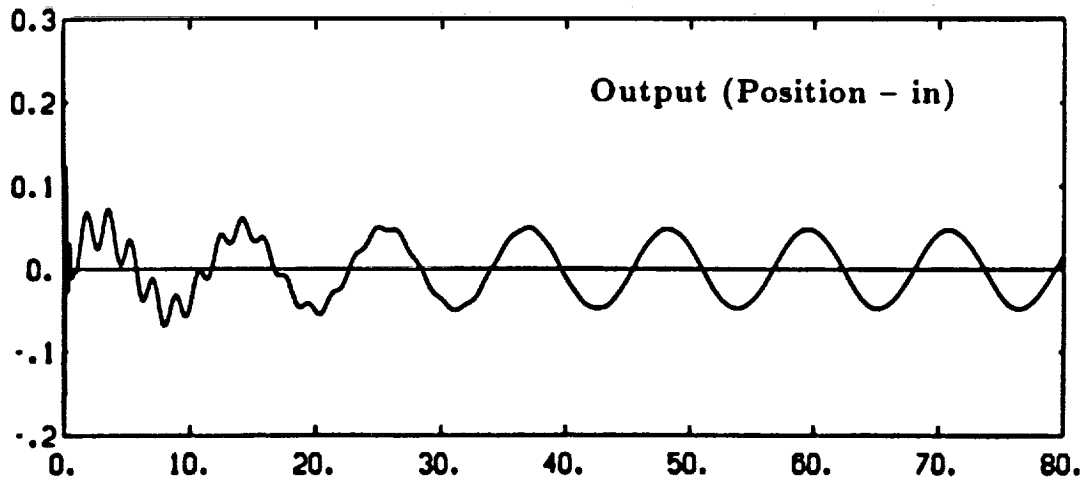
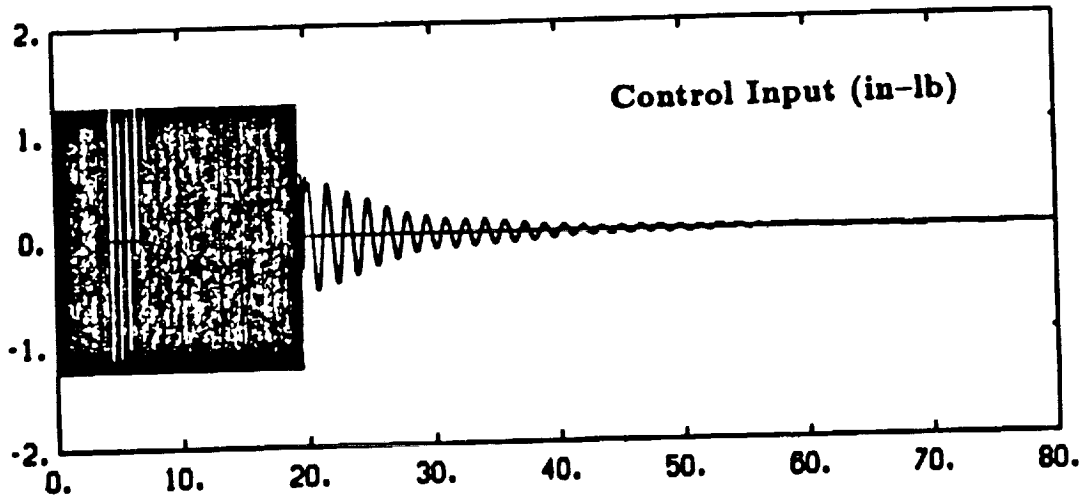


Figure 5.2: LQ Control without constraints



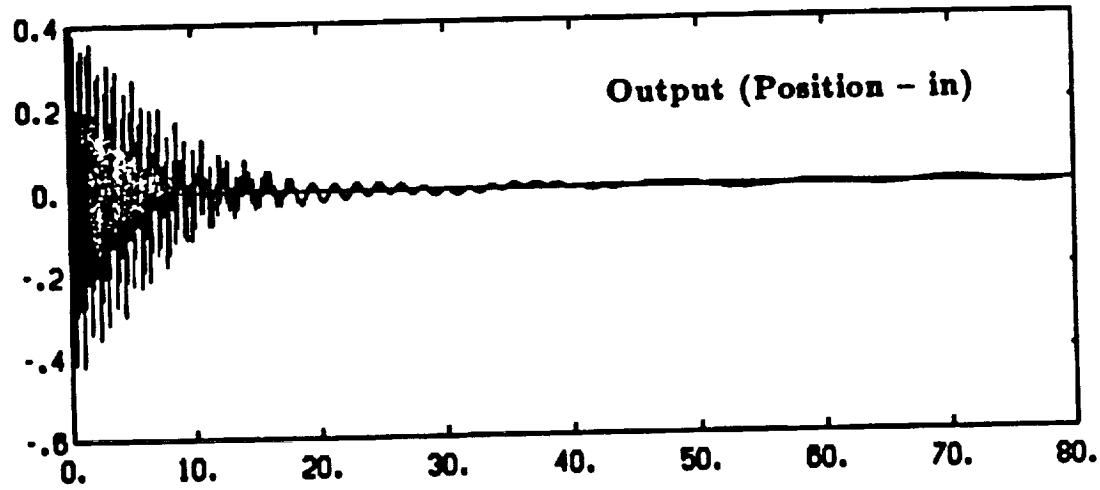


Figure 5.3: LQ Control with constraints

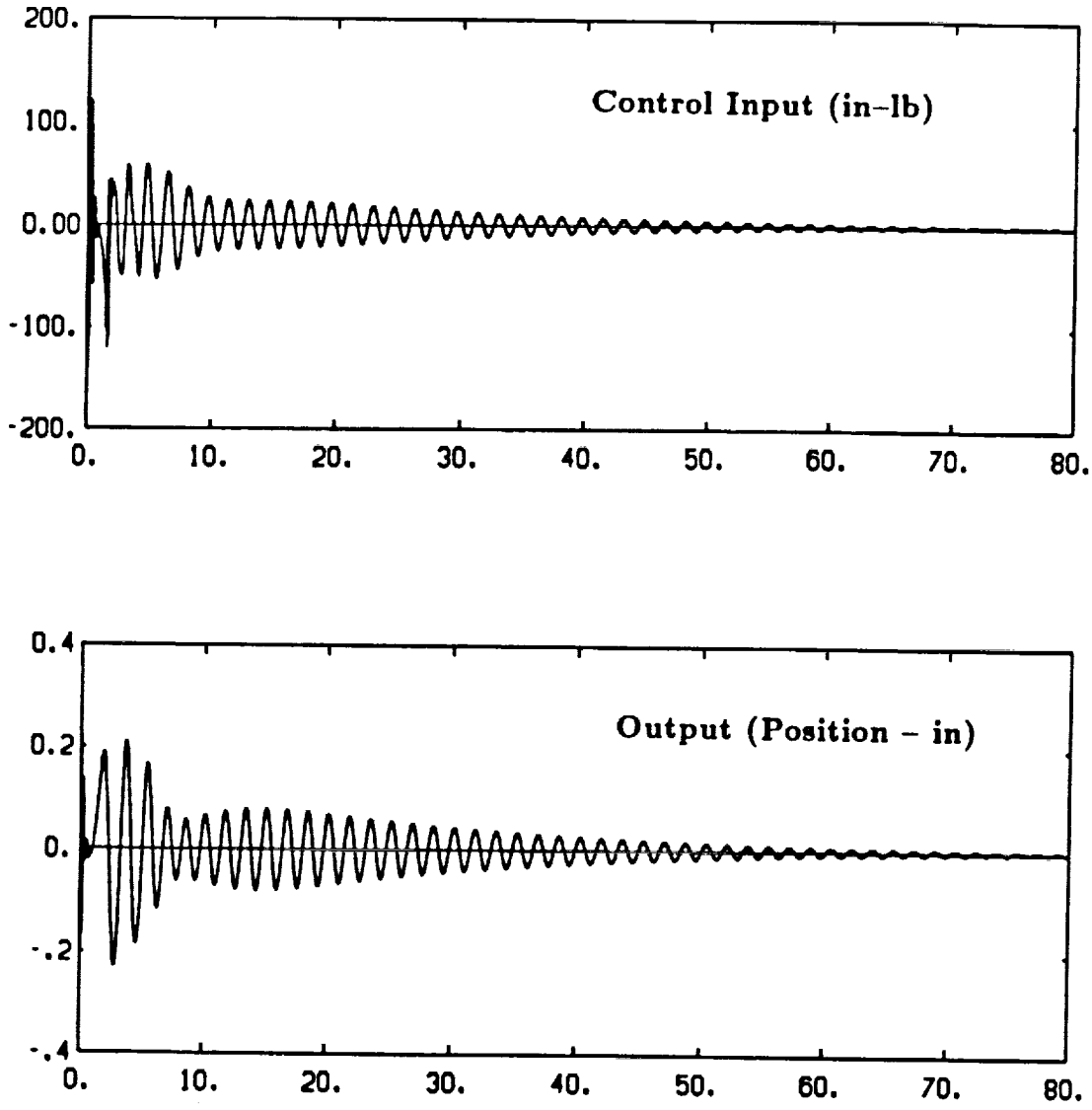
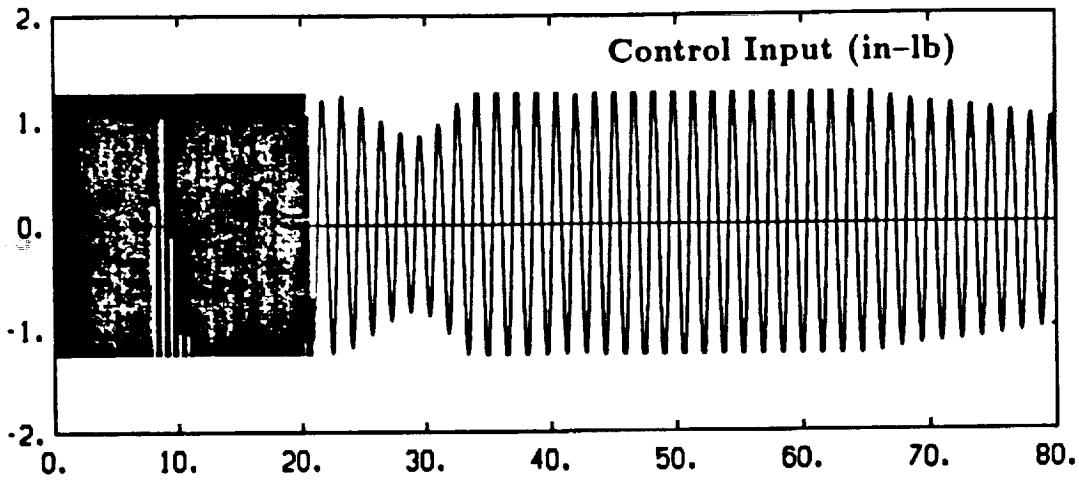


Figure 5.4: Adaptive LQ without constraints (no unmodeled dynamics)



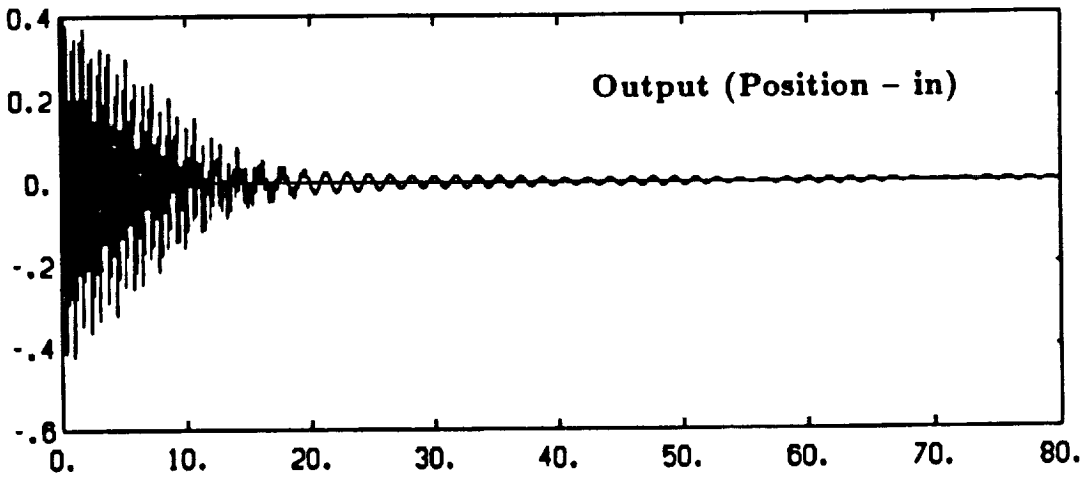


Figure 5.5: Adaptive LQ with constraints (no unmodeled dynamics)

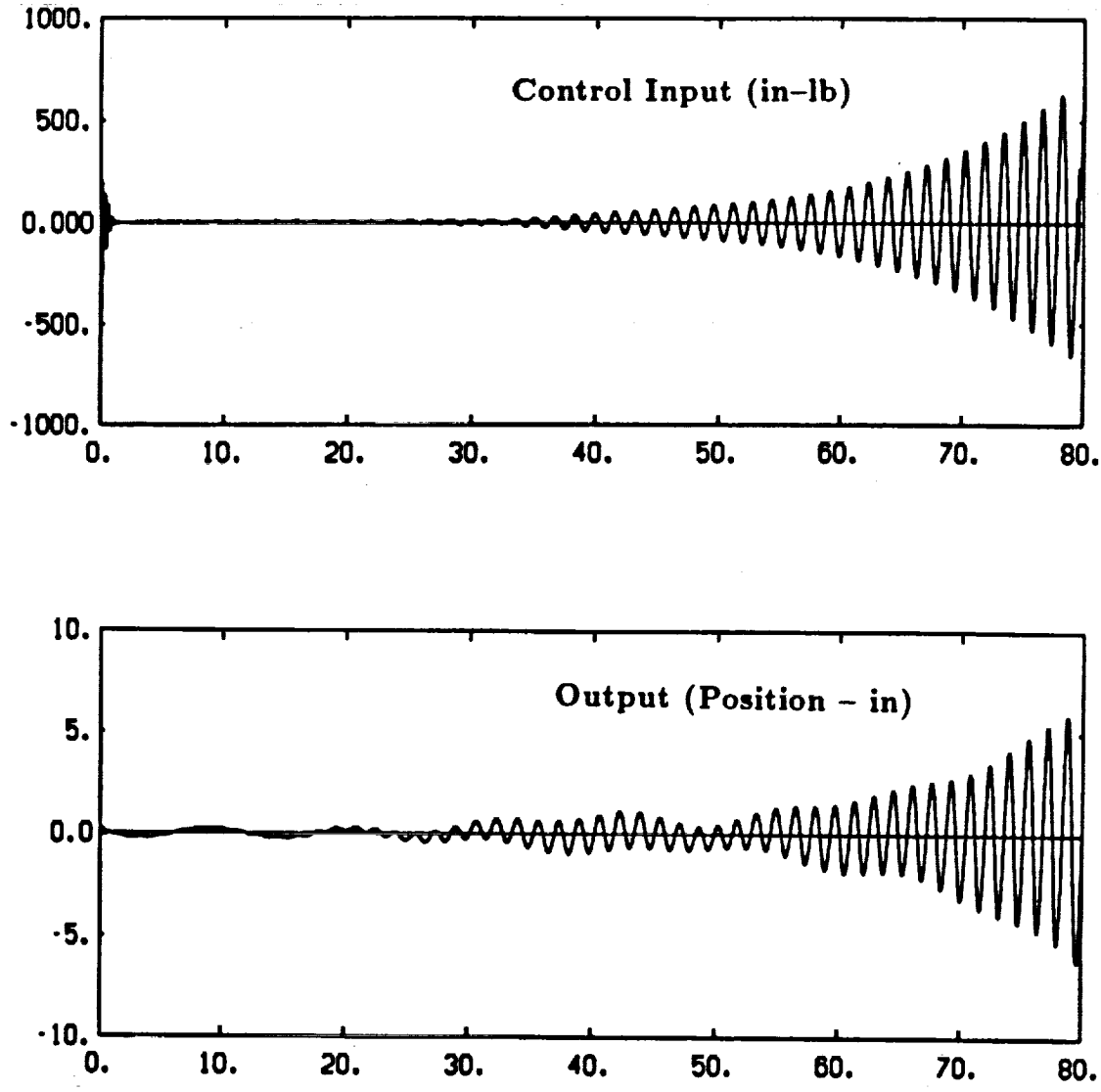


Figure 5.6: Adaptive LQ without constraints (full model)

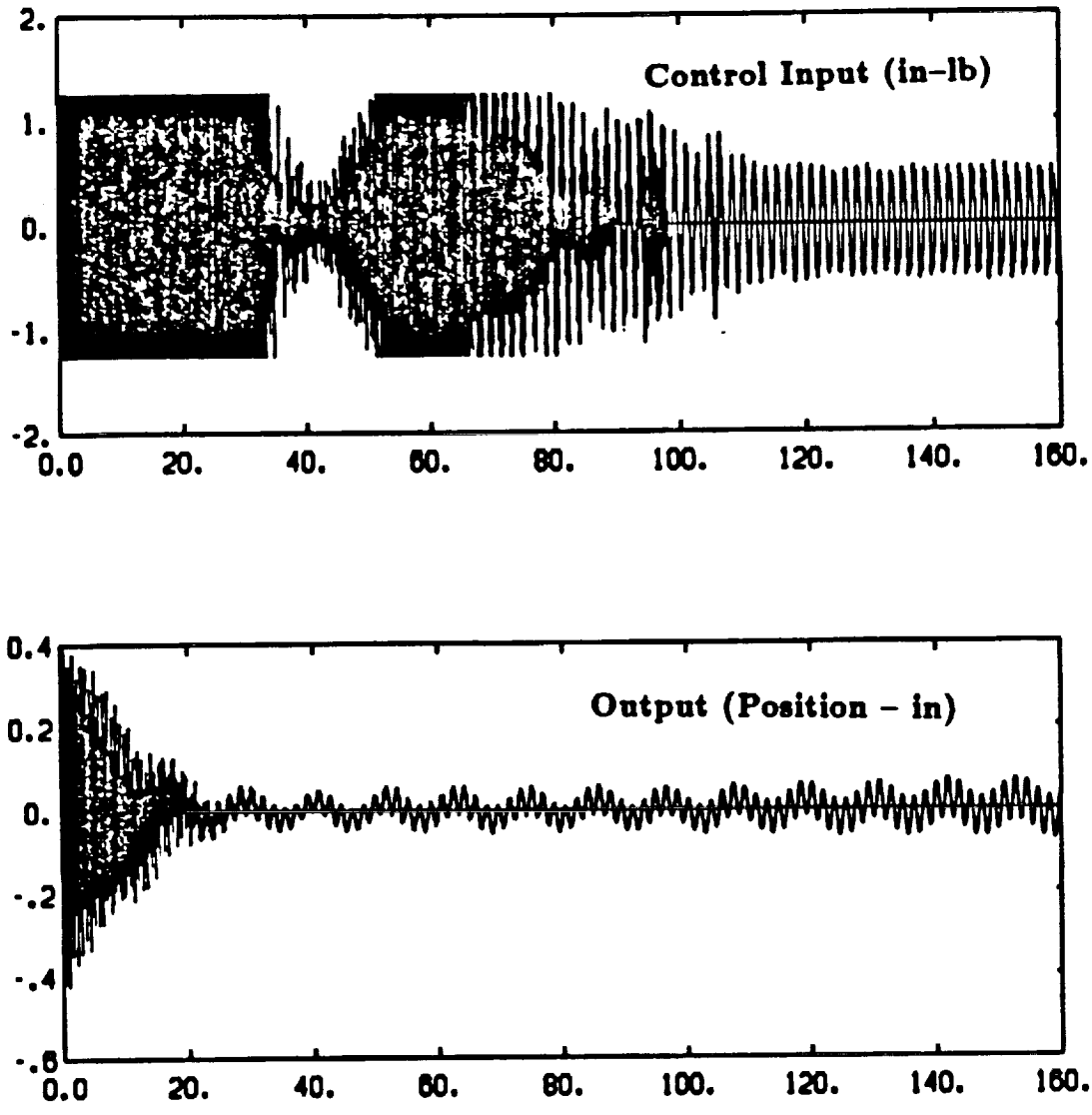


Figure 5.7: Adaptive LQ with constraints (full model)

§ 3 Robust Indirect Adaptive Control

The robustness of the adaptive controller with respect to unmodeled dynamics can certainly be improved. One option currently being investigated is the incorporation of a variable dead zone into the identification scheme for the system parameters [9]. The motivation for adding a dead zone into the identification scheme is clear if we express the tracking error as $e(k) = e_i(k) + e_m(k)$ where $e_i(k)$ represents an error in identification of the system parameters and $e_m(k)$ denotes the error due to disturbances or unmodeled dynamics. If the tracking error is due mainly to identification error, then certainly parameter estimation should continue. However, if the tracking error arises mainly from unmodeled dynamics or disturbances, continuing with the identification could move the parameter estimates further from the true values and thus degrade system performance. The variable dead zone is just an attempt to extract a reasonable approximation of the identification error from the measurable tracking error using both on-line information and prior system information. Thus, whenever the tracking error is due mainly to unmodeled dynamics, the identification scheme is "shut off." A variable dead zone has the advantage of simplicity; that is, it can be easily incorporated into most existing adaptive controllers. However, designing the dead zone is not a trivial problem and this choice will affect the performance of the system.

The design of the dead zone has thus far been investigated for some simple examples which have characteristics of flexible structures control problems. By way of example, then, consider a two mode flexible structure described by

$$\frac{d}{dt}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -4 & 0 & -.04 & 0 \\ 0 & -100 & 0 & -1 \end{bmatrix}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ .5 \\ 1 \end{bmatrix} u(t) \tag{5.14}$$

$$y(t) = [0 \quad 0 \quad 0.5 \quad 1]\begin{bmatrix} q \\ \dot{q} \end{bmatrix} \quad . \tag{5.15}$$

The reduced order model used for designing the adaptive controller is given by

$$\frac{d}{dt}\begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -4 & -.04 \end{bmatrix}\begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} u(t) \tag{5.16}$$

$$y(t) = [0 \quad 0.5]\begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} + e_m(t) \quad . \tag{5.17}$$

The estimate for the identification error $e_i(k)$ used in the least-squares algorithm was:

$$e_i(k) = \begin{cases} 0 & \text{if } |e(k)| < d(k) \\ e(k) - d(k) & \text{if } e(k) > d(k) \\ e(k) + d(k) & \text{if } e(k) < -d(k) \end{cases} , \tag{5.18}$$

where $d(k)$ represents an upper bound on the magnitude of the modeling error $e_m(k)$ and is given by

$$m(k) = (0.951)m(k-1) + u(k) \tag{5.19}$$

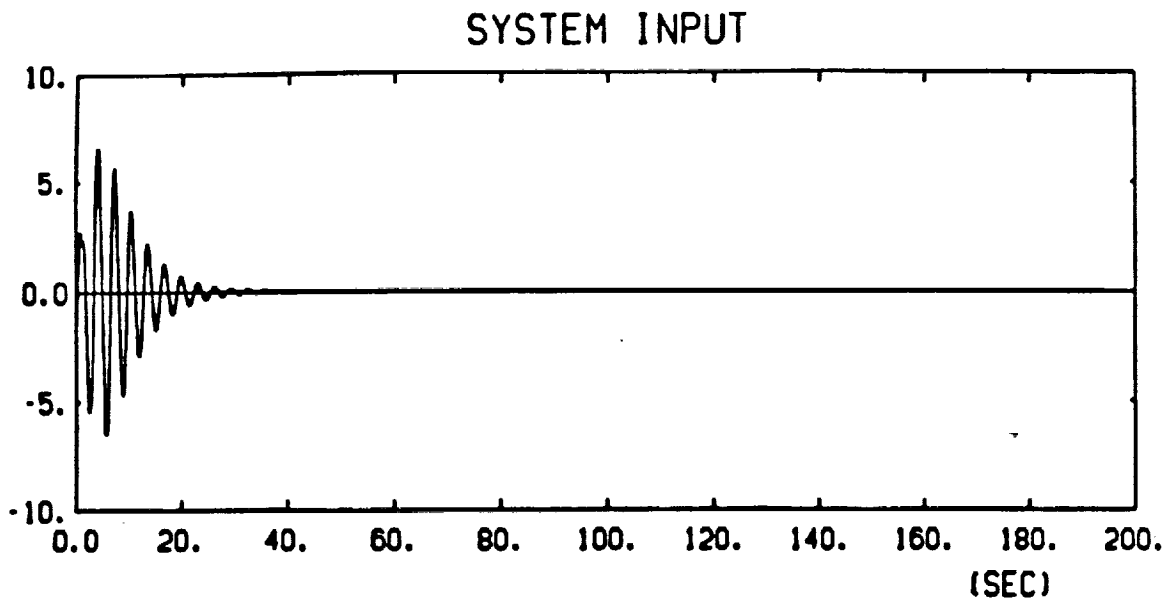
$$d(k) = (1.026)\,|m(k)| \quad . \tag{5.20}$$

The control law chosen was the adaptive LQ control law previously discussed for SCOLE.

Figure 5.8 shows the control input and the resulting output when the parameters of the reduced order model are assumed to be known exactly. The output was computed from the full order model.

Figure 5.9 shows the control input and resulting output when estimation is included but a dead zone is not incorporated into the least-squares algorithm. The initial parameter estimation errors ranged from 10-20%. As in the case of SCOLE, the unmodeled dynamics cause the adaptive control loop to go unstable resulting in an input and output which grow without bound.

Finally, the dead zone described above was incorporated into the estimation scheme. As shown in Figure 5.10, both the control input and the resulting output converge to zero.



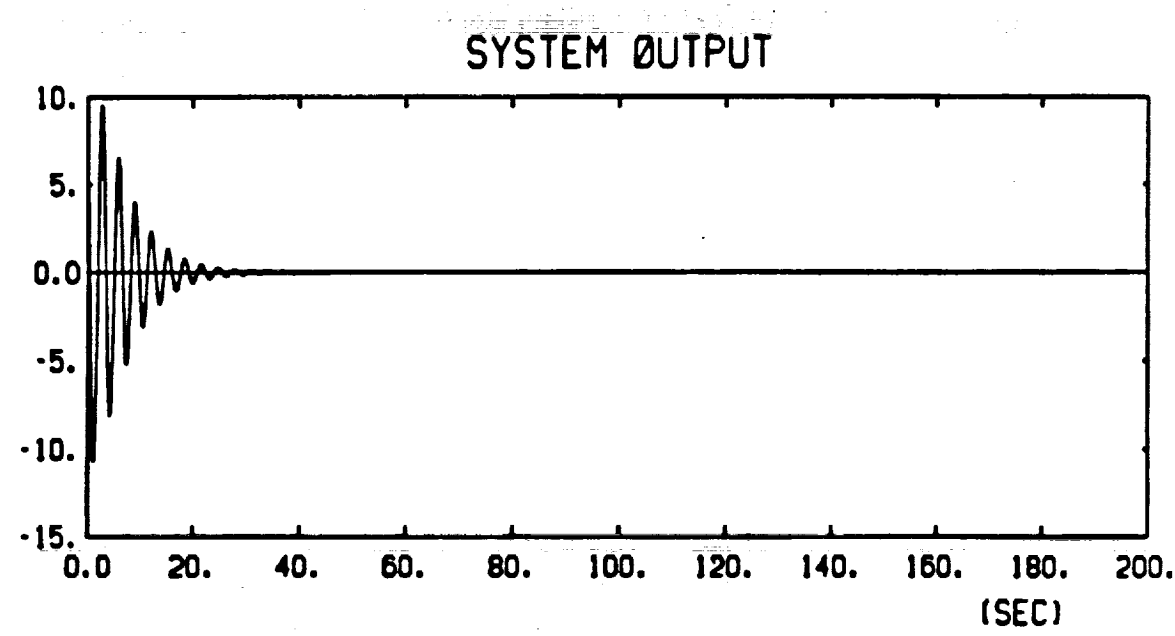
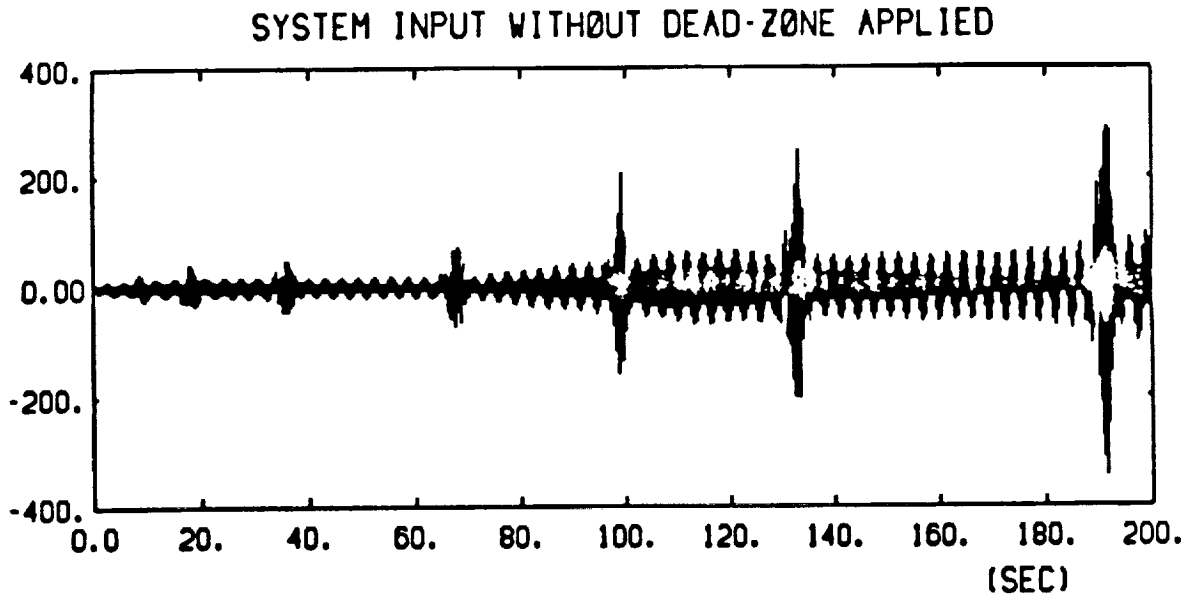


Figure 5.8: LQ control (unmodeled dynamics present)



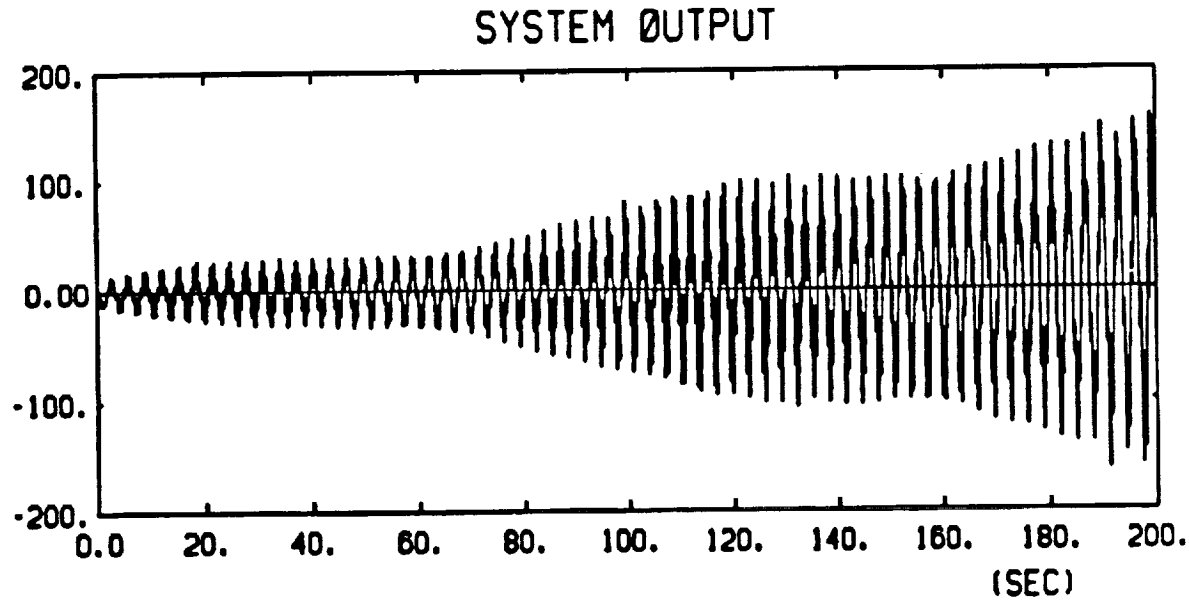


Figure 5.9: Adaptive LQ control without the dead zone

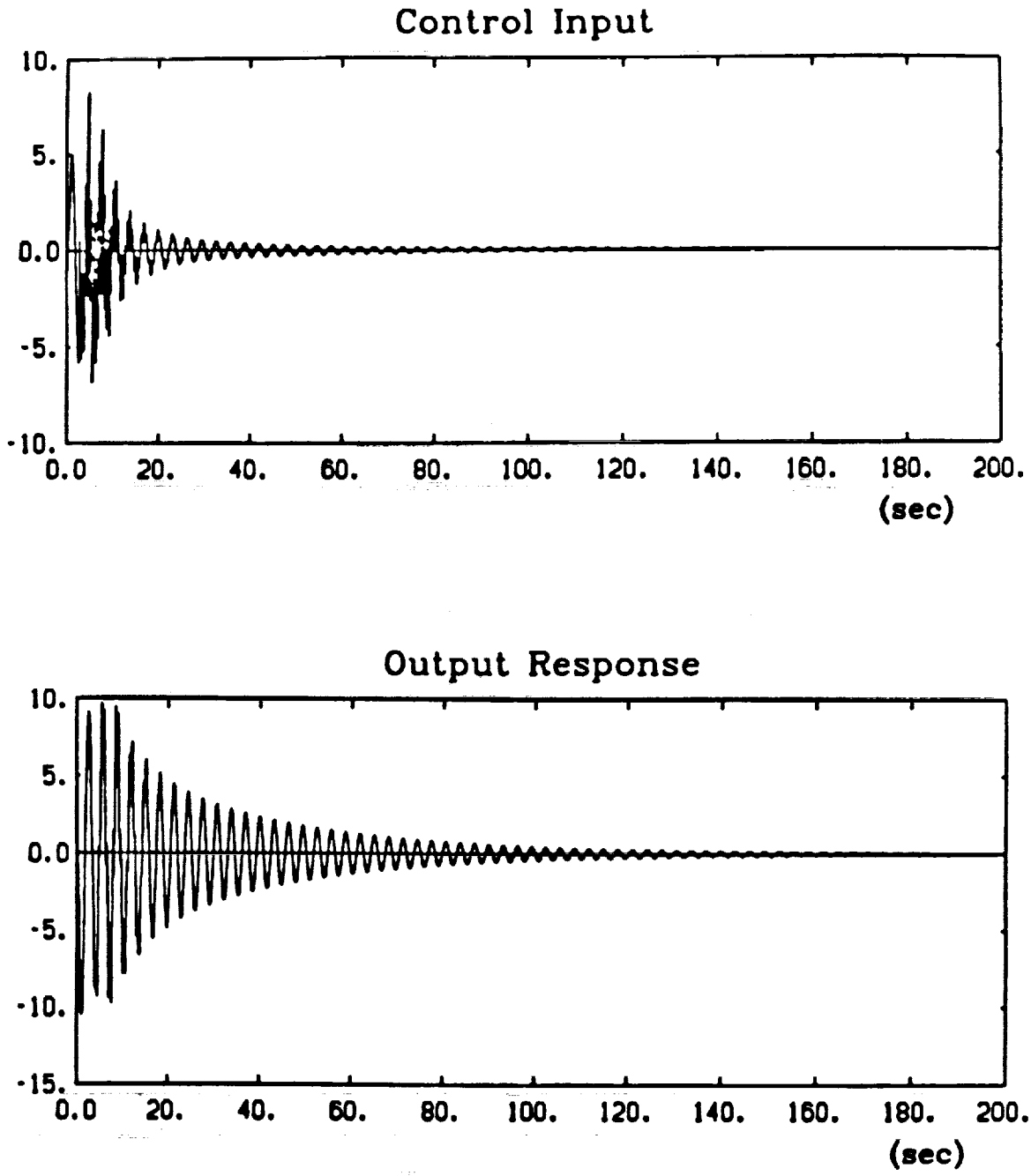


Figure 5.10: Adaptive LQ control with the dead zone

5.4 Discussion

Initial results on the incorporation of a variable dead zone into the least-squares estimation scheme look promising. Current work centers on the design of the dead zone for SCOLE. Furthermore, a better choice of modes will be made in the reduced order model for the SCOLE to reduce control and observation spillover. As mentioned, one of the unmodeled low order pendulum modes (0.08 Hz) appears in the output position response even in the non-adaptive case.

Another approach for improving robustness is also being investigated. The empirical transfer function estimation (ETFE) method will be used to identify the system parameters [10]. Using finite length discrete Fourier transforms (DFTs) of the input/output data, the ETFE method provides both an estimate of the plant transfer function and a frequency-dependant upper bound on the modeling error between the estimated transfer function and the actual plant transfer function. This identification scheme has been shown to possess certain desirable robustness characteristics for use in adaptive control [11]. Because it requires DFTs of the input/output data, the ETFE method will require more computational time than the variable dead zone identification scheme. However, Tzes and Yurkovich have shown that with modest computational load (less than 10 ms) the ETFE scheme may be carried out on-line, with a recursive least squares estimator, to accurately identify frequency response characteristics for a slewing flexible beam [12]. Moreover, it is shown in [12] that the frequency response information thus obtained is useful in designing an LQ-adaptive controller with frequency weighting; details of this work will appear in a future report.

6. LABORATORY EXPERIMENTATION

6.1 Visits to Langley

May 1987

In May of 1987 we visited the LaRC facility and worked with Mr. Jeff Williams for two days in the laboratory preparing for control implementation. Our progress was limited due to unforeseen problems with the computer and compiler systems. We were, however, successful in familiarization with the current set-up, and have since completed the designs described herein, and have completed coding for use on the structure.

December 1987

The next visit to Langley was in December of 1987 for five full days of implementation and experimentation. All control designs and necessary software were transported via floppy disks, so that the on-site work began by downloading the various control law programs and their data files for compilation on the the SCOLE computer, a Charles River 68000 based system. A variety of differences between the Vax 11/785 FORTRAN code and the Unix/Green Hill FORTRAN compiler were identified so that first two days were consumed in making the code changes, compiling, and testing the simulator code. During the software preparation, hardware problems were also delineated. For example, the reaction wheels at the reflector/-base had not been used before, and several wiring checks had to be carried out.

The third day began with experimentation involving the algorithm with the least computational requirements, the observer/pole placement design. The initial plan was to run this algorithm at 100 hertz but the computational requirements caused the sample time to be exceeded before the control law was complete; the algorithm did, however, operate at 40 hertz without a time exceeded warning. We had over estimated the Floating Point Operations per Second (FLOPS) capability of the computer, primarily because the previous hardware configuration was such that the floating point operations were done by a Sky floating point board. This board has capabilities of 3 micro second operations or 330,000 FLOPS. Since then, however, the computing environment for the SCOLE set-up has been changed to a 68020/68881 processor, and the compiler has also been changed. Because of this unforeseen problem, some time was spent benchmarking various operations; it appears that the best expected performance is now 100,000 FLOPS. Utilizing the Sky board again, with the current configuration, did not result in increased performance. In the interest of time, we terminated our efforts at increasing FLOP speed at that point.

Because of the unexpected slower speed capabilites, most of the control laws brought were too slow in their present form, since most of the algorithms used continuous time observer equations. The algorithms can, however, be made more efficient, and current efforts are directed in this vein. The inirect adaptive control designs, which are already in discrete time form, may require more FLOP capability than the SCOLE computer can provide at the present time.

Two algorithms were marginally acceptable in terms of achievable sampling period, specifically, the pole placement and MEOP algorithms, although the maximum achievable sampling rate for the MEOP design was 10 hertz. Before experimentation with these control laws, various tests were run to determine if the amplifers were functioning properly and if wire phasing was correct. Due to concerns with the amplifiers, the reaction wheel actuators were run at 5 in-lb or 30% of the maximum rating.

Both of the aforementioned algorithms showed some undesirable behaviour. To determine some rationale for the poor results, the outputs from the observers were recorded, indicating that several of the modes which should be pure sinusoids were heavily distorted. This in turn indicated that noise and observation spillover may be responsible for poor observer convergence. The SCOLE Sensor conditioners are two pole low pass filters with a cutoff frequency of 10 hertz. Our state space model of the SCOLE structure upon which the observers were designed includes modal frequencies of a maximum 5 hertz. Observation spillover from the unmodeled modes in the 5 to 10 hertz range may account for some of the distortion. Channel noise seemed to be significant relative to the signal input data. A data sample of channel noise was collected and brought back to determine noise power and spectrum.

On day five, system identification experiments/data collection and a simple control law (described below) were priorities. To identify various input output relationships, white noise inputs were applied to the various actuators individually. All the outputs were collected for each actuator run. A thousand data points were collected at sampling rates of 20 or 50 hertz. During this data collection, we noted that the low authority reaction wheels mounted near the top of the mast provided little excitation of any of the modes. This data is currently being used for investigation of various system identification techniques, primarily on–line schemes such as the above–mentioned ETFE approach.

6.2 Software

The software to implement the control algorithms described in previous sections was developed with the following goals. One, the code must be implementable on OSU computers (Vax 11/785 and MicroVax II) simulating all experimental hardware

dependent subroutine calls (A/D, D/A, timers, and so on) with code to run a state space model of SCOLE. The control law code is validated by this process since the SCOLE computer debugging facilities are limited. Also, the simulation produces data for comparison to actual data output. The same code used for the simulations is used to run the experiment by linking in the runtime libraries instead of simulation libraries.

Secondly, the software is designed with standard hooks to the control law subroutine. This allows one standard interface to the experimental hardware to be designed, implemented, and validated. To implement a control law, only two subroutines must be written, interfacing to the main logic through sensor–in actuator–out data arrays. All other tasks are performed by the standard interface. These tasks are

- Set sample rate, number of samples, sensor bias levels, and so on;
- Check if control law calculation time exceeds sampling interval;
- Sample analog inputs;
- Remove input bias, reorder the data array to standard model order, and convert to usable values;
- Integrate all sensor values;
- Scale and limit output values;
- Convert to DAC array ordering, output analog outputs;
- Save all input and output values.

6.3 Results

To complete our most recent visit, we chose to implement a simple rate feedback law, consisting of a proportional feeback term for each rate sensor and reaction wheel actuator in each plane. The feedback gains were chosen by exciting the modes in each plane individually and choosing appropriate values according to the observed behaviour. The final data collected was done by exciting the system in all three planes and running all three actuators simultaneously. Table 6.1 shows the actual feedback gains. From the table, we note that the X direction actuators were relatively sensitive and could not tolerate large gains. Figures 6.1 and 6.2 show the actual sensor outputs and actuator inputs, respectively, for the rate feedback law to an initial disturbance.

Table 6.1: Feedback Gains for Rate Feedback Control Scheme

X Axis	-5
Y Axis	-100
Z Axis	-50

The results from this experiment imply that the output from the X axis rate feedback gyro is always positive, not a sinusoid oscillating about zero. This would essentially indicate that the position error is positive and unbounded, suggesting a failure either in the rate gyro itself or in the measurement process[1]. This probably accounts for some of problems encountered with the more sophisticated algorithms, and may account for the low feedback gain required on this axis during the rate feedback experiment. The performance in the other axes was satisfactory.

[1] Jeff Williams has been made aware of the problem and is investigating.

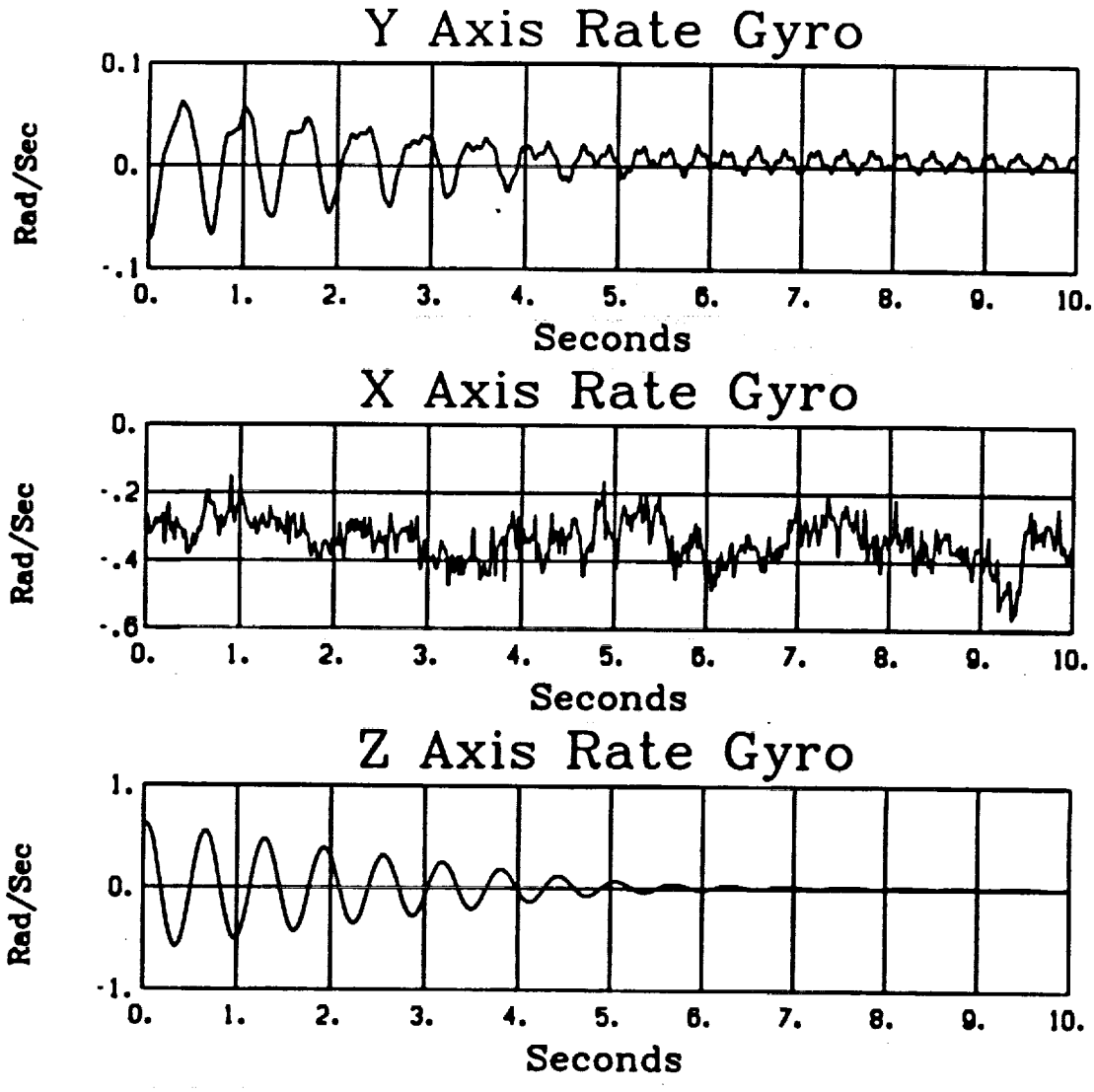


Figure 6.1: Actual sensor outputs, Rate Feedback design

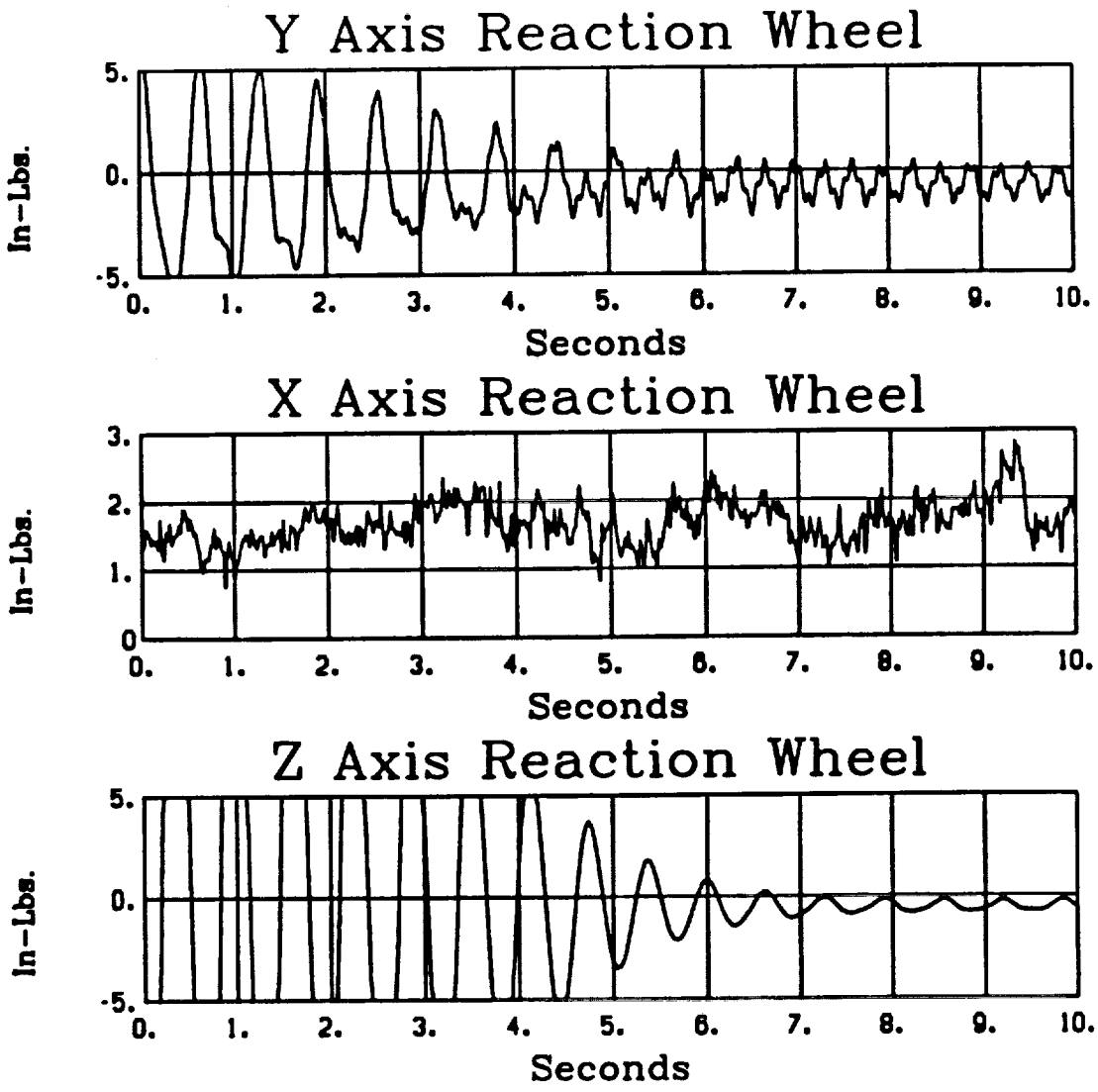


Figure 6.2: Actual actuator inputs, Rate Feedback design

7. COFS CONTROLLER DESIGN

7.1 Introduction

In spite of its cancellation, the original Mast Flight System provides (and will probably continue to provide) an interesting model that can be used to test in simulation different control approaches. Finite element models of varying complexity exist, experiments have been performed on the individual actuators.

The basic element in the Mast Flight System is a 60.7 meter long, triangular cross section truss structure. The truss has 54 bays with LDCM actuators distributed along the length. There are four primary actuators located at the tip of the beam so as to produce both x and y axis forces and z axis torques. The actuators at the intermediate stations are smaller devices and can be utilized to check decentralized control issues[1].

Our effort in the Control Research Lab of the Ohio State University was concerned with three issues:

1. Design of an optimal decentralized controller.
2. Design of a decentralized controller through overlapping decompositions and loop–shaping.
3. Design of low–order controllers through fixed–order, direct optimal projection techniques.

Throughout this section we will be referring to certain data which was supplied to us at different times from NASA LaRC. These may not be the same versions that were finally accepted and certified or utilized in NASA's final design effort. Be that as it may, the data used here is still very representative of such a flight system.

7.2 Design of an Optimal Decentralized Controller

7.2.1 Background Theory

Consider

$$\dot{x} = Ax + \sum_{i=1}^{\nu} B_i u_i \qquad ; x(0) = x_0 \tag{7.1a}$$

[1] In the analysis to follow identical actuators have been assumed.

$$y_i = C_i x \qquad ;i = 1,\ldots,\nu \tag{7.1b}$$

with cost

$$J = \int_0^\infty (x^T Q x + 2\sum_{i=1}^{\nu} x^T N_i u_i + \sum_{i=1}^{\nu} u_i^T R_i u_i)\, dt \tag{7.2}$$

and the following feedback structure constraint:

$$u_i = K_i\, y_i \qquad ;i = 1,\ldots,\nu \quad . \tag{7.3}$$

It can be shown [13,14] that the solution of the above optimal control problem reduces to solving the following system of non-linear algebraic equations:

$$\begin{cases} A_c^T P + P A_c + \hat{Q} = 0 \\ A_c L + L A_c^T + X_0 = 0 \end{cases} \tag{7.4}$$

and

$$\nabla_{K_i} J = B_i^T P L C_i^T + R_i K_i C_i L C_i^T + N_i^T L C_i^T = 0 \qquad ;i = 1,\ldots,\nu \tag{7.5}$$

where

$$A_c = A + \sum_{i=1}^{\nu} B_i K_i C_i \tag{7.6a}$$

$$\hat{Q} = Q + \sum_{i=1}^{\nu} C_i^T K_i^T R_i K_i C_i + \sum_{i=1}^{\nu} (N_i K_i C_i + C_i^T K_i^T N_i^T) \tag{7.6b}$$

$$X_0 = x_0\, x_0^T \quad . \tag{7.6c}$$

Many approaches have been introduced in the literature to solve the above system of equations. The approach pursued in our software package[2] is a gradient method.

Extension to the basic problem can be done by considering the following decentralized dynamic stabilizing compensator:

$$\dot{z}_i = F_i z_i + G_i y_i \tag{7.7}$$

$$u_i = H_i z_i + N_i y_i \qquad ;i = 1,\ldots,\nu \quad . \tag{7.8}$$

[2]The software package DOLORES was developed by Professor Ü. Özgüner, Mr. Khorrami and Mr. Tien for another project

The problem of dynamic output feedback is reduced to the previous problem by augmenting the state space of the system and the stabilizing compensators, that is,

$$\begin{bmatrix} \dot{x} \\ z_1 \\ z_2 \\ \vdots \\ z_\nu \end{bmatrix} = \begin{bmatrix} A & \vdots & 0 \\ \cdots & \vdots & \cdots \\ 0 & \vdots & 0 \end{bmatrix} \begin{bmatrix} x \\ \cdots \\ z_1 \\ z_2 \\ \vdots \\ z_\nu \end{bmatrix} + \sum_{i=1}^{\nu} \begin{bmatrix} B_i & 0 \\ \cdots & \cdots \\ 0 & 0 \\ \vdots & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_i \\ \cdots \\ v_i \end{bmatrix} \quad , \tag{7.9}$$

$$\bar{y}_i = \begin{bmatrix} C_i & 0 & \ldots & 0 \\ 0 & 0 & \hat{C}_i & 0 \end{bmatrix} \begin{bmatrix} x \\ \cdots \\ z_1 \\ z_2 \\ \vdots \\ z_\nu \end{bmatrix} \tag{7.10}$$

where $v_i = \dot{z}_i$ and $\hat{C}_i$ is provided by the user. The cost may be further modified to

$$J = \frac{1}{2} \int_0^\infty (x^T Q x + 2 \sum_{i=1}^{\nu} x^T N_i u_i + \sum_{i=1}^{\nu} u_i^T R_i^1 u_i + \sum_{i=1}^{\nu} \dot{z}_i^T R_i^2 \dot{z}_i)\, dt \quad . \tag{7.11}$$

7.2.2 Design of an Optimal Decentralized Controller

The modes and mode shapes used in this study are listed below:

Mode	Freq. (Hz)	Bay 28	Tip
(1) 1st x-z	0.187	.0149	.0486
(4) 2nd x-z	1.36	.0647	-.0243
(7) 3rd x-z	3.93	.0111	.0167
(10) 4th x-z	6.84	-.0567	-.0107

Each of the actuators with their associated (inner-loop) compensators (already designed) can be described by a fourth order model:

$$\dot{x}_c = A_c x_c + B_{c_1} \delta_c + B_{c_2} p \tag{7.12}$$

$$f = C_c x_c + D_{c_1} \delta_c + D_{c_2} p \tag{7.13}$$

where,

$$A_c = \begin{bmatrix} -\tau & 0 & 0 & 0 \\ 0 & -\tau & (1-g_2)\tau & 0 \\ 0 & 0 & 0 & 1 \\ \frac{K_0}{m} & -\frac{K_0}{m} & -\frac{g_2 K_0}{m} & 0 \end{bmatrix} \tag{7.14}$$

$$B_{c_1} = \begin{bmatrix} (1-g_1)\tau \\ 0 \\ 0 \\ \frac{g_1 K_0}{m} \end{bmatrix} \tag{7.15}$$

$$B_{c_2} = \begin{bmatrix} 0 \\ -(1-g_2)\tau \\ 0 \\ \frac{g_2 K_0}{m} \end{bmatrix} \tag{7.16}$$

$$C_c = \begin{bmatrix} -K_0 & K_0 & g_2 K_0 & 0 \end{bmatrix} \tag{7.17}$$

$$D_{c_1} = -g_1 K_0 \tag{7.18}$$

$$D_{c_2} = -g_2 K_0 \quad . \tag{7.19}$$

In the above, δ_c is the control input, p is the displacement of the structure (in the x direction) at the location of the actuator, and f is the force output of the actuator as applied to the structure. The parameters used in our study are, $\tau = 9.9298 s^{-1}$, $g_1 = 3$, $g_2 = 6$, $K_0 = 83.333$ and $m = 11.4$ kg.

A two–mode model is used to describe the x–z bending plane dynamics of the structure:

$$\dot{x}_s = A_s x_s + B_s^t f^t + B_s^{28} f^{28} \tag{7.20}$$

$$p^i = \Gamma_s^i x_s \tag{7.21}$$

$$y^i = \ddot{p}^i = C_s^i x_s + D_s^{i,t} f^t + D_s^{i,28} f^{28} \tag{7.22}$$

where i takes on the values "t" and "28" and these denote the quantities at the tip and bay 28 locations, respectively.

The equations given above can now be combined to give a state–space model with 12 states. In doing this 0.2 and 0.3% damping was assumed for the structure. A quadratic regulator was designed with second order dynamics admitted for the compensators in the feedback controllers.

7.3 Decentralized Design using Overlapping Decompositions and Loop–Shaping

7.3.1 Background Theory

Consider a system with two decentralized control agents described by

$$\dot{x} = Ax + B_1u_1 + B_2u_2, \tag{7.23}$$

$$y_1 = C_1x \tag{7.24}$$

$$y_2 = C_2x \tag{7.25}$$

where $x \in R^n$ is the state, $u_i \in R^{m_i}$ is the input, and $y_i \in R^{m_i}$ is the output of the i^{th} control agent (i = 1,2). Let the state x be partitioned as

$$x = \begin{bmatrix} x_1 \\ x_0 \\ x_2 \end{bmatrix} \tag{7.26}$$

where $x_i \in R^{n_i}$ ($i = 1,0,2$). The partitioning is usually done in such a way that x_i corresponds to the part of the state space which is *strongly* observable and controllable only by the i^{th} control agent ($i = 1,2$), and x_0 corresponds to the part which is strongly observable and controllable by the both agents [15].

Consider the transformation:

$$\hat{x} = Tx = \begin{bmatrix} x_1 \\ x_0 \\ \cdots \\ x_0 \\ x_2 \end{bmatrix} \in R^{\hat{n}}, \qquad \hat{n} = n + n_0\,. \tag{7.27}$$

The *expansion* of the original system (7.23) with respect to the transformation (7.27) is given by (see [15] for developments)

$$\dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}u, \quad \hat{x}(0) = Tx_0 \in R^{\hat{n}} \tag{7.28}$$

$$y = \hat{C}\hat{x} \tag{7.29}$$

where

$$\hat{A} \triangleq \begin{bmatrix} \hat{A}_1 & \hat{A}_{12} \\ \hat{A}_{21} & \hat{A}_2 \end{bmatrix} = TAT^I + M_A \quad , \tag{7.30}$$

$$\hat{B} \triangleq \begin{bmatrix} \hat{B}_1 & \hat{B}_{12} \\ \hat{B}_{21} & \hat{B}_2 \end{bmatrix} = TB + M_B \quad , \tag{7.31}$$

$$\hat{C} \triangleq \begin{bmatrix} \hat{C}_1 & \hat{C}_{12} \\ \hat{C}_{21} & \hat{C}_2 \end{bmatrix} = CT^I + M_C \quad , \tag{7.32}$$

T^I is a generalized inverse of T satisfying $T^I T = I$, and M_A, M_B, and M_C are complementary matrices satisfying appropriate inclusion conditions. It can be shown that the response $x(t)$ of (7.23) is related to the response $\hat{x}(t)$ of (7.27) by

$$x(t) = T^I \hat{x}(t) \qquad \forall t \geq 0 \quad . \tag{7.33}$$

Furthermore (7.23) and (7.27) have equivalent input/output descriptions

$$\hat{G}(s) \triangleq \hat{C}(sI - \hat{A})^{-1}\hat{B} \equiv G(s) \triangleq C(sI - A)^{-1}B \quad . \tag{7.34}$$

Now consider the *uncoupled expanded model* described by

$$\dot{\tilde{x}} = \tilde{A}\tilde{x} + \tilde{B}u \tag{7.35}$$

$$y = \tilde{C}\tilde{x} \tag{7.36}$$

where

$$\tilde{A} = \begin{bmatrix} \hat{A}_1 & 0 \\ 0 & \hat{A}_2 \end{bmatrix} \tag{7.37}$$

$$\tilde{B} = \begin{bmatrix} \hat{B}_1 & 0 \\ 0 & \hat{B}_2 \end{bmatrix} \tag{7.38}$$

$$\tilde{C} = \begin{bmatrix} \hat{C}_1 & 0 \\ 0 & \hat{C}_2 \end{bmatrix} \tag{7.39}$$

The transfer function matrix (TFM) for the uncoupled expanded model is given by

$$\tilde{G}(s) \triangleq \tilde{C}(sI - \tilde{A})^{-1}\tilde{B} = \text{block diag}\,(G_1(s), G_2(s)) \tag{7.40}$$

where $G_i(s)$ is the TFM of the *local model*

$$\dot{\tilde{x}}_i = \hat{A}_i\tilde{x}_i + \hat{B}_i u_i \tag{7.41}$$

$$y_i = \hat{C}_i\tilde{x}_i \quad , \tag{7.42}$$

for agent i ($i = 1, 2$). Note that this model differs from one that is obtained by simply ignoring the interactions in the original system as it retains portions of the state space which affects each channel.

Let $\hat{G}(s)$ and $\tilde{G}(s)$ be related by

$$\hat{G}(s) = \tilde{G}(s)(I + E(s)) \tag{7.43}$$

where $E(s)$ is the so-called *multiplicative error matrix* between the true TFM $G(s)$ (or equivalently $\hat{G}(s)$) and the *uncoupled expanded TFM* $\tilde{G}(s)$. An upper bound on the norm of $E(j\omega)$ can be found as

$$\bar{\sigma}(E(j\omega)) \leq \frac{\bar{\sigma}(\hat{G}(j\omega) - \tilde{G}(j\omega))}{\underline{\sigma}(\tilde{G}(j\omega))} \triangleq \epsilon(\omega) \tag{7.44}$$

where $\bar{\sigma}(\cdot)$ and $\underline{\sigma}(\cdot)$ indicate, respectively, the maximum and the minimum singular values of the indicated matrices.

In practice, model (7.23)-(7.25) may not represent the physical system exactly, that is there may be some modeling uncertainties present. In such a case, a total error function, $e_m(\omega)$, must be defined, for example as

$$e_m(\omega) = e(\omega)e_c(\omega) + e_0(\omega) \tag{7.45}$$

to represent the error between the uncoupled expended system model and the true system. Here, $e(\omega)$ represents an upper bound on $\bar{\sigma}(E(j\omega))$, such as the one given in (7.44), $e_c(\omega) \geq 1$ accounts for modeling uncertainties in interactions, and $e_0(\omega) \geq 0$ accounts for uncertainties in subsystem models.

Once the total error function is chosen as in (7.45), our approach proceeds with designing decentralized controllers based on local models (7.41)-(7.42). These controllers are designed such that the local stability robustness requirement [15],

$$\bar{\sigma}(T_i(j\omega)) < \frac{1}{e_m(\omega)} \quad , \qquad \forall\omega \tag{7.46}$$

and acceptable performance requirements are satisfied at each local station. Here

$$T_i(s) \triangleq (I + \tilde{G}_i(s)K_i(s))^{-1}\tilde{G}_i(s)K_i(s) \ , \qquad i = 1, 2 \tag{7.47}$$

is the closed-loop design TFM for the i^{th} agent.

7.3.2 Design of the Decentralized Controllers

An initial try at decomposition was made with the dynamics of the individual actuator/sensors assigned to separate subsystems and the two-mode model being kept in the overlapping portion. Thus the subsystem states are described by:

$$\tilde{x}_1 = \begin{bmatrix} x_c^t \\ x_s \end{bmatrix}, \qquad \tilde{x}_2 = \begin{bmatrix} x_s \\ x_c^{28} \end{bmatrix}. \tag{7.48}$$

This is a design choice, although there indeed are guidelines in the selection process. In fact, with a larger model for the structure, portions of the structure model (for example, the "flexible modes") can also be assigned to separate subsystems as based on a measure of effectiveness. In a structure like a beam this would be directly related to the mode-shape values at the individual actuators.

An upper bound on the error introduced due to the interactions between the subsystems is determined as

$$e(\omega) = \frac{\omega}{(\omega^2+1)^{\frac{1}{2}}\left((\frac{\omega}{10})^2+1\right)^{\frac{1}{2}}}. \tag{7.49}$$

Furthermore, it is assumed that the modeling uncertainties can be represented by:

$$e_c(\omega) = \left((\frac{\omega}{10})^2+1\right)^{\frac{1}{2}} \tag{7.50}$$

and

$$e_0(\omega) = \frac{\omega\left((\frac{\omega}{10})^2+1\right)^{\frac{1}{2}}}{((10\omega)^2+1)^{\frac{1}{2}}} \tag{7.51}$$

These functions, together with the final $e_m(\omega)$ given by (7.45), are plotted in Figure 7.1.

With the present decomposition, Linear Quadratic Gaussian controllers were designed for each subsystem to obtain good performance while satisfying the stability-robustness requirements. Gaussian white noise processes of intensity 1 and 100 are assumed to be present respectively at the input and the output of each subsystem. Local quadratic performance indices are chosen with control weightings $\rho_1 = \rho_2 = 100$, and state weightings $Q_i = L_i^T L_i$ $(i = 1, 2)$, where

$$L_1 = \begin{bmatrix} 10^{-3}C_c & 0 \\ 0 & 100\Gamma_s^t \end{bmatrix}, \tag{7.52}$$

and

$$L_2 = \begin{bmatrix} 0 & 10^{-3}C_c \\ 100\Gamma_s^{28} & 0 \end{bmatrix}. \tag{7.53}$$

The resulting control and filtering gains are:

$$K_1 = [.0297 \quad -.0297 \quad -.0.888 \quad .2329 \quad -.2171 \quad .3369 \quad .1076 \quad .0103] \tag{7.54}$$

$$K_2 = [-.0625 \quad .0736 \quad -.4039 \quad -.0384 \quad .0147 \quad -.0147 \quad -.0439 \quad .0409] \tag{7.55}$$

$$H_1 = \begin{bmatrix} -.0764 \\ -.0272 \\ .0104 \\ .0917 \\ -.0233 \\ -.0509 \\ .0361 \\ .0101 \end{bmatrix}, \quad H_2 = \begin{bmatrix} -.0033 \\ -.0315 \\ -.1281 \\ -.0862 \\ -.1445 \\ -.0135 \\ .0101 \\ .01817 \end{bmatrix}. \tag{7.56}$$

Thus the individual controllers are of the form

$$u_i = -K_i z_i \tag{7.57}$$

where

$$\dot{z}_i = (A_i - B_i K_i - H_i C_i) z_i + H_i (y_i - D_i u_i) \quad . \tag{7.58}$$

Magnitudes of the resulting closed-loop TFM's are plotted together with the stability bounds in Figures 7.2 and 7.3. It is observed that the stability robustness requirement (7.46) is indeed satisfied.

As a final check, the controllers were applied to a 4-mode "truth model". The results are summarized in Table 7.1. It is observed that 13% damping in the first and 5% damping in the second mode were achieved. Furthermore this was accomplished with relatively small control and filtering gains. The design is also robust to modeling errors and plant variations. The closed-loop system is guaranteed to remain stable as long as the perturbations are bounded by $\epsilon_m(\omega)$.

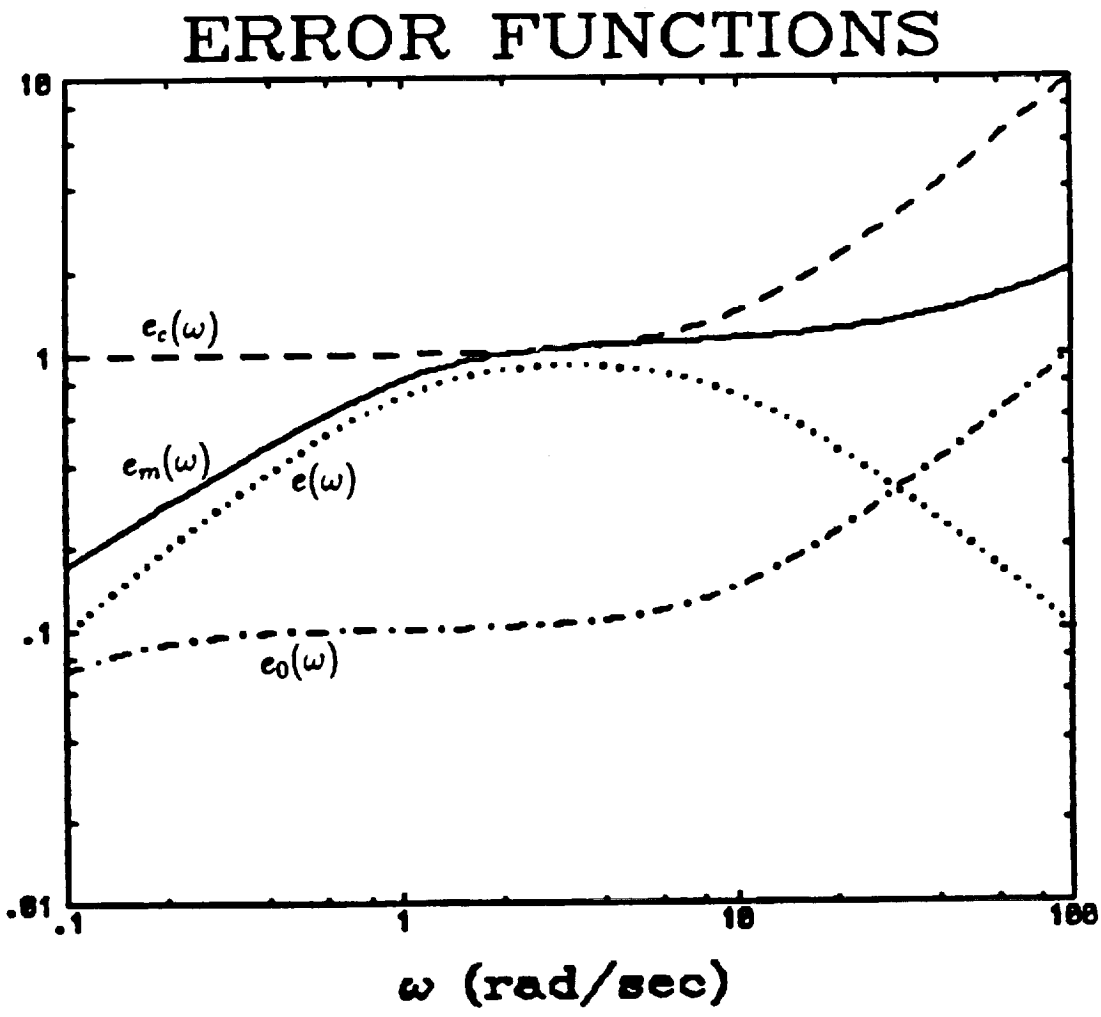


Figure 7.1: Total error function and its components

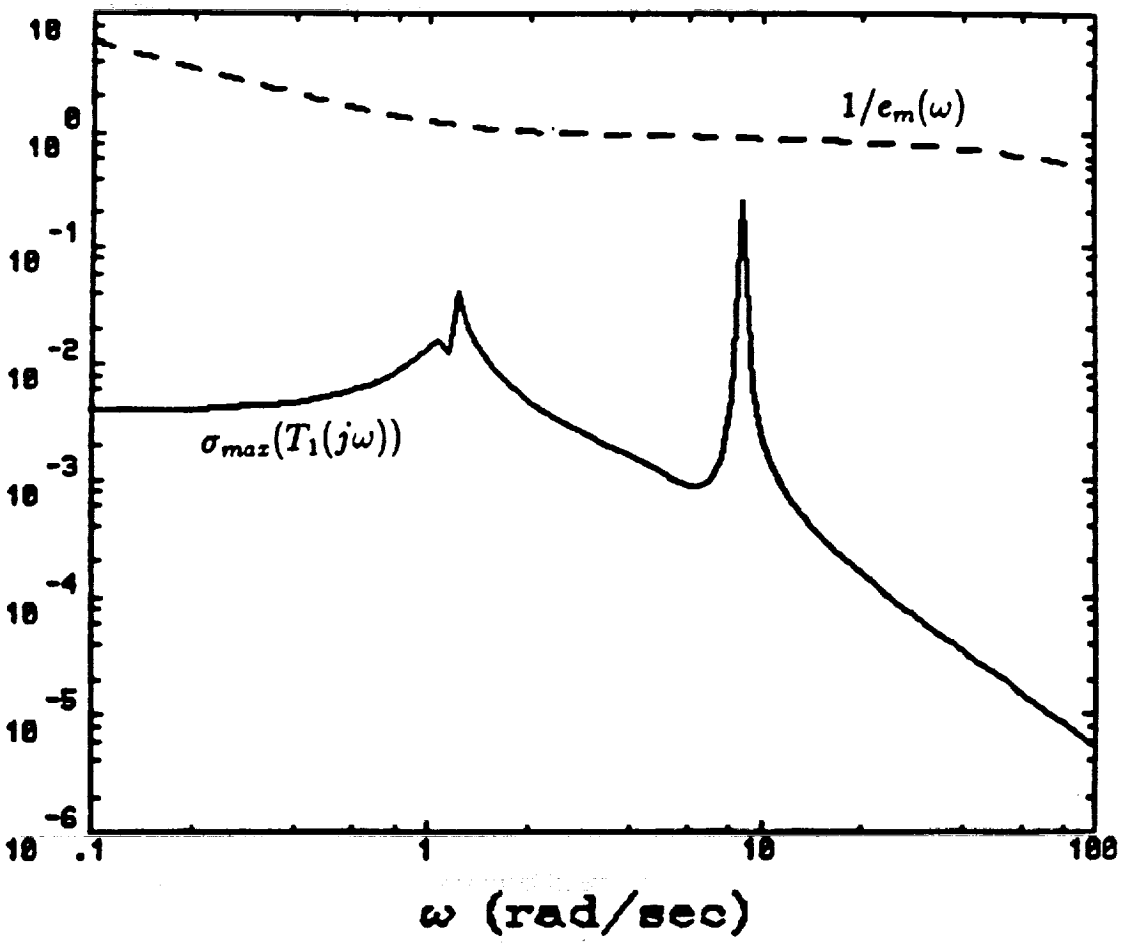


Figure 7.2: Error bound, Maximum singular value (Station 1)

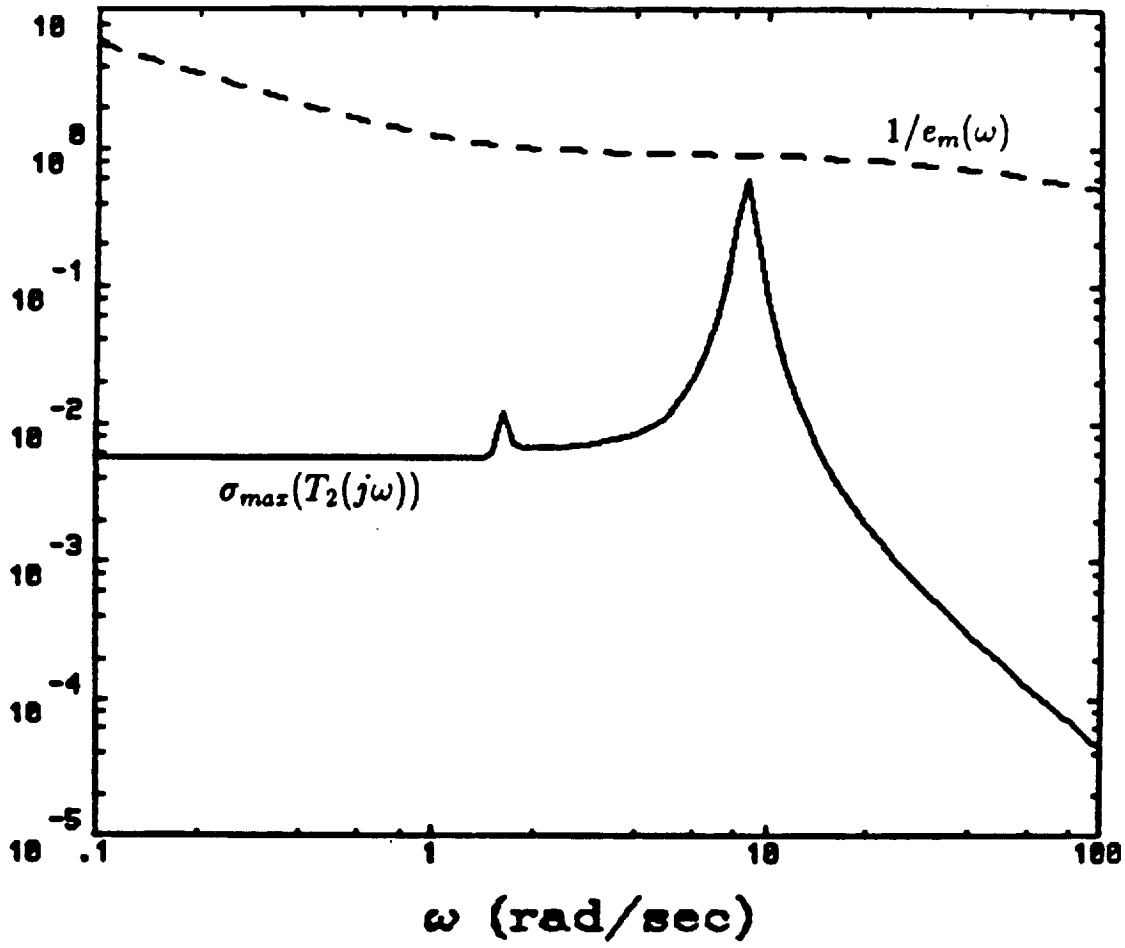


Figure 7.3: Error bound, Maximum singular value (Station 2)

Table 7.1: Decentralized Control Design Results

Mode	*Structural Eigenvalues (damping, %)*			
Freq.	OPEN-LOOP		CLOSED-LOOP	
(Hz)	Without actuators	With actuators	Design Model	"Truth" Model
0.187	$-0.0023 \pm j1.1750$ (0.2)	$-0.0038 \pm j1.1556$ (0.33)	$-0.1511 \pm j1.1724$ (12.78)	$-0.1514 \pm j1.1727$ (12.80)
1.36	$-0.0256 \pm j8.5451$ (0.3)	$-0.1342 \pm j8.6144$ (1.56)	$-0.4862 \pm j8.5773$ (5.66)	$-0.4396 \pm j8.6359$ (5.08)
3.93	$-0.1235 \pm j24.693$ (0.5)	$-0.1248 \pm j24.696$ (0.5)		$-0.1236 \pm j24.6972$ (0.50)
6.84	$-0.2149 \pm j42.977$ (0.5)	$-0.2186 \pm j42.995$ (0.51)		$-0.1954 \pm j43.0042$ (0.45)

7.4 Direct Low–Order Controller Design

In this section we discuss application of the direct (fixed–order) design approach to reduced order controller design for the COFS mast. The details of the optimal projection design methodology, with the maximum entropy stochastic modeling of the uncertainties (MEOP approach), were discussed in earlier in this report. We thus omit further detail here, and refer the reader to Equation (3.1), (3.2) for the general system and controller structure, Equation (3.3) for the cost criterion, and Equations (3.4)–(3.7) for the coupled Riccati and Lyapunov equations to be solved in the MEOP approach.

In the design results to follow, the model employed is as given above, for a full 10 mode representation. We note, however, that the designs of this section do not include the effects of actuator dynamics.

7.4.1 LQG Parameter Selection

For comparison purposes we first present the results of an LQG design (we omit details here) involving the full 20 state model. The design objective is to design compensation to dampen oscillation of the COFS mast structure as much as possible while avoiding high gains. The four design parameters for this exercise are R_1, R_2 (see Equation (3.3)) in the controller Riccati equation and V_1, V_2 in the filter Riccati equation.

After trying several different combinations of the design parameters and conducting simulations, the following parameters were chosen:

$$R_1 = 10^{-2} \begin{bmatrix} 0 & 0 \\ 0 & \tilde{R} \end{bmatrix} \quad , \quad R_2 = 10^{-3} I_{4\times 4} \quad , \tag{7.59}$$

where

$$\tilde{R} = \text{diag} \{ 20,\ 20,\ 10,\ 10,\ 5,\ 5,\ 5,\ 1,\ 1,\ 1 \} \quad , \tag{7.60}$$

and

$$V_1 = \begin{bmatrix} 0 & 0 \\ 0 & I_{20\times 20} \end{bmatrix} \quad , \quad V_2 = 10^{-2} I_{4\times 4} \quad . \tag{7.61}$$

The results of this design (referred to later as the "20th order controller") are presented in comparative plots of the next section.

Table 7.2: LQG & OP Controllers

Order	20	16	14	12	10	8	6
Cost	2.848	2.848	2.849	2.849	2.867	2.895	2.932

Table 7.3: LQG & MEOP Controllers

Order	20	16	14	12	10	8	6
Cost	3.681	3.561	3.561	3.562	3.565	3.568	3.804

7.4.2 Reduced–Order Design Results

For the optimal projection design, Table 7.2 lists the cost for each design, beginning with the full 20th order controller, including OP designs of order 16, 14, 12, 10, 8, and 6. Taking into account the vibrational energy of the mast flight system, controller designs with orders down to about 12 provide acceptable damping relative to the full LQG design; the energy profiles of each controller design, and the open loop case, are given in Figure 7.4, where the energy expression is given by Equation (3.11). For the MEOP designs, where robustification (using the above-mentioned noise intensities) was carried out with respect to the modal frequencies of the first bending modes in all three planes, the results are given in Table 7.4.2 and in Figure 7.5. Again, a 12th order compensator is acceptable considering both the cost and the ability to dissipate energy.

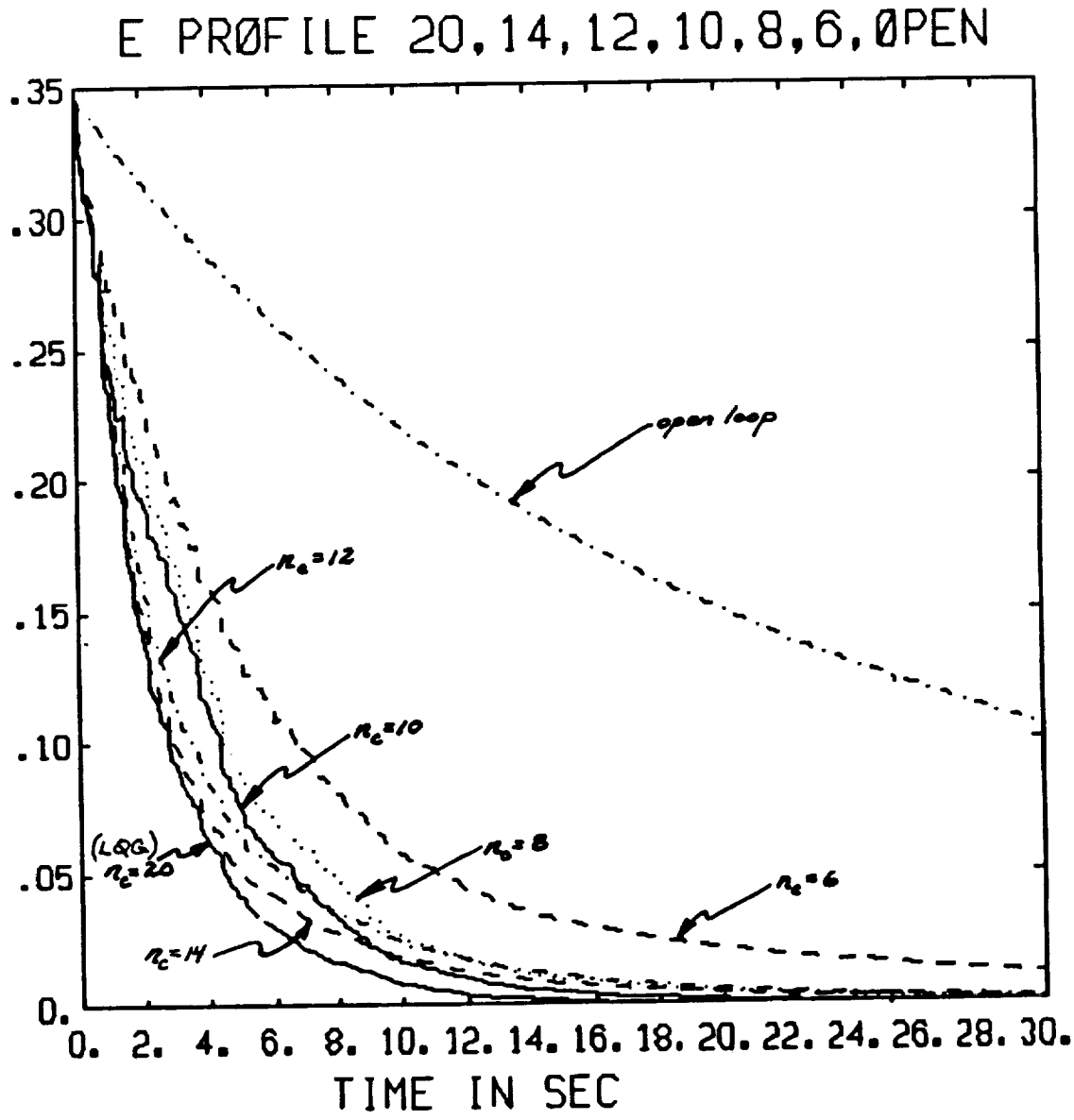


Figure 7.4: LQG & OP Controllers

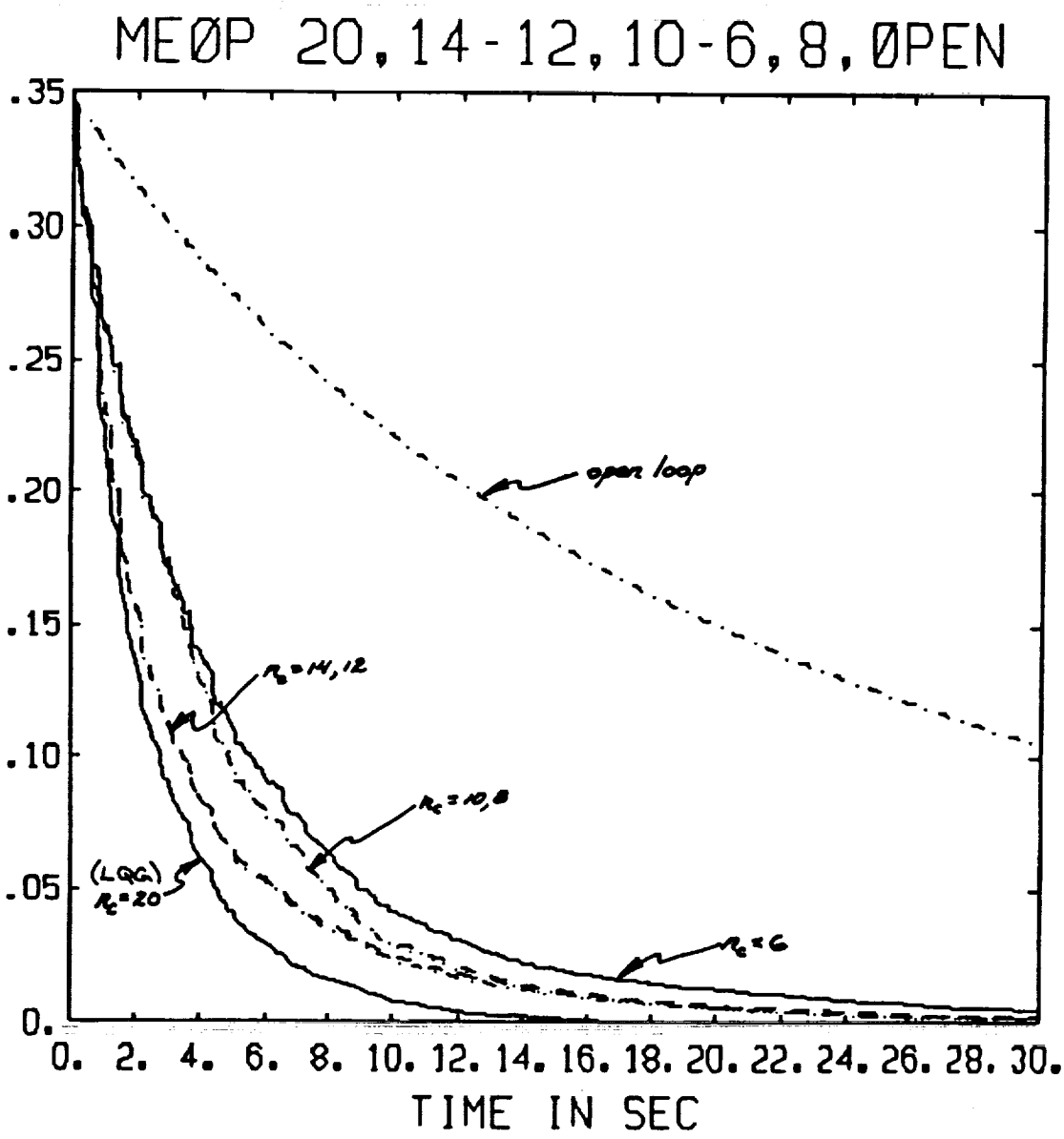


Figure 7.5: LQG & MEOP Controllers

8. FUTURE DIRECTIONS

We have identified several areas to pursue for continuation in this research, not only for the SCOLE configuration but for the MiniMast apparatus as well. These directions for future investigation are summarized below.

Nonadaptive Control Approaches

The issue of spillover or the robustness with respect to unmodeled dynamics is another important subject in large space structures control. Large flexible structures are inherently infinite dimensional, but we can only work on a finite dimensional model due to the constraints in computer capability. In this regard, the system response to unmodeled dynamics should also be investigated in the future. We are, therefore, focusing much of our attention in current and future design efforts towards *robust* controller designs.

To this end, we intend to investigate asymptotic LQG synthesis procedures, such as LQG/LTR, as well as the MEOP design approach. We continue to believe that the MEOP design approach is a very useful tool for control design in the application area of large space structures. This is due to the fact that the compensator order can be reduced significantly from the full order LQG design, allowing easy implementation. A comparison between the various techniques in terms of stability robustness to variations in all system parameters (for SCOLE) should be carried out.

Adaptive Control Approaches

Preliminary results presented in this report have shown that the indirect adaptive LQ controller is initially very effective in vibration damping but effects of unmodeled modes can eventually cause the control input and output to grow without bound. The variable dead zone method and the empirical transfer function estimation scheme are being pursued as viable options for improving the robustness of the indirect adaptive LQ controller.

Another method for improving robustness is to introduce an adaptive filter in order to reduce the effects of control and observation spillover. Methods for designing this filter on-line are currently under investigation. Preliminary results on simple models indicate that the addition of filtering improves both the transient response and the identification of system parameters.

System Identification for Control

The empirical transfer function estimation (ETFE) method is under current investigation for real-time identification of general ARMA model formulations. Using finite length discrete Fourier transforms (DFTs) of the input/output data, the ETFE

method provides both an estimate of the plant transfer function and a frequency–dependent upper bound on the modeling error between the estimated transfer function and the actual plant transfer function. We have obtained new results along these lines for on–line estimation which enables the design of a frequency weighted optimal LQ–adaptive controller. We believe that the ETFE approach is particularly well suited to flexible structure identification and control due to the frequency response profiles typical of such systems.

As for off–line identification techniques, we are investigating experiment design and input synthesis algorithms for flexible structure identification in a related project within the Control Research Laboratory at Ohio State. We therefore hope to incorporate many of those ideas in future work with modeling and identification of SCOLE and MiniMast.

Variable Structure Control

Due to the dynamic coupling of rigid body motion and flexible modes during slew maneuvers, as well as to the inherent complexities resulting from nonlinear dynamics, we believe the application of *Variable Structure Control* to be extremely promising for control of slewing. The SCOLE apparatus and accompanying model offers a suitable testbed for such ideas. We have examined the feasibility of VSC for vibration suppression in our previous work on the NASA Grid and SCOLE, and based on the our experience we propose to investigate such techniques for control of *slewing* on the SCOLE model. Indeed, experiments to date have been performed at NASA/LaRC [16] employing "on–off", bang–bang type controls on the apparatus, and several works have recently appeared advocating such ideas for general rapid slewing, reorientation maneuvers of large space structures [17].

Time Optimal Control

Time Optimal slewing of the SCOLE structure presents some extremely interesting problems. Initial experimental results have been reported by Montgomery *et al* [18] where the attitude control problem for SCOLE was treated as time–optimal uncoupled pitch and roll motions. The model that the basic design was based on included only the rigid body modes; the effects of the vibrational modes were dealt with by an ad–hoc scheme. Recently Barbieri and Özgüner [19] have been able to derive the time–optimal feedback control for systems with both rigid body and purely vibrational modes. The SCOLE configuration will be utilized as an example for evaluating these control strategies.

Experimentation

From our most recent visit to LaRC and subsequent evaluation, we have determined appropriate actions to take for improvement. Since a large portion of the difficulties

experienced centered on the computing power available, particularly evident in the observer–based designs, we will be investigating the following solutions. After noise problem analysis, we propose to install a Butterworth filter to narrow the effective sensor bandwidth and eliminate excess sensor noise and unmodeled dynamics (modal frequencies). In order to speed up computation time, we plan to add state space discretization routines to discretize the continuous time observer equations at runtime. This will essentially result in double to triple the current maximum sample rates allowing us to try the MRAC and MEOP control laws again with a much faster sampling rate. Moreover, we need to delineate the problem with the X–axis rate gyro and take corrective action.

The experimental software program is installed on the SCOLE computer. With the cooperation of a technician at Langley, we are now prepared to run experiments over the phoneline. In the coming months we intend to take advantage of this capability, provided that help is available at LaRC. With regard to the computer setup in the laboratory, we feel that the overall procedure could be improved with the availability of some sort of local plotting routines, or graphics capabilities, to evaluate control law performance at runtime. Also, some sort of diagnostic programs to evaluate system behaviour before any control laws are tried would be a tremendous help.

9. REFERENCES

[1] D. C. Hyland and D. S. Bernstein, "The optimal projection equations for fixed-order dynamic compensation," *IEEE Transactions on Automatic Control*, vol. AC–29, pp. 1034–1037, November 1984.

[2] D. S. Bernstein and D. C. Hyland, "The Optimal Projection/Maximum Entropy approach to designing low-order, robust controllers for flexible structures," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 745–752, Ft. Lauderdale, Florida, December 1985.

[3] M.–F. Cheung and S. Yurkovich, "On the robustness of MEOPdesign versus asymptotic LQG synthesis," *IEEE Transactions on Automatic Control*, vol. AC–33, no. 9, , September 1988.

[4] R. K. Yedavalli, "Perturbation bounds for robust stability in linear system models," *International Journal of Control*, vol. 42, no. 6, pp. 1507–1517, 1985.

[5] K. K. D. Young, "Controller design for a manipulator using theory of variable structure systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-8, pp. 101–109, February 1978.

[6] U. Özgüner, S. Yurkovich, J. Martin, and F. Al–Abbass, "Decentralized control experiments on NASA's flexible grid," in *Proceedings of the 1986 American Control Conference*, pp. 1045–1051, Seattle WA, June 1986.

[7] F. Al-Abbass and U. Özgüner, "Decentralized model reference adaptive system using a variable structure control," in *Proceedings of the 24th Conference on Decision and Control*, pp. 1473–1478, Ft. Lauderdale, Florida, December 11–13, 1985.

[8] K. A. Ossman and E. W. Kamen, "Adaptive regulation of MIMO linear discrete–time systems without requiring a persistent excitation," *IEEE Transactions on Automatic Control*, vol. AC–32, no. 5, pp. 397–404, May 1987.

[9] G. Kreisselmeier, "A robust indirect adaptive–control approach," *International Journal of Control*, vol. 43, no. 1, pp. 161–175, January 1986.

[10] L. Ljung, *System Identification Theory For The User*. Englewood Cliffs, NJ: Prentice–Hall, 1987.

[11] R. LaMaire, L. Valavani, M. Athans, and G. Stein, "A frequency–domain estimator for use in adaptive control systems," in *Proceedings of the American Control Conference*, pp. 238–244, Minneapolis, MN, June 1987.

[12] A. Tzes and S. Yurkovich, "A new approach to frequency domain identification for flexible structure control," *Trans. ASME, J. Meas., Dyn., and Control*, 1988. (submitted).

[13] W. S. Levine, T. L. Johnson, and M. Athans, "Optimal limited state variable feedback controllers for linear systems," *IEEE Transactions on Automatic Control*, vol. AC–16, no. 6, pp. 785–792, December 1971.

[14] J. C. Geromel and J. Bernussou, "Optimal decentralized control of dynamic systems," *Automatica*, vol. 18, no. 5, pp. 545–557, 1982.

[15] A. Iftar and U. Özgüner, "Local LQR/LTR controller design for decentralized systems," *IEEE Transactions on Automatic Control*, vol. AC–32, no. 10, , October 1987.

[16] J. P. Williams, D. Sparks, Y. Shenhar, and R. C. Montgomery, "Attitude control system testing on SCOLE," in *Proceedings of the Sixth VPI & AIAA Symposium on Dynamics and Control of Large Structures*, June 1987. (to appear).

[17] H. Öz and U. Özgüner, "Variable structure system conytol of flexible spacecraft," in *Proceedings of the AIAA/AAS Astrodynamics Conference*, p. , Seattle, WA, 1984.

[18] R. C. Montgomery, J. Shenhar, and J. P. Williams, "On–line identification and attitude control for SCOLE," in *Proc. AIAA Guidance, Navigation and Control Conference*, pp. 950–958, Monterey, CA, August 1987.

[19] E. Barbieri and Ümit Özgüner, "Time–optimal rest–to–rest control of flexible structures," February 1988, (submitted).

PARAMETER IDENTIFICATION FOR VIBRATION CONTROL OF SCOLE

by

Dean W. Sparks, Jr.,
Raymond C. Montgomery,
Robin C. Elder,
and
Danette B. Lenox

NASA Langley Research Center
Hampton, VA 23665-5225

Paper to be presented at the
1988 ASME Winter Annual Meeting

Chicago, Illinois
November 27 - December 2, 1988

PARAMETER IDENTIFICATION FOR VIBRATION CONTROL OF SCOLE

Dean W. Sparks, Jr.*, R. C. Montgomery†, Robin C. Elder‡, and Danette B. Lenox**

NASA Langley Research Center
Hampton, VA 23665-5225

ABSTRACT

This paper documents the linear least-square identification procedure used to obtain an empirical model of the vibrational dynamics of SCOLE, a laboratory apparatus used to test parameter identification techniques and control laws for large, flexible space structures. Testing is done autonomously by exciting the structure from a quiescent state with torque wheels and recording the time history data of rate gyro sensors. The torque wheels are then shut down and free-decay data is recorded. The free-decay portion of the data is analyzed using the Fast Fourier transform to determine the best model order to use in modelling the response. Linear least-square analysis is then used to select the parameters that best fit the output of an Autoregressive (AR) model to the data. The control effectiveness of the torque wheels is then determined using the excitation portion of the test data, again using linear least squares. This report describes the system model assumed and the experimental apparatus and procedures used. Also, typical experimental data are presented that reflect the performance of the identification algorithms.

INTRODUCTION

Control of distributed-parameter systems is a technology that will be required for future spacecraft, but there are both theoretical and practical problems which must be overcome to create an effective design capability within the control systems community. Recognizing this, NASA has built a national research facility, the Spacecraft Control Laboratory Experiment (SCOLE) [1,2], to support this research effort. This facility provides researchers with a highly flexible test article, sensors, actuators, and digital control processing capability. The test article mimics the Space Shuttle with a large, flexible, offset-feed antenna attached to the payload bay (figures 1 and 2).

Much interest has been expressed by the research community concerning SCOLE. This is reflected in the technical output of two workshops held at the NASA Langley Research Center [3,4]. Using SCOLE, control laws for a multi-input/output structural dynamics system can be implemented in real time from any remote site that has a computer terminal and modem communications capability. In order to design effective control systems, the technical community needs a good model. Previously, work has been reported on the empirical determination of the rigid body rotational dynamics of SCOLE (reference 5). To this time, however, there has been no comprehensive testing of its vibration dynamics.

The technique used to identify the dynamics of SCOLE is an extension of the one developed in [6]. The model used to represent the dynamics is an Autoregressive Moving Average (ARMA) model. In [6] the test data was assumed to contain only a single mode and the test input was modified to produce only that mode. That technique is valid for structures wherein that can be achieved. Unfortunately, for SCOLE, generating a response that contains only one mode is very difficult. Thus, a new procedure had to be evolved. The object of this paper is to develop and tutorially describe a method applicable to SCOLE that is

* Aero-Space Technologist, Spacecraft Control Branch.
† Senior Research Scientist, Spacecraft Control Branch.
‡ VPI&SU Aerospace Engineering Co-op, Spacecraft Control Branch.
** Computer System Manager, PRC Kentron, Inc.

capable of handling multiple modes in the test data.

The report is organized as follows: first, the experimental apparatus and the models used to represent the vibration dynamics are described. Then, the parameter identification procedure and computational algorithms are presented. Next, typical examples of the experimental work are presented. Presentation and discussion of these experimental results complete the report.

SCOLE APPARATUS

The SCOLE hardware and support software is described in detail in [2]. In this report, only those elements of SCOLE used in this research are described. Referring to the schematic diagram, figure 2, SCOLE contains two major structural elements of interest: a planar, hexagonal, tubular structure representing an antenna reflector; and a single tubular flexible mast connecting the antenna to the platform. For this experiment, the platform is fixed to ground; only the mast and antenna portions are dealt with in this paper. The system actuators consist of three torque wheels that produce torque in three mutually orthogonal directions (see figures 1 and 2).

Experiments are run on SCOLE using a digital computer that has a UNIX-like operating system called UNOS. Programming is accomplished in a combination of C and FORTRAN 77 programs. The computer has analog-to-digital converters used for sampling the rate gyro data, digital-to-analog converters used to command the torque wheels, and a process timer which achieves precise internal timing of the data sampling process. This equipment has been added to the original system along with software drivers which can be evoked from either C or FORTRAN 77 programs.

SYSTEM MODELS

The model we seek for SCOLE should incorporate the actual natural frequencies, damping factors, and control effectiveness coefficients of the system. To this end, each mode of the vibrational dynamics of SCOLE is modeled as a single-input, single-output system described by

$$\dot{x} = Ax + Bu \tag{1}$$

wherein

$$A = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\zeta\omega \end{bmatrix}, \quad \text{and} \quad B = \begin{bmatrix} 0 \\ b \end{bmatrix} \tag{2}$$

x - modal state vector
u - control input vector
b - modal parameter of actuator location
ω - natural frequency of mode
ζ - damping ratio of mode

The output is of the form

$$y = Hx \tag{3}$$

For a rate sensor, $H = [0\ c]$, c is the mode slope at the sensor location.

For digital control implementation, the control input is assumed constant over the uniform sample time interval of T seconds, and the continuous-time model is converted to its discrete-time equivalent by integration of equation (2) over the interval. Thus, the difference equation describing the motion appears as

$$x_{k+1} = \Phi x_k + \Gamma u_k \tag{4}$$

where $\Phi = e^{AT}$, $\Gamma = \int^T e^{At} dt\ B = (\Phi - I)A^{-1}B$ since A is nonsingular. The matrix I is a 2x2 identity matrix.

An autoregressive form of the discrete time model may be accomplished by taking the z-transform of the equation (4) and solving for the sampled sensor output, y_k, in terms of the input actuator

$$y_k = a_1 y_{k-1} + a_2 y_{k-2} + (\tilde{b}_1 u_{k-1} + \tilde{b}_2 u_{k-2})b \tag{5a}$$

where

$$a_1 = \Phi_{11} + \Phi_{22}$$

$$a_2 = \Phi_{12}\Phi_{21} - \Phi_{11}\Phi_{22}$$

$$\tilde{b}_1 = \Gamma_{22}$$

$$\tilde{b}_2 = \Phi_{21}\Gamma_{12} - \Phi_{11}\Gamma_{22}$$

$$b = c \cdot b$$

Taking the z transform of equation (5a) and arranging like terms (and letting the a coefficients absorb their respective signs) results in the output equation form

$$(z^2 + a_1 z + a_2)Y(z) = (\tilde{b}_1 + \tilde{b}_2)U(z)b \tag{5b}$$

The above derivations are for the single mode case. A more general form of the sampled output equation, which includes multiple modes, is presented. This case still deals with single input, single output only. It can be shown that a decoupled multiple mode state space system can be written in block diagonal form, using the A matrix of equation (2) as the block elements for each mode. Because of the diagonal property of this system and the fact that the corresponding output matrix has the H matrix of equation (3) as elements, the z transformed multiple mode output equation can be written in the following compact form

$$d(z)\ Y(z) = \sum_{i=1}^{N} b_i\ (n_1^i z + n_2^i) \prod_{\substack{j=1 \\ j\neq i}}^{N} d_j(z)\ U \qquad (6)$$

where $d(z) = \prod_{j=1}^{N} d_j(z)$, $d_j(z) = z^2 + a_1^i z + a_2^i$

and N is the number of modes. The super- and subscripted quantities refer to the ith and jth modes and the b_i terms are the modal control effectiveness coefficients of the torque wheel. The $a_{1,2}^i$ coefficients are the AR coefficients of the ith mode and $n_{1,2}^i$ can be directly computed given $a_{1,2}^i$. Thus, the model satisfies

$$(z^{2N} + \alpha_1 z^{2N-1} + \ldots + \alpha_{2N} z^0)\ Y =$$

$$\sum_{i=1}^{N} (\beta_{i,1} z^{2N-1} + \beta_{i,2} z^{2N-2} + \ldots + \beta_{i,2N} z^0) b_i U \qquad (7)$$

where: $\alpha_1, \ldots, \alpha_{2N}$ - coefficients of d(z)

$\beta_{i,1}, \ldots, \beta_{i,2N}$ - coefficients of the product of righthand side of equation (6)

This is the model form used to represent the dynamics for each test conducted herein.

PARAMETER IDENTIFICATION PROCEDURES

Linear least-square estimation (LSE) is used to identify the difference equation (7) parameters. This method was selected because of its computation efficiency and implementation simplicity [7,8]. The identification process is carried out for each mode and each actuator. A structural excitation test is conducted wherein the structure, initially at rest, is sinusoidally forced by a single actuator at the predicted natural frequency of the mode of interest. The actuator is turned off when the output of the sensor of interest reaches a predetermined magnitude. The data recording is continued to obtain free-decay data.

The test data is processed in a two-step operation. For the first step, the AR coefficients ($\alpha_1, \alpha_2, \ldots$) of the model are identified using the free-decay portion of the data. For the second step, the identified AR coefficients are used to obtain the control effectiveness coefficients ($b_1, b_2, \ldots$) with data taken from the excitation portion.

Step 1 - Identification of the Free Decay Coefficients

First, the spectral content of the free-decay portion of the data is examined to determine the number of modes present in the data. The order of the model is taken to be twice the number of modes determined. Then, the free-decay data is digitally filtered to suppress noise an the signal due to any modes not wanted in the model. Finally, the filtered data is processed using the standard least-squares estimation to identify the AR coefficients for the number of modes selected for modelling.

The identification of the AR ($a_{1,2}^i$) coefficients for each mode generally depends on the data base used in the estimation. As more significant data is added, the estimates should converge to a value and the variance of the estimates will improve to a limit, which depends on the measurement noise and the model. After this "convergence" occurs the mean and variance of the estimates should remain constant. Therefore, the variation of the estimates is examined as data is added to the data base and the mean of the estimates is taken over the last several data base additions. To ascertain confidence in the estimates, the variance of the estimates is also checked.

The natural frequency and damping factor are obtained from a_1 and a_2 by finding the roots z of its characteristic equation and using the relation $z=e^{sT}$ in the primary strip, where T is the uniform sampling period. The following equations for the natural frequency and damping factor are obtained for both the overdamped and underdamped cases:

Overdamped -- $a_1^2/4 + a_2 > 0$ (8a)

$$s_{1,2} = \frac{1}{T} \ln\left[\frac{a_1}{2} \pm \sqrt{\frac{a_1^2}{4} + a_2}\right] \qquad (8b)$$

$$\omega = \sqrt{s_1 s_2} \qquad (8c)$$

$$\zeta = -(s_1+s_2)/2\omega \qquad (8d)$$

Underdamped -- $a_1^2/4 + a_2 < 0$ (9a)

$$\sigma = \frac{1}{2T} \ln(-a_2) \qquad (9b)$$

$$\omega_d = \frac{1}{T} \tan^{-1}\sqrt{-1 - 4a_2/a_1^2} \qquad (9c)$$

$$\omega = \sqrt{\omega_d^2 + \sigma^2} \qquad (9d)$$

$$\zeta = -\sigma/\omega \qquad (9e)$$

Step 2 - Identification of Actuator Control Effectiveness Parameters

Once the values of the AR coefficients for the mode of interest are determined, a similar linear least-squares scheme used in [6] is employed to obtain the control effectiveness of the torque wheel actuator with respect to the mode of interest.

From equation (7), an error equation can be defined as

$$e_k = y_k + \sum_{m=1}^{2N} \alpha_m y_{k-m} - \sum_{i=1}^{N} (\beta_{i,1} u_{k-1} + \ldots + \beta_{i,2N} u_{k-2N}) b_i \qquad (10)$$

To calculate the b_i's by least squares, the parameter J, defined in equation (11), is minimized with respect to the b_i's. The J parameter represents the total error for all the samples taken.

$$J = 1/2 \sum_{k=1}^{s} e_k^2 \qquad (11)$$

s - total number of k samples

The resulting least square equation is in matrix form, shown in equation (12).

$$V = P^{-1} Q \qquad (12)$$

V - Nx1 vector, with b_i's as elements

P - NxN matrix, with element P_{ij}

$$= \sum_{k=1}^{s} \left[\sum_{m=1}^{2N} \beta_{i,m} u_{k-m} \circ \sum_{m=1}^{2N} \beta_{j,m} u_{k-m} \right]$$

Q - Nx1 vector, with element Q_i

$$= \sum_{k=1}^{s} \left[(y_k + \sum_{m=1}^{2N} \alpha_m y_{k-m}) \circ \sum_{m=1}^{2N} \beta_{i,m} u_{k-m} \right]$$

For the case of a single mode, the above equation should be the same as the least square equation in [6].

RESULTS AND DISCUSSIONS

A finite-element analysis of SCOLE has been conducted to calculate the modes and frequencies of interest. Figure 3 shows the first vibration mode shape and frequency and that of the third mode. The first mode is essentially bending. The second mode is also a bending mode and looks similar to mode 1 except that the bending is orthogonal to that of mode 1. The third mode is a torsional rotation of the reflector about the mast. The remaining modes are higher in frequency and appear similar to the classical Euler bending modes of a flexible beam. The predicted natural frequency of the first mode is .43 Hz. This is the frequency used during the excitation portion of the testing for the mode 1 tests. Figure 4a shows the excitation portion of the mode 1 test with the x-axis (the 1-axis of figure 3) torque wheel. The data was sampled at 50 samples/second and the actuator commands were updated at this same frequency. The free-decay portion is shown in figure 4b. Although the time scales both start at 0, the free-decay portion is, in fact, a direct extension of the excitation portion.

Step 1 -- The free-decay portion of the data run is first analyzed to determine the number of modes present in the data. Figure 5a is a Fast Fourier transform of the free-decay data. The transform is only plotted to the Nyquist frequency. Experience has shown that the accommodation of noise in a finite data base does not allow one to estimate to the Nyquist frequency but to about one decade down. In this case about 5 Hz. Since the data includes a significant peak at around 10 Hz, it should be filtered to remove this mode and the noise in the signal. The sensor data was digitally filtered using an 8th order Butterworth filter with the cutoff frequency at 3 Hz. This was taken to provide significant attenuation of the 10 Hz peak in the unfiltered spectrum. Figure 5b shows the spectral content of the filtered data. Indeed, the 10 Hz. mode has been removed and three modes remain. The separation of the magnitudes of these remaining modes is a factor of 200. This means that extraction of the coefficients of the AR model for all three modes will be difficult with a limited data base. Hence, we select a two mode model for the purpose of estimating the dynamics of the free-decay data. The ARMA model is thus of order 4.

Least square estimation is used to determine the AR parameters from the data. The variation of the α_i coefficients as data is added to the data base for estimation is shown in figure 6. The abscissa is interpreted as follows: the last point (100) corresponds to the estimate with the entire free-decay data base (this includes 1465 data points sampled at 50 samples/sec.) and the first point (0) corresponds to a data base of 1365 points, approximately 27 seconds. This variation is typical and the other two α parameters have a similar variation. The mean and variance of the parameters are:

$\alpha_1 = 3.8403 \quad var(\alpha_1) = 8.567(10)^{-8}$

$\alpha_2 = -5.6792 \quad var(\alpha_2) = 6.787(10)^{-8}$

$\alpha_3 = 3.8363 \quad var(\alpha_3) = 6.220(10)^{-8}$

$\alpha_4 = -0.9980 \quad var(\alpha_4) = 6.650(10)^{-9}$

The characteristic polynomial corresponding to these coefficients has factors which correspond to mode natural frequencies and damping ratios of (see equations 8 and 9):

$\omega_1 = .4610$ Hz $\quad \zeta_1 = .0011$

$\omega_2 = 3.1479$ Hz $\quad \zeta_2 = .0024$

Step 2 -- The data for the frequency and damping ratio obtained in step 1 has been used in the algorithm of [6]. Examination of the excitation portion of the test data (figure 4b), reveals the presence of a disturbance induced when the torque wheel is started. It is believed that this is the result of start-up friction in the torque wheel assembly. This disturbance disappears in about 7 seconds. Hence, this part of the data was not included in the data base for estimation. The variation of the least square control effectiveness estimate as the last 400 samples are added to the data base is shown in figure 7. For the lowest number of points in the data base (0 on the abscissa) the estimate is seen to not have converged and more data is needed. As points are added the estimate statistics converge. It is important to use the mean taken over a converged portion of figure 7 and not a single point. The value of the control effectiveness mean taken over the 400 data points of figure 7 is .0171 and the variance is $1.0576(10)^{-6}$.

SUMMARY

This paper describes the linear least-square identification procedure used to obtain an empirical model of the vibrational dynamics of SCOLE. Empirical data is obtained autonomously by exciting the structure from a quiescent state with torque wheels and observing the time history data of rate gyro sensors. The torque wheels are then shut down and free-decay data is recorded. The free-decay portion of the data is analyzed using the digital Fourier transform to determine the best model order to use in modelling the response. Linear least-square analysis is then used to select the parameters that best fit the output of an autoregressive (AR) model to the data. The control effectiveness of the torque wheels is then determined using the excitation portion of the test data, again using linear least squares. Typical experimental data are presented that reflect the performance of the identification algorithms.

REFERENCES

1. Taylor, L. W., Jr. and A. V. Balakrishnan: A Laboratory Experiment Used to Evaluate Control Laws for Flexible Spacecraft ... NASA/IEEE Design Challenge. Proceedings of the Fourth VPI&SU Symposium on Dynamics and Control of Large Structures, Blacksburg, VA, June 1983.

2. Williams, J. P. and R. A. Rallo: Description of the Spacecraft Control Laboratory Experiment (SCOLE) Facility. NASA TM-89057, February 1986.

3. Taylor, L. W. (Compiler): Spacecraft Control Laboratory Experiment, SCOLE Workshop -- Proceedings of a Workshop Concerning the NASA/IEEE Design Challenge. NASA Langley Research Center, December 6-7, 1984.

4. Taylor, L. W. (Compiler): 2nd Annual SCOLE Workshop -- Proceedings of a Workshop Concerning the NASA Design Challenge. NASA TM 89048, December 9-10, 1985.

5. Montgomery, R. C., Shenhar, J., and J. P. Williams: On-Line Identification and Attitude Control for SCOLE. Proceedings of the AIAA Guidance, Navigation, and Control Conference, August, 1987.

6. Montgomery, R. C. and T. Lazarus: Recent Developments in the Experimental Identification of the Dynamics of a Highly Flexible Grid. ASME Paper Number 87-WA/DSC-19, presented at the 1987 ASME Winter Annual Meeting, December 13-18, 1987.

7. Montgomery, R. C. and Sundararajan, N.: Identification of the Dynamics of a Two-Dimensional Grid Using Least-Square Lattice Filters. The Journal of the Astronautical Sciences. Vol. 33, No. 1, January-March 1985, pp, 35-47.

8. Montgomery, R. C., Williams, J. P., Lazarus, T. L., and Nelson, P.E.: Control Effectiveness Characterization for State Estimation and Control on a Highly Flexible Grid. Proceedings of the AIAA Guidance, Navigation and Control Conference, August 18-20, 1986.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

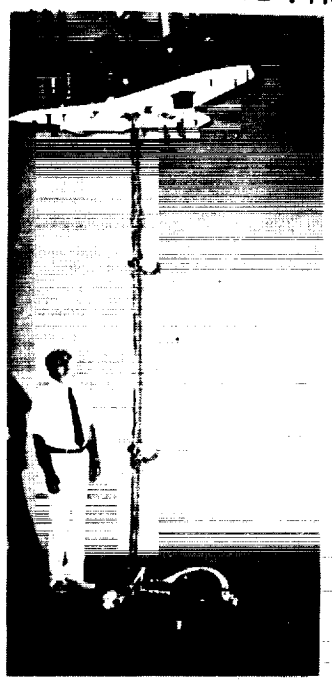


Figure 1.- Photograph of the Spacecraft COntrol Laboratory Experiment - SCOLE

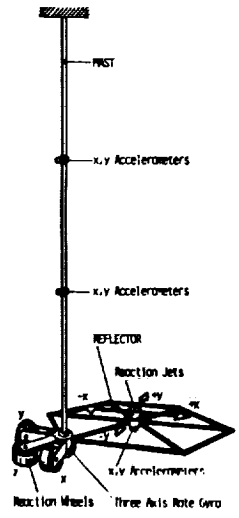


Figure 2.- Schematic of SCOLE

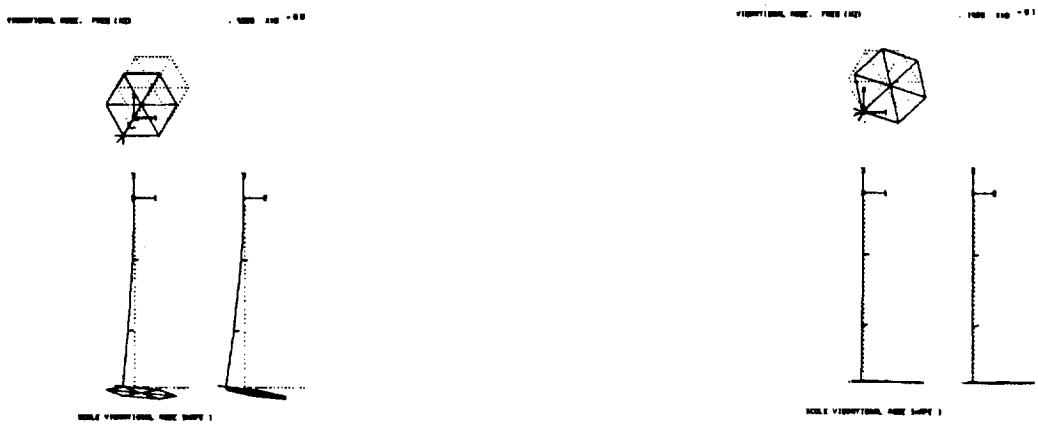


Figure 3.- First bending and torsional mode shapes and frequencies.

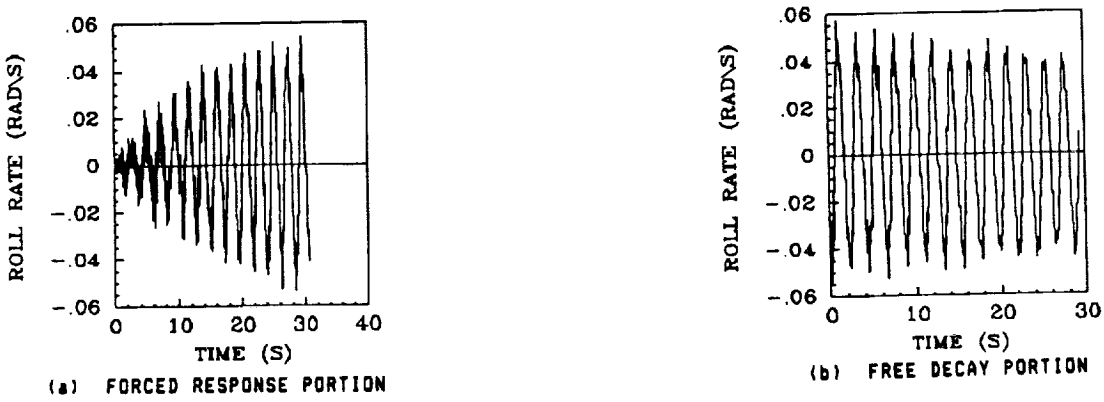


Figure 4.- Typical torque wheel test data including the excitation and free-decay portions.

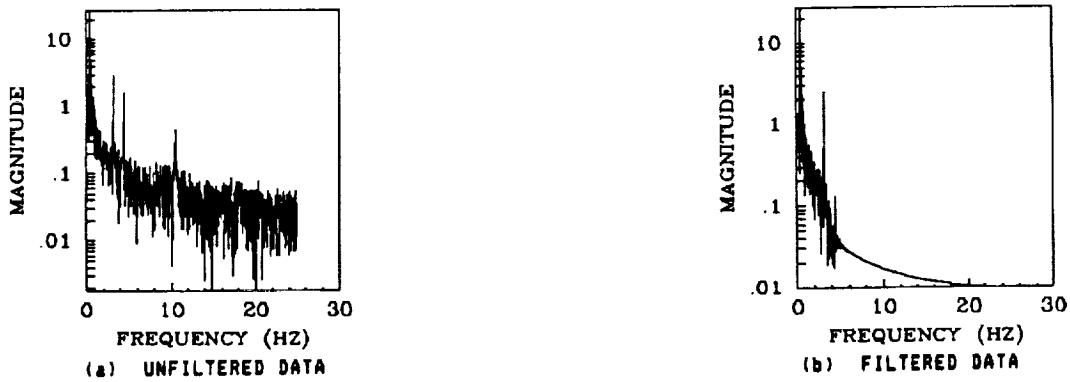


Figure 5.- Fast Fourier transform magnitude plots of the rate gyro output during the free-decay portion.

Figure 6.- Sample AR parameters identified using a two-mode model.

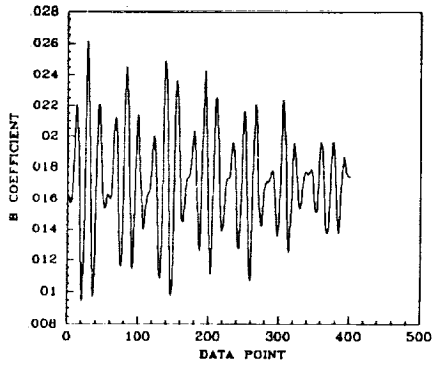


Figure 7.- The control effectiveness parameter for the first mode using an average of the AR parameters of a single mode model.

Identification and Control of Large Flexible Spacecraft

Sue Harris

Y. P. Kakad

University of North Carolina at Charlotte
Charlotte, N. C.

555

IDENTIFICATION AND CONTROL OF LARGE FLEXIBLE SPACECRAFT

Sue Harris

Y. P. Kakad

Department of Electrical Engineering
University of North Carolina at Charlotte
Charlotte, N. C. 28223

ABSTRACT

Future NASA missions will depend on significantly large, flexible, and complicated structures in outer space. These structures will require very stringent pointing and vibration suppression requirements. The design of the active controllers to achieve these objectives requires accurate knowledge of the dynamic behavior of these structures. The structures are lightly damped distributed parameter systems and have an infinite number of frequencies of vibration. This paper presents a system identifcation technique that represents the structural dynamics in terms of a reduced-order modal model. The finite-element method is used to reduce the order of the physical model, and then the model is transformed from a physical to a modal form. The maximum likelihood method is used to parameterize the modal model, based on experimental observations of the structural dynamics.

NOMENCLATURE

$\underline{q}(t)$	Generalized Coordinate Vector
$\underline{u}(t)$	Input Vector
$\underline{y}(t)$	Predicted Output
M	Mass Matrix
K	Stiffness Matrix
B_p	Input Influence Matrix
C_p	Output Influence Matrix
Ω	Eigenvalue Matrix
Φ	Eigenfunction Matrix
$\underline{\omega}$	Modal Frequency
$\underline{\eta}(t)$	Modal Displacement Vector
$\ddot{\underline{\eta}}(t)$	Modal Acceleration Vector
$\underline{\Theta}(t)$	Vector of System Parameters
ζ	Damping Ratio
$J(\underline{\Theta})$	Cost Function
T	Period of Oscillation
$\ddot{\underline{y}}(t)$	Predicted Acceleration Vector
$\underline{z}(t)$	Observed Acceleration Vector
G	Covariance Matrix of Noise
$\underline{p}(z)$	Position Vector of Instrument on Mast
L	Length of the Mast
ρ	Density of the Mast
A_m	Cross-sectional Area of the Mast
E	Modulus of Elasticity of the Mast
I	Area Moment of Inertia of the Mast
G_ψ	Shear Modulus of the Mast
M_1	Mass of the Shuttle End of the Mast
M_2	Mass of the Reflector End of the Mast
J_1	Mass Moment of Inertia of the Shuttle Mass
J_2	Mass Moment of Inertia of the Reflector Mass
I_ψ	Polar Moment of Inertia of the Mast
$\nabla J(\underline{\Theta})$	Jacobian Matrix
$H(\underline{\Theta})$	Hessian Matrix

INTRODUCTION

Many future NASA missions would utilize significantly large, flexible and very complicated structures in outer space. These large space structures would require very stringent pointing and vibration suppression requirements. The active controller that can acheive these objectives will have to be designed with very accurate knowledge of the dynamic behavior of the structure to ensure performance robustness to a variety of disturbances and uncertainties. It is recognized by control engineers that there are certain inherent problems in the design of active controllers for this class of large flexible space structures. A space structure is a very lightly damped distributed parameter system and hence has an infinite number of frequencies of vibration. A large number of these frequencies have extremely low values and they are very closely spaced together. Additionally, numerical solutions of the higher frequencies tend to yield large errors in their calculations. Thus, it is a challenge to design a state-of-the-art multivariable control system for a lower order model of a large flexible space structure.

Several techniques for designing control systems for vibration suppresion for flexible spacecraft have been proposed, and various computer and experimental studies have been conducted to validate these control schemes. An experimental test article was

designed under the cognizance of the Spacecraft Control Branch at NASA Langley Research Center for testing additional methods and for directly comparing the various competing techniques on a common experimental model (SCOLE).

This test article consists of a softly supported dynamic model of a flexible beam and reflector grillage attached to the space shuttle. The control objective includes the task of directing the line-of-sight (LOS) of the antenna-like configuration toward a fixed target, under conditons of limited control authority and random disturbances.

IDENTIFICATION OF SCOLE ASSEMBLY

The first task in the control system design is that of system identification. The dynamic behavior of the experimental model must be described analytically. The finite-element method can be used as a structural analysis method to define the reduced-order model for the infinite-dimensional SCOLE test assembly. The finite- element method represents the continuous body as a finite number of discrete basic elements that each possess mass and stiffness properties that can be assembled to form mass and stiffness matrices.

System identification combines experimental observations and theoretical predictions to form an accurate model of an unknown system. The observations are used to quantify the unknown model parameters and to minimize the error between the observed system dynamics and the model-predicted dynamics.

The SCOLE test assembly can be modeled as a free-free beam with rigid end masses. The objective of the identification process is to describe the dynamics of the antenna and mast with respect to the shuttle body. The observations that will be used to parameterize the system are the linear and angular acceleration rates of the mast and antenna. The available data consists of two sets of measurements of mast accelerations in the x and y planes, one set of measurements of the x and y accelerations of the reflector, and the angular acceleration about x, y, and z measured at the junction of the mast and antenna. Only the motion of the antenna and mast with respect to the shuttle body is of interest so that the pitch and roll of the antenna adequately describes the motion in the z-direction.

The data set thus describes the system as one that can be represented by ten degrees-of-freedom. The frame of reference is chosen so that the origin is at the universal joint suspension point of the test article with the z-axis running along the center of the mast. The desired form of the model is one that transforms the physical properties of mass and stiffness into the modal properties of natural frequency and damping ratio. The ten degrees-of-freedom of the physical model results in a truncated modal model with five modes.

BASIC ANALYTICS

The structural damping of the assembly is very light and is neglected in the physical model. The model can thus be written as

$$M\ddot{\underline{q}}(t) + K\underline{q}(t) = B_p \underline{u}(t) \tag{1}$$

$$\underline{y}(t) = C_p \underline{q}(t) \tag{2}$$

where $\underline{q}(0) = \underline{q}_0$, $\dot{\underline{q}}(0) = \dot{\underline{q}}_0$ are the initial conditions.

Let $\Omega =$ diagonal $\left|\omega_1^2 \ \omega_2^2 \ \ldots \ \omega_n^2\right|$ be the eigenvalue matrix corresponding to the physical model and Φ be the corresponding eigenfunction matrix of the eigenvalue problem.

It can be shown that

$$M\Phi\Omega = K\Phi. \tag{3}$$

Now normalize Φ so that

$$\Phi^T M \Phi = I \tag{4}$$

and let

$$\underline{q}(t) = \Phi\underline{\eta}(t). \tag{5}$$

This results in the modal transformation

$$\ddot{\underline{\eta}}(t) + \Omega\underline{\eta}(t) = B_m \underline{u}(t) \tag{6}$$

$$\underline{y}(t) = C_m \underline{\eta}(t) \tag{7}$$

where $B_m = \Phi^T B_p$ and $C_m = C_p\Phi$, and the initial conditions are $\underline{\eta}_0 = \Phi^{-1}\underline{q}_0$, $\dot{\underline{\eta}}_0 = \Phi^{-1}\dot{\underline{q}}$. The modal equation is equivalent to the physical equation in the sense that it produces the same input-output description of the structure. The viscous damping can be modeled in terms of ζ_i, the damping ratio of the ith mode. Including the damping and reordering the state variables, the dynamic model can be written as

$$\dot{\underline{\Theta}} = A\underline{\Theta}(t) \tag{8}$$

$$\underline{Y}(t) = C\underline{\Theta}(t). \tag{9}$$

$$\underline{\Theta} = \left|\dot{\eta}_1 \ \eta_1 \ \dot{\eta}_2 \ \cdots \ \dot{\eta}_n \ \eta_n\right|^T$$

where $\dot{\eta} = \omega =$ natural frequency vector and $\eta = \zeta =$ damping ratio vector.

$$A = \begin{vmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\omega_1^2 & -2\zeta_1\omega_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -- & -- & 0 & 0 \\ 0 & 0 & -- & -- & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -\omega_n^2 & -1\zeta_n\omega_n \end{vmatrix}$$

$$B = \left| 0 \ (B_m)_1 \ 0 \ (B_m)_2 \ \dots \ (B_m)_n \right|^T$$

$$C = \left| (C_m)_1 \ 0 \ (C_m)_2 \ 0 \ \dots \ 0 \right|$$

where $\underline{\Theta}(t)$ is the vector of unknown parameters to be adjusted.

ALGORITHM DEVELOPMENT

The system identification algorithm uses the maximum likelihood method to improve the accuracy of the finite-element model. The model is assummed to be of the same form as the true model and converges to the true value as the parameters converge to their actual values. The cost functional , $J(\underline{\Theta})$, is minimized so that the error betwen the model-predicted output, $\underline{y}(t)$, and the observed output, $\underline{z}(t)$, approaches zero. The algorithm uses the modified Newton-Raphson method to minimize the error.

The cost functional, $J(\underline{\Theta})$ can be expressed as

$$J(\underline{\Theta}) = \frac{1}{2T}\int_0^T \left|\underline{z}(t) - \underline{\ddot{y}}(t)\right|^T G^{-1} \left|\underline{z}(t) - \underline{\ddot{y}}(t)\right| dt \quad (10)$$

Let

$$V = \left|\underline{z}(t) - \underline{\ddot{y}}(t)\right|$$

and the individual component of vector $\underline{\ddot{y}}(t)$ is given as

$$\ddot{y}_i(t) = -\omega_i^2(K_i)(C_{1i}\sin\omega_i t + C_{2i}\cos\omega_i t). \quad (11)$$

For lateral vibrations

$$K_i = A_i\sin\beta_{1i}\epsilon_j + B_i\cos\beta_{1i}\epsilon_j + C_i\sinh\beta_{1i}\epsilon_j + D_i\cosh\beta_{1i}\epsilon_j$$

and for torsional vibrations

$$K_i = E_i\sin\beta_{2i}\epsilon_j + F_i\cos\beta_{2i}\epsilon_j$$

where the subscript j indicates the sensor locations along the z axis.

The physical properties of the test assembly are used to evaluate the output vector $\underline{\ddot{y}}(t)$ as follows

$$\epsilon_j = \frac{p_j}{L}$$

$$\beta_{1i} = \omega_i^{1/2} \cdot L\left(\frac{\rho \cdot A_m}{E \cdot I}\right)^{\frac{1}{4}}$$

$$\beta_{2i} = \omega_i \cdot L\left(\frac{\rho}{G_\psi}\right)^{1/2}. \quad (12)$$

The coefficients A, B, C, D, E, and F are evaluated from the boundary conditions and C_1 and C_2 are evaluated from the initial conditions. The first coefficient of the spatial parameters (A, E) is set equal to one and the resulting system is reduced so that the following system of equations is obtained

$$a_i = \frac{M_1 \cdot \beta_{1i}}{\rho \cdot A_m \cdot L} \quad (13)$$

$$b_i = \frac{M_2 \cdot \beta_{1i}}{\rho \cdot A_m \cdot L}$$

$$c_i = \frac{J_1 \cdot exp\,(3ln\,(\beta_{1i}))}{\rho \cdot A_m \cdot exp\,(3ln\,(L))}$$

$$d_i = b_i\cos\beta_{1i} + \sin\beta_{1i}$$

$$e_i = b_i\sinh\beta_{1i} + \cosh\beta_{1i}$$

$$f_i = b_i\cosh\beta_{1i} + \sinh\beta_{1i}$$

$$g_i = \cos\beta_{1i} - b_i\sin\beta_{1i}$$

$$A_i = 1.0$$

$$B_i = \frac{g_i + g_i a_i c_i - e_i + c_i e_i a_i + 2c_i f_i}{d_i + a_i + c_i a_i d_i + c_i a_i^2(1 - e_i) + e_i a_i + f_i - c_i a_i f_i}$$

$$D_i = \frac{-2c_i + B_i(1 - c_i a_i)}{1 + c_i a_i}$$

$$C_i = 1 + a_i(B_i + D_i)$$

$$E_i = 1.0$$

$$F_i = \frac{-\rho \cdot L \cdot I_\psi}{(\beta_{2i})^2 \cdot J_1}.$$

The first measurements in each data set , $\underline{z}(t_1)$, are used to evaluate the initial modal conditions and the coefficients of the modal parameter as follows

$$\eta_{0i} = \frac{z_i(t_1)}{\dot{\eta}_i} = \frac{z_i(t_1)}{\omega_i}$$

$$\dot{\eta}_{0i} = \frac{z_i(t_1)}{-\dot{\eta}_i^2} = \frac{z_i(t_1)}{-\omega_i^2}$$

$$C_{1i} = \eta_{0i} \tag{14}$$

$$C_{2i} = \frac{\dot{\eta}_{0i} + \eta_i \dot{\eta}_i \eta_{0i}}{\dot{\eta}_i (1 - \eta_i^2)^{1/2}} = \frac{\omega_{0i} + \zeta_i \omega_i \zeta_{0i}}{\omega_i (1 - \zeta_i^2)^{1/2}}.$$

The cost function, $J(\underline{\Theta})$, is minimized by adjusting the vector of system parameters, $\underline{\Theta}$, by an iterative method in which the new parameter estimate, $\underline{\Theta}_{n+1}$, is computed on the basis of the present estimate $\underline{\Theta}_n$.

$$\underline{\Theta}_{n+1} = \underline{\Theta}_n - H^{-1}\underline{\Theta}_n \nabla J(\underline{\Theta}_n) \tag{15}$$

where the gradient of the cost function, $\nabla J(\underline{\Theta}_n)$ is

$$\nabla J(\underline{\Theta}_n) = \left| \frac{\partial J(\underline{\Theta})}{\partial \Theta_1} \quad \cdots \quad \frac{\partial J(\underline{\Theta})}{\partial \Theta_n} \right|^T \tag{16}$$

$$\frac{\partial J(\underline{\Theta}_n)}{\partial \Theta_i} = \frac{-1}{T} \int_0^T \left| \frac{\partial \underline{y}(t)}{\partial \Theta_i} \right|^T G^{-1} V_i \, dt \quad i = 1,2,\ldots,10$$

and the Hessian, $H(\underline{\Theta})$ is

$$H(\underline{\Theta}) = [h_{ij}(\underline{\Theta})] \quad i = 1,2,\ldots,10 \tag{17}$$

$$h_{ij}(\underline{\Theta}) = \frac{1}{T} \int_0^T \left| \frac{\partial \underline{y}(t)}{\partial \Theta_i} \right|^T G^{-1} \left| \frac{\partial \underline{y}(t)}{\partial \Theta_j} \right| dt$$

$$\frac{\partial \underline{y}(t)}{\partial \Theta_k} = \frac{\partial [K_i \eta_i]}{\partial \omega_i} = K_i \frac{\partial \eta_i}{\partial \omega_i} + \frac{\partial K_i}{\partial \omega_i} \eta_i.$$

$$k = 1,2,\ldots,10 \quad i = 1,2,\ldots,5$$

Let

$$s = (1 - \zeta_i^2)^{1/2}$$

$$\eta_i = e^{(-\zeta_i \omega_i t)}(C_{1i} \cos\omega_i s_i t + C_{2i} \sin\omega_i s_i t) + R_i \tag{19}$$

where R_i is a constant.

$$\frac{\partial \eta_i}{\partial \omega_i} = K_i [-\zeta_i t e^{-\zeta_i \omega_i t}(C_{1i} \cos\omega_i s_i t + C_{2i} \sin\omega_i s_i t) +$$

$$e^{-\zeta_i \omega_i t}(C_{1i} \sin\omega_i s_i t + C_{2i} \cos\omega_i s_i t)] \tag{20}$$

$$\frac{\partial K_i}{\partial \omega_i} = \frac{\partial K_i}{\partial \beta_i} \cdot \frac{\partial \beta_i}{\partial \omega_i}$$

For lateral vibrations $i = 1,2,3$

$$\frac{\partial K_i}{\partial \omega_i} = \epsilon(A_i \cos\beta_{1i}\epsilon - B_i \sin\beta_{1i}\epsilon + C_i \cosh\beta_{1i}\epsilon +$$

$$D_i \sinh\beta_{1i}\epsilon) \frac{L}{2\omega_i^{1/2}} \left| \frac{\rho \cdot A_m}{E \cdot I} \right|^{\frac{1}{4}}$$

while for torsional vibrations $i = 4,5$

$$\frac{\partial K_i}{\partial \omega_i} = \epsilon(E_i \cos\beta_{2i}\epsilon - F_i \sin\beta_{2i}\epsilon) L \left| \frac{\rho}{G_\psi} \right|^{1/2}.$$

$$\frac{\partial y_j}{\partial \Theta_k} = \frac{\partial (K_j \eta_j)}{\partial \omega_j} = K_j \frac{\partial \eta_j}{\partial \zeta_j} \tag{22}$$

$$k = 1,2,\ldots,10 \quad j = 1,2,\ldots,5$$

$$\frac{\partial \eta_j}{\partial \zeta_j} = K_j [-\omega_j t e^{-\zeta_j \omega_j t}(C_{1j} \cos\omega_j s_j t + C_{2j} \sin\omega_j s_j t)$$

$$+ e^{-\zeta_j \omega_j t}(C_{1j} \frac{\zeta_j \omega_j t}{s_j} \sin\omega_j s_j t$$

$$- C_{2j} \frac{\zeta_j \omega_j t}{s_j} \cos\omega_j s_j t]. \tag{23}$$

APPLICATION TO EXPERIMENTAL DATA

The algorithm is used to evaluate eight sets of data and each set represents different initial conditions in order to excite vibrations corresponding to roll, pitch, and yaw motions. The system identfication is carried out for each data set to yeld the modal model. Each data set is a sequence of 1200 observations that were sampled at a 0.025 s sampling rate. The parameters are evaluated for each observation and then the resulting set of parameters are evaluated using statistical analysis to define the dominant natural modes.

CONCLUSIONS

The identification technique presented in this paper has been found to be very effective in modeling the SCOLE test article and for active controller designs. The method has some convergence problems but has the potential to be extremely useful for solving the system identification problem of other large flexible spacecrafts.

REFERENCES

1. L. W. Taylor, Jr. and A. V. Balakrishnan, "A Mathematical Problem And A Spacecraft Control Experiment (SCOLE) Used to Evaluate Control Laws for Flexible Spacecraft... NASA/IEEE Design Challenge," NASA Langley Research Center, Spacecraft Control Branch (unpublished), January 1984.

2. Jeffrey P. Williams and Rosemary A. Rallo, "Description of the Spacecraft Control Control Laboratory Experiment (SCOLE) Facility," NASA TM 89048, December, 1985.

3. Nghia Chi Nguyen, "System Identification of Space Structures," Thesis, University of North Carolina at Charlotte, 1988.

4. D. K. Robertson, "Analysis of Lateral and Torsional Vibration Characteristics of Beams and Shafts With End Located Rotational Masses," NASA TM 84593, May 1984.

5. Y. P. Kakad, "Dynamics of Spacecraft Control Laboratory Experiment (SCOLE) Slew Maneuvers," NASA CR 4098, October, 1987.

6. Y. P. Kakad, "Slew Manuver Control of the Spacecraft Control Laboratory Experiment (SCOLE)," Proceedings of AIAA Guidance, Navigation and Control Conference, pp 629-634, August 1986.

7. Harold W. Sorenson, Parameter Estimation: Marcel Dekker, Inc, New York, 1980.

8. Paul L. Meyer, Introductory Probability and Statistical Applications: Addison-Wesley Publishing Co, Mass., 1965.

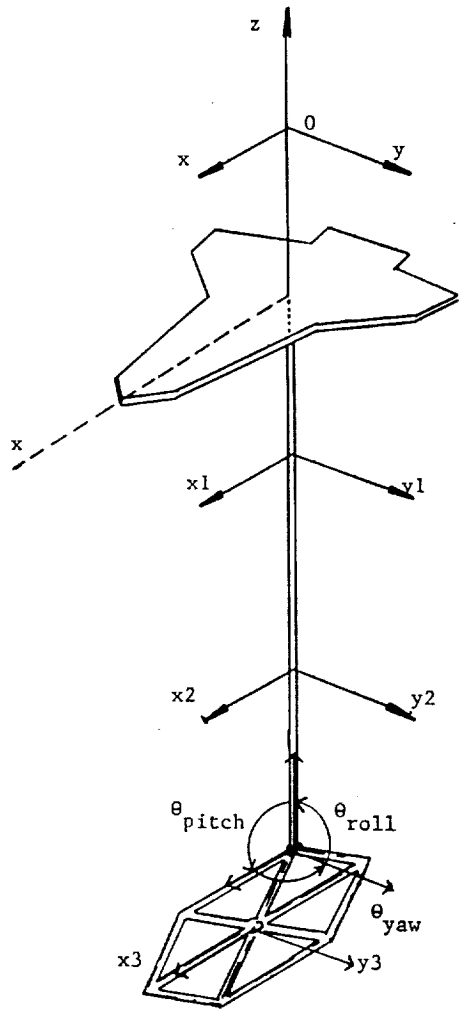


Figure 1

SCOLE Assembly and Measured Output Locations

DESCRIPTION OF THE SPACECRAFT CONTROL LABORATORY EXPERIMENT (SCOLE) FACILITY

Jeffrey P. Williams and
Rosemary A. Rallo

561

SUMMARY

The Spacecraft Control Laboratory Experiment is a facility for the investigation of control techniques for large flexible spacecraft. The control problems to be studied are slewing maneuvers and pointing operations. The facility implements the salient characteristics of a flexible satellite with distributed sensors and actuators.

The flexible satellite is represented by a continuous structure consisting of a large mass and inertia connected to a small mass and inertia by a slender, flexible beam. The structure is suspended by a single cable mounted to a universal joint at the system C. G. The sensors for the experiment consist of aircraft quality rate sensors and servo-accelerometers. The shuttle attitude will be determined through a combination of inertial measurements and optical sensing techniques. Actuators for the experiment consist of Control Moment Gyros, reaction wheels, and cold gas thrusters. Computational facilities consist of micro-computer-based central processing units with appropriate analog interfaces for implementation of the primary control system, the attitude estimation algorithm and the CMG steering law. Details of the experimental apparatus and the system software are presented in this paper.

ABSTRACT

A laboratory facility for the study of control laws for large flexible spacecraft is described in the following paper. The facility fulfills the requirements of the Spacecraft Control Laboratory Experiment (SCOLE) design challenge for a laboratory experiment, which will allow slew maneuvers and pointing operations. The structural apparatus is described in detail sufficient for modelling purposes. The sensor and actuator types and characteristics are described so that identification and control algorithms may be designed. The control implementation computer and real-time subroutines are also described.

INTRODUCTION

A modelling and control design challenge for flexible space structures has been presented to the technical community by the NASA and IEEE (ref. 1). The Spacecraft Control Laboratory Experiment (SCOLE) was constructed to provide a physical test bed for the investigation and validation techniques developed in response to the design challenge. The control problems to be studied are slewing maneuvers and pointing operations. The slew is defined as minimum time maneuver to bring the antenna line-of-sight (LOS) pointing to within an error limit of the pointing target. The second control objective is to rotate about the line of sight and stabilize about the new attitude while keeping the LOS error within the bound δ. The SCOLE problem is defined as two design challenges. The first challenge is to design control laws, using a given set of sensors and actuators, for a mathematical model of a large antenna attached to the space shuttle by a long flexible mast. The second challenge is to design and implement the control laws on a structural model of the system in a laboratory environment. This report gives preliminary specifications of the laboratory apparatus so that interested investigators may begin design and simulation for the laboratory experiment.

The laboratory experiment shown in figure 1 attempts to implement the definition of the modelling and control design challenge within reasonable limits of the 1-g atmospheric environment. The experimental facility exhibits the essential SCOLE characteristics of a large mass/inertia (space shuttle model) connected to a

small mass/inertia (antennae reflector) by a flexible beam. Control sensors and actuators are typical of those which the control designer would have to deal with on an actual spacecraft. Some trades are made in terms of structure, sensors, actuators, and computational capability in order to develop the experiment in a timely and cost-effective manner. To this end, the basic structure is made of homogeneous, continuous elements. It is suspended from a steel cable with the positive z-axis of the shuttle pointing up, thus minimizing the static bending of the antenna mast. The suspension point is a two-degree-of-freedom gimbal for pitch and roll with yaw freedom supplied by the suspension cable. The sensors are aircraft quality rate sensors and servo-accelerometers. The shuttle attitude will be determined through a combination of inertial measurements and optical sensing techniques.

The shuttle control moments are provided by a pair of two-axis control moment gyros (CMG's). Mast-mounted control torques can be applied by a pair of two-axis reaction wheels. The reflector-based forces are provided by solenoid-actuated cold-air thrusters. Reflector mounted torque devices are a trio of high-authority reaction wheels. Computational facilities consist of micro-computer-based central processing units with appropriate analog interfaces for implementation of the primary control system, the attitude estimation algorithm, and the CMG steering law. All of the elements which make up the SCOLE experiment are described in detail in the following text.

The description of the apparatus covers five major groups: The basic structural elements are described and pertinent dimensions and structural properties are provided. The sensor locations and their dynamic properties are presented. The actuator locations and estimated dynamic properties are also given. The mass properties of the combined structure, sensor and actuator system are given. Finally, the computing system and analog interfaces are described.

The contents of this report are considered accurate at the time of publication. All of the planned SCOLE components are implemented and are available to the user at a raw signal level. However, due to continued refinement of some of the components, specific details of the system may change over the life-time of the experimental apparatus.

STRUCTURES

The SCOLE is comprised of three basic structures, the shuttle, the mast, and the reflector panel. The assembly of these individual components and the global reference frame are shown in figure 2.

The shuttle planform is made from a 13/16-inch steel plate and has overall dimensions of 83.8 by 54.0 inches. Its total weight is 501.7 pounds. The shuttle's center-of-mass is located 3.4 inches below the experiment's point of suspension, and 26.8 inches forward of the tail edge (fig. 3).

The mast is 120 inches long. It is made from stainless steel tubing and weighs 4.48 pounds. One-inch thick manifolds are mounted to the mast at each end. The assembly of these parts and their dimensions are shown in figure 4.

The reflector panel is hexagonal in shape, made from welded aluminum tubing, and weighs 4.76 pounds (fig. 5). It is located 126.6 inches below the SCOLE's point of suspension. The center of the reflector is located at 12.0 inches in the x direction and 20.8 inches in the y direction from the end of the mast.

The complete system is suspended from an 11-foot cable attached at the system center-of-gravity via a universal joint. Roll and pitch rotational freedom is provided by pillow-block ball bearings which have an estimated break-out torque of 0.1 ft-lb. The universal joint is shown in figure 6. It is fixed to the shuttle plate, and the system center-of-gravity is made to coincide with the center-of-rotation by means of an adjustable counter balance system.

SENSORS

The sensors for the experiment consists of nine servo-accelerometers and two, 3-axis rotational rate sensing units. An optical sensor will provide yaw attitude of the shuttle. The power supplies for these sensors are mounted on the shuttle plate to minimize the number of large gauge wires which must cross the universal joint suspension point. Only a single 115 VAC cable and 33 signal wires cross the universal joint. The wires for the sensors are routed on the shuttle and along the mast.

Accelerometers

All nine accelerometers have a frequency response which is nearly flat up to 350 Hz. Linearity is within 0.17 percent of the full-scale output. A typical calibration is presented in figure 7. Individual calibrations are available on request.

The shuttle-mounted accelerometers shown in figure 8a sense the x, y, and z accelerations. These sensors are distributed away from the suspension point to aid inertial attitude estimation. The locations and sensitive axis are shown in figure 8b.

The mast-mounted accelerometers shown in figure 9a sense x and y acceleration at locations about one-third of the mast length from each end. The positions and sensing axis of the devices are shown in figure 9b.

The reflector-mounted accelerometers are shown in figure 10a. They are positioned in the center of the reflector below the thrusters and sense the x and y accelerations. The coordinates and sensing axis of the devices are shown in figure 10b.

Rate Sensors

The rotational rate sensors are three-axis, aircraft-quality instruments. The frequency response is approximately flat to 1 Hz and -6 db at 10 Hz. Linearity is about 0.6 percent full scale. A typical calibration is shown in figure 11. The range is 60 deg/sec for the yaw and pitch axis and 360 deg/sec. for roll. The threshold is 0.01 deg/sec.

The shuttle-mounted rate sensor package, shown in figure 12a, senses three-axis, rigid body angular rates of the shuttle plate. Its coordinates and sensing axis are presented in figure 12b.

The mast-mounted rate sensor package, shown in figure 13a, senses three-axis angular rates at the reflector end of the mast. Its coordinates and sensing axis are presented in figure 13b.

The sensor information required for control system design is summarized in Table I. The sensor type is listed in column 2, and its sensed variable is listed in column 3. The analog interface channel is listed next. The coordinates of the parts with respect to the universal joint are listed in the next three columns. The sensitivities in terms of analog-to-digital converter units are listed next. The error list shows the RMS deviation of the rate sensors or the percentage of full-scale linearity error for the accelerometers. The linear range of the instruments is listed in the next-to-last column.

OPTICAL SENSOR

An optical sensor will be provided to determine yaw attitude of the shuttle. The optical sensor is a planar photo-diode with appropriate optics mounted on the ground. The outputs of the sensor are proportional to the position of an infared light source on the shuttle. The sensor data is processed by a dedicated micro-controller and then sent to the main CPU over a serial data link for transformation to attitude angles. No photographs or calibration data are available for this device.

ACTUATORS

The actuators consist of both proportional and on-off controllers. Shuttle attitude control is provided by a pair of two-axis control moment gyros (CMG's). Mast vibration suppression can be achieved with a pair of orthogonally mounted reaction wheel actuators positioned at two stations on the mast. Reflector forces are provided by four cold gas jets. Reflector torques are provided by three orthogonally mounted reaction wheels at the end of the mast. As with the sensors, all devices are inertial, and the power supplies and amplifiers are mounted on the shuttle. Fifteen command signal wires cross the universal joint. All actuators were manufactured in house.

Control Moment Gyros

The CMG's each have two gimbals which are equipped with individual direct drive DC torque motors. The momentum wheel is mounted in the inner gimbal and driven by two permanent magnet DC motors. The nominal operational momentum is about 2.5 ft-lb-sec. The gimbal torque motors are driven by current amplifiers so the output torque will be proportional to the command voltage sent to the amplifier. The gimbal torquers will produce +/- 1.5 ft-lbs at frequencies up to 1kHz. The gimbals are instrumented with tachometers and sine-cosine poteniometers to facilitate decoupled control of the shuttle attitude angles. A dedicated computer will be used to control the CMG gimbals. Routines will be provided so that users may command decoupled shuttle torques or gimbal torque commands.

The sensitivity calibration curve of a typical gimbal motor is shown in figure 14. No other calibration data are available for the CMG's.

The forward CMG is shown in figure 15a. Note that the outer gimbal is fixed and parallel to the pitch axis of the shuttle. The inner gimbal is nominally oriented so that the spin axis of the momentum wheel is parallel to the shuttle z-axis. The second CMG is mounted at the rear of the shuttle so that the outer gimbal is

parallel to the z-axis. The inner gimbal is nominally oriented so that the rotor spin axis is parallel to the shuttle x-axis. The coordinates and nominal axis of actuation of the CMG's are shown in figure 15b.

Reaction Wheels

The mast-mounted reaction wheels consist of aluminum disks with inertia of about 0.00027 lb-ft-sec^2 mounted directly on the drive shaft of a 20 oz-in permanent magnet DC motor. The motors are powered by high bandwidth current amplifiers. A torque sensitivity plot is presented in figure 16. No other calibration data are available. A typical reaction wheel assembly is shown in figure 17a. The two actuator locations and their axis of actuation are shown in figure 17b.

The mast end mounted reaction wheels consist of D C permanent magnet pancake motors which are mounted with the armature fixed to the structure. The stator and case of the motor are allowed to rotate via a slip ring assembly, thus providing high inertia mass to fixed mass efficiency.

The motors are powered by high bandwidth current amplifiers. The torque capability of these devices is estimated to be about 50 oz-in. No sensity plot or other calibration data is presently available. The three-axis reaction wheel assembly is shown in figure 18a. The actuator locations and their axis of actuation are shown in figure 18a.

Thrusters

The control forces on the reflector are provided by solenoid actuated cold gas jets. The thrusters are mounted in the center of the reflector and act in the x-y plane. The jets are supplied by a compressed air tank mounted on the shuttle. The pressurized air travels through the mast to the solenoid manifold, which gates the air flow between the regulated supply tank and the thrusters as shown in figure 19. Thrust is initiated by opening the solenoid with a discrete command. The rise time and transient oscillation of thrust is shown in figure 20. The magnitude and duration of the thrust before the air supply is depleted at 60-psi nozzle pressure is shown in figure 21. The pertinent data from figures 20 and 21 are tabulated in Table II.

The thrusters are shown in figure 22a. Their location and axis of actuation are shown in figure 22b.

The actuator information required for control system design is summarized in Table III. The actuator type and direction of action are listed in Column 2. The analog interface channels are listed in Column 3. The coordinates of the devices are listed in the next three columns. The sensitivities of the actuators in terms of digital-to-analog converter units are shown in Column 7. An estimate of the thruster RMS deviations exhibited in figure 19 is presented as error data. No other error data are available. The maximum range of the system actuators is shown in the next-to-last column.

MASS PROPERTIES

The position and weight of the various pieces of equipment, which collectively form the SCOLE apparatus, are cataloged in Table IV. Distances are measured from the point of suspension to the approximate center-of-mass of each component. Each major component is listed in the second column of the table. The x, y, z coordinates are listed next. The weight of each component is listed in Column 6. The remaining columns are the mass moments and moments of inertia. The totals for the complete system are presented on the bottom row.

COMPUTER SYSTEM

The main computer for control law implementation will be a micro-computer based on the Motorola M68000 microprocessor. The computer has 2.0 M-byte of random access memory and a 40 M-byte hard disk. The operating system is based on UNIX with C, Fortran and Pascal compilers available for applications programming. The computer has 12 serial ports and 1 parallel port. Terminals are connected on four of the ports and an answer-only modem is attached to another. One port is used for an originate-only modem. A line printer is attached to another port. The optical sensor is connected to a serial port. The IBM PC, which is used to drive the CMG's, is also connected to a serial port.

Analog interfaces consist of a four-bit, output-only discrete channel: an 8 bit discrete output port, an 8-bit discrete input port, 8 digital-to-analog converters, and 64 analog-to-digital converters. All converters are 12-bit devices with a range of +/-10v. These interfaces are shown schematically in figure 23. The CMG control software required for the PC should be relatively transparent to the controls designer who will be operating on the CRDS computer.

Subroutines for accessing the analog interfaces and setting the digital sampling interval are described in Appendix A. The most commonly used routines are listed below.

For accessing the analog devices:

- getadc - read the analog-to-digital converters
- setdac - set the digital-to-analog converters
- thrust - set the cold-gas thrusters.

To control the sampling interval:

- rtime - sets the sample period marks the beginning of a real-time loop.

A time-line of the synchronization of the sample interval using the routine rtime and the analog interface routine usage is shown in figure 24. The basic operation is as follows:

> The user first calls rtime with flag=.true., and a valid sample interval. After setting the timer period, the routine starts the user's real-time routine and the interval clock. The user routine will use some or all of the subroutine calls shown. When the user computations and actuator commands are complete, the routine must return to the top of the real-time loop and once again call rtime. If this occurs before the end of the sample interval, the time-out condition will be inhibited and rtime will wait for the next rising edge of the sample interval clock and then return

to the calling program. If the user computations take longer than the sample interval, a time-out condition will be signaled to the operator when rtime is called. The user may choose to ignore the condition and continue or may take specific steps to alleviate the condition.

The procedure for logging on to the computer is as follows:
Set communication parameters to 1200 baud, 7 bit, even parity.
Dial in to the Langley data communication switching system at 804-865-4037. When connected, type a carriage return.
To the system prompt "ENTER RESOURCE CODE" type "acrl."
Wait until "GO" and the "name:" prompt appear on the screen.
Type in your log-in information. All investigators will be given a three-letter log-in name (usually the university affiliation).

Some useful system commands are listed below:
To transfer a file to the experiment computer from a smart terminal, type **cat> filename <cr>** and then enable the upload function of the local terminal. When the upload is complete, type **<cntl>d.**
To transfer a file from the experiment computer to a smart terminal, type **cat filename** then enable the download function of the local terminal and type a **<cr>.**

To list the contents of a directory, type **l <cr>.**

To look at a file, type **p filename.**

To compile a FORTRAN program and link with real-time system commands and

TCS graphics, type **frt filename.**

Note: **filename** must have the extension .for. The executable code will be under filename without the .for extension. To run, simply type **filename** without any extension.

To list the system commands, type **l /bin.** to get a description of any command, type **describe command.**

System user guides will be available upon request from the Spacecraft Control Branch, M/S 161, NASA Langley Research Center, Hampton, VA, 23665-5225. Other details for operating in a real-time mode will be provided at the time of implementation.

CONCLUDING REMARKS

The SCOLE laboratory facility is an experimental apparatus which permits ground-based investigation of identification and control algorithms for large space structures. The facility exhibits structural dynamics similar to those expected on the large satellites. The sensors and actuators are typical of those, which may be used on an operational satellite. The computational system is reasonably sized with current technology processors and permits ready access to the facility for interested investigators.

The description of the structural assembly, the sensor and actuator configuration, and software provided in this paper should be sufficient for SCOLE investigators to begin designing identification and control algorithms for the SCOLE facility.

APPENDIX

REAL-TIME SYSTEM SUBROUTINES

Analog I/O system command.

NAME:

getadc Samples the analog-to-digital converters.

COMMAND:

getadc [-s]

DESCRIPTION:

This command is used to sample the analog-to-digital converters and display selected channels at the terminal. The +/- 10.0 volt input range is scaled to +/- 1.0 units so a single bit is worth .00049 units. The channels to be displayed are selected with the -s option. This option displays a menu which allows the user to set the print flags for the individual ADC channels. Specific choices are:

1) turn on all print flags,
2) turn off all print flags,
3) turn on a range of print flags,
4) turn off a range of print flags,
5) display current print flags,
6) save current print flags.

If the command is executed without the -s option, the last set of print flags is used to selectively display the ADC channels.

USES:

getadc.flags

DIRECTORY:

/bin

SOURCE:

/usr/csc/ele/getadc.c

Analog I/O system command.

NAME:

getadc Samples the analog-to-digital converters.

COMMAND:

getadc [-s]

DESCRIPTION:

This command is used to sample the analog-to-digital converters and display selected channels at the terminal. The +/- 10.0 volt input range is scaled to +/- 1.0 units so a single bit is worth .00049 units. The channels to be displayed are selected with the -s option. This option displays a menu which allows the user to set the print flags for the individual ADC channels. Specific choices are:

1) turn on all print flags,
2) turn off all print flags,
3) turn on a range of print flags,
4) turn off a range of print flags,
5) display current print flags,
6) save current print flags.

If the command is executed without the -s option, the last set of print flags is used to selectively display the ADC channels.

USES:

getadc.flags

DIRECTORY:

/bin

SOURCE:

/usr/csc/ele/getadc.c

Analog I/O system command.

NAME:

setdac Sets the digital-to-analog converters.

COMMAND:

setdac [-s] [-0]

DESCRIPTION:

This command is used to set the voltage on a range of digital-to-analog output channels. The +/- 1.0 unit output range is scaled to +/- 10.0 volts so a single unit is worth .0049 volts. The channels to be set are selected by executing the command with the -s option. This option displays a request for the range of channels to be set, and then queries for individual channel values in terms of units. All DAC channels may be set to zero by executing the command with the "-0" option.

USES:

Nothing.

DIRECTORY:

/bin

SOURCE:

/usr/csc/ele/setdac.c

UNOS system command.

NAME:

lterm Terminal emulator.

COMMAND:

term [-s] [-S]

DESCRIPTION:

This command connects the user terminal to the Langley central data communication switch at 1200 baud. This is a dumb terminal emulator which provides rudimentary file transfer capabilities. No attempt is made to emulate control codes of any particular terminal for editing purposes.

The emulator commands are as follow:

@	Return to UNOS (operating system.)
!	To download a file.
^	To upload a file.
~	To excute a system command.
?	For help.

The options are:

-s	300 baud
-S	900 baud

Upload means to transfer a file from the Charles River computer to the remote computer. The remote computer must have some mechanism for receiving the text.

Download means to transfer a file from the remote computer to the Charles River computer. No attempt is made to check for existence of the receiving file name before saving the downloaded file.

USES:

/doc/cmds/lterm.help

DIRECTORY:

/bin

SOURCE:

/jpw/lterm.c

Analog I/O system command.

NAME:

8751_test Test the serial communication link to the 8751 boards.

COMMAND:

8751_test

DESCRIPTION:

This command facilitates verification and calibration of the 8751 micro-controller interface over the RS-232 serial ports. The command queries for voltages to be output by the digital-to-analog converters on the 8751 boards. The data input is in terms of units with 2047 equal to 9.9951 volts and -2048 equal to -10.0000 volts. If the input line contains only one value, all active boards are sent that value. Otherwise, individual values are sent.

USES:

motint()
motsub()

DIRECTORY:

/bin

SOURCE:

/usr/rdb/8751_test.for

Analog I/O system subroutine.

NAME:

getadc() (C callable) Sample a range of analog-to-digital converters.

CALL:

```
int error, first_adc, last_adc;
float adc_data_pointer;
int getadc( first_adc, last_adc, &adc_data_pointer)
error = getadc(first_adc, last_adc, &adc_data_pointer)
```

DESCRIPTION:

This subroutine samples a range of analog-to-digital converters. The +/- 10.0 volt input range is scaled to +/- 1.0 units so a single bit is worth .00049 units. The arguments are:

first_adc	(int)	First converter to be sampled (numbering starts from zero.)
last_adc	(int)	Last converter to be sampled (maximum is 63.)
&adc_data_pointer	(*)	Starting location for storing sample data. Data are floating point values with a range of +/- 1.0.

RETURNS:

error = 0 indicates valid transfer.
error = indicates bad range.

USES:

Nothing

LIBRARY: None

SOURCE:

/usr/csc/ele/getadc_c.c

Analog I/O system subroutine.

NAME:

setdac() (C callable) Set a range of digital-to-analog conveters.

CALL:

```
int error, first dac, last dac;
float dac data pointer;
int setdac( first dac, last dac, & dac data pointer )
```

DESCRIPTION:

This subroutine sets a range of digital-to-analog converters. The +/- 1.0 unit output range is scaled to +/- 10.0 volts so a single unit is worth .0049 volts. The arguments are:

first dac	(int)	First converter to be set (numbering starts from zero.)
last_dac	(int)	Last converter to be set (maximum is 7.)
&dac_data_pointer	(*)	Starting location DAC data. Data are floating point values with a range of +/- 1.0.

RETURNS:

```
error = 0  Indicates valid transfer.
error =-1  Indicates bad range.
error > 0  Indicates "error" number of data words out of range.
```

USES:

Nothing.

LIBRARY:

None.

SOURCE:

/usr/csc/ele/setdac_c.c

Analog I/O system subroutine.

NAME:

getadc() (Fortran callable) Sample a range of analog-to-digital converters.

CALL:

```
integer error, getadc
error =  getadc(first_adc, last_adc, adc_data_array)
```

DESCRIPTION:

This subroutine samples a range of analog-to-digital converters. The +/- 10.0 volt input range is scaled to +/- 1.0 units so a single bit is worth .00049 units. The arguments are:

first_adc	(integer)	First converter to be sampled (numbering starts from zero.)
last_adc	(integer)	Last converter to be sampled (maximum is 63.)
adc_data_array	(real)	Starting location for storing sample data. Data are floating point values with a range of +/- 1.0.

RETURNS:

error = 0 indicates valid transfer.
error =-1 indicates bad range.

USES:

getadc_w.j.

LIBRARY:

/lib/acrl_rt_lib_f.j

SOURCE:

/usr/csc/ele/getadc_f.c

Analog I/O system subroutine.

setdac() (Fortran callable) Set a range of digital-to-analog converters.

CALL:

```
integer error, setdac
error = setdac( first_dac, last_dac, dac_data_arrray )
```

DESCRIPTION:

This subroutine sets a range of digital-to-analog converters. The +/- 1.0 unit output range is scaled to +/- 10.0 volts so a single unit is worth .0049 volts. The arguments are:

first_dac	(integer)	First converter to be set (numbering starts from zero.)
last_dac	(integer)	Last converter to be set (maximum is 7.)
dac_data_array	(real)	Starting location DAC data. Data are floating point values with a range of +/- 1.0.

RETURNS:

error = 0	Indicates valid transfer.
error =-1	Indicates bad range.
error > 0	Indicates "error" number of data words out of range.

USES:

setdac_w.j

LIBRARY:

/lib/acrl_rt_lib_f.j

SOURCE:

/usr/csc/ele/setdac_f.c

Analog I/O system subroutine.

NAME:

motsub() (Fortran callable) Send scaled voltages to the 8751 micro-controllers which in turn set individual DACs.

CALL:

```
integer torque
call motsub ( torque )
```

DESCRIPTION:

This subroutine sends the motor torque command data to the 8751 micro-controller boards which in turn load the data into the digital-to-analog converters.

torque(6) (integer) Array of dimension 6 which contains the scaled data to be output on the 8751 DAC's. The range of the data is +2047 for +9.9951 volt output to -2048 for -10.0000 volt output.

REQUIRES:

Call to motint.

RETURNS:

Nothing.

USES:

motsub_w.j

LIBRARY:

/lib/acrl_rt_lib_f.j

SOURCE:

/jpw/8751COM/torsub.c

Analog I/O system subroutine.

NAME:

finish() (Fortran callable) Close serial ports to 8751s.

CALL:

finish()

DESCRIPTION:

This subroutine closes the serial communication lines to the 8751s. There are no arguments.

RETURNS:

Nothing.

USES:

finish_w.j.

LIBRARY:

/lib/acrl_rt_lib_f.j

SOURCE:

/jpw/8751COM/torsub.c

Analog I/O system subroutines.

NAME:

thrust (Fortran callable) Set the discrete ports to activate the thrusters.

CALL:

```
integer*2 thrust = thrust ( x,y )
```

DESCRIPTION:

This subroutine sets the states of the four discrete outputs on the Parallel Interface/Timer. It was designed for the thrusters on the SCOLE facility. The arguments can have one of three values: 1, 0, or -1 corresponding to positive, none, and negative thrust respectively.
The arguments are:

x (integer) State of x thruster.

y (integer) State of y thruster.

REQUIRES:
Nothing.

RETURNS:
Nothing.

USES:
thrust_w.j

LIBRARY:
/lib/acrl_rt_lib_f.

SOURCE:
/jpw/TIMER/pitdsc.j

Analog I/O system subroutine.

NAME:

rtime() (fortran callable) Mark the start of the real-time loop and change the sample time interval if required.

CALL:

```
call rtime(tau,flag,k)
```

DESCRIPTION:

An internal memory mapped timer is used to control the timing of real-time operations in the Charles River Computer. The programmable clock is required to generate a start pulse and a stop pulse for each sampling interval of the control process. The maximum interval is eighty-five seconds and the minimum interval is 5 micro-seconds. The timer is a Motorola M68230 Parallel Interfac Timer (PI/T) which provides versitile double buffered parallel interfaces and 24-bit programmable timer for M68000 systems.

Note: The call to rtime() should be made just inside the real time loop. The arguments are:

tau (real) is the sample period,
flag (logical) is the indicator to either maintain the same value of tau or pass in a new value,
k (integer) is the timout parameter. If k is returned from rtime containing a 1, this indicates a normal return. If k is returned a 0, a timout has occurred, and appropriate action should be taken. The user must supply his/her own timout procedure.

REQUIRES:

Nothing.

RETURNS:

k

LIBRARY:

/lib/acrl_rt_lib_f.j
Add " /jpw/rtime.obj " to FORTRAN compile command.

USES:

tmset()
Init()
rtwate()
cktim()
pint_w.j

SOURCE:

/jpw/TIMER/pint.c
/jpw/TIMER/pint_w.m
/jpw/rtime.for

REFERENCES

1. Taylor, L. W., Jr., and Balakrishnan, A. V.: A Laboratory Experiment Used To Evaluate Control Laws for Flexible Spacecraft...NASA/IEEE Design Challenge, June 1983, pp.1-2.

TABLE I. SENSOR PARAMETERS FOR CONTROL SYSTEM DESIGN

Component No.	Type	Sensed Variable	Signal Source	X Coord. (inches)	Y Coord. (inches)	Z Coord. (inches)	Sensitivity	Error	Range	Bias
1	Rate Gyro 1	s roll s pitch s yaw	ADC 0 ADC 1 ADC 2	27.8 " "	0.0 " "	-5.3 " "	0.0393 units/s^{-1} 0.2351 units/s^{-1} 0.2354 units/s^{-1}	.58 deg/s rms 0.55 deg/s rms 0.53 deg/s rms	+/-360 (deg/s) +/-60 (deg/s) +/-60 (deg/s)	2.458 v 2.361 v 2.564 v
2	Rate Gyro 2	r roll r pitch r yaw	ADC 3 ADC 4 ADC 5	0.0 " "	0.0 " "	-129.3 " "	0.0395 units/s^{-1} 0.2300 units/s^{-1} 0.2300 units s^{-1}	0.43 deg/s rms 0.10 deg/s rms 0.12 deg/s rms	+/-360 (deg/s) +/-60 (deg/s) +/-60 (deg/s)	2.457 v 2.703 v 2.779 v
3	Accelerom.	sx	ADC 6	9.0	0.0	-4.5	0.1 units/g	0.17 %F.S.	+/-20 g	0.00108 v
4	Accelerom.	sy	ADC 7	0.0	-11.5	-4.5	0.1 units/g	0.17 %F.S.	+/-20 g	0.00066 v
5	Accelerom.	sz	ADC 8	-6.3	0.0	-4.5	0.1 units/g	0.17 %F.S.	+/-20 g	0.00145 v
6	Accelerom.	m1x	ADC 9	-1.3	0.0	-41.0	0.1 units g	0.17 %F.S.	+/-20 g	0.00065 v
7	Accelerom.	m1y	ADC 10	0.0	-1.3	-41.0	0.1 units g	0.17 %F.S.	+/-20 g	0.00050 v
8	Accelerom.	m2x	ADC 11	-1.3	0.0	-87.8	0.1 units g	0.17 %F.S.	+/-20 g	0.00001 v
9	Accelerom.	m2y	ADC 12	0.0	-1.3	-87.8	0.1 units g	0.17 %F.S.	+/-20 g	0.00077 v
10	Accelerom.	rx	ADC 13	10.00	20.8	-129.3	0.1 units g	0.17 %F.S.	+/-20 g	0.00103 v
11	Accelerom.	ry	ADC 14	12.00	18.8	-129.3	0.1 units g	0.17 %F.S.	+/-20 g	0.00049 v

1 unit = 10 volt

s = shuttle
m1 = upper mast location
m2 = lower mast location
r = reflector
F.S. = full scale
ADC = Analog-to-digital converter channel

TABLE II. TYPICAL THRUSTER DATA FOR SIXTY PSI NOZZLE PRESSURE.

Peak Thrust	0.641 lb
Steady State Thrust	0.32 lb
Rise Time	0.032 seconds
Thrust Duration	24 seconds

TABLE III. ACTUATOR PARAMETERS FOR CONTROL SYSTEM DESIGN

Component No.	Type	Actuation Direction	Signal Source	X Coord. (inches)	Y Coord. (inches)	Z Coord. (inches)	Sensitivity	Noise (mv rms)	Range	Bias
12	CMG (for.)	s roll	tty5	35.3	0.0	4.8	0.3448 (units/ft-lb)	-	+/- 1.5 ft-lb	0
		s pitch	tty5	"	"	"	0.3448 (units/ft-lb)	-	+/- 1.5 ft-lb	0
13	CMG (aft)	s pitch	tty5	-32.9	0.0	8.8	0.3448 (units/ft-lb)	-	+/- 1.5 ft-lb	0
		s yaw	tty5	"	"	"	0.3448 (units/ft-lb)	-	+/- 1.5 ft-lb	0
14	Thrusters	rX	DISC 1	12.5	21.0	-124.3	.35 lbs on/off	0.0252 lb rms	0.35 lb avg	0
		-rX	DISC 2	"	"	"	.35 lbs on/off	0.0252 lb rms	0.35 lb avg	0
		rY	DISC 3	"	"	"	.35 lbs on/off	0.0252 lb rms	0.35 lb avg	0
		-rY	DISC 4	"	"	"	.35 lbs on/off	0.0252 lb rms	0.35 lb avg	0
15	Reaction Wheel	rx	DAC 0	0.0	-6.0	-125.8	not available	-	not available	0
		ry	DAC 1	-6.0	0.0	-125.8	not available	-	not available	0
		rz	DAC 2	-4.5	-4.5	-125.8	not available	-	not available	0
16	Reaction Wheel	m1X	DAC 3	4.0	0.0	-41.0	0.05 units/oz-in	-	+20 oz-in	0
17	Reaction Wheel	m1Y	DAC 4	0.0	4.0	-41.0	0.05 units/oz-in	-	+20 oz-in	0
18	Reaction Wheel	m2X	DAC 5	4.0	0.0	-87.8	0.05 units/oz-in	-	+20 oz-in	0
19	Reaction Wheel	m2Y	DAC 6	0.0	4.0	-87.8	0.05 units/oz-in	-	+20 oz-in	0

1 unit = 10 volt
s = shuttle
m1 = upper mast location
m2 = lower mast location
r = reflector
F.S. = full scale
DAC = Digitial-to-analog converter channel
DISC = Discrete output channel

TABLE IV. SCOLE APPARATUS MASS PROPERTIES AND COMPONENT LOCATIONS

Component No.	Component Type	X Coord. (in.)	Y Coord. (in.)	Z Coord. (in.)	Weight (lb.)	Moment X	Moment Y	Moment YZ	I XX	I YY	I ZZ	I xy	I xz	I yz
						(lb. inches)			(slugs * in. 2)			(slugs * inches 2)		
	Sensor													
1	Rate gyro s	27.8	0.0	-5.3	1.69	46.9	0.0	-8.9	1.4	41.9	40.4	0.0	7.7	0.0
2	Rate gyro r	0.0	0.0	-129.3	1.69	0.0	0.0	-218.5	877.5	877.5	0.0	0.0	0.0	0.0
3	Accelerometer sx	9.0	0.0	-4.5	0.17	1.5	0.0	-0.8	0.1	0.5	0.4	0.0	0.2	0.0
4	Accelerometer sy	0.0	-11.5	-4.5	0.17	0.0	-0.2	-0.8	0.8	0.1	0.7	0.0	0.0	-0.3
5	Accelerometer sz	-6.3	0.0	-4.5	0.17	-1.1	0.0	-0.8	0.1	0.3	0.2	0.0	-0.2	0.0
6	Accelerometer m1x	-1.3	0.0	-41.0	0.17	-0.2	0.0	-7.0	8.9	8.9	0.0	0.0	-0.3	0.0
7	Accelerometer m1y	0.0	-1.3	-41.0	0.17	0.0	-0.2	-7.0	8.9	8.9	0.0	0.0	0.0	-0.3
8	Accelerometer m2x	-1.3	0.0	-87.8	0.17	-0.2	0.0	-14.9	40.7	40.7	0.0	0.0	-0.6	0.0
9	Accelerometer m2y	0.0	-1.3	-87.8	0.17	0.0	-0.2	-14.9	40.7	40.7	0.0	0.0	0.0	-0.6
10	Accelerometer rx	10.0	20.8	-129.3	0.17	1.7	3.5	-22.0	90.5	88.8	2.8	-1.1	6.9	14.3
11	Accelerometer ry	12.0	18.8	-129.3	0.17	2.0	3.2	-22.0	90.1	89.0	2.6	-1.2	8.2	12.9

TABLE IV. SCOLE APPARATUS MASS PROPERTIES AND COMPONENT LOCATIONS (CONT'D)

Component No.	Component Type	X Coord. (in.)	Y Coord. (in.)	Z Coord. (in.)	Weight (lb.)	Moment X	Moment Y (lb. inches)	Moment Z	I XX	I YY (slugs *	I ZZ in. 2)	I xy	I xz (slugs * inches 2)	I yz
	Actuators													
12	CMG (for.)	35.3	0.0	4.8	34.27	1208.0	0.0	162.8	24.0	1346.5	1322.4	0.0	-179.3	0.0
13	CMG (aft)	-32.9	0.0	8.8	34.27	-1127.5	0.0	299.9	81.5	1233.5	1152.0	0.0	308.3	0.0
14	Thrusters	12.0	20.8	-124.3	1.00	12.0	20.8	-124.3	493.3	484.3	17.9	-7.8	46.6	80.8
15	Reaction Wheel X	0.0	-6.0	-125.8	4.28	0.0	-25.7	-538.4	2108.3	2103.5	4.8	0.0	0.0	-101.0
16	Reaction Wheel Y	6.0	0.0	-125.8	4.28	-25.7	0.0	-538.4	2103.5	2108.3	4.8	0.0	-101.0	0.0
17	Reaction Wheel Z	-4.5	-4.5	-125.8	4.28	-19.3	-19.3	-538.4	2106.2	2106.2	5.4	-2.7	-75.7	-75.7
18	Reaction Wheel 1X	4.0	0.0	-41.0	1.45	5.8	0.0	-59.4	75.7	76.4	0.7	0.0	7.4	0.0
19	Reaction Wheel 2X	4.0	0.0	-87.8	1.45	5.8	0.0	-127.3	347.1	347.9	0.7	0.0	15.9	0.0
20	Reaction Wheel 1Y	0.0	4.0	-41.0	1.45	0.0	5.8	-59.4	76.4	75.7	0.7	0.0	0.0	7.4
21	Reaction Wheel 2Y	0.0	4.0	-87.8	1.45	0.0	5.8	-127.3	347.9	347.1	0.7	0.0	0.0	15.9
22	Solenoid	4.0	7.0	-123.8	5.50	22.0	38.5	-680.9	2626.2	2620.6	11.1	-4.8	85.1	148.9
	Power Supplies													
23	lambda 1	-9.5	-10.3	0.0	26.00	-247.0	-267.8	0.0	85.7	72.9	158.5	-79.5	0.0	0.0
24	lambda 2	-9.5	10.3	0.0	26.00	-247.0	-267.8	0.0	85.7	72.9	158.5	79.5	0.0	0.0
25	lambda 3	-9.5	-10.3	5.5	26.00	-247.0	-267.8	143.0	110.1	97.3	158.5	-79.5	42.5	46.0
26	lambda 4	-9.5	10.3	5.5	26.00	-247.0	267.8	143.0	110.1	97.3	158.5	79.5	42.5	-46.0
27	28v	-12.3	-10.5	5.0	7.25	-89.2	-76.1	36.3	30.5	39.7	58.9	-29.3	13.9	11.9
28	variable	-11.5	10.8	5.8	30.00	-345.0	322.5	172.5	138.5	154.0	230.9	115.9	62.0	-57.9

TABLE IV. SCOLE APPARATUS MASS PROPERTIES AND COMPONENT LOCATIONS (CONT'D)

Component No.	Component Type	X Coord. (in.)	Y Coord. (in.)	Z Coord. (in.)	Weight (lb.)	Moment X	Moment Y	Moment YZ	I XX	I YY	I ZZ	I xy	I xz	I yz
						(lb. inches)			(slugs * in. 2)			(slugs * inches 2)		
29	5 +/-15v	-8.5	-5.6	0.4	2.5	-19.5	-12.9	0.9	2.3	5.2	7.4	-3.4	0.2	0.2
30	6 +5v	-5.3	-10.0	10.8	4.00	-21.0	-40.0	43.2	26.9	17.9	15.8	-6.6	7.1	13.5
31	-5v	-2.3	-10.0	10.8	4.00	-9.2	-40.0	43.2	26.9	15.1	13.1	-2.9	1.5	13.5
	Counter Balance													
32	wing +	-10.0	18.5	10.8	79.75	-797.5	1475.4	1724.6	2005.9	1405.9	1095.3	461.1	538.9	997.0
33	wing -	-10.0	-18.5	11.8	79.75	797.5	-1475.4	1724.6	2005.9	1405.9	1095.3	-461.1	538.9	997.0
34	nose	46.0	0.0	0.0	10.00	460.0	0.0	0.0	0.0	657.1	657.1	0.0	0.0	0.0
35	air tank	15.5	0.0	10.0	35.25	546.4	0.0	352.5	109.5	372.5	263.0	0.0	-170.7	0.0
36	power mount	-10.3	0.0	-2.6	19.13	-196.1	0.0	-50.3	4.1	66.5	62.4	0.0	-16.1	0.0
37	regulator	4.0	-9.3	0.3	4.00	16.0	-37.0	1.0	10.6	2.0	12.6	4.6	-0.1	0.3
38	bracket (aft.)	-26.8	0.0	6.5	13.00	-348.4	0.0	84.5	17.1	307.0	290.0	0.0	70.8	0.0
39	power amplifier	-15.5	0.0	7.5	5.98	-92.7	0.0	44.9	10.4	55.1	44.6	0.0	21.7	0.0
40	manifold bottom	0.0	0.0	-124.5	1.68	0.0	0.0	-209.2	808.7	808.7	0.0	0.0	0.0	0.0
41	manifold top	0.0	0.0	-4.3	3.19	0.0	0.0	-13.6	1.8	1.8	0.0	0.0	0.0	0.0
42	misc. 1	-7.5	-22.5	-2.5	5.00	-37.5	-112.5	-12.5	79.6	9.7	87.3	-26.4	-2.9	-8.8
43	misc. 2	42.5	0.0	-2.5	0.30	12.8	0.0	-0.8	0.1	16.9	16.8	0.0	1.0	0.0
	Structure													
44	shuttle	4.9	0.0	-3.4	501.73	2458.5	0.0	-1693.3	1902.8	6967.7	8510.2	0.0	259.3	0.0

TABLE IV. SCOLE APPARATUS MASS PROPERTIES AND COMPONENT LOCATIONS (CONT'D)

Component No.	Component Type	X Coord. (in.)	Y Coord. (in.)	Z Coord. (in.)	Weight (lb.)	Moment X	Moment Y	Moment Z	I XX	I YY	I ZZ	I xy	I xz	I yz
						(lb. inches)			(slugs * in. 2)			(slugs * inches 2)		
45	mast	0.0	0.0	-64.8	4.48	0.0	0.0	-290.1	747.7	747.7	0.0	0.0	0.0	0.0
46	reflector	12.0	20.8	-125.8	4.76	57.1	99.0	-598.6	2428.2	2385.3	134.9	-384.3	-223.0	36.9
	Total				1019.0	-82.3	133.3	-790.9	22407.3	29957.2	15817.8	-349.9	1398.3	69.1

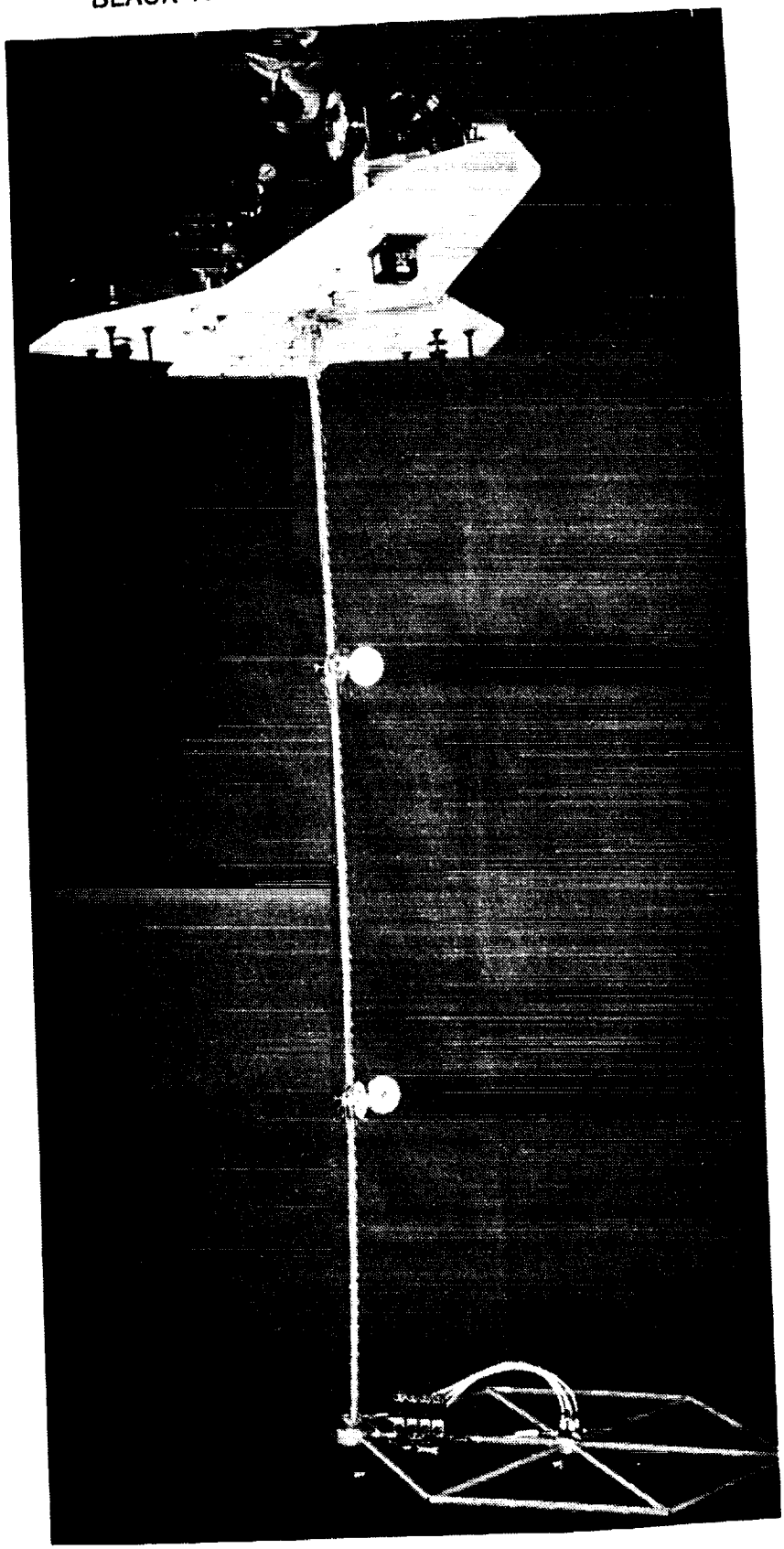


Figure 1. The SCOLE experiment apparatus.

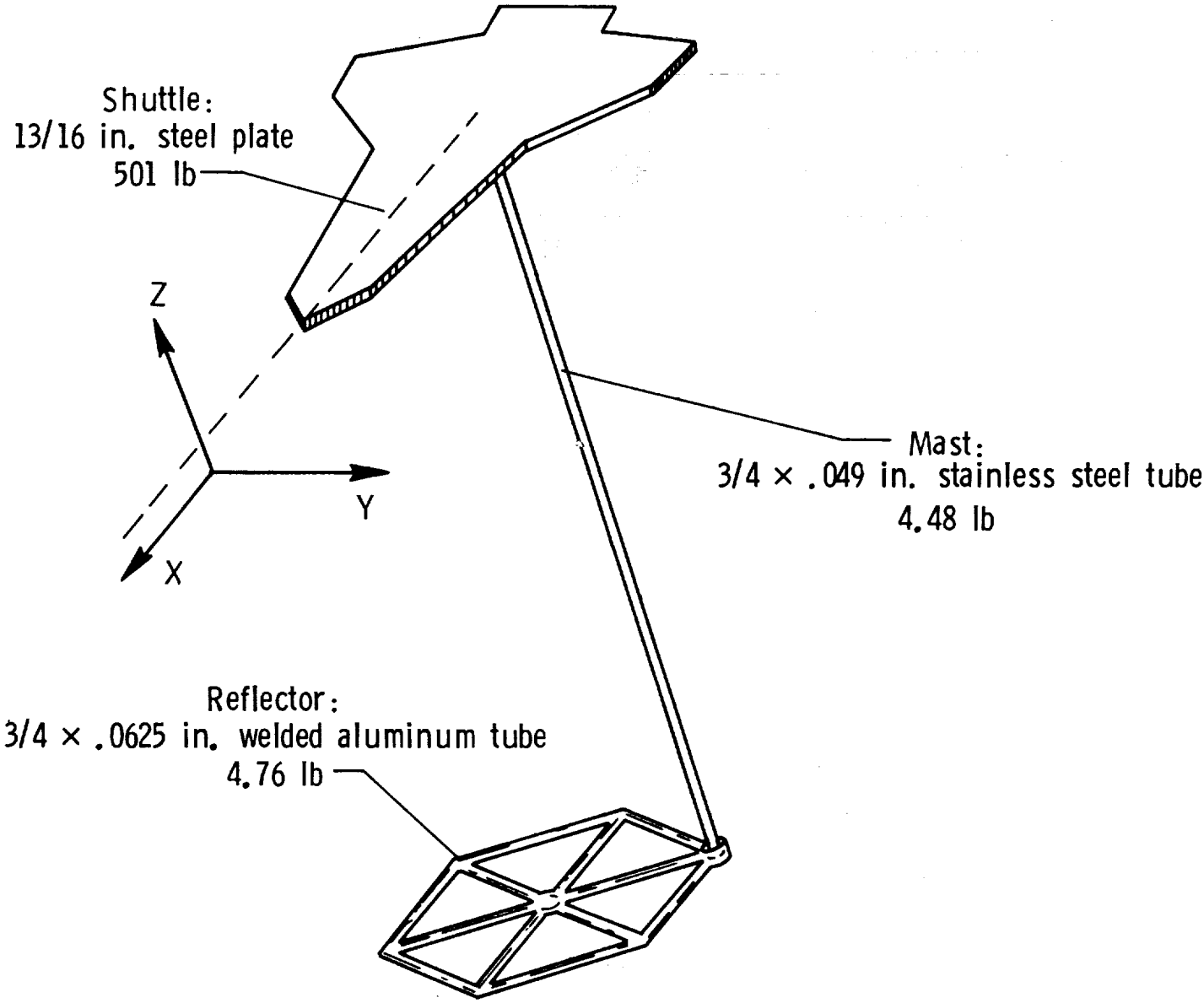


Figure 2. Basic SCOLE structural assembly.

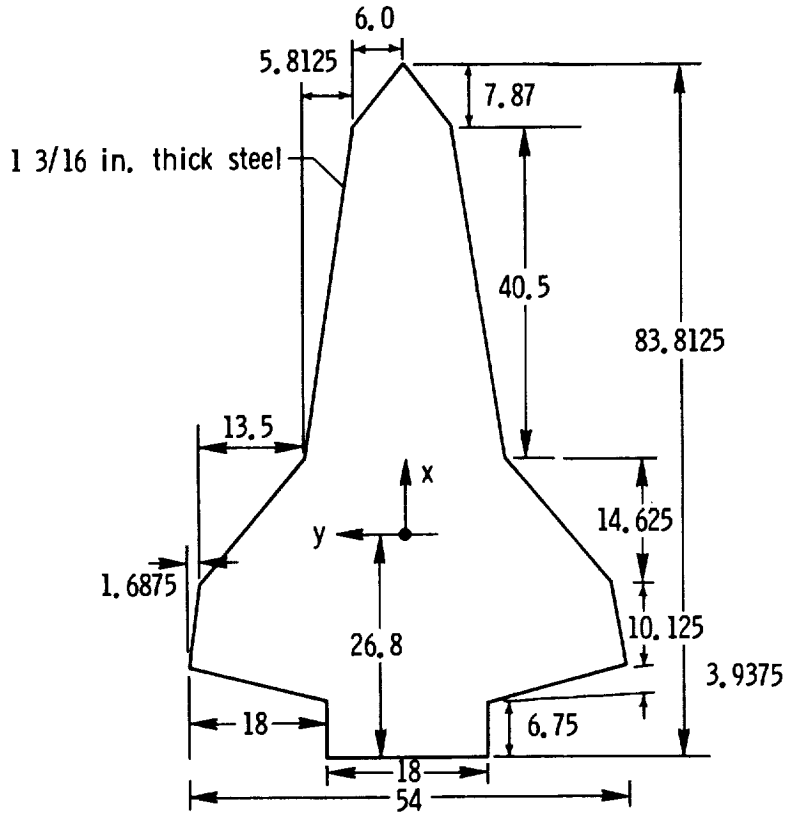


Figure 3. Shuttle planform.

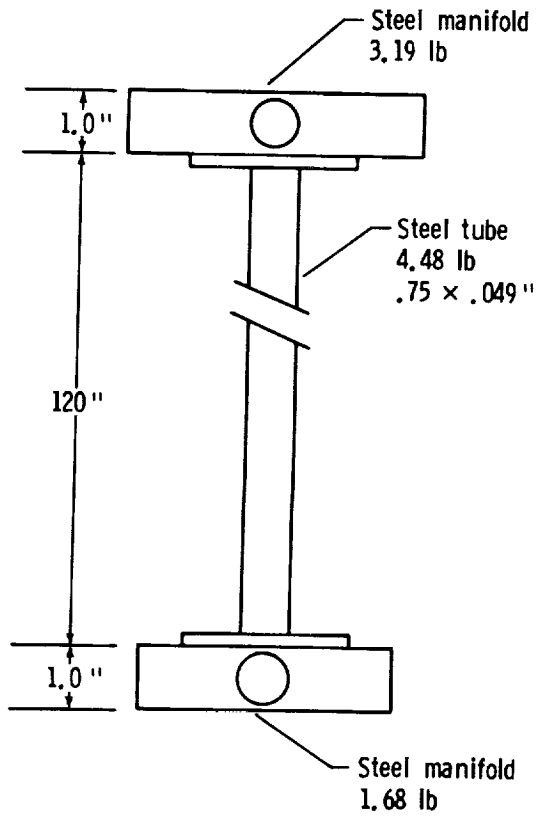


Figure 4. Mast and manifold assembly.

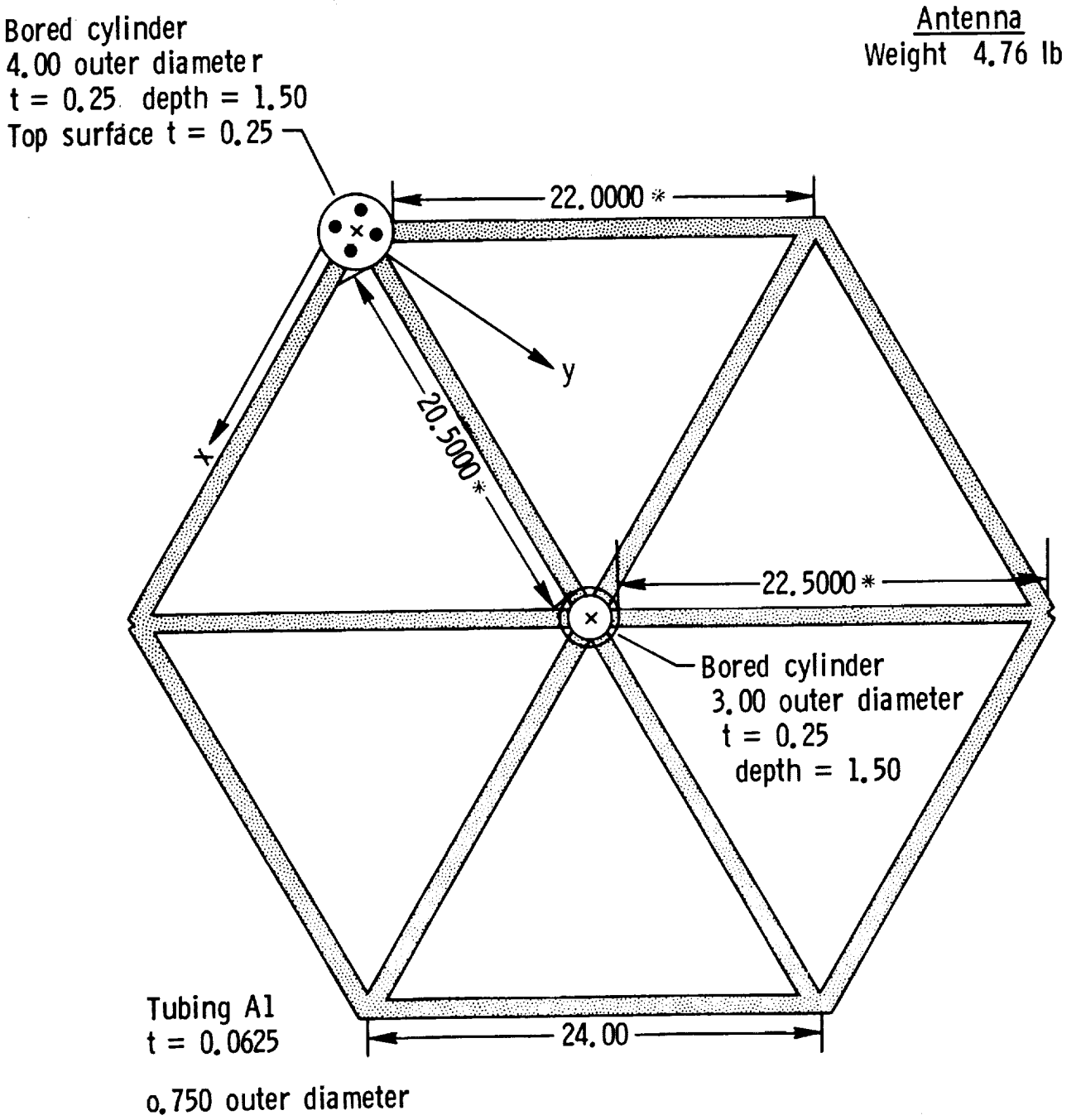


Figure 5. Reflector assembly.

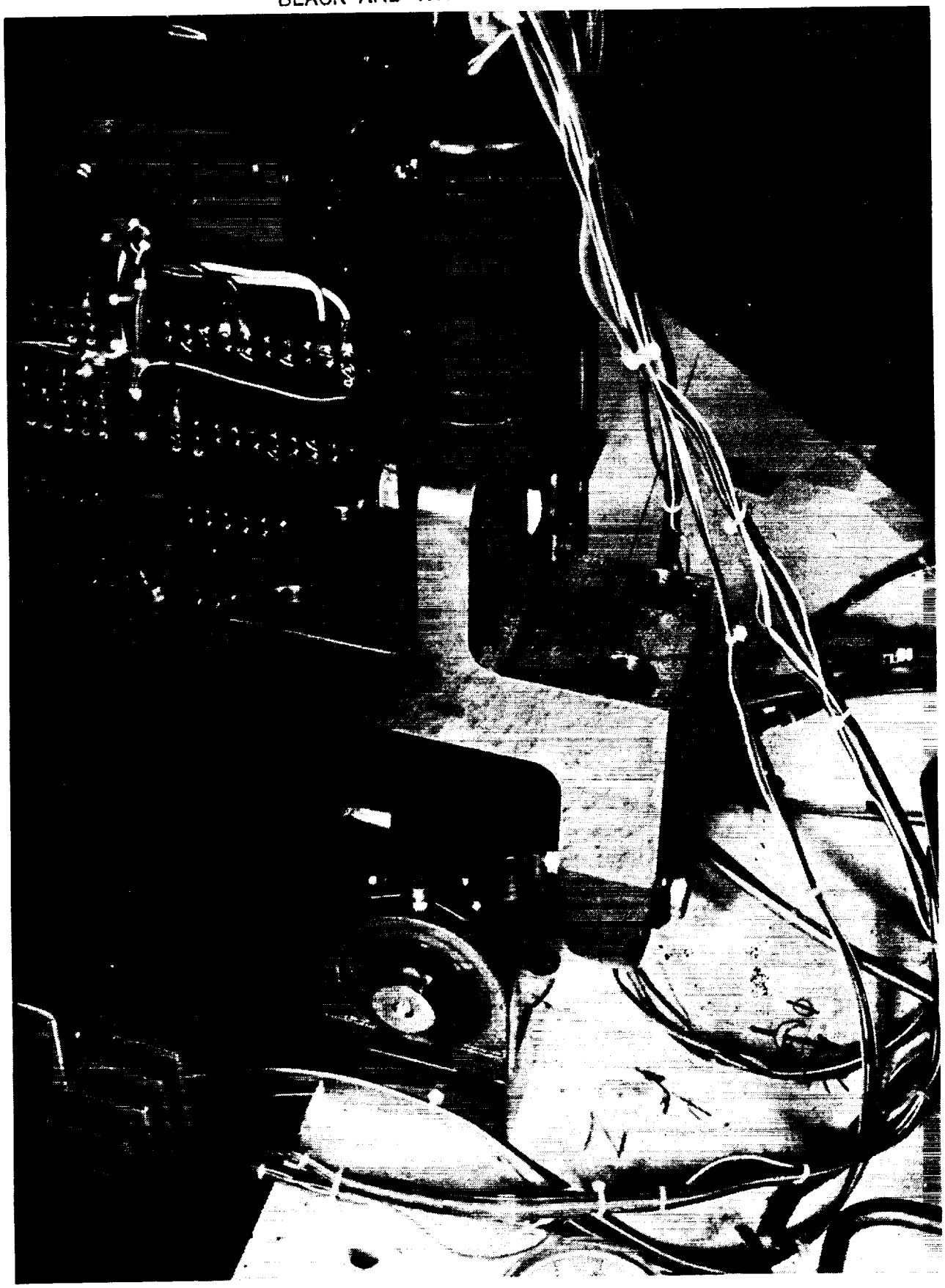
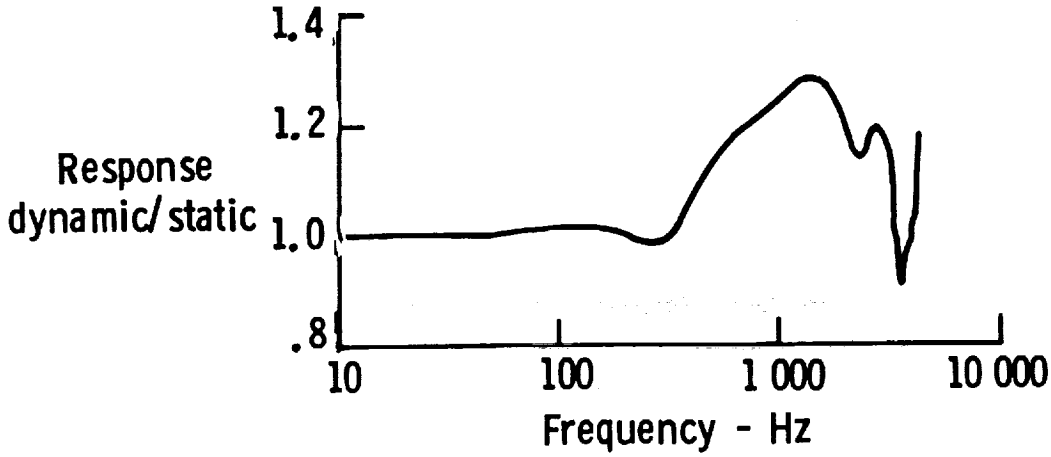


Figure 6. Universal joint suspension point.



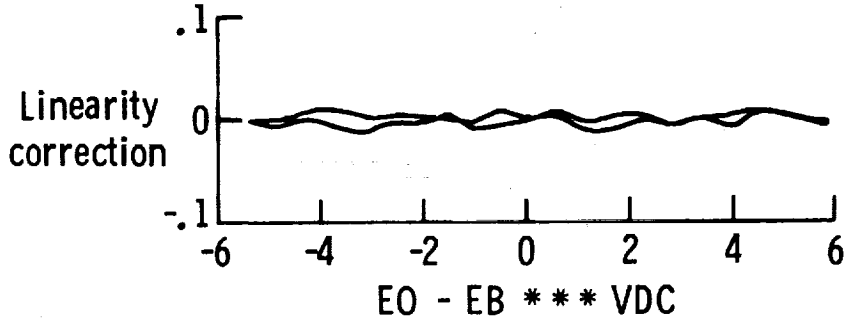


Figure 7. Typical accelerometer calibration.

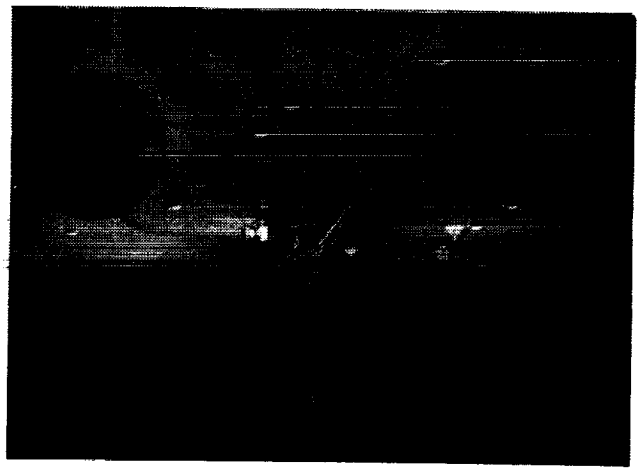


Figure 8a. Shuttle-mounted accelerometers.

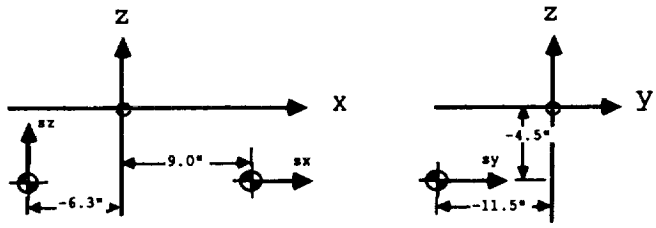


Figure 8b. Coordinates and sensing axis of shuttle-mounted accelerometers.

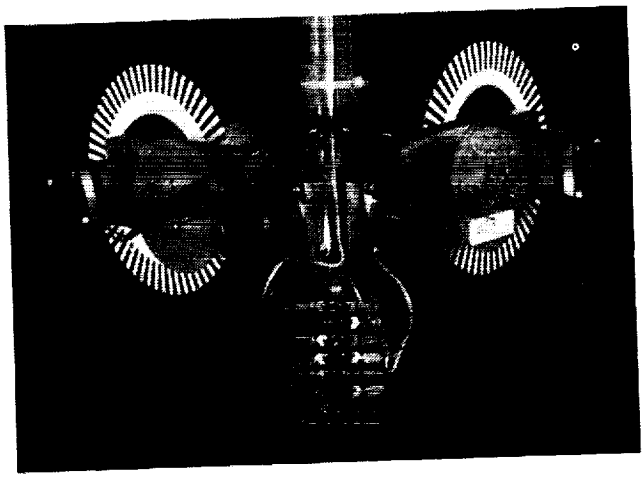


Figure 9a. Mast-mounted accelerometers.

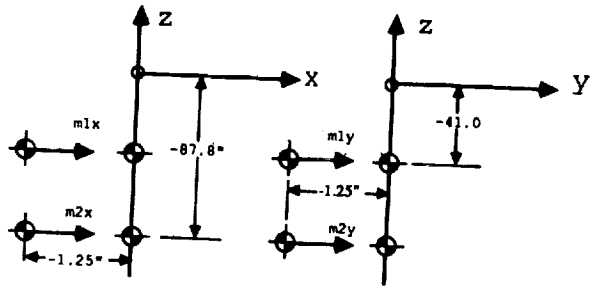


Figure 9b. Coordinates and sensing axis of mast-mounted accelerometers.

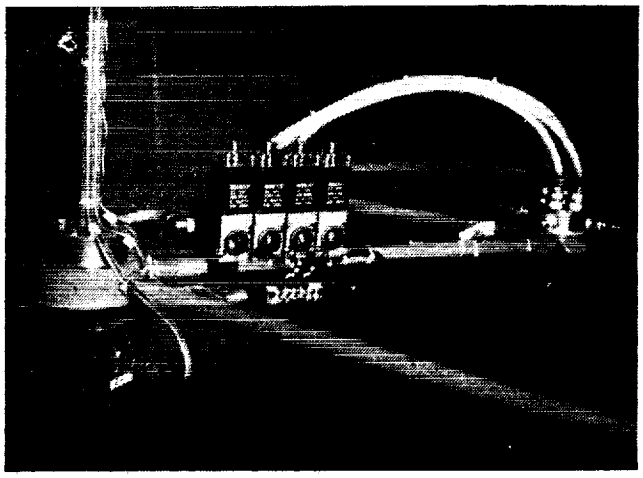


Figure 10a. Reflector-mounted accelerometers.

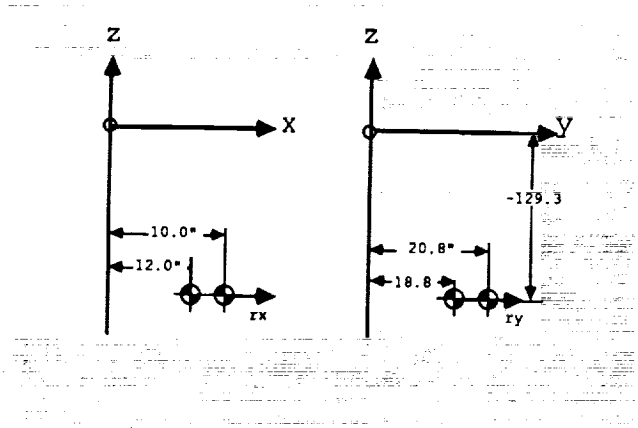
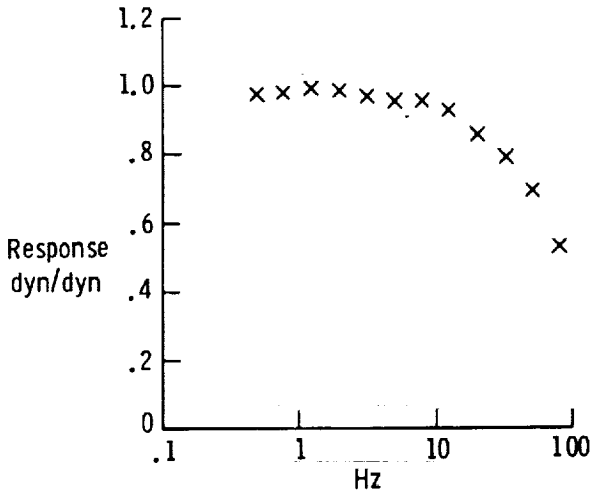


Figure 10b. Coordinates and sensing axis of reflector mounted accelerometers.



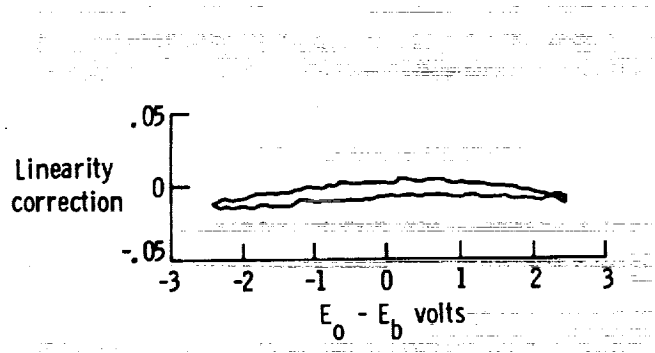


Figure 11. Typical rate sensor calibration.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

Figure 12a. Shuttle-mounted, three-axis rate sensor.

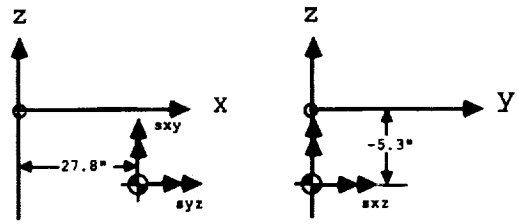


Figure 12b. Coordinates and sensing axis of shuttle-mounted rate sensor.

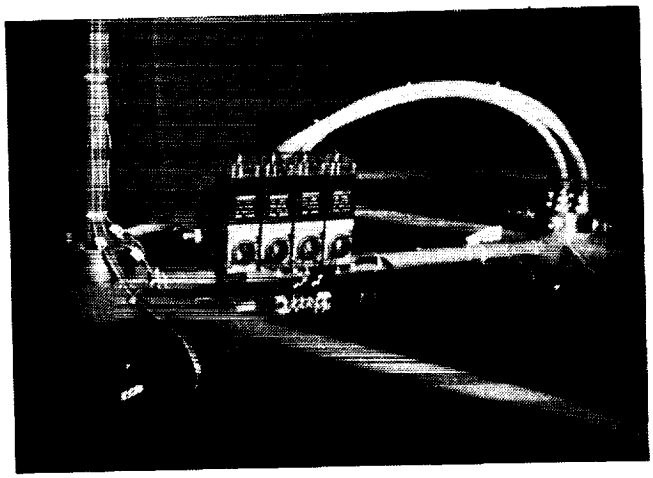


Figure 13a. Mast-end-mounted, three-axis rate sensor.

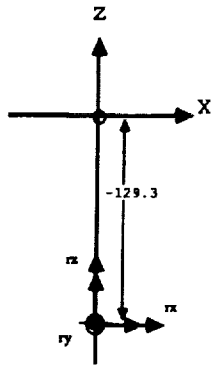


Figure 13b. Coordinates and sensing axis of mast-end-mounted rate sensor.

ORIGINAL PAGE IS
OF POOR QUALITY

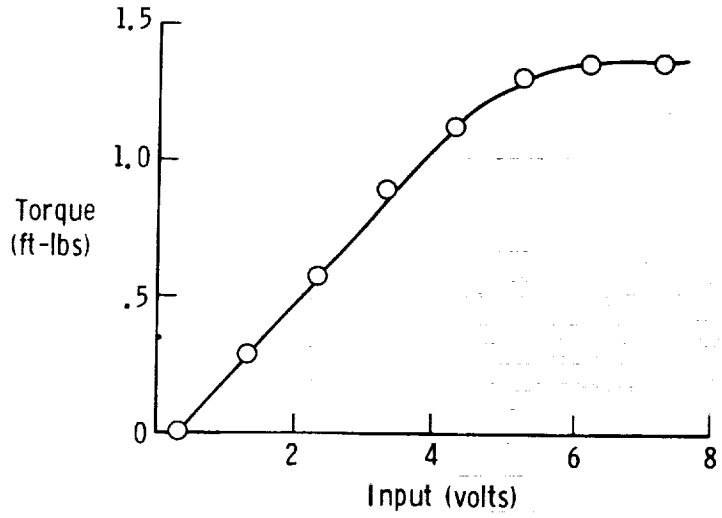


Figure 14. Typical CMG gimbal torque sensitivity curve.

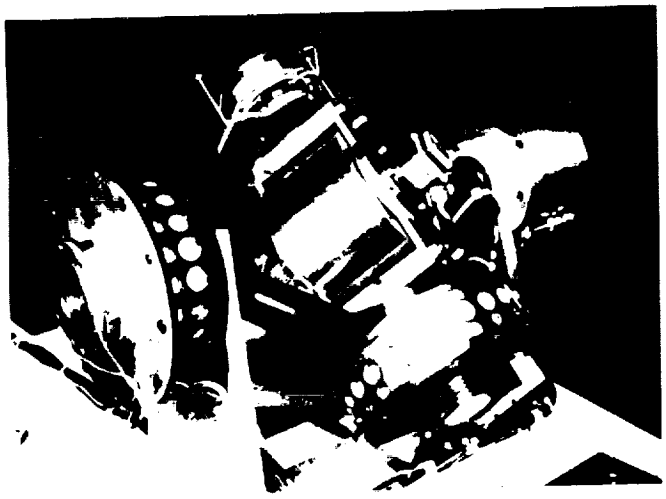


Figure 15a. Shuttle mounted forward CMG.

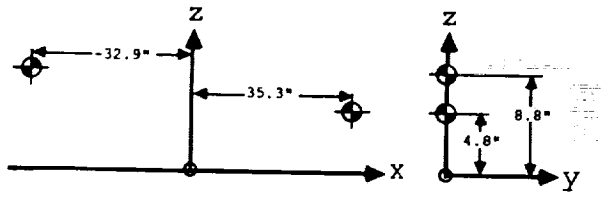


Figure 15b. Coordinates and actuation axis of shuttle-mounted CMG's.

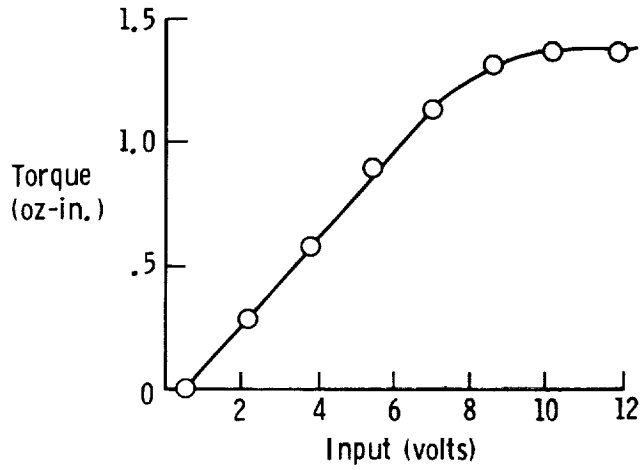


Figure 16. Typical reaction wheel torque sensitivity curve.

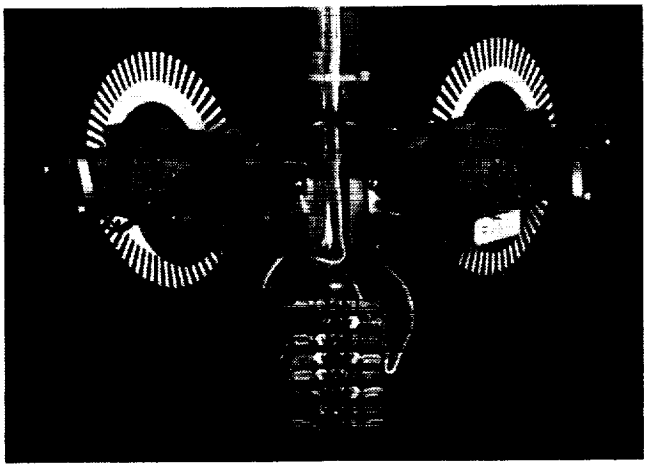


Figure 17a. Mast-mounted reaction wheels.

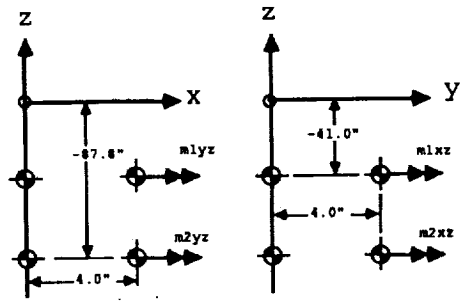


Figure 17b. Coordinates of mast-mounted reaction wheels.

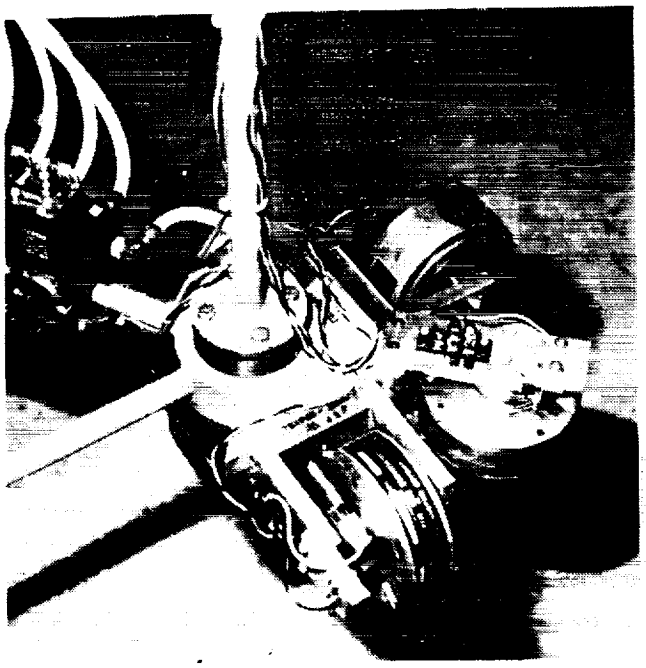


Figure 18a. Mast-end mounted reaction wheels.

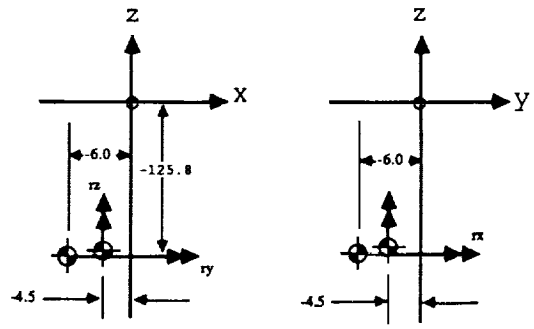


Figure 18b. Coordinate and axis of actuation for mast-end mounted reaction wheels.

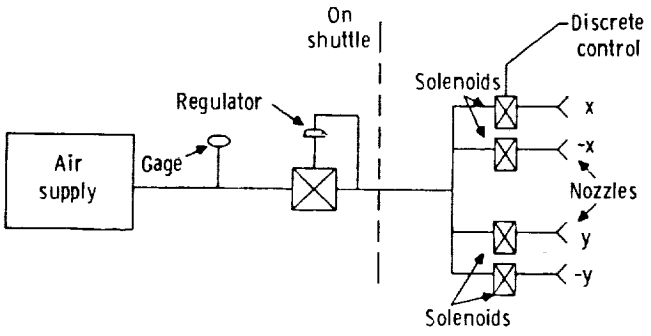


Figure 19. Thruster air supply schematic.

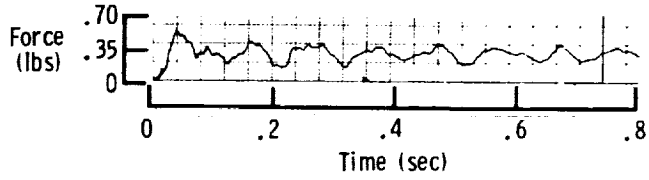


Figure 20. Thruster startup and transients.

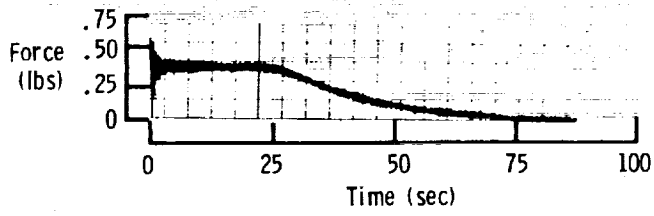


Figure 21. Thrust magnitude and duration.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

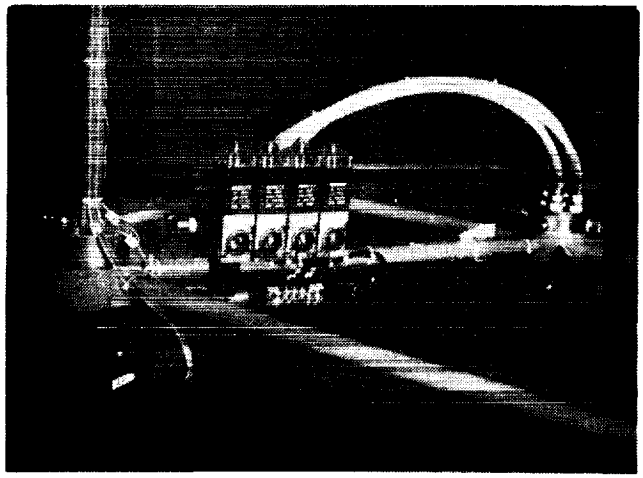


Figure 22a. Reflector-mounted thrusters.

ORIGINAL PAGE IS
OF POOR QUALITY

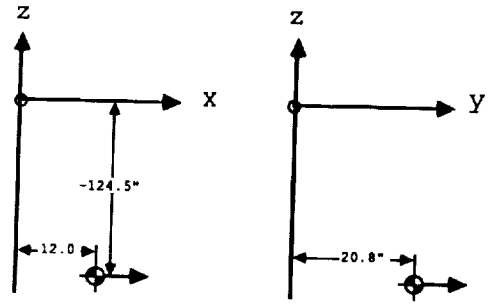


Figure 22b. Coordinates of reflector-mounted thrusters.

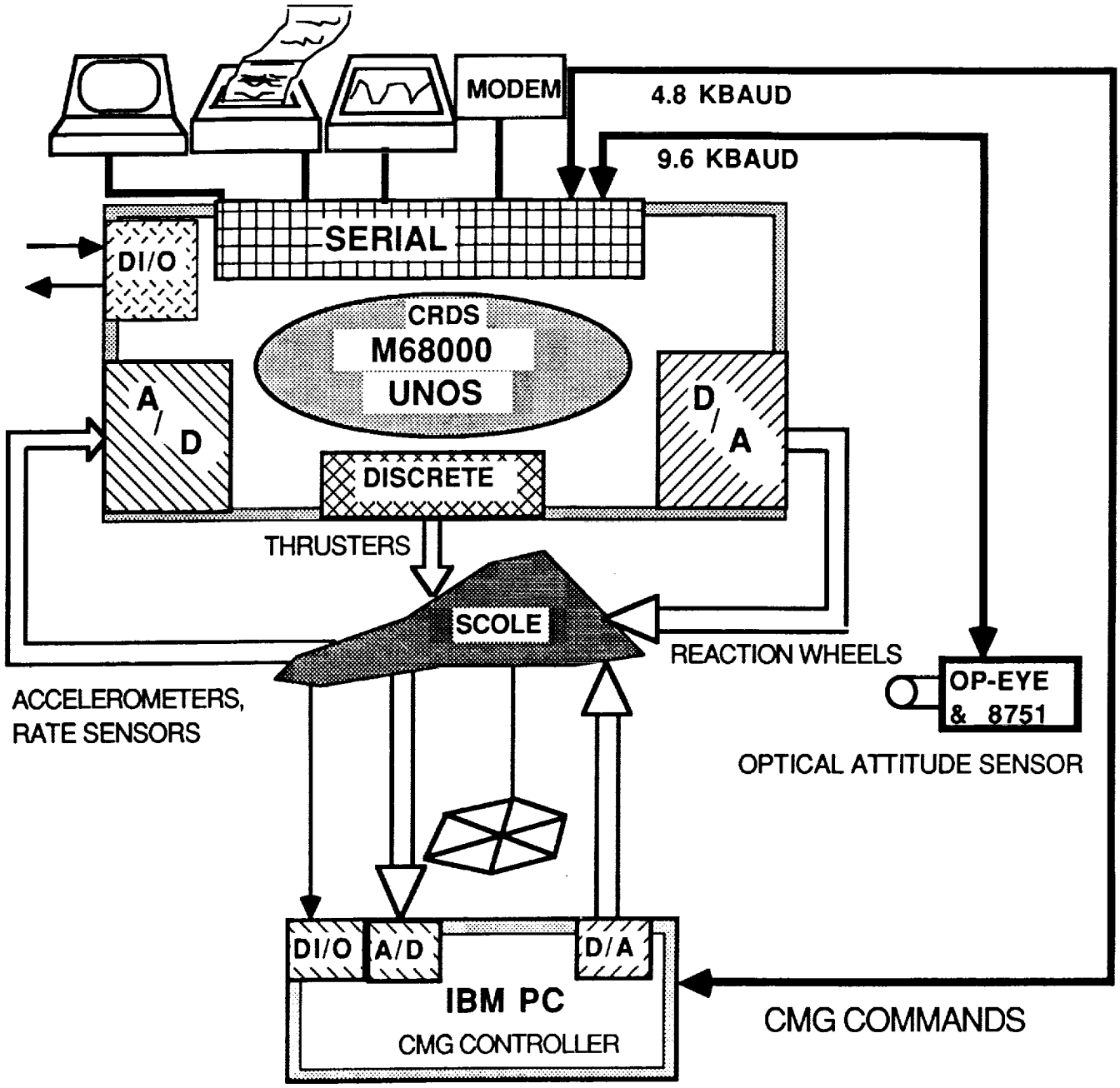


Figure 23. SCOLE computer interfaces.

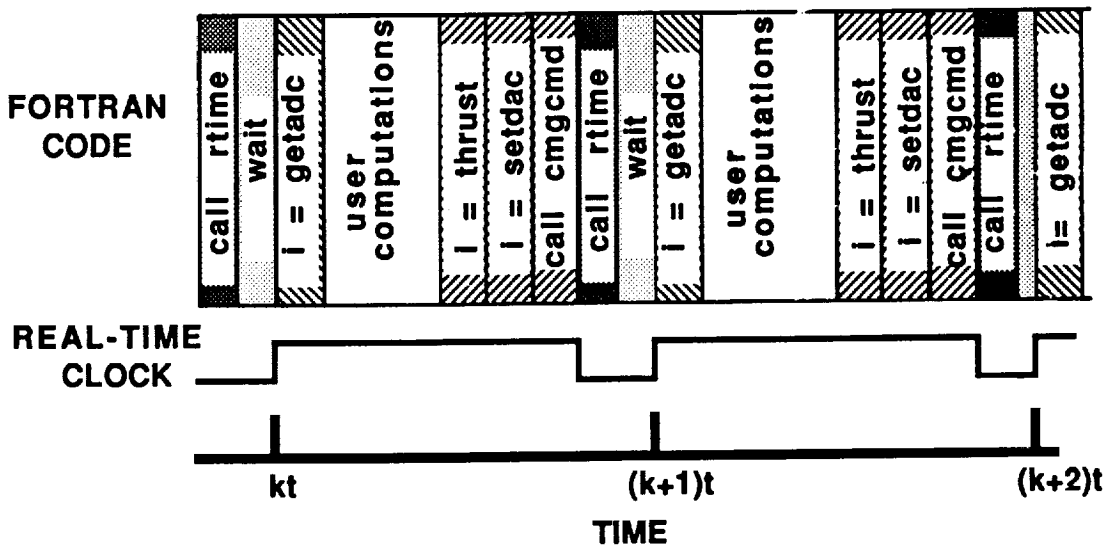


Figure 24. Real-time sample interval.

ANALYTIC REDUNDANCY MANAGEMENT FOR SCOLE

Raymond C. Montgomery

Spacecraft Control Branch
NASA Langley Research Center
Hampton, VA 23665

Presented at the
5th Annual SCOLE Workshop
Lake Arrowhead, CA

October 31, 1988

OUTLINE

SUMMARIZE THE GRID SCHEME OF WILLIAMS AND MONTGOMERY

DESCRIPTION OF THE LQG DESIGN FOR THE SCHEME

PLANS FOR COMPLETION OF THE WORK

OBJECTIVE OF WORK

DEVELOP & TEST A PRACTICAL SENSOR ARM SCHEME USING SCOLE

APPROACH

USE SCHEME PREVIOUSLY DEVELOPED FOR THE GRID BY WILLIAMS AND MONTGOMERY

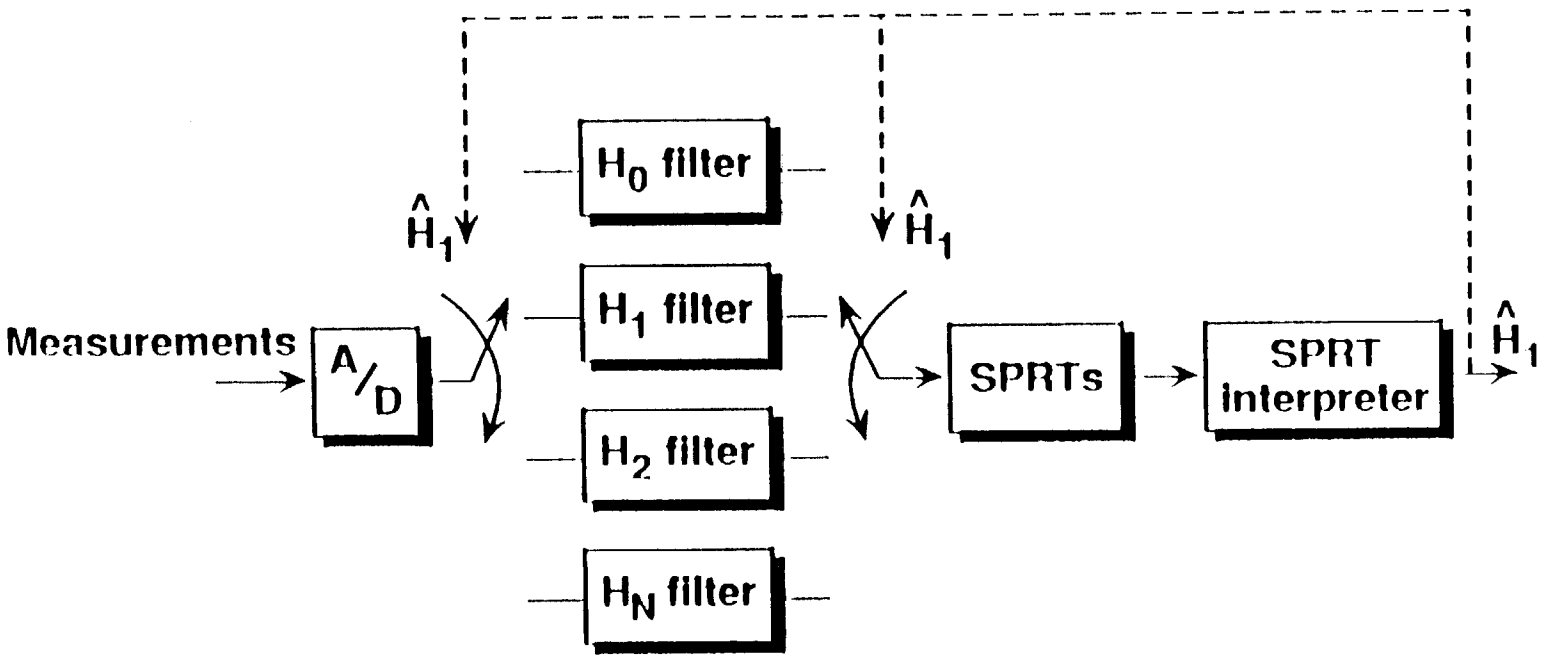
GRID ARM SCHEME – SUMMARY

USE SINGLE, ON–LINE, KALMAN FILTER

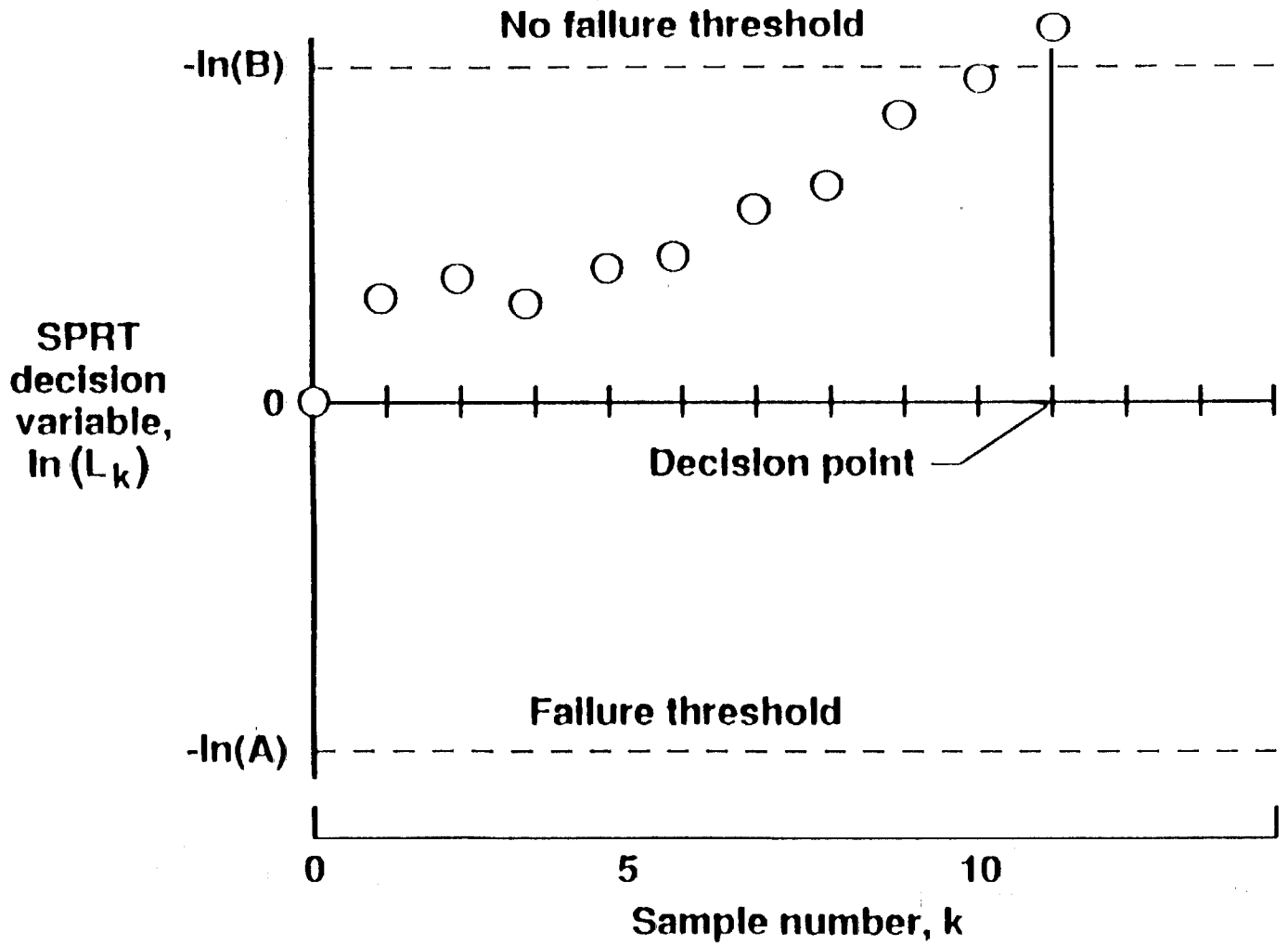
USE SPRT TO CHECK THE ZERO MEAN CHARACTER OF THE ESTIMATED MEASUREMENT ERROR

IF FAILURE IS DETECTED, ISOLATE USING FAILURE SIGNATURE IN THE ESTIMATED MEASUREMENT ERRORS

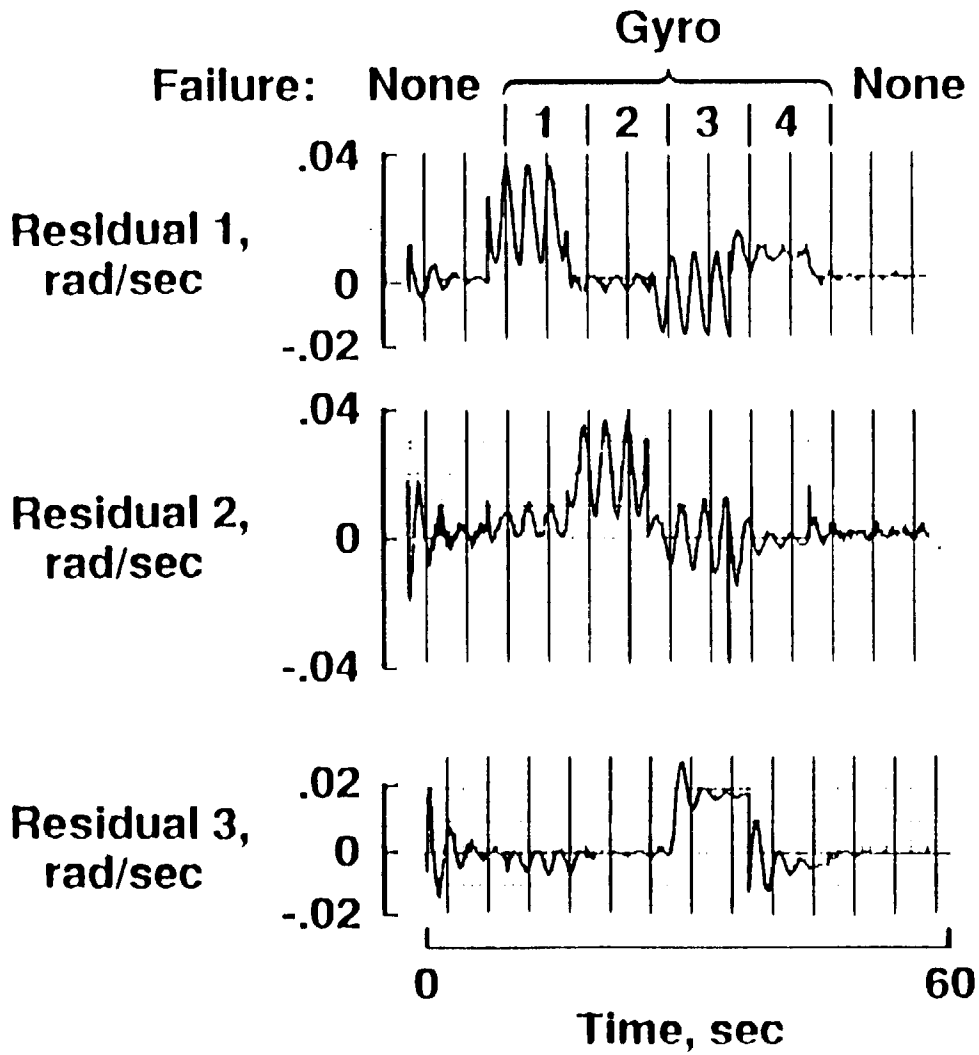
Logic for Failure Detection and Reconfiguration



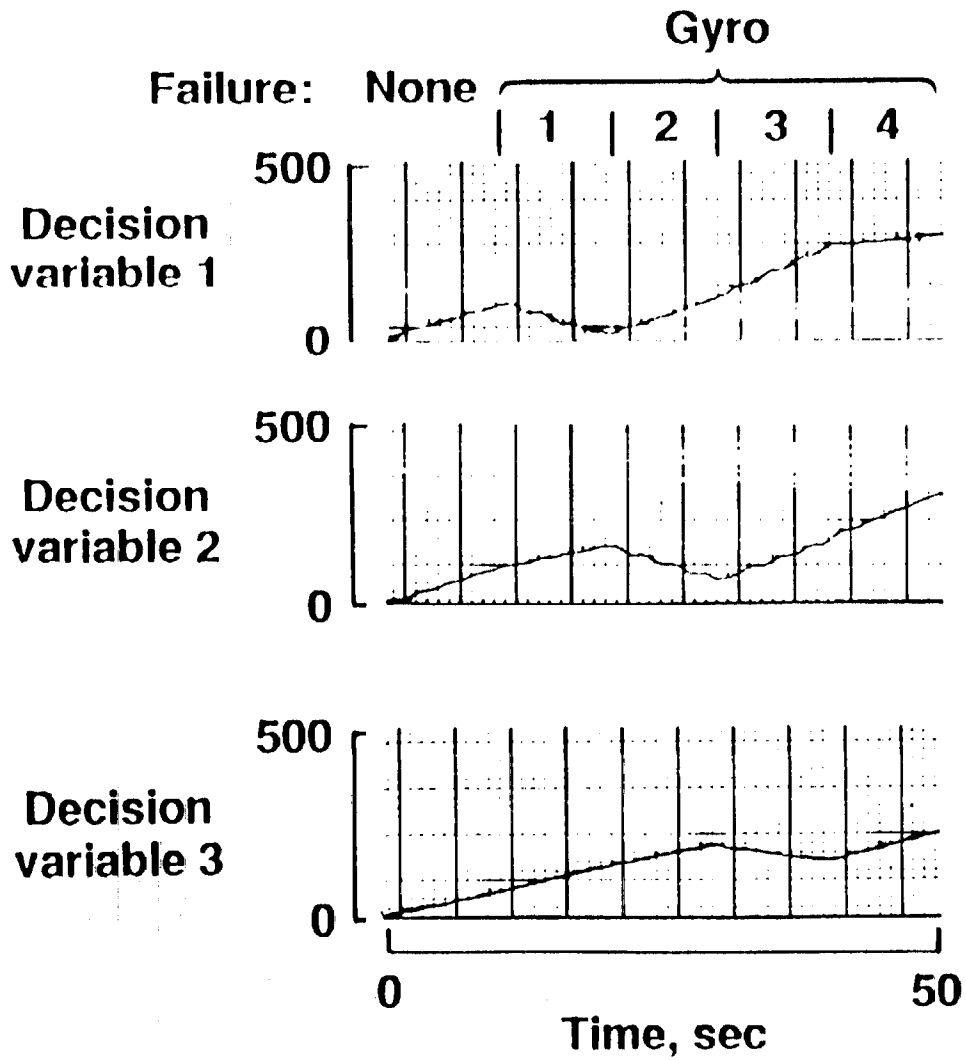
Behavior of the SPRT Decision Variable



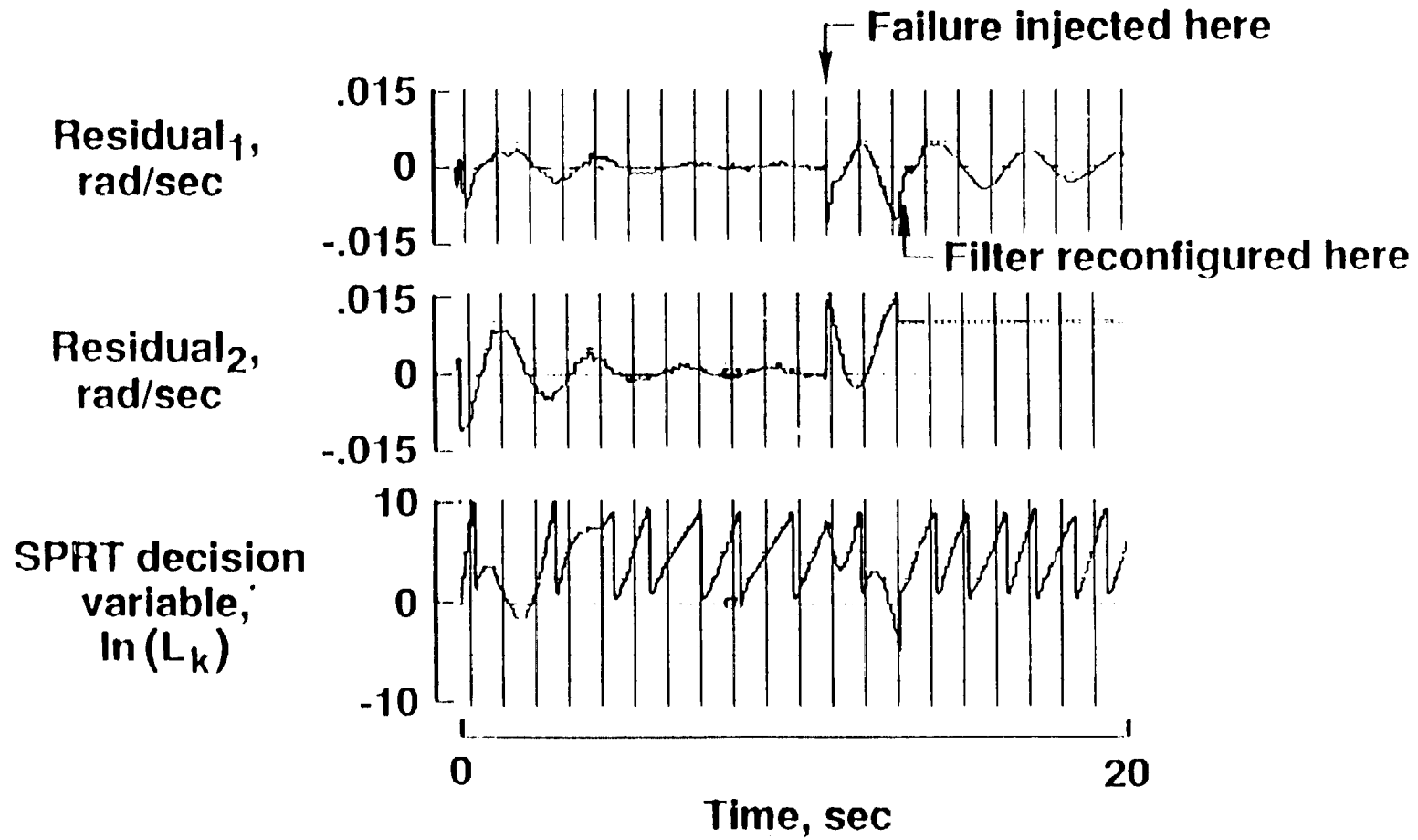
Effect of Bias Failure on Estimation
Langley Grid Experiment



SPRT Decision Variable
Langley Grid Experiment



Failure Detection and Reconfiguration Langley Grid Experiment



SCOLE ARM LQG DESIGN

DESIGN MODEL -- MODAL MODEL

FIXED SCOLE PLATFORM MODEL

NO RIDID BODY MODES

5 LOWEST FREQUENCY VIBRATION MODES

JETS INCLUDED IN FILTER, NOT IN REGULATOR

TORQUE WHEELS USED FOR THE REGULATOR

SCOLE CONFIGURATION

SCOLE PLATFORM FIXED

SENSORS --

MID-MAST AND REFLECTOR ACCELEROMETERS

RATE GYROS ON MAST TIP

ACTUATORS --

JETS ON REFLECTOR

TORQUE WHEELS AT MAST TIP

FUTURE PLANS

VALIDATE NOMINAL KALMAN FILTER

TEST SPRT ON NOMINAL

DESIGN FOR NULL FAILURES OF SENSORS

VALIDATE FAILURE CASE DESIGNS

TEST OVERALL ARM FDI PERFORMANCE

TO BE COMPLETED BY BY MID JUNE '88

Spacecraft COntrols Laboratory Experiment

SCOLE

Software User's Manual

February 1, 1989

619

ABSTRACT

The computer software available to those implementing identification and control algorithms on the Spacecraft Control Labratory Experiment (SCOLE) is described in the following paper. This paper is intended to be used as a guide and reference material.

TABLE OF CONTENTS

INTRODUCTION

SECTION I

The Charles River Data System, commonly referred to as Chuck or the CRDS, is currently being used as the Aerospace Control Research Lab (ACRL) micro-computer for realtime programming for the SCOLE (Spacecraft Controls Laboratory Experiment). The CRDS is based on the Motorola 68020 microprocessor. The operating system is UNIX V with a Korn Shell. The computer has twelve serial ports and one parallel port. Terminals, five answer only communication lines, an originate only communication line, and a line printer are connected to the serial ports.

Accessing the CRDS:

The Charles River Data System (CRDS) can be accessed through LaTS (group name ACRL, group members 59382, 59383, 59384, 59385) or the Communication Machinery Corporation (CMC) Internet hardware and software. (Internet address 128.155.23.20) The following Internet applications are available for the UNIX-UNIX system communication:

1. Remote copy (rcp)
   -transfers file and directory copies
2. Remote login (rlogin)
   -establishes remote UNIX system login
3. Remote shell (rsh)
   -executes one UNIX command on a remote system

The following Internet applications are available for the UNIX-UNIX and/or UNIX-non-UNIX system communications:

1. File transfer (ftp)
   -transfers file and directory copies
2. Network virtual terminal (telnet)
   -establishes remote system login

Upon establishing communications with the CRDS, the user will be prompted for a login name and password. The CRDS system administrator will provide the user with a login name; the user is encouraged to add a password.

Editors:

The Charles River Data System supports three editors-ved, vi, and ed. On-line help is available for each editor by typing "help ved", "man vi | more", "man ed | more". Ved and vi are full screen editors. Ed is a line editor.

File Naming Conventions:

File naming conventions exist on the CRDS. A table of appropriate file name extensions follows. These conventions must be followed to ensure proper functioning of the system procedures.

PROGRAM TYPE	EXTENSION
C source programs	.c
assembly language	.m
object files	.o
FORTRAN source programs	.for
Pascal source programs	.pas
editor backup files	.bak (ved editor only)
listing files	.lst
pre-processed assembly language	.s (FORTRAN 77)

Control Characters:

Control characters influence terminal processes. To use a control character, strike the CTRL key followed by the letter corresponding to the desired command. A table of control characters commonly used on the CRDS follows.

CONTROL CHARACTER	COMMAND EXECUTED
^c	stops command execution returns to prompt level
^d	end of file user logged out
^q	continues display halted with ^s
^r	redisplays current input line
^s	suspends display such as listing contents of file to the screen
^u	erases current line
^t	suspends command execution

Commands:

On-line help is available for the following commands by typing "help" or "describe" followed by the command name:

addname, ar, archive, build, cat, cc, ccom68, cd, change, chmod, chown, command, common, compare, contig, copy, count, cpw, cut, date, debe, debug, delete, describe, difference, diskusage, dump,echo, email, false, find, gcc, hexodec, hist, id, kill, ld, link, list, lpd, lpmaint, lpq, lpr, makedir, match, message, mount, move, nice, p, paste, pmask, #, prfmt, ps, pwd, repeat, resume, setdate, setmask, setpri, setuser, size, sleep, space, su, suname, suspend, sysversion, tail, ted, tee, term, time, translit, true, ttymodes, unmount, uname, unique, uptime, version, wall, wcc, when, whenq, whenrun, which, who, write, yes.

The following summary presents the most commonly used commands on the CRDS:

Command	Description
cat "file(s)"	Displays contents of file(s) or standard input if not supplied
cd "dir"	Change working directory to 'dir'
cp "file1" "file2"	Copy "file1" to "file2"
cp "file(s)" "dir"	Copy "file(s)" into "dir"
date	Displays the date and time
echo "args"	Displays "args"
ln "file1" "file2"	Link "file1" to "file2"
ln "file(s)" "dir"	Link "file(s)" into "dir"
ls "file(s)"	List "file(s)"
ls "dir(s)"	List files in "dir(s)" or in current directory if "dir(s)" is not specified
mkdir "dir(s)"	Create directory "dir(s)"
mv "file1" "file2"	Move "file1" to "file2" (simply rename it if both reference the same directory)
mv "file(s)" "dir"	Move "file(s)" into directory "dir"
ps	List information about active processes
pwd	Display current working directory path
rm "file(s)"	Remove "file(s)"
rmdir "dir(s)"	Remove empty directory "dir(s)"
sort "files(s)"	Sort lines of "file(s)" or standard input if not supplied
wc "file(s)"	Count the number of lines, words, and characters in "file(s)" or standard input if not supplied
who	Display who's logged in

[Source: Charles River Data System's Commands Manual]

Subroutines:

On-line help is available for the following commands by typing "help" or "describe" followed by the subroutine names:

abs, acos, alarm, asctime, asin, astoc, astod, astof, astoi, astol, atan, atan2, atod, atof, atoi, atol, atoq, atos, bsearch, capitalize, ceil, clock, close, conv, cos, cosh, creat, ctime, cvtime, dayofyear, delay, drand, drand48, dup, dup2, cprintf, execl, execle, execlp, execv, execve, execvp, exit, exp, fabs, fbufcnt, fclose, fexecl, fexecle, fexecv, fexecve, fflush, fgetc, fgetline, fileopen, filepos, fileread, fileseek, filesize, filewrite, find, findline, floor, flush, fmod, fopen, format, fprintf, fputc, fputs, frac, fread, frexp, fscanf, fseek, fshow, fsstat, fstat, ftell, ftime, ftoes, ftofs, ftok, fwrite, gamma, gcvt, getallargs, getargs, getbroken, getc, getchar, getcmask, getcwd, gethz, getline, gettime, gmtime, gtty, hypot, index, ioctl, is_addressable, itoq, J0, kill, ldexp, ln, localtime, log, log10, longjmp, lrand, lsearch, lseek, ltoq, map_io_page, modf, open, parseranges, pause, peekc, pipe, poly, pow, proctime, putc, putchar, qadd, qadd3, qcmp, qdiv, qdiv3, qdivi, qmod, qmod3, qmul, qmul3, qneg, qneg2, qsort, qsub, qsub3, qtos, qtoi, qtol, qtoq, qtos, quicksort,

rand, random, randomize, read, rewind, scanf, setjmp, setwd, signal, sin, sinh, sleep, spfun, sprintf, srand, sscanf, ssignal, stat, status, stoq, strcat, strcatl, strchr, strcmp, strcpy, streql, strindex, strlen, strncat, strncmp, strncpy, strrchr, system, tan, tanh, tell, time, times, timestr, timetostr, ttyname, wait, y0.

Compilers:

FORTRAN 77, C, and Pascal compilers are available on the CRDS. On-line help is available for each compiler by typing "help" followed by f77 (FORTRAN 77), cc (C), and pascal (Pascal).

Many macros exist to compile, link and assemble code. The following is a brief description of each macro. On-line help is available for cm, af, and as by typing "help" followed by the macro name.

cm:

file: FORTRAN 77 main program source code "main.for"
command: cm main (prefix only)
use: produce object code

as:

file: FORTRAN 77 subroutine source code "sub.for"
command: as sub (prefix only)
use: produce object code

af:

file: FORTRAN 77 main program source code "main2.for"
command: af main2 (prefix only)
use: produce executable code by linking object code to all available system libraries (SCOLE, math, realtime, strip chart included)
executable code is stored in the file "main2"
Note: This macro is not to be used with main programs calling subroutines not found in the system libraries.

C:

file: C main program or subroutine source code "main.c" or "sub.c"
command: C main or C subc (prefix only)
use: produce C object code main.o or subc.o

S:

file: FORTRAN pre-processed assembly code "file.s" produced with f77 command
command: S file (prefix only)
use: produce assembly language code file.m

A:

file: FORTRAN assembly code "file.m" produced with S macro
command: A file (prefix only)
use: produce object code file.o ready to be linked.

Note: macros cm and as perform the same functions as the f77 command used in conjunction with the S and A macros.

Linker:

The CRDS linker command is "ld". On-line help is available for the linker by typing "help ld". The standard link line is as follows:

ld FORTRAN main.o C/FORTRAN subroutine(s).o -l(library names) o=fn

ld C main.o C/FORTRAN subroutine(s).o -l(library names) o=fn

(fn = executable file name)

One or more of the following libraries may be linked to the object code:

LIBRARY			COMMAND
scole	:	SCOLE FORTRAN library	(-lscole)
real	:	C realtime library	(-lreal)
mt8500	:	C MT8500 strip chart recorder library	(-lmt8500)
matx	:	FORTRAN matrix library	(-lmatx)
unos	:	UNOS system library Note: This library must always be linked and must be linked as the last library.	(-lunos)

On-Line Help:

Every effort has been made to make the CRDS a user-friendly machine. Help and system updates are updated as required. The following menu presents all of the on-line help and update reports of the CRDS.

Should any questions arise pertaining to software and/or hardware problems with the CRDS, please feel free to contact Danette B. Lenox at x46619 or x44473 ((804)864-6619 or (804)864-4473).

On-line help is available on the following topics:

1. SCOLE Help:
   A. type "h_make" for help with the SCOLE program build macro "make"
   B. type "flow" for a copy of the SCOLE program flowchart
2. System bug report:
   A. UNOS VII bugs type "b_UNVII"
   B. UNIX bugs type "b_UNIX"
   C. C compiler bugs type "b_UNCII"
   D. FORTRAN compiler bugs type "b_FORTRAN"

3. System updates:
   A. type "updates"
4. Basic system subroutine and command help:
   A. type "list_all" to see a listing of the names of the basic commands for which help is available
5. Compiler help:
   A. type "absoft" for help with the absoft FORTRAN77 compiler
   B. type "comp_pascal" for help with the PASCAL compiler
6. Library help:
   A. type "real" for help with the real time library
   B. type "math" for help with the matrix math library
   C. type "LINPACK" for help with the LINPACK libraries
   D. type "mt8500" for help with the MT8500 strip chart recorder library

SCOLE SOFTWARE OVERVIEW

SECTION II

The SCOLE model is operated by a set of software generically referred to as the SCOLE program. Control laws can be easily implemented and tested on the model through minor SCOLE software alterations.

1. SCOLE Files

Each user must create a directory called "SCOLE" on his account. Subroutine and main program source and object code will be located on directory SCOLE. Files main.for, cpmenu.for, preexp.for, posexp.for, rtloop.for, makefile, gains.dat and select.set on directory /lib/SCOLE/TEMPLATE should be copied onto the user's SCOLE directory.

These files perform the following functions:

A. main.for :
   .. source code main program

B. cpmenu.for:
   .. source code for controls parameters menu

C. posexp.for :
   .. source code for post-experiment menu

D. preexp.for :
   .. source code for pre-experiment menu

E. rtloop.for :
   .. source code for real time loop

F. makefile :
   ..makefile used to create SCOLE program (refer to topic #2)

G. gains.dat :
   ..data file used by SCOLE library routines altered by variable setting in menus

H. select.set :
   ..data file used by SCOLE library routines altered by variable setting in menus

Each of these files should be altered according to the user's requirements. The codes have been "commented" to provide guidelines for alterations. All additional source code for subroutines written by the user for implementation of the SCOLE contol law should be located on SCOLE and must have the appropriate extension.

2. Building SCOLE Program

The command "make" builds the SCOLE program with the code located on the user's SCOLE directory using the directions found in makefile, the math library (/lib/libmatx.a), the strip chart library (/lib/libmt8500.a), the real time library (/lib/libreal.a), the UNOS library (/lib/libunos.a), and the SCOLE library (/lib/libscole.a).

The command is invoked by typing:

make [exec=name].

The SCOLE program executables are by default written to file 'scole'. Should the user desire to save the executables in another file, simply append "exec = name" where name is an arbitrary file name to which the executables are written.

The formation of the program is based on relationships expressed in 'makefile' as follows:

The modification dates of source code on the user's SCOLE directory are compared to that of the corresponding object code on SCOLE. If the date of the source code is more recent, the subroutine is recompiled. The modification dates of the two SCOLE include files (/HOLD/SCOLE/varibs.comm and /HOLD/SCOLE/ blk.comm) and any user supplied include files are also checked. If these dates are more recent than the object codes including the files, the corresponding source is recompiled. If errors occur in compilation, formation of the SCOLE program will not be successful. FORTRAN syntax errors can be noted by studying the ".lst" file related to the subroutine in question. If no errors occur during compilation, the object code is linked to the libraries listed above, forming a set of executable code stored in the file scole or "name". Watch out for "unresolved externals". When user adds files to the SCOLE directory, the source name must be added to the FSUBS list in makefile and the object name must be added to the FOBJ list in makefile. When the user adds include files to the SCOLE directory the file name must be added to the INCVC list.

3. Running the SCOLE Program

The SCOLE program is run by simply typing scole or "name".
If the user encounters timeouts during the real time execution, increase the sample period and please contact the system administrator for information on running the program with priority privileges.

4. Scole Software Variables

The following variables are commonly used in the SCOLE software environment and are passed by way of common blocks in include file /HOLD/SCOLE/varibs.comm:

VARIABLE	DESCRIPTION
shurgr	shuttle rate gyro roll
shurgp	shuttle rate gyro pitch
shurgy	shuttle rate gyro yaw
refrgr	reflector rate gyro roll
refrgp	reflector rate gyro pitch
refrgy	reflector rate gyro yaw
shaccx	shuttle acceleration x direction
shaccy	shuttle acceleration y direction
shaccz	shuttle acceleration z direction
m1accx	mast station 1 -upper- acceleration x direction
m1accy	mast station 1 -upper- acceleration y direction
m2accx	mast station 2 -lower- acceleration x direction
m2accx	mast station 2 -lower- acceleration y direction
refacx	reflector acceleration x direction
refacy	reflector acceleration y direction
shtorr	shuttle torque roll
shtorp	shuttle torque pitch

shtory	shuttle torque yaw
refwhx	reflector wheel x direction
refwhy	reflector wheel y direction
refwhz	reflector wheel z direction
m1whx	mast station 1 -upper- wheel x direction
m1why	mast station 1 -upper- wheel y direction
m2whx	mast station 2 -lower- wheel x direction
m2why	mast station 2 -lower- wheel y direction
tx	thrust x direction
ty	thrust y direction
phi	bank angle
theta	pitch angle
psi	yaw angle
p	shuttle rotational velocity about x
q	shuttle rotational velocity about y
r	shuttle rotational velocity about z
fx	thruster gains x direction
fy	thruster gains y direction
ctheta	cosine theta
stheta	sine theta
cpsi	cosine psi
spsi	sine psi
cphi	cosine phi
sphi	sine phi
bls	thruster least square coefficient

5. SCOLE FORTRAN Library Routines

The following subroutines are found in the SCOLE library. Users are not given privileges to alter the subroutines.

MT8500 STRIP CHART RECORDER SUBROUTINES:

chan	change number of strip chart channels
cscale	accept new strip chart scale factors
lbmenu	set strip chart recorder labels
screc	replays saved data to strip chart
spdset	set strip chart recorder speed
stmenu	strip chart recorder menu
strcht	send data to the strip chart recorder
strset	select program variable to be output to the strip chart recorder

PARAMETER DEFINITION AND FILE I/O SUBROUTINES:

colint	gain and dead band levels for thruster slew menu
ditime	duration and interval time change
ertrap	error trapping in keyboard input
iselek	user menu selection option
lhelp	on-line help
readf	reads standard SCOLE data files
skipln	skips line
tscole	get current date

CMG SUBROUTINES: (CMG software in process of being updated)

cmgtor	send 3 axis shuttle torque commands to cmg controller
cmgtst	cmg test
gettor	get shuttle torque commands from joystick
gimbal	cmg gimbals - four torque commands to gimbals
sensor	prints scaled sensor measurements from cmgs

MEASUREMENT/DATA ACQUISITION SUBROUTINES:

attit	determine roll and pitch attitude of shuttle from accelerometers and rate sensor
damint	thruster damping menu
getac	get accelerometer data
getyz	get y and z data from the op_eye
lsqang	least square shuttle attitude estimation
sbias	determines sensor bias by averaging over twenty seconds
slwint	accepts gain and dead band levels
toggle	toggle electromagnet (not in use)

OPTICAL SENSOR SUBROUTINES:
(Optical Sensor software in process of being updated)

opacal	calibrate optical sensor
opang	returns yaw and pitch angles as determined from optical sensor outputs
opashw	prints yaw and pitch angles as determined from optical sensor outputs
opatst	test optical sensor
opmenu	sets up optical sensor calibration and test
qqstop	stops program
yz	get y and z data from the op_eye

DATA OUTPUT SUBROUTINES:

stofil	store experimental data in arrays to be output to file
stoset	select program variable to be saved on file
uwrite	output data to file

6. SCOLE Realtime Library Calls

The realtime library consists primarily of the following FORTRAN callable C routines. Help is available for each routine by typing "describe" followed by the routine name in question.

getadc -
This subroutine samples a range of analog-to-digital converters. The +/- 10.0 volt input range is scaled to +/- 1.0 units so a single unit is worth .0049 volts.

iswtch -
This subroutine checks the state of a switch as low or high.

oswtch -
This subroutine sets the state of a switch as low or high.

rtime -

An internal memory mapped timer is used to control the timing of real-time operations in the Charles River Computer. The programmable clock is required to generate a start pulse and a stop pulse for each sampling interval of the control process. The maximum interval is eighty-five seconds and the minimum interval is 5 micro-seconds. The timer is a Motorola M68230 Parallel Interface Timer (PI/T) which provides versatile double buffered parallel interfaces and 24-bit programmable timer.

setdac -

This subroutine samples a range of digital-to-analog converters. The +/- 1.0 unit output range is scaled to +/- 10.0 volts so a single unit is worth .0049 volts.

strip -

This function outputs data to the analog strip chart recorder.

thrust -

This function sets the states of the four discrete outputs on the Parallel Interface/Timer. It was designed for the thrusters on the SCOLE facility. The arguments can have one of three values: 1, 0, or -1 corresponding to positive, none, and negative thrust respectively.

rnum -

This function returns the current date since Jan 1, 1980 in seconds.

mprep -

This function clears the terminal screen (Wyse50 terminal).

mtfeed-

This function advances the paper in the MT8500 chart recorder to the next top-of-form mark (form feed).

mtrls-

This function releases the MT8500 chart recorder from the real-time mode and leaves it in a ready state.

mtstar-

This function starts the MT8500 chart recorder's waveform printing - printing of scale grid can be turned off and on via subroutine grid argument.

mtstop-

This function stops the waveform printing from the MT8500 chart recorder.

mtspee-

This function sets the waveform printing speed of the MT8500 recorder.

mtchan-

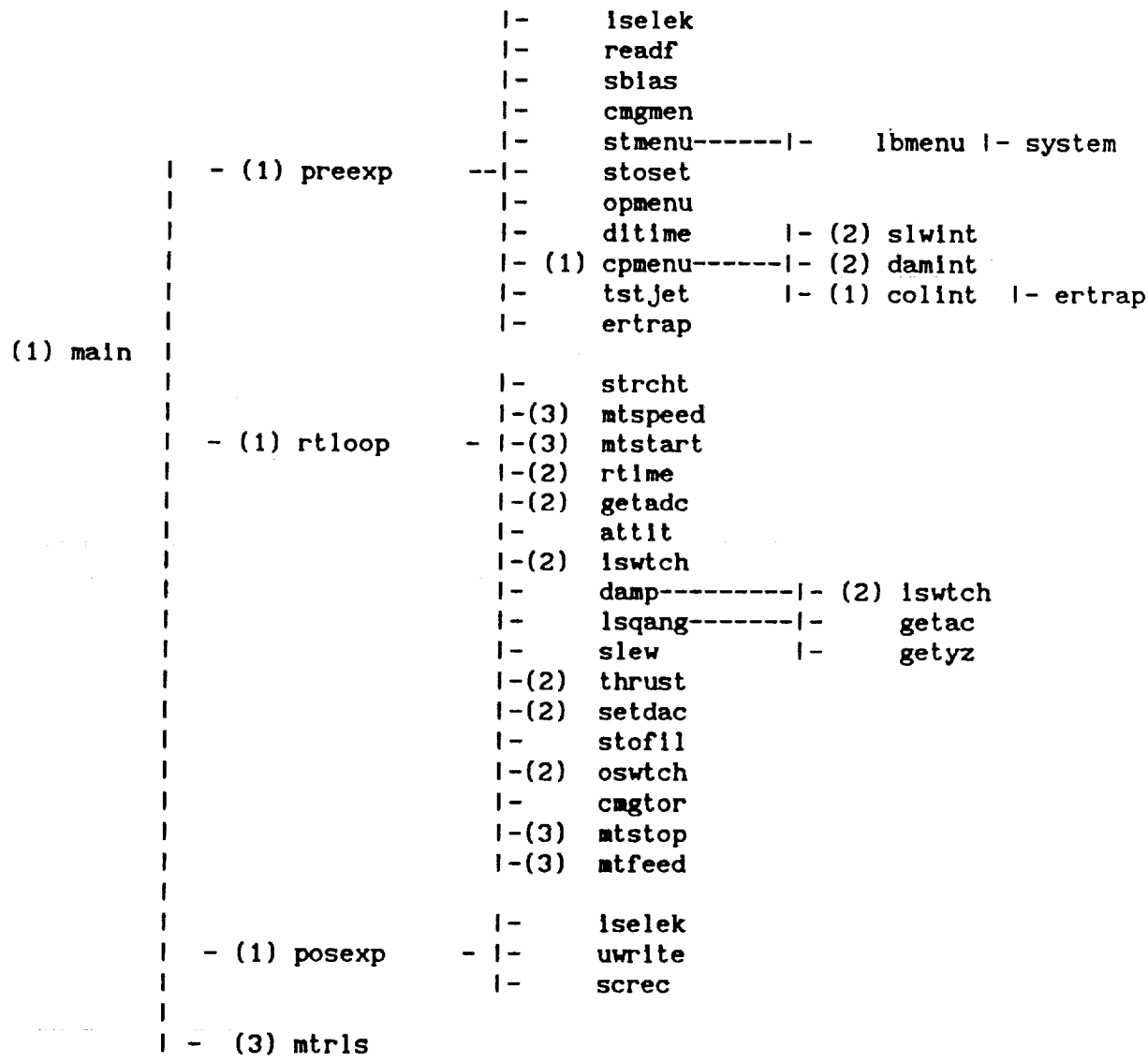
This function sets the number of channels to be printed from the MT8500 recorder.

cmgtor- (software in process of being updated)
This function commands the torque wheels.

A table of the C subroutines commonly called from the SCOLE program follows.

SUBROUTINE/ FUNCTION	LIBRARY	C DECLARATIONS	FORTRAN DECLARATIONS
thrust	real	int thrust(x,y) int *x, *y	integer thrust integer x, y
iswtch	real	short iswtch(n) int *n	logical*2 iswtch integer n
oswtch	real	short oswtch(state,n) int *state, *n	integer*2 oswtch integer n, state
rtime	real	int rtime(flag,tau) float *tau int *flag	logical*4 rtime real tau logical flag
setdac	real	long setdac(dac_data_pointer, last_dac,first_dac)	integer*4 setdac real dac_data_pointer, last_dac,first_dac
getadc	real	long getadc(adc_data_pointer, last_adc,first_adc)	integer*4 getadc real adc_data_pointer, last_adc,first_adc
mtfeed	mt8500	long mtfeed()	integer*4 mtfeed
mtrls	mt8500	long mtrls()	integer*4 mtrls
mtstar	mt8500	long mtstar(grid) int *grid	integer*4 mtstar integer grid
mtstop	mt8500	long mtstop()	integer*4 mtstop
mtspee	mt8500	long mtspee(units_arg,speed_arg) char *units_arg float *speed_arg	integer*4 mtspee character*10 units_arg real speed_arg
mtchan	mt8500	long mtchan(nchan) long *nchan	integer*4 mtchan integer nchan
cmgtor	cmg	long cmgtor(torz,tory, torz) real torz, tory, torx (software in process of being updated)	integer*4 cmgtor
strip	real	long strip(data) int *data	integer*4 strip real data
mprep	real	long mprep()	integer*4 mprep
rnum	real	long rnum(ans) long *ans	integer*4 rnum integer*4 ans

7. SOFTWARE FLOWCHART



(1) user file
(2) C realtime library routine (/lib/libreal.a)
(3) C strip chart library routine (/lib/libmt8500.a)
all other routines from SCOLE library (/lib/libscole.a)

SCOLE TEMPLATE FILES SOURCE CODE(/lib/SCOLE/TEMPLATE)

SECTION III

```
      program  main
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                    D O C U M E N T A T I O N                          c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c   This main program calls the 3 main modules - preexperiment,
c   real time loop and postexperiment. This is the main calling module.
c
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c    D A T A  /  V A R I A B L E  D I C T I O N A R Y                    c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
      include /HOLD/SCOLE/varibs.comm
c
      integer iabrt
      common /store/ bias(25)
      common /scale/  sfsrr, sfsrp, sfsry, sfrrr, sfrrp, sfrry,
     +                sfsax, sfsay, sfsaz, sfm1x, sfm1y, sfm2x, sfm2y,
     +                sfrax, sfray, sfstr, sfstp, sfsty
      common /dimen/  xd, yd, zd
      common /cdebug/dbug
      dimension stodat(10,10000)
c
      logical dbug
      character ans*3
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c              P R O C E D U R E     S E C T I O N                       c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      dbug=.false.
      iabrt = 0
c
c       USERS CAN ADD CODE TO READ FILES, PERFORM INITIALIZATION OPERATIONS,
c       OR WHATEVER ONE DESIRES IN THIS CODE
c
 10   if(dbug)print*,'DEBUG MODE ON'
c
c       THE MAIN PROGRAM FIRST CALLS SUBROUTINE PREEXP WHICH PRESENTS
c       THE PRE-EXPERIMENT MENU BY WHICH USERS CAN INTERACTIVELY SET
c       UP PARAMETERS
c
      call preexp (iabrt)
c
c       IABRT IS A VARIABLE SET TO 1 IN "PREEXP.FOR" WHEN THE USER CHOOSES
c       TO EXIT THE SCOLE PROGRAM
c
      if(dbug)print *,'call preexp from main'
      if (iabrt.eq.1) goto 10000
      if(dbug)print *,'call rtloop from main'
c
c       NEXT THE MAIN PROGRAM CALLS SUBROUTINE RTLOOP WHICH CONTAINS THE
c       REAL TIME LOOP OR THE ACTUAL EXPERIMENT
```

```
c
      call rtloop(stodat)
c
c       STODAT IS THE ARRAY IN WHICH DATA ARE SAVED IN THE REAL TIME LOOP
c       VARIABLES TO SAVE ARE CHOSEN BY SELECTING THE MAIN MENU OPTION TO
c       SET-UP THE DATA FILE
c
      if(dbug)print *,'call posexp from main'
c
c       AFTER THE EXPERIMENT IS COMPLETE, THE MAIN PROGRAM CALLS POSEXP -
c       THE POST-EXPERIMENT MENU FOR DATA PROCESSING
c
      call posexp (iabrt,stodat)
      if (iabrt.eq.1) goto 10000
      goto 10
10000 continue
c
c     RELEASE MT8500 STRIP CHART RECORDER
c
      im = mtrls()
      stop
      end
c
c*************************************************************************
c                                                                        *
c    Block data subprogram                                               *
c                                                                        *
c*************************************************************************
c
c       USERS SHOULD NOT DELETE ANY OF THE INCLUDE FILES BUT ARE FREE
c       TO ADD ANY OF THEIR OWN FILES OR DATA
c
      blockdata const
c
      include /HOLD/SCOLE/varibs.comm
      include /HOLD/SCOLE/blk.comm
c
      end
```

```
      subroutine  cpmenu
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c      This is the menu for 'SET UP CONTROL PARAMETERS' module.
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     D A T A  /  V A R I A B L E          D I C T I O N A R Y         c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      integer     ians, mode, lhno, idef, iopt
      character   nans*1, procse*20, hlpfil*20
      common /info/ procse, hlpfil
      common/cdebug/dbug
      logical dbug
      integer*4 iprep, mprep
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c               P R O C E D U R E      S E C T I O N                    c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      if(dbug)print*,'Subroutine cpmenu: '
c
c     USERS CAN ADD AND/OR DELETE CONTROL PARAMETER OPTIONS IN THIS MENU
c
c     THIS MENU IS PRESENTED AS AN ILLUSTRATION
c
10    iprep = mprep()
      write(*, * )'-------------------------------------------'
      write(*, * )'|        ** CONTROL PARAMETER MENU **      |'
      write(*, * )'-------------------------------------------'
      write(*, * )'|  1) SET THRUSTER DAMPING PARAMETER.      |'
      write(*, * )'|  2) SET SLEW PARAMETER.                  |'
      write(*, * )'|  3) SET COLLOCATED MAST DAMPER PARAMETERS|'
      write(*, * )'|  4) RETURN TO PREVIOUS MENU.  [DEFAULT]  |'
      write(*, * )'-------------------------------------------'
      imaxp = 4
      lhno = 10
      idef = 4
      imode = 7
      call iselek (imaxp,imode,ians,nans,lhno,idef)
      if (nans.eq.'M') then
      goto 10
      else
      iopt = ians
      end if
c
c     USERS SHOULD INCLUDE/DELETE OPTION CODE IN THIS IF...THEN...ELSE BLOCK
c     TO CORRESPOND TO HIS/HER OPTIONS
      if (iopt.eq.1) then
      call damint
      else if (iopt.eq.2) then
      call slwint
      else if (iopt.eq.3) then
      call colint
      else if (iopt.eq.4) then
      goto 10000
      end if
      goto 10
10000 return
      end
```

```
      subroutine  posexp (iabrt,stodat)
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                    D O C U M E N T A T I O N                          c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c     This is the menu for the 'POST-EXPERIMENT MENU' module
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     D A T A  /  V A R I A B L E          D I C T I O N A R Y        c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      save dbug
      dimension stodat(10,10000)
      integer     iabrt,  ians, mode, lhno, maxp, idef, runs
      character   nans*1, procse*20, hlpfil*20
      common /time/ period, ttime, runs
      common /info/procse,hlpfil
      common/cdebug/dbug
      logical dbug
      integer*4 iprep, mprep
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c               P R O C E D U R E      S E C T I O N                   c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      if(dbug)print*,'Subroutine posexp: '
c
c       OUTPUT FILE FORMATS INCLUDE MATRIX-X, MATLAB, FORMATTED, UNFORMATTED
c       FILES OPTIONS PRESENTED IN LIBRARY ROUTINE UWRITE
c
10    iprep = mprep()
c
c       USER IS GIVEN THE CHOICE AT THE END OF THE EXPERIMENT TO
c       SAVE DATA IN A FILE, REPLAY IT TO THE STRIP CHART, OR
c       END THE EXPERIMENT
c
      write(*,*)
      write(*,*)'----------------------------------'
      write(*,*)'|   ** POST-EXPERIMENT MENU **   |'
      write(*,*)'----------------------------------'
      write(*,*)'|  1) PREPARE FOR NEXT RUN       |'
      write(*,*)'|  2) SAVE DATA TO OUTPUT FILE   |'
      write(*,*)'|  3) REPLAY SAVED DATA TO STRIP |'
      write(*,*)'|             CHART              |'
      write(*,*)'|  4) STOP EXPERIMENT            |'
      write(*,*)'----------------------------------'
c
      lhno = 10
      mode = 6
      idef = 0
      maxp = 4
      call iselek (maxp,mode,ians,nans,lhno,idef)
      if (nans.eq.'M') then
          goto 10
      else if (ians.eq.1) then
          goto 10000
      else if (ians.eq.2) then
          call uwrite(runs,period,stodat)
      else if (ians.eq.3) then
          call screc
```

```
      else
          write(*,124)
124       format(' Are you sure ? (y/n) - ',\)
          maxp = 0
          mode = -13
          call iselek (maxp,mode,ians,nans,lhno,idef)
          if (nans.eq.'Y') then
             iabrt = 1
             goto 10000
          end if
      endif
      goto 10
10000 continue
c
c     USERS CAN ADD POST-PROCESSINF CODE AT THIS POINT
c
      end
```

```
      subroutine  preexp (iabrt)
ccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c   This is the menu for the 'PRE-EXPERIMENT MODULE' cc
ccccccccccccccccccccccccccccccccccccccccccccccccccccccc

ccccccccccccccccccccccccccccccccccccc
c           DOCUMENTATION             c
ccccccccccccccccccccccccccccccccccccc

ccccccccccccccccccccccccccccccccccccc
c         PROCEDURE SECTION           c
ccccccccccccccccccccccccccccccccccccc

c
      include /HOLD/SCOLE/varibs.comm
      common /info/  procse, hlpfil
      common /cdebug/dbug
      logical dbug
      integer    c, rret, iabrt, runs, ians, idef, mode, lhno, maxp
      integer system
      integer*4 mprep, iprep
      character  schar*1, nans*1, procse*20, hlpfil*20
      if(dbug)print*,'Subroutine preexp:'
c
c     Set default values
c
      iabrt  = 0
c
c       SCOLE LIBRARY ROUTINE TO READ FILES GAINS.DAT AND SELECT.SET ON SCOLE
c
      call readf
      if(dbug)print*,'Returned from readf:'
1000  continue
900   iprep=mprep()
c
c       TTIME AND PERIOD ARE SUPPLIED BY THE USER AS REQUESTED IN THE MAIN
c       MENU
c
      runs=ttime/period
c
c       USERS SHOULD READ ALL OF THEIR DATA FILES AT THIS POINT
c       ALL VARIABLES READ SHOULD BE PASSED AMONG THE SUBROUTINES
c       ON HIS ACCOUNT BY MEANS OF INCLUDE FILES
c
c       USERS ARE FREE TO ADD/DELETE MENU OPTIONS AS LONG AS THEY TAKE
c       INTO ACCOUNT HOW THE OPTIONS ARE USED AND WHAT SUBROUTINES
c       THEY CALL
c
      write(*, * )'-----------------------------------------'
      write(*, * )'|        ** PRE-EXPERIMENT MAIN MENU **    |'
      write(*, * )'-----------------------------------------'
      write(*, * )'|  1) OBTAIN RATE SENSOR AND ACCELEROMETER |'
      write(*, * )'|     BIAS.                                |'
      write(*, * )'|  2) SET UP AND TEST CMGS.                |'
      write(*, * )'|  3) SET UP STRIP CHART.                  |'
      write(*, * )'|  4) SET UP DATA FILE.                    |'
      write(*, * )'|  5) SET UP OPTICAL SENSOR                |'
```

```
      write(*, * )'|  6) SET UP DURATION AND INTERVAL TIME.    |'
      write(*, * )'|  7) SET UP CONTROL PARAMETERS.            |'
      write(*, * )'|  8) TEST JET THRUSTERS.(NOT OPERATIONAL)  |'
      write(*, * )'|  9) START EXPERIMENT                      |'
      write(*, * )'| 10) TOGGLE DEBUG MODE                     |'
      write(*, * )'| 11) ENTER OPERATING SYSTEM                |'
      write(*, * )'| 12) QUIT.                                 |'
      write(*, * )'-------------------------------------------'
c
      lhno = 10
      mode = 6
      idef = 0
      maxp = 12
      call iselek (maxp,mode,ians,nans,lhno,idef)
      iprep = mprep()
      if (nans.eq.'M') then
         goto 900
      else
         iopt = ians
      end if
c
      if (iopt.eq.1) then
         call sbias(period)
      else if (iopt.eq.2) then
         print*,'*******THIS SUBROUTINE NOT IMPLEMENTED*******'
c        call cmgmen
      else if (iopt.eq.3) then
         call stmenu
      else if (iopt.eq.4) then
         call stoset
      else if (iopt.eq.5) then
         call opmenu
      else if (iopt.eq.6) then
         call ditime(period,ttime)
      else if (iopt.eq.7) then
         call cpmenu
      else if (iopt.eq.8) then
c         call tstjet
c         this subroutine is not in use at this time 1-20-87
          print*,'*******THIS SUBROUTINE NOT IN USE AT THIS TIME*******'
          go to 900
      else if (iopt.eq.9) then
          goto 10000
      else if (iopt.eq.10) then
          dbug = .not. dbug
      else if (iopt.eq.11) then
          write(*,*)' Use ^D to return to this program'
          is = system("command")
      else if (iopt.eq.12) then
          write(*,124)
124       format(' Are you sure ? (y/n) - ',\)
          maxp = 0
          mode = -13
          call iselek (maxp,mode,ians,nans,lhno,idef)
          if (nans.eq.'Y') then
            iabrt = 1
            goto 10000
```

```
          else
            goto 900
          end if
        endif
        goto 900
c
100     format (a,f7.2,a,f6.4,a)
c
9996    call ertrap(ios,1,0)
        goto 1000
9997    call ertrap(ios,1,0)
        goto 900
10000   return
        end
```

```
c*********************************************************************
c                                                                     *
c   This is the main subroutine for the real-time loop - all          *
c   estimation, control, and data storage functions should be         *
c   accomplished in this subroutine.                                  *
c                                                                     *
c*********************************************************************
c
      subroutine rtloop(stodat)
c
c       USE COMMON BLOCKS TO PASS DATA BETWEN REAL TIME ROUTINES TO
c       TO IMPROVE PROCESSING SPEED.
c
      include /HOLD/SCOLE/varibs.comm
      save
      dimension dac0(8)
      dimension stodat(10,10000)
      dimension adc(18), dac(8), sfsens(18),  adcs(18)
      common/cdebug/dbug
      common/store/bias(25)
      common /scale/  sfsrr, sfsrp, sfsry, sfrrr, sfrrp, sfrry,
     +                sfsax, sfsay, sfsaz, sfm1x, sfm1y, sfm2x, sfm2y,
     +                sfrax, sfray, sfstr, sfstp, sfsty
      common /dimen/ xd,yd,zd
c
c     The Absoft compiler accepts only the first array argument and first
c     element of the common block to which the first array argument is
c     equivalenced.  All subsequent equivalences are performed in the order
c     of array elements and common block variables
      equivalence ( adcs(1),   shurgr)
c     equivalence ( adcs(1),   shurgr)
c     +           ,( adcs(2),  shurgp)
c     +           ,( adcs(3),  shurgy)
c     +           ,( adcs(4),  refrgr)
c     +           ,( adcs(5),  refrgp)
c     +           ,( adcs(6),  refrgy)
c     +           ,( adcs(7),  shaccx)
c     +           ,( adcs(8),  shaccy)
c     +           ,( adcs(9),  shaccz)
c     equivalence ( adcs(10), m1accx)
c     +           ,( adcs(11), m1accy)
c     +           ,( adcs(12), m2accx)
c     +           ,( adcs(13), m2accy)
c     +           ,( adcs(14), refacx)
c     +           ,( adcs(15), refacy)
c     +           ,( adcs(16), shtorr)
c     +           ,( adcs(17), shtorp)
c     +           ,( adcs(18), shtory)
c
      equivalence (dac(1),  refwhx)
c
      equivalence (sfsens(1),  sfsrr)
c     equivalence (sfsens(1),  sfsrr)
c     +           ,(sfsens(2),  sfsrp)
c     +           ,(sfsens(3),  sfsry)
c     +           ,(sfsens(4),  sfrrr)
```

```
c     +          ,(sfsens(5),  sfrrp)
c     +          ,(sfsens(6),  sfrry)
c     +          ,(sfsens(7),  sfsax)
c     +          ,(sfsens(8),  sfsay)
c     +          ,(sfsens(9),  sfsaz)
c     +           ,(sfsens(10), sfm1x)
c     +          ,(sfsens(11), sfm1y)
c     +          ,(sfsens(12), sfm2x)
c     +          ,(sfsens(13), sfm2y)
c     +          ,(sfsens(14), sfrax)
c     +          ,(sfsens(15), sfray)
c     +          ,(sfsens(16), sfstr)
c     +          ,(sfsens(17), sfstp)
c     +          ,(sfsens(18), sfsty)
c
c Function declarations
c
      integer*2 oswtch,io
      logical*2 iswtch, il, ils
      logical*4 rtime
      integer*4 cmgtor
      integer*4 iprep, mprep
      integer*4 setdac , getadc , thrust
      integer*4 mtspeed, mtstart, mtstop, mtrls, mtfeed
c
      character*1 schar
c
      logical flag, reset, cmgclr,dbug
      logical*4 insw(8)
      integer c, rret
      integer runs
      integer reply, x, y, dampx, dampy, slewx, slewy
c
      data insw  / 8*.false. /
      data dac0  / 8*0.0 /
      if(dbug)print*,'Subroutine rtloop: '

c
c
c         DELETE THIS CALL TO STRCHT IF THE USER DOES NOT WISH TO
c         USE THE MT8500 STRIP CHART RECORDER
c
      call strcht
      iprep = mprep()
c
      write(*,*)
      write(*,124)
124   format(' EXPERIMENT RUNNING.
     1/,'       WAIT FOR MENU')
c
      reset = .true.
      flag = .true.
c
      refrgr=0.00
      refrgp=0.00
      refrgy=0.00
c
```

```
      shtorr = 0.0
      shtorp = 0.0
      shtory = 0.0
c
c       START STRIP CHART RECORDER - MT8500
c       OPTIONAL CODE - STRIP CHART NOT NECESSARY
c       DELETE LINES BETWEEN ASTERISKS IF STRIP CHART NOT DESIRED
c*****************************************************************
      im = mtspeed(strspd, trim(spdunt)//char(0))
      im = mtstart(1)
c*****************************************************************
c
c       DO LOOP FOR REAL TIME EXECUTION
c       RUNS = DURATION/SAMPLE RATE AS SET IN MAIN MENU
c
      do 1 i=1,runs
c
      if(dbug)print*,'.....call rtime'
c
c       'IF(RTIME(PERIOD,FLAG))' THEN TIMEOUT HAS OCCURRED
c       "RTIME" PERFORMS THE DISCRETE SAMPLE INTERVAL SYNCHRONIZATION
c
      if (rtime(period,flag)) then
910           write(*,125)
125           format(' Do you wish to continue ? (y/n) - ',\)
              read(*,'(a',err=910) schar
              if ( schar .eq. 'n' .or. schar .eq. 'N' ) goto 10000
              ir = rtime(period,.true.)
      endif
c
c     Acquire sensor data, remove bias, and scale.
c
c       GET MEASUREMENT DATA AND SCALE
c
c       SEE NASA TM89057, "DESCRIPTION OF THE SPACECRAFT CONTROL LABORATORY
c       EXPERIMENT (SCOLE) FACILITY" FOR INFORMATION ON THE SENSOR CHANNELS
c
c       SEE EQUIVALENCE AT THE TOP OF THIS SUBROUTINE FOR SCOLE VARIABLE
c       ASSOCIATION WITH "ADC"
c
c       BIAS IS DETERMINED BY SUBROUTINE "SBIAS.FOR"
c
      ig = getadc(0,17,adc)
c
      do 10 kk = 1,18
10          adcs(kk) = (adc(kk) - bias(kk)) / sfsens(kk)
c
c       OPTIONAL - KALMAN FILTER-BASED SHUTTLE EULER ANGLE ESTIMATE
c
      call attit( reset )
c
c       THE CODE BETWEEN THE ASTERIKS IS AN EXAMPLE OF A SET OF CONTROL
c       LAWS INCORPORATING THE SWITCH BOX FOR OPERATOR INTERVENTION
c
c       USERS SHOULD PUT THEIR OWN CONTROL LAW OR ESTIMATION ALGORITHM HERE
c
c
```

```
c*********************************************************************
      x = 0
      y = 0
c
c   Determine vibration suppression commands only if slew is not activated
c
      dampx = 0
      dampy = 0
      if(  iswtch(1) ) then
             call damp( dampx, dampy )
             x = dampx
             y = dampy
      endif
c
c   Determine slew commands only if vibration suppresion is not activated
c
      slewx=0
      slewy=0
      if(  iswtch(3) ) then
             call slew( slewx, slewy )
             x = slewx
             y = slewy
      endif
      if( iswtch(1) .and. iswtch(3) ) then
             if ( slewx+dampx .eq. 0)then
                    x = 0
             else
                    x = isign(1, slewx + dampx )
             endif
             if ( slewy+dampy .eq. 0)then
                    y = 0
             else
                    y = isign(1, slewy + dampy )
             endif
      endif
      if( iswtch(6) )then
             x = 0
      endif
      tx = x
      ty = y
      it = thrust(x,y)
c
c       USER SETS DACS TO ARRAY DAC VALUES
c
      is = setdac(0,6,dac)
c
c       DATA SENT TO STRIP CHART ( DELETE LINE IF NOT USING STRIP CHART)
c
c
c       USER SHOULD COMMENT OUT THIS CALL TO STRCHT IF HE/SHE DOES
c       NOT WISH TO USE THE MT8500 STRIP CHART RECORDER
c
      call strcht
c
c       DATA SAVED IN ARRAY FOR POST-EXPERIMENT PROCESSING
c
      call stofil(1,stodat)
```

```
ccccccccccccccccccccccccccccccccccc
c
c*****************************************************************
c
1     continue
ccccccccccc END OF REAL TIME LOOP cccc
333   continue
10000 continue
c
c       USER MUST SHUT DOWN ACTUATORS FOR SAFETY REASONS
c
c     SHUTDOWN ACTUATORS!
c
      is = oswtch( 1, 0 )
      it = thrust( 0, 0 )
      is = setdac(0,7,dac0)
      ic = cmgtor(0.,0.,0.)
c
      shtorr = 0.0
      shtorp = 0.0
      shtory = 0.0
c
c
c       END STRIP CHART SESSION
c       (OPTIONAL - DELETE  NEXT TWO LINES IF NOT USING STRIP CHART)
      im = mtstop()
      im = mtfeed()
      return
      end
```

SCOLE Makefile

```
.SUFFIXES:  .for .o
FSUBS  = cpmenu.for posexp.for preexp.for rtloop.for
FMAIN =  main.for
OSUBS  = cpmenu.o  posexp.o preexp.o rtloop.o
OMAIN = main.o
INCVC=/HOLD/SCOLE/varibs.comm
INCBC=/HOLD/SCOLE/blk.comm
LIBS = /lib/libscole.a /lib/libreal.a /lib/libmt8500.a /lib/libmatx.a
exec  = scole
SCOLE:  $(OMAIN) $(OSUBS) $(LIBS)
        ld main.o $(OSUBS) -lscole  -lreal -lmt8500 -lmatx -lunos o=$(exec)
        delete -s *bak *lst
        echo $(exec) re-created
.for.o  :
        f77 -lrujkp -z 240  $<
        /sys/pp $*.s>$*.m
        aa -argi2s $*.m o=$*.j
        reltocoff $*.j $*.o
        delete -s $*.s $*.m $*.j
        echo  $*.for compiled
$(OMAIN): $(FMAIN) $(INCVC) $(INCBC)
        f77 -lujkmp -z 240  $*.for
        /sys/pp $*.s>$*.m
        aa -argi2s $*.m o=$*.j
        reltocoff $*.j $*.o
        delete -s $*.s $*.m $*.j
        echo  $*.for compiled
 $(OSUBS): $(INCVC) $(INCBC)
        f77 -lrujkp -z 240  $*.for
        /sys/pp $*.s>$*.m
        aa -argi2s $*.m o=$*.j
        reltocoff $*.j $*.o
        delete -s $*.s $*.m $*.j
        echo  $*.for compiled
.IGNORE :
```

```
.SILENT :
.PRECIOUS: $(FMAIN) $(FSUBS) $(OSUBS) $(OMAIN)
```

SCOLE INCLUDE FILES

SECTION IV

The file /HOLD/SCOLE/varibs.comm is included in the majority of the SCOLE library subroutines and should be included in user supplied subroutines.

The file contains the following common blocks and declarations.

```
ccccccccccccccccccccc $include file VARIBS.COMM ccccccccccccccc
      implicit real (m)
      common /varibs/ shurgr, shurgp, shurgy, refrgr, refrgp, refrgy,
     +                shaccx, shaccy, shaccz, m1accx, m1accy, m2accx,
     +                m2accy, refacx, refacy, shtorr, shtorp, shtory,
     +                refwhx, refwhy, refwhz, m1whx,  m1why,  m2whx,
     +                m2why,  tx,     ty,     phi,    theta,  psi,
     +                p,      q,      r,      fx,     fy,
     +                ctheta, stheta, cpsi,   spsi,   cphi,   sphi,
     +                bls
c
      common /gains / dgain,  dblx,   dbly,
     +                sgainr, sgainp, sgainy, dbphi,  dbthe,  dbpsi,
     +                clx, cly, clz
c
      character*6 vdat,sdat,varib,cnum
      character*20 unit,sunit,vunit
      common /strdat/ sf(8),varib(42),unit(42),cnum(42),fsr(8),sfmt,
     + istosel(15),vdat(15),vunit(15),istrsel(8),sdat(8),sunit(8)
c
      common /consts/ pi
c
      common /time /  period, ttime,  runs
c
      character*10 spdunt
      common /mt8500/ strspd, spdunt
c
      common /chng_var/change(8)
      logical change
ccccccccccccccccccccc END OF $include file VARIBS.COMM ccccccccccc
```

A description of the elements of each common block follows.

COMMON VARIBS:

VARIABLE	DESCRIPTION
shurgr	shuttle rate gyro roll
shurgp	shuttle rate gyro pitch
shurgy	shuttle rate gyro yaw
refrgr	reflector rate gyro roll
refrgp	reflector rate gyro pitch
refrgy	reflector rate gyro yaw
shaccx	shuttle acceleration x direction
shaccy	shuttle acceleration y direction

shaccz	shuttle acceleration z direction
m1accx	mast station 1 -upper- acceleration x direction
m1accy	mast station 1 -upper- acceleration y direction
m2accx	mast station 2 -lower- acceleration x direction
m2accx	mast station 2 -lower- acceleration y direction
refacx	reflector acceleration x direction
refacy	reflector acceleration y direction
shtorr	shuttle torque roll
shtorp	shuttle torque pitch
shtory	shuttle torque yaw
refwhx	reflector wheel x direction
refwhy	reflector wheel y direction
refwhz	reflector wheel z direction
m1whx	mast station 1 -upper- wheel x direction
m1why	mast station 1 -upper- wheel y direction
m2whx	mast station 2 -lower- wheel x direction
m2why	mast station 2 -lower- wheel y direction
tx	thrust x direction
ty	thrust y direction
phi	bank angle
theta	pitch angle
psi	yaw angle
p	shuttle rotational velocity about x
q	shuttle rotational velocity about y
r	shuttle rotational velocity about z
fx	thruster gains x direction
fy	thruster gains y direction
ctheta	cosine theta
stheta	sine theta
cpsi	cosine psi
spsi	sine psi
cphi	cosine phi
sphi	sine phi
bls	thruster least square coefficient

COMMON GAINS:

VARIABLE	DESCRIPTION
dgain	thruster damping routine gain controller
dblx	thruster damping routine dead band level x velocity
dbly	thruster damping routine dead band level y velocity
sgainr	thruster slew routine gain roll
sgainp	thruster slew routine gain pitch
sgainy	thruster slew routine gain yaw
dbphi	thruster slew routine roll dead band level
dbthe	thruster slew routine pitch dead band level
dbpsi	thruster slew routine yaw dead band level
clx	collocated mast damper thruster slew routine gain roll
cly	collocated mast damper thruster slew routine gain pitch
clz	collocated mast damper thruster slew routine gain yaw

COMMON STRDAT:

VARIABLE	DESCRIPTION
sf, fsr	scale factors on eight dac channels sent to MT5800 strip chart recorder
varib	forty-two SCOLE supplied variables from which user chooses fifteen to save to data file and eight to record on the MT8500 strip chart recorder
unit	forty-two units corresponding to the forty-two SCOLE supplied variable choices above (varib)
cnum	character string array consisting of elements from '1' to '42'
istosel	array of indices of fifteen elements chosen from forty-two element varib array to be saved in a data file
vdat	variable name (character) of fifteen varib array elements chosen to be saved in a data file
vunit	units of fifteen varib array elements chosen to be saved in a data file
istrsel	array of indices of eight elements chosen from forty-two element varib array to be recorded on the MT8500 strip chart recorder
sdat	variable name (character) of eight varib array elements chosen to be recorded on the MT8500 strip chart recorder
vdat	units of eight varib array elements chosen to be recorded on the MT8500 strip chart recorder

COMMON TIME:

VARIABLE	DESCRIPTION
period	real time loop sampling period (sec)
ttime	real time loop duration (sec)
runs	number of passes in real time loop (ttime/period)

COMMON MT8500:

VARIABLE	DESCRIPTION
strspd	MT8500 strip chart recorder speed
spdunt	MT8500 strip chart recorder speed units

COMMON CHNG_VAR:

VARIABLE	DESCRIPTION
change	array to correlate variables sent to MT8500 strip and apropriate units

/HOLD/SCOLE/blk.comm

The file /HOLD/SCOLE/blk.comm is included in the SCOLE template main program located on /lib/SCOLE/TEMPLATE.

The file contains the following common blocks and declarations.

COMMON SCALE:

VARIABLE	DESCRIPTION
sfsrr	scale factor shuttle rate gyro roll
sfsrp	scale factor shuttle rate gyro pitch
sfsry	scale factor shuttle rate gyro yaw
sfrrr	scale factor reflector rate gyro roll
sfrrp	scale factor reflector rate gyro pitch
sfrry	scale factor reflector rate gyro yaw
sfsax	scale factor shuttle acceleration x
sfsay	scale factor shuttle acceleration y
sfsaz	scale factor shuttle acceleration z
sfm1x	scale factor mast station 1 -upper- acceleration x
sfm1y	scale factor mast station 1 -upper- acceleration y
sfm2x	scale factor mast station 2 -lower- acceleration x
sfm2y	scale factor mast station 2 -lower- acceleration y
sfrax	scale factor reflector acceleration x
sfray	scale factor reflector acceleration y
sfstr	scale factor shuttle torque roll
sfstp	scale factor shuttle torque pitch
sfsty	scale factor shuttle torque yaw

The values of the above listed variables correspond to the eighteen elements of array sfsens in blk.comm.

Scale factors have units of unity/engineering unit.
The units for the rate sensors are units/radian/second.
The units for the accelerometers are units/inch/second^2.

COMMON CMGSET: (CMG software in process of being updated)

VARIABLE	DESCRIPTION
rpm1	cmg rpm - default set to 3000
rpm3	cmg rpm - default set to 3000
sf1	scale factor cmg - default set to 10.
sf2	scale factor cmg - default set to 10.
sf3	scale factor cmg - default set to 10.
pd	servo gain prop. - default set to 60.
ti	servo gain integ - default set to .0005
td	servo gain deriv - default set to 9000.
b2	servo gain damp - default set to .9925

The values of array sfsens are set in data statement sfsens.

```
      data sfsens / 7.860, 11.2351, 4.7080, .1975, 1.15, 1.15,
     +              .00518, .00518, .00259,
     +              4*2.591e-3, 2*5.18e-4, 3*1. /
```

The forty-two character elements of array varib are defined in data statement varib.

```
      data varib /
     +       'shurgr', 'shurgp', 'shurgy', 'refrgr', 'refrgp', 'refrgy',
     +       'shaccx', 'shaccy', 'shaccz', 'm1accx', 'm1accy', 'm2accx',
     +       'm2accy', 'refacx', 'refacy', 'shtorr', 'shtorp', 'shtory',
     +       'refwhx', 'refwhy', 'refwhz', 'm1whx',  'm1why',  'm2whx',
     +       'm2why',  'tx',     'ty',     'phi',    'theta',  'psi',
     +       'p',      'q',      'r',      'fx',     'fy',
     +       'ctheta', 'stheta', 'cpsi',   'spsi',   'cphi',   'sphi',
     +       'bls'  /
```

The forty-two character units of array unit are defined in data statement unit.

```
      data unit /
     +       6*'rad/s',9*'in/s**2',10*'ft-lbs',2*'lbs',
     +       3*'rad',3*'rad/s',2*'N',6*'units','units'/
```

The forty-two character elements of array cnum are defined in data statement cnum.

```
         data cnum /'1','2','3','4','5','6','7','8','9','10',
     +   '11','12','13','14','15','16','17','18','19','20',
     +   '21','22','23','24','25','26','27','28','29','30',
     +   '31','32','33','34','35','36','37','38','39','40','41','42'/
```

SCOLE LIBRARY FILES

SECTION V

SUBROUTINE ATTIT

```
CALL:  call attit(reset)
       logical reset
```

Kinematic observer used to determine the roll and pitch attitude of the shuttle from the shuttle accelerometers and rate sensor. Reset is input as true for the first call to attit.

SUBROUTINE CHAN

```
CALL:   call chan
```

User prompted for number of channels to be printed on the MT8500 strip chart recorder (1-8 channels).

SUBROUTINE CMGMEN
(CMG software in process of being updated)

```
CALL:   call cmgmen
```

User presented a cmg set-up menu as follows:

```
CMG SETUP MENU
1. INPUT RPM
2. TAKE ADC TARE
3. TEST STEERING LAW CONTROL
4. TEST GIMBAL COMMAND CONTROL
5. CHANGE SERVO GAINS
6. CHANGE INPUT SCALE FACTOR
7. CHECK CMG SENSOR OUTPUTS
8. RETURN TO PREVIOUS MENU [DEFAULT]
```

```
Option 1:       sets rpm1, rpm3
Option 2:       sets array adct(12)
Option 3:       calls subroutine cmgtst
Option 4:       calls subroutine gimbal
Option 5:       sets pk, ti, td, b2
Option 6:       sets sf1, sf2, sf3
Option 7:       calls subroutine sensor
```

SUBROUTINE CMGTOR
(CMG software in process of being updated)

```
CALL:   img = cmgtor(xcmd,ycmd,zcmd)
        FORTRAN callable C routine
        integer*4 cmgtor, img
```

Three axis shuttle torque commands (x, y, z) sent to cmg controller.

SUBROUTINE CMGTST
(CMG software in process of being updated)

CALL: call cmgtst

Cmgs tested.

SUBROUTINE COLINT

CALL: call colint

User presented the following menu to accept the gain and dead-band levels for the thruster slew routine as shown in the documentation for the real time loop - rtloop.

```
-------------------------------------------
| ** COLOCATED MAST DAMPER MENU    **     |
-------------------------------------------
|  1) SET THE GAIN FOR ROLL                |, clx
|  2) SET THE GAIN FOR PITCH               |, cly
|  3) SET THE GAIN FOR YAW                 |, clz
|  4) RETURN TO MAIN MENU [DEFAULT]        |
-------------------------------------------
```

Values of clx, cly, clz passed in common block gains in include file /HOLD/SCOLE/varibs.comm.

SUBROUTINE CSCALE

CALL: call cscale

User prompted for scale factors by which to scale data being sent to the MT8500 strip chart recorder.

SUBROUTINE DAMINT

CALL: CALLED FROM SUBROUTINE CPMENU (call damint)

User presented following menu to accept gain and dead band levels for thruster damping routine.
Variables dgain, dblx, dbly, set and passed in common block gains in include file /HOLD/SCOLE/varibs.comm

```
-------------------------------------------
|      ** THRUSTER DAMPING MENU **        |
-------------------------------------------
| 1) SET THE GAIN FOR THE CONTROLLER      |, dgain
| 2) SET DEAD BAND LEVEL OF X-VELOCITY    |, dblx
| 3) SET DEAD BAND LEVEL OF Y-VELOCITY    |, dbly
| 4) RETURN TO MAIN MENU [DEFAULT]        |
-------------------------------------------'
```

SUBROUTINE DITIME

```
CALL:   call ditime(period, ttime)
        period: sample period - set in subroutine ditime
        ttime : run duration  - set in subroutine ditime
```

User inputs test duration time (ttime) and sample period (period).

SUBROUTINE ERTRAP

```
CALL:   call ertrap(ercode,ertype,mode)
        ercode - passed in value of ios in unsuccessful file open
                 (ie. open(..,iostat=ios,..))
        ertype - passed in as 1 to execute subroutine code
                 passed in as 0 to abort program
        mode   - passed in as 0 to continue program
                 passed in as 1 to abort program
```

Subroutine called after a file open as follows:

```
        call ertrap(ios,1,0).
```

SUBROUTINE GETAC

CALL: CALLED FROM SUBROUTINE LSQANG (call getac(ax,ay,az))

Subroutine gets data from accelerometers.

```
        ax set based on adc6
        ay set based on adc7
        az set based on adc8
```

SUBROUTINE GETTOR

```
CALL:   call gettor
```

Shuttle torque commands gotten from joystick.

SUBROUTINE GIMBAL
(CMG software in process of being updated)

```
CALL:   call gimbal(test)
        logical test
```

Four torque commands sent to gimbals.
If switch 2 is set on the switch box the subroutine code is not executed.

SUBROUTINE ISELEK(MAXP,MODE,IANS,NANS,LHNO,IDEF)

CALL: call iselek(maxp,mode,ians,nans,lhno,idef)
maxp, mode, ians, lhno and idef are integers
nans should be declared character*1 in the calling subroutine

The user chooses a menu selection line based on the mode chosen.
The following modes are available.

MODE	DESCRIPTION
1	1-maxp, H, M, Q or *
2	1-maxp, H, Q or *
3	1-maxp, H, M or *
4	1-maxp, H or *
5	1-maxp, Q or *
6	1-maxp, H, M, Q
7	1-maxp, H, Q
8	1-maxp, H, M
9	1-maxp, H
10	1-maxp, Q
11	1-maxp, M
12	1-maxp
13	Y or N
14	[Y] or N
15	[Y] or [N]
16	H, Y or N
17	H, [Y] or N
18	H, Y or [N]
19	[MORE] enter <cr> to continue ...
20	Enter <cr> to continue ...

Where,
maxp = maximum range of selection.
[xx] = default value enclosed in brackets.
H = display help message; returns 'M' in nans and 0 in ians.
M = returns'M' in nans.
Q = gives user option to abort program; if not confirmed an 'M' will be return in nans and 0 in ians.
* = return 'A' in nans and 0 in ians.
<cr> = return 'C' in nans and 0 in ians.

IANS passed out of iselek equal to zero if user inputs invalid selection number entered else equal to valid selection number input.

NANS passed out of iselek as follows:

C	modes 19, 20
Y	modes 14, 17; all yes choices with modes 13, 16, 18
N	modes 15, 18; all no choices with modes 13, 16, 17
D	choices 1 - maxp
M	choices H, M, Q
A	choice *

If nans equals M the user should direct program flow back to the menu in question. (ie., if(nans.eq.'M')go to xxx).

IDEF is input equal to MAXP.

LHNO is input with modes incorporating the H option for on-line help as follows:

0	user does not know what standard SCOLE subroutines are documented on-line; menu presented subroutines for which on-line help is available
1	user knows which subroutine for which help is desired; user inputs name in argument hlpfil
2	user desires help for a routine that is not one of the SCOLE standard routines; a help file for the subroutine is located on the SCOLE working directory; user inputs file name prefix in argument hlpfil (file extension is ".doc").

In the calling subroutine, LHNO should be set to 1 or 2 and hlpfil should be declared character*20 and be set to the name of the calling subroutine. If lhno is 2 then the user must have created a help file on the current working directory with prefix equal to the subroutine name and suffix '.doc'.

The common block info in subroutine iselek and lhelp passes parameters procse and hlpfil.

SUBROUTINE LBMENU

CALL: CALLED FROM SUBROUTINE STMENU (call lbmenu)
USERS SHOULD NOT NEED TO CALL THIS SUBROUTINE

User presented a menu as follows with options to choose the MT8500 strip chart recorder labels.

```
-------------------------------------------------
|              ** CHOOSE LABELS **              |
-------------------------------------------------
|  1) USE PROGRAM VARIABLE NAMES AS DESCRIPTORS.  |
|  2) USE FULL DESCRIPTION OF PROGRAM VARIABLES.  |
|  3) CHOOSE FILE TO USE FOR LABEL DATA.          |
|  4) CREATE CUSTOM LABELS FILE.                  |
|  5) RETURN TO PREVIOUS MENU.  (DEFAULT)         |
-------------------------------------------------
```

SUBROUTINE LHELP(HLPFIL,LHNO)

CALL: call lhelp(hlpfil,lhno)

This subroutine will allow the user to obtain on-line help for the standard SCOLE subroutines and/or user supplied subroutines.

LHNO is integer input as follows:

0	user does not know what standard SCOLE subroutines are documented on-line; menu presented of help available for subroutines
1	user knows which subroutine for which help is desired; user inputs name in argument hlpfil
2	user desires help for a routine that is not one of the SCOLE standard routines; a help file for the subroutine is located on the SCOLE working directory; user inputs file name prefix in argument hlpfil (file extension is ".doc").

Hlpfil must be declared character*20 in the calling subroutine and set to the name of the subroutine for which help is desired if lhno is set to 1 and set to the prefix of the name of the help file if lhno is set to 2.

The common block info as found in subroutines iselek and lhelp passes parameters procse and hlpfil.

```
common /info/ procse, hlpfil
```

This subroutine is called from subroutine iselek when modes including the H menu option are chosen.

Users can include a call to lhelp from a menu if iselek is not used.

SUBROUTINE LSQANG

CALL: call lsqang(ctheta,stheta,cpsi,spsi,cphi,sphi)

Subroutine determines least square angle estimation.
Arguments set in subroutine lsqang as follows

```
ctheta - cosine theta
stheta - sine theta
cpsi   - cosine psi
spsi   - sine psi
cphi   - cosine phi
sphi   - sine phi.
```

SUROUTINE READF

CALL: call readf

SCOLE formatted direct access files gains.dat and select.set, copied onto the user's account from /lib/SCOLE/TEMPLATE, are read.

The section of the code that reads the files follows.

```
c
c  Read from "gains.dat"
c
        open(5,err=9996,iostat=ios,form='formatted',access='direct',
     +  recl=80,file='gains.dat',status='old')
        if(dbug)print*,'Subroutine readf -file gains.dat opened'
        read(5,25,rec=2)dgain,dblx,dbly
25      format(3(e12.5,1x))
        read(5,26,rec=1)ttime,period
26      format(2(e12.5,1x))
        read(5,27,rec=3)sgainr,sgainp,sgainy,dbphi,dbthe,dbpsi
27      format(6(e12.5,1x))
        read(5,39,rec=4)clx,cly,clz
39      format(3(e12.5,1x))
        close(5)
c
c  Read from "select.set"
c
        open(6,err=9996,iostat=ios,form='formatted',access='direct'
     +  recl=112,file='select.set',status='old')
        read(6,28,rec=1)(fsr(i),i=1,4)
        read(6,28,rec=2)(fsr(i),i=5,8)
28      format(4(e12.5,1x))
        read(6,2000,rec=3)strspd,spdunt
2000    format(f10.2,1x,a10)
        read(6,29,rec=4)(istosel(i),i=1,15)
29      format(15(i3,1x))
        read(6,30,rec=5)(vdat(i),i=1,15)
301     format(3(a20,1x))
30      format(15(a6,1x))
        read(6,301,rec=6)(vunit(i),i=1,3)
        read(6,301,rec=7)(vunit(i),i=4,6)
        read(6,301,rec=8)(vunit(i),i=7,9)
        read(6,301,rec=9)(vunit(i),i=10,12)
        read(6,301,rec=10)(vunit(i),i=13,15)
        read(6,31,rec=11)(istrsel(i),i=1,8)
        read(6,32,rec=12)(sdat(i),i=1,8)
        read(6,301,rec=13)(sunit(i),i=1,3)
        read(6,301,rec=14)(sunit(i),i=4,6)
        read(6,304,rec=15)(sunit(i),i=7,8)
304     format(2(a20,1x))
        close(6)
c
31      format(8(i3,1x))
32      format(8(a6,1x))
321     format(8(a20,1x))
```

To alter values in these files, files should be opened as shown and records read and written with the indicated formats.

VARIABLE	USE	SET IN SUBROUTINE	COMMON BLOCK IN /HOLD/SCOLE/VARIBS.COMM
dgain	controller gain	damint.for	gains
dblx	dead band level x velocity	damint.for	gains
dbly	dead band level y velocity	damint.for	gains
ttime	run duration	ditime.for	parameter
period	sample period	ditime.for	parameter
sgainr	gain roll thruster slew	slwint.for	gains
sgainp	gain pitch thruster slew	slwint.for	gains
sgainy	gain yaw thruster slew	slwint.for	gains
dbphi	roll dead band level thruster slew	slwint.for	gains
dbthe	pitch dead band level thruster slew	slwint.for	gains
dbpsi	yaw dead band level thruster slew	slwint.for	gains
clx	gain roll colocated mast damper	colint.for	gains
cly	gain pitch colocated mast damper	colint.for	gains
clz	gain yaw colocated mast damper	colint.for	gains
fsr	MT8500 scale factors	cscale.for	strdat
strspd	MT8500 speed	spdset.for	strdat
spdunt	MT8500 speed units	spdset.for	strdat
istosel	array of indices of 15 variables of 42 SCOLE variables to be saved to a data file	stoset.for	strdat
vdat	array of 15 variable names selected to be saved to data file	stoset.for	strdat
vunit	array of 15 variable units selected to be saved to data file	stoset.for	strdat
istrsel	array of indices of 8 variables of 42 SCOLE variables to be sent to the MT8500	strset.for	strdat
sdat	array of 8 variable names selected to be sent to the MT8500	strset.for	strdat
sunit	array of 8 variable units selected to be sent to the MT8500	strset.for	strdat

SUBROUTINE SBIAS

CALL: call sbias(tau)
tau - sample period for 20 second averaging

Subroutine determines sensor bias by averaging over 20 seconds. Bias values passed in array bias in common store and written to file calibration.dat. Bias(1) bias for adc(1), bias(2) bias for adc(2), etc.

SUBROUTINE SLWINT

CALL: CALLED FROM SUBROUTINE CPMENU (call slwint)

User presented the following menu to set the gain and dead-band levels for the thruster slew routine.

```
-------------------------------------------
|      ** THRUSTER SLEW MENU    **         |
-------------------------------------------
|  1) SET THE GAIN FOR ROLL                |, sgainr
|  2) SET THE GAIN FOR PITCH               |, sgainp
|  3) SET THE GAIN FOR YAW                 |, sgainy
|  4) SET ROLL DEAD BAND LEVEL             |, dbphi
|  5) SET PITCH DEAD BAND LEVEL            |, dbthe
|  6) SET YAW  DEAD BAND LEVEL             |, dbpsi
|  7) RETURN TO MAIN MENU [DEFAULT]        |
-------------------------------------------
```

Variables passed in common gains in include file /HOLD/SCOLE/varibs.comm

SUBROUTINE SPDSET

CALL: call spdset

User given option to change MT8500 strip chart recorder speed.

SUBROUTINE STOFIL

CALL: call stofil(n,stodat)
n - pass number in real time loop
stodat - array of data points to be stored; dimensioned 15 by runs where 15 is the number of variables to be saved and runs is the number of passes in the real time loop calculated to be duration/period.

Subroutine called in real time loop for data point storage.

SUBROUTINE STOSET

CALL: call stoset

User selects program variables (15) to be output to data file from the 42 SCOLE variables saved in common block varibs in include /HOLD/SCOLE/varibs.comm

```
      common /varibs/ shurgr, shurgp, shurgy, refrgr, refrgp, refrgy,
     +                shaccx, shaccy, shaccz, m1accx, m1accy, m2accx,
     +                m2accy, refacx, refacy, shtorr, shtorp, shtory,
     +                refwhx, refwhy, refwhz, m1whx,  m1why,  m2whx,
     +                m2why,  tx,     ty,     phi,    theta,  psi,
     +                p,      q,      r,      fx,     fy,
     +                ctheta, stheta, cpsi,   spsi,   cphi,   sphi,
     +                bls
```

SUBROUTINE STRCHT

CALL: call strcht

The values of the eight SCOLE variables selected in subroutine strset sent to the MT8500 strip chart recorder. These values are scaled by the scale factors set in subroutine cscale before being sent to the strip chart recorder. Subroutine called in a real time loop.

SUBROUTINE STRSET

CALL: call strset

User selects program variables (8) to be output to the MT8500 strip chart recorder from the 42 SCOLE variables saved in common block varibs in include /HOLD/SCOLE/varibs.comm.

```
      common /varibs/ shurgr, shurgp, shurgy, refrgr, refrgp, refrgy,
     +                shaccx, shaccy, shaccz, m1accx, m1accy, m2accx,
     +                m2accy, refacx, refacy, shtorr, shtorp, shtory,
     +                refwhx, refwhy, refwhz, m1whx,  m1why,  m2whx,
     +                m2why,  tx,     ty,     phi,    theta,  psi,
     +                p,      q,      r,      fx,     fy,
     +                ctheta, stheta, cpsi,   spsi,   cphi,   sphi,
     +                bls
```

SUBROUTINE TSCOLE

CALL: call tscole(nm,nd,ny)

The current date is output in integer form: month - nm, day - nd, and year - ny.

SUBROUTINE UWRITE

CALL: call uwrite(nsamp,tau,stodat)
nsamp - number of samples, precalculated to be duration / period
tau - sample period
stodat - array of data points stored

User presented the following menu of options to save run data to a file after experiment complete.

```
****************************************
            DATA OUTPUT MENU
****************************************
    1.  CRDS DATA FILE FOR ANALYSIS
    2.  MATRIX-X DATA FILE
    3.  MATLAB DATA FILE
```

MT8500 STRIP CHART RECORDER LIBRARY ROUTINES CALLED BY USERS

SECTION VI

SUBROUTINE MTCHAN

NAME: mtchan(nchan)

DESCRIPTION: This subroutine sets the number of channels to be printed from the MT8500.

CALL:

FORTRAN:

```
integer*4 mtchan, kepone
integer nchan
kepone = mtchan(6)
```

C :

```
integer nchan;
mtchan(nchan);
```

INPUT ARGUMENTS:

nchan integer 1-8 = number of channels to be printed.

OUTPUT ARGUMENTS:

return: 1 = successful call
0 = unsuccessful call

REQUIRES:

Nothing

LIBRARY: /lib/libmt8500.a

SUBROUTINE MTDATE

NAME: mtdate()

DESCRIPTION: This subroutine sets the time and date on the MT8500 chart recorder.
(This subroutine is not normally called by applications - date is automatically set by other routines.)

CALL:

FORTRAN:

```
integer*4 mtdate, im
im = mtdate()
```

C :

```
mtdate();
```

INPUT ARGUMENTS:

None

OUTPUT ARGUMENTS:

return: 1 = successful call
0 = unsuccessful call

REQUIRES:
Nothing

USES:
cvtime

LIBRARY:
/lib/libmt8500.a

SUBROUTINE MTFEED

NAME: mtfeed()

DESCRIPTION: This subroutine advances the paper in the MT8500 chart recorder to the next top-of-form mark (form feed).

CALL:

FORTRAN:
```
integer*4 mtfeed, j
j=mtfeed()
```
C :
```
mtfeed();
```

INPUT ARGUMENTS:
None

OUTPUT ARGUMENTS:
return : 1 (no check for a successful call)

REQUIRES:
Nothing

LIBRARY: /lib/libmt8500.a

SUBROUTINE MTFILE

NAME: mtfile('fname')

DESCRIPTION: This subroutine sets the channel labels on the MT8500 chart recorder according to the filename supplied in fname.

CALL:

FORTRAN:
```
integer*4 mtfile, igor
character*(*) fname
igor = mtfile("llabel.doc")
```
C :
```
char[] fname;
mtfile (fname);
```

INPUT ARGUMENTS:
fname string pointing to sequential data file of labels

OUTPUT ARGUMENTS:

return: 1 = successful call
0 = unsuccessful call

REQUIRES:

All labels must end in a <CR>, be written sequentially, and be < 100 characters in length.

LIBRARY: /lib/libmt8500.a

SUBROUTINE MTINIT

NAME: mtinit(nchan)

DESCRIPTION: This subroutine initializes the MT8500 chart recorder for use from its "Ready" state. Communication lines are opened between the chart recorder and the CRDS on /dev/tty4. The MT8500 is initialized to the "Realtime" state and passed the number of channels to be printed. Finally, date and time are passed to the recorder via the mtdate subroutine.

This subroutine is NOT called by user programs!

CALL:

FORTRAN:

```
integer*4 mtinit, i
integer nchan
i=mtinit(nchan)
```

C :

```
int nchan;
mtinit(nchan);
```

INPUT ARGUMENTS:

nchan {integer 1-8} = number of channels to be printed.

OUTPUT ARGUMENTS:

return: 1 = successful call
0 = unsuccessful call

REQUIRES:

Nothing

LIBRARY:

/lib/libmt8500.a

SUBROUTINE MTRAY

NAME: mtray(a_name)

DESCRIPTION: This subroutine sets the eight channel labels on the sMT8500 chart recorder according to the array of labels in a_name.

CALL:

FORTRAN:

```
integer*4 mtray, lupe
dimension a_name(8)
character a_name*100
lupe = mtray(a_name)
```

INPUT ARGUMENTS:

a_name = array of 0 to 8 label strings

OUTPUT ARGUMENTS:

return: 1 = successful call
0 = unsuccessful call

REQUIRES:

Each label string must end with a "0" and be < 100 characters in length.

LIBRARY: /lib/libmt8500.a

SUBROUTINE MTRLS

NAME: mtrls()

DESCRIPTION: This subroutine releases the MT8500 chart recorder from the real-time mode and leaves it in a ready state.

CALL:

FORTRAN:

```
integer*4 mtrls, l
l = mtrls()
```

C :

```
l = mtrls();
```

INPUT ARGUMENTS:

None

OUTPUT ARGUMENTS:

return value: 1 = successful call
0 = unsuccessful call

REQUIRES:

Nothing

LIBRARY: /lib/libmt8500.a

SUBROUTINE MTSTOP

NAME: mtstop()

DESCRIPTION: Stop waveform printing from the MT8500 chart recorder.

CALL:

FORTRAN:

```
integer*4 mtstop, jove
jove = mtstop()
```

C :

```
i = mtstop();
```

INPUT ARGUMENTS:

None

OUTPUT ARGUMENTS:

return value: 1 = successful call
0 = unsuccessful call

REQUIRES:

Nothing

LIBRARY: /lib/libmt8500.a

MATH LIBRARY ROUTINES

SECTION VII

SUBROUTINE ADD

NAME: ADD()

DESCRIPTION: This subroutine performs matrix addition of input matrices [A] and [B]; sum matrix output in [C].

CALL: call add(a,na,b,nb,c,nc,m,n)

INPUT ARGUMENTS:

A,B	matrices packed by columns in one dimensional arrays
NA,NB	dimension vectors of [A] and [B] respectively NA(1)=NB(1)=number of rows of [A] and [B] NA(2)=NB(2)=number of columns of [A] and [B]
M	number of rows of [A] and [B]
N	number of columns of [A] and [B]

OUTPUT ARGUMENTS:

C	[C]=[A]+[B]; sum matrix packed by columns in one dimensional array
NC	dimension vector of [C] NC(1)=NA(1)=NB(1)=M NC(2)=NA(2)=NB(2)=N

REQUIRES:

NA(1)=NB(1)=NC(1)=M
NA(2)=NB(2)=NC(2)=N

LIBRARY:

/lib/libmatx.a

SUBROUTINE DISCRT

NAME: DISCRT()

DESCRIPTION: This subroutine computes exp(a*tau) and its integral using the series expansion definition for up to nterms.

CALL: call discrt(a,na,phi,nphi,gam,ngam,n,tau,nterms)

INPUT ARGUMENTS:

A	matrix packed by columns in one dimensional array
NA	dimension vector of [A] NA(1)=number of rows of [A] NA(2)=number of columns of [A]
N	number of rows and columns of [A]
NTERMS	number of terms to expand

OUTPUT ARGUMENTS:

PHI	matrix packed by columns in one dimensional array at [PHI]=e

GAM — matrix packed by columns in one dimensional array as
[GAM]=integral e ds from 0 to tau

NPHI, NGAM — dimension vectors [PHI] and [GAM] respectively

NPHI(1)=NGAM(1)=NA(1)=N
NPHI(2)=NGAM(2)=NA(2)=N

REQUIRES:

NPHI(1)=NGAM(1)=NA(1)=N
NPHI(2)=NGAM(2)=NA(2)=N

LIBRARY: /lib/libmatx.a

FUNCTION DOT

NAME: DOT()

DESCRIPTION: This function calculates the dot product of two vectors, a and b.

CALL: caldot = dot (a,b,n)

INPUT ARGUMENTS:

A,B — column vectors
N — dimension of vector [A]

OUTPUT ARGUMENTS:

DOT — dot product of input vectors

REQUIRES:

Nothing

LIBRARY: /lib/libmatx.a

SUBROUTINE EQUATE

NAME: EQUATE()

DESCRIPTION: This subroutine equates two matrices; [B]=[A].

CALL: call equate(a,na,b,nb,m,n)

INPUT ARGUMENTS:

A — matrix packed by columns in one dimensional array
NA — dimension vector of [A]
NA(1)=number of rows of [A]
NA(2)=number of columns of [A]
M — number of rows of [A]
N — number of columns of [A]

OUTPUT ARGUMENTS:

B — [B]=[A]; equivalent matrix packed by columns in one

```
                dimensional array
        NB      dimension vector of [B]
                NB(1)=M
                NB(2)=N

REQUIRES:
        NA(1)=NB(1)=M
        NA(2)=NB(2)=N

LIBRARY:    /lib/libmatx.a
```

FUNCTION FMAG

```
NAME:  FMAG()

DESCRIPTION: This function finds the magnitude of a vector A.

CALL:       calmag = fmag(a,n)

INPUT ARUGMENTS:
        A       column vector
        N       dimension of vector A

OUTPUT ARUGMENTS:
        fmag    magnitude of vector A

REQUIRES:
        Nothing

LIBRARY:    /lib/libmatx.a
```

SUBROUTINE IDENT

```
NAME:  IDENT()

DESCRIPTION: This subroutine creates an identity matrix; [A]=[I].

CALL:       call ident(a,na,n)

INPUT ARGUMENTS:
        A       matrix packed by columns in one dimensional array
        NA      dimension vector of [A]
                NA(1)=number of rows of [A]
                NA(2)=number of columns of [A]
        N       number of rows and columns of [A]

OUTPUT ARGUMENTS:
        A       [A]=[I]; identity matrix packed by columns in one
                dimensional array

REQUIRES:
        NA(1)=NA(2)=N
        call to NULL
```

LIBRARY: /lib/libmatx.a

SUBROUTINE MULT

NAME: MULT()

DESCRIPTION: This subroutine performs matrix multiplication of input matrices [A] and [B]; product matrix output in [C].

CALL: call mult(a,na,b,nb,c,nc,l,m,n)

INPUT ARGUMENTS:

A,B	matrices packed by columns in one dimensional arrays
NA,NB	dimension vectors of [A] and [B] respectively NA(1)=number of rows of [A] NA(2)=number of columns of [A] NB(1)=number of rows of [B] NB(2)=number of columns of [B]
L	number of rows of [A] and [C]
M	number of columns of [A] and rows of [B]
N	number of columns of [B] and [C]

OUTPUT ARGUMENTS:

C	[C]=[A]*[B]; product matrix packed by columns in one dimensional array
NC	dimension vector of [C] NC(1)=NA(1)=L NC(2)=NB(2)=N

REQUIRES:

NA(1)=NC(1)=L
NA(2)=NB(1)=M
NB(2)=NC(2)=N

LIBRARY: /lib/libmatx.a

SUBROUTINE MULT3

NAME: MULT3()

DESCRIPTION: This subroutine performs matrix multiplication of input matrices [A],[B], and [C]; product matrix output in [D].

CALL: call mult3(a,b,c,d,ar,ac,bc,cc)

INPUT ARGUMENTS:

A,B,C	matrices packed by columns in one dimensional arrays
AR	number of rows of [A] and [D]
AC	number of columns of [A] and rows of [B]
BC	number of columns of [B] and rows of [C]
CC	number of columns of [C] and [D]

OUTPUT ARGUMENTS:

D	[D]=[A]*[B]*[C]; product matrix packed by columns in one dimensional array

REQUIRES:

Nothing

LIBRARY: /lib/libmatx.a

SUBROUTINE NEG

NAME: NEG()

DESCRIPTION: This subroutine renders the negative of a matrix ; [B]=-1.0*[A].

CALL: call neg(a,na,b,nb,m,n)

INPUT ARGUMENTS:

A	matrix packed by columns in one dimensional array
NA	dimension vector of [A] NA(1)=number of rows of [A] NA(2)=number of columns of [A]
M	number of rows of [A]
N	number of columns of [A]

OUTPUT ARGUMENTS:

B	[B]=-1.0*[A]; scaled matrix packed by columns in one dimensional array
NB	dimension vector of [B] NB(1)=M NB(2)=N

REQUIRES:

NA(1)=NB(1)=M
NA(2)=NB(2)=N

LIBRARY: /lib/libmatx.a

SUBROUTINE NULL

NAME: NULL()

DESCRIPTION: This subroutine nulls a matrix; [A]=[0].

CALL: call null(a,na,n,m)

INPUT ARGUMENTS:

A	matrix packed by columns in one dimensional array
NA	dimension vector of [A] NA(1)=number of rows of [A] NA(2)=number of columns of [A]
N	number of rows of [A]
M	number of columns of [A]

OUTPUT ARGUMENTS:

A	[A]=[0]; null matrix packed by columns in one dimensional array

REQUIRES:

NA(1)=N
NA(2)=M

LIBRARY: /lib/libmatx.a

SUBROUTINE PRNT

NAME: PRNT()

DESCRIPTION: This subroutine prints a matrix for output to a printer; 132 columns are used; matrix is printed 1 row by 9 columns.

CALL: call prnt(a,na,nrow,mcol,name)

INPUT ARGUMENTS:

A	matrix packed by columns in one dimensional array
NA	dimension vector of [A] NA(1)=number of rows of [A] NA(2)=number of columns of [A]
NROW	number of rows of [A]
MCOL	number of columns of [A]
NAME	descriptor of matrix to be printed; may be passed as a string or as a variable declared as a character

OUTPUT ARGUMENTS:
NONE

REQUIRES:

NA(1)=NROW
NA(2)=MCOL

LIBRARY: /lib/libmatx.a

SUBROUTINE PRNT1

NAME: PRNT1()

DESCRIPTION: This subroutine prints a matrix for output to a terminal.

CALL: call prnt1(a,na,n,m,name)

INPUT ARGUMENTS:

A	matrix packed by columns in one dimensional array
NA	dimension vector of [A] NA(1)=number of rows of [A] NA(2)=number of columns of [A]
N	number of rows of [A]

M — number of columns of [A]
NAME — descriptor of matrix to be printed; may be passed as a string or as a variable declared as a character

OUTPUT ARGUMENTS:
NONE

REQUIRES:
NA(1)=N
NA(2)=M

LIBRARY: /lib/libmatx.a

SUBROUTINE PRNTF

NAME: PRNTF()

DESCRIPTION: This subroutine prints a matrix to a file.

CALL: call prntf(a,na,nrow,mcol,name)

INPUT ARGUMENTS:
A — matrix packed by columns in one dimensional array
NA — dimension vector of [A]
NA(1)=number of rows of [A]
NA(2)=number of columns of [A]
NROW — number of rows of [A]
MCOL — number of columns of [A]
NAME — descriptor of matrix to be printed; may be passed as a string or as a variable declared as a character

OUTPUT ARGUMENTS:
NONE

REQUIRES:
NA(1)=NROW
NA(2)=MCOL
Output file must be opened and closed in calling program; unit number 6 is used.

LIBRARY: /lib/libmatx.a

SUBROUTINE SCALE

NAME: SCALE()

DESCRIPTION: This subroutine scales a matrix by a scalar; [B]=c[A].

CALL: call scale(a,na,b,nb,m,n,c)

INPUT ARGUMENTS:
A — matrix packed by columns in one dimensional array
NA — dimension vector of [A]

NA(1)=number of rows of [A]
NA(2)=number of columns of [A]

M — number of rows of [A]

N — number of columns of [A]

C — scale factor passed by value or as variable

OUTPUT ARGUMENTS:

B — [B]=c[A]; scaled matrix packed by columns in one dimensional array

NB — dimension vector of [B]
NB(1)=M
NB(2)=N

REQUIRES:

NA(1)=NB(1)=M
NA(2)=NB(2)=N

LIBRARY: /lib/libmatx.a

SUBROUTINE SUBTRACT

NAME: SUBTRACT()

DESCRIPTION: This subroutine performs matrix subtraction of input matrices [A] and [B]; difference matrix output in [C].

CALL: call subtract(a,na,b,nb,c,nc,m,n)

INPUT ARGUMENTS:

A,B — matrices packed by columns in one dimensional arrays

NA,NB — dimension vectors of [A] and [B] respectively
NA(1)=NB(1)=number of rows of [A] and [B]
NA(2)=NB(2)=number of columns of [A] and [B]

M — number of rows of [A] and [B]

N — number of columns of [A] and [B]

OUTPUT ARGUMENTS:

C — [C]=[A]-[B]; difference matrix packed by columns in one dimensional array

NC — dimension vector of [C]
NC(1)=NA(1)=NB(1)=M
NC(2)=NA(2)=NB(2)=N

REQUIRES:

NA(1)=NB(1)=NC(1)=M
NA(2)=NB(2)=NC(2)=N

LIBRARY: /lib/libmatx.a

FUNCTION TRACE

NAME: TRACE()

DESCRIPTION: This function finds the trace of a matrix.

CALL: t = trace(a,na,n)

INPUT ARGUMENTS:

A	matrix packed by columns in one dimensional array
NA	dimension vector of [A] NA(1)=number of rows of [A] NA(2)=number of columns of [A]
N	number of rows and columns of [A]

OUTPUT ARGUMENTS:

TRACE	trace of the input matrix

REQUIRES:

NA(1)=NA(2)=N

LIBRARY: /lib/libmatx.a

SUBROUTINE TRANP

NAME: TRANP()

DESCRIPTION: This subroutine takes the transpose of a matrix; $[B]=[A]^T$.

CALL: call tranp(a,na,b,nb,m,n)

INPUT ARGUMENTS:

A	matrix packed by columns in one dimensional array
NA	dimension vector of [A] NA(1)=number of rows of [A] NA(2)=number of columns of [A]
M	number of rows of [A]
N	number of columns of [A]

OUTPUT ARGUMENTS:

B	$[B]=[A]^T$; transposed matrix packed by columns in one dimensional array
NB	dimension vector of [B] NB(1)=N NB(2)=M

REQUIRES:

NA(1)=NB(2)=M
NA(2)=NB(1)=N
call to EQUATE

LIBRARY: /lib/libmatx.a

SECTION VIII

APPENDIX

APPENDIX A

SCOLE INSTRUMENT CABINET PATCHING

CABINET 1		CABINET 2		CABINET 3
CMG 3:				
T-Motor 26-not connected				
T-Motor 27-not connected				
Tach 28	->	neff 25	->	DT-ADC 11
Tach 29	->	neff 26	->	DT-ADC 12
Res1 30	->	neff 27	->	DT-ADC 5
Res1 31	->	neff 28	->	DT-ADC 6
Res3 32	->	neff 29	->	DT-ADC 7
Res3 33	->	neff 30	->	DT-ADC 8
		neff 31-not connected		
		neff 32-not connected		
CMG 1:				
T-Motor 34-not connected				
T-Motor 35-not connected				
Tach 36	->	neff 18	->	DT-ADC 9
Tach 37	->	neff 19	->	DT-ADC 10
Res1 38	->	neff 20	->	DT-ADC 1
Res1 39	->	neff 21	->	DT-ADC 2
Res3 40	->	neff 22	->	DT-ADC 3
Res3 41	->	neff 23	->	DT-ADC 4
		neff 24-not connected		
ACCELEROMETERS:				
1 (shaccx)	->	neff 9	->	ADC 7
2 (shaccy)	->	neff 10	->	ADC 8
3 (shaccz)	->	neff 11	->	ADC 9
4 (m1accx)	->	neff 12	->	ADC 10
5 (m1accy)	->	neff 13	->	ADC 11
6 (m2accx)	->	neff 14	->	ADC 12
7 (m2accy)	->	neff 15	->	ADC 13
8 (refacx)	->	neff 16	->	ADC 14
9 (refacy)	->	neff 17	->	ADC 15
RATE TRANSDUCERS:				
Shuttle 10 (shurgr)	->	neff 1	->	ADC 1
Shuttle 11 (shurgp)	->	neff 2	->	ADC 2
Shuttle 12 (shurgy)	->	neff 3	->	ADC 3
Mast 13 (refrgr)	->	neff 4	->	ADC 4
Mast 14 (refrgp)	->	neff 5	->	ADC 5
Mast 15 (refrgy)	->	neff 6	->	ADC 6
PROOF INERTIA				
16 (m1whx)	->		->	DAC 4
17 (m1why)	->		->	DAC 5
18 (m2whx)	->		->	DAC 6
19 (m2why)	->		->	DAC 7
20 (refwhx)	->		->	DAC 1
21 (refwhy)	->		->	DAC 2
22 (refwhz)	->		->	DAC 3
23-not connected				
24-not connected				

25-not connected

BUFFER			
1	->	->	MT-8500 1
2	->	->	MT-8500 2
3	->	->	MT-8500 3
4	->	->	MT-8500 4
5	->	->	MT-8500 5
6	->	->	MT-8500 6
7	->	->	MT-8500 7
8	->	->	MT-8500 8

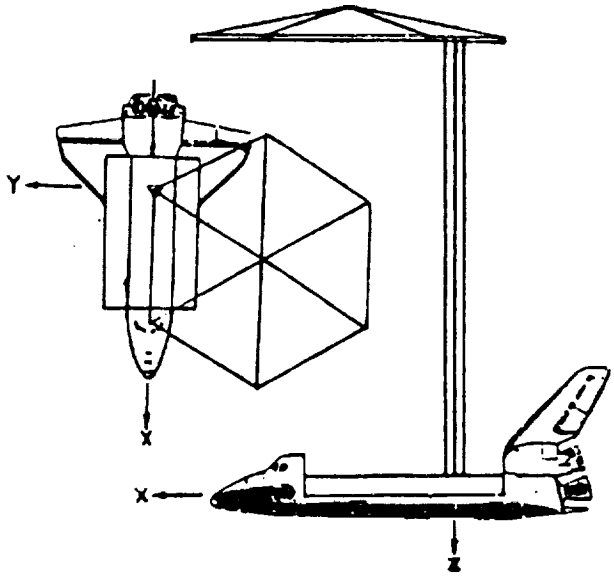
A MATHEMATICAL PROBLEM AND A SPACECRAFT CONTROL LABORATORY EXPERIMENT (SCOLE) USED TO EVALUATE CONTROL LAWS FOR FLEXIBLE SPACECRAFT... NASA/IEEE DESIGN CHALLENGE

by

Lawrence W. Taylor, Jr.
Spacecraft Control Branch
NASA Langley Research Center
Hampton, VA 23665

and

A. V. Balakrishnan
Chairman, IEEE Subcommittee on Large Space Structures, COLSS
System Sciences Department
University of California at Los Angeles
Los Angeles, CA



NASA IEEE

A MATHEMATICAL PROBLEM AND A SPACECRAFT CONTROL LABORATORY EXPERIMENT (SCOLE) USED TO EVALUATE CONTROL LAWS FOR FLEXIBLE SPACECRAFT... NASA/IEEE DESIGN CHALLENGE

by

Lawrence W. Taylor, Jr.
Spacecraft Control Branch
NASA Langley Research Center
Hampton, VA 23665

and

A. V. Balakrishnan
Chairman, IEEE Subcommittee on Large Space Structures, COLSS
System Sciences Department
University of California at Los Angeles
Los Angeles, CA

SUMMARY

The problem of controlling large, flexible space systems has been the subject of considerable research. Many approaches to control system synthesis have been evaluated using computer simulation. In several cases, ground experiments have also been used to validate system performance under more realistic conditions. There remains a need, however, to test additional control laws for flexible spacecraft and to directly compare competing design techniques. In this paper an NASA program is discussed which has been initiated to make direct comparisons of control laws for, first, a mathematical problem, then an experimental test article is being assembled under the cognizance of the Spacecraft Control Branch at the NASA Langley Research Center with the advice and counsel of the IEEE Subcommittee on Large Space Structures. The physical apparatus will consist of a softly supported dynamic model of an antenna attached to the Shuttle by a flexible beam. The control objective will include the task of directing the line-of-sight of the Shuttle/antenna configuration toward a fixed

target, under conditions of noisy data, limited control authority and random disturbances. The open competition started in the early part of 1984. Interested researchers are provided information intended to facilitate the analysis and control synthesis tasks. A workshop is planned for early December at the NASA Langley Research Center to discuss and compare results.

INTRODUCTION

Many future spacecraft will be large and consequently quite flexible. As the size of antennae is increased, the frequencies of the first flexible modes will decrease and overlap the pointing system bandwidth. It will no longer be possible to use low gain systems with simple notch filters to provide the required control performance. Multiple sensors and actuators, and sophisticated control laws will be necessary to ensure stability, reliability and the pointing accuracy required for large, flexible spacecraft.

Control of such spacecraft has been studied with regard given to modeling, order reduction, fault management, stability and dynamic system performance. Numerous example applications have been used to demonstrate specific approaches to pertinent control problems. Both computer simulations and laboratory experiment results have been offered as evidence of the validity of the approaches to control large, flexible spacecraft. Concerns remain, however, because of the chronic difficulties in controlling these lightly damped large-scale systems. Because of these concerns and because of the desire to offer a means of comparing technical approaches directly, an NASA/IEEE Design Challenge is being offered. An

experimental test article is being assembled under the cognizance of the Spacecraft Control Branch at the NASA Langley Research Center with the advice and counsel of the IEEE (COLSS) Subcommittee on Large Space Structures. This Spacecraft Control Laboratory Experiment (SCOLE) will serve as the focus of a design challenge for the purpose of comparing directly different approaches to control synthesis, modeling, order reduction, state estimation and system identification.

The configuration of the SCOLE will represent a large antenna attached to the Space Shuttle orbiter by a flexible beam. This configuration was chosen because of its similarity to proposed space flight experiments and proposed space-based antenna systems. This paper will discuss the "Design Challenge" in terms of both a mathematical problem and a physical experimental apparatus. The SCOLE program is not part of any flight program.

SYMBOLS

a	acceleration vector ft/sec^2
A	beam cross section area
c	observation matrix
d	noise contaminating direction cosine matrix measurements
e	line-of-sight error
E	modulus of elasticity
f	concentrated force expressions
F_4	force vector
g	concentrated moment expressions
GI	torsional rigidity
I	moment of inertia matrix for entire Shuttle/antenna configuration

I_1	moment of inertia matrix, Shuttle body
I_4	moment of inertia matrix, reflector body
I_ϕ	beam cross section moment of inertia, roll bending
I_Θ	beam cross section moment of inertia, pitch bending
I_Ψ	beam polar moment of inertia, yaw torsion
L	length of the reflector mast, beam
M_1	control moment applied to the Shuttle body
M_4	control moment applied to the reflector body
M_D	disturbance moment applied to the Shuttle body
m	mass of entire Shuttle/antenna configuration
m_1	mass of Shuttle body
m_4	mass of reflector body
P	mass density of beam
s	beam position variable
T_1	direction cosine matrix, Shuttle body $()_{earth} = T_1()_{Shuttle\ body}$
T_4	direction cosine matrix, reflector body $()_{earth} = T_4()_{reflector\ body}$
v_1	inertial velocity, Shuttle body
v_4	inertial velocity, reflector body
u_ϕ	lateral deflection of beam bending in y-z plane
u_Θ	lateral deflection of beam bending in x-z plane
u_Ψ	angular deflection of beam twisting about z axis
X,Y,Z	position variables
Δ	displacement of proof-mass actuator
δ	line-of-sight pointing requirement
ε	noise contaminating angular velocity measurements

Θ, ϕ, Ψ pitch, roll, heading

ζ damping ratio

τ noise contaminating acceleration measurements

ω_1 angular velocity of Shuttle body

ω_4 angular velocity of reflector body

DISCUSSION

The objective of the NASA-IEEE Design Challenge concerning the control of flexible spacecraft is to promote direct comparison of different approaches to control, state estimation and systems identification. The design challenge has principal parts, the first using a mathematical model, and the second using laboratory experimental apparatus. The specific parts of the Spacecraft Control Laboratory Experiment (SCOLE) program will be discussed in detail.

Control Objectives

The primary control task is to rapidly slew or change the line-of-sight of an antenna attached to the space Shuttle orbiter, and to settle or damp the structural vibrations to the degree required for precise pointing of the antenna. The objective will be to minimize the time required to slew and settle, until the antenna line-of-sight remains within the angle δ. A secondary control task is to change direction during the "on-target" phase to prepare for the next slew maneuver. The objective is to change attitude and stabilize as quickly as possible, while keeping the line-of-sight error less than δ.

Math Model Dynamics

The initial phase of the design challenge will use a mathematical model of the Shuttle orbiter/antenna configuration. It is necessary to obtain a balance, of course, between complex formulations which might be more accurate and simplified formulations which ease the burden of analysis.

The dynamics are described by a distributed parameter beam equation with rigid bodies, each having mass and inertia at either end. One body represents Space Shuttle orbiter; the other body is the antenna reflector. The equations for the structural dynamics and Shuttle motion are formed by adding to the rigid-body equations of motion, beam-bending and torsion equations. The boundary conditions at the ends of the beam contain the forces and moments of the rigid Shuttle and reflector bodies. The nonlinear kinetmatics couples the otherwise uncoupled beam equations. Additional terms represent the action of two, 2-axis proof-mass actuators at locations on the beam chosen by the designer.

The rigid-body equations of motion for the Shuttle body are given by:

$$\dot{\omega}_1 = - I_1^{-1}(\tilde{\omega}_1 I_1 \omega_1 + M_1 + M_D + M_{B,1})$$

$$\dot{v} = \frac{F_{B,1}}{m_1}$$

Similarly, for the reflector body,

$$\dot{\omega}_4 = - I_4^{-1}(\tilde{\omega}_4 I_4 \omega_4 + M_4 + M_{B,4})$$

$$\dot{v}_4 = \frac{F_4 + F_{B,4}}{m_4}$$

The direction cosine matrices defining the attitudes of the Shuttle and reflector bodies are given by:

$$\dot{T}_1^T = - \tilde{\omega}_1 T_1^T$$

$$\dot{T}_4^T = - \tilde{\omega}_4 T_4^T$$

The direction cosine matrices defining the attitudes of the Shuttle and the reflector bodies are related to the beam end conditions.

$$T_4 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Delta\phi & -\sin\Delta\phi \\ 0 & \sin\Delta\phi & \cos\Delta\phi \end{bmatrix} \begin{bmatrix} \cos\Delta\Theta & 0 & \sin\Delta\Theta \\ 0 & 1 & 0 \\ -\sin\Delta\Theta & 0 & \cos\Delta\Theta \end{bmatrix} \begin{bmatrix} \cos\Delta\Psi & -\sin\Delta\Psi & 0 \\ \sin\Delta\Psi & \cos\Delta\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} T_1$$

where:

$$\Delta\Psi = u_\Psi \Big|_{s=L} - u_\Psi \Big|_{s=0}$$

$$\Delta\Theta = \frac{\partial u_\Theta}{\partial s} \Big|_{s=L} - \frac{\partial u_\Theta}{\partial s} \Big|_{s=0}$$

$$\Delta\phi = \frac{\partial u_\phi}{\partial s} \Big|_{s=L} - \frac{\partial u_\phi}{\partial s} \Big|_{s=0}$$

The equations of motion for the flexible beam-like truss connecting the reflector and Shuttle bodies consist of standard beam bending and torsion partial differential equations with energy dissapative terms which enable damped modes with constant characteristics for fixed, though dynamic, end conditions. The system of equations can be viewed as driven by changing end conditions and forces applied at the locations of the proof-mass actuators.

ROLL BEAM BENDING:

$$PA \frac{\partial^2 u_\phi}{\partial t^2} + 2\zeta_\phi \sqrt{PA\ EI_\phi} \frac{\partial^3 u_\phi}{\partial s^2 \partial t} + EI_\phi \frac{\partial^4 u_\phi}{\partial s^4} = \sum_{n=1}^{4} \left[f_{\phi,n} \delta(s-s_n) + g_{\phi,n} \frac{\partial \delta}{\partial s} (s-s_n) \right]$$

PITCH BEAM BENDING:

$$PA \frac{\partial^2 u_\Theta}{\partial t^2} + 2\zeta_\Theta \sqrt{PA\ EI_\Theta} \frac{\partial^3 u_\Theta}{\partial s^2 \partial t} + EI_\Theta \frac{\partial^4 u_\Theta}{\partial s^4} = \sum_{n=1}^{4} \left[f_{\Theta,n} \delta(s-s_n) + g_{\Theta,n} \frac{\partial \delta}{\partial s} (s-s_n) \right]$$

YAW BEAM TORSION:

$$PI_\Psi \frac{\partial^2 u_\Psi}{\partial t^2} + 2\zeta_\Psi I_\Psi \sqrt{GP} \frac{\partial^3 u_\Psi}{\partial s^2 \partial t} + GI_\Psi \frac{\partial^2 u_\Psi}{\partial s^2} = \sum_{n=1}^{4} g_{\Psi,n} \delta(s - s_n)$$

where:

$$f_{\phi,1} = m_1 \left. \frac{\partial^2 u_\phi}{\partial t^2} \right|_{s=0} \qquad \{\text{SHUTTLE BODY FORCE}\}$$

$$f_{\phi,2} = m_2 \left. \frac{\partial^2 u_\phi}{\partial t^2} \right|_{s=s_2} + m_2 \frac{\partial^2 \Delta_{\phi,2}}{\partial t^2} \qquad \{\text{PROOF-MASS ACTUATOR FORCE}\}$$

$$f_{\phi,3} = m_3 \left.\frac{\partial^2 u_\phi}{\partial t^2}\right|_{s=s_3} + m_3 \frac{\partial^2 \Delta_{\phi,2}}{\partial t^2} \qquad \{\text{PROOF-MASS ACTUATOR}\}$$

$$f_{\phi,4} = m_4 \left.\frac{\partial^2 u_\phi}{\partial t^2}\right|_{s=130} - I_{zz,4} \frac{\partial^2 u_\psi}{\partial t^2} /32.5 + F_y \qquad \{\text{REFLECTOR BODY FORCE}\}$$

$$f_{\Theta,1} = m_1 \left.\frac{\partial^2 u_\Theta}{\partial t^2}\right|_{s=s_1} \qquad \{\text{SHUTTLE BODY FORCE}\}$$

$$f_{\Theta,2} = m_2 \left.\frac{\partial^2 u_\Theta}{\partial t^2}\right|_{s=s_2} + m_2 \frac{\partial^2 \Delta_{\Theta,2}}{\partial t^2} \qquad \{\text{PROOF-MASS ACTUATOR FORCE}\}$$

$$f_{\Theta,3} = m_3 \left.\frac{\partial^2 u_\Theta}{\partial t^2}\right|_{s=s_3} + m_3 \frac{\partial^2 \Delta_{\Theta,2}}{\partial t^2} \qquad \{\text{PROOF-MASS ACTUATOR FORCE}\}$$

$$f_{\Theta,4} = m_4 \left.\frac{\partial^2 u_\Theta}{\partial t^2}\right|_{s=130} - I_{zz,4} \frac{\partial^2 u_\psi}{\partial t^2} /18.75 - F_x \qquad \{\text{REFLECTOR BODY FORCE}\}$$

$$\begin{pmatrix} g_{\phi,1} \\ g_{\Theta,1} \\ g_{\Psi,1} \end{pmatrix} = I_1 \dot{\omega}_1 + \omega_1 I_1 \omega_1 + M_1 + M_D \qquad \{\text{SHUTTLE BODY, MOMENTS}\}$$

$$\begin{pmatrix} g_{\phi,2} \\ g_{\Theta,2} \\ g_{\Psi,2} \end{pmatrix} = 0 \qquad \{\text{PROOF-MASS ACTUATOR, MOMENT}\}$$

$$\begin{pmatrix} g_{\phi,3} \\ g_{\Theta,3} \\ g_{\Psi,3} \end{pmatrix} = 0 \qquad \{\text{PROOF-MASS ACTUATOR, MOMENT}\}$$

$$\begin{pmatrix} g_{\phi,4} \\ g_{\Theta,4} \\ g_{\Psi,4} \end{pmatrix} = I_4 \dot{\omega}_4 + \omega_4 I_4 \omega_4 + M_4 + \tilde{R}_B F_{B,4} \qquad \{\text{REFLECTOR BODY, MOMENT}\}$$

The angular velocity of the reflector body is related to the Shuttle body by:

$$\omega_4 = \begin{pmatrix} \frac{\partial^2 u_\phi}{\partial s \partial t}\Big|_{s=L} \\ \frac{\partial^2 u_\Theta}{\partial s \partial t}\Big|_{s=L} \\ \frac{\partial u_\Psi}{\partial t}\Big|_{s=L} \end{pmatrix} - \begin{pmatrix} \frac{\partial^2 u_\phi}{\partial s \partial t}\Big|_{s=0} \\ \frac{\partial^2 u_\Theta}{\partial s \partial t}\Big|_{s=0} \\ \frac{\partial u_\Psi}{\partial t}\Big|_{s=0} \end{pmatrix} + \omega_1 \qquad \tilde{R}_B = \begin{bmatrix} 0 & 130 & 0 \\ -130 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The line-of-sight error described in figure 2 is affected by both the pointing error of the Shuttle body and the misalignment of the reflector due to the deflection of the beam supporting the reflector. The line-of-sight is defined by a ray from the feed which is reflected at the center of the reflector. Its direction in the Shuttle body coordinates is given by:

$$R_{LOS} = \frac{-R_R + R_F + 2\left[R_A^T(R_R - R_F) \cdot R_A\right]}{\left|\left|R_R - R_F - 2\left[R_A^T R_R - R_F\right] \cdot R_A\right|\right|}$$

where:

R_F is the feed location (3.75, 0, 0)

R_R is the location of the center of the reflector (18.75, -32.5, -130)

R_A is a unit vector in the direction of the reflector axis in Shuttle body coordinates

The vector R_A can be related to the direction cosine attitude matrices for the Shuttle body, T_1, and the reflector body, T_4, by

$$R_A = \left[T_1^T T_4\right] \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

The relative alignment of the reflector to the Shuttle body is given by $T_1^T T_4$ which is a function of the structural deformations of the beam.

The line-of-sight error, e, is the angular difference between the target direction, given by the unit vector, D_T, and the line-of-sight direction in Earth axes, T_1R_{LOS}.

$$e = \text{ARCSIN} \left| D_T \times T_1 R_{LOS} \right| \quad \text{or} \quad \text{ARCSIN} \left| \tilde{D}_T T_1 R_{LOS} \right|$$

Computer programs are available which generate time histories of the rigid body and the mode shapes and frequencies for the body-beam-body configuration for "pitch" bending, "roll" bending and "yaw" twisting. Since the modes are based on solving explicitly the distributed parameter equations (without damping and without kinematic coupling) there is no limit to the number of modal characteristic sets that can be generated by the program. It will be the analyst's decision as to how many modes need to be considered.

Laboratory Experiment Description

The second part of the design challenge is to validate in the laboratory, the system performance of the more promising control system designs of the first part. The experimental apparatus will consist of a dynamic model of the Space Shuttle orbiter with a large antenna reflector attached by means of a flexible beam. The dynamic model will be extensively instrumented and will have attached force and moment generating devices for control and for disturbance generation. A single, flexible tether will be used to suspend the dynamic model, allowing complete angular freedom in yaw, and limited freedom in pitch and roll. An inverted position will be used to let the reflector mast to hang so that gravity effects on mast bending will be minimized. The dynamics of the laboratory model will of necessity be different from the mathematical model discussed earlier.

Design Challenge, Part One

For part one of the design challenge, the following mathematical problem is addressed. Given the dynamic equations of the Shuttle/antenna configuration, what control policy minimizes the time to slew to a target and to stabilize so that the line-of-sight (LOS) error is held, for a time, within a specified amount, δ. During the time that the LOS error is within δ, the attitude must change 90° to prepare for the next slew maneuver. This was previously referred to as the sescondary control task. The maximum moment and force generating capability will be limited. Advantage may be taken of selecting the most suitable initial alignment of the Shuttle/antenna about its assigned initial RF axis, line-of-sight. Random, broad band-pass disturbances will be applied to the configuration. Two proof-mass, force actuators may be positioned anywhere along the beam. The design guidelines are summarized below:

1. The initial line-of-sight error is 20 degrees.

$$e(o) = 20 \text{ degrees}$$

2. The initial target direction is straight down.

$$D_T = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

3. The initial alignment about the line-of-sight is free to be chosen by the designer. Advantage may be taken of the low value of moment of inertia in roll. The Shuttle/antenna is at rest initially.

4. The objective is to point the line-of-sight of the antenna and stabilize to within 0.02 degree of the target as quickly as possible.

$$\delta = 0.02 \text{ degree}$$

5. Control moments can be applied at 100 Hz sampling rate to both the Shuttle and reflector bodies of 10,000 ft-lb for each axis. The commanded moment for each axis is limited to 10,000 ft-lb. The actual control moment's response to the commanded value is first-order with a time constant of 0.1 second.

   For the rolling moment applied to the Shuttle body:

$$-10^4 \leq M_{X,1,\text{command}} \leq 10^4$$

$$M_{X,1}(n+1) = e^{-0.1}\, M_{X,1}(n) + (1 - e^{-0.1})\, M_{X,1,\text{command}}(n)$$

   Equations for other axes and for the reflector body are similar.

6. Control forces can be applied at the center of the reflector in the X and Y directions only. The commanded force in a particular direction is limited to 800 lbs. The actual control force's response to the commanded value is first-order with a response time of 0.1 second.

   For the side for applied to the reflector body:

$$-800 \leq F_{Y,\text{command}} \leq 800$$

$$F_Y(n+1) = e^{-0.1}\, F_Y(n) + (1 - e^{-0.1})\, F_{Y,\text{command}}(n)$$

   Equations for X-axis are similar.

7. Control forces using two proof-mass actuators (each having both X and Y axes) can be applied at two points on the beam. The strokes are limited to ± 1 ft, and the masses weight 10 lbs each. The actual stroke follows a first-order response to limited commanded values.

For the X-axis of the proof-mass actuator at s_2:

$$-1 \leq \Delta_{X,2,\text{command}} \leq 1$$

$$\Delta_{X,2}(n+1) = e^{-0.1}\,\Delta_{X,2}(n) + (1 - e^{-0.1})\,\Delta_{X,2,\text{command}}(n)$$

Equations for other axes and locations are similar.

8. The inertial attitude direciton cosine matrix for the Shuttle body lags in time the actual values by 0.01 second and are made at a rate of 100 samples per second. Each element of the direction cosine measurement matarix is contaminated by additive, uncorrelated Gaussian noise having an rms value of 0.001. The noise has zero mean.

$$T_{s,\text{measured}}(n+1) = T_{s,\text{true}}(n) + \begin{bmatrix} d_{11}(n) & d_{12}(n) & d_{13}(n) \\ d_{21}(n) & d_{22}(n) & d_{23}(n) \\ d_{31}(n) & d_{32}(n) & d_{33}(n) \end{bmatrix}$$

where:

$$E\{d_{ij}(n)\} = 0$$

$$E\{d_{ij}(n)d_{kL}(n)\} = 0 \qquad \text{for } i \neq k \text{ or } j \neq L$$

$$E\{d_{ij}(n)d_{ij}(n+k)\} = 0 \qquad \text{for } k \neq 0$$

$$= [.001]^2 \qquad \text{for } k = 0$$

9. The angular velocity measurements for both the Shuttle and reflector bodies pass through a first-order filter with 0.05 sec time constant and lag in time the actual values by 0.01 second and are made at a rate of 100 samples per second. Each rate measurement is contaminated by additive, Gaussian, uncorrelated noise having an rms value of 0.02 degree per second. The noise has zero mean.

For example:

$$\omega_{1,X,\text{measured}}(n+1) = \omega_{1,X,\text{filtered}}(n) + \varepsilon_{1,X}(n)$$

$$E\{\varepsilon_{1,X}(n)\ \varepsilon_{1,X}(n+k)\} = 0 \qquad \text{for } k \neq 0$$

$$= (.02)^2 \qquad \text{for } k = 0$$

where

$$\dot{\omega}_{1,X,\text{filtered}} = -20\ \omega_{1,X,\text{filtered}} + 20\ \omega_{1,X,\text{true}}$$

10. Three-axis accelerometers are located on the Shuttle body at the base of the mast and on the reflector body at its center. Two-axes (X and Y) accelerometers are located at intervals of 10 feet along the mast. The acceleration measurements pass through a first-order filter with a 0.05 second time constant and lag in time the actual values by 0.01 second, and are made at a rate of 100 samples per second. Each measurement is contaminated by Gaussian additive, uncorrelated noise having an rms value of 0.05 ft/sec^2.

For example:

$$a_{1,X,\text{measured}}(n+1) = a_{1,X,\text{filtered}}(n) + \tau_{1,X}(n)$$

$$E\{\tau_{1,X}(n)\ \tau_{1,X}(n+k)\} = 0 \qquad \text{for } k \neq 0$$

$$= (.05)^2 \qquad \text{for } k = 0$$

where:

$$\dot{a}_{1,X,\text{filtered}} = -20\, a_{1,X,\text{filtered}} + 20\, \omega_{1,X,\text{true}}$$

11. Gaussian, uncorrelated step-like disturbances are applied 100 times per second to the Shuttle body in the form of 3-axes moments, having rms values of 100 ft-lbs. These disturbances have zero mean.

    For example:

$$E\{M_{D,X}(n)\ M_{D,X}(n+k)\} = 0 \qquad \text{for } k \neq 0$$

$$= (100)^2 \qquad \text{for } k = 0$$

In summary, the designer's task for part one is to: (1) derive a control law for slewing and stabilization, coded in FORTRAN; (2) select an initial attitude in preparation for slewing 20 degrees; and (3) select two positions for the 2-axes proof-mass actuators. An official system performance assessment computer program will be used to establish the time required to slew and stabilize the Shuttle/antenna configuration.

Design Challenge, Part Two

As in part one, the task is to minimize the time to slew and stabilize a Shuttle/antenna configuration. The difference is that in part two of the design challenge, a physical laboratory model will be used instead of the dynamic equations of part one. The constraints on total moment and force generation capability will apply to part two, as for part one. Again, the analyst may select the initial alignment about the assigned initial RF line-of-sight. Disturbances will be injected into the Shuttle/antenna model. The designer's task will be similar to that for part one.

CONCLUDING REMARKS

A Design Challenge, in two parts, has been offered for the purpose of comparing directly different approach to controlling a flexible Shuttle/antenna configuration. The first part of the design challenge uses only mathematical equations of the vehicle dynamics; the second part uses a physical laboratory model of the same configuration. The Spacecraft Control Laboratory Experiment (SCOLE) program is being conducted under the cognizance of the Spacecraft Control Branch at the NASA Langley Research Center. The NASA/IEEE Design Challenge has the advice and counsel of the IEEE-COLSS Subcommittee on Large Space Structures. Workshops will be held to enable investigators to compare results of their research.

MASS CHARACTERISTICS

	CG LOCATION, FT X	Y	Z	WEIGHT, LB	I_{XX} SLG-FT^2	I_{YY} SLG-FT^2	I_{ZZ} SLG-FT^2	I_{XY} SLG-FT^2	I_{XZ} SLG-FT^2	I_{YZ} SLG-FT^2
SHUTTLE	0	0	0	205,000	905,443	6,789,100	7,086,601	0	145,393	0
MAST, CG	0	0	-65.	400	17,495	17,495	0	0	0	0
REFLECTOR, CG	18.75	-32.5	-130.	400	4,969	4,969	9,938	0	0	0
REFLECTOR, ATTACHMENT POINT					18,000	9,336	27,407	-7,570	0	0
TOTAL	.036	-.063	-.379	205,800	1,132,508	7,007,447	7,113,962	-7,555	115,202	52,293

The moment of inertia becomes:

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} = \begin{bmatrix} 1,132,508 & 7,555 & -115,202 \\ 7,555 & 7,007,447 & -52,293 \\ -115,202 & -52,293 & 7,113,962 \end{bmatrix}$$

$$I_1 = \begin{bmatrix} 905,443 & 0 & -145,393 \\ 0 & 6,789,100 & 0 \\ -145,393 & 0 & 7,086,601 \end{bmatrix}$$

$$I_4 = \begin{bmatrix} 4,969 & 0 & 0 \\ 0 & 4,969 & 0 \\ 0 & 0 & 9,938 \end{bmatrix}$$

$m = 6391.30$ slugs

$m_1 = 6366.46$ slugs

$m_2 = 0.3108$ slugs

$m_3 = 0.3108$ slugs

$m_4 = 12.42$ slugs

$PA = 0.09556$ slugs/ft

$EI_\phi = 4.0 \times 10^7$ lb-ft^2

$\zeta_\phi = .003$

$PA = 0.09556$ slugs/ft

$EI_\theta = 4.0 \times 10^7$ lb-ft^2

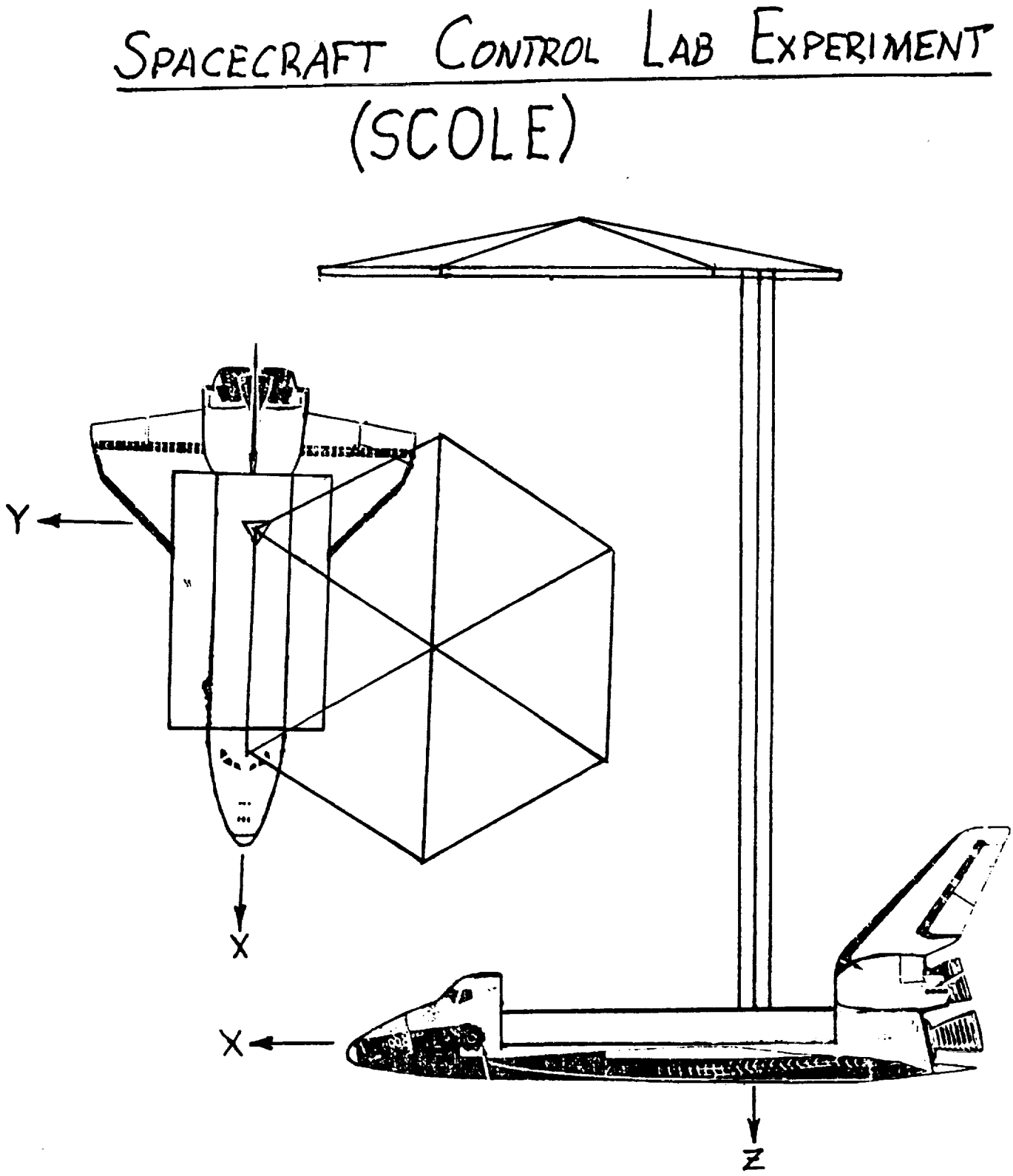
$\zeta_\theta = .003$

$PI_\psi = 0.9089$ slug-ft

$GI_\psi = 4.0 \times 10^7$ lb-ft^2

$\zeta_\psi = .003$

Figure 1. Drawing of the Shuttle/Antenna Configuration.



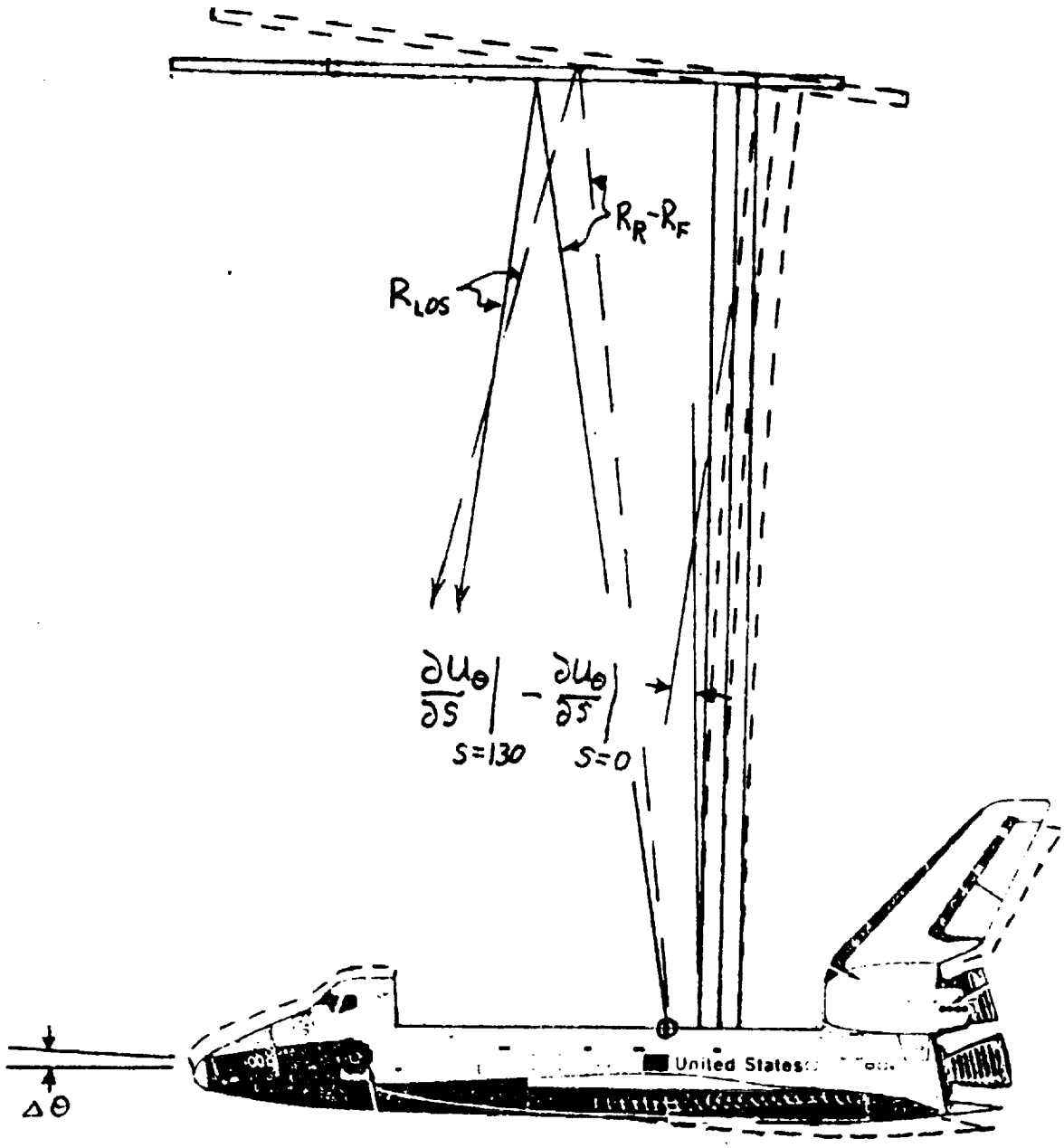
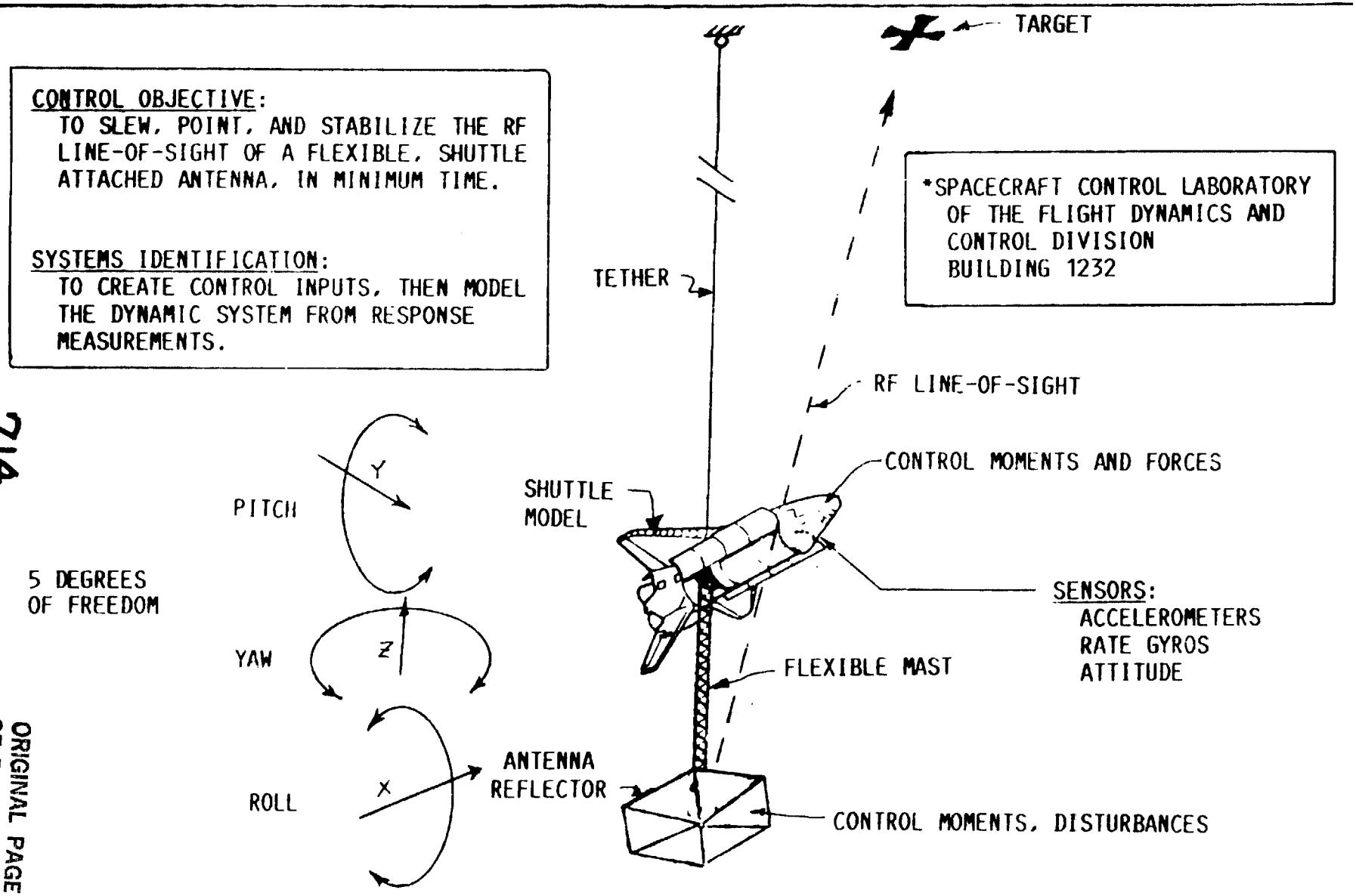


Figure 2.- Schematic of the effect of bending on the line-of-sight pointing error.

Figure 3. SPACECRAFT CONTROL LABORATORY* EXPERIMENT (SCOLE)



714

ORIGINAL PAGE IS OF POOR QUALITY

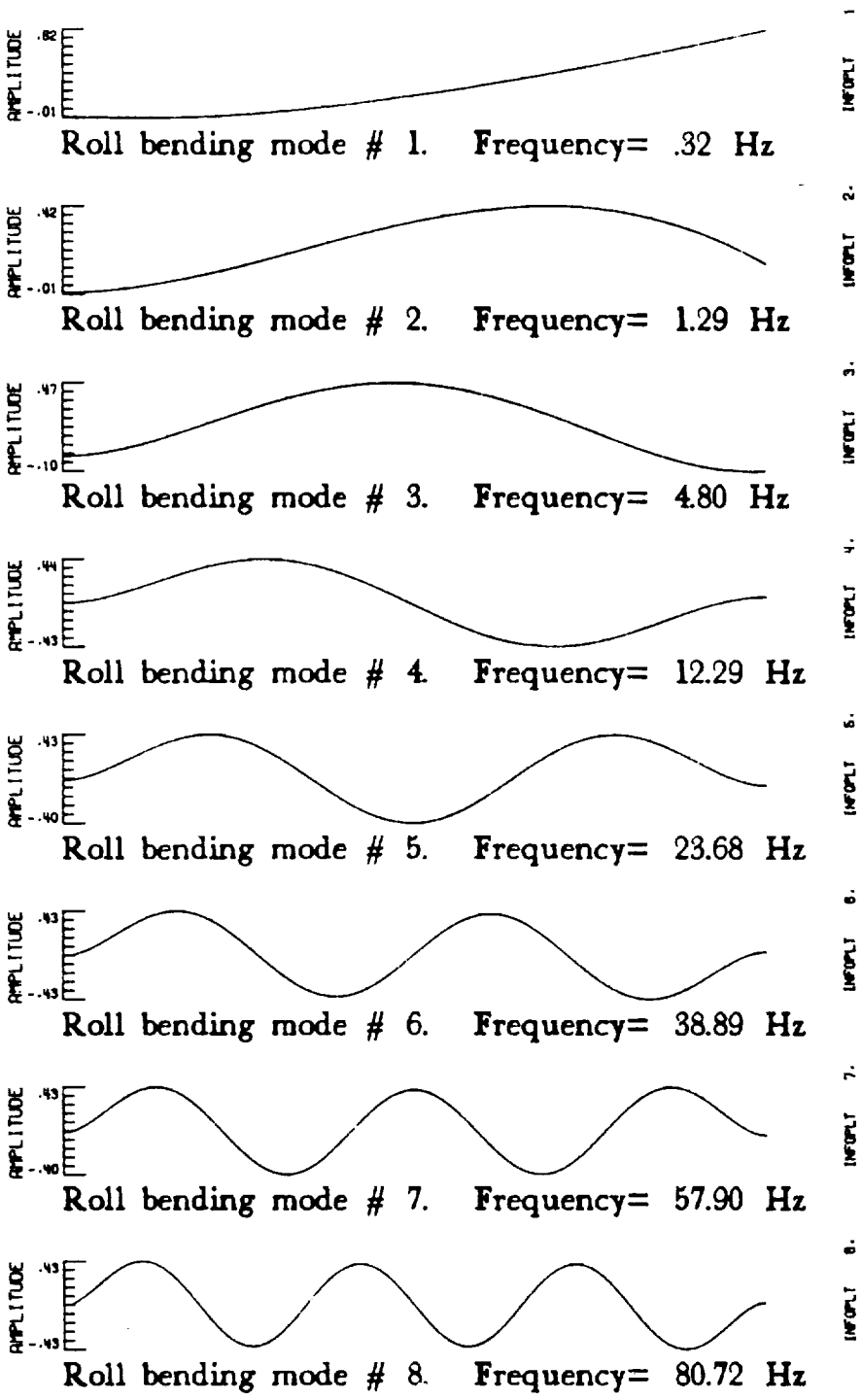


Figure 4a.- Plots of normalized roll bending mode shapes for SCOLE configuration.

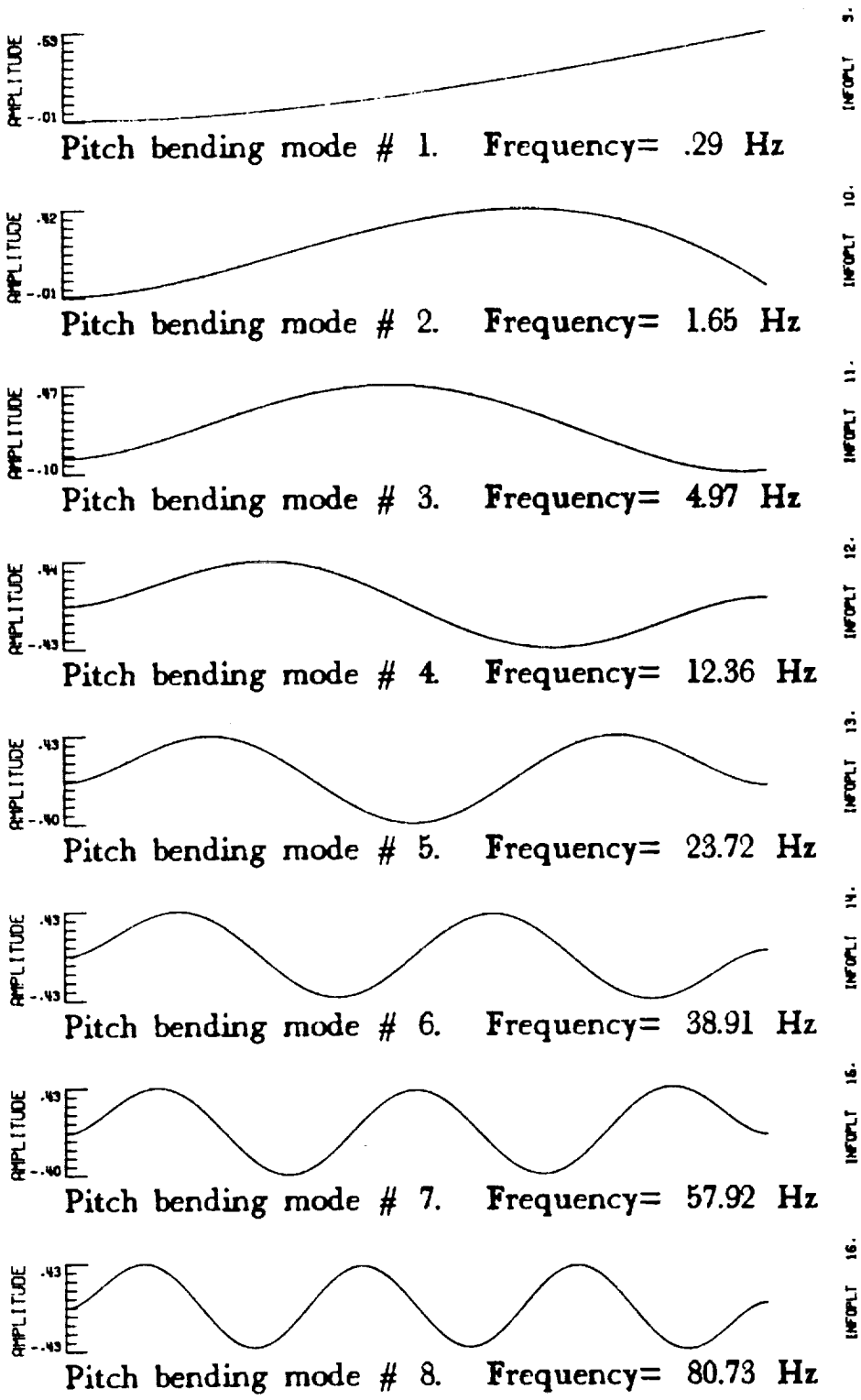


Figure 4b.- Plots of normalized pitch bending mode shapes for SCOLE configuration.

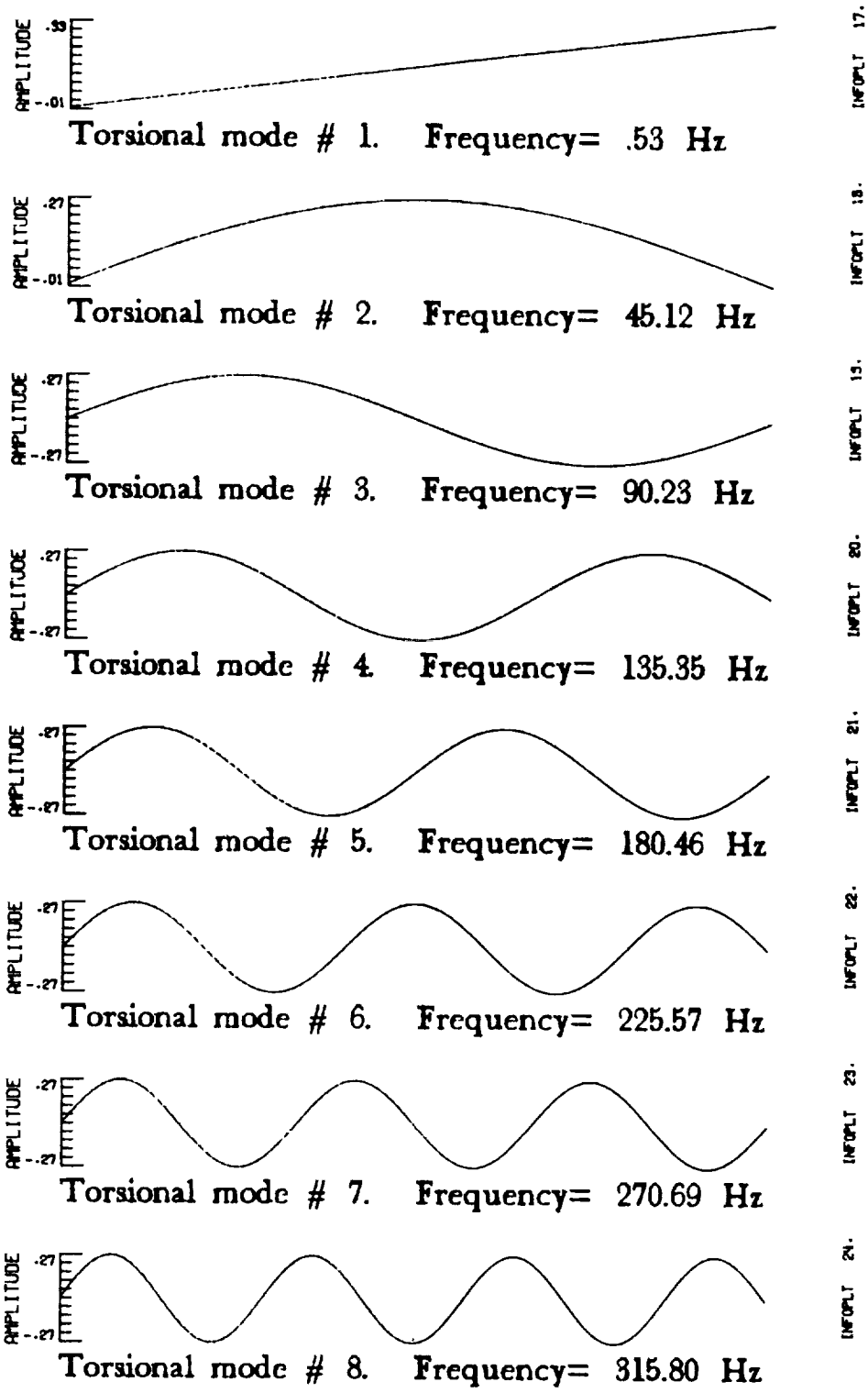


Figure 4c.- Plots of normalized torsional mode shapes for SCOLE configuration.

STATUS AND FUTURE PLANS OF THE SCOLE EXPERIMENTAL FACILITY

by

J. Shenhar

NASA Langley Research Center
Hampton, Va 23665

Fifth Annual SCOLE Workshop
Lake Arrowhead, California
October 31 – November 1, 1988

IN THE PRESENTATION ...

* INTRODUCTION TO THE SCOLE EXPERIMENT
* LATEST SYSTEM UPGRADE AND CURRENT STATUS OF THE SCOLE FACILITY
* RESEARCH ACTIVITY (IN HOUSE & GI PROGRAM)
* SOME EXPERIMENT RESULTS
  Movie: The Optimal Attitude Control Testing on SCOLE
* FUTURE PLANS
* SUMMARY

INTRODUCTION TO THE SCOLE EXPERIMENT

* THE NASA–IEEE DESIGN CHALLENGE (Taylor–Balakrishnan, 1984)

Control of flexible spacecraft to promote direct comparison of different approaches to:
- Control
- State Estimation
- System Identification

Using: 1. A mathematical model
2. A laboratory experimental apparatus

* CONTROL OBJECTIVES

- Rapid slew or change of line–of–sight of an antenna attached to the space Shuttle orbiter
- Damp the structural vibrations to a prescribed degree
- Minimize the time for slew and settle to a given orientation
- Change direction during the "on–target" phase to prepare for the next slew maneuver

INTRODUCTION (CONT.)

* THE SPACECRAFT CONTROL LABORATORY EXPERIMENT – S C O L E

  - Experiment conceived to provide a common design challenge for interested investigators
  - Develop and validate control methods for large flexible spacecraft configuration
  - Demonstration of slewing and pointing control problems along with vibration suppression
  - Utilizing antenna–like structural configuration and inertial sensors and actuators
  - Variety of inertial quality sensor and actuator types
    + Accelerometers, rate sensors, optical position sensors
    + Thrusters, control moment gyros, reaction wheels
  - Real time computing capability

LATEST SYSTEM UPGRADE AND CURRENT STATUS OF THE SCOLE FACILITY

LATEST UPGRADE:

* DUAL CAMERA ATTITUDE MEASUREMENT SYSTEM
* CAMERA POSITION IDENTIFICATION AND SCOLE ATTITUDE TRACKING ALGORITHMS
* POWER AMPLIFIERS TO ACCOMODATE THE HI–TORQUE REACTION WHEELS
* REAL TIME COMPUTER UPGRADE – HARDWARE AND SOFTWARE
* SCOLE DOCUMENTATION – HARDWARE AND SOFTWARE
* DEVELOPMENT OF CMGs STEERING LAW – IN PROGRESS

LATEST SYSTEM UPGRADE AND CURRENT STATUS OF THE SCOLE FACILITY (CONT.)

UNDER THE CURRENT STATUS, SCOLE IS CAPABLE OF CONDUCTING THE FOLLOWING TESTS:

* GENERATE VIBRATION DATA FILES FOR DISTRIBUTION
* ON–LINE PARAMETER IDENTIFICATION
* VIBRATION SUPPRESSION – CANTILEVER MODE TESTING
* FIXED PIVOT POINT – 3 DOF TESTING
* FREE PIVOT POINT – 5 DOF TESTING

RESEARCH ACTIVITY (IN–HOUSE AND GI PROGRAM)

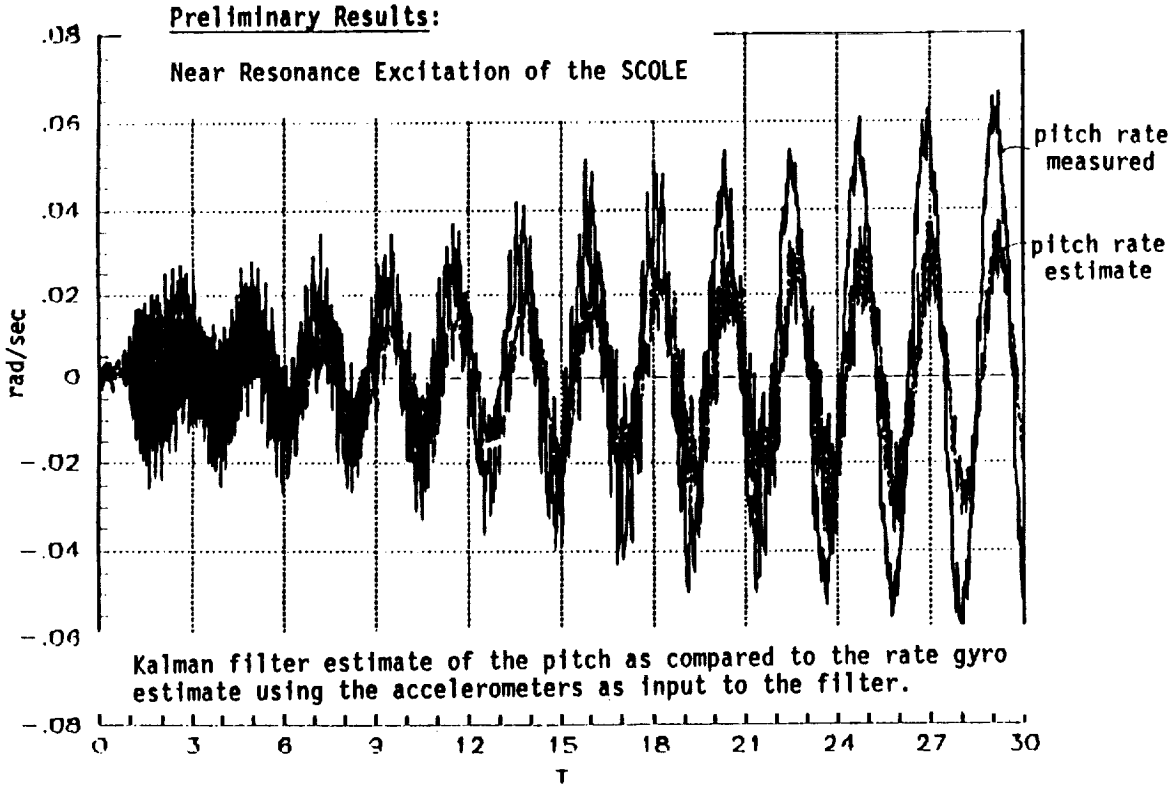
* NONLINEAR CONTROL DESIGN AND RELIABILITY ISSUES
* DEVELOP AND TEST A DISTRIBUTED ADAPTIVE CONTROL SYSTEM
* DECENTRALIZED CLOSED LOOP MANEUVER
* DISTRIBUTED OUTPUT MODEL–FOLLOWING CONTROL LAW
* PARAMETER IDENTIFICATION AND PERTURBATION METHOD CONTROLLER
* ANALYTIC REDUNDANCY MANAGEMENT FOR SYSTEM WITH APPRECIABLE STRUCTURAL DYNAMICS
* RIGID BODY ATTITUDE TRACKING ALGORITHMS
* CONTROL MOMENT GYRO STEERING LAW

SOME EXPERIMENTAL RESULTS

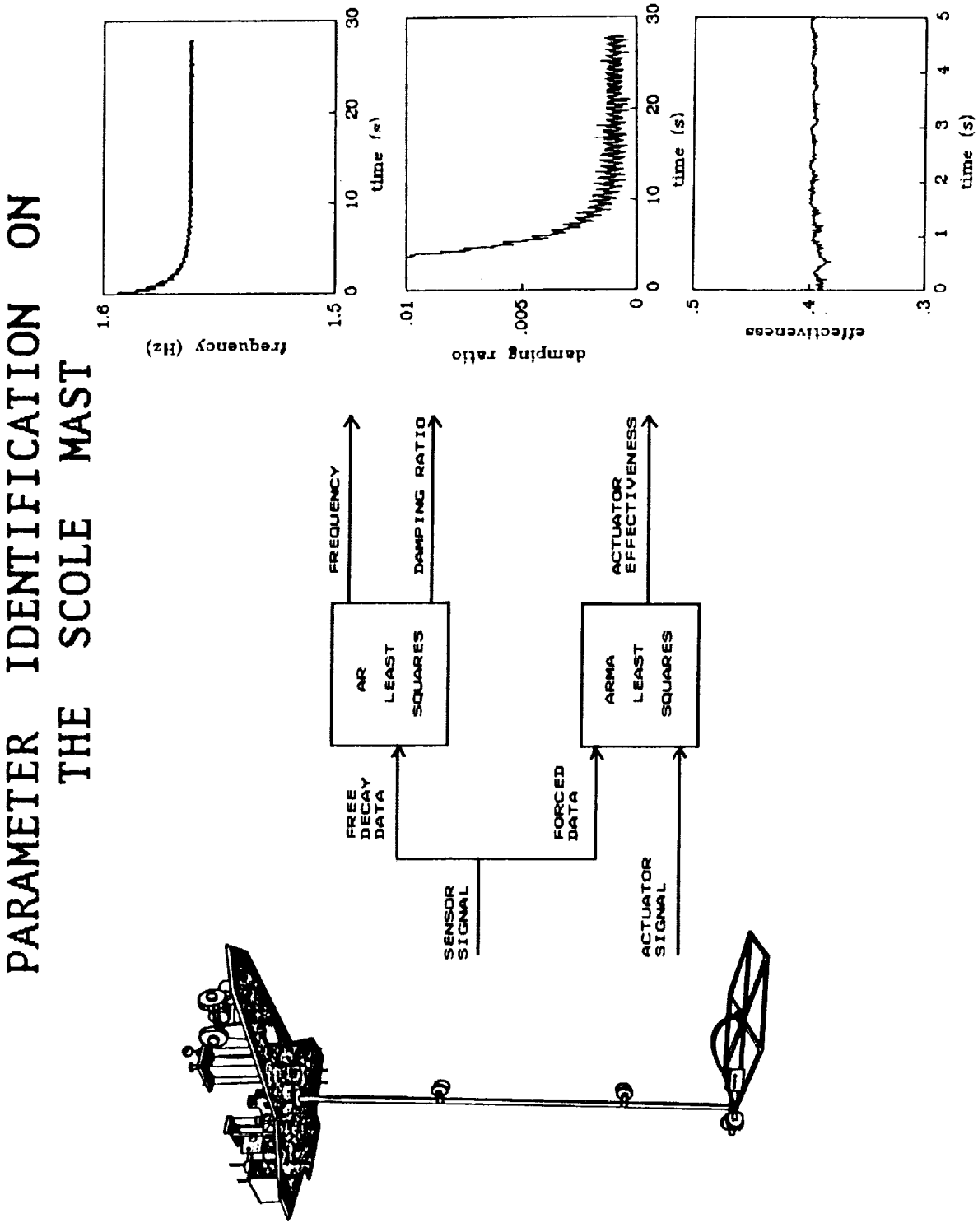
* KALMAN FILTER ESTIMATE
* PARAMETER IDENTIFICATION ON THE SCOLE MAST
* ON–LINE IDENTIFICATION AND ATTITUDE CONTROL FOR SCOLE

* MOVIE: SCOLE ATTITUDE CONTROL

KALMAN FILTER ESTIMATE



PARAMETER IDENTIFICATION ON THE SCOLE MAST



ON-LINE IDENTIFICATION AND ATTITUDE CONTROL FOR SCOLE

Parameter Identification

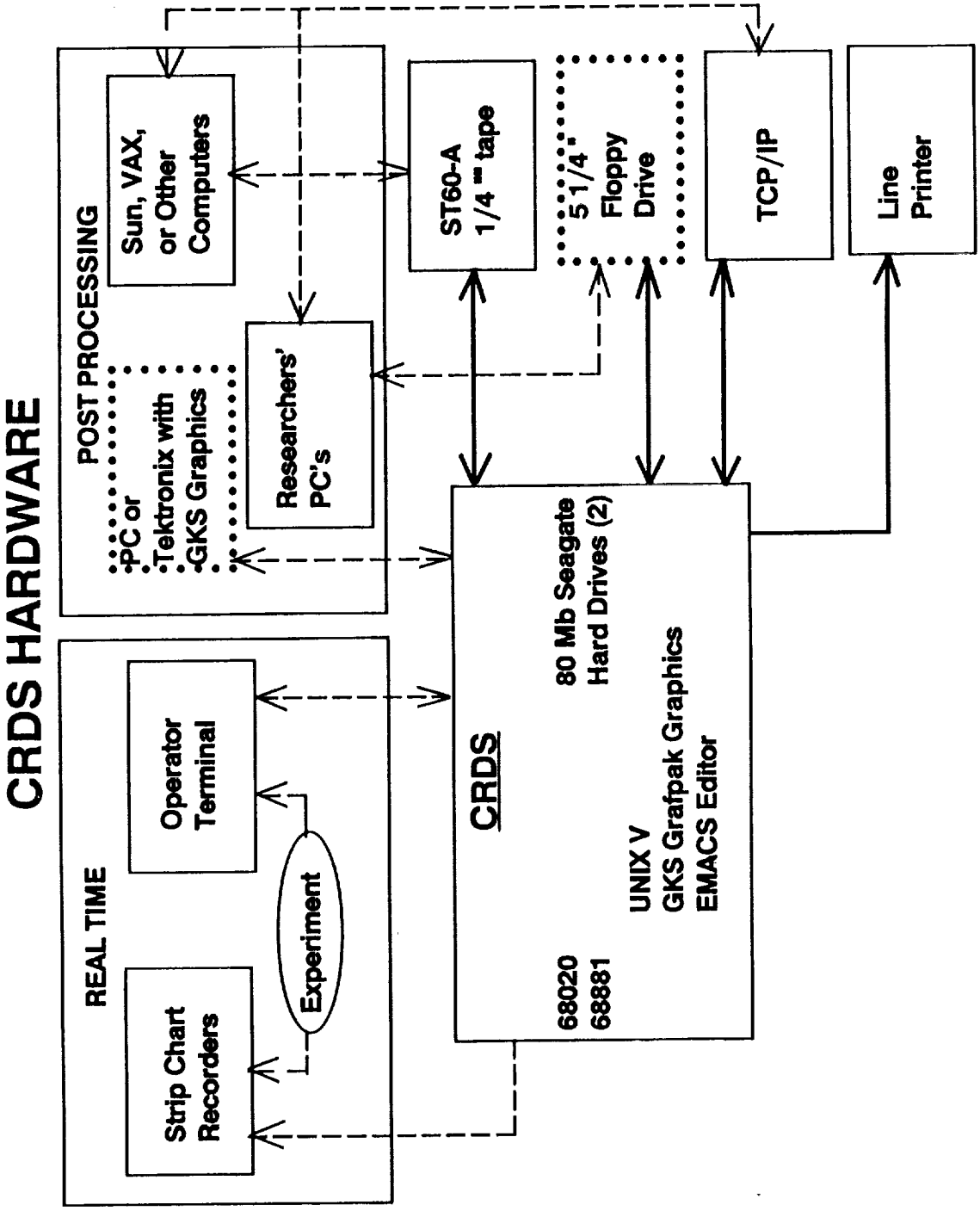
* Modeling for Identification and Control
* Free Response Coefficients
* Jet Actuators Control Effectiveness Parameters

Attitude Control

* Implementing Rigid Body Minimum-Time Control Law
* Comparing Test Results to Computer Simulation to Verify Performance of the Control Law on a Real Time System
* Minimum-Time Slew Maneuver Accomplished
  - Attitude Error was 2%

THE REAL TIME SCOLE COMPUTER

* CHARLES RIVER DATA SYSTEM (CRDS) – UNIVERS 32 MICRO–COMPUTER BASED ON THE MOTOROLA M68020 MICROPROCESSOR AND M68881 FLOATING POINT CO–PROCESSOR
* RANDOM ACCESS MEMORY 4.0 M–BYTE
* TWO 80 M–BYTE HARD DISKS
* OPERATING SYSTEM IS BASED ON AT&T UNIX SYSTEM V
* COMPILERS – FORTRAN, C, PASCAL


CRDS HARDWARE
POST PROCESSING
Sun, VAX, or Other Computers
PC or Tektronix with GKS Graphics
Researchers' PC's
ST60-A 1/4" tape
5 1/4" Floppy Drive
TCP/IP
Line Printer
REAL TIME
Operator Terminal
Experiment
Strip Chart Recorders
CRDS
80 Mb Seagate Hard Drives (2)
UNIX V
GKS Grafpak Graphics
EMACS Editor
68020
68881

CRDS SCOLE SOFTWARE

* MENU DRIVEN SCOLE EXPERIMENT OPERATING ENVIRONMENT
* USER–FRIENDLY FORTRAN & C SUBROUTINES AND FUNCTIONS TO CONTROL EXPERIMENT HARDWARE
* CAPABILITY TO STORE REALTIME SENSOR & ACTUATOR DATA IN MATRIX–X, MATLAB AND OTHER POST PROCESSING SOFTWARE PACKAGES

CRDS OPERATING ENVIRONMENT

* TCP/IP NETWORKING CAPABILITY
* USER CHOICE TO OPERATE IN UNIX V BOURNE SHELL OR UNOS VIII OPERATING SYSTEM
* FORTRAN, C, AND PASCAL COMPILERS AVAILABLE
* GKS GRAFPAK GRAPHICS
* SYSTEM ADMINISTRATOR HELP

FUTURE PLANS

* DEVELOPMENT OF CMGs STEERING LAW TO ACCOMODATE:
  - Maneuver bounds: yaw +/− 45 degrees
    pitch +/− 20 degrees
    roll +/− 20 degrees
  - Generation of CMG gimbal rate command that results in the desired control torque

* BENCHMARK PROGRAMS THAT ILLUMINATE:
  - Computer architectural issues that affect the speed of software
  - Hardware performance of the SCOLE system
  - Control effectiveness parameters

* REAL TIME COMPUTER UPGRADE (HARDWARE–SOFTWARE) ACCORDING TO FUTURE NEEDS

* DOCUMENTATION
  - Complete software user manual
  - Experimental setup diagrams
  - Various wiring diagrams to accomodate the in–house user

SUMMARY

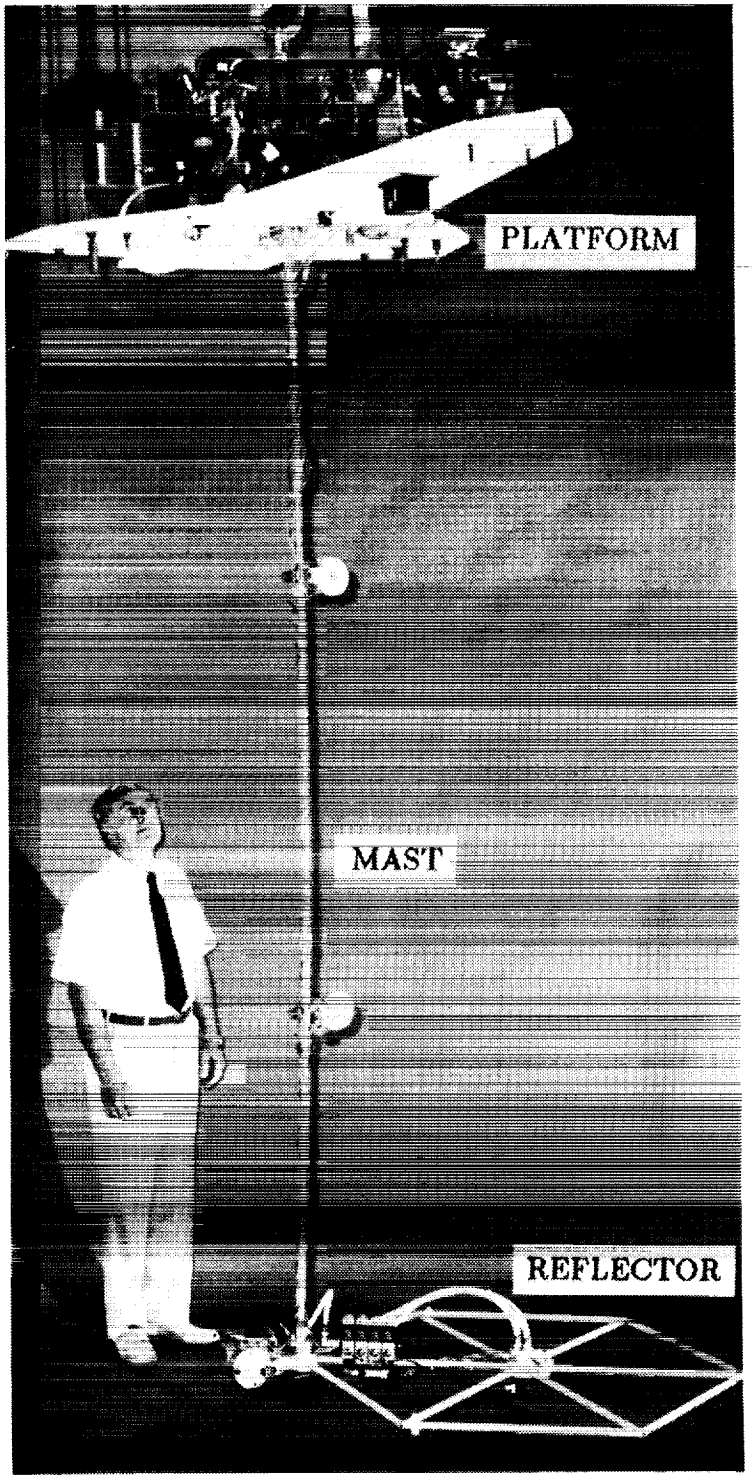
* THE SCOLE LABORATORY FACILITY IS PROVIDED AS A TEST–BED FOR EVALUATION OF CONTROL LAWS FOR LARGE FLEXIBLE STRUCTURES

* THE SCOLE APPARATUS IS EQUIPPED WITH SENSORS, ACTUATORS AND A REAL TIME COMPUTER THAT MAY ACCOMODATE VARIOUS CONTROL LAWS AND TESTS IN THE RESEARCH OF LARGE FLEXIBLE STRUCTURES

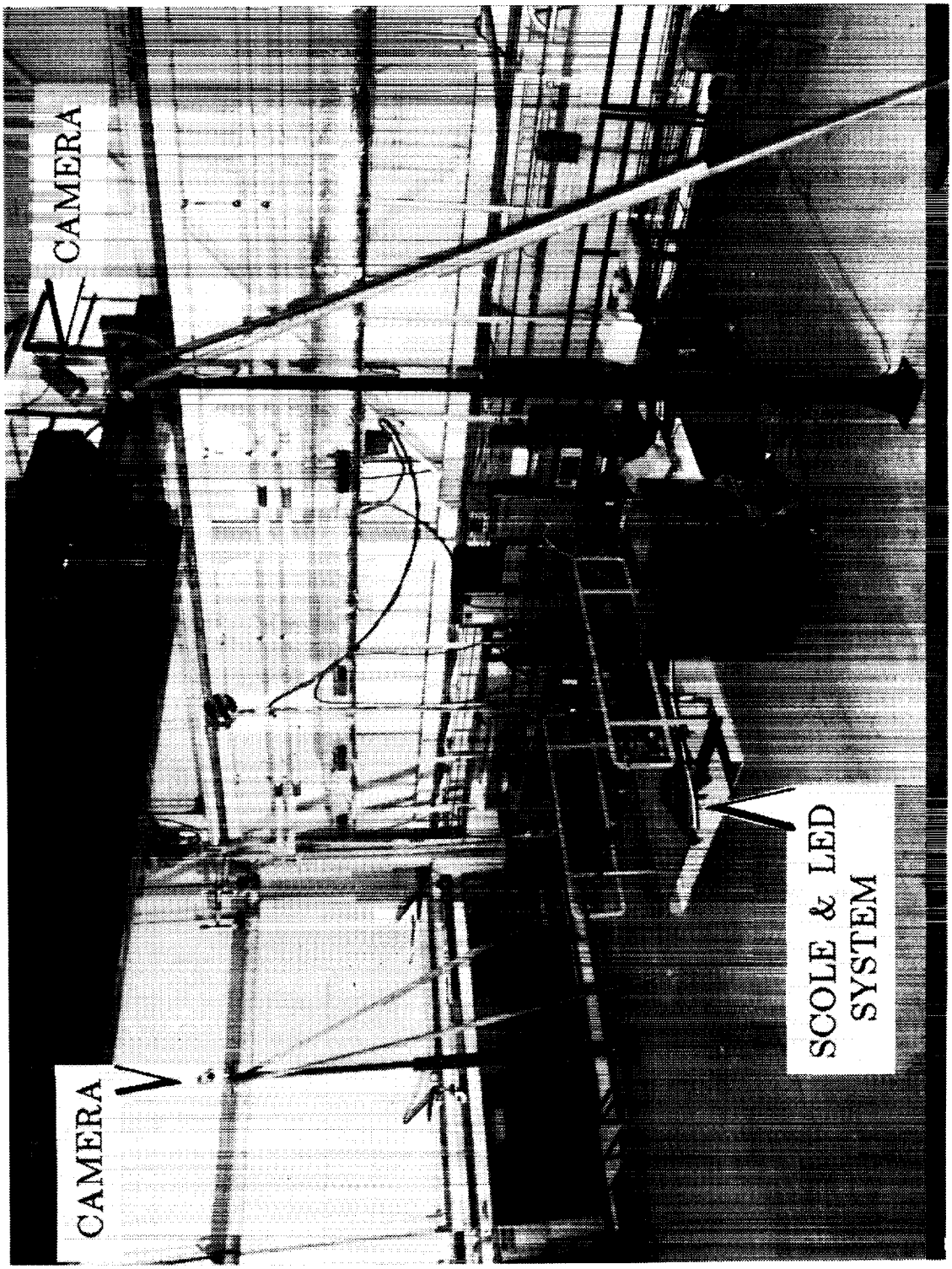
* SEVERAL CONTROL LAWS TESTED ON THE SCOLE SHOWED GOOD AGREEMENT WITH THEORY AND COMPUTER SIMULATIONS

* GUEST INVESTIGATORS ARE INVITED NOT ONLY TO DERIVE CONTROL LAWS BUT ALSO TO TEST THEM IN–HOUSE ON THE SCOLE FACILITY

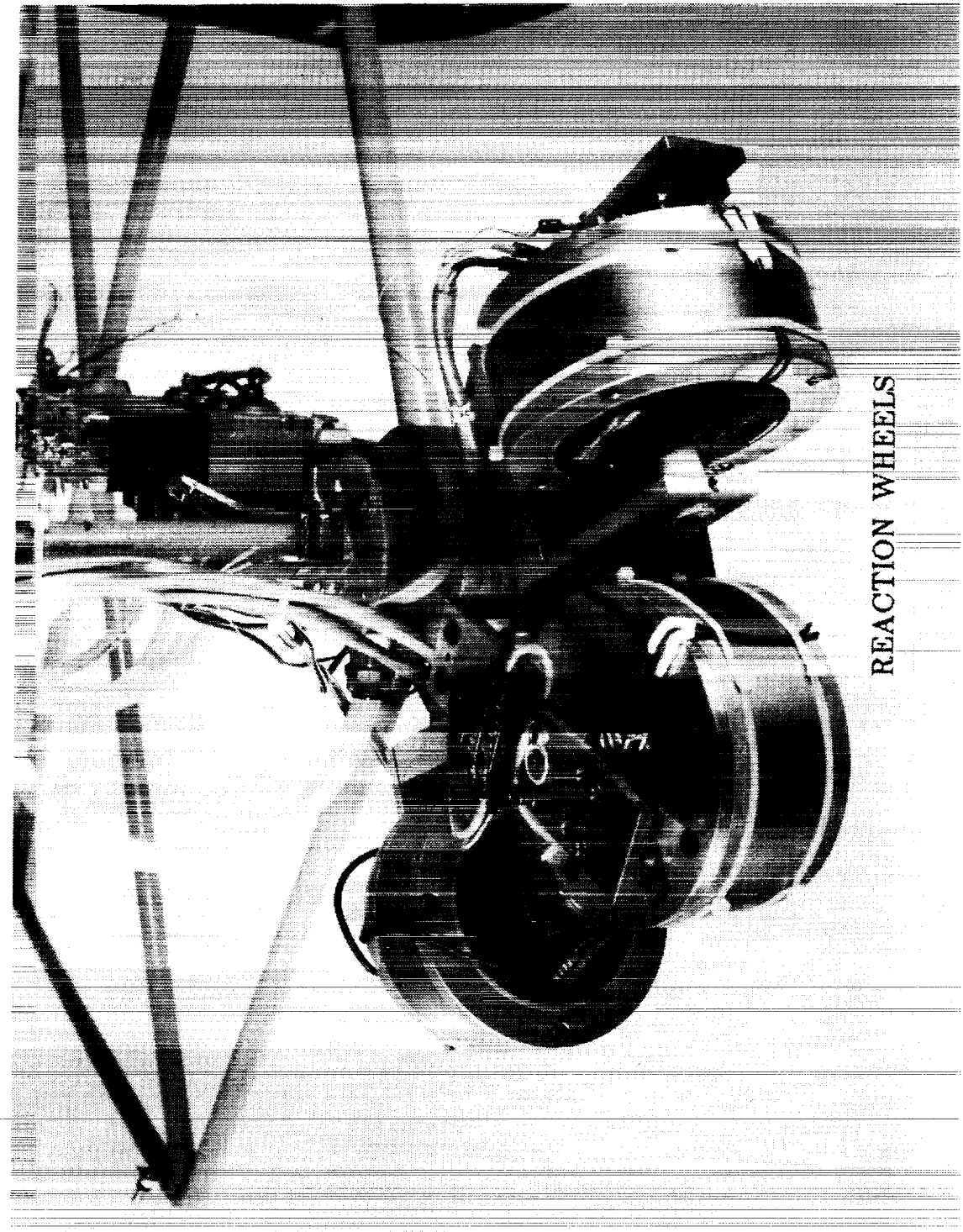
THE SCOLE EXPERIMENT APPARATUS



ATTITUDE TRACKING SYSTEM

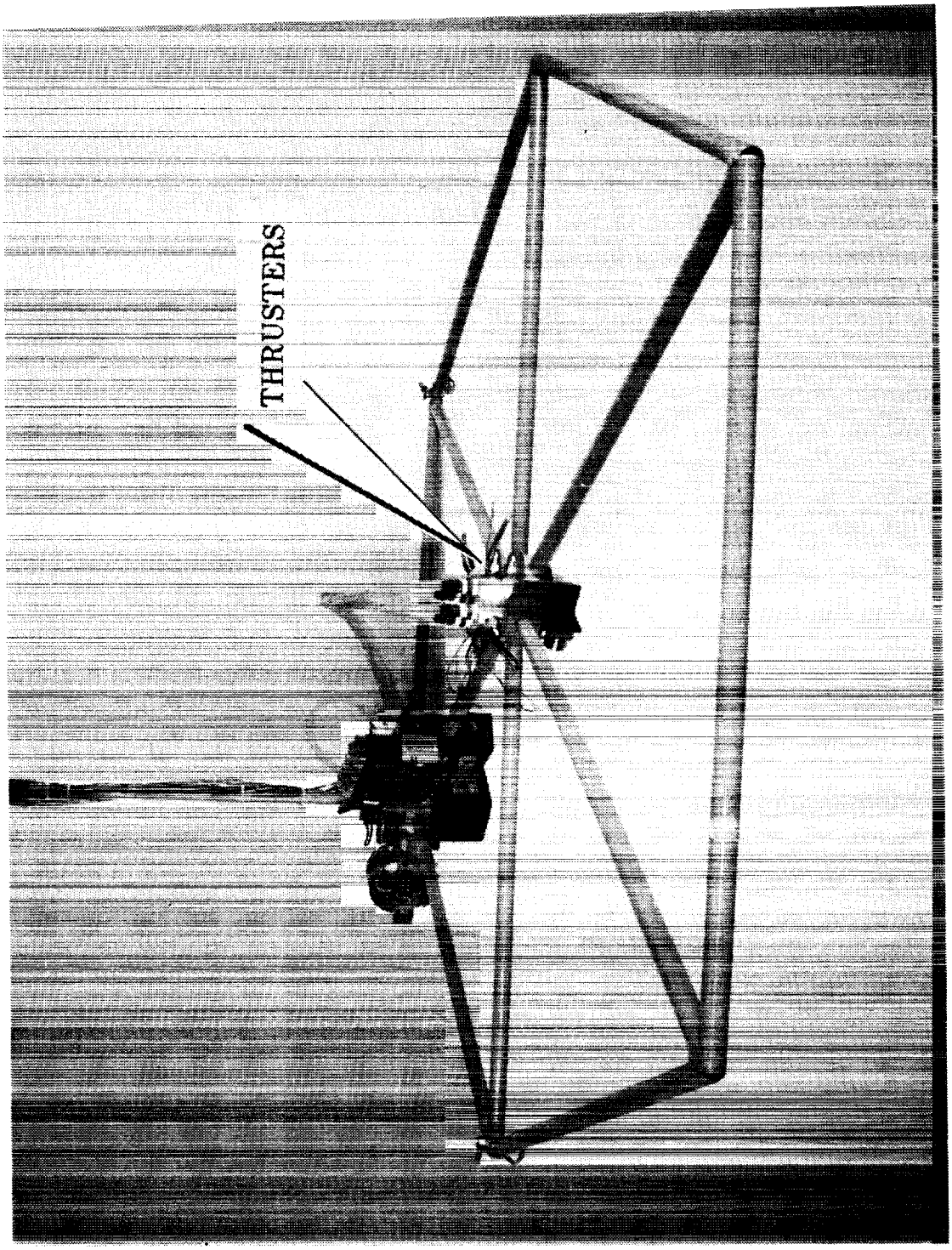


MAST–END MOUNTED REACTION WHEELS

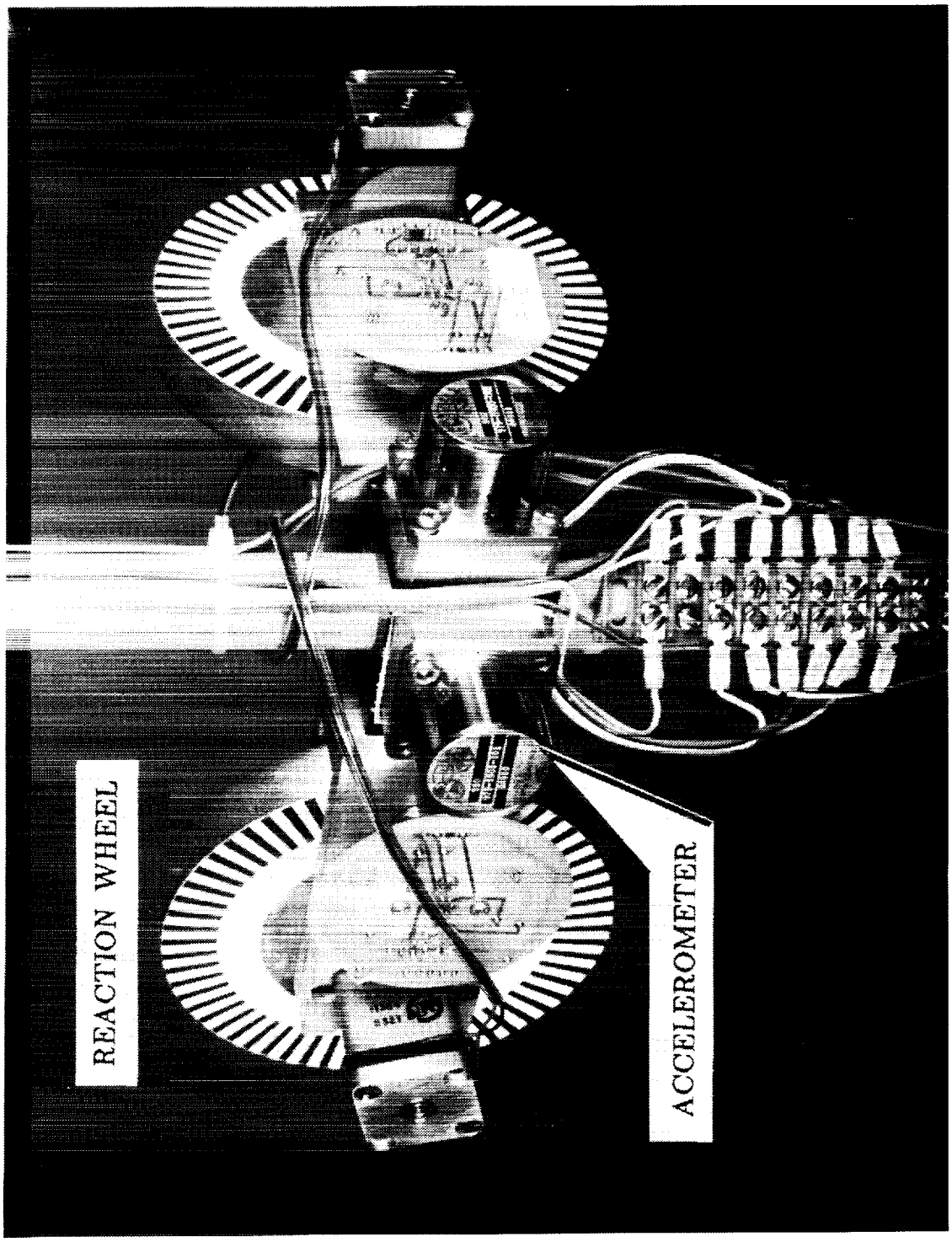


ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

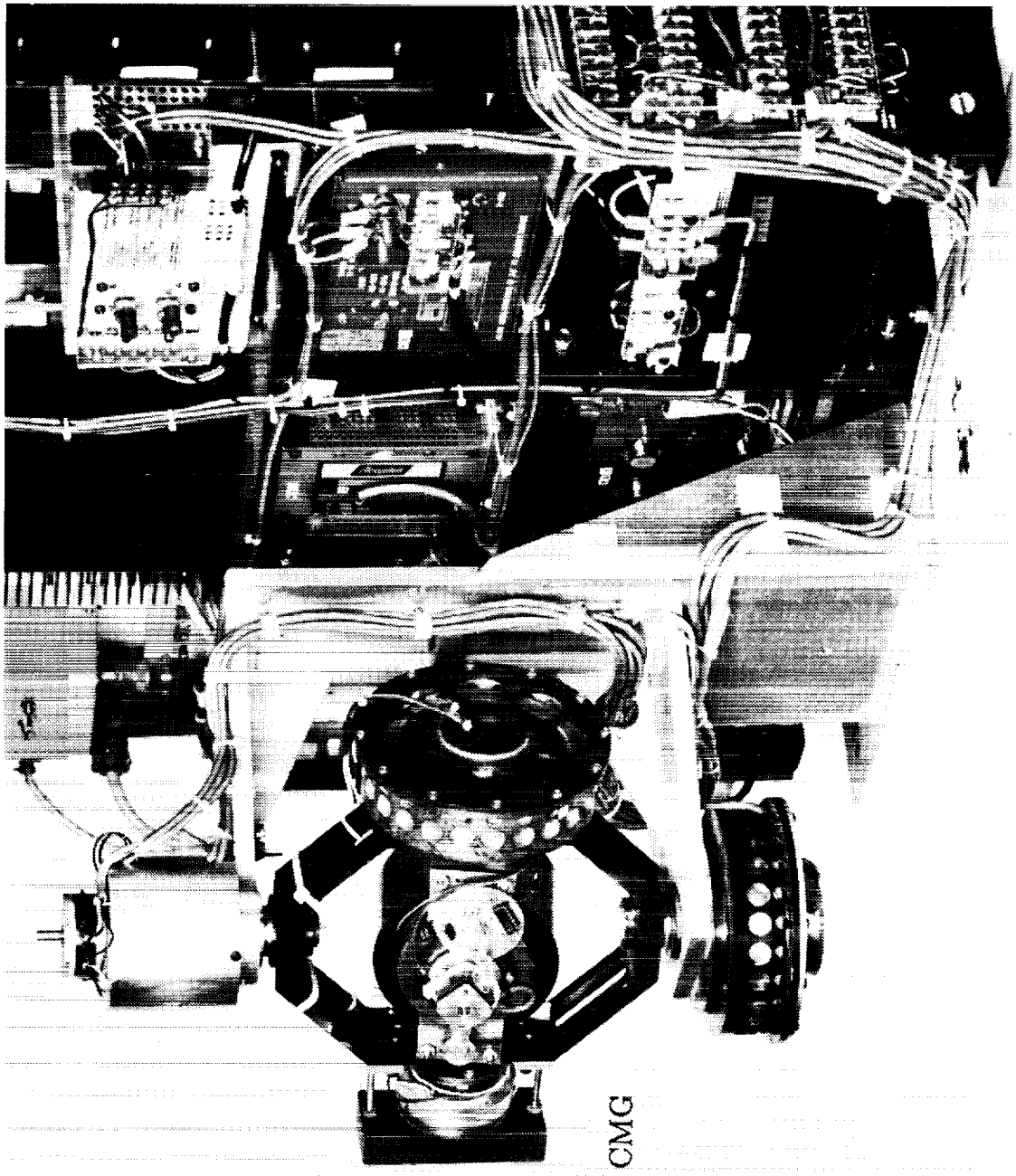
REFLECTOR–MOUNTED THRUSTERS



MAST–MOUNTED REACTION WHEELS AND ACCELEROMETERS



PLATFORM–MOUNTED REAR CONTROL MOMENT GYRO (CMG)



NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CP-10057, Part 2		

4. Title and Subtitle	5. Report Date
5th Annual NASA Spacecraft Control Laboratory Experiment (SCOLE) Workshop	December 1990
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Lawrence W. Taylor, Jr. (Compiler)	
	10. Work Unit No. 506-46-11-01
9. Performing Organization Name and Address	11. Contract or Grant No.
NASA Langley Research Center Hampton, VA 23665	
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Conference Publication
National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

This publication is a collection of papers presented at the Fifth Annual Spacecraft Control Laboratory Experiment (SCOLE) Workshop held at the Hilton Lodge, Lake Arrowhead, California, October 31, 1988. The papers address the modeling, systems identification, and control synthesis for the Spacecraft Control Laboratory Experiment (SCOLE) configuration.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Large Flexible Spacecraft Control Structural Dynamics	Unclassified-Unlimited Subject Category - 18

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclasssified	Unclassified	371	A16

NASA FORM 1626 OCT 86 For sale by the National Technical Information Service, Springfield, Virginia 22161-2171

NASA-Langley, 1990

PRECEDING PAGE BLANK NOT FILMED