NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ACTIVE DAMPING OF VIBRATIONS ON SPACE STATION FREEDOM USING LINEAR QUADRATIC GAUSSIAN CONTROL AND H∞ CONTROL

by

Jacqueline R. McClusky

December 1991

Thesis Advisor: Jeffrey B. Burl

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution is unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION: Naval Postgraduate School; 6b OFFICE SYMBOL (If applicable): EC	7a NAME OF MONITORING ORGANIZATION: Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code): Monterey, CA 93943-5000	7b ADDRESS (City, State, and ZIP Code): Monterey, CA 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION; 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO; PROJECT NO; TASK NO; WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification): ACTIVE DAMPING OF VIBRATIONS ON SPACE STATION FREEDOM USING LINEAR QUADRATIC GAUSSIAN CONTROL AND H∞ CONTROL

12 PERSONAL AUTHOR(S): MCCLUSKY, Jacqueline R.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month Day)	15 PAGE COUNT
Master's Thesis	FROM _____ TO _____	1991 December	100

16 SUPPLEMENTARY NOTATION: The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the US Government.

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	active damping; H∞ control; vibration; linear quadratic Gaussian control

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Active damping of modal oscillation is critical to the success of future versions of Space Station Freedom. Vibratory motion may be induced by external disturbances such as solar and gravity gradient torques, extra vehicular and experimental activity, aerodynamic forces, the earth's magnetic field, and space shuttle docking. Linear proof mass actuators can provide control on the space station to achieve this damping effect. Two control algorithms, Linear Quadratic Gaussian control and H∞ control are applied to a model of Space Station Freedom. The results compare the robustness, stability and performance of the Space Station under the effects of each of the two control algorithms.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL: BURL, Jeffrey B.	22b TELEPHONE (Include Area Code): 408-646-2390; 22c OFFICE SYMBOL: EC/Bl

DD Form 1473, JUN 86 — Previous editions are obsolete — S/N 0102-LF-014-6603

Approved for public release; distribution is unlimited

Active Damping of Vibrations on Space Station Freedom
Using Linear Quadratic Gaussian Control and H∞ Control

Jacqueline R. McClusky
Lieutenant, United States Navy
B.S., United States Naval Academy, 1984

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1991

Author: _____
Jacqueline R. McClusky

Approved by: _____
Thesis Advisor, Jeffrey B. Burl

_____
Second Reader, Robert D. Strum

_____
Michael A. Morgan, Chairman
Department of Electrical and Computer Engineering

ABSTRACT

Active damping of modal oscillation is critical to the success of future versions of Space Station Freedom. Vibratory motion may be induced by external disturbances such as solar and gravity gradient torques, extra vehicular and experimental activity, aerodynamic forces, the earth's magnetic field, and space shuttle docking. Linear proof mass actuators can provide control on the space station to achieve this damping effect. Two control algorithms, Linear Quadratic Gaussian control and H∞ control are applied to a model of Space Station Freedom. The results compare the robustness, stability, and performance of the Space Station under the effects of each of the two control algorithms.

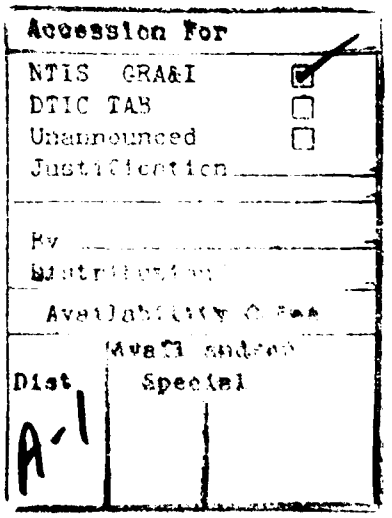

TABLE OF CONTENTS

I. INTRODUCTION.....1

A. OVERVIEW.....1

B. SPACE STATION FREEDOM AND CONTROL SYSTEMS....2

C. VIBRATION SUPPRESSION ON SPACE STATION FREEDOM.....4

D. ORGANIZATION OF THESIS.....5

II. CONTROL SYSTEM DESIGNS.....6

A. CONTROL ALGORITHMS.....10

1. The Linear Quadratic Regulator.....10

2. The Kalman Filter.....12

3. Linear Quadratic Gaussian Control.....14

4. H∞ Control.....16

B. REDUCED ORDER MODELLING.....20

III. ANALYSIS AND SIMULATION.....22

A. FULL ORDER CONTROLLERS.....22

1. Performance Measure.....22

2. Open Loop System.....24

3. Linear Quadratic Gaussian Controller.....25

a. Simulation of the Linear Quadratic Regular.....25

b. Simulation of Kalman Filter.....33

4. H∞ Controller.....38

B. REDUCED ORDER CONTROLLERS.....45

1. Linear Quadratic Gaussian Controller.....45

2. H∞ Controller.....48

C. A COMPARISON OF LQG AND H∞ CONTROLLERS.....49

IV. CONCLUSIONS.....51

A. SUMMARY OF OBSERVATIONS.....51

B. FURTHER RESEARCH AND DEVELOPMENT.....52

APPENDIX A Zeros and Poles of Plant.....53

APPENDIX B Computer Programs.....54

LIST OF REFERENCES.....93

INITIAL DISTRIBUTION LIST.....95

ACKNOWLEDGEMENT

I would like to express my sincere appreciation to the ECE staff, Professor Jeff Burl, and Professor Strum for their untiring devotion and patience. Their assistance and insight were invaluable to me in the completion of my studies.

I would also like to thank my husband, Ken, for being my friend throughout this endeavor.

I. INTRODUCTION

A. OVERVIEW

A long time quest for scientists and engineers has been the development of a permanent space station. This pursuit has resulted in the ongoing development of Space Station Freedom which is intended to be a permanently-manned orbiting base by the end of the century. The identification of the placement of actuators and sensors, diminishing effects of extra vehicular activity and zero gravity on crew members, developing of solar panels for energy, and damping structural vibrations are some of the tasks and obstacles involved in the design of Space Station Freedom. Of particular interest is the control of vibratory motion on the space station. Classical control techniques have been employed in the past to solve the vibration problem. This approach alone is limited due to its inability to address the issues of robustness, sensitivity, and fault tolerance in a multivariable setting. Modern advances in control systems and computer technology have equipped designers with the necessary tools to overcome these limitations.

The present study is aimed at developing a control algorithm for damping structural vibrations in the presence of parameter variations and unmodelled dynamics. In order to achieve this goal, $H\infty$ control and Linear Quadratic Gaussian

control will be evaluated and applied to a model of Space Station Freedom.

B. SPACE STATION FREEDOM AND CONTROL SYSTEMS

Space travel has been envisioned by astronomers and scientist since approximately 300 A.D. [Ref. 1]. Early references to space travel were only stories with no scientific foundation. Later in the 1700's, Galileo first viewed the stars and moons when he envisioned these heavenly bodies as places for man to visit. About 1860, Jules Verne detailed the venture of three crew members who soared in a spacecraft to the moon in *De La Terre a' la Lune (From the Earth to the Moon)*. The ideas of space travel persevered into the 1900's. After World War II, the space station concept and serious proposals for a manned spacecraft originated. Manned spaceflight became a national goal after President Kennedy's pronouncement to send a man to the moon. Hence, the Mercury, Gemini, and Apollo space programs resulted.

The manned lunar landing was a prominent event in the space program. The diversion of attention to the lunar landing missions led to the decline of interest toward building a permanent space station. Later, mandated budgetary constraints imposed by Congress did not improve the situation. In 1982, Space Station Freedom became a national goal [Refs. 2 and 3]. Development of Space Station Freedom was divided into eight phases. Control and stabilization were two phases

that focused on limiting the effects of structural vibratory motion.

The beginnings of control theory has been traced from Huygins in 1673 to Maxwell in 1868 [Ref. 4]. Maxwell was the first to discuss stability of feedback control systems whereas Bode, Routh, and Nyquist later addressed this issue. Feedback control became an important topic at approximately the same time the space concept emerged. Control design was prominent due to the construction of rockets, missiles, and communication systems developed during WWII. This effort undoubtedly impacted the space design process. With the need for space control technology, Kalman, Pontryagin, and Bellman investigated at optimal control and state space modelling [Refs. 5 and 6] while Joseph, Tou and Simon developed compensators that estimated the states of a system in the presence of stochastic noise [Ref. 7].

Most of the systems developed during this time were single-input/single-output (SISO). However, the requirement for controlling systems with multiple inputs and multiple outputs (MIMO) was becoming important. Zames [Ref. 8] introduced a feedback control design which addressed multivariable systems affected by external perturbations. Classical and feedback control techniques in the frequency domain were extended to MIMO systems with the advent of the $H\infty$ (H-infinity) design methods. The $H\infty$ design methodology allows the designer to directly consider the contradictory

requirements of system performance, sensitivity reduction, robustness and disturbance attenuation in multivariable control systems. The H∞ controller, is consequently, a natural choice for suppression of vibrations due to disturbances and unmodelled dynamics aboard Space Station Freedom.

C. VIBRATION SUPPRESSION ON SPACE STATION FREEDOM

Active damping of modal oscillation is critical to the success of future versions of Space Station Freedom [Ref. 9]. The vibratory motion may be induced by external disturbances such as solar and gravity gradient torques, extra vehicular and experimental activity, aerodynamic forces, the earth's magnetic field, and space shuttle docking. Damping this vibratory motion reduces the disruption of onboard experiments and communication and remote sensor pointing errors. Linear proof mass actuators can provide control on the space station to achieve this damping effect [Ref. 10].

Currently, there is little documented research that provides an analysis and comparison of different control algorithms. It is, therefore, the intent of this research to apply two control algorithms, Linear Quadratic Gaussian control and H∞ control to a model of Space Station Freedom. The results will compare the robustness, stability, and performance of Space Station Freedom under the effects of each of the two control algorithms.

D. ORGANIZATION OF THESIS

Linear Quadratic Gaussian control and H∞ control are two algorithms used to solve the vibration problem. Chapter II provides the mathematical model of a space station. The data used in this study are furnished by the McDonnell Douglas Astronautics Company. The two control algorithms are presented in Chapter II. Because of the complexity and high order of the model, reduced-order controllers are also presented in Chapter II. Chapter III examines the application of the two control systems to the model and a comparison of the controllers. Conclusions and recommendations for future study are presented in Chapter IV.

II. CONTROL SYSTEM DESIGNS

The space station exhibits low frequency, lightly damped vibrations due to its size, construction and composition. Figure 1 is a representation of a dual-keel space station provided courtesy of the McDonnell Douglas Astronautics Company.

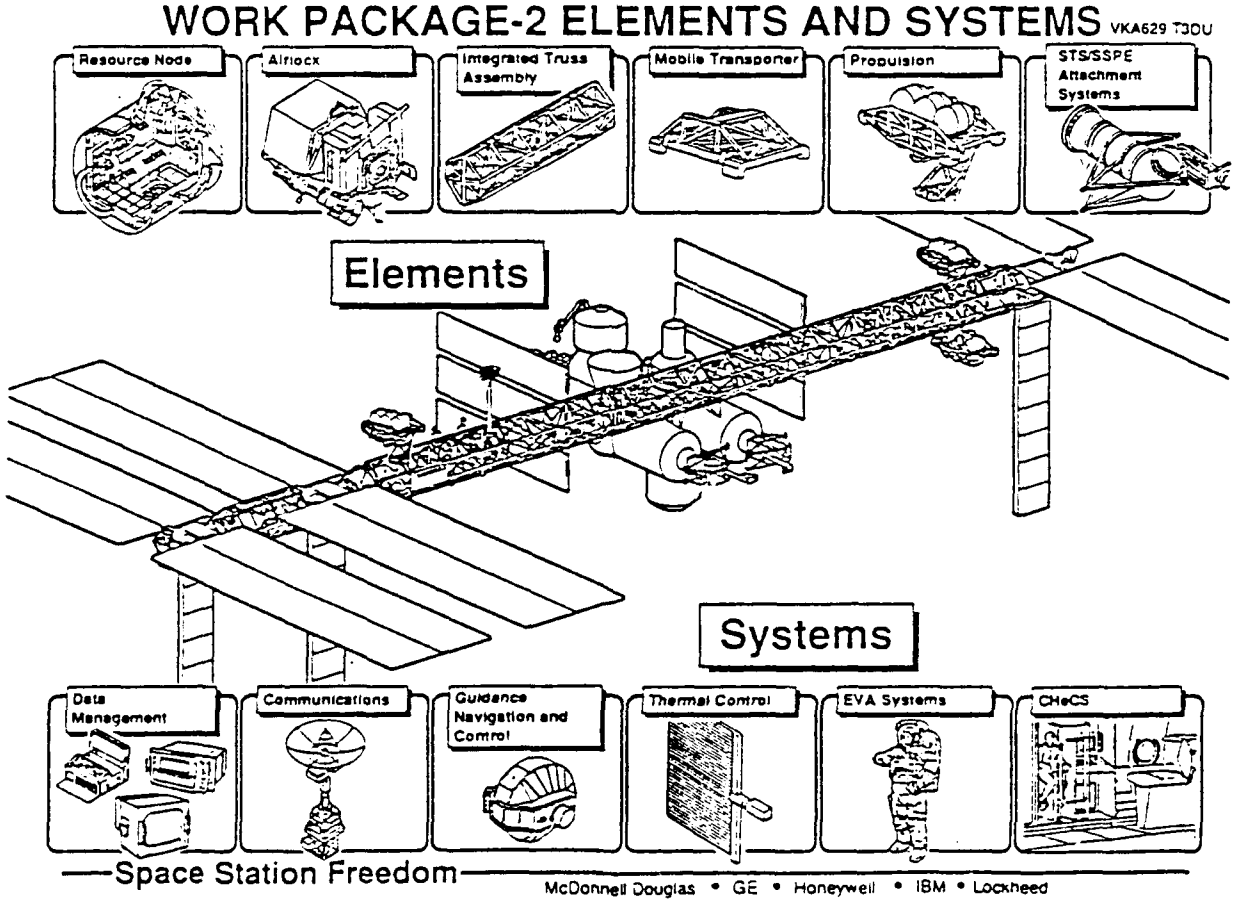

Figure 1 A Mathematical Model of Space Station Freedom

A mathematical model is required to analyze the motion of this space station. The modal equation can be found by applying finite element analysis to the space station dynamics. The motion is then a linear combination of the natural modes. The solution of the modal equation is of the form [Ref. 11]:

$$\mathbf{q}(t) = \sum_{i=1}^{\infty} \Phi_i \, \eta_i(t) \qquad \text{(2-1)}$$

where $\mathbf{q}(t)$ is the displacement of the structure, Φ_i is the modal vector or natural mode shape, and $\eta_i(t)$ is the modal amplitude of the i^{th} mode at time t. The modal amplitudes can be found by solving the uncoupled second-order differential equations:

$$\ddot{\eta}_i(t) + d\,\omega_i\,\dot{\eta}_i + \omega_i^2\,\eta(t) = \Phi_i(p) \cdot \mathbf{f}_p(t) \qquad \text{(2-2)}$$

where d is a scalar structural damping coefficient, ω_i is the i^{th} natural modal frequency and $\mathbf{f}$ is an external forcing function. Solving for the highest order derivative in Equation 2-3 to obtain a state space representation of the modal matrix gives:

$$\begin{aligned} \ddot{\eta}_i(t) &= -d\omega_i^2\dot{\eta}_i(t) + \omega_i\eta(t) \\ &= \Phi_i(p) \cdot \mathbf{f}_p(t) \quad i = 1,2,\ldots 10 \end{aligned} \qquad \text{(2-3)}$$

and let

$$\begin{aligned} x_1(t) &= \eta_i(t) \\ \dot{x}_1(t) &= x_2 = \dot{\eta}_i(t) \\ \dot{x}_2(t) &= -d w_i x_2 - w_i^2 x_1 + \Phi_i(p) \cdot f_i(t) = \ddot{\eta}(t) \end{aligned} \tag{2-4}$$

where

$$\begin{aligned} f_p(t) &= u(t) \\ \Phi_i(p) &= b_i(p) . \end{aligned} \tag{2-5}$$

The matrix state equation is:

$$\begin{bmatrix} \ddot{\eta}_i \\ \dot{\eta}_i \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -w_i^2 & -d w_i \end{bmatrix} \begin{bmatrix} \eta_i \\ \dot{\eta}_i \end{bmatrix} + \begin{bmatrix} 0 \\ b_i(p) \end{bmatrix} u \quad i = 1,2,\ldots 10 \tag{2-6}$$

A measurement, taken at a point p, is given by:

$$\begin{aligned} y(t) &= C\, x(t) \\ &= [\Phi_1(p) \;\; 0 \ldots \Phi_{10}(p) \;\; 0] \begin{bmatrix} \eta_1 \\ \dot{\eta}_1 \\ \cdot \\ \cdot \\ \cdot \\ \dot{\eta}_{10} \end{bmatrix} \end{aligned} \tag{2-7}$$

The data provided 100 modes of vibration and their corresponding natural frequencies [Ref. 12]. In this study, however, only the first ten modes are considered. Tables 1 and 2 present these ten modes and their natural frequencies, respectively. Vibration may occur in the translational or

rotational direction. Furthermore, node 23, the shuttle docking point is analyzed.

TABLE 1
LIST OF TRANSLATIONAL MODAL VECTORS

MODE	NODE 23		
	X	Y	Z
1	2.7006e-02	-2.7171e-13	7.8058e-12
2	-3.2282e-13	2.7006e-02	4.7006e-13
3	1.7084e-12	1.5979e-13	2.7006e-02
4	2.6363e-13	3.2302e-03	3.3864e-03
5	-4.9299e-03	2.1923e-05	2.5267e-02
6	-4.2493e-03	2.0629e-02	1.3326e-04
7	-1.4465e-03	-5.9130e-04	-7.2815e-03
8	6.4308e-03	3.7219e-04	-1.9019e-03
9	-7.0786e-05	3.6509e-04	-1.2529e-04
10	-1.0788e-03	2.6420e-04	2.6420e-04

TABLE 2
NATURAL FREQUENCIES

Modes	Natural Frequencies
1	0.32374935
2	0.34734376
3	0.37886664
4	0.38108557
5	0.39618959
6	0.40531164
7	0.40761414
8	0.41520832
9	0.43176959
10	0.44403511

A. CONTROL ALGORITHMS

There are a number of linear control algorithms available to a designer. The Linear Quadratic (LQ) regulator, Kalman filter, Linear Quadratic Gaussian (LQG) control, and H∞ control are applied and analyzed in this thesis. The Loop Transfer Recovery (LTR) technique is also presented.

1. The Linear Quadratic Regulator

The LQ regulator is an algorithm that results in an optimum controller for a given system. A performance measure is minimized given that the states of the system are fully accessible. Kirk [Ref. 13] states that the objective is to determine the control signals that will cause a process to satisfy the physical constraints and at the same time minimize (or maximize) some performance measure. The selection of the

performance measure is based on the objectives and imagination of the designer.

In addition to the performance measure, the formulation of the LQ regulator requires a mathematical model of the space station. The mathematical model utilizes the linear, time-invariant state equation:

$$\boldsymbol{x}(k+1) = \Phi\, \boldsymbol{x}(k) + \Gamma\, \boldsymbol{u}(k) \qquad \textbf{(2-8)}$$

where Φ is an n by n matrix, Γ an n by r matrix, $\mathbf{x}(k)$ an n-dimensional state vector. $\mathbf{u}(k)$ a r-dimensional control vector which minimizes the performance measure:

$$J = \sum_{k=0}^{N-1} [\boldsymbol{x}^T(k)\ \boldsymbol{Q}\, \boldsymbol{x}(k) + \boldsymbol{u}^T(k)\ \boldsymbol{R}\, \boldsymbol{u}(k)] \qquad \textbf{(2-9)}$$

in which **Q** is the square, symmetric state weighting matrix and **R** is the control weighting matrix. To guarantee a unique solution, **Q** must be positive semi-definite and **R** positive definite. The **Q** matrix provides a penalty for deviation from equilibrium while **R** provides a penalty for using control. Consequently, increasing **Q** increases the penalty on the state vector, while increasing **R** increase the cost of control applied to the system. The performance function is minimized when the linear optimal control law, $\mathbf{u}(k) = -\mathbf{K}\mathbf{x}(k)$ is implemented. The optimal gain matrix **K** is the solution of the

algebraic Ricatti equation that drives the states to zero. Figure 2 is a block diagram of the system:

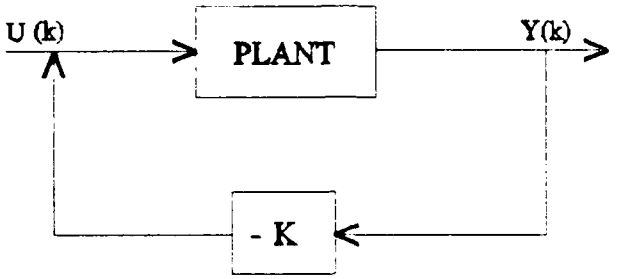

Figure 2 The Linear Quadratic Regulator

The LQ regulator has good stability and robustness properties, but assumes that no noise exists in the system and all states are available for measurement and feedback. This assumption poses a major drawback for the LQ regulator and may be overcome by applying the Kalman filter.

2. The Kalman Filter

The Kalman Filter is an observer that provides optimal state estimation in the presence of noise. It is a recursive algorithm that provides the "best" estimate of the states of a linear system given only the input and output of the system. The linear, time-invariant system whose state is to be estimated is:

$$\begin{aligned} \boldsymbol{x}(k+1) &= \Phi\, \boldsymbol{x}(k) + \Gamma_1\, \boldsymbol{u}(k) + \Gamma_2\, \boldsymbol{w}(k) \\ \boldsymbol{y}(k) &= \boldsymbol{C}\, \boldsymbol{x}(k) + \boldsymbol{v}(k) \end{aligned} \qquad \textbf{(2-10)}$$

where Φ is an n by n matrix, Γ_1 and Γ_2 are n by r matrices, $\mathbf{u}(k)$ is a known input, $\mathbf{x}(k)$ is a state matrix, $\mathbf{w}(k)$ is random plant driving noise and $\mathbf{v}(k)$ is random measurement noise. Both $\mathbf{w}(k)$ and $\mathbf{v}(k)$ are white noise vectors with zero mean, i.e., $E[w] = E[v] = 0$. The covariance matrix for the plant driving noise is $E[\mathbf{w}\mathbf{w}^T] = \mathbf{W}$ and the measurement noise covariance matrix is $E[\mathbf{v}\mathbf{v}^T] = \mathbf{V}$. The plant driving noise and measurement noise are independent and uncorrelated.

The optimal state estimate:

$$\hat{\boldsymbol{x}}(k+1|k+1) = \hat{\boldsymbol{x}}(k+1|k) + \boldsymbol{G}[\boldsymbol{y}(k+1) - \hat{\boldsymbol{y}}(k+1|k)] \qquad \textbf{(2-11)}$$

where $\hat{\mathbf{x}}(k+1|k)$ denotes the estimate of the state at time k+1 given measurements through time k and $\hat{\mathbf{y}}(k+1|k)$ is the estimate of the measurement at time k+1 given measurement through time k. A summary of the other equations that represent a steady-state Kalman filter are:

$$\begin{aligned} \hat{\boldsymbol{x}}(k+1|k) &= \Phi \boldsymbol{x}'(k|k) + \Gamma_1\, \boldsymbol{u}(k) + \Gamma_2\, \boldsymbol{w}(k) \\ \hat{\boldsymbol{y}}(k+1|k) &= \boldsymbol{C}\hat{\boldsymbol{x}}(k+1|k) \end{aligned} \qquad \textbf{(2-12)}$$

A general block diagram of a Kalman filter is given in Figure 3 [Ref. 14]:

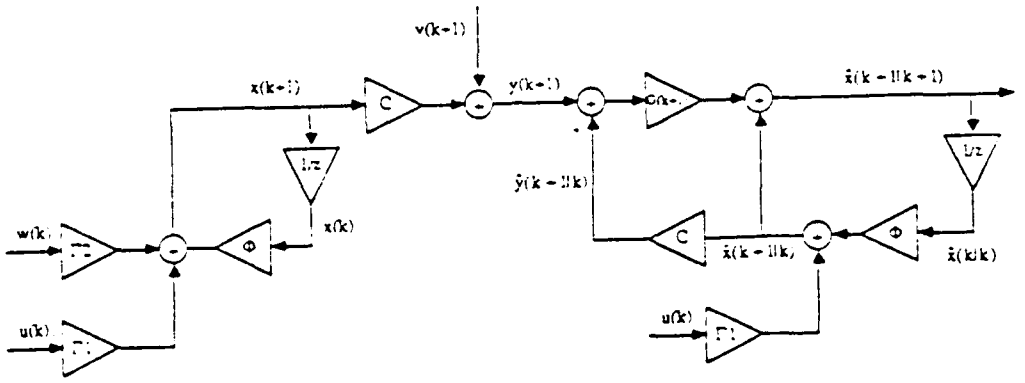

Figure 3 The Kalman Filter

The Kalman filter is an optimum process provided the stochastic noise processes are white and Gaussian [Ref. 12].

3. Linear Quadratic Gaussian Control

The Linear Quadratic Gaussian (LQG) control is a combination of the LQ regulator and the Kalman filter designed in separate stages. The results derived separately for the optimal control and estimation problem are still valid. A typical block diagram of the system is depicted in Figure 4.

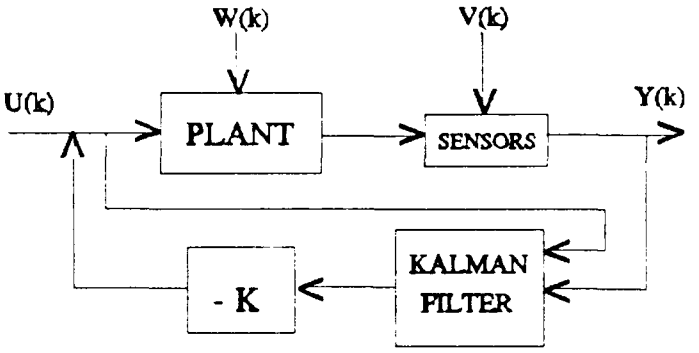

Figure 4 The Linear Quadratic Gaussian Controller

The LQG control system utilizes the full-state feedback of the LQ regulator applied to the estimated states from the Kalman filter. The robustness properties of the feedback system diminishes when the Kalman filter is implemented to estimate the states [Ref. 15]. Two criteria that can be used to assess the robustness of the system are the phase and gain margins. The gain margin is the amount of additional gain that can be added to the system without causing instability. The phase margin is the amount of additional phase delay in the plant and/or compensator that can be added before generating instability. In order to improve the robustness as defined by the phase and gain margins, loop transfer recovery techniques may be applied [Ref. 16].

Loop transfer recovery (LTR) is a technique to recover the robustness properties lost due to using state estimates acquired by the Kalman filter. In order to apply this

algorithm, the system must be causal and have minimum phase, i.e., all the zeros strictly lie inside the unit circle. Fictitious noise, describing the unmodelled dynamics, is implanted at the plant input thereby adjusting the estimator design. Consequently, the LQG controller becomes more robust to gain and phase changes at the plant input as the noise increases [Ref. 17]. However, robustness is inversely related to the performance. The system becomes suboptimal with the addition of the fictitious noise. This forces the designer to form a compromise between robustness or performance.

4. H∞ Control

Since no design model can approximate a physical plant perfectly, modelling errors will always exist. The LQ regulator, LQG control, and the Kalman filter possess reasonable tolerance to modelling errors. However, these control techniques lack the provisions to directly design for robustness to unmodelled dynamics. The H∞ controller is an optimal method that is capable of addressing these factors using frequency domain techniques. The H∞ performance function puts limits on the maximum value of the disturbance frequency response.

The H∞ controller is similar in structure to that found in the classical control design as illustrated in Figure 5. G(s) is the open loop transfer function, K(s) is the controller, U is the known input, and Y is the error.

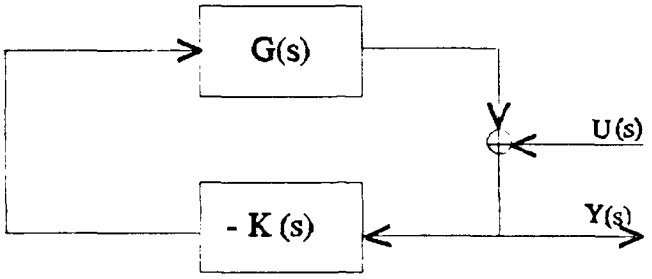

Figure 5 Closed Loop System with H∞ Controller

The objective of H∞ is to achieve the desired robustness and performance response in the presence of unmodelled dynamics. An analysis of the response of the multivariable system in Figure 5 may be ascertained by an inspection of the closed loop transfer function matrix singular values [Ref. 18] which will be denoted as $\sigma[\cdot]$. H∞ employs these singular values to generate the performance measure. The performance measure is the maximum singular value of the closed loop transfer function over a frequency range, i.e., minimizing the H∞ norm. The singular values of the closed loop system can be used to quantify its stability margins [Ref 18]. The conceptual H∞ design procedure consists of four basic operations which are described below.

First, the design problem is formulated as a "standard problem" as depicted in Figure 5 where the objective is to design a controller such that the closed loop system is

internally stable. Minimizing the H∞ norm of the transfer function from the disturbance to the output and the measurement noise to the output, is also an objective. Secondly, an analysis to determine the sensitivity of the system is required. The sensitivity, S(s), is a measure of how much the closed loop transfer function changes with a small change in G(s) and is defined as $[I + G(s)K(s)]^{-1} = Y(s)/U_1(s)$. The complementary sensitivity, T(s) = $[I + G(s)K(s)]^{-1} G(s)K(s) = - Y(s)/U_2(s)$, where $U_1(s)$ and $U_2(s)$ are the disturbance and measurement noises, respectively. A graphical representation is illustrated in Figure 6.

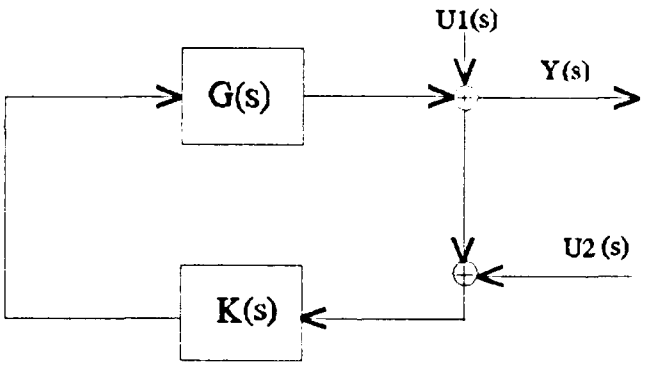

Figure 6 Closed Loop System with Disturbance and Measurement Noise

The sensitivity determines the disturbance attenuation. This attenuation performance specification may be written in the real frequency domain as $\bar{\sigma}(S(j\omega)) \leq |W_1^{-1}(j\omega)|$ where $|W_1^{-1}(j\omega)|$ is the desired disturbance attenuation or performance factor and $\bar{\sigma}$ denotes the largest singular value. Errors from

multiplicative perturbations may affect the robustness whereas the specification $\bar{\sigma}(T(j\omega)) \leq |W_3^{-1}(j\omega)|$ where $|W_3^{-1}(j\omega)|$ limits the effects of measurement noise and provides robustness in the presence of these perturbations. Figure 7 depicts the closed loop system with added weighting functions, $W_1(s)$ and $W_3(s)$.

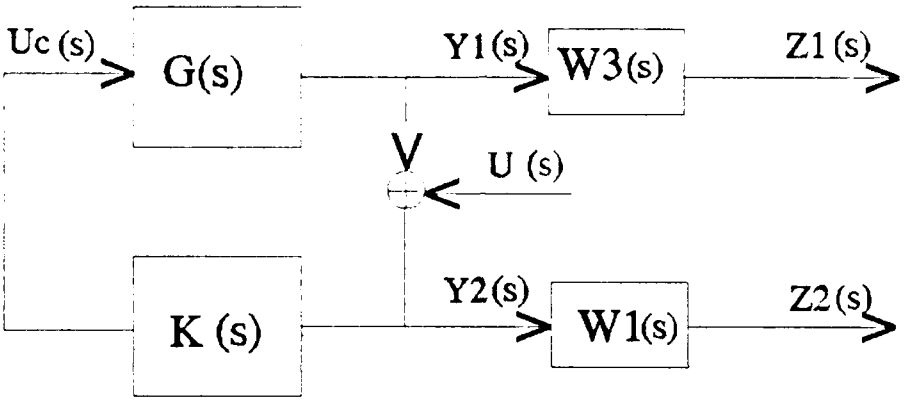

Figure 7 Augmented System with Weighting Functions

The input/output behavior of the system is given by:

$$\begin{aligned} y(s) &= -G(s)K(s)\,[I + G(s)K(s)]^{-1}\, m(s) + [I + G(s)K(s)]^{-1}\, d(s) \\ &= -T(s)\, m(s) + S(s)\, d(s) \end{aligned} \qquad \textbf{(2-13)}$$

Third, an augmented system containing the plant and the weighting functions are produced. This augmented system can sometimes be of high-dimension and reduced order modelling

would be warranted (to be discussed in section II.C).

Fourth, the controller is designed for the system. A controller is selected that stabilizes the system and has an H∞ norm less than one for the closed loop system. This restriction on the H∞ norm guarantees that the sensitivity and complementary sensitivity meet the given specifications. The resulting closed loop system is illustrated in Figure 8.

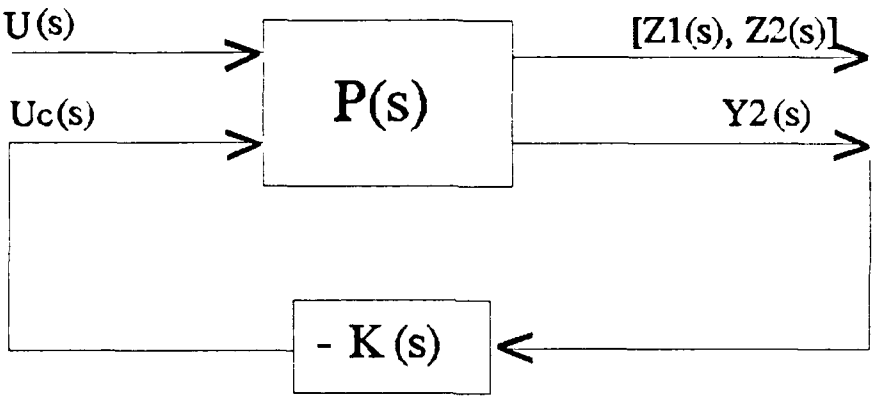

Figure 8 The H ∞ Controller

B. REDUCED ORDER MODELLING

Most mathematical models of space structures are of high order which requires a large computational effort, CPU time, and complex hardware. An alternative is to reduce the order of the model, thereby alleviating some of the complexities of

the design problem. The control design is then synthesized using the reduced-order model. The reduced order design methods, however, do not always produce feedback designs that remain stable in the presence of unmodelled dynamics. Three approaches to model reduction are (1) approximation of the plant with a low order model, (2) balance realizations by organization of state by order of controllability and by (3) truncation of frequencies. There is no simple guideline to determine which model reduction technique will produce the best result or what degree of approximation is necessary to be applied to this model. The reduced order controller for this thesis, however, will be obtained by truncation of frequencies.

III. ANALYSIS AND SIMULATION

This chapter is concerned with the design of the LQG and H∞ controllers for the model discussed in Chapter II. The response of the model due to external forces when both control algorithms are applied is also presented. Two techniques, full and reduced order LQG and H∞ controllers, were employed for analysis and simulations. Finally, a comparison of both control algorithms is provided.

A. FULL ORDER CONTROLLERS

1. Performance Measure

The objective for implementing the LQG and H∞ control is to find a stable controller that minimizes the performance measure. The total vibrational energy plus a control energy term represent the performance measure. The potential and kinetic energy equations written in terms of the modal amplitudes and modal velocities are:

$$P.E.(k) = \frac{1}{2}\sum_{i=1}^{10} w_i^2 \eta_i^2(k) \qquad (3.1)$$

and

$$K.E.(k) = \frac{1}{2}\sum_{i=1}^{10} \dot{\eta}_i(k)\,. \qquad (3.2)$$

The total energy consists of the potential and kinetic energies of the structure at any point in time and can be written in terms of modal state vectors as:

$$\begin{aligned} T.E.(k) &= P.E.(k) + K.E.(k) \\ &= \frac{1}{2}\sum_{i=1}^{10} [\eta_i(k) \;\; \dot{\eta}_i(k)] \; Q_i \begin{bmatrix} \eta_i(k) \\ \dot{\eta}_i(k) \end{bmatrix} \end{aligned} \qquad (3.3)$$

where Q_i is the state weighting matrix defined as:

$$Q_i = \begin{bmatrix} \omega_i^2 & 0 \\ 0 & 1 \end{bmatrix}. \qquad (3.4)$$

Adding the control energy, the performance measure Equation 2.10 may be written as:

$$J = \sum_{i=1}^{10} \boldsymbol{x}_i^T Q \boldsymbol{x}_i + \boldsymbol{u}^T R \boldsymbol{u} \qquad (3.5)$$

where R is a positive definite matrix.

The performance measure of the system may be determined from the system response to a disturbance. The disturbance of interest is an impulse represented as $\delta[\cdot]$, called the delta or Dirac function.

The impulse can be used to generate the initial conditions for the system which can be found following the derivation in Reference 19. The state equation with the impulse input is given as:

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + \boldsymbol{B}\delta(t). \qquad (3.6)$$

Integrating both sides of Equation 3.6 using $-\epsilon$ and ϵ as the lower and upper limits respectively:

$$\boldsymbol{x}(\epsilon) - \boldsymbol{x}(-\epsilon) = \boldsymbol{B} + \int_{\epsilon}^{-\epsilon} A\boldsymbol{x}(t)\,dt. \qquad (3.7)$$

Taking the limit as ϵ approaches 0 to obtain $x(0^+)$, where 0^+ indicate the initial condition an arbitrarily short time after the impulse occurs, yields:

$$x(0^+) = \mathbf{B}. \qquad (3.8)$$

2. Open Loop System

The open loop system is analyzed to assist in understanding the closed loop system. The stability of the open loop system is determined from the eigenvalues of the plant matrix. Appendix A lists these negative, complex conjugate poles. The Bode plot of the plant also illustrates the stability of the open loop system as depicted in Figure 9. The zeros of the plant are also listed in Appendix A. Note that the plant is a minimum phase system.

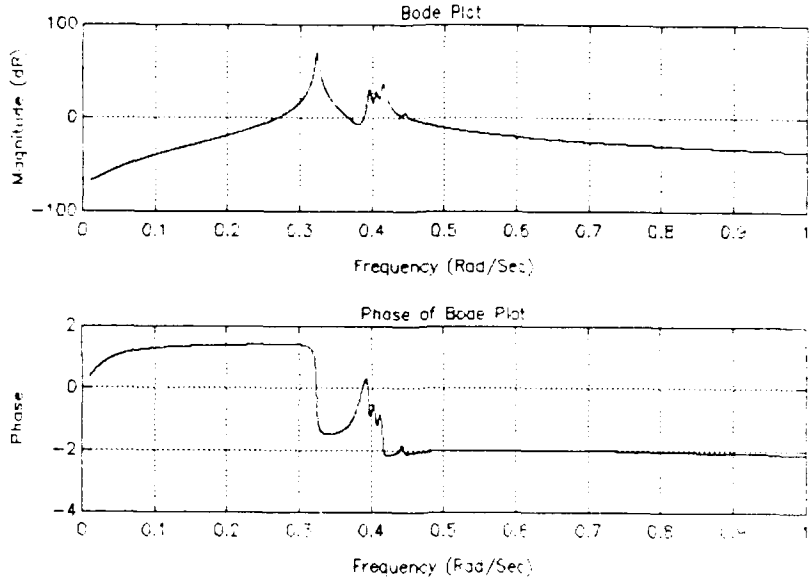

Figure 9 Open Loop Plant Bode Plot

3. Linear Quadratic Gaussian Controller

a. Simulation of the Linear Quadratic Regulator

The LQG controller is based on the separation principle in which the LQ regulator and the Kalman filter are constructed separately. The steady-state LQG controller utilizes two gain matrices by Equation 3.9

$$\begin{bmatrix} \boldsymbol{x}(k+1) \\ \hat{\boldsymbol{x}}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & 0 \\ GC & F-\Gamma K \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(k) \\ \hat{\boldsymbol{x}}(k) \end{bmatrix} + \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} u(k)$$
$$u_{out}(k) = [0 \;\; K] \begin{bmatrix} \boldsymbol{x}(k) \\ \hat{\boldsymbol{x}}(k) \end{bmatrix} \qquad (3.9)$$

where **G** and **K** are the gain matrix obtained from the Kalman filter from the LQ regulator.

A simulation of the model with a unit impulse input and no control is first examined. This provides a reference to refer to when a controller is later added to the

system. The simulation program begins with an interactive phase. This allows the user to select the values of certain parameters that are not fixed by either the program or input data. The parameters that may be varied are:

- node location of applied disturbance
- axial direction of the disturbance
- sampling time
- simulation time
- damping factor

The model was simulated for 200 seconds and a sampling time of one second which was based on the period of the highest sampling frequency of the system. The sampling time was approximately ten times faster than the highest natural frequency, resulting in a minimal amount of aliasing. A damping factor of 0.1 was used in order to yield a lightly damped structure and mode 1 in the horizontal translational direction was analyzed. The five parameters mentioned above were kept constant throughout the simulation.

The transient response of both mode 1 and the total displacement due to an impulse input is shown in Figure 10. The system settles in approximately 800 seconds when simulated without control. The ten natural modes are nearly equal in frequency. As a result, a combination in which 2 modes reinforce each other and cancellation may then result. This is called the beat phenomenon.

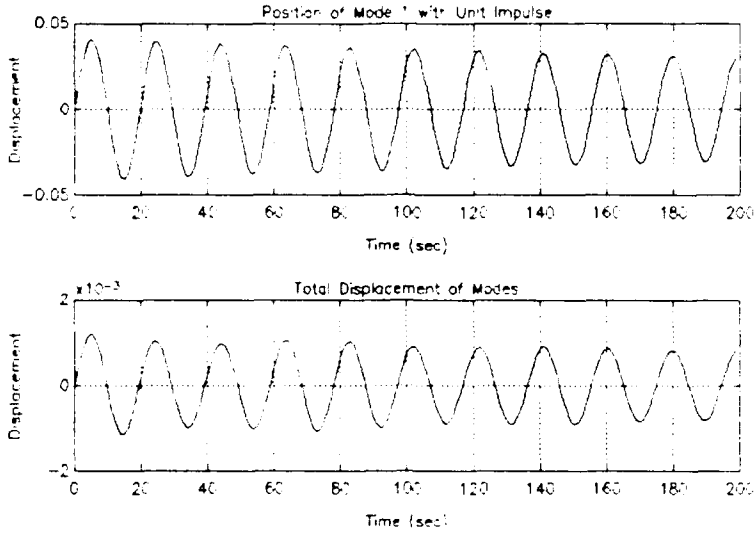

Figure 10 Modal Position and Displacement with No Control

Meirovitch [Ref. 20] states that any motion of the system can be regarded at any time as a superposition of the natural modes each multiplied by some constant. The contribution of each mode is represented by the input coupling of each vibrational mode as shown in Figure 11. It can be seen that modes 2 through 4 do not significantly contribute to the response of the system.

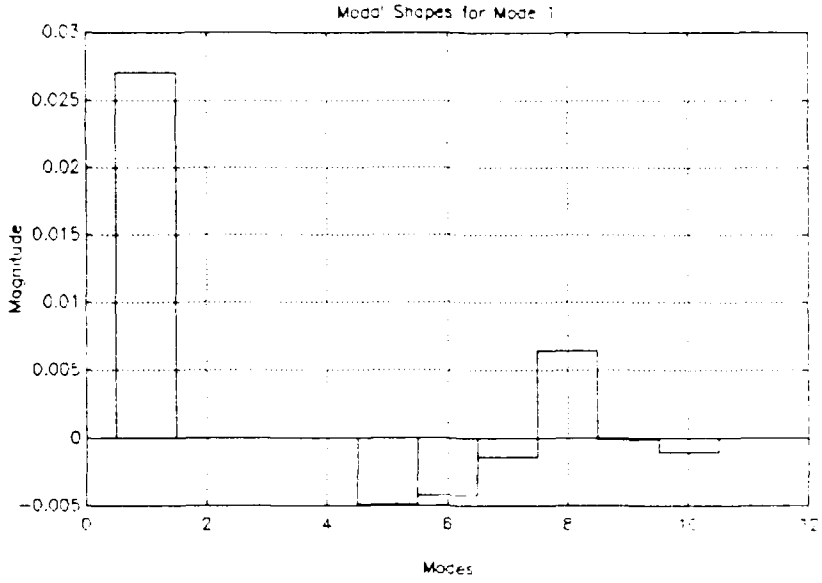

Figure 11 Input Coupling for All Modes

The total energy and energy per mode dissipated by the system is shown in Figure 12. Although the system response deviates from the equilibrium position, the total energy dissipated by the structure is minimal. As expected, modes 2 through 4 have virtually no energy. This is because they are uncoupled to the input.

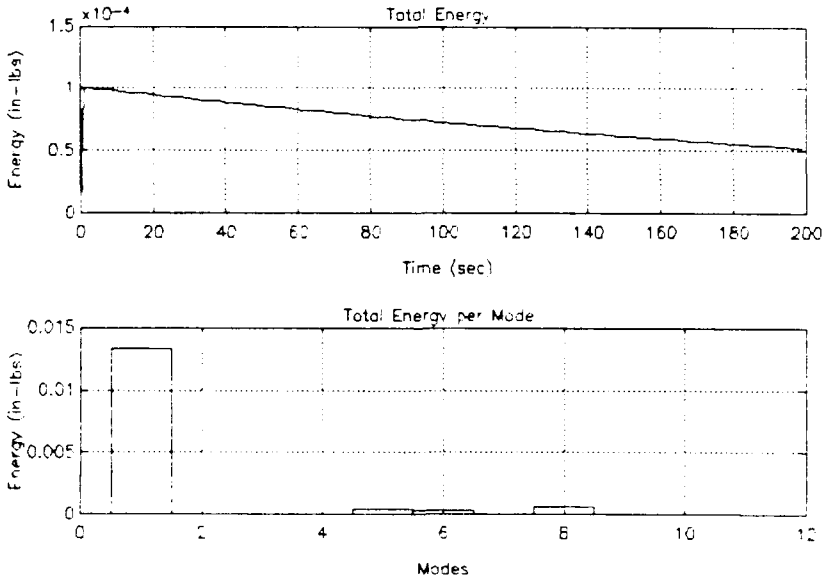

Figure 12 Total Energy and Energy Per Mode Dissipated

The second simulation was performed with the LQ regulator control applied to the system. The design of the LQ regulator for the LQG controller utilized the performance measure in Equation 3.5 with R equal to 1. The stability of the system is crucial to the evaluation of the system performance and was determined by evaluating the number of encirclements of the point (0,-1) which represents the number of poles with positive real parts. This evaluation is called the Nyquist criterion. The Nyquist criterion is useful because it directly displays how a change in gain affects stability as depicted in Figure 13. The second plot in Figure 13 is an enlargement of the real and imaginary axes from -2 to 2. Table 3 lists the matrix gain values, **K**.

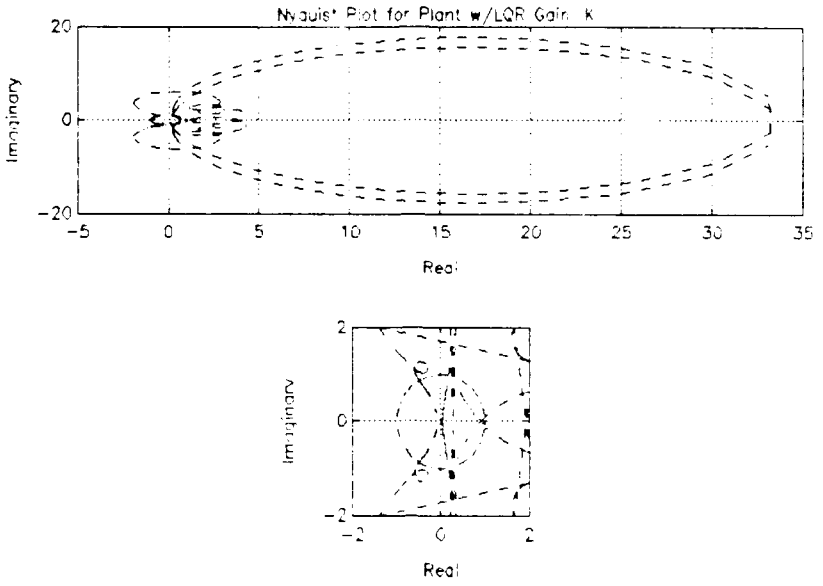

Figure 13 Nyquist Plot with LQ Regulator Gain, K

Table 3 Gain Values

Mode 1

-7.8672E-02	1.2409E+00
-8.0024E-12	-6.3944E-11
-1.0611E-11	3.9083E-10
-2.3457E-12	5.9271E-11
4.8464E-02	-7.2182E-01
7.2167E-03	-6.1610E-01
-2.8356E-03	-2.1306E-01
6.6404E-02	8.1423E-01
-1.3024E-03	-1.4196E-02

The LQ regulator damps the oscillatory motion with an impulse input as illustrated in Figure 14. The transient response settles in approximately 140 seconds.

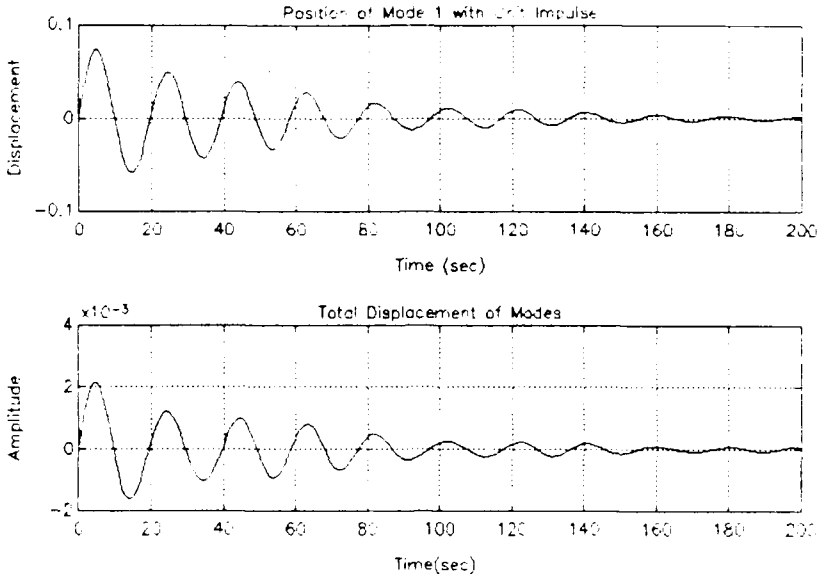

Figure 14 Modal Position with LQ Regulator for Mode 1

Figure 15 illustrates the total energy and energy per mode of the system controlled by the LQ regulator. Using full state feedback, the energy is dissipated rapidly compared to Figure 12 (without control). The system suppresses the vibratory motion rapidly and the higher frequency modes damp out faster with very little energy lost.

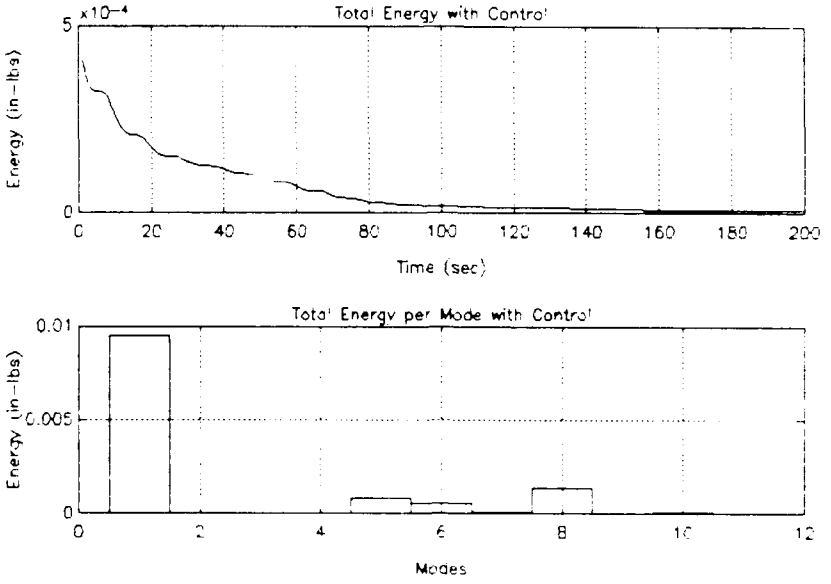

Figure 15 Total Energy Dissipated with LQ Regulator

Figure 16 illustrates the amount of control energy used to suppress the oscillatory motion. Most of the initial control energy was used for mode 1 and the residual for higher frequency modes. It was observed during simulation that R was inversely proportional to the control energy and directly proportional to the settling time.

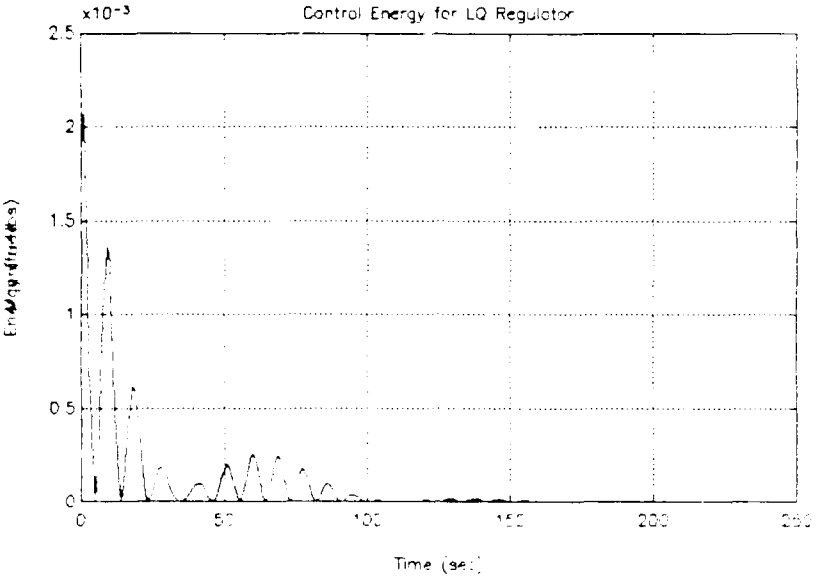

Figure 16 Control Energy for LQ Regulator

The LQ regulator provides relatively good results but requires perfect measurements of the state. Noise is inherent in any system and the Kalman filter will be required to estimate these states.

b. Simulation of Kalman Filter

The addition of the Kalman filter completes the formulation of the LQG controller as depicted in Figure 4. The parameters to be determined for the Kalman filter are the scalar plant W and measurement V covariance matrices. This was a difficult process. The sensor or measurement noise covariance matrix value was determined by varying its value and examining the system response. Several values for W were attempted with the model. Initially, $W=800^{2/3}/40$ and $V=0.005$ were used. The value for W was too high and required adjustment. This indicated that the Kalman filter was

converging much faster than was reasonable. Consequently, the measurement noise was increased and W was lowered which yielded better results. It was determined that a value for V = 0.005 and W = 40 provided the best response for this system as shown in Figure 17.

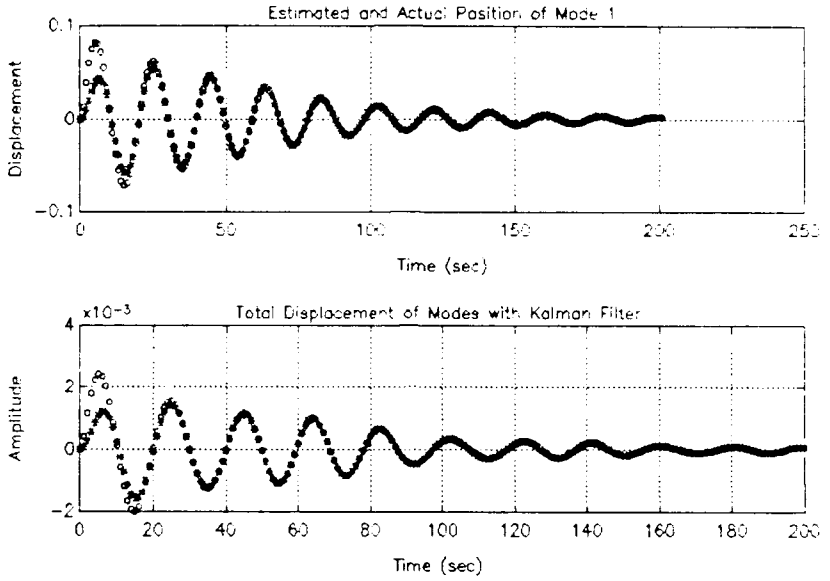

Figure 17 Modal Position and Total Displacement with Kalman Filter

Figure 18 depicts the control input with the Kalman filter. A comparable control input to that used in the LQ regulator was required to suppress the vibrations.

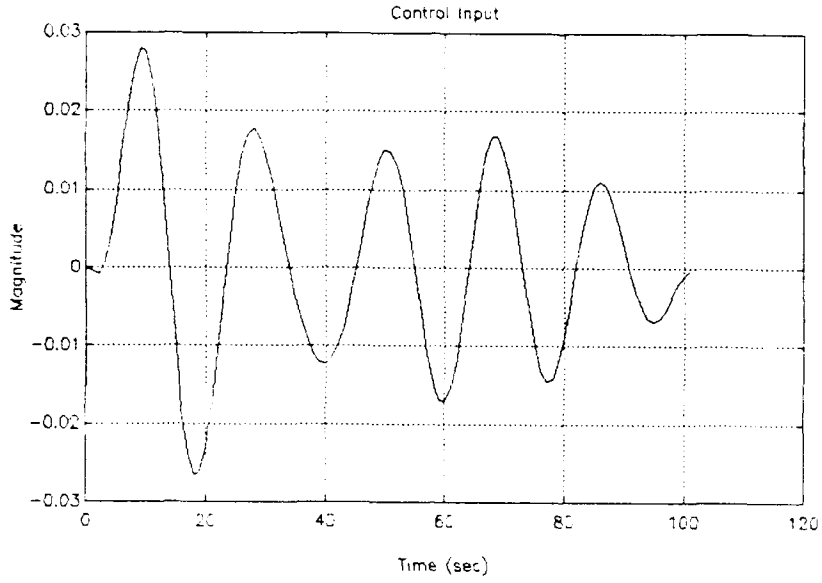

Figure 18 Control Input with Kalman Filter

Figure 19 illustrates that the energy dissipated by the Kalman filter is essentially the same as that of the LQ regulator (Figure 15).

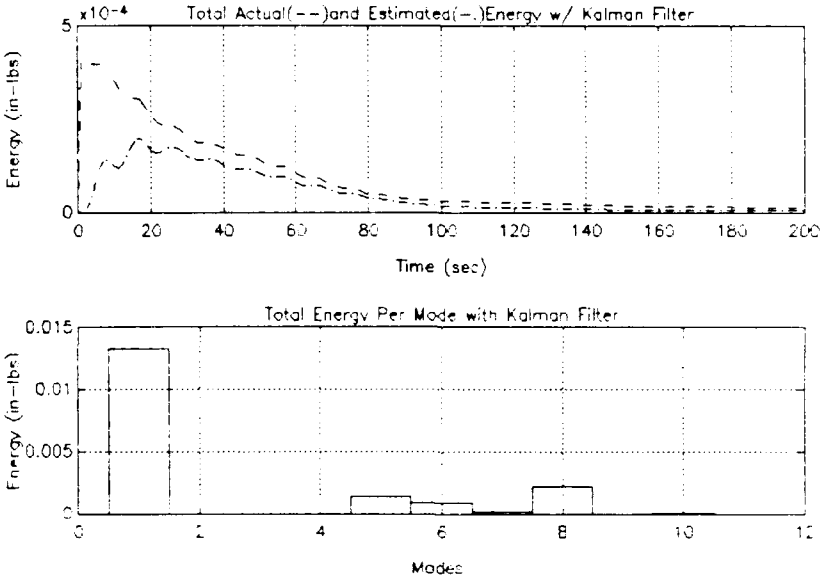

Figure 19 Energy Dissipated by Kalman Filter

Less control energy was required early in the response with the Kalman filter to control the vibratory motion since the estimates have not yet converged to the states. Figure 20 shows the control energy required by the LQG controller.

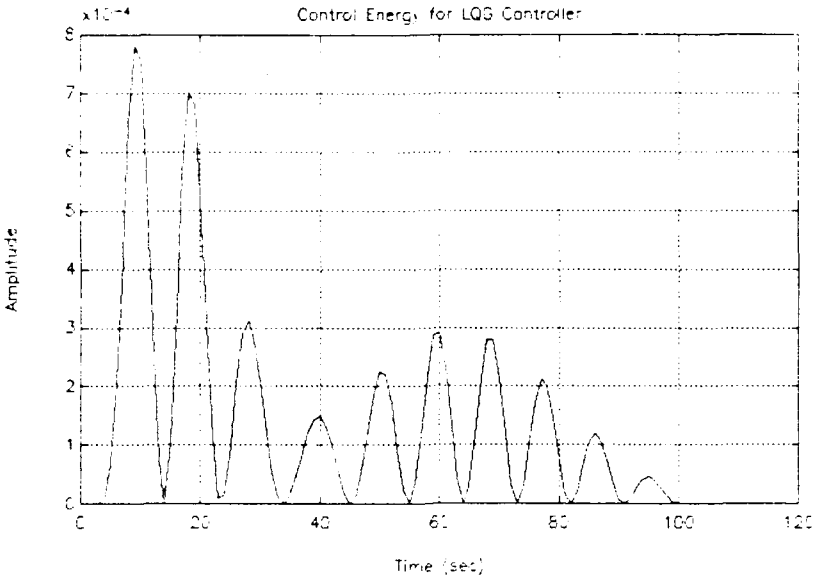

Figure 20 Control Energy with LQG Controller

The LQG controller not only guarantees the stability of the closed loop system, but also minimizes the performance measure. With the Kalman filter and plant input noise added, an approximation of the states yielded suitable performance. Although the performance of the system is relatively good, it may no longer be robust to perturbations in the plant.

To increase the robustness of the system, loop transfer recovery was applied. As a result, the phase and gain margins were improved to the values indicated in Figure 21. The second plot in Figure 21 is an enlargement of the real and imaginary axes from -2 to 2. A compromise was made in terms of the robustness and performance of the system.

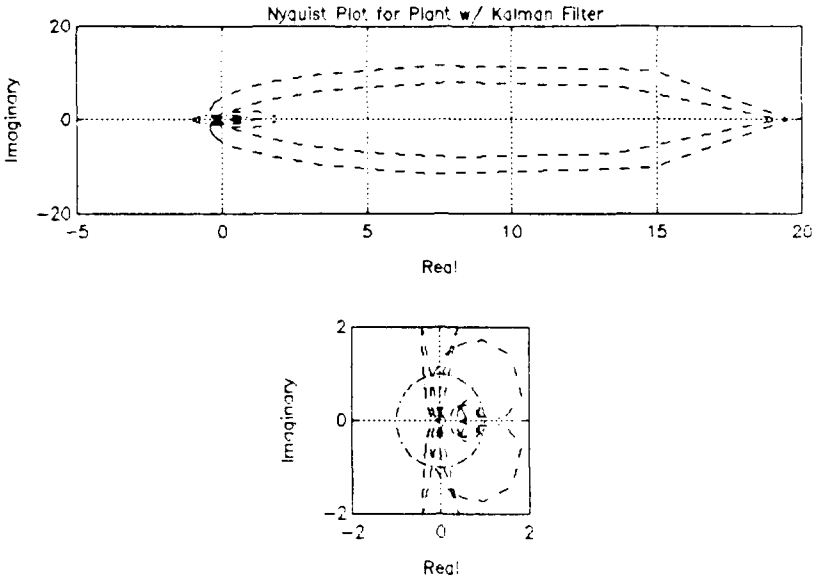

Figure 21 Nyquist Plot for Plant with Kalman Filter

Although the LQG controller is the optimal controller for the modelled noise disturbances, the H∞ design provides a controller that attenuates disturbances in a specified frequency range and can address robustness directly.

4. H∞ Controller

The design of the H∞ controller is based on the desired frequency response characteristics. Dependent upon this configuration is the selection of performance and robustness specifications by the designer. The determination of the specifications required yields the necessary weighting functions. As discussed in section II.B.4, the H∞ controller design is analyzed in four steps. As a reminder, the four steps are to formulate a problem with a stable feedback compensator, determine the sensitivity of the system, augment

the plant with weighting functions, and apply H∞ controller design to the system.

The design of the H∞ controller is based on the space structure example in Reference 21. In selecting the weighting functions, the 0 dB crossover frequency of $W_1(s)$ must be sufficiently below the 0 dB crossover frequency of $W_3(s)$. This is to ensure that a solution exists. W_1 is selected as a second-order system described by:

$$W_1(s) = \frac{\gamma\ 100(1 + \frac{s}{500})^2}{(1 + \frac{s}{100})^2} \qquad \textbf{(3.10)}$$

where γ is a design parameter. $W_1(s)$ is the sensitivity specification where an attenuation of 100:1 to 10 rad/sec (16 hertz) is required. The robustness specification has a closed loop bandwidth 100 rad/sec (300 hertz). $W_3(s)$ is selected as a derivative function of the first order:

$$W_3(s) = \frac{s}{200} \qquad \textbf{(3.11)}$$

Figure 22 illustrates that the requirement of the crossover frequencies is met.

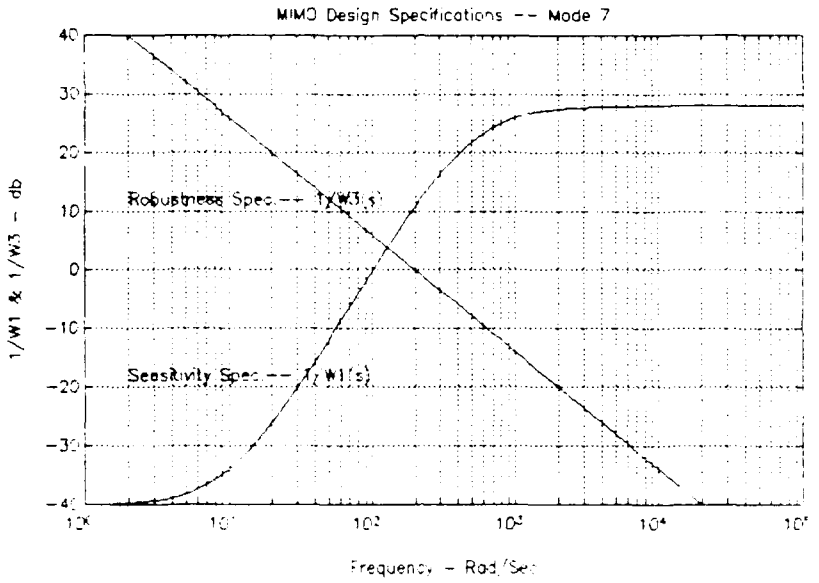

Figure 22 Robustness and Sensitivity Specification

The next step in this design process was to augment the plant matrix to generate Figure 7. The maximum singular values of the sensitivity and complementary sensitivity functions must be less than one where the closed loop system is of the form:

$$\overline{\sigma}(jw) = \overline{\sigma}\begin{bmatrix} W_1(jw)\ S(jw) \\ W_3(jw)\ T(jw)) \end{bmatrix} \leq 1. \qquad \textbf{(3.11)}$$

Once the weighting functions are added to the system, the H∞ controller generated acp, bcp, ccp, and dcp in state space form:

$$\begin{aligned} \dot{\boldsymbol{x}}_c &= acp\ \boldsymbol{x}_c + bcp\ \boldsymbol{y} \\ \boldsymbol{u}_c &= ccp\ \boldsymbol{x}_c + dcp\ \boldsymbol{y} \end{aligned} \qquad \textbf{(3.12)}$$

where $\mathbf{x}_c$ is the controller state vector, $\mathbf{y}$ is the output, and $\mathbf{u}_c$ is the control input. Combining the controller equations and the plant equations yields the closed loop system in Equation 3.13:

$$\begin{bmatrix} \dot{x} \\ \dot{x}_c \end{bmatrix} = \begin{bmatrix} A + C\,B\,dcp & B\,ccp \\ bcp\,C & acp \end{bmatrix} \begin{bmatrix} x \\ x_c \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} w. \qquad (3.13)$$

The inverse of both weighting functions are the bounds on the frequency response of the sensitivity and complementary sensitivity functions of the closed loop system. A plot of both is provided in Figures 23 and 24. Gamma equal to 1.5 was used. The requirement to limit the sensitivity to not exceed the W_1^{-1} was met.

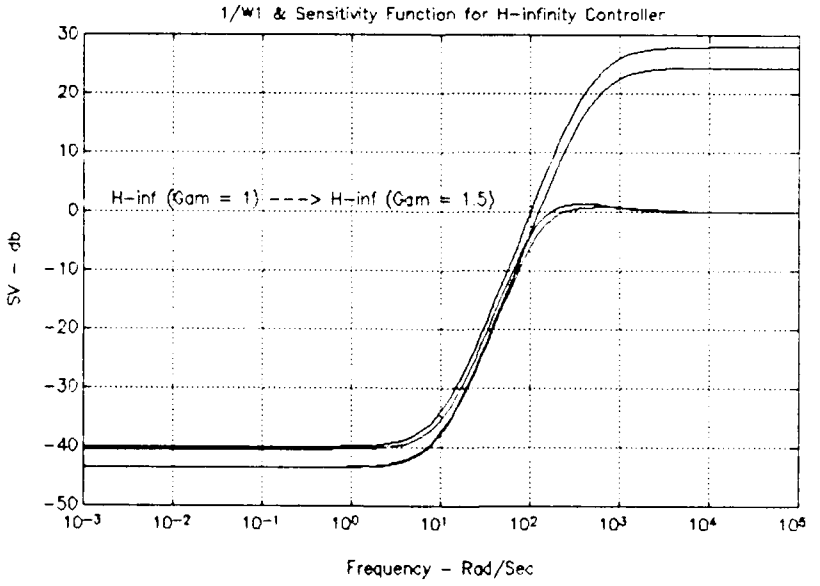

Figure 23 1/W1 & Sensitivity Function for H∞ Controller

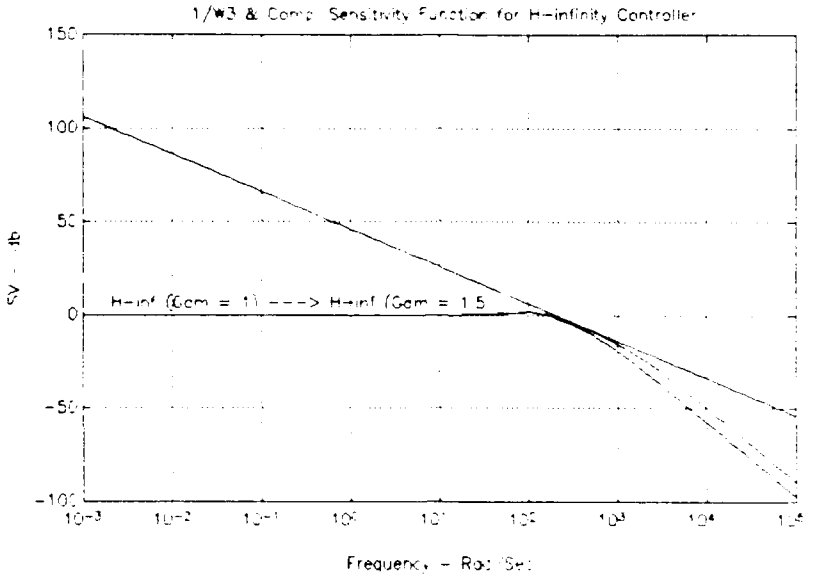

Figure 24 1/W3 & Complementary Sensitivity Function for H∞ Controller

The controller acts as a filter that generates the control from the plant output. Applying the H∞ controller resulted in a very fast closed loop system as shown in Figure 25:

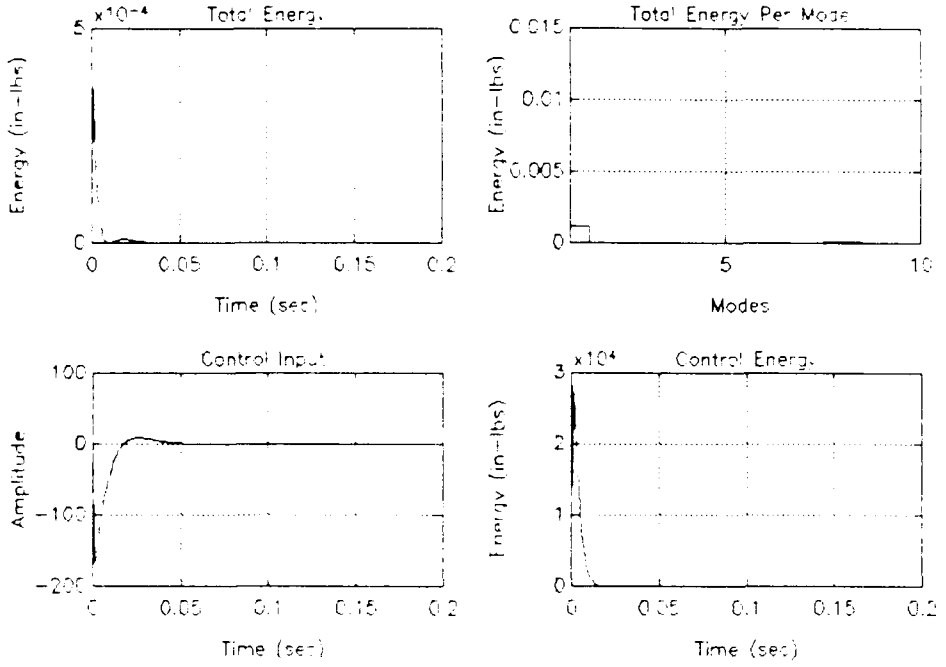

Figure 25 Energy and Control Input with H∞ Controller

The H∞ controller designed with weighting functions $W_1(s)$ and $W_3(s)$ had a large bandwidth of 500 rad/sec which added too much control to the system. An additional weighting factor was added to the performance measure to lessen the control applied to the system. The transfer function R(jw) was constrained: $R(jw) \leq W_2(jw)$. R has no common name, but is defined as:

$$\frac{U_c(s)}{U_1(s)} = R(s) = F(s)\,[I + G(s)K(s)]^{-1} \qquad \textbf{(3.14)}$$

where R(jw) is the ratio of the control input to the disturbance input in the real frequency domain. $W_2(s)$ was selected to be:

$$W_2(s) = \frac{10^9(s + 0.22)}{s + 2.4e10^3}. \qquad \textbf{(3.15)}$$

There was no significant difference as a result of this modification.

The next modification to the specifications was lowering the bandwidth of the system. Several values for the bandwidth from 100 rad/sec to 1 rad/sec were tried. The frequency range from 1.25 rad/sec to 2 rad/sec provided good response. A bandwidth of 2, however, yielded the best response and is shown in Figure 26.

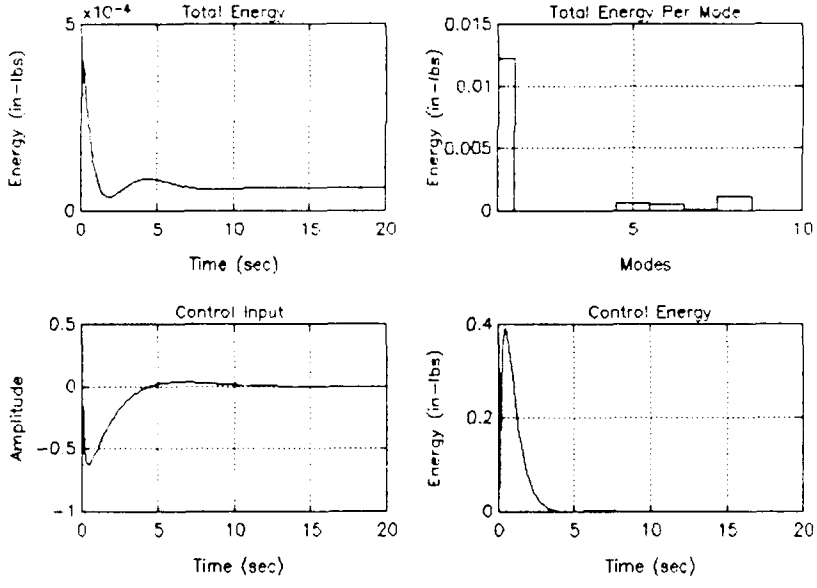

Figure 26 Energy and Control Input with H∞ Controller

Figure 26 illustrates that less control energy and control input was required to actively suppress the vibrations. Moreover, mode 1 required the most control and was suppressed

immediately with residual control for the other modes. The total energy in the system dissipates in about 20 seconds.

The full order design for the H∞ controller was based on trial and error. A better design may exist. The full order controller yielded good results. The total energy to suppress the vibrations with the H∞ controller was greater than that with LQG controller. However, it required only 20 seconds vice 140 seconds. The full order controllers require more CPU time, hence reduced order controllers are implemented and examined.

B. REDUCED ORDER CONTROLLERS

1. Linear Quadratic Gaussian Controller

The reduced order models are obtained by truncating the high frequency modes. The performance measure and design parameters remain the same as those for the full order controller. Controllers are designed for the reduced order models and applied to the full order system. The simulation software allows the user to specify the number of modes to be controlled through an interactive phase where the user is queried for the number of modes to be controlled. As the number of modes to be controlled decreases, less energy is dissipated as shown in Figure 27.

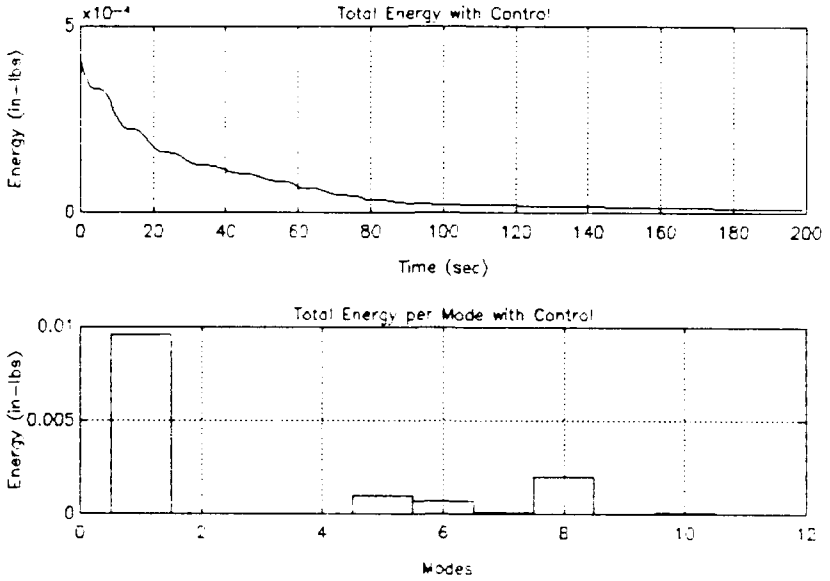

Figure 27 Total Energy Dissipated with 7 Modes Controlled

Figure 28 shows that the control energy also decreased. This is because there are fewer modes contributing to the control. The more modes controlled the closer the reduced-order controller was to the full-order design.

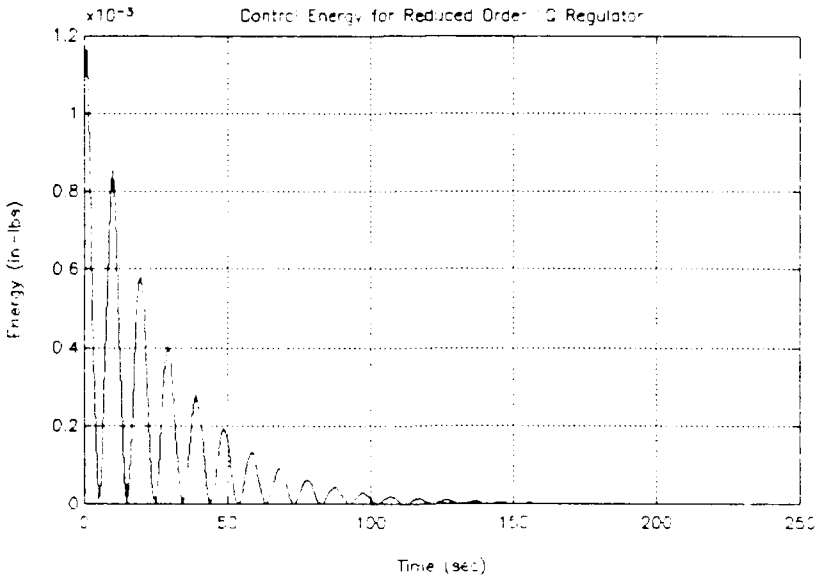

Figure 28 Control Energy with 7 Modes Controlled

With the Kalman filter added, the LQG controller exhibits good response with the impulse input. The actual and estimated energy with the reduced-order Kalman filter is shown in Figure 29:

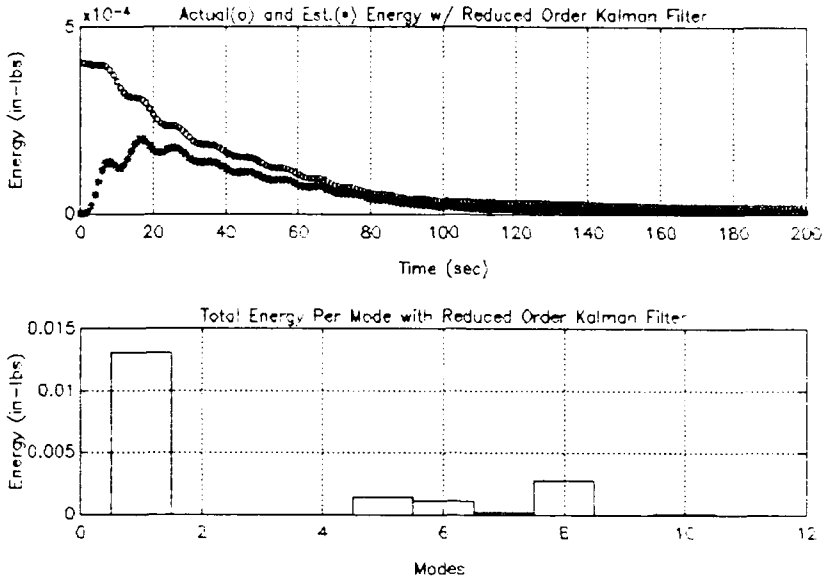

Figure 29 Reduced Order Controller with 7 Modes Controlled

The estimation of the actual position required approximately the same amount of time as Figure 18 (full order controller).

2. H∞ Controller

The design of the H∞ order reduced controller is similar to that of the LQG controller. A bandwidth of 2 rad/sec was used for the system again. Figure 30 illustrates damping of the modes and shows the amount of control and control energy with 7 modes controlled:

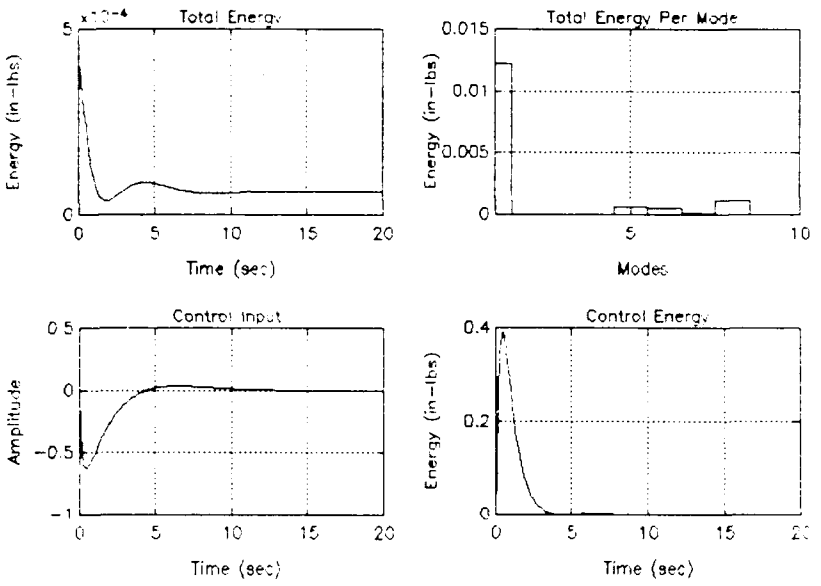

Figure 30 Energy and Control Input for H∞ Reduced Controller

The performance of the reduced-order system was reasonable and used less energy than the full order controller.

C. A COMPARISON OF LQG AND H∞ CONTROLLERS

Both design processes involved some insight on behalf of the designer to formulate the performance measures, i.e., the plant and measurement noise covariance matrices for the LQG controller and the weighting functions for the H∞ controller. The LQG controller exhibited good stability margins and also minimized the performance measure specifying by two weighting functions Q and R. The H∞ controller specified the performance measure using the weighting functions, $W_1(s)$ and $W_3(s)$. Thus, both controllers required the selection of weighting functions.

As expected the LQG controller provided good full order response with modelled noise. For robustness, another controller may be better suited because no guidelines exist to tailor the LQG controller for unmodelled dynamics. The H∞ controller is designed for such a condition.

The full-order controller, of course, exhibited better response than the reduced order controller. However, the reduced-order controllers worked reasonably and used less energy for control.

An indication of the robustness of the LQG and H∞ controller is the performance of the reduced-order controllers. Both controllers proved to be reasonably robust to unmodelled dynamics.

IV. CONCLUSIONS

The significant observations and results from the simulations generated are summarized. Areas that require further consideration for research and development are highlighted.

A. SUMMARY OF OBSERVATIONS

The design of the full order controllers was based on the full order model of the system. As a result, the "best" performance of the system was expected. The LQG controller illustrated good performance in the presence of modelled noise dynamics. However, it lost some of its robustness properties due to estimating the states with the Kalman filter. Applying LTR, the robustness was increased by implanting fictitious noise, thereby yielding a robust suboptimal system. The H∞ controller is an optimal method directly designed for robustness to unmodelled dynamics.

The H∞ controller design is simple and yet complicated for implementation. It, like the design of the LQG controller, requires some perspective in its design. The weighting functions used for augmentation, provide a tool for specifying robustness. The weighting functions characterize the specifications on disturbance attenuations and robustness. Simulations of the system with both the LQG and H∞ controller were then performed.

Simulation results indicated that both dynamic controllers provided good results. The LQG controller provided good response with less control energy required than the H∞ controller. The reduced order LQG and H∞ controllers both achieved good response when truncating the high frequency modes. Simulations of the system with 7 modes controlled were presented. It was shown that the reduced order controller actively suppressed the modes selected to be controlled. The reduced order controllers used less energy and control as expected.

B. FURTHER RESEARCH AND DEVELOPMENT

1. The range of natural frequency values should be increased in order to better evaluate the system response.

2. The model should be evaluated with random noise input disturbances.

3. μ-synthesis should be applied to the model and then compared to the LQG and H∞ controllers.

Appendix A
Zeros and Poles of Plant

Zeros	Poles
-4.5920e+07-1.2343e+05i	-1.6200e-03+3.2372e-01i
4.5920e+07+1.2343e+05i	-1.6200e-03-3.2372e-01i
-2.2193e-03-4.4395e-01i	-1.7350e-03+3.4727e-01i
-2.1600e-03-4.3173e-01i	-1.7350e-03-3.4727e-01i
-2.0590e-03-4.1223e-01i	-1.8900e-03+3.7881e-01i
-1.9631e-03-3.9319e-01i	-1.8900e-03-3.7881e-01i
-2.0148e-03-4.0345e-01i	-1.9050e-03+3.8105e-01i
-2.2200e-03+4.4407e-01i	-1.9050e-03-3.8105e-01i
-2.1600e-03+4.3173e-01i	-1.9800e-03+3.9619e-01i
-2.0590e-03+4.1223e-01i	-1.9800e-03-3.9619e-01i
-2.0148e-03+4.0345e-01i	-2.0250e-03+4.0533e-01i
-2.0391e-03+4.0753e-01i	-2.0250e-03-4.0533e-01i
-1.9631e-03+3.9319e-01i	-2.0400e-03+4.0767e-01i
-1.7350e-03-3.4727e-01i	-2.0400e-03-4.0767e-01i
-1.8900e-03-3.7881e-01i	-2.0750e-03+4.1521e-01i
-1.9050e-03-3.8105e-01i	-2.0750e-03-4.1521e-01i
-1.7350e-03+3.4727e-01i	-2.1600e-03+4.3174e-01i
-1.8900e-03+3.7881e-01i	-2.1600e-03-4.3174e-01i
-1.9050e-03+3.8105e-01i	-2.2200e-03+4.4407e-01i
-2.2193e-03+4.4395e-01i	-2.2200e-03-4.4407e-01i

Appendix B
Computer Programs

```
**************************************************************************
*                        LQGCTC.M                                        *
* This program generates the Linear Quadratic Regulator for the          *
*  the system.                                                           *
**************************************************************************

% In order to run this program some of the variables used in the
% model simulation program are required.

if exist('modenbr') == 1
  load lqgc_c.mat
else
  var
end

%For Linear Gaussian Control, the system is modeled as:
% aida_dot = A*aida + B1u  + B2w
%        y = C*aida + v
% where w is the plant driving noise and v is the measurement noise
% Control input is u= -K*aida where aida is the modal amplitude.
% Using the ten second order matrices, the response should
% be quicker than that of the twentieth order .

%Plant matrices for 20th order system to obtain full order
%state feedback controller

dv=zeros(19,1);
  for i=1:19
    if(rem(i,2)~=0);
      dv(i,1)=1;
    end
  end

A=diag(dv,1)+diag([-0.1048,0,-0.1206,0,-0.1435,0,-0.1452,0,...
 -0.15697,0,-0.1643,0,-0.1662,0,-0.1724,0,-0.1864,0,-0.1972],-1)+...
diag([0,-0.00324,0,-0.00347,0,-0.00378,0,-0.00381,0,-0.00396,0,...
     -0.00405,0,-0.00408,0,-0.00415,0,-0.00432,0,-0.00444],0);
B=[0;b(1);0;b(2);0;b(3);0;b(4);0;b(5);0;b(6);0;b(7);0;b(8);0;b(9);...
  0;b(10)];
Q=diag([f(1),1,f(2),1,f(3),1,f(4),1,f(5),1,f(6),1,f(7),1,f(8),...
       1,f(9),1,f(10),1],0);
```

```
R = 1;

% Full Order State Feedback where u = -K*[aida1,aida2,aida3,...]'
% Phi represents the discrete 2x2 matrix for each mode of vibration
% at natural frequency. These matrices can be found on the diagonals
% of this twentieth order matrix. Del also represents each mode of
% vibration. First two row entries correspond to a modal matrix
% at specified natural frequency.

ts = 1;                              %Sampling Time
[Phi,Del] = c2d(A,B,ts);
aida(:,1) = B;                 % Initialization
K = dlqr(Phi,Del,Q,R);

for n = 1:200                              % Time
  u = -K*aida(:,n);                        % No control added
    for mode = 1:10
      Phimode = Phi(2*mode-1:2*mode,2*mode-1:2*mode);
      Delmode = Del(2*mode-1:2*mode);
      Qe = Q(2*mode-1:2*mode,2*mode-1:2*mode);    % Energy Q matrix

% Aida gives the matrix for twenty states for each point in time.
% Next actual state calculated

     aida(2*mode-1:2*mode,n + 1) = Phimode*aida(2*mode-1:2*mode,n)...
                          + Delmode * u ; %  control added

% At each point in time the energy of each mode is calculated using
% Energy in specified mode  = 0.5[aida aida_d]Q[aida
%                                               aida_d]
     E(mode,n) = 0.5*[aida(2*mode-1,n) aida(2*mode,n)] *...
          Qe *[aida(2*mode-1,n);aida(2*mode,n)];
  end
end

% Plot of Total Dissipation of Modes
  for m = 1:10
    out_d(:,m) = aida(2*m-1,:)'*B(2*m);
  end

out_tot = sum(out_d');

% Plot of Total Displacement
time_o = 0:1:200;
```

```
plot(time_o,out_tot);
xlabel('Time(sec) ');ylabel('Amplitude');grid;
title('Total Displacement of Modes');
gtext(['R =', num2str(R) ]);
clg;

% Energy Calculation
E_total=sum(E);
time=0:1:199;
subplot(211),plot(E_total);grid;
title('Total Energy with Control');
xlabel('Time');
gtext(['R = ',num2str(R)]);pause

% Plot of Total Energy per Mode
Em_total=sum(E');
modes=1:1:10;
plot(modes,Em_total);grid;
title('Total Energy per Mode with Control');
gtext(['R = ',num2str(R)]);
```

```
**************************************************************************
*                          KALMAN.M                                      *
* This program generates the Kalman filter for the LQG controller.       *
**************************************************************************

% In order to run this program some of the variables used in the
% model simulation program are required. This program simulates the model
% with estimated states using steady state Kalman filter.

clg
format short e
format compact

if exist('b') == 1
  load var.mat
else
  var
end

% Plant matrices
dv=zeros(19,1);
  for i=1:19
    if(rem(i,2)~=0);
      dv(i,1)=1;
    end
  end

A=diag(dv,1)+diag([-0.1048,0,-0.1206,0,-0.1435,0,-0.1452,0,...
 -0.15697,0,-0.1643,0,-0.1662,0,-0.1724,0,-0.1864,0,-0.1972],-1)+...
diag([0,-0.00324,0,-0.00347,0,-0.00378,0,-0.00381,0,-0.00396,0,...
     -0.00405,0,-0.00408,0,-0.00415,0,-0.00432,0,-0.00444],0);
B=[0;b(1);0;b(2);0;b(3);0;b(4);0;b(5);0;b(6);0;b(7);0;b(8);0;b(9);...
  0;b(10)];
C=[b(1) 0 b(2) 0 b(3) 0 b(4) 0 b(5) 0 b(6) 0 b(7) 0 ...
  b(8) 0 b(9) 0 b(10) 0];
D=0;
Q=diag([f(1),1,f(2),1,f(3),1,f(4),1,f(5),1,f(6),1,f(7),1,f(8),...
       1,f(9),1,f(10),1],0);
R=1;
N=200;

% Initialize the random inputs to the same for each run.
rand('normal');
rand('seed',0);              % Sets the seed to 0 when Matlab is entered
```

```
% Initialization of the state and the estimate, x_hat
% and the error covariance matrix.
aida_hat(:,1)=zeros(B);
aida(:,1)=zeros(B);
old_P=zeros(A);

% Desired state
aida_d=zeros(B);

ts=1;                              %Sampling Time
W=input('Input W: ');
V=input('Input V: ');
[Phi,Del]=c2d(A,B,ts);
G=real(dlqe(Phi,Del,C,W,V));          % Steady State Optimal Gain
K=real(dlqr(Phi,Del,Q,R));

for k=1:N

% Plant simulation
  ww=zeros(N,1);ww(1)=1;
  u(k)=-K*(aida_hat(:,k)-aida_d);
  aida(:,k+1)=Phi*aida(:,k) + Del*u(k) + Del*ww(k);
  vv(k+1)=sqrt(V)*rand;
  y(k+1)=C*aida(:,k+1) + vv(k+1);

% Estimates updated
  aida_hat(:,k+1) = Phi*aida_hat(:,k) + Del *u(k);
  y_hat(k+1)      = C *aida_hat(:,k+1);
  aida_hat(:,k+1) = aida_hat(:,k+1) + G*(y(k+1)-y_hat(k+1));
  t(k)=k-1;

  for mode=1:10
     E_k(mode,k)=0.5*[aida(2*mode-1,k) aida(2*mode,k)]*...
     Q(2*mode-1:2*mode,2*mode-1:2*mode)*...
     [aida(2*mode-1,k);aida(2*mode,k)];
  end
end

   u(N+1)=-K*aida_hat(:,N+1);
   t(N+1)=N+1;

% Plots
subplot(211)
plot(t,aida(modenbr,:),'*',t,aida_hat(modenbr,:),'o');
```

```
title(['Estimated and Actual Position of Mode ',num2str(modenbr)]);
xlabel('seconds');ylabel('Displacement');grid;
pause

subplot(212),plot(t,u);grid;
title('Control Input');pause
%meta kalman
clg

% Plot of Total Dissipation of Modes
for d=1:10
   k_disp(:,d)=aida(2*d-1,:)'*B(2*d);
   kh_disp(:,d)=aida_hat(2*d-1,:)'*B(2*d);
end

t_disp=sum(k_disp');
th_disp=sum(kh_disp');

% Plot of Total Displacement
ktime=0:1:N;
plot(ktime,t_disp,'--',ktime,th_disp,'-.');
xlabel('Time(sec) ');ylabel('Amplitude');grid;
title('Total Displacement of Modes with Kalman Filter ');
%meta kalman
pause

% Plot of Total Energy
K_energy=0.5*(aida'*Q*aida);
Khenergy=0.5*(aida_hat'*Q*aida_hat);clg;
plot(ktime,diag(K_energy),'--',ktime,diag(Khenergy),'-.');grid;
title('Total Actual(--)and Estimated(-.)Energy w/ Kalman Filter');
xlabel('Time');ylabel('Magnitude');
pause;
%meta kalman
clg

% Plot of Total Energy per Mode
modes=1:1:10;
Emk_tot=sum(E_k');clg;
plot(modes,Emk_tot);grid;
title('Total Energy Per Mode with Kalman Filter');xlabel('Modes ');
ylabel('Amplitude');pause
%meta kalman
```

```
**************************************************************************
*                          JOSE.M                                        *
* This program generates the H∞ controller for the system. This program* 
* has been modified to reflect the specification design for             *
* this particular model.                                                *
**************************************************************************

% JOSE Large space structure design demonstration
%
% R. Y. Chiang & M. G. Safonov 3/88
% Copyright (c) 1988 by the MathWorks, Inc.
% All Rights Reserved.
% ------------------------------------------------------------
clc
disp('        < < Demo #3: MIMO Large Space Structure Design Example > >')
disp('                                          _________')
disp('        Secondary Mirror ----> -ooo-          ^')
disp('                              / \ / \         |')
disp('                              |  X  |         |')
disp('                              | / \ |         |')
disp('                              -------         |')
disp('                              | \ / |         |')
disp('                              |  X  |         |')
disp('                              | / \ |         ')
disp('                  Lens ------>  ---O---      7.4 Meters')
disp('                              |       |       ')
disp('                             |  \   /  |      |')
disp('                             |   \ /   |      |')
disp('                             |    X    |      |')
disp('                            |    / \    |     |')
disp('                            |   /   \   |     |')
disp('                           /   /     \   \    |')
disp('        Primary Mirror --> (_OOOOOO____\___)   |')
disp('                            \            /    |')
disp('                             \__________/ ___v___')
disp(' ')
disp('                              (strike a key to continue ...)')
pause

if exist ('b') = = 1
  load var.mat
else
  var
end
```

```
format short
format compact

dv=zeros(19,1);
  for i=1:19
     if (rem(i,2)~=0);
        dv(i,1)=1;
     end
  end

ag =diag(dv,1)+diag([-0.1048,0,-0.1206,0,-0.1435,0,-0.1452,0,...
  -0.15697,0,-0.1643,0,-0.1662,0,-0.1724,0,-0.1864,0,-0.1972],-1)+...
diag([0,-0.00324,0,-0.00347,0,-0.00378,0,-0.00381,0,-0.00396,0,...
    -0.00405,0,-0.00408,0,-0.00415,0,-0.00432,0,-0.00444],0);
bg=[0;b(1);0;b(2);0;b(3);0;b(4);0;b(5);0;b(6);0;b(7);0;b(8);0;b(9);...
  0;b(10)];
cg=[b(1) 0 b(2) 0 b(3) 0 b(4) 0 b(5) 0 b(6) 0 b(7) 0 ...
    b(8) 0 b(9) 0 b(10) 0];
dg = 0;
clc
disp(' ')
disp('  State-Space of the Large Space Structure:')
disp('  (after colocated rate feedback and model reduction)')
ag
bg
disp('                              (strike a key to continue ...)')
pause
clc
cg
dg
disp('                              (strike a key to continue ...)')
pause
clc
disp(' ')
disp(' ')
disp('   Poles of the Plant (stable):')
disp(' ')
disp('   ------------------------------------------------------------')
disp('      poleg = eig(ag)      % Computing the poles of the plant')
disp('   ------------------------------------------------------------')
poleg  = eig(ag)
pause
disp(' ')
disp(' ')
```

```
disp('                              (strike a key to continue ...)')
pause
clc
disp(' ')
disp('  ')
disp('   Transmission Zeros of the Plant (minimum phase):')
disp(' ')
disp('   ----------------------------------------------------------------')
disp('    tzerog = tzero(ag,bg,cg,dg)   % Computing the transmission zeros')
disp('   ----------------------------------------------------------------')
tzerog = tzero(ag,bg,cg,dg)      % Computing the transmission zeros
disp(' ')
disp(' ')
disp(' ')
disp('                              (strike a key to continue ...)')
pause
clc
disp(' ')
disp('    - - - Computing SV Bode plot of the open loop plant - - -')
w = logspace(-3,5,100);
svg = sigma(ag,bg,cg,dg,1,w); svg = 20*log10(svg);
disp(' ')
disp(' ')
disp(' ')
disp(' ')
disp(' ')
disp('                    (strike a key to see the SV Bode plot of G(s) ...)')
pause
semilogx(w,svg)
title('MIMO -- Mode 1: Open Loop Bode Plot ')
xlabel('Frequency - Rad/Sec')
ylabel('SV - db')
grid
pause
% meta bode
clc
disp(' ')
disp('                        < < Design Specifications > >  ')
disp(' ')
disp('     1). Robustness Spec. : closed loop bandwidth -- 200 r/s (30 hz)')
disp('         Associated Weighting:')
disp('    ')
disp('                   -1     200')
disp('                 W3(s) = ------ * I       (fixd)')
```

```
disp('                          s      2x2')
disp('   ')
disp(' ')
disp('      2). Performance Spec.: sensitivity reduction of at least 100:1')
disp('                             up to approx. 100 r/s')
disp('          Associated Weighting:')
disp(' ')
disp('                    -1        -1   0.01(1 + s/10)^2')
disp('                  W1(s) = Gam  * ---------------------- * I')
disp('                                    (1 + s/500)^2        2x2')
disp('   ')
disp('          where "Gam" in our design goes from 1 --> 1.5.')
coef=input('Enter the coef value');
fac=500/coef;
k=200/fac;
nuw3i = [0 k]; dnw3i = [1 0];
svw3i = bode(nuw3i,dnw3i,w); svw3i = 20*log10(svw3i);
nuw1i = conv([1/(10/fac) 1],[1/(10/fac) 1]);
dnw1i = 100*conv([1/coef 1],[1/coef 1]);
svw1i = bode(nuw1i,dnw1i,w); svw1i = 20*log10(svw1i);
disp(' ')
disp(' ')
disp('                  (strike a key to see the plot of the weightings ...)')
pause
axis([0 5 -40 40])
semilogx(w,svw1i,w,svw3i)
grid
title('MIMO Design Specifications -- Mode 1')
xlabel('Frequency - Rad/Sec')
ylabel('1/W1 & 1/W3 - db')
text(2,-20,'Sensitivity Spec.-- 1/W1(s)')
text(2,10,'Robustness Spec.-- 1/W3(s)')
pause
% meta weight
axis
clc
disp('                      < < Problem Formulation > >')
disp(' ')
disp('      Form an augmented plant P(s) with these two weighting functions:')
disp(' ')
disp('                  1). W1 penalizing error signal "e"')
disp('  ')
disp('                  2). W3 penalizing plant output "y"')
disp(' ')
```

```
disp('     and find a stabilizing controller F(s) such that the Hinf-norm')
disp('     of TF Ty1u1 is minimized and less than one, i.e.')
disp(' ')
disp('                    min |Ty1u1|    < 1,')
disp('                    F(s)       inf')
disp(' ')
disp('     where ')
disp('                  |              -1|')
disp('           Ty1u1 = | Gam*W1*(I + GF) |   = | Gam * W1 * S  |')
disp('                  |              -1|     |  W3 * (I - S) |')
disp('                  |  W3*GF*(I + GF) |')
disp(' ')
disp(' ')
disp(' ')
disp('                          (strike a key to continue ...)')
pause
clc
disp(' ')
disp(' ')
disp('                        < < DESIGN PROCEDURE > >')
disp(' ')
disp('          * * * * * * * * * * * * * * * * * * * * * * * * * * * * *')
disp('          *    [Step 1]. Do plant augmentation (run augtf.m or      *')
disp('          *              augss.m)                                    *')
disp('          *                                                          *')
disp('          *    [Step 2]. Balanced the augmented plant for better     *')
disp('          *              numerical condition (run OBALREAL.M)        *')
disp('          *                                                          *')
disp('          *    [Step 3]. Do H-inf synthesis (run HINF.M)             *')
disp('          *                                                          *')
disp('          *    [Step 4]. Redo the plant augmentation and balancing   *')
disp('          *              for a new "Gam" --> 1.5 and rerun HINF.M    *')
disp('          * * * * * * * * * * * * * * * * * * * * * * * * * * * * *')
disp(' ')
disp(' ')
disp('                          (strike a key to continue ...)')
pause
clc
disp(' ')
disp(' ')
disp('          Assign the cost coefficient "Gam" --> 1 ')
disp('      ')
disp('          this will serve as the baseline design ....')
disp(' ')
```

```
disp(' ')
Gam = input('            Input the cost coefficient "Gam" = ');
disp(' ')
disp('    ----------------------------------------------------------------')
disp('       % Plant augmentation of the LSS:')
disp('         sysg = [ag bg;cg dg]; xg = 4;')
disp('         w1 = [Gam*dnw1i;nuw1i;Gam*dnw1i;nuw1i];')
disp('         w2 = [];')
disp('         w3 = [dnw3i;nuw3i;dnw3i;nuw3i];')
disp('         [A,B1,B2,C1,C2,D11,D12,D21,D22] = augtf(sysg,xg,w1,w2,w3);')
disp('    ----------------------------------------------------------------')
sysg = [ag bg;cg dg]; xg = 20;
w1 = [Gam*dnw1i;nuw1i];
%nuw2i=10e9*[1 0.12];dnw2i=[1 2.4e3];
%w2=[dnw2i;nuw2i];
w2 = [];
w3 = [dnw3i;nuw3i];                          %dnw3i;nuw3i];
[A,B1,B2,C1,C2,D11,D12,D21,D22]=augtf(sysg,xg,w1,w2,w3);
disp(' ')
disp('    - - - State-Space (A,B1,B2,C1,C2,D11,D12,D21,D22) is ready for')
disp('          the Small-Gain problem - - -')
disp(' ')
%disp('    ----------------------------------------------------------------')
%disp('     [aa,bb,cc,mm,tt] = obalreal(A,[B1 B2],[C1;C2])   % Balancing P(s)')
%disp('     A = aa; B1 = bb(:,1:2); B2 = bb(:,3:4); ')
%disp('     C1 = cc(1:4,:); C2 = cc(5:6,:);')
%disp('    ----------------------------------------------------------------')
%[aa,bb,cc,mm,tt] = obalreal(A,[B1 B2],[C1;C2]);
%A = aa; B1 = bb(:,1:2); B2 = bb(:,3:4); C1 = cc(1:4,:); C2 = cc(5:6,:);
disp(' ')
disp(' ')
disp('                              (strike a key to continue ...)')
pause
clc
disp(' ')
disp(' ')
disp(' ')
disp('    ----------------------------------------------------------------')
disp('     hinf      % Running script file HINF.M for H-inf optimization')
disp('    ----------------------------------------------------------------')
hinf
disp(' ')
disp(' ')
disp('                              (strike a key to continue ...)')
```

```
pause
pltopt          % Preparing singular values for plotting
svw1i1 = svw1i; hsvs1 = svs; hsvt1 = svt; hsvtt1 = svtt;
disp(' ')
disp(' ')
disp('                              (strike a key to continue ...)')
pause
clc
disp(' ')
disp(' ')
disp('    After a few iterations, we found a new Gam of 1.5 can push the')
disp(' ')
disp('    H-inf cost function close to its limit. ')
disp(' ')
disp(' ')
disp('          Input "Gam" --> 1.5, and try HINF again .....')
disp(' ')
disp(' ')
Gam = input('            Input the cost coefficient "Gam" = ');
disp(' ')
disp('    ------------------------------------------------------------------')
disp('       % Adjust plant augmentation:')
disp('         w1 = [Gam*dnw1i;nuw1i;Gam*dnw1i;nuw1i];')
disp('         [A,B1,B2,C1,C2,D11,D12,D21,D22] = augtf(sysg,xg,w1,w2,w3);')
disp('    ------------------------------------------------------------------')
w1 = [Gam*dnw1i;nuw1i];                 %Gam*dnw1i;nuw1i];
[A,B1,B2,C1,C2,D11,D12,D21,D22]=augtf(sysg,xg,w1,w2,w3);
%[aa,bb,cc,mm,tt] = obalreal(A,[B1 B2],[C1;C2]);
%A = aa; B1 = bb(:,1:2); B2 = bb(:,3:4); C1 = cc(1:4,:); C2 = cc(5:6,:);
disp(' ')
disp(' ')
disp('                              (strike a key to continue ...)')
pause
hinf
disp(' ')
disp(' ')
disp('                              (strike a key to continue ...)')
pause
pltopt
svw1i2 = svw1i; hsvs2 = svs; hsvt2 = svt; hsvtt2 = svtt;
disp(' ')
disp(' ')
disp('                 (strike a key to see the plots of the comparison ...)')
pause
```

```
semilogx(w,svw1i1,w,hsvs1,w,svw1i2,w,hsvs2)
title('Mode 1: 1/W1 & Sensitivity Function for H-infinity Controller')
xlabel('Frequency - Rad/Sec')
ylabel('SV - db')
grid
text(0.002,0,'H-inf (Gam = 1) ---> H-inf (Gam = 1.5)')
% meta sens
pause
semilogx(w,svw3i,w,hsvt1,w,hsvt2)
title(' 1/W3 & Comp. Sensitivity Function for H-infinity Controller')
xlabel('Frequency - Rad/Sec')
ylabel('SV - db')
grid
text(0.002,0,'H-inf (Gam = 1) ---> H-inf (Gam = 1.5)')
% meta compl
pause
semilogx(w,hsvtt1,w,hsvtt2)
title('Mode 1: H-infinity Design Cost Function ')
xlabel('Frequency - Rad/Sec')
ylabel('SV - db')
grid
text(0.002,-10,'H-inf (Gam = 1) ---> H-inf (Gam = 1.5)')
% meta cost
pause
clc
disp(' ')
disp(' ')
disp('            << 8-State H-inf Controller (Gam = 1.5) >>')
disp('  ')
disp('   Poles of Controller :')
polecp = eig(acp)
disp(' ')
disp('                              (strike a key to continue ...)')
pause
clc
disp('   State-Space of the 8-State H-inf Controller:')
disp('      First 6 columns of the A matrix:')
acp(:,1:6)
disp('                              (strike a key to continue ...)')
pause
clc
disp(' ')
disp('      Last two columns of the A matrix:')
acp(:,7:8)
```

```
disp('                              (strike a key to continue ...)')
pause
clc
bcp
disp('                              (strike a key to continue ...)')
pause
clc
ccp
dcp
disp('                              (strike a key to continue ...)')
pause
clc
disp('   ')
disp('    Poles of closed-loop TF matrix Ty1u1:')
poletyu = eig(acl)
disp('                              (strike a key to continue ...)')
pause
%
save hinfcc.mat ag bg cg dg acp bcp ccp dcp b f modenbr A;
clear
load hinfcc.mat
% ----- End of JOSE.M --- RYC/MGS %
```

```
*************************************************************************
* This program simulates the model with the H∞ controller generated*
* by the modified Matlab Robust Control Toolbox program Jose.m      *
*************************************************************************

% In order to run this program some of the variables used in the
% model simulation program are required. This program simulates the
% model with the H-infinity controller.

clg
format short
format compact

if exist('acp')  = = 0
    jose
else
   load hinfcc.mat
   load var.mat
end

Q=diag([f(1),1,f(2),1,f(3),1,f(4),1,f(5),1,f(6),1,f(7),1,f(8),...
       1,f(9),1,f(10),1],0);
N=200;

% Initialize the random inputs to the same for each run.
rand('normal');

% Initialization of the state and control state vector
xcl=[bg;zeros(bcp)];
aida(:,1)=xcl(1:20,1);
y(1)=cg*aida(:,1);
ww=zeros(N,1);

% Calculation of controller
ts=0.1;
acl=[ag-bg*dcp*cg -bg*ccp;bcp*cg acp];
bcl=[bg;zeros(bcp)];
[Phi_cl,Del_cl]=c2d(acl,bcl,ts);

for n=1:N

% Plant simulation with the H-infinity controller of the form:
%                xc_dot = acp*xc + bcp*y
%                   uc = -(ccp*xc + dcp*y)
```

```
xcl(:,n+1) = Phi_cl*xcl(:,n) + Del_cl*ww(n);
aida(:,n+1)=xcl(1:20,n+1);
lccp=length(ccp);
uc=-ccp*xcl(21:20+lccp,:)-dcp*cg*xcl(1:20,:);

% Calculation of energy in system
        for mode =1:10
          E_h(mode,n)=0.5*[aida(2*mode-1,n) aida(2*mode,n)]*...
          Q((2*mode-1):(2*mode),(2*mode-1):(2*mode))*...
          [aida(2*mode-1,n);aida(2*mode,n)];
        end
end

% Plots
subplot(211)
time=0:1:N;
plot(time,aida(modenbr,:));
title([ ' Position of Mode ',num2str(modenbr)]);
xlabel('seconds');ylabel('Displacement');
grid;
%meta pos1
pause

% Plot of Total Dissipation of Modes
for d=1:10
  h_disp(:,d)=aida(2*d-1,:)'*bg(2*d);
end

t_disp=sum(h_disp');

% Plot of Total Displacement
htime=0:1:N;clg;
plot(htime,t_disp);
xlabel('Time(sec) ');ylabel('Amplitude');grid;
title('Total Displacement of Modes with H-infinity Controller');
%meta hindisp
pause

% Plot of Total Energy
E_htot=sum(E_h);
time=0:ts:19.9;
subplot(221),;plot(time,E_htot);grid;
title('Total Energy');
xlabel('Time (sec)');ylabel('Energy (in-lbs)');
```

```
pause;axis;
%meta test2

% Plot of Total Energy per Mode
modes=1:1:10;
Emh_tot=sum(E_h');
subplot(222),axis([1 10 0 0.015]);plot(modes,Emh_tot);grid;
title('Total Energy Per Mode ');
xlabel('Modes ');ylabel('Energy (in-lbs)');pause
%meta energy8

subplot(223),axis;plot(uc);grid;
title(['Control Input for Mode ',num2str(modenbr)]);
xlabel(' Time (sec)');ylabel('Energy (in-lbs)');
%meta control1
pause

subplot(224),
plot(diag(uc'*uc));title('Control Energy');grid;
xlabel('Time (sec)');ylabel('Energy (in-lbs)');
%meta fullw2
```

```
*************************************************************************
*                          LQGCTCR.M                                    *
* This program is designed to generate a reduced order LQ regulator     *
* controller. The number of modes to be controlled is determined when   *
* the user is queried by the program.                                   *
*************************************************************************
% In order to run this program some of the variables used in the
% model simulation program are required.

if exist('modenbr') == 1
  load var.mat
else
  var
end


%Plant matrices for 20th order system for full order
%state feedback controller

dv=zeros(19,1);
  for i=1:19
    if(rem(i,2)~=0);
      dv(i,1)=1;
    end
  end

A=diag(dv,1)+diag([-0.1048,0,-0.1206,0,-0.1435,0,-0.1452,0,...
 -0.15697,0,-0.1643,0,-0.1662,0,-0.1724,0,-0.1864,0,-0.1972],-1)+...
diag([0,-0.00324,0,-0.00347,0,-0.00378,0,-0.00381,0,-0.00396,0,...
     -0.00405,0,-0.00408,0,-0.00415,0,-0.00432,0,-0.00444],0);
B=[0;b(1);0;b(2);0;b(3):0;b(4);0;b(5);0;b(6):0;b(7);0;b(8);0;b(9);...
  0;b(10)];
Q=diag([f(1),1,f(2),1,f(3),1,f(4),1,f(5),1,f(6),1,f(7),1,f(8),...
       1,f(9),1,f(10),1],0);
R=1;

% Number of modes to be controlled
nbr_mode=input('How many modes do you desire to control (1:10)? ');

% The following procedure is for a reduced order controller.
% The user selects the number of modes to be controlled.

ts=1;                          %Sampling Time
```

```
[Phi,Del]=c2d(A,B,ts);
aida(:,1)=B;                    % Initialization
Phi_r=Phi(1:nbr_mode*2,1:nbr_mode*2);
Del_r=Del(1:nbr_mode*2);
Q_r=Q(1:nbr_mode*2,1:nbr_mode*2);
K_r=dlqr(Phi_r,Del_r,Q_r,R);
K=[K_r zeros(1,20-nbr_mode*2)];

for n=1:100                             % Time
  u=-K*aida(:,n);                       % No control added
    for mode=1:10
      Phimode=Phi(2*mode-1:2*mode,2*mode-1:2*mode);
      Delmode=Del(2*mode-1:2*mode);
      Qe=Q(2*mode-1:2*mode,2*mode-1:2*mode);    % Energy Q matrix

% Aida gives the matrix for twenty states for each point in time.
% Next actual state calculated

     aida(2*mode-1:2*mode,n+1)=Phimode*aida(2*mode-1:2*mode,n)...
                        + Delmode * u ; %  control added

% At each point in time the energy of each mode is calculated using
% Energy in specified mode  =  0.5[aida aida_d]Q[aida
%                                                aida_d]

  E(mode,n)=0.5*[aida(2*mode-1,n) aida(2*mode,n)] *...
          Qe *[aida(2*mode-1,n);aida(2*mode,n)];

  end
end

% Plot of Total Dissipation of Modes

  for m=1:10
    out_d(:,m)=aida(2*m-1,:)'*B(2*m);
  end

out_tot=sum(out_d');

% Plot of Total Displacement
time_o=0:1:100;
plot(time_o,out_tot);
xlabel('Time(sec) ');ylabel('Amplitude');grid;
title('Total Displacement of Modes');
```

```
gtext(['R =', num2str(R) ]);
 meta redd
clg;

E_total=sum(E);
time=0:1:99;
subplot(211),plot(time,E_total);grid;
title('Total Energy with Control');
xlabel('Time');
gtext(['R = ',num2str(R)]);
gtext(['Number of Modes Controlled: ',num2str(nbr_mode)]);
pause

% Plot of Total Energy per Mode
Em_total=sum(E');
modes=1:1:10;
subplot(212),plot(modes,Em_total);grid;
title('Total Energy per Mode with Control');
gtext(['R = ',num2str(R)]);
```

```
**************************************************************************
*                          KALMANR.M                                     *
* This program generates the reduced order controller for the Kalman     *
* filter.                                                                *
**************************************************************************

% In order to run this program some of the variables used in the
% model simulation program are required.
% This program simulates the model with estimated states using
% steady state Kalman filter.

clg
format short e
format compact

if exist('b') == 1
  load var.mat
else
  var
end

% Plant matrices
dv=zeros(19,1);
  for i=1:19
    if(rem(i,2)~=0);
      dv(i,1)=1;
    end
  end

A=diag(dv,1)+diag([-0.1048,0,-0.1206,0,-0.1435,0,-0.1452,0,...
 -0.15697,0,-0.1643,0,-0.1662,0,-0.1724,0,-0.1864,0,-0.1972],-1)+...
diag([0,-0.00324,0,-0.00347,0,-0.00378,0,-0.00381,0,-0.00396,0,...
     -0.00405,0,-0.00408,0,-0.00415,0,-0.00432,0,-0.00444],0);
B=[0;b(1);0;b(2);0;b(3);0;b(4);0;b(5);0;b(6);0;b(7);0;b(8);0;b(9);...
  0;b(10)];
C=[b(1) 0 b(2) 0 b(3) 0 b(4) 0 b(5) 0 b(6) 0 b(7) 0 ...
  b(8) 0 b(9) 0 b(10) 0];
D=0;
Q=diag([f(1),1,f(2),1,f(3),1,f(4),1,f(5),1,f(6),1,f(7),1,f(8),...
       1,f(9),1,f(10),1],0);
R=1;
N=200;
nbr_mode=input('Number of modes to control: ');
```

```
% Initialize the random inputs to the same for each run.
rand('normal');
rand('seed',0);   % Sets the seed to 0 when Matlab is entered

% Initialization of the state and the estimate, x_hat
% and the error covariance matrix.
aida_hat(:,1)=zeros(B);
aida(:,1)=zeros(B);
old_P=zeros(A);

% Desired state
aida_d=zeros(B);

ts=1;                              %Sampling Time
W=40;
V=0.005;
[Phi,Del]=c2d(A,B,ts);
G=real(dlqe(Phi,Del,C,W,V));          % Steady State Optimal Gain
K=real(dlqr(Phi,Del,Q,R));            % Gain matrix K

%For reduced order matrices
Phi_kr=Phi(1:nbr_mode*2,1:nbr_mode*2);
Del_kr=Del(1:nbr_mode*2);
Q_kr=Q(1:nbr_mode*2,1:nbr_mode*2);
K_kr=dlqr(Phi_kr,Del_kr,Q_kr,R);
K=[K_kr zeros(1,20-nbr_mode*2)];

for k=1:N

% Plant simulation
  ww(k)=zeros(N,1);ww(1)=1;
  u(k)=-K*(aida_hat(:,k)-aida_d);
  aida(:,k+1)=Phi*aida(:,k) + Del*u(k) + Del*ww(k);
  vv(k+1)=sqrt(V)*rand;
  y(k+1)=C*aida(:,k+1) + vv(k+1);

% Estimates updated
  aida_hat(:,k+1) = Phi*aida_hat(:,k) + Del *u(k);
  y_hat(k+1)      = C *aida_hat(:,k+1);
  aida_hat(:,k+1) = aida_hat(:,k+1) + G*(y(k+1)-y_hat(k+1));

 for mode=1:10
     E_k(mode,k)=0.5*[aida(2*mode-1,k) aida(2*mode,k)]*...
     Q(2*mode-1:2*mode,2*mode-1:2*mode)*...
```

```
      [aida(2*mode-1,k);aida(2*mode,k)];
  end
end

u(N+1)=-K*aida_hat(:,N+1);
t(N+1)=N+1;

% Plots
subplot(211)
plot(t,aida(modenbr,:).'o',t,aida_hat(modenbr,:),'*');
title(['Estimated and Actual Position of Mode ',num2str(modenbr)]);
xlabel('Time(sec)');ylabel('Displacement');
grid;%gtext(['W = ',num2str(W)]);gtext(['V = ',num2str(V)]);
%gtext('w/ rand plant disturbance and meas noise');
%gtext(['Reduced Order Controller/Kalman Filter:',num2str(nbr_mode)]);
pause

% Plot of Total Dissipation of Modes
for d=1:10
   k_disp(:,d)=aida(2*d-1,:)'*B(2*d);
   kh_disp(:,d)=aida_hat(2*d-1,:)'*B(2*d);
end

t_disp=sum(k_disp');
th_disp=sum(kh_disp');

% Plot of Total Displacement
ktime=0:1:N;subplot(212)
plot(ktime,t_disp,'--',ktime,th_disp,'-.');
xlabel('Time(sec) ');ylabel('Amplitude');grid;
title(' Total Displacement of Modes with Reduced Order Kalman Filter');
meta kalrdiw3
pause
clg

K_energy=0.5*(aida'*Q*aida);
Khenergy=0.5*(aida_hat'*Q*aida_hat);subplot(211)
plot(ktime,diag(K_energy),'o',ktime,diag(Khenergy),'*');grid;
title('Actual(o) and Est.(*) Energy w/ Reduced Order Kalman Filter');
xlabel('Time');ylabel('Magnitude');
pause;

% Plot of Total Energy per Mode
modes=1:1:10;
```

```
Emk_tot=sum(E_k');
subplot(212),plot(modes,Emk_tot);grid;
title('Total Energy Per Mode with Reduced Order Kalman Filter');
xlabel('Modes ');ylabel('Amplitude');pause
meta kalrepw3
clg

plot(u);grid;xlabel('Time (sec)');ylabel('Magnitude');
title('Control Input with Reduced Order Kalman Filter');pause
clg

plot(t,diag(u'*u));
grid;title('Control Energy with Reduced Order Kalman Filter');
xlabel('Time (sec)');ylabel('Amplitude');
```

```
*************************************************************************
*                          JOSER.M                                      *
* This program generates the reduced order H∞ controller for            *
* the Hinfcr.m simulation program.                                      *
*************************************************************************

% JOSER Large space structure design demonstration
%
% R. Y. Chiang & M. G. Safonov 3/88
% Copyright (c) 1988 by the MathWorks, Inc.
% All Rights Reserved.
% ------------------------------------------------------------
clc
disp('       < < Demo #3: MIMO Large Space Structure Design Example > >')
disp('                                        _________')
disp('       Secondary Mirror ----> -ooo-      ^')
disp('                             / \ / \      |')
disp('                             |  X  |      |')
disp('                             | / \ |      |')
disp('                             -------      |')
disp('                             | \ / |      |')
disp('                             |  X  |      |')
disp('                             | / \ |      ')
disp('                Lens ------>  ---O---    7.4 Meters')
disp('                             |     |      ')
disp('                            |   \ /   |    |')
disp('                            |  \   /  |    |')
disp('                            |  \   /  |    |')
disp('                            |  \   /  |    |')
disp('                            |    X    |    |')
disp('                           |    / \    |   |')
disp('                           |   /   \   |   |')
disp('                          /   /     \   \  |')
disp('       Primary Mirror --> (_OOOOOO____\___)  |')
disp('                           \           /   |')
disp('                            \_________/ ___v___')
disp(' ')
disp('                              (strike a key to continue ...)')
pause

if exist('b') == 1
  load var.mat
else
  var
```

```
end

nbr_mode=input('Enter the number of modes to control: ');

format short
dv=zeros(19,1);
  for i=1:19
     if (rem(i,2)~=0);
        dv(i,1)=1;
     end
  end

ag =diag(dv,1)+diag([-0.1048,0,-0.1206,0,-0.1435,0,-0.1452,0,...
  -0.15697,0,-0.1643,0,-0.1662,0,-0.1724,0,-0.1864,0,-0.1972],-1)+...
diag([0,-0.00324,0,-0.00347,0,-0.00378,0,-0.00381,0,-0.00396,0,...
    -0.00405,0,-0.00408,0,-0.00415,0,-0.00432,0,-0.00444],0);
bg=[0;b(1);0;b(2);0;b(3);0;b(4);0;b(5);0;b(6);0;b(7);0;b(8);0;...
   b(9);0;b(10)];
cg = [b(1),0,b(2),0,b(3),0,b(4),0,b(5),0,b(6),0,b(7),0,b(8),0,...
     b(9),0,b(10),0];
dg = 0;
clc
% To reduce the order of the model, the size of the matrix is
% determined the number of modes to control
ag_r=ag(1:nbr_mode*2,1:nbr_mode*2);
bg_r=bg(1:nbr_mode*2);
cg_r=cg(1,1:nbr_mode*2);
dg_r=0;
disp(' ')
disp('   State-Space of the Large Space Structure:')
disp('   (after colocated rate feedback and model reduction)')
%ag
%bg
%disp('                              (strike a key to continue ...)')
%pause
%clc
%cg
%dg
%disp('                              (strike a key to continue ...)')
%pause
%clc
disp(' ')
disp(' ')
disp('    Poles of the Plant (stable):')
```

```
disp(' ')
disp('    ------------------------------------------------------------')
disp('      poleg = eig(ag)      % Computing the poles of the plant')
disp('    ------------------------------------------------------------')
poleg  = eig(ag_r)
disp(' ')
disp(' ')
disp('                                (strike a key to continue ...)')
pause
clc
disp(' ')
disp('    ')
disp('    Transmission Zeros of the Plant (minimum phase):')
disp(' ')
disp('    ------------------------------------------------------------')
disp('      tzerog = tzero(ag,bg,cg,dg)   % Computing the transmission zeros')
disp('    ------------------------------------------------------------')
tzerog = tzero(ag_r,bg_r,cg_r,dg)      % Computing the transmission zeros
disp(' ')
disp(' ')
disp(' ')
disp('                                (strike a key to continue ...)')
pause
clc
disp(' ')
disp('     - - - Computing SV Bode plot of the open loop plant - - -')
w = logspace(-3,5,100);
svg = sigma(ag_r,bg_r,cg_r,dg,1,w); svg = 20*log10(svg);
disp(' ')
disp(' ')
disp(' ')
disp(' ')
disp(' ')
disp('                    (strike a key to see the SV Bode plot of G(s) ...)')
pause
semilogx(w,svg)
title('MIMO Large Space Structure Open Loop')
xlabel('Frequency - Rad/Sec')
ylabel('SV - db')
grid
pause;%meta jack1
clc
disp(' ')
disp('                        < < Design Specifications > >  ')
```

```
disp(' ')
disp('      1). Robustness Spec. : closed loop bandwidth -- 200 r/s (30 hz)')
disp('         Associated Weighting:')
disp('     ')
disp('                  -1      200')
disp('               W3(s) = ------ * I       (fixd)')
disp('                       s      2x2')
disp('    ')
disp(' ')
disp('      2). Performance Spec.: sensitivity reduction of at least 10:1')
disp('                          up to approx. 10 r/s')
disp('         Associated Weighting:')
disp(' ')
disp('                  -1      -1  0.01  (1 + s/10)^2')
disp('               W1(s) = Gam  * ---------------------- * I')
disp('                                 (1 + s/500)^2        2x2')
disp('    ')
disp('         where "Gam" in our design goes from 1 --> 1.5.')
coef=input('Enter the coef value');
fac=500/coef;
k=200/fac;
nuw3i = [0 k]; dnw3i = [1 0];
nuw1i = conv([1/(10/fac) 1],[1/(10/fac) 1]);
dnw1i = 100*conv([1/coef 1],[1/coef 1]);
%k = 200;
%nuw3i=[0 k];dnw3i = [1 0];
svw3i = bode(nuw3i,dnw3i,w); svw3i = 20*log10(svw3i);
%nuw1i = conv([1/10 1],[1/10 1]);dnw1i = 100*conv([1/500 1],[1/500 1]);
svw1i = bode(nuw1i,dnw1i,w); svw1i = 20*log10(svw1i);
disp('  ')
disp('  ')
disp('               (strike a key to see the plot of the weightings ...)')
pause
axis([0 5 -40 40])
semilogx(w,svw1i,w,svw3i)
grid
title('MIMO LSS Design Example -- Design Specifications')
xlabel('Frequency - Rad/Sec')
ylabel('1/W1 & 1/W3 - db')
text(2,-20,'Sensitivity Spec.-- 1/W1(s)')
text(2,10,'Robustness Spec.-- 1/W3(s)')
pause;%meta jack1
axis
clc
```

```
disp('                  << Problem Formulation >>')
disp(' ')
disp('     Form an augmented plant P(s) with these two weighting functions:')
disp(' ')
disp('               1). W1 penalizing error signal "e"')
disp('  ')
disp('               2). W3 penalizing plant output "y"')
disp(' ')
disp('     and find a stabilizing controller F(s) such that the Hinf-norm')
disp('     of TF Ty1u1 is minimized and less than one, i.e.')
disp('  ')
disp('                     min |Ty1u1|    < 1,')
disp('                     F(s)       inf')
disp(' ')
disp('     where ')
disp('                    |              -1|')
disp('           Ty1u1 = | Gam*W1*(I + GF) |   = | Gam * W1 * S   |')
disp('                    |              -1|     | W3 * (I - S) |')
disp('                    |  W3*GF*(I + GF) |')
disp('  ')
disp('  ')
disp('  ')
disp('                          (strike a key to continue ...)')
pause
clc
disp('  ')
disp(' ')
disp('                          << DESIGN PROCEDURE >>')
disp('  ')
disp('          * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *')
disp('          *    [Step 1]. Do plant augmentation (run augtf.m or       *')
disp('          *              augss.m)                                     *')
disp('          *                                                           *')
disp('          *    [Step 2]. Balanced the augmented plant for better     *')
disp('          *              numerical condition (run OBALREAL.M)         *')
disp('          *                                                           *')
disp('          *    [Step 3]. Do H-inf synthesis (run HINF.M)            *')
disp('          *                                                           *')
disp('          *    [Step 4]. Redo the plant augmentation and balancing  *')
disp('          *              for a new "Gam" --> 1.5 and rerun HINF.M     *')
disp('          * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *')
disp('  ')
disp('  ')
disp('                          (strike a key to continue ...)')
```

```
pause
clc
disp(' ')
disp(' ')
disp('          Assign the cost coefficient "Gam" --> 1 ')
disp('      ')
disp('          this will serve as the baseline design ....')
disp(' ')
disp(' ')
Gam = input('            Input the cost coefficient "Gam" = ');
disp(' ')
disp('    ----------------------------------------------------------------')
disp('       % Plant augmentation of the LSS:')
disp('         sysg = [ag bg;cg dg]; xg = 4;')
disp('         w1 = [Gam*dnw1i;nuw1i;Gam*dnw1i;nuw1i];')
disp('         w2 = [];')
disp('         w3 = [dnw3i;nuw3i;dnw3i;nuw3i];')
disp('         [A,B1,B2,C1,C2,D11,D12,D21,D22] = augtf(sysg,xg,w1,w2,w3);')
disp('    ----------------------------------------------------------------')
sysg = [ag_r bg_r;cg_r dg]; xg = nbr_mode*2;
w1 = [Gam*dnw1i;nuw1i];              %Gam*dnw1i;nuw1i];
w2 = [];
w3 = [dnw3i;nuw3i];                  % dnw3i;nuw3i];
[A,B1,B2,C1,C2,D11,D12,D21,D22]=augtf(sysg,xg,w1,w2,w3);
disp(' ')
disp('    - - - State-Space (A,B1,B2,C1,C2,D11,D12,D21,D22) is ready for')
disp('          the Small-Gain problem - - -')
disp(' ')
disp('    ----------------------------------------------------------------')
disp('     [aa,bb,cc,mm,tt] = obalreal(A,[B1 B2],[C1;C2])   % Balancing P(s)')
disp('     A = aa; B1 = bb(:,1:2); B2 = bb(:,3:4); ')
disp('     C1 = cc(1:4,:); C2 = cc(5:6,:);')
disp('    ----------------------------------------------------------------')
%[aa,bb,cc,mm,tt] = obalreal(A,[B1 B2],[C1;C2]);
%A = aa; B1 = bb(:,1:2); B2 = bb(:,3:4); C1 = cc(1:4,:); C2 = cc(5:6,:);
disp(' ')
disp(' ')
disp('                                (strike a key to continue ...)')
pause
clc
disp(' ')
disp(' ')
disp(' ')
disp('    ----------------------------------------------------------------')
```

```
disp('    hinf      % Running script file HINF.M for H-inf optimization')
disp('   ------------------------------------------------------------')
hinfr
disp(' ')
disp(' ')
disp('                              (strike a key to continue ...)')
pause
pltoptr          % Preparing singular values for plotting
svw1i1 = svw1i; hsvs1 = svs; hsvt1 = svt; hsvtt1 = svtt;
disp(' ')
disp(' ')
disp('                              (strike a key to continue ...)')
pause
clc
disp(' ')
disp(' ')
disp('    After a few iterations, we found a new Gam of 1.5 can push the')
disp(' ')
disp('    H-inf cost function close to its limit. ')
disp(' ')
disp(' ')
disp('           Input "Gam" --> 1.5, and try HINF again .....')
disp(' ')
disp(' ')
Gam = input('             Input the cost coefficient "Gam" = ');
disp(' ')
disp('    ------------------------------------------------------------')
disp('        % Adjust plant augmentation:')
disp('          w1 = [Gam*dnw1i;nuw1i;Gam*dnw1i;nuw1i];')
disp('          [A,B1,B2,C1,C2,D11,D12,D21,D22] = augtf(sysg,xg,w1,w2,w3);')
disp('    ------------------------------------------------------------')
w1 = [Gam*dnw1i;nuw1i];                    %Gam*dnw1i;nuw1i];
[A,B1,B2,C1,C2,D11,D12,D21,D22]=augtf(sysg,xg,w1,w2,w3);
%[aa,bb,cc,mm,tt] = obalreal(A,[B1 B2],[C1;C2]);
%A = aa; B1 = bb(:,1:2); B2 = bb(:,3:4); C1 = cc(1:4,:); C2 = cc(5:6,:);
disp(' ')
disp(' ')
disp('                              (strike a key to continue ...)')
pause
hinf
disp(' ')
disp(' ')
disp('                              (strike a key to continue ...)')
pause
```

```
pltoptr
svw1i2 = svw1i; hsvs2 = svs; hsvt2 = svt; hsvtt2 = svtt;
disp(' ')
disp(' ')
disp('              (strike a key to see the plots of the comparison ...)')
pause
semilogx(w,svw1i1,w,hsvs1,w,svw1i2,w,hsvs2)
title('H-inf LSS Design -- 1/W1 & Sensitivity Func.')
xlabel('Frequency - Rad/Sec')
ylabel('SV - db')
grid;
text(0.002,0,'H-inf (Gam = 1) ---> H-inf (Gam = 1.5)')
pause;%meta jack1
semilogx(w,svw3i,w,hsvt1,w,hsvt2)
title('H-inf LSS Design -- 1/W3 & Comp. Sens. Func.')
xlabel('Frequency - Rad/Sec')
ylabel('SV - db')
grid
text(0.002,0,'H-inf (Gam = 1) ---> H-inf (Gam = 1.5)')
pause;%meta jack1
semilogx(w,hsvtt1,w,hsvtt2)
title('H-inf LSS Design -- Cost function Ty1u1')
xlabel('Frequency - Rad/Sec')
ylabel('SV - db')
grid
text(0.002,-10,'H-inf (Gam = 1) ---> H-inf (Gam = 1.5)')
pause;%meta jack1

clc
disp(' ')
disp(' ')
disp('          << 8-State H-inf Controller (Gam = 1.5) >>')
disp('  ')
disp('  Poles of Controller :')
polecp = eig(acp_r)
disp(' ')
disp('                          (strike a key to continue ...)')
pause
clc
disp('  State-Space of the 8-State H-inf Controller:')
disp('    First 6 columns of the A matrix:')
acp_r(:,1:nbr_mode*2)
disp('                          (strike a key to continue ...)')
pause
```

```
clc
disp(' ')
%disp('       Last two columns of the A matrix:')
%acp(:,7:8)
disp('                                (strike a key to continue ...)')
pause
clc
bcp_r
disp('                                (strike a key to continue ...)')
pause
clc
ccp_r
dcp_r
disp('                                (strike a key to continue ...)')
pause
clc
disp('   ')
disp('    Poles of closed-loop TF matrix Ty1u1:')
poletyu = eig(acl)
disp('                                (strike a key to continue ...)')
pause
%
% ----- End of JOSEDEMO.M --- RYC/MGS %
save hinfcr.mat acp_r bcp_r ccp_r dcp_r b ag bg cg dg nbr_mode modenbr f;
clear
load hinfcr.mat
```

```
*************************************************************************
*                          HINFCR.M                                     *
* This program simulates the model with the reduced order H∞ controller *
*************************************************************************
% In order to run this program some of the variables used in the
% model simulation program are required. This program simulates the
% model with the Reduced Order H-infinity controller.

clg
format short e
format compact

if exist('acp') == 0
  joser
else
  load hinfcr.mat
  load var.mat
end

% Plant matrices
dv=zeros(19,1);
  for i=1:19
    if(rem(i,2)~=0);
      dv(i,1)=1;
    end
  end

Q=diag([f(1),1,f(2),1,f(3),1,f(4),1,f(5),1,f(6),1,f(7),1,f(8),...
       1,f(9),1,f(10),1],0);
N=200;

% Initialize the random inputs to the same for each run.
rand('normal');
rand('seed',0);   % Sets the seed to 0 when Matlab is entered

% Initialization of the state and control state vector
aida(:,1)=bg;
xcl_r=zeros([bg;bcp]);
y(1)=cg*aida(:,1);
ww=zeros(N,1);ww(1)=1;

%Calculation of controller and Conversion to discrete time
ts=0.1;
acl_r=[ag-bg*dcp*cg -bg*ccp;bcp*cg acp];
```

```
bcl_r=[bg;zeros(bcp)];
[Phi_clr,Del_clr]=c2d(acl_r,bcl_r,ts);

for n=1:N

% Plant simulation with the H-infinity controller of the form:
%                xc_dot = acp*xc + bcp*y
%                   uc = ccp*xc + dcp*y

% Controller state updated and input updated
xcl_r(:,n+1) = Phi_clr*xcl_r(:,n) + Del_clr*ww(n);
aida(:,n+1)=xcl_r(1:20,n+1);
lccp_r=length(ccp);
uc_r=-ccp*xcl_r(21:20+lccp_r,:)-dcp*cg*xcl_r(1:20,:);

% Calculation of energy in system
        for mode = 1:10
          E_hr(mode,n)=0.5*[aida(2*mode-1,n) aida(2*mode,n)]*...
          Q(2*mode-1:2*mode,2*mode-1:2*mode)*...
          [aida(2*mode-1,n);aida(2*mode,n)];
        end
end

% Plots
time=0:1:N;
plot(time,aida(modenbr,:));
title([' Position of Mode ',num2str(modenbr)]);
xlabel('seconds');ylabel('Displacement');
grid;%gtext('w/ impulse plant disturbance ');%meta jack
pause

% Plot of Total Dissipation of Modes

for d=1:10
  h_disp(:,d)=aida(2*d-1,:)'*bg(2*d);
end

t_disp=sum(h_disp');

% Plot of Total Displacement
htime=0:1:N;clg;
plot(htime,t_disp');
xlabel('Time (sec) ');ylabel('Amplitude');grid;
title('Total Displacement of Modes w/ H-infinity Reduced Controller');
```

```
%meta hinfcr
pause

% Plot of Total Energy
time=0:ts:19.9;
subplot(221),plot(time,sum(E_hr));
title('Total Energy ');grid;
xlabel('Time (sec)');ylabel('Energy (in-lbs)');
pause;
%meta testr2

% Plot of Total Energy per Mode
modes=1:1:10;
Emh_tot=sum(E_hr');subplot(222)
subplot(222),axis([1 10 0 0.015]);plot(modes,Emh_tot);grid;
title('Total Energy Per Mode ');
xlabel('Modes ');ylabel('Energy (in-lbs)');pause
%meta hinfcr

subplot(224),
plot(diag(uc_r'*uc_r));title('Control Energy');grid;
xlabel(' Time (sec)');ylabel(' Energy (in-lbs)');
pause
meta red15_7;

clg;plot(uc_r);grid;
title('Control Input');
xlabel(' Time (sec)');ylabel('Energy (in-lbs)');
%meta contred7
pause
```

LIST OF REFERENCES

1. Osman, Tony, *Space History*, St. Martin's Press, New York,, NY, 1985, p. 18.

2. McCurdy, Howard E., *The Space Station Decision*, John Hopkins University Press, Baltimore, MD, 1990, p. 66.

3. National Aeronautics and Space Administration, *Space Station Task Force*, Noyes Publication, Park Ridge, NJ, pp. 10-11.

4. Franklin, Gene F., Powell, J., David, Abbas, Emani-Naeii, *Feedback Control of Dynamic Systems*, Addison-Wesley Publishing Company, Inc., 1986, p. 7.

5. Franklin, Gene F., Powell, J., David, Abbas, Emani-Naeii, *Feedback Control of Dynamic Systems*, Addison-Wesley Publishing Company, Inc., 1986, p. 9.

6. Kirk, Donald E., *Optimal Candle Theory*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1970, p. 53.

7. Friedland, Bernard, *Control System Design*, McGraw-Hill, Inc., 1986, p. 469.

8. Zames, G., "Feedback and Optimal Sensitivity:Model Reference Transformations, Multiplicatives, Seminorms, and Approximate Inverses," IEEE Transaction on Automatic Control, vol. AC-26, pp. 301-320.

9. PHONCON between author and Hardeval, John, McDonnell Douglas Astronautics Co., 20 Nov 91.

10. PHONCON between author and Hardeval, John, McDonnell Douglas Astronautics Co., 26 Nov 91.

11. Meirovitch, Leonard, *Elements of Vibration Analysis*, McGraw-Hill, Inc., 1986, New York, NY, p. 164.

12. McDonnell Douglas Space Systems Company, Space Station Division, 5301 Bolsa Avenue, Huntington Beach, CA.

13. Kirk, Donald G., *Optimal Control Theory*, Prentice-Hall, inc., Englewood Cliffs, NJ, 1970, p. 10.

14. Burl, Jeff B., "Linear Optimal Estimation and Control," unpublished class notes, Naval Postgraduate School, Monterey, CA, Sep 91.

15. Doyle, J.C., Stein, G., "Robustness with Observers," IEEE transaction on Automatic Control, vol. AC-24, p. 93.

16. Anderson, Brian D.O., Moore, John B., *Optimal Control*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1990, pp. 236-237.

17. Anderson, Brian D.O., Moore, John B., *Optimal Control*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1990, p. 237.

18. Doyle, J.C., Stein, G., "Multivariable Feedback Design: Concepts for a Classical/Modern Synthesis," IEEE Transaction on Automatic Control, vol. AC-26, 1981, pp. 4-16.

19. Burl, Jeff B., "Impulse Response," unpublished class notes, Naval Postgraduate School, Sep 91.

20. Meirovitch, Leonard, *Elements of Vibration Analysis*, McGraw-Hill, Inc., 1986, New York, NY, p. 396.

21. Safonor, M.G., Chiang, R.Y., Flasher, H., "H∞ Robust Control Synthesis for a Large Space Structure," proceedings of American Control Conference, 15 Aug 87, pp. 2038-2045.

INITIAL DISTRIBUTION LIST

Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
Library, Code 52 Naval Postgraduate School Monterey, California 93943-5002	2
Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
Prof. J. B. Burl, Code EC/Bl Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
Prof. R. B. Strum, Code EC/St Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
Commander, Naval Ocean Systems Command c/o Lieutenant Jacqueline R. McClusky San Diego, California 92152-5000	1
Naval Research Laboratory 4500 Overlook Drive Washington, DC 20390	1