Air Force Institute of Technology
AFIT Scholar

Theses and Dissertations Student Graduate Works

3-1999

Robust Eigenstructure Assignment Using Positive Definite Output Feedback Control

Adam G. Harris

Follow this and additional works at: https://scholar.afit.edu/etd

Part of the Aerospace Engineering Commons

Recommended Citation

Harris, Adam G., "Robust Eigenstructure Assignment Using Positive Definite Output Feedback Control" (1999). *Theses and Dissertations*. 5162.
https://scholar.afit.edu/etd/5162

This Thesis is brought to you for free and open access by the Student Graduate Works at AFIT Scholar. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AFIT Scholar. For more information, please contact AFIT.ENWL.Repository@us.af.mil.

AFIT/GAE/ENY/99M-06

Robust Eigenstructure Assignment Using Positive Definite Output Feedback Control

THESIS

Adam G. Harris, B.S.
First Lieutenant, USAF

AFIT/GAE/ENY/99M-06

Approved for public release; distribution unlimited

DTIC QUALITY INSPECTED 2

19990409 011

Robust Eignstructure Assignment Using Positive Output Feedback Control

Disclaimer

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the United States Government.

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GAE/ENY/99M-06

Robust Eigenstructure Assignment Using Positive Definite Output Feedback Control

THESIS

Adam G. Harris, B.S.
First Lieutenant, USAF

AFIT/GAE/ENY/99M-06

Approved for public release; distribution unlimited

Robust Eignstructure Assignment Using Positive Output Feedback Control

Disclaimer

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the United States Government.

AFIT/GAE/ENY/99M-06

Robust Eigenstructure Assignment Using Positive Definite Output Feedback Control

THESIS

Presented to the Faculty of the Graduate School of Engineering of the Air Force Institute of Technology Air University In Partial Fulfillment for the Degree of

Master of Science in Aeronautical Engineering

Adam G. Harris, B.S.

First Lieutenant, USAF

Air Force Institute of Technology

Wright-Patterson AFB, Ohio

March 1999

Approved for public release; distribution unlimited

AFIT/GAE/ENY/99M-06

Robust Eigenstructure Assignment Using Positive Definite Output Feedback Control

Adam G. Harris, B.S.

First Lieutenant, USAF

Approved:

Bradley S. Liebst

Dr. Bradley S. Liebst
Committee Chairman

26 Feb 99
Date

Maj. Jeffrey S. Turcotte
Committee member

26 Feb 99
Date

Curtis H. Spenny

Dr. Curtis H. Spenny
Committee member

26 Feb 99
Date

Dedication

I dedicate this work to my family. My wife Audrey was my constant support and the most wonderful person that I know. My dog Polly was truly man's best friend. So, to my family who enriched my soul and was always there with love and open hearts I dedicate this eighteen month effort.

Acknowledgments

I would first like to acknowledge the help and guidance of Dr. Brad Liebst. This thesis began with his idea and was strongly influenced by his encouragement and constant mentorship. I would also like to thank the readers on my committee, Maj. Jeff Turcotte and Dr. Curtis Spenny. Their effort helped focus this research and provided a check on this work that was invaluable.

The works of John Junkins helped immensely with the mathematics and literature review I used to begin my research. Also, the AFIT theses that came before me provided a framework and starting point for the layout for this work.

Table Of Contents

Page

List of Figures

Page

List of Tables

List of Symbols

English Symbols

Symbol	Definition
M	Mass or inertial matrix
C	Damping matrix
K	Stiffness matrix
D	Control matrix
$\mathbf{x}$	State vector, or eigenvector in eq (68)
$\mathbf{y}$	Output vector, or eigenvector in eq (68)
$\mathbf{z}$	First order form state vector, or eigenvector in eq (68)
$\mathbf{u}$	Actuator vector
$\mathbf{v}$	Eigenvector
G_P	Position gain matrix
G_R	Rate gain matrix
A	System matrix
B	Input matrix
J	Cost function value
V	Eigenvalue
$\mathcal{R}$	Real number set
∞	Infinity

Greek Symbols

Symbol	Definition
λ	Eigenvalue
Λ	Diagonal matrix of eigenvalues

Φ	Open-loop modal matrix
η	Vector of modal coordinates
ω	Natural frequency
ϕ	Natural mode
Ω	Diagonal matrix of natural frequencies squared
ξ	Transformed eigenvectors
Ψ	Closed-loop eigenvectors
$\hat{\omega}$	Closed-loop eigenvalues
α	Eigenvalue of A matrix
β	Eigenvalue of B matrix
γ	Eigenvalue of C matrix
ϵ	Small value much less than one
ζ	Damping factor

Subscripts

Symbol	Definition
e	Equilibrium state (Lyapunov stability section)
e	Eigenvalue (Cost function section)
v	Eigenvector
a	Achievable (eigenvalue)
d	Desired (eigenvalue)
s	Specific value
n	Natural (frequency)
d	Damped (frequency)

Superscript

Symbol	Definition
T	Transpose of a matrix
-1	Inverse of a matrix
$(\dot{\ })$	Differentiation with respect to time
$(\hat{\ })$	New damping or stiffness matrix
0	Zeroth order
1	First order
2	Second order

Abbreviations

Abbreviation	Definition
AFIT	Air Force Institute of Technology
PACOSS	Passive and Active Control of Space Structures
AIAA	American Institute of Aeronautics and Astronautics
LQR	Linear Quadratic Regulator
DOF	Degree-Of-Freedom

AFIT/GAE/ENY/99M-06

Abstract

This research investigates a method of eigenstructure assignment for the classes of controllers known as positive definite output feedback control. The research began with a closed loop system with output feedback control that utilizes a collocated actuator-sensor pair. It is shown that this system is robust because it is globally stable for positive definite feedback. Analytical analysis is then completed to determine gains that minimize a cost function that is the error between the desired and achievable (assuming positive definite output feedback) eigenstructure. Examples are given at the end of the thesis to validate the computer code and theory results.

Robust Eigenstructure Assignment Using Positive Definite Output Feedback Control

Chapter 1 - Introduction

1.1 Background

Many researchers have examined the problem of controlling flexible systems, the most obvious Air Force application being the control of large space structures. These systems include future space stations which have many flexible appendages, large antenna satellites, as well as systems such as the Air Force Institute of Technology's (AFIT) Passive and Active Control of Space Structures (PACOSS) experiment. Weight considerations due to the high cost of current space lift systems drive designers to choose low mass and highly flexible materials which are easily excited by vibrations in daily operations. Passive (material considerations) and active control of these vibrational problems has given rise to a large body of research material. This thesis will only cover active control and leaves passive control to the material designers and researchers, though it will draw on the methods of finite element modeling and vibrations theory. Specifically, this work will examine the active control methods using eigenvalue assignment.

Previous theses by Robinson [8] and Huckabone [1] give algorithms for eigenvalue assignment using linear quadratic regulator (LQR) techniques. Robinson used MatLab™ and the LQR routines to minimize a cost function of weighted eigenvalues. His main complaint was the run time required for MatLab™ on a Compaq 286, which in one case ran for seventeen hours before the PC ran out of memory. The next thesis by Huckabone converted the algorithm to FORTRAN to save computer time and used a cost function that included both the eigenvalues and the eigenvectors. He noticed a ten fold increase in compute time for the FORTRAN algorithm over Robinson's MatLab™ routines. Finally, a thesis by Solomon [9] extended Huckabone's work on the LQR problem to cross correlation weighting and applied the algorithms in FORTRAN to two helicopter systems. Since computer speeds have increased rapidly and look like they will continue to do so, this thesis will use MatLab™ for an eigenstructure algorithm but will not use the LQR technique. The reason to use LQR is for the stability robustness guarantees. This thesis will use positive definite output feedback to robustly guarantee stability.

John L. Junkins and Youdan Kim wrote a book for the AIAA Education Series titled "Introduction to Dynamics and Control of Flexible Structures." [3] In this work, Junkins described output feedback and provided a literature review which is current as of the publishing date of 1993. This book provided an understanding of Lyapunov stability as it applies to the problem of output feedback, as well as a great beginning for a literature review. Junkins also edited a work for the Progress in Astronautics and Aeronautics series which contained many of the papers cited in his literature review [2].

1.2 Problem Statement

Many dynamical systems are modeled using Newton's laws or Lagrange's equations. The result is a second order system of linear constant coefficient differential equations. This class of systems can be mathematically described by the equations of motion

$$M\ddot{\mathbf{x}} + C\dot{\mathbf{x}} + K\mathbf{x} = D\mathbf{u} \tag{1}$$

where $\mathbf{x} \in \Re^n$ and $\mathbf{u} \in \Re^m$ are the state and control (actuator) vectors respectively, M is the $n \times n$ positive definite symmetric mass matrix, C is the $n \times n$ positive semidefinite symmetric structural damping matrix, K is the $n \times n$ positive semidefinite stiffness matrix, D is the $n \times m$ control influence matrix, and $\dot{( )}$ is differentiation with respect to time.

As an introduction to output feedback and taking advantage of collocated sensors and actuators, the control and measurement equations can be written as

$$\mathbf{y} = D^T\mathbf{x} \tag{2}$$

$$\dot{\mathbf{y}} = D^T\dot{\mathbf{x}} \tag{3}$$

$$\mathbf{u} = -G_P\mathbf{y} - G_R\dot{\mathbf{y}} \tag{4}$$

where $\mathbf{y} \in \Re^m$ is the output (sensor) vector, and due to the assumption of collocation of the sensors and actuators, D is the same control influence matrix as in Equation 1, and G_P and G_R are the $m \times m$ position and rate feedback gains. Equations 2 and 3 can be substituted into Equation 4. Then that equation can be substituted back into Equation 1 and everything can be taken to the left hand side:

$$M\ddot{\mathbf{x}} + (C + DG_R D^T)\dot{\mathbf{x}} + (K + DG_P D^T)\mathbf{x} = 0. \tag{5}$$

This equation can be simplified notationally to the following, which is used in discussions of Lyapunov Stability

$$M\ddot{\mathbf{x}} + (C + \hat{C})\dot{\mathbf{x}} + (K + \hat{K})\mathbf{x} = 0 \tag{6}$$

where $\hat{C} = DG_R D^T$ and $\hat{K} = DG_P D^T$.

The problem statement, then, is to place the eigenvalues and eigenvectors (hereafter referred to as the eigenstructure) using Equation 5. This work examines a cost function that minimizes the difference between the actual and desired eigenstructure. Finally, this thesis presents an algorithm for creating computer code.

1.3 Methodology

The research for this thesis included a literature review, mathematical proofs, and a computer algorithm for designing a feedback control system. This effort began with a review of many of the sources available in the AFIT Library. As detailed in the introduction background, three previous theses by AFIT students provided a framework and excellent references to begin the literature search. Also, the AIAA Education Series title "Introduction to Dynamics and Control of Flexible Structures" by John L. Junkins and Youdan Kim provided an invaluable literature review for output feedback material up to 1993. [3]

The mathematical proofs presented throughout this work have been given in the past, but have never been combined and used for this application. This thesis represents a melding of many different areas of research from dynamics, to vibrations, to material analysis.

Finally, the computer algorithm began with a review of work done by Huckabone [1] and Lee [5] to validate the new and updated code. After coding this algorithm in MatLab™ and testing it on previous systems with known results, the code was then used on the example problems presented later in the final chapter.

1.4 Organization

This thesis is organized around creating a control system for a flexible structure. Figure 1 illustrates how each chapter corresponds to a step in the creation process.

This thesis begins with the mathematical theory which is necessary to carry out the eigenstructure assignment algorithm. The theory in Chapter Two discusses Lyapunov stability and positive definite output feedback. Chapter Three presents a proof of the bounds on eigenvalue placement that is original to this work. Then, the eigenvalue assignment system is discussed in Chapter Four. Finally, Chapter Five provides some examples which validate the algorithm and show its usefulness. The appendices are used for the computer code.

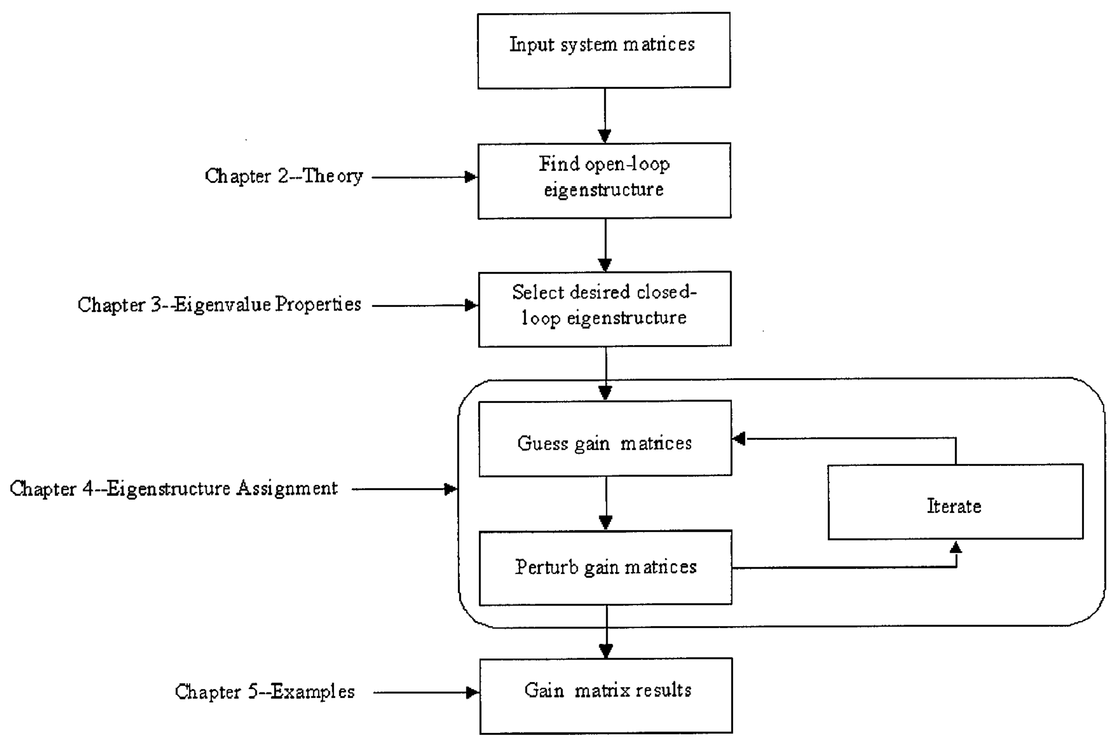

Figure 1. Block diagram of eigenvalue assignment process

Chapter 2 - Theory

This chapter begins with a review of Lyapunov stability and a detailed account of how to guarantee asymptotic stability for second order equations of motion. Then output feedback is introduced showing that the stability of the controlled closed-loop system is guaranteed by choosing positive definite gain matrices. Finally, a method of assigning values in the gain matrices with a cost function is used to drive achievable eigenvalues and eigenvectors (i.e. the eigenstructure) optimally close to the desired eigenstructure.

The two main equations used in this chapter and throughout this thesis are the second order structural equations of motion and the state space representation of a multivariable, linear, time-invariant feedback system. These equations are related by the following derivation beginning with the second order equations of motion.

$$M\ddot{\mathbf{x}} + C\dot{\mathbf{x}} + K\mathbf{x} = D\mathbf{u} \tag{7}$$

where $\mathbf{x} \in \Re^n$and $\mathbf{u} \in \Re^m$are the state and control(actuator) vectors respectively, M is the $n \times n$ positive definite symmetric mass matrix, C is the $n \times n$ positive semidefinite symmetric structural damping matrix, K is the $n \times n$ positive semidefinite stiffness matrix, D is the $n \times m$ control influence matrix, and $(\dot{\ })$ is differentiation with respect to time.

A variable, $\mathbf{z}$, is defined as

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{bmatrix} \tag{8}$$

so,

$$M\dot{\mathbf{z}} = \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}} \\ \ddot{\mathbf{x}} \end{bmatrix} = \begin{bmatrix} 0 & M \\ -K & -C \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{bmatrix} + \begin{bmatrix} 0 \\ D \end{bmatrix} \mathbf{u} \tag{9}$$

and, since M is non-singular, it is invertible, and

$$\dot{\mathbf{z}} = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}C \end{bmatrix} \mathbf{z} + \begin{bmatrix} 0 \\ M^{-1}D \end{bmatrix} \mathbf{u} \tag{10}$$

which is the same as the standard state space form

$$\dot{\mathbf{z}} = A\mathbf{z} + B\mathbf{u}. \tag{11}$$

Therefore, the second order system and the first order state space representation are equivalent. They are just different forms of the same system of equations. The second order system matrices M, C, and K will be n x n while the first order A matrix will be $2n$ x $2n$. The state space representation will be used in the eigenstructure assignment algorithm, but the second order system will be used to show the stability guarantees of positive definite output feedback control and the properties of the closed-loop eigenvalues.

2.1 Lyapunov Stability

The following theorems for asymptotic stability are found in Junkins [3] for a continuous, finite-dimensional dynamic system which can be described by the first order nonlinear equations

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t), \qquad \mathbf{x} \in \mathcal{R}^n. \tag{12}$$

Junkins notes that the function $\mathbf{f}(\mathbf{x}, t)$ is continuous and at least piecewise differentiable one or more times with respect to all arguments. Also note that these theorems are descriptions of the stability of a system from some reference equilibrium state.

Theorem 1 *The equilibrium state* $\mathbf{x}_e$ *is stable if there exists a continuously differentiable function* U *such that*
(1) $U(\mathbf{x}_e) = 0$
(2) $U(\mathbf{x}) > 0$ *for all* $\mathbf{x} \neq \mathbf{x}_e$
(3) $\dot{U}(\mathbf{x}) \leq 0$ *for all* $\mathbf{x} \neq \mathbf{x}_e$

Theorem 2 *The equilibrium state* $\mathbf{x}_e$ *is globally asymptotically stable if there exists a continuously differentiable function* U *such that*
(1) $U(\mathbf{x}_e) = 0$
(2) $U(\mathbf{x}) > 0$ *for all* $\mathbf{x} \neq \mathbf{x}_e$
(3) $\dot{U}(\mathbf{x}) < 0$ *for all* $\mathbf{x} \neq \mathbf{x}_e$
(4) $U(\mathbf{x}) \longrightarrow \infty$ *as* $\| \mathbf{x} \| \longrightarrow \infty$

For the closed-loop system of equations we write the following:

$$M\ddot{\mathbf{x}} + (C + \hat{C})\dot{\mathbf{x}} + (K + \hat{K})\mathbf{x} = 0. \tag{13}$$

The terms M, C, and K are the inertial, damping, and stiffness matrices of a structure. The $\hat{C}$ and $\hat{K}$ are the velocity control and displacement control matrices described in section 1.2 and further explored in the next section. For Lyapunov stability, we first define a candidate function, which in this structural case is the total mechanical energy, as

$$U = \frac{1}{2}\dot{\mathbf{x}}^T M\dot{\mathbf{x}} + \frac{1}{2}\mathbf{x}^T(K + \hat{K})\mathbf{x}. \tag{14}$$

Then, we differentiate the candidate function with respect to time and get

$$\dot{U} = \frac{1}{2}\ddot{\mathbf{x}}^T M\dot{\mathbf{x}} + \frac{1}{2}\dot{\mathbf{x}}^T M\ddot{\mathbf{x}} + \frac{1}{2}\dot{\mathbf{x}}^T(K + \hat{K})\mathbf{x} + \frac{1}{2}\mathbf{x}^T(K + \hat{K})\dot{\mathbf{x}}. \tag{15}$$

Since U is a scalar, all four terms in the above equation are scalars. Also, the transpose of a scalar is equal to that same scalar, and since $M = M^T$, $K = K^T$, and $\hat{K} = \hat{K}^T$, we can rewrite the equation as

$$\dot{U} = \dot{\mathbf{x}}^T M\ddot{\mathbf{x}} + \dot{\mathbf{x}}^T(K + \hat{K})\mathbf{x} = \dot{\mathbf{x}}^T[M\ddot{\mathbf{x}} + (K + \hat{K})\mathbf{x}]. \tag{16}$$

Finally, rearranging once more and using Equation 13 we find

$$\dot{U} = \dot{\mathbf{x}}^T[-(C + \hat{C})\dot{\mathbf{x}}] = -\dot{\mathbf{x}}^T(C + \hat{C})\dot{\mathbf{x}} \tag{17}$$

Returning to the definition of Lyapunov stability, first we look at U in Equation 14. M is positive definite and K is usually positive semidefinite. If there are no rigid body modes present, K will be positive definite and U will be greater than zero. In cases where rigid body modes are present, if the system is controllable, $\hat{K}$ can be chosen so that $(K + \hat{K})$ is positive definite. Finally, if K has unstable modes, special care needs to be taken to ensure that $\hat{K}$ is chosen such that $(K + \hat{K})$ becomes positive definite. Thus, we can choose $\hat{K}$ so that our Lyapunov function, U, is positive definite.

Now we turn our attention to Equation 17 for $\dot{U}$. $\dot{U}$ will be negative semidefinite as long as $(C + \hat{C})$ is positive semidefinite. The $\hat{C}$ may again have to be chosen to negate rigid body modes just as was done with the stiffness matrix, but usually only adjustments to the stiffness matrix will be necessary. Junkins proves that these conditions lead to asymptotic stability and states a conclusion that ". . . if the system is controllable, then the closed-loop system is at least asymptotically stable if the gain matrices are chosen properly." [3, pg. 88] The proper choice of gain matrices refers only to the conditions above which state that $(K + \hat{K})$ must be positive definite and that $(C + \hat{C})$ must be positive semidefinite.

The next section explores positive definite output feedback. Using positive definite output feedback $\hat{C}$ and $\hat{K}$ will be shown to be positive definite. Therefore, we will guarantee the four conditions for asymptotic stability. We have just shown conditions two and three. The first condition, $U(\mathbf{x}_e) = 0$, is satisfied simply by choosing the equilibrium point where the total mechanical energy for the system is zero. This will occur when the system is at rest and the position is at an equilibrium state. The fourth condition is satisfied by setting the position and velocity states to infinity in the total mechanical energy equation. The mechanical energy will go to infinity as the states goes to infinity. We see that all four conditions can be satisfied, so with the proper feedback law we can guarantee that the system will be asymptotically stable.

2.2 Positive Definite Output Feedback

This thesis uses positive definite output feedback and will show how this control law guarantees asymptotic stability. The second order equations of motion are repeated here for convenience

$$M\ddot{\mathbf{x}} + C\dot{\mathbf{x}} + K\mathbf{x} = D\mathbf{u} \tag{18}$$

Using output feedback and taking advantage of collocated sensors and actuators, the control equations can be written as

$$\mathbf{y} = D^T\mathbf{x} \tag{19}$$

$$\dot{\mathbf{y}} = D^T\dot{\mathbf{x}} \tag{20}$$

$$\mathbf{u} = -G_P\mathbf{y} - G_R\dot{\mathbf{y}} \tag{21}$$

where $\mathbf{y} \in \Re^m$ is the output (sensor) vector, D is the same control influence matrix as in Equation 7, and G_P and G_R are the $m \times m$ position and rate gain matrices. Equations 19 and 20 can be substituted into Equation 21. Then, that equation can be substituted back into Equation 18 and everything can be taken to the left hand side

$$M\ddot{\mathbf{x}} + (C + DG_RD^T)\dot{\mathbf{x}} + (K + DG_PD^T)\mathbf{x} = 0 \tag{22}$$

Please note that the assumption that the sensors and actuator are linear and instantaneous in operation was used to write the control law. If G_R and G_P are positive definite, then DGD^T for both position and rate feedback will be symmetric positive semidefinite. The gain matrices will always be positive definite if we chose them using a Cholesky decomposition of the form

$$G_R = L_R L_R^T \quad G_P = L_P L_P^T \tag{23}$$

where L_R is defined as

$$L_R = \begin{bmatrix} r_{11} & 0 & 0 & \cdots & 0 \\ r_{21} & r_{22} & 0 & \cdots & 0 \\ r_{31} & r_{32} & r_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & r_{n3} & \cdots & r_{nn} \end{bmatrix} \tag{24}$$

and L_P is defined as

$$L_p = \begin{bmatrix} p_{11} & 0 & 0 & \cdots & 0 \\ p_{21} & p_{22} & 0 & \cdots & 0 \\ p_{31} & p_{32} & p_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & p_{n3} & \cdots & p_{nn} \end{bmatrix} \tag{25}$$

where the r_{nn} and p_{nn} are scalars so that both of the matricies are lower triangular. Therefore, when we multiply L by its transpose we get a gain matrix of

$$LL^T = \begin{bmatrix} l_{11}^2 & l_{11}l_{21} & \cdots & l_{11}l_{n1} \\ l_{11}l_{21} & l_{22}^2 & \cdots & l_{21}l_{n1} + l_{22}l_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ l_{11}l_{n1} & l_{21}l_{n1} + l_{22}l_{n2} & \cdots & l_{nn}^2 \end{bmatrix} \tag{26}$$

where $L = L_R$ or L_P and $l_{ij} = r_{ij}$ or p_{ij} so that the diagonal terms are always positive, and the gain matrices will be positive definite. As was shown in the previous section, this class of controllers will guarantee asymptotic stability of the closed-loop system. An important part of this analysis is that since G_R and G_P are positive definite then M, $K + DG_PD^T$, and $C + DG_RD^T$ will always be positive definite for a controllable system. This result is independent of any reduced order model used, and regardless of inaccuracies in the parameter values used in the design. Only the predicted performance, or optimality, will be reduced as a result of modelling errors. [3, pg. 359]

Use of this control method must take into consideration the controllability of the system. As shown in the previous section, if there is an unstable mode the control system of sensors and actuators must first and foremost stabilize the system. Therefore, special care must be taken to ensure the positive definiteness of M, $K + DG_P D^T$, and $C + DG_R D^T$. In a practical sense, most flexible structures and most systems will have positive definite mass and stiffness with positive semidefinite damping. Therefore, the gain matrices can be used exclusively to optimize, by some method, the controlled closed-loop system.

We can conclude that if we chose positive definite gain matrices our closed-loop system will always be asymptotically stable. The system will be robust in the sense that stability is preserved in the face of modeling errors and reduced order models. The assumption that we have perfect sensors and perfect collocation of sensors must be taken into account in any experimental or real system and further robustness should be built into the assignment of gain matrices. So we have seen that with positive definite output feedback we can guarantee asymptotic stability for a controllable system, and now we must explore a method of controlling the system to a designer's specific desires.

2.3 Eigenstructure Assignment

Now that we can guarantee stability, the next question is how to pick our gain matrices to give a desired eigenstructure. Junkins and others have used techniques to find an "optimal" solution in the sense that he minimized the condition number of his closed-loop eigenvalues [3] [4]. Others have used Linear Quadratic Regulator, or LQR, techniques as an optimization method [8] [1] [9]. This thesis will use pole placement techniques and minimize the difference between an actual and desired eigenstructure. The number of eigenvalues and eigenvectors to be placed is limited by the number of actuators and sensors used on a system. For every sensor with velocity and feedback measurements, the real and imaginary parts of a complex conjugate eigenvalue pair can be placed. To that end, if we wish to specify the eigenvalues only, we want to minimize the following cost function

$$J = \sum_{i=1}^{n} F_{e_i} (\lambda_{d_i} - \lambda_{a_i})^2 \tag{27}$$

where F_{e_i} is the i$^{\text{th}}$ eigenvalue weighting, λ_{d_i} is the i$^{\text{th}}$ desired closed-loop eigenvalue, and λ_{a_i} is the i$^{\text{th}}$ achievable closed-loop eigenvalue. Remember that the closed loop equations of motion were written in matrix form as

$$M\ddot{\mathbf{x}} + (C + \hat{C})\dot{\mathbf{x}} + (K + \hat{K})\mathbf{x} = 0. \tag{28}$$

If we rearrange it to first order form we get

$$\dot{\mathbf{z}} = \begin{bmatrix} 0 & I \\ -M^{-1}(K + \hat{K}) & -M^{-1}(C + \hat{C}) \end{bmatrix} \mathbf{z} = A\mathbf{z} \tag{29}$$

We can then find the eigenvalues of A by solving the characteristic equation

$$\det(\lambda_i I - A) = 0 \tag{30}$$

or we can let a software package solve the problem for us. This work utilized the software package MatLab™. These eigenvalues from Equation 30 are then compared to the designers desired eigenvalues and the cost function is minimized using a numerical algorithm which is found in many software packages.

If the designers have enough sensors and actuators they can also move the eigenvectors to a desired location. This is done by adding the eigenvectors to the cost function and replacing the previous one with

$$J = \sum_{i=1}^{n} \left[F_{e_i}(\lambda_{d_i} - \lambda_{a_i})^2 + (V_{d_i} - V_{a_i})^T F_{v_i}(V_{d_i} - V_{a_i}) \right] \tag{31}$$

where V_{d_i} is the i$^{\text{th}}$ desired eigenvector, V_{a_i} is the i$^{\text{th}}$ achievable eigenvector, and F_{v_i} is the i$^{\text{th}}$ eigenvector weighting matrix (usually a positive semi-definite diagonal matrix). Please note that the eigenvectors must be normalized by some method so that a comparison between the achieved and desired will be a comparison of merit. A derivation for one method of normalization can be found in Huckabone's thesis [1]. It is sufficient to say that his method normalizes the complex eigenvectors to a magnitude of one so that they can be compared with the normalized desired eigenvalues. Since this thesis utilized the software packages MatLab™, the normalization scheme is done automatically when the eigenvectors are calculated by the software.

The next chapter of this thesis will look at possible values for the eigenvalues using the technique of positive definite output feedback control. Then the equations in this theory chapter will be put to use in the robust eigenstructure assignment algorithm and the computer code will be explained.

Chapter 3 - Eigenvalue Properties

The previous chapter introduced positive definite output feedback control. This chapter will explore properties of eigenvalues for different systems. First, this chapter will review the properties of an undamped open-loop system. Then this work will move to the closed-loop system using only displacement output feedback control. In the closed-loop section, a proof is offered for the possible values of the controlled eigenvalues. The next section contains information about damping and how it affects the eigenvalues. That section will take a look at velocity output feedback control. Finally, the entire damped system with displacement and velocity feedback will be explored.

3.1 Open-loop System

Suppose we begin with a simple system describing a flexible structure without damping which will be designated as the open loop equations of motion

$$M\ddot{\mathbf{x}} + K\mathbf{x} = D\mathbf{u}. \tag{32}$$

where M and K are symmetric nxn matrices. Next, a modal coordinate transformation is introduced as follows:

$$\mathbf{x}(t) = \Phi\boldsymbol{\eta}(t) \tag{33}$$

where Φ is the open-loop modal matrix, and $\boldsymbol{\eta}$ is the vector of modal coordinates. The modal matrix is found by solving the following eigenvalue problem, which results from assuming a solution of the form $x_i = \phi_i e^{j\omega t}$ and $\mathbf{u} = 0$:

$$\omega_i^2 \phi_i = M^{-1} K \phi_i \tag{34}$$

for the n natural frequencies, ω_i, and the n eigenvectors, ϕ_i. Then the open-loop modal matrix is

$$\Phi = [\phi_1, \phi_2, \cdots, \phi_n] \tag{35}$$

Transforming the open-loop equations of motion in Equation 32 with Equation 33, we end up with the following equations:

$$M\Phi\ddot{\boldsymbol{\eta}} + K\Phi\boldsymbol{\eta} = D\mathbf{u}. \tag{36}$$

Then we pre-multiply by Φ^T to get

$$\Phi^T M \Phi \ddot{\boldsymbol{\eta}} + \Phi^T K \Phi \boldsymbol{\eta} = \Phi^T D \mathbf{u}. \tag{37}$$

Now in classical modal analysis the eigenvectors are assumed to be normalized such that

$$\Phi^T M \Phi = I \tag{38}$$

$$\Phi^T K \Phi = diag\left[\omega_i^2\right] = \Omega \tag{39}$$

So we end up with the following transformed equations of motion

$$\ddot{\boldsymbol{\eta}} + \Omega \boldsymbol{\eta} = \hat{D} \mathbf{u} \tag{40}$$

where $\hat{D} = \Phi^T D$. It can also be seen that the left hand side of Equation 40 is a set of n uncoupled equations of motion. Each of these equations can be solved separately by assuming a solution of the form

$$\boldsymbol{\eta} = \xi e^{j\lambda t} \tag{41}$$

which when substituted into Equation 40 with the right hand side equal to zero gives

$$-\lambda^2 \xi e^{j\lambda t} + \Omega \xi e^{j\lambda t} = 0. \tag{42}$$

Now divide by $e^{j\lambda t}$ to get the characteristic equations of each of the uncoupled equations

$$[\lambda^2 I - \Omega]\xi = 0. \tag{43}$$

So the eigenvalues (λ_i) equal the natural frequencies of the open-loop, undamped system (ω_i), and the eigenvectors, ξ_i, are the unit n^{th} order basis vectors. This development assumes that the eigenvectors from repeated eigenvalues have been orthogonalized.

3.2 Closed-loop System with Displacement Feedback Control

The purpose of this proof is to show that the following statement is true:

The natural frequencies of a structural system of equations can only increase when positive definite output feedback control is implemented.

This proof begins by defining the term positive definite and then gives two lemmas to support the overall proof.

3.2.1 Definitions and Lemmas

Definition 1 *Each of the following tests is a necessary and sufficient condition for the real symmetric matrix A to be positive definite: [10, 331]*

(I) $\mathbf{x}^T A\mathbf{x} > 0$ *for all nonzero vectors x.*

(II) *All the eigenvalues of A satisfy* $\lambda_i > 0$.

Lemma 1 *If G is positive definite then* $\Phi^T DGD^T\Phi$ *is positive semi-definite.*
Proof. *If G is positive definite then* $\mathbf{x}^T G\mathbf{x} > 0$ *for all* $\mathbf{x}$. *Now take* $\mathbf{x}=D^T\Phi$ *and* $\mathbf{x}^T G\mathbf{x} = \Phi^T DGD^T\Phi > 0$ *so it is positive definite unless* $D^T\Phi$ *has a zero, which causes it to be positive semi-definite.* ■

Lemma 2 *When eigenvalues of symmetric matrices A, B and C are* α_i, β_i *and* γ_i *respectively where all eigenvalues are arranged in non-increasing order and when* $C = A+B$, *then* $\alpha_s + \beta_n \leq \gamma_s \leq \alpha_s + \beta_1$. *So that when B is added to A then all of the eigenvalues are changed by an amount which lies between the smallest and greatest eigenvalues of B.*
Proof. *See Wilkinson. [11, 101-2]* ■

3.2.2 Problem Formulation

Suppose we begin again with the simple system describing a flexible structure without damping which we designated the open loop equations of motion

$$M\ddot{\mathbf{x}} + K\mathbf{x} = D\mathbf{u}. \tag{44}$$

As in previous sections, we will utilize positive definite output feedback control. However, we will begin this proof with only collocated, displacement feedback so that we have the following control law.

$$\mathbf{u} = -G_P\mathbf{y} \tag{45}$$

$$\mathbf{y} = D^T\mathbf{x} \tag{46}$$

Now combining these equations we have

$$M\ddot{\mathbf{x}} + K\mathbf{x} = -DG_P D^T\mathbf{x} \tag{47}$$

which will be designated as the closed loop equations of motion. Next, the modal coordinate transformation is introduced as previously shown

$$M\Phi\ddot{\eta} + K\Phi\eta = -DG_P D^T\Phi\eta \tag{48}$$

Then we bring the right hand side over to the left hand side

$$M\Phi\ddot{\eta} + (K + DG_P D^T)\Phi\eta = 0 \tag{49}$$

and pre-multiply by Φ^T to get:

$$\Phi^T M\Phi\ddot{\boldsymbol{\eta}} + \Phi^T(K + DG_P D^T)\Phi\boldsymbol{\eta} = 0 \tag{50}$$

Now with classical modal analysis the eigenvectors are normalized so that

$$\Phi^T M\Phi = I \tag{51}$$

$$\Phi^T K\Phi = \Omega \tag{52}$$

So we end up with the following transformed equations of motion:

$$\ddot{\boldsymbol{\eta}} + (\Omega + \Phi^T DG_P D^T\Phi)\boldsymbol{\eta} = 0 \tag{53}$$

Next assume that

$$\boldsymbol{\eta} = \Psi e^{j\hat{\omega}t} \tag{54}$$

then Equation 53 becomes, with some rearrangement,

$$\hat{\boldsymbol{\omega}}^2\Psi = (\Omega + \Phi^T DG_P D^T\Phi)\Psi \tag{55}$$

In the notation of Wilkinson (see Lemma 2) we then have the open-loop matrix from Equation 40

$$A \equiv \Omega, \tag{56}$$

the control matrix from the right hand side of Equation 48

$$B \equiv \Phi^T DG_P D^T\Phi, \tag{57}$$

and finally the closed-loop matrix

$$C \equiv [\Omega + \Phi^T DG_P D^T\Phi] \tag{58}$$

from Equation 53. The eigenvalues of A are designated α_i, which we showed in the previous section are equal ω_i^2. The eigenvalues of B are all non-negative since B is positive semidefinite. Finally, the eigenvalues of C are $\gamma_i = \hat{\omega}_i^2$ where $\hat{\omega}_i^2$ are the closed-loop eigenvalues. As a reminder of lemma 2, Wilkinson showed that when

$$C = A + B \tag{59}$$

then,

$$\alpha_s + \beta_n \leq \gamma_s \leq \alpha_s + \beta_1 \tag{60}$$

where β_n is the smallest eigenvalue of B and β_1 is the largest eigenvalue of B. The conservative case is when β_n is equal to zero, then we can say

$$\omega_s^2 \leq \gamma_s \leq \omega_s^2 + \beta_1 \tag{61}$$

However,

$$\gamma_s = \hat{\omega}_s^2 \tag{62}$$

where $\hat{\omega}_s$ are the closed-loop system natural frequencies. So Equation 61 becomes

$$\omega_s^2 \leq \hat{\omega}_s^2 \leq \omega_s^2 + \beta_1 \tag{63}$$

or taking the square root

$$\omega_s \leq \hat{\omega}_s \leq \sqrt{\omega_s^2 + \beta_1}. \tag{64}$$

Therefore, each of the natural frequencies of the controlled system lie on the real number line somewhere greater than the original open-loop natural frequencies and less than $\sqrt{\omega_s^2 + \beta_1}$. In other words, the magnitude of the natural frequencies will always increase. Their squares can increase up to the magnitude of the largest eigenvalue of the control matrix, B.

In a practical sense, the maximum natural frequency increment due to feedback, β_1, is limited by the control power for a given application.

3.3 Open-loop System with Damping

Now if the system has damping and/or velocity feedback control, then the result on the natural frequencies is similar, but not quite the same as above. Meirovitch states that

When the damping coefficients are small, a meaningful approximation can be obtained by using the modal matrix as the transformation matrix . . . This in effect implies that the uncoupled equations can be used when damping is small without causing serious errors. Physically this means that when damping is sufficiently small that coupling is a second order effect. [6, pg. 388-433]

When damping is sufficiently small, as it is in many structural applications, and we keep the velocity control matrix sufficiently small, we can approximate the natural frequencies with the open-loop undamped system of equations. As we have shown previously, the natural frequencies can only increase with the general output feedback control with only displacement control. So in the general case, the eigenvalues will follow the proof in the previous section up to second order effects.

A proof of the damping case can be found in Natsiavs' and Beck's work [7]. Natsiavs' proof showed a methodology for separating a damping matrix into a diagonalized matrix for standard modal analysis and a matrix with the diagonal terms equal to zero. This allows one to analyze a system using classical modal techniques and also find a second order correction term for small off-diagonal coupling. The following proof will use a similar methodology, but whereas Natsiavas used only small coupling effects, this proof will look at damping matrices where all of the damping elements can be considered small.

The previous section on natural frequencies only used displacement or position feedback. In this section we will concentrate on small amplitude velocity or rate feedback. Thus, the positive definite output rate feedback control law is written as

$$DG_RD^T = \epsilon C \tag{65}$$

where $\epsilon \ll 1$. Then, we start with the equations of motion with a small damping matrix

$$M\ddot{\mathbf{x}} + \epsilon C\dot{\mathbf{x}} + K\mathbf{x} = \mathbf{0}. \tag{66}$$

Now let λ_n^0 be the n^{th} eigenvalue and $\hat{\mathbf{x}}_n$ be the n^{th} eigenvector for the undamped problem

$$((\lambda_n^0)^2 M + K)\hat{\mathbf{x}}_n = \mathbf{0}. \tag{67}$$

Next construct the modal matrix $\Phi = [\hat{\mathbf{x}}_1 \cdots \hat{\mathbf{x}}_n]^T$ and normalize so that

$$\Phi^T M \Phi = I, \quad \Phi^T K \Phi = \Lambda \quad \Phi^T \epsilon C \Phi = \epsilon \hat{C}. \tag{68}$$

So we now have for the undamped system

$$((\lambda_n^0)^2 I + \Lambda)\mathbf{x}_n = 0 \tag{69}$$

Notice that the $\mathbf{x}_n$ are now the modal eigenvectors of the undamped system.

Using the original system in Equation 66 and using the modal transformation in Equation 68 the eigenvalue problem can be expressed as

$$((\lambda_n)^2 I + \lambda_n \epsilon \hat{C} + \Lambda)\mathbf{v}_n = 0. \tag{70}$$

Now, for a damping matrix of the form expressed by Equation 65, the eigenvalues and eigenvectors of Equation 66 are expressed to second order as

$$\lambda_n = \lambda_n^0 + \epsilon\lambda_n^1 + \epsilon^2\lambda_n^2 \tag{71}$$

$$\mathbf{v}_n = \mathbf{x}_n + \epsilon\mathbf{y}_n + \epsilon^2\mathbf{z}_n. \tag{72}$$

The notation of the past few equations requires some explination. The λ_n^0, λ_n^1 and λ_n^2 terms are the zeoreth, first and second order terms in ϵ of the eigenvalue. The term $(\lambda_n)^2$ denotes the eigenvalue squared. Therefore, using the expansion in Equation 71 the term $(\lambda_n)^2$ equals $(\lambda_n^0 + \epsilon\lambda_n^1 + \epsilon^2\lambda_n^2)^2$. Substituting Equations 71 and 72 into 70 gives

$$[(\lambda_n^0 + \epsilon\lambda_n^1 + \epsilon^2\lambda_n^2)^2 I + (\lambda_n^0 + \epsilon\lambda_n^1 + \epsilon^2\lambda_n^2)\epsilon\hat{C} + \Lambda](\mathbf{x}_n + \epsilon\mathbf{y}_n + \epsilon^2\mathbf{z}_n) = 0 \tag{73}$$

Collecting the terms with the same order of ϵ gives

$$((\lambda_n^0)^2 I + \Lambda)\mathbf{x}_n = 0 \tag{74}$$

$$((\lambda_n^0)^2 I + \Lambda)\mathbf{y}_n = -(2\lambda_n^0\lambda_n^1 I + \lambda_n^0\hat{C})\mathbf{x}_n \tag{75}$$

$$((\lambda_n^0)^2 I + \Lambda)\mathbf{z}_n = -(2\lambda_n^0\lambda_n^1 I + \lambda_n^0\hat{C})\mathbf{y}_n - [(2\lambda_n^0\lambda_n^1 + (\lambda_n^1)^2)I + \lambda_n^1\hat{C})\mathbf{x}_n. \tag{76}$$

The first of these equations is the same as Equation 69 using modal coordinates for the undamped system. So as expected with ϵ to the zeroth order we have no damping and hence we have the equation for the undamped system. Examining the $\mathrm{n^{th}}$ row of Equation 75 we notice the left hand side $((\lambda_n^0)^2 I + \Lambda)$ is a diagonal matrix that by definition has a zero at its $\mathrm{n^{th}}$ diagonal element. This element corresponds to the modal eigenvector, $\mathbf{x}_n$, which is equal to zero every where except at its $\mathrm{n^{th}}$ element, which equals one. In other words, on the right hand side the modal eigenvectors,$\mathbf{x}_n$, will pick off only the $\mathrm{n^{th}}$ diagonal elements of $(2\lambda_n^0\lambda_n^1 I + \lambda_n^0\hat{C})$ and the left hand side will be equal to zero. So

$$2\lambda_n^0\lambda_n^1 I + \lambda_n^0\hat{C}_{nn} = 0 \tag{77}$$

or,

$$\lambda_n^1 = -\frac{\hat{C}_{nn}}{2} \tag{78}$$

Therefore, to order ϵ we have

$$\lambda_n = \lambda_n^0 - \epsilon\frac{\hat{C}_{nn}}{2} \tag{79}$$

But,

$$\lambda_n^0 = \pm j\omega_n \tag{80}$$

so to first order in ϵ

$$\lambda_n = -\epsilon\frac{\hat{C}_{nn}}{2}\lambda_n \pm j\omega_n. \tag{81}$$

An eigenvalue of the form

$$\lambda = -\alpha \pm j\beta \tag{82}$$

has a damped natural frequency, ω_d

$$\omega_d = \beta \tag{83}$$

and a damping factor, ζ

$$\zeta = \frac{1}{\sqrt{\left(\frac{\beta}{\alpha}\right)^2 + 1}} \tag{84}$$

Consequently, for the damped system to first order in ϵ,

$$\omega_d = \omega_n \tag{85}$$

and,

$$\zeta = \frac{1}{\sqrt{\left(\frac{2\omega_n}{\epsilon\hat{C}_{nn}}\right)^2 + 1}} \tag{86}$$

Expanding the above in a Taylor series and retaining only the terms of order ϵ we get

$$\zeta = \epsilon\frac{\hat{C}_{nn}}{2\omega_n} \tag{87}$$

This has shown, as Meirovitch stated, that when damping is sufficiently small, its effect on the damped natural frequency is second order. Thus we can rely on the proof in the previous section which gave the constraint that the undamped natural frequencies (and our damped natural frequencies as well due to Equation 85) of the closed-loop system will always increase compared to the open-loop natural frequencies. Therefore, the answer to our original question: do the natural frequencies of a structural system of equations only increase when positive definite output feedback control is implemented, is yes if the positive definite rate feedback matrix G_R is sufficiently small.

Chapter 4 - Eigenstructure Assignment

This chapter will explore the computer code used in this thesis. First, we will summarize the equations developed in earlier chapters of this thesis. Next, we will look at the programing considerations for this work, including the existing MatLab$^{\text{TM}}$ routines, as well as, new routines developed for this thesis. Then, we will look at the algorithms to develop this code in any computer language. Finally we will examine the usage of the ear.m program.

4.1 Algorithm Equations

The main equations in the program have been developed through this thesis. The main equations used in the program ear.m begin with the open-loop equations of motion

$$M\ddot{\mathbf{x}} + C\dot{\mathbf{x}} + K\mathbf{x} = \mathbf{0} \tag{88}$$

and end with the achieved closed-loop equations of motion

$$M\ddot{\mathbf{x}} + (C + DG_RD^T)\dot{\mathbf{x}} + (K + DG_PD^T)\mathbf{x} = \mathbf{0}. \tag{89}$$

The eigenvalues and eigenvectors are calculated by writing these equations in first order form

$$\dot{\mathbf{z}} = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}C \end{bmatrix} \mathbf{z} \tag{90}$$

and,

$$\dot{\mathbf{z}} = \begin{bmatrix} 0 & I \\ -M^{-1}(K + DG_PD^T) & -M^{-1}(C + DG_RD^T) \end{bmatrix} \mathbf{z} \tag{91}$$

and then using the existing MatLab$^{\text{TM}}$ subroutine eig.m for the calculation. The program ear.m then plots the resulting eigenvalues.

The cost function for just the eigenvalues is

$$J = \sum_{i=1}^{n} F_{e_i}(\lambda_{d_i} - \lambda_{a_i})^2 \tag{92}$$

and the cost function for both eigenvalues and eigenvectors is

$$J = \sum_{i=1}^{n} \left[F_{e_i}(\lambda_{d_i} - \lambda_{a_i})^2 + (V_{d_i} - V_{a_i})^T F_{v_i}(V_{d_i} - V_{a_i}) \right]. \tag{93}$$

These are the main equations for the program. Next, we will look at the programing considerations using MatLab$^{\text{TM}}$ and these main equations to get a useful output.

4.2 Programing Considerations

As stated in the introduction, previous theses by Robinson [8] and Huckabone [1] give algorithms for eigenvalue assignment using linear quadratic regulator (LQR) techniques. Robinson used MatLab^{TM} and the LQR routines to minimize a cost function of weighted eigenvalues, and then Huckabone converted the algorithm to FORTRAN to save computer time and used a cost function that included both the eigenvalues and the eigenvectors. He noticed a ten fold increase in compute time for the FORTRAN algorithm over Robinson's MatLab^{TM} routines.

Robinson wrote his code in 1990 and Huckabone wrote the FORTRAN code in 1991. It has been nine years since Robinson's thesis, and the current law of compute time is that processor speed doubles every eighteen months. So, since Robinson's time we should have doubled six times. He was able to use a Compaq 286, while at the time of this writing the state of the art for a personal computer is a Pentium II 450 MHz processor from Intel. Huckabone's FORTRAN code gave him a ten times increase over Robinson's MatLab^{TM} code. However, with computer speeds of today, most of the example problems in this thesis ran in about three to five seconds, and the examples with a four degree-of-freedom system took less than two minutes.

The algorithm for this thesis is similar to the algorithms used by both Robinson and Huckabone. The difference is in the control method. Robinson and Huckabone used the linear quadratic regulator while this thesis has explored the use of positive definite output feedback control.

4.2.1 Existing Subroutines

The main program structure and cost function structure were taken from Huckabone [1], but MatLab^{TM} has many useful existing subroutines which proved to be advantageous for this work.

The main subroutine used by the program ear.m is fmins.m. The subroutine fmins.m uses a Nelder-Mead type simplex search method and the inputs are the cost function name and a vector of initial conditions. The subroutine eig.m was used many times in the program. That subroutine finds the eigenvalues and eigenvectors of a matrix. Also, the subroutine sort.m sorts a vector in ascending order of magnitude. These subroutines work within the ear.m main program to optimize the difference between the desired eigenstructure and the achieved eigenstructure. The cost function is minimized within fmins.m and the result is sent back to the main program.

4.2.2 Newly Developed Subroutines

The subroutines developed for this thesis and used in the main program ear.m are getdata.m, eigsort.m, plotinit.m, value.m, and structure.m. These subroutines must be in the same directory as the main program ear.m for the program to run.

The subroutine getdata.m is the input file and is fully explained in the section "Using the program." Next, eigsort.m is a subroutine that takes the desired eigenvalues, eigenvectors, and weighting matrices for the eigenstructure assignment algorithm and sorts the eigenvalues in ascending order with the sort routine discussed above. Then, it sorts the desired eigenvectors and weighting matrices so that they match the originally intended eigenvalues.

The routine plotinit.m initializes the plotting area for the graphical output of this program. The routine uses the magnitude of the largest desired eigenvalue to scale the axis for the final output.

Finally, the subroutines value.m and structure.m are the cost functions used by fmins.m. These two routines are discussed in detail in the following sections.

4.2.3 Program Flow

The program for this thesis has a main program and two different cost functions. The main program has an algorithm as follows:

4.2.3.1 Main Program

Input data from file

Variables:

Mass Matrix

Stiffness Matrix

Damping Matrix

Control Matrix

Desired Eigenvalues

Desired Eigenvectors

Weighting Factors

Sort the eigenvalues and eigenvectors by eigenvalue magnitude

Plot the open-loop eigenvalues

Initialize gain matrices

Call "fmins.m" to minimize cost function

Plot the closed-loop eigenvalues

4.2.3.2 Cost function for eigenvalues only

Both of the cost functions in this program are called by the MatLab™ subroutine fmins.m. The first cost function calculates the difference between the achievable and desired eigenvalues using the following algorithm:

Form gain matrices

Find eigenvalues

Sort eigenvalues

Plot eigenvalues

Calculate cost function

4.2.3.3 Cost function for both eigenvalues and eigenvectors

The second cost function calculates the difference between the achievable and desired eigenvalues and eigenvectors. The algorithm is similar to the first cost function but adds a step for the eigenvectors as shown in the following algorithm:

Form gain matrices

Find eigenvalues and eigenvectors

The M, C, K, D matrices are the mass, damping, stiffness, and control matrices. These depend on the characteristics of the open-loop system, and as we will examine with some examples, the D matrix depends on the placement of the collocated sensor-actuator pairs. The D matrix must have the same number of rows as the mass, stiffness, and damping matrices, and it must have the same number of columns as sensor-actuator pairs.

The ed vector contains the desired eigenvalues. Please note that in the example section, and in most cases, the desired eigenvalues will occur in complex conjugate pairs. The F vector is the weighting vector for the desired eigenvalues. This can be used by the designer to induce the program to direct its search toward the most desired eigenvalues. If the F variable is left blank, then the ear.m program will automatically assign ones for all the eigenvalues so that each one is weighted with the same values.

The vecd matrix is a matrix of the desired eigenvectors. The program will automatically normalize the eigenvectors, but the user must ensure that the eigenvectors are entered in the same order as the desired eigenvalues. Finally, Fvec is a weighting matrix for the desired eigenvectors. Usually this will be the identity matrix when using the structure cost function. If Fvec is set to a default of zeros when using the structure cost function, then the program will only consider the eigenvalues in the cost function. This is exactly what is done in the value cost function, but the numerical subroutine does not even attempt to calculate the eigenvector part of the cost function so that compute time will be shorter.

In the final chapter of this thesis we will explore some of these programing considerations, as well as demonstrate the usefulness of this program.

Chapter 5 - Examples

This chapter will begin with some simple examples and move to more complex models. The first step is to look at a single degree-of-freedom (DOF) system and to explore some of the eigenvalue properties proved in this thesis. Then, we will move on to a second order system and investigate some of the properties of the computer code. As a final validation step, we will look at a four degree-of-freedom system and see how the computer code handles higher order systems. Finally, we will look at a flexible truss example.

5.1 Single DOF system

The single spring-mass system is the simplest example of how the eigenvalue assignment algorithm works. First, let us take an example that graphically looks like Figure 2.

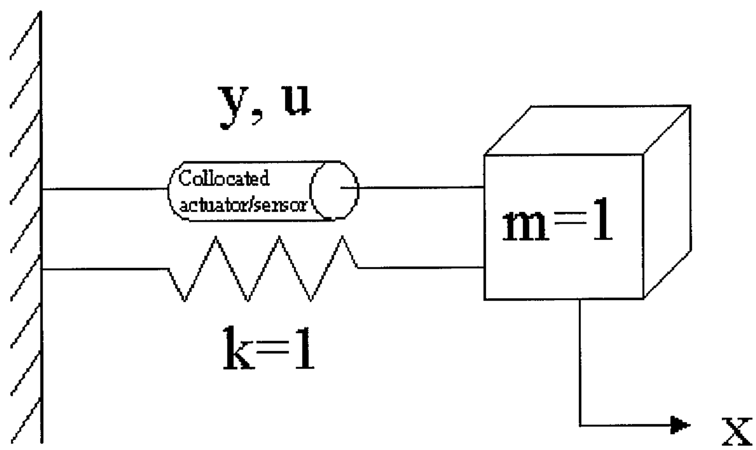

Figure 2. Single DOF spring-mass system

The mathematical equation of motion for the system in Figure 2 is

$$\ddot{x} + x = u. \tag{94}$$

The open-loop characteristic equation is

$$\lambda^2 + 1 = 0 \tag{95}$$

so the eigenvalues of this equation are

$$\lambda_{OL} = \pm i. \tag{96}$$

If we close the loop, the new equation of motion is

$$\ddot{x} + G_R\dot{x} + (1 + G_P)x = 0. \tag{97}$$

Then the closed-loop characteristic equation is

$$\lambda^2 + G_R\lambda + (1 + G_P) = 0, \tag{98}$$

and, from the quadratic equation, the closed-loop eigenvalues are

$$\lambda_{CL} = \frac{-G_R \pm \sqrt{G_R^2 - 4(1 + G_P)}}{2}. \tag{99}$$

If we desire closed-loop eigenvalues of -1 $\pm\, i$, then in this single DOF problem it can be verified using Equation 99 that $G_R = 2$ and $G_P = 1$. Note that with higher order systems we will not be able to solve for the gain matrices in closed form.

Therefore, in the computer program, we enter the following for the single DOF system shown in Table 1:

M	C	K	D
1	0	1	1

Table 1. Single DOF system properties

Now, when the program ear.m is asked to find the desired eigenvalues of -1 $\pm\, i$, the output is shown in Figure 3. The x marks on the imaginary axis are the open-loop poles of the system at $\pm i$ There are asterisks at the desired values of -1 $\pm\, i$. In this case there are also circles at -1 $\pm\, i$ to indicate the achieved closed-loop poles. The dots on the figure indicate the intermediate steps of the numerical gradient search.

Table 2 corresponding to Figure 3, which compares the desired with the achieved eigenvalues, and shows the position and rate gains, as well as the natural frequency and damping of the achieved poles.

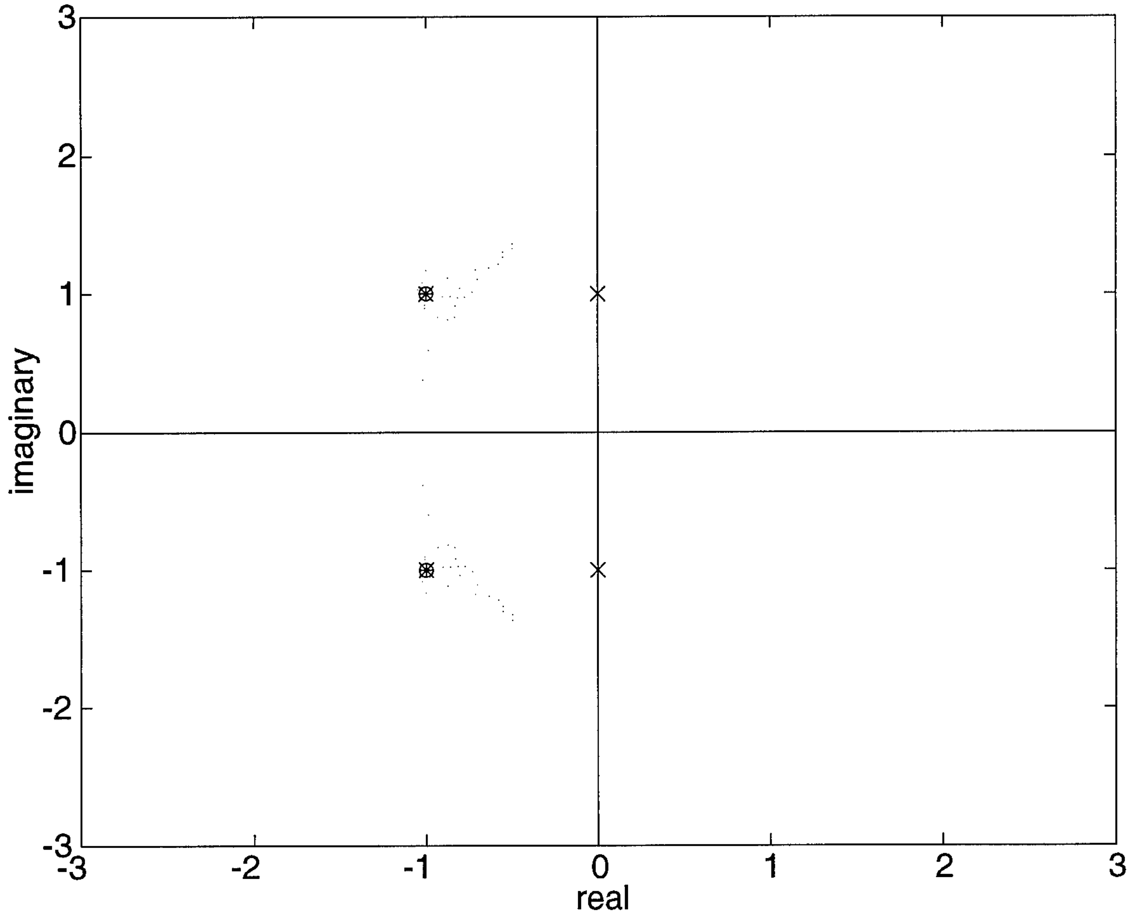

Figure 3. One degree-of-freedom system

G_P	G_R	desired	achieved	nat. freq.	damping
1	2	$-1 \pm i$	$-1 \pm i$	1.4142	0.7071

Table 2. One degree-of-fredom results

Now, since we have seen how this computer code works, let us look at problems we might encounter. The first problem was presented in the proof which showed that the natural frequency can only increase. Let the damping remain the same, but let us choose a set of desired eigenvalues with a natural frequency smaller than the open-loop natural frequency of one. For this example, the desired eigenvalues chosen were -0.5 $\pm$ $0.5i$. While these eigenvalues have the same damping as the previous example, the desired natural frequency is the magnitude of the eigenvalue or 0.7071.

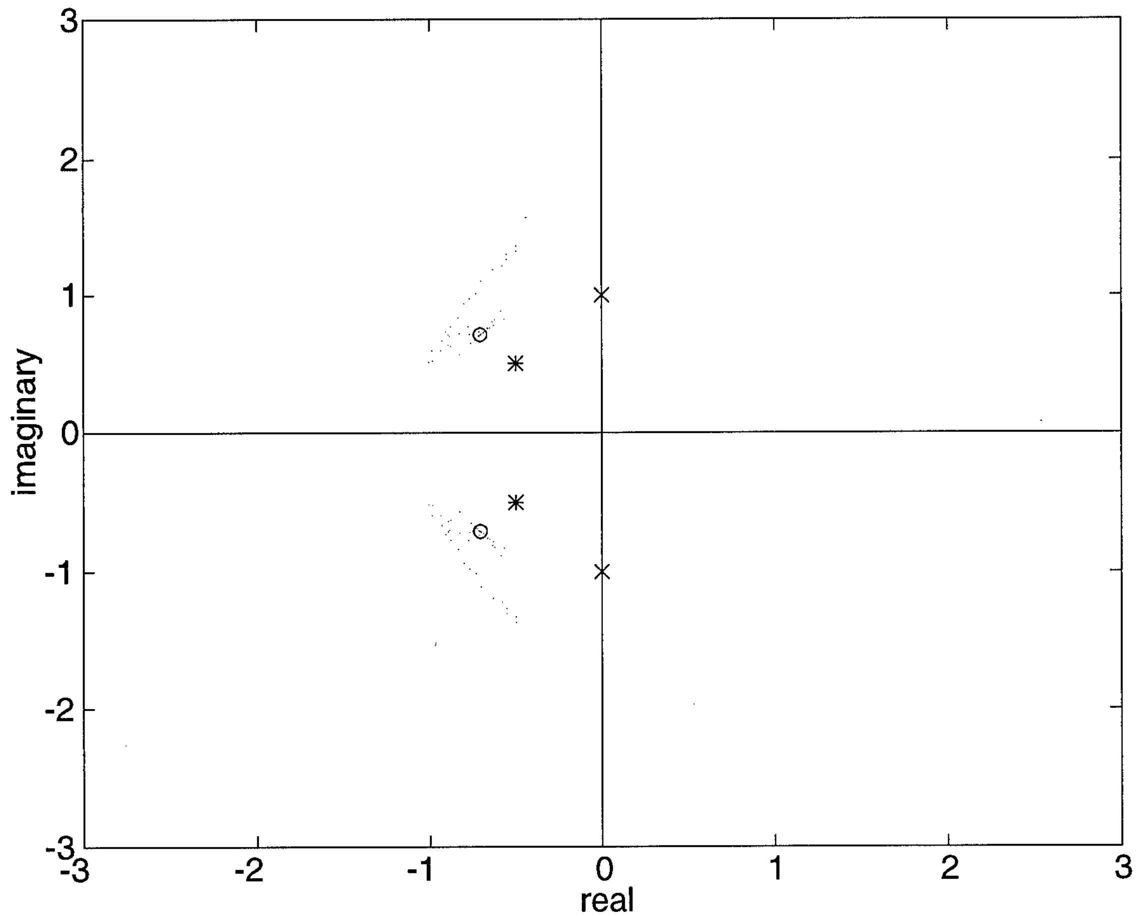

Figure 4. One degree-of-freedom with small desired poles

As can be seen in Figure 4 and Table 3, we did not achieve our desired eigenvalues. In fact, notice that the gradient search found the closest value to the desired poles with the constraint that the natural frequency could not be less than one.

Next, just as a precaution we will test the computer algorithm with very large desired eigenvalues in Figure 5 and Table 4.

The previous case shows that the eigenvalues can be placed a relatively large distance from the open-loop position. However, notice how large the gain matrices had to become to achieve the larger eigenvalues. As a practical matter, the control designer must always ensure that a design does not exceed the control power available for a given system.

Now that we see the achievable space for the desired eigenvalues, we can examine the effects of damping. Remember that our proof made the assumption that we would have small damping so that we could prove that damping effects were a second order perturbation to the eigenvalues. First, we will look at the response when we desire large damping.

In Figure 6 and Table 5, we asked for poles at $-0.5 \pm 0.1i$ with a large amount of damping, $\zeta = 0.9806$, and we also desired a natural frequency of 0.5099 which we know, since it is less than 1, cannot be achieved. The numerical algorithm again ended up at the closest point to the desired eigenvalues while constraining the natural frequency to be greater than 1. This means that even with large damping our proof still held. This would be a very interesting discovery since our closed-loop damping matrix, $C + DG_RD^T$, is larger than either the mass or stiffness matrix. However, we must contain our enthusiasm until we can explore a model with more than one DOF.

Finally, we can ensure that the program will handle small damping by asking for poles at $-0.1 \pm 0.5i$. This can be seen in Figure 7

G_P	G_R	desired	achieved	nat. freq.	damping
0	1.4142	$-0.5 \pm 0.5i$	$-0.7071 \pm 0.7071i$	1	0.7071

Table 3. One degree-of-freedom with small desired results

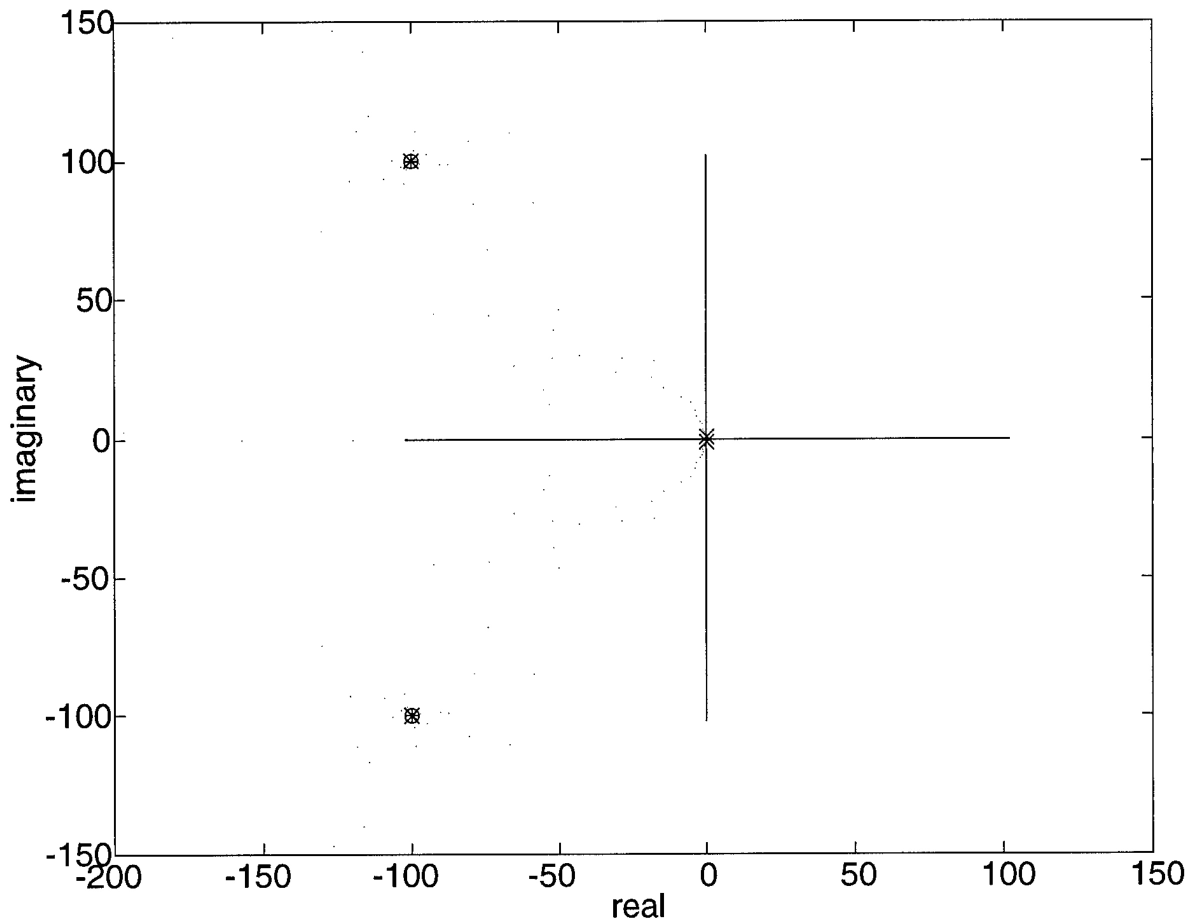

Figure 5. One degree-of-freedom with large desired eigenvalues

G_P	G_R	desired	achieved	nat. freq.	damping
19999	200	$-100 \pm 100i$	$-100 \pm 100i$	141.4214	0.9806

Table 4. One degree-of-freedom with large desired results

G_P	G_R	desired	achieved	nat. freq.	damping
0	1.9611	$-0.5 \pm 0.1i$	$-0.9806 \pm 0.1962i$	1	0.9806

Table 5. One degree-of-freedom with large damping results

G_P	G_R	desired	achieved	nat. freq.	damping
0	.3922	$-0.1 \pm 0.5i$	$-.1961 \pm .9806i$	1	0.1961

Table 6. One degree-of-freedom with small damping results

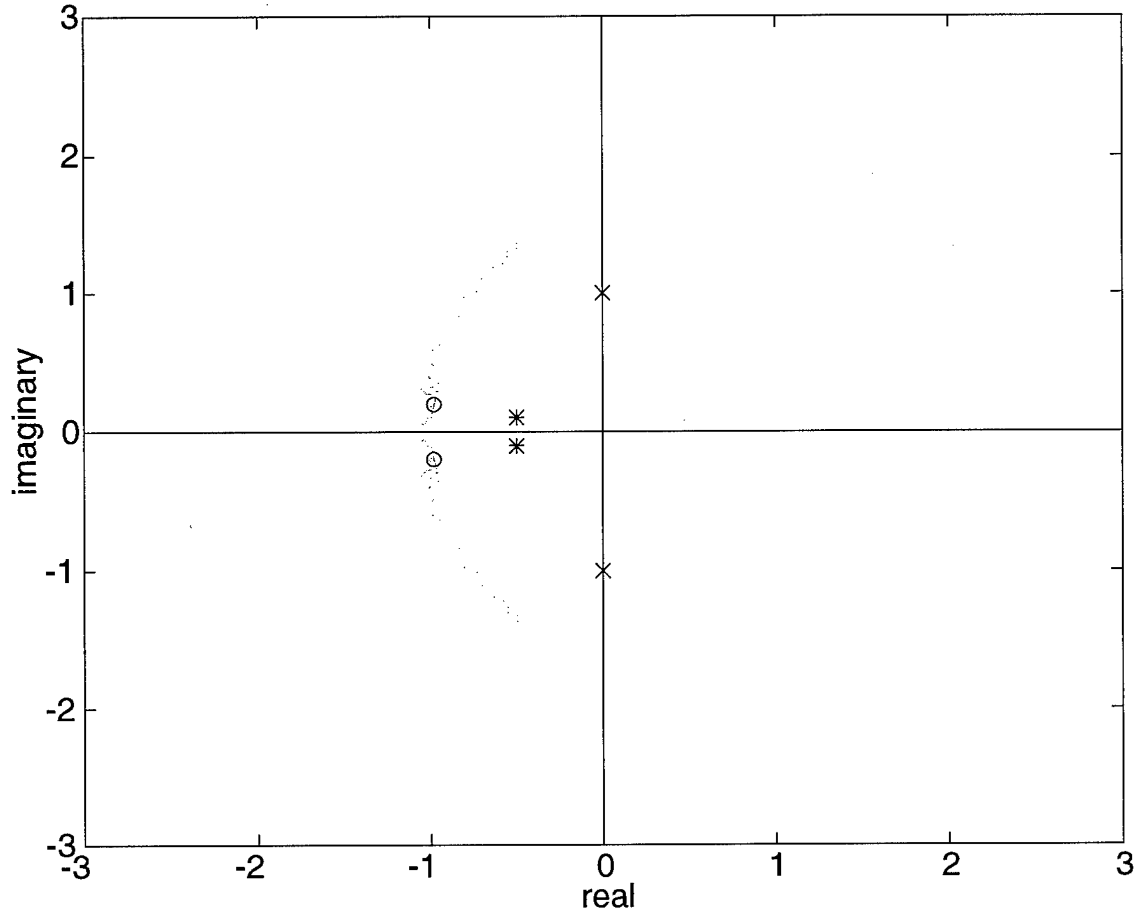

Figure 6. One degree-of-freedom system with large damping

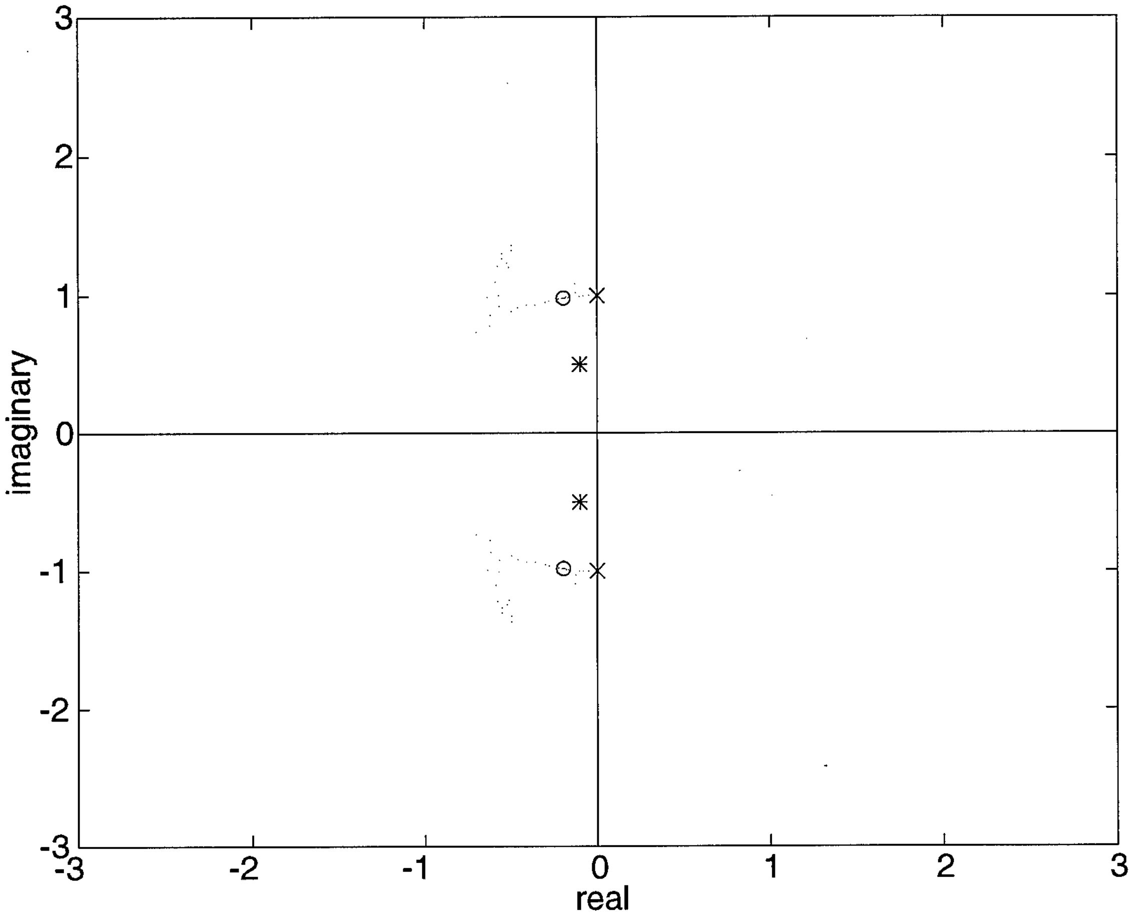

Figure 7. One degree-of-freedom system with small damping

5.2 Two DOF system

A two DOF system is best introduced using a model with two masses and two springs. This system, with collocated actuator-sensor pairs located between the masses, is shown in Figure 8.

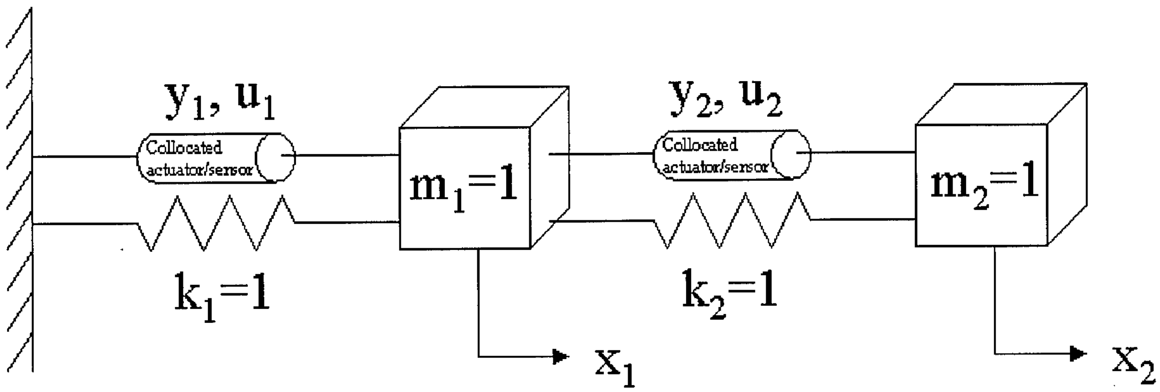

Figure 8. Two DOF spring-mass system

The equations of motion for this system are constructed using the values in Table 7.

The open-loop eigenvalues are on the imaginary axis at $\pm.681i$ and $\pm1.681i$. The first step is to ensure that we can move the poles to a location with damping and a larger natural frequency. This example is shown in Figure 9 and Table 8.

Again, we see that the numerical algorithm has no problem with the baseline system. The next two figures will validate the program. First, in Figure 10 and Table 9, we desire eigenvalues with larger natural frequencies than either of the open-loop eigenvalues.

Second, in Figure 11 and Table 10, we examine the result of desired eigenvalues which are both smaller than the open-loop eigenvalues.

Notice that we did not achieve the desired eigenvalues, but also notice that the natural frequency of the second eigenvalue decreased. Our rule that the natural frequency would not get smaller than the open-loop natural frequencies did not hold in this case. There are two reasons for this result. The first is the large amount of damping present in the second pole. Another reason

M	C	K	D
$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$

Table 7. Two DOF system properties

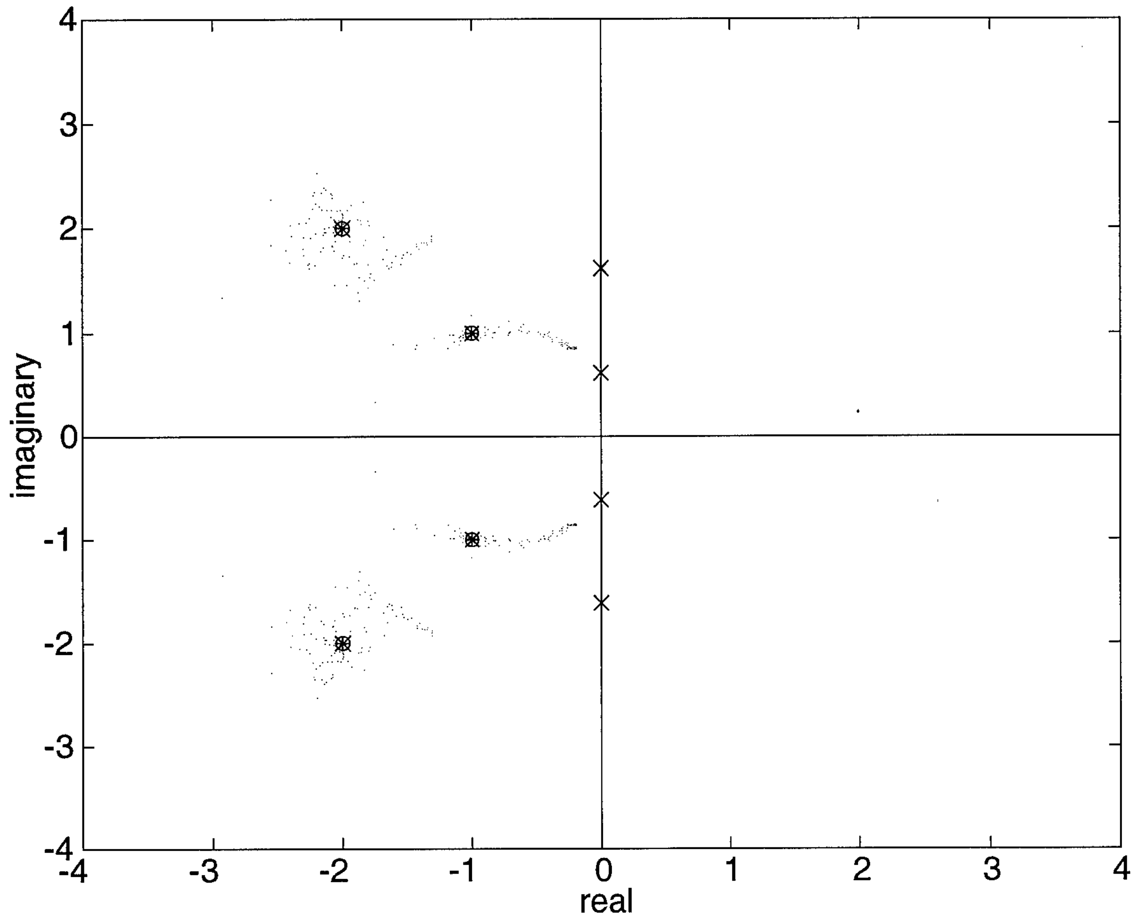

Figure 9. Two degrees-of-freedom system

G_P	G_R	desired	achieved	nat. freq.	damping
$\begin{bmatrix} 1.9303 & -0.0074 \\ -0.0074 & 4.4599 \end{bmatrix}$	$\begin{bmatrix} 3.8137 & -0.0089 \\ -0.0089 & 1.0843 \end{bmatrix}$	$-1 \pm i$	$-1 \pm i$	1.4142	0.7071
$det = 8.6089$	$det = 4.1349$	$-2 \pm 2i$	$-2 \pm 2i$	2.8284	0.7071

Table 8. Two degrees-of-freedom results

G_P	G_R	desired	achieved	nat. freq.	damping
$\begin{bmatrix} 8.4956 & -0.0615 \\ -0.0615 & 14.1652 \end{bmatrix}$	$\begin{bmatrix} 7.0519 & -0.059 \\ -0.059 & 1.433 \end{bmatrix}$	$-2 \pm 2i$	$-2 \pm 2i$	2.8284	0.7071
$det = 120.3373$	$det = 10.0500$	$-3 \pm 3i$	$-3 \pm 3i$	4.2426	0.7071

Table 9. Two degrees-of-freedom with larger desired results

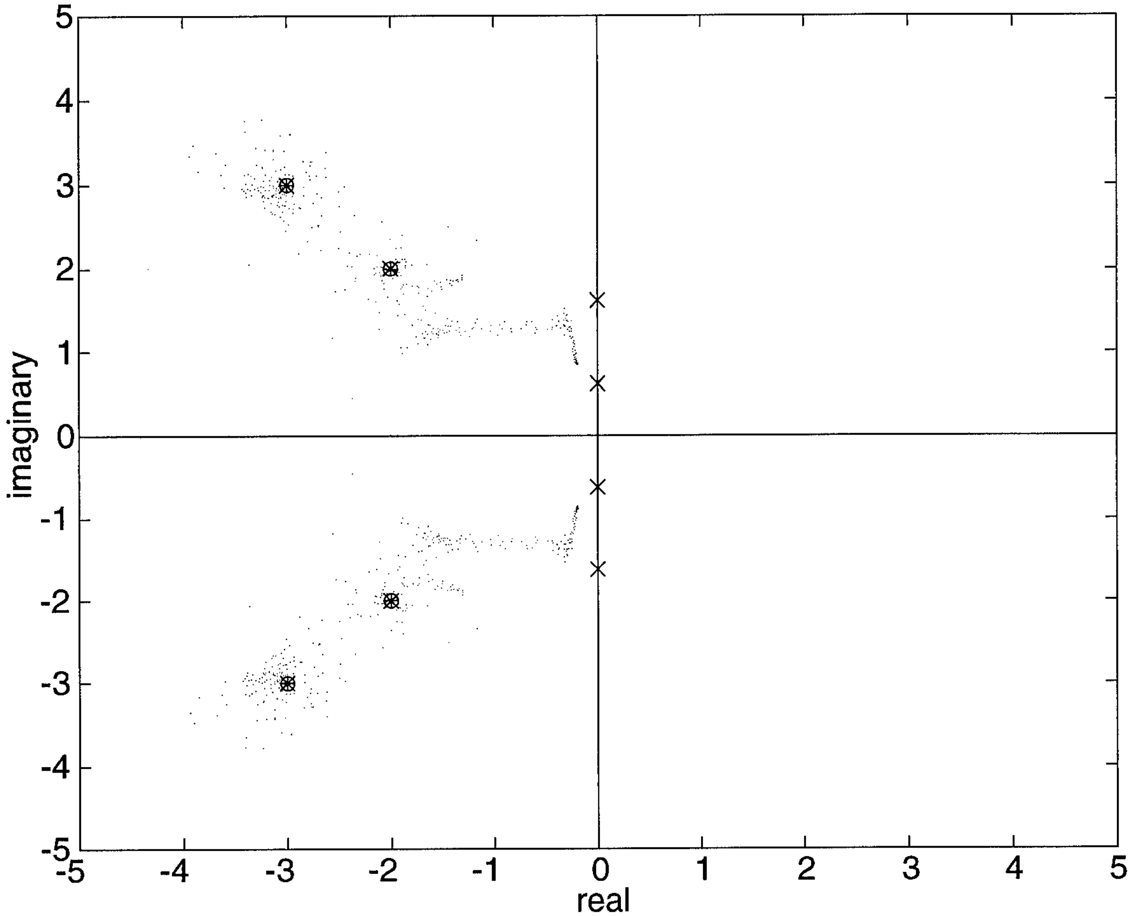

Figure 10. Two degrees-of-freedom with larger desired eigenvalues

G_P	G_R	desired
$10^{-3} * \begin{bmatrix} 0.1164 & 0.1088 \\ 0.1088 & 0.1016 \end{bmatrix}$	$\begin{bmatrix} 0.0001 & 0.0000 \\ 0.0000 & 1.2558 \end{bmatrix}$	$-0.2 \pm 0.2i$
$det = 1.4249 * 10^{-13}$	$det = 7.8972 * 10^{-5}$	$-0.3 \pm 0.3i$

achieved	nat. freq.	damping
$-0.0479 \pm 0.6518i$	0.6535	0.0732
$-1.2079 \pm 0.9396i$	1.5303	0.7893

Table 10. Two degee-of-freedom with small desired results

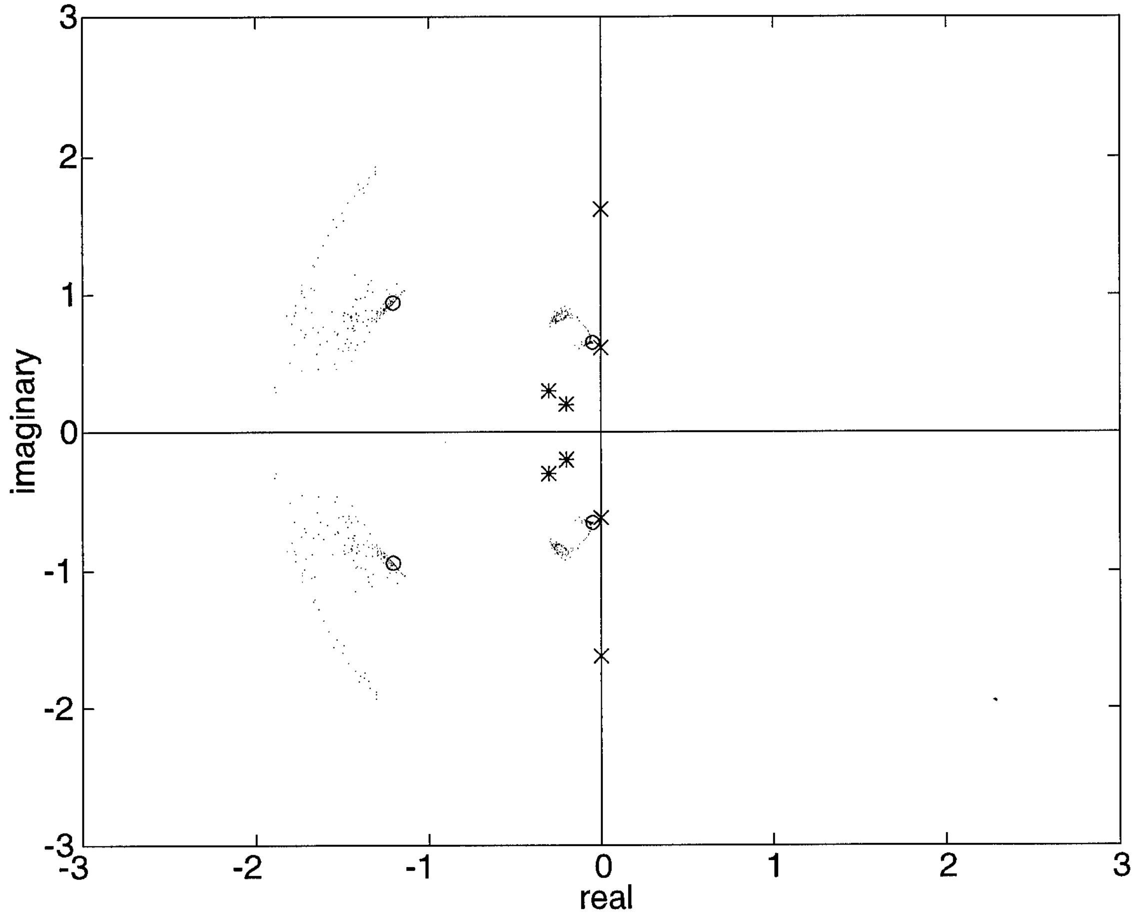

Figure 11. Two degrees-of-freedom with smaller desired eigenvalues

for this discrepancy is numerical error. When the algorithm is performing a gradient search with numbers on the order of 10^{-4}, truncation errors must be considered. The way to avoid this problem is for the designer to first find the open-loop eigenvalues and make sure that all of the desired eigenvalues have a larger natural frequency than the open-loop natural frequencies.

Now, let us turn to large damping, as we did with the single DOF system. In Figure 12 and Table 11, the natural frequencies of the desired eigenvalues are slightly smaller than the natural frequencies of the open-loop system, but the desired damping is very large.

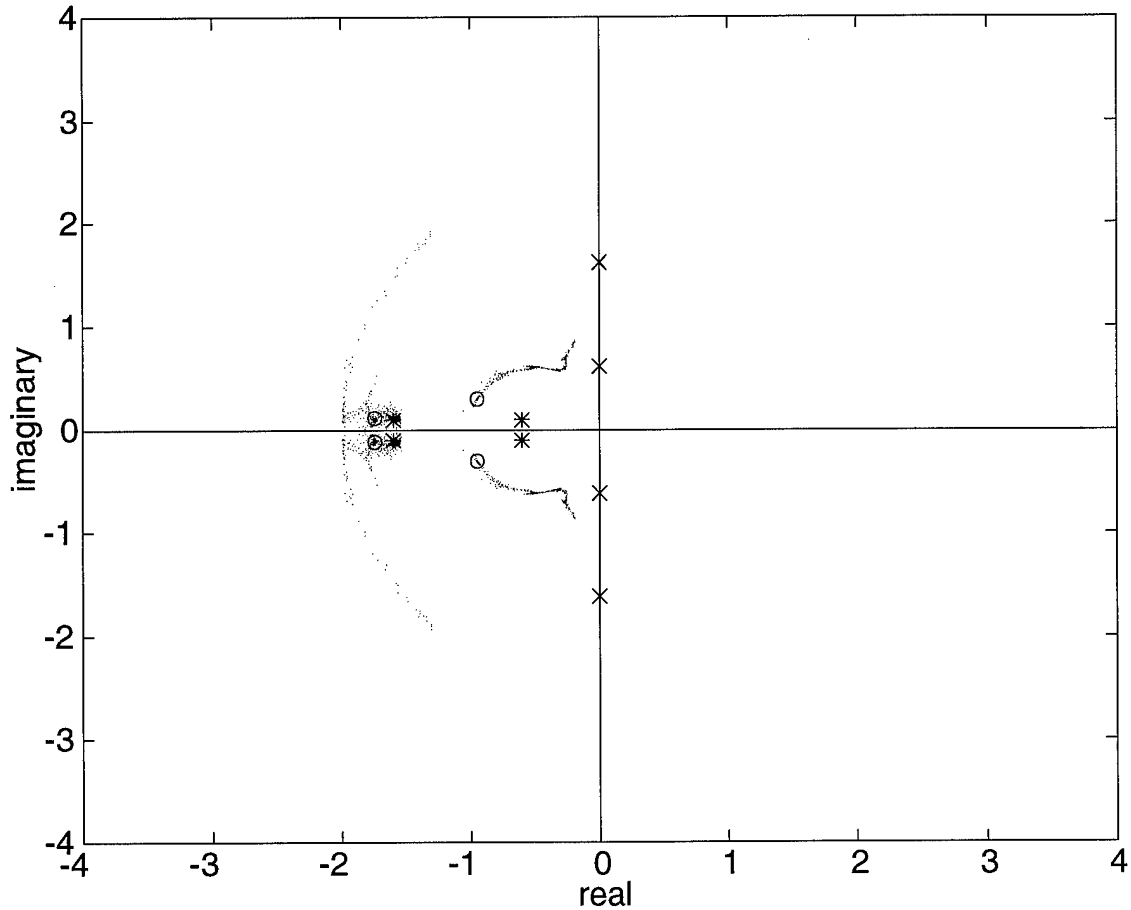

Figure 12. Two degrees-of-freedom with large damping desired

Notice how neither of the natural frequencies was smaller than the open-loop natural frequencies of 0.618 and 1.618. Also, the smaller pair of eigenvalues did not even move close to the desired values. So, while the single DOF model in the previous section showed a lower bound on the closed-loop natural frequency equal to that of the open-loop natural frequency, the rule does not hold with higher order models and large damping factors. In this case, the lower bound is larger than the predicted value of the proof. The difference is the large damping in G_R which makes damping larger than just a second order effect.

The next area to explore is what happens when we have a different number of sensors than the order of the system. First, we will look at the result when we use three sensors. Then we will see what happens when we have only one sensor with this two DOF system. In the rest of the examples, the desired natural frequencies will always be larger than the open-loop natural frequencies. Also, the desired frequencies will be chosen so as not to be larger than the next greater open-loop eigenvalue as was done in Figure 10.

The three sensor case would look graphically like Figure 13.

G_P	G_R	desired	achieved	nat. freq.	damping
$\begin{bmatrix} 0.0000 & 0.0000 \\ 0.0000 & 2.0307 \end{bmatrix}$	$\begin{bmatrix} 2.5978 & -0.0045 \\ -0.0045 & 1.3908 \end{bmatrix}$	$-1 \pm .1i$	$-0.9491 \pm 0.3002i$	0.9954	0.9534
$det = 6.3270 * 10^{-7}$	$det = 3.6129$	$-2 \pm .1i$	$-1.7451 \pm 0.1148i$	1.7489	0.9978

Table 11. Two degrees-of-freedom with large damping results

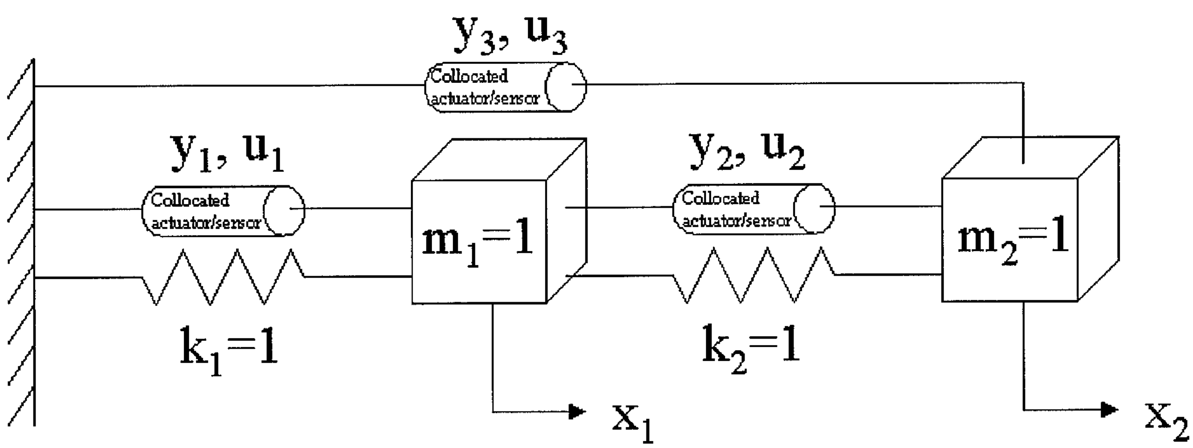

Figure 13. Two DOF system with three sensors

The result can be seen in Figure 14 and Table 12 .

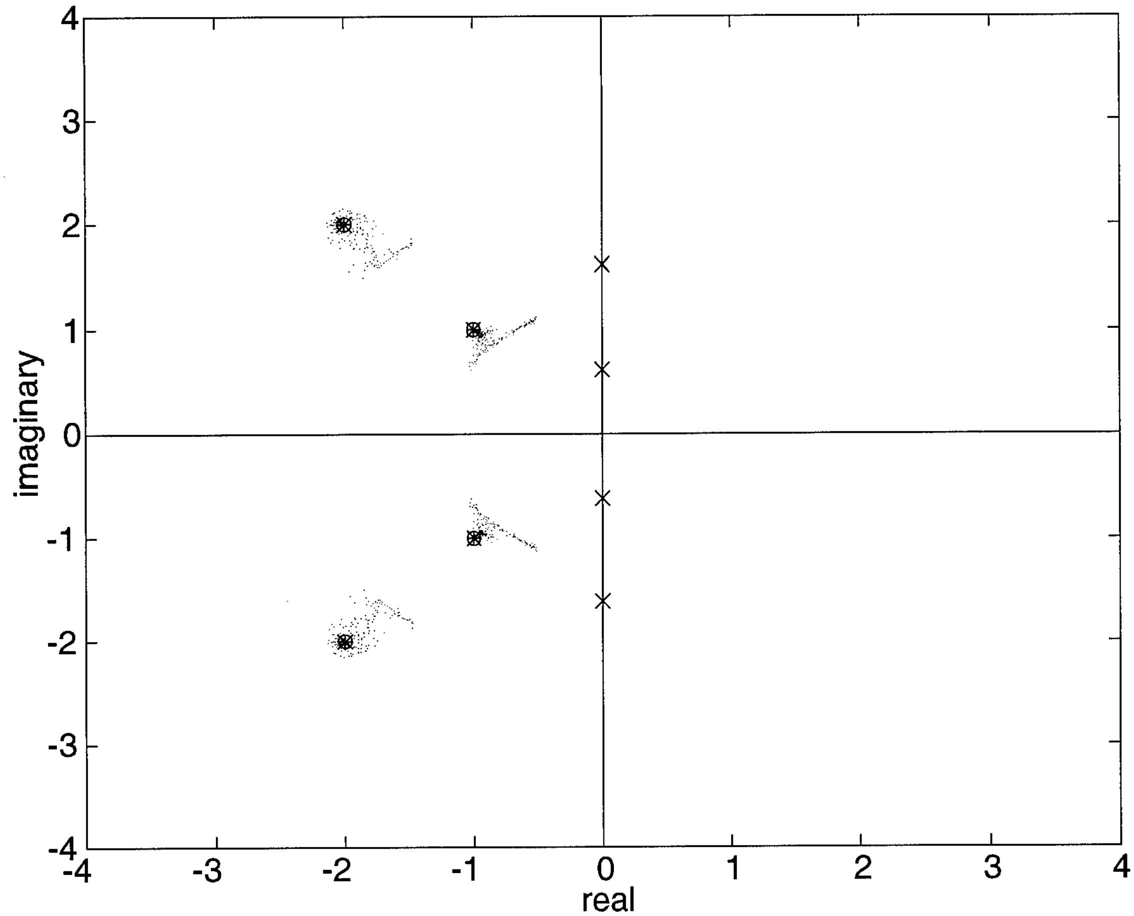

Figure 14. Two degrees-of-freedom with three sensors

The difference between this example and the example with two sensors is that the numerical gradients search narrows in on the desired eigenvalues much more quickly. Also, with an additional sensor, the gain matrices are larger so that they have more values to iterate with in the numerical minimization routine. In actual systems, the designer will usually have the opposite of this case, where there are fewer sensors than eigenvalues to control.

The first example, with only one sensor on a two DOF system, has the sensor between the two masses as in Figure 15.

This sensor placement results in Figure 16 and Table 13.

G_P	G_R	desired
$\begin{bmatrix} 1.5718 & 0.0000 & -0.0008 \\ 0.0000 & 2.0185 & -0.0011 \\ -0.0008 & -0.0011 & 1.4737 \end{bmatrix}$	$\begin{bmatrix} 2.4394 & -0.0035 & -0.0021 \\ -0.0035 & 0.9629 & -0.0019 \\ -0.0021 & -0.0019 & 1.6382 \end{bmatrix}$	$-1 \pm i$
$det = 4.6747$	$det = 3.8478$	$-2 \pm 2i$
achieved	nat. freq.	damping
$-1 \pm i$	$-1 \pm i$	.7071
$-2 \pm 2i$	2.8284	.7071

Table 12. Two degrees-of-freedm with three sensors results

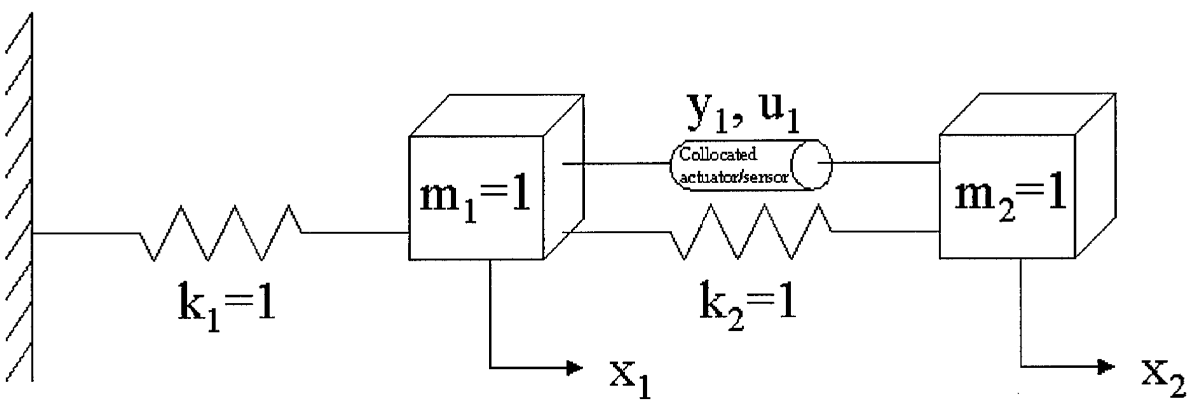

Figure 15. Two DOF system with one sensor

G_P	G_R	desired	achieved	nat. freq.	damping
2.7560	2.0070	$-1 \pm i$	$-0.0078 \pm 0.6861i$	0.6861	0.0114
		$-2 \pm 2i$	$-1.9992 \pm 1.9955i$	2.8247	0.7078

Table 13. Two degrees-of-freedom with one sensor results

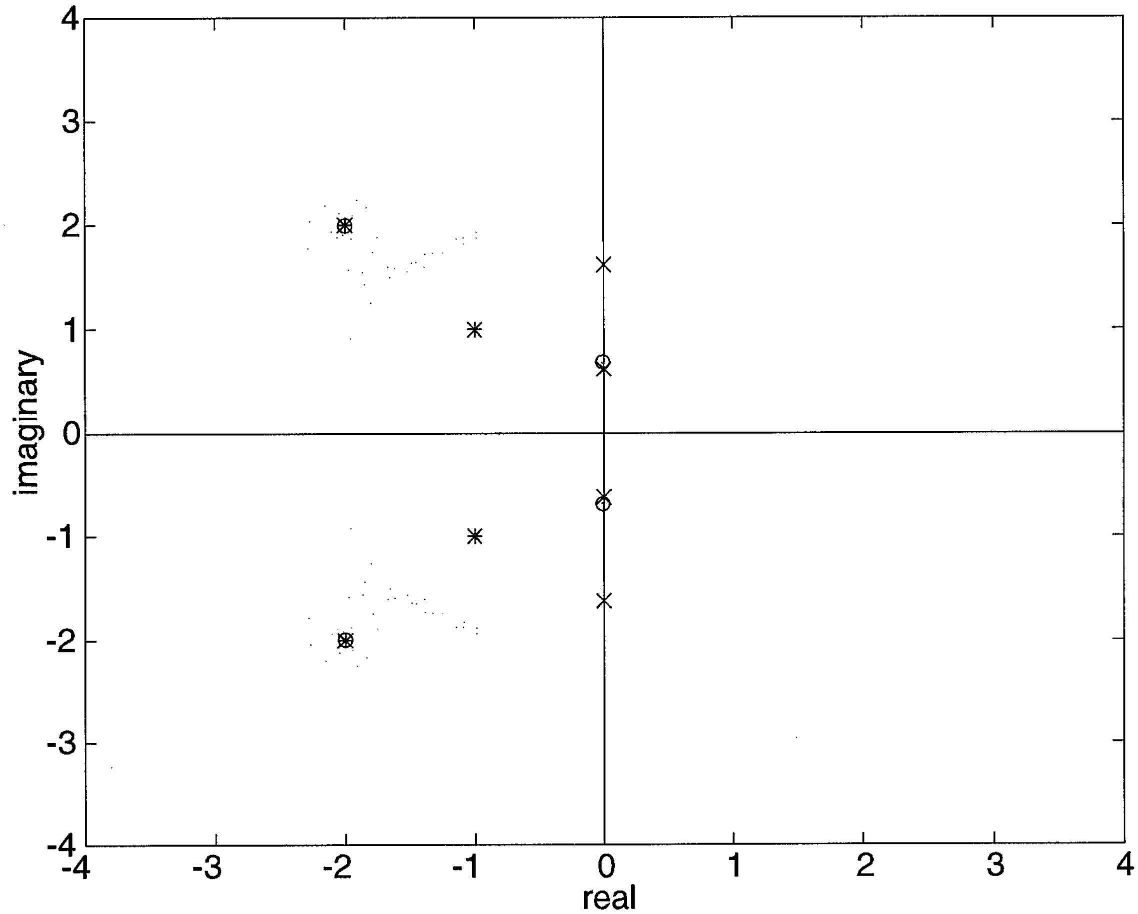

Figure 16. Two degrees-of-freedom with one sensor

The vibrations analyst will recognize that the sensor placement is such that it can only measure one of the modes of the system. In fact, the eigenvectors for the open-loop system show that the frequency at 0.618 has a mode shape where both of the masses move together. The eigenvectors for the second frequency have a mode shape where the masses move in opposite directions. So, in effect, the sensor placement for this example dictated that we could only change the second frequency. Now, we can change our sensor-actuator pair to measure the first mode shape, which looks like Figure 17. This figure is mainly for visualization, since the control matrix is set up so that the sensor measures $x_1 + x_2$, which is difficult to graphically display.

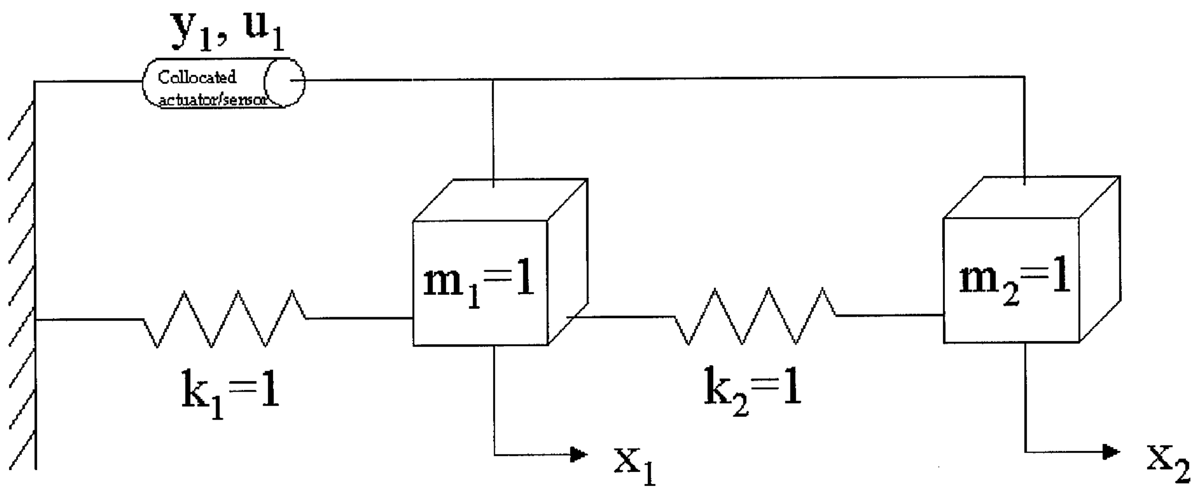

Figure 17. Two DOF sytem with one sensor in new location

This new sensor placement results in Figure 18 and Table 14.

The first mode has been moved by changing the control, or D, matrix and changing the desired eigenvalues to reflect that only one pole would be controlled. Finally, an interesting study is when this two DOF system is not attached to a ground. We are going to again place the sensor-actuator pair between the masses, but our open-loop system will have two poles at the origin corresponding to rigid-body motion. This system graphically looks like Figure 19.

The output is shown in Figure 20 and Table 15.

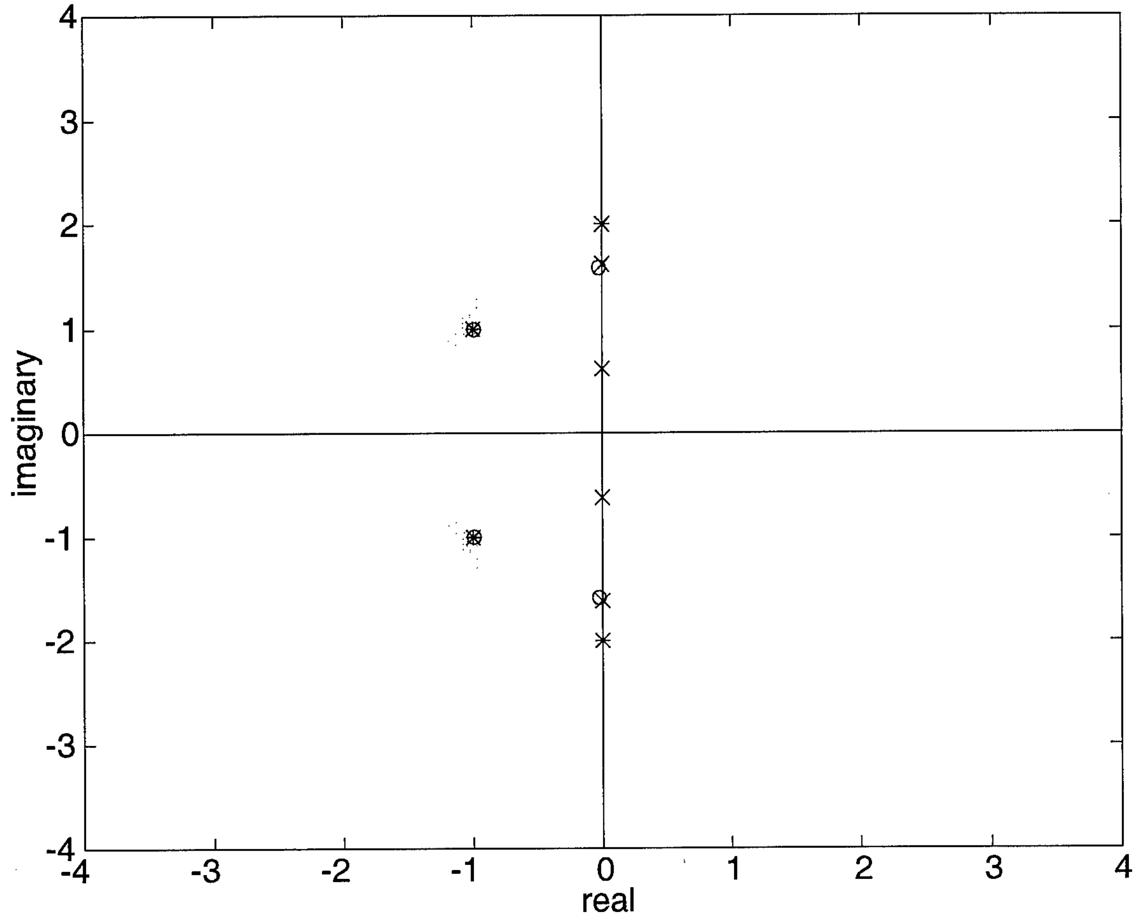

Figure 18. Two degrees-of-freedom with new sensor placement

G_P	G_R	desired	achieved	nat. freq.	damping
0.7901	1.0157	$-1 \pm i$	-0.9911 $\pm$ 0.9937i	1.4035	0.7062
		$\pm 2i$	-0.0245 $\pm$ 1.5851i	1.5853	0.0155

Table 14. Two degree-of-fredom with new sensor placement results

G_P	G_R	desired	achieved	nat. freq.	damping
$3.2791*10^{-8}$	1	$0\pm0i$	0	0	0
		$-1\pm1i$	$-1\pm1i$	1.4142	0.7071

Table 15. Two degrees-of-freedom with rigid-body modes results

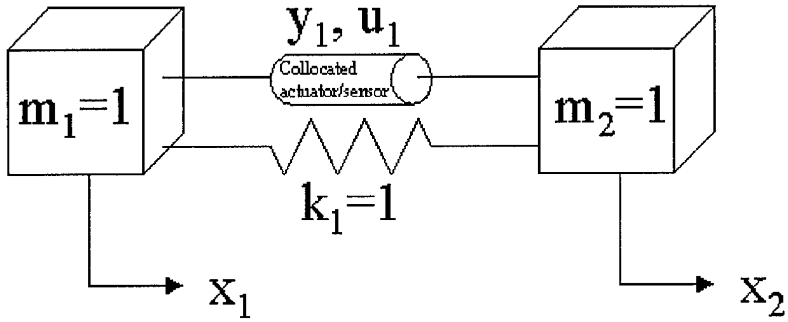

Figure 19. Two DOF system with rigid-body modes

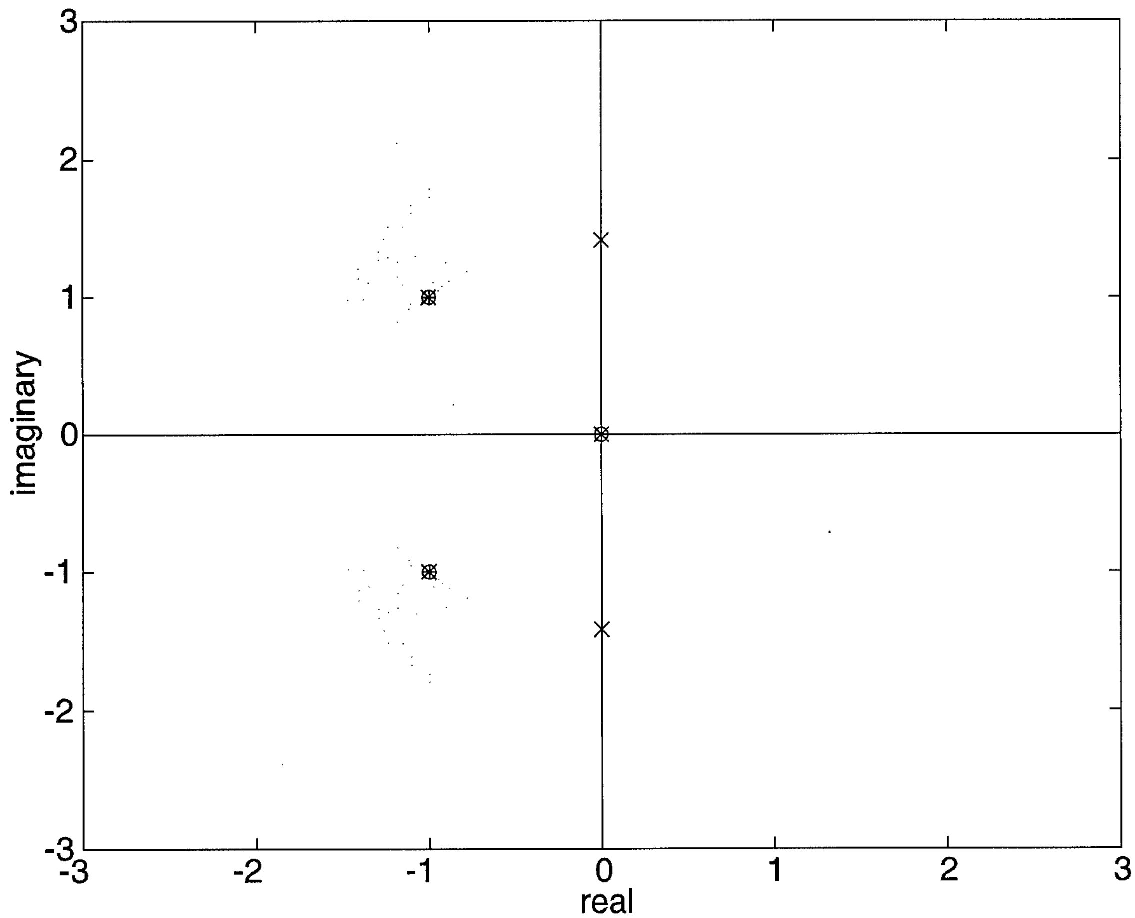

Figure 20. Two degrees-of-freedom with rigid-body modes

Notice that this system acted like the system in Figure 16 in that the second mode eigenvalues where moved to the desired location. In this case, the first mode eigenvalues are at the origin. In most space applications, vibrations are controlled by one system while the translation of the rigid body is controlled by a separate and distinct system.

5.3 Four DOF system

We will use the four mass system shown in Figure 21 to further explore the case in which we have fewer sensors than degrees-of-freedom.

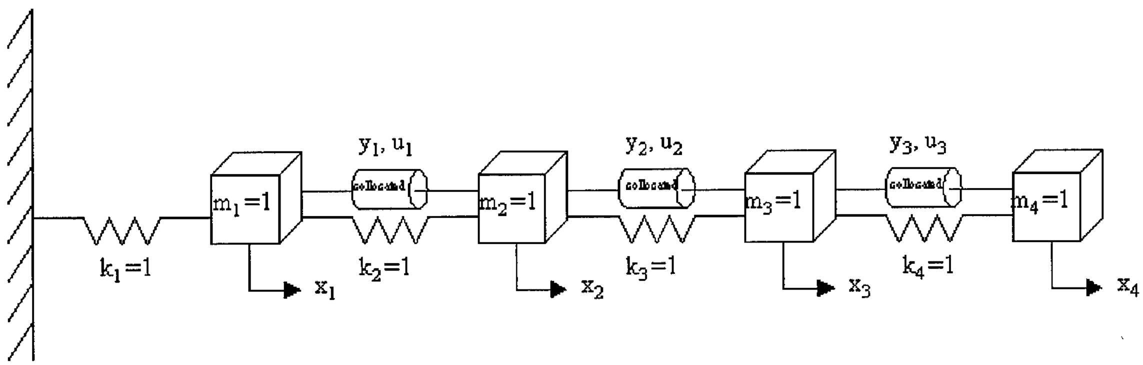

Figure 21. Four DOF system with three sensors

The results of the computer run can be seen in Figure 22 and Table 16. Notice that we have the same problem as we did with the two DOF system in that we do not move the smallest eigenvalue. While the theory section showed that we can only move as many eigenvalues as we have sensor/actuators, this case shows that there is an additional controllability/observability issue to contend with. We will explore this further in the next section with the 29 degree-of-freedom truss example.

For now, notice that we can still control a Four DOF system if it has a rigid body mode and we have three sensors/actuator pairs. This is shown graphically in Figure 23 and the results are shown in Figure 24 and Table 17.

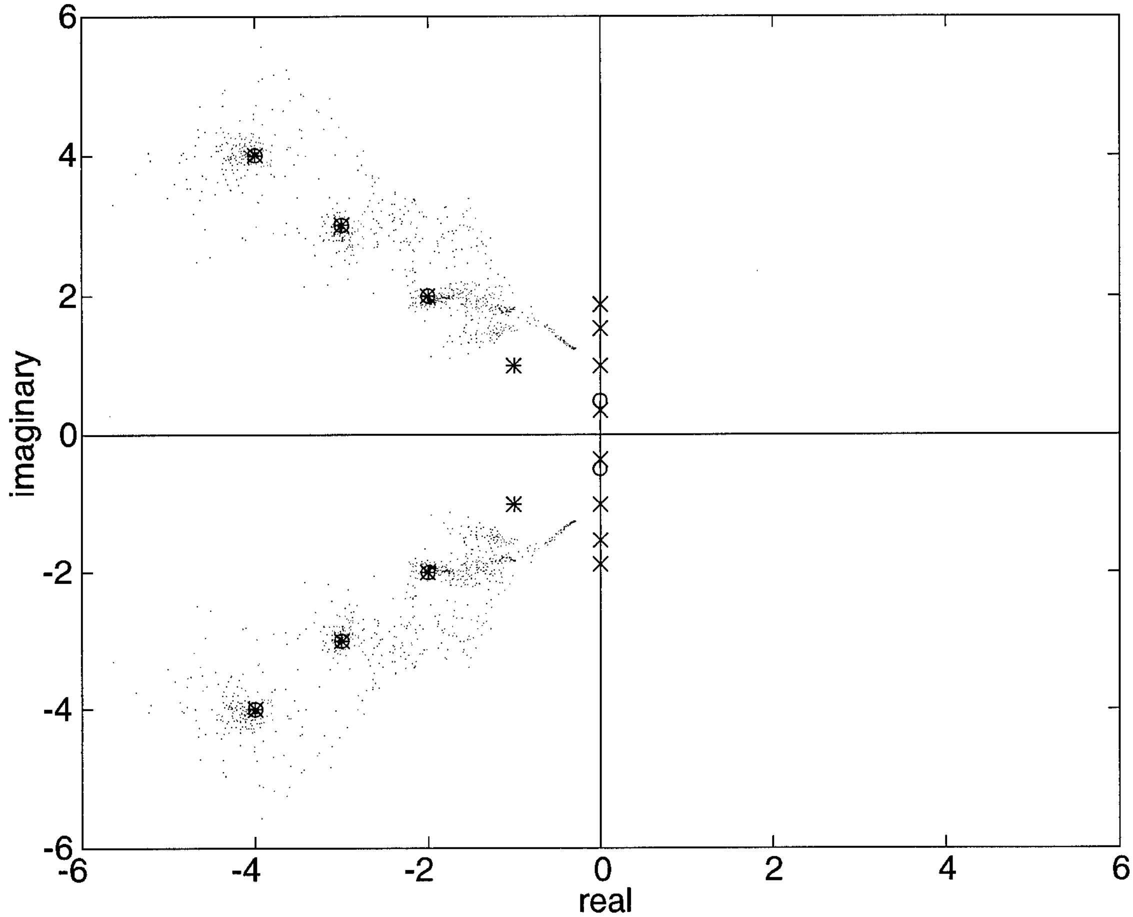

Figure 22. Four degrees-of-freedom with three sensors

G_P	G_R		
$\begin{bmatrix} 8.1219 & -0.0022 & -0.0113 \\ -0.0022 & 5.4504 & -0.0163 \\ -0.0113 & -0.0163 & 0.2237 \end{bmatrix}$	$\begin{bmatrix} 2.0370 & -0.0003 & 0.0001 \\ -0.0003 & 2.4575 & -0.0012 \\ 0.0001 & -0.0012 & 1.5039 \end{bmatrix}$		
$det = 9.8988$	$det = 24.7742$		
desired	achieved	nat. freq.	damping
$0 \pm 0i$	$0 \pm 0i$	0	0
$-1 \pm 1i$	$-1 \pm 1i$	1.4142	0.7071
$-2 \pm 2i$	$-2 \pm 2i$	2.8284	0.7071
$-3 \pm 3i$	$-3 \pm 3i$	4.2426	0.7071

Table 16. Four degrees-of-freedom with three sensors results

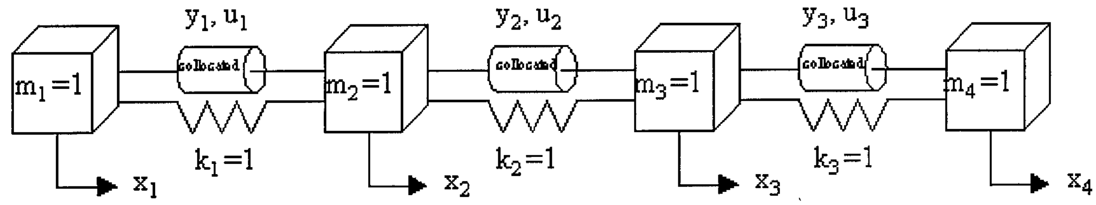

Figure 23. Four DOF system with rigid body modes

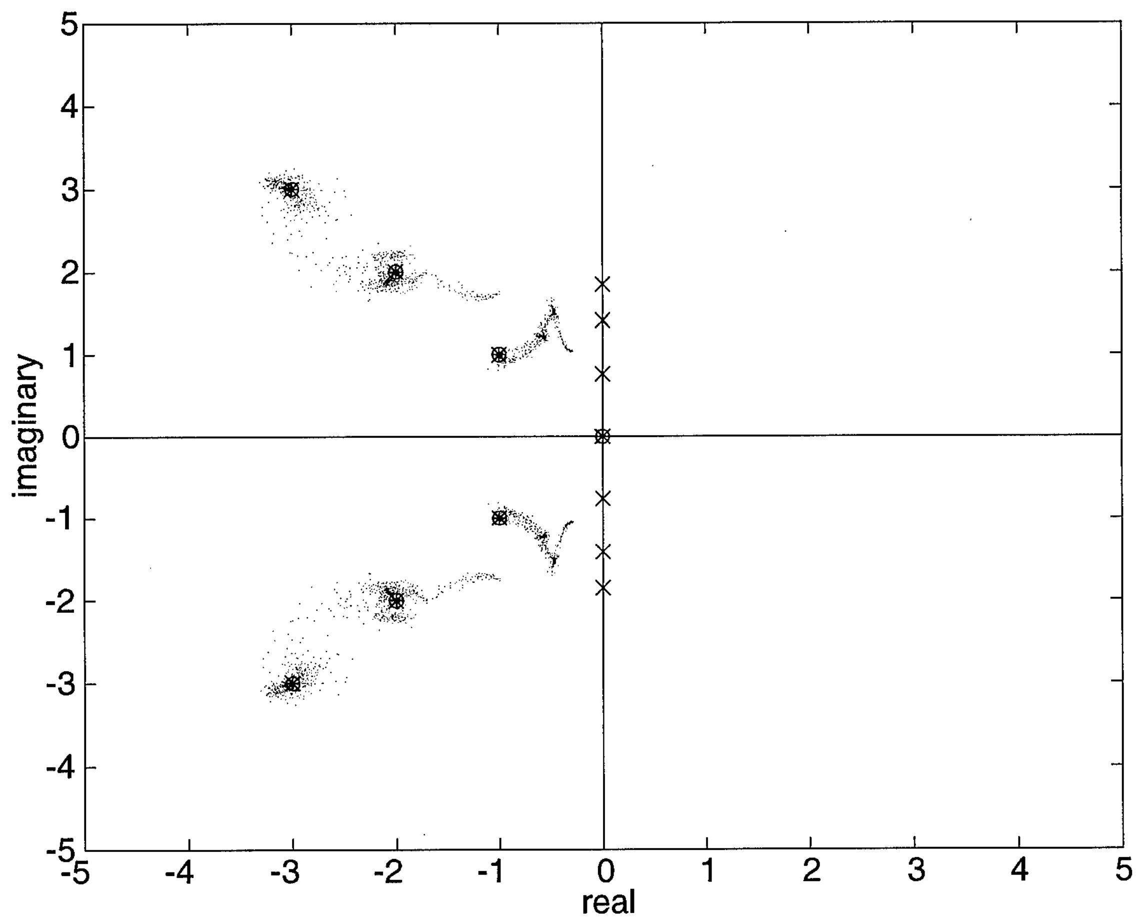

Figure 24. Four degrees-of-freedom with rigid body modes

G_P	G_R		
$\begin{bmatrix} 16.3530 & -0.0744 & 0.0067 \\ -0.0744 & 16.1186 & -0.0239 \\ 0.0067 & -0.0239 & 2.6464 \end{bmatrix}$	$\begin{bmatrix} 2.3347 & -0.0047 & -0.0042 \\ -0.0047 & 3.9421 & -0.0312 \\ -0.0042 & -0.0312 & 2.6920 \end{bmatrix}$		
$det = 697.5459$	$det = 24.7742$		
desired	achieved	nat. freq.	damping
$1 \pm 1i$	$-0.0023 \pm 0.4846i$	0.4846	0.0048
$-2 \pm 2i$	$-2.0009 \pm 1.9995i$	2.8288	0.7074
$-2 \pm 2i$	$-3.0009 \pm 3.0002i$	4.2434	0.7072
$-2 \pm 2i$	$-4.0006 \pm 4.0004i$	5.6576	0.7071

Table 17. Four degrees-of-freedom with rigid-body modes results

5.4 Truss system

The final example we will explore is the 29 degree-of-freedom truss in Figure 25. There are 16 nodes on this system, and each can move in the horizontal and vertical directions. A simple finite element computer algorithm was used to develop the mass and stiffness matrices for this system.

The mass and stiffness were input into ear.m, and many sensor/actuator combinations were tried. The objective was to control the first five modes of the system, which we have seen with the two and four DOF systems to be difficult without the correct sensor/actuator placement. Also, each of the runs of this system took approximately an hour of computation time, so guessing the sensor placement to control the first five modes became prohibitive. An example run is shown in Figure 26.

The general problem of output feedback control does not follow the traditional analysis for controllability and observability. Thus, further analysis and further research should be conducted to analyze the controllability and observability of this specific control problem. Until that time, this method is limited to either increasing the number of sensors or reducing the model order. However, model reduction may affect the mode shapes of the full order system.

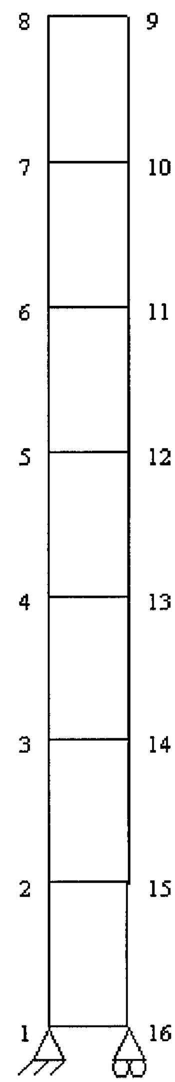

Figure 25. 29 degree-of-freedom planar truss

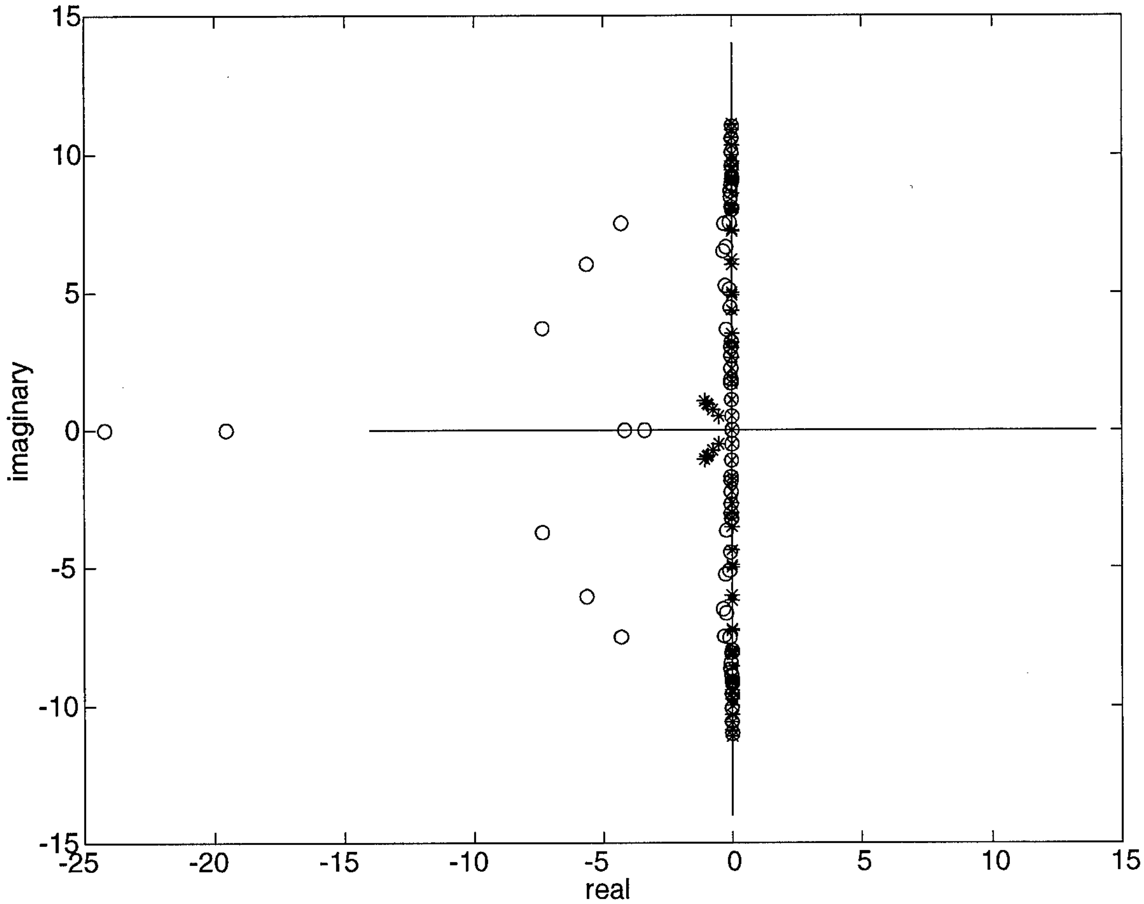

Figure 26. Truss problem which shows the difficulty of sensor placement

Chapter 6 - Conclusion and Recommendations

6.1 Summary

As a summary, we can look once again at the figure given in the introduction shown here as Figure 27.

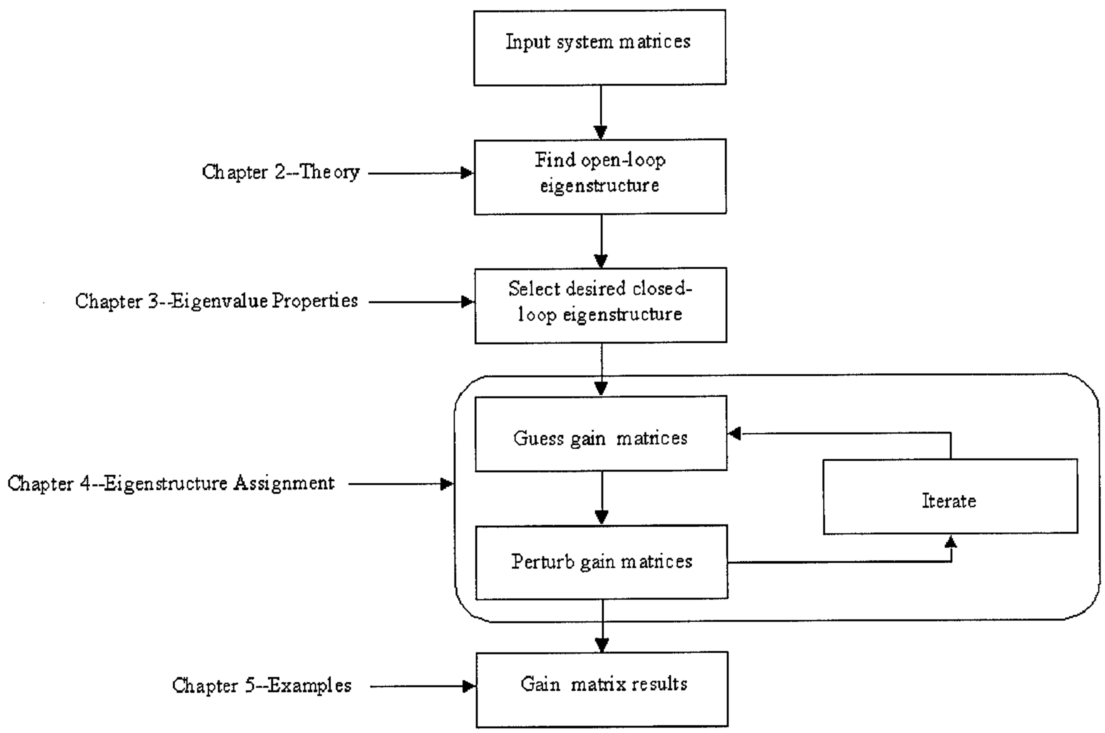

Figure 27. Block diagram of eigenvalue assignment process

This thesis began with the mathematical theory necessary to carry out the eigenstructure assignment algorithm. Chapter Two discussed Lyapunov stability and positive definite output feedback, which provided a framework for investigating the open-loop eigenvalues of a system, ensuring stability of a closed-loop system, and exploring the specific methodology used in this thesis.

Chapter Three gave a proof of the bounds on eigenvalue placement that was completely original to this work. That proof showed that the eigenvalues of a system using positive definite output feedback control will always increase because of the positive definite gain matrices. The proof not only explored an undamped system, but showed that a system with small damping will follow the same rule as the undamped system.

Next, the eigenvalue assignment system was discussed in Chapter Four, and a discussion of the computer code used in this work was given. Finally, some examples were provided which both validated the algorithm and showed its usefulness.

With all of the research accomplish in this thesis, there are still plenty of topics for future research and study. The next section discusses a few of the myriad of possible topics.

6.2 Recommendations For Further Study

This thesis effort has given rise to a number of future research topics. Perhaps the most important is to look at sensor/actuator placement. This could be done through a computer algorithm or possibly the problem could be solved in closed form. The researcher wishing to pursue this type of work could start with the controllability/observability criteria for full state feedback and adapt it to output feedback. Perhaps, if there is no closed form solution, another cost function could be created which finds an optimal sensor/actuator arrangement for a given system.

The next area of study would be non-collocated sensors. There are two definitions for this problem in the literature. The first is to assume the designer wanted collocated sensors, but could not get them exactly aligned. Research could then be conducted to measure the error when there was a given offset in the hardware placement. The second definition of non-collocation dealt with purposefully locating the sensor away from the actuator on, for example, a flexible beam. An example of this research would determine if a better sensor placement could be found to measure the eigenvalues of a system, and the actuator could still control specific modes.

Further, non-symmetric gain matrices could be explored using the same algorithm of this thesis to determine if there is any improvement in the eigenstructure placement. The new algorithm may speed up processing time and provide some insight into the problem of optimal sensor placement.

Finally, a study could be conducted to combine this thesis research with an optimum pole placement method. Through out this thesis the desired eigenvalues and eigenvectors were assumed to be given by a designer. However, there is extensive literature on optimization that could help a researcher examine a method of pole placement that optimizes other criteria than examined here, such as other stability robustness measures.

Appendices

APPENDIX A - Computer Code

This appendix includes the computer code used for this thesis including EAR.M and its sub-routines.

A.1 EAR.M

```
%------------------------------------------
%
% EIGENSTRUCTURE ASSIGNMENT ALGORITHM
%
% Author: Adam G. Harris Date: 31 March 1999
%
% This routine implements the eigenstructure assignment algorithm
% used in this thesis. The input file called "getdata.m" must be
% created before running this routine.
%
% fuctions calls: getdata.m
% eigsort.m
% plotinit.m
%
%------------------------------------------
clear all
ea=[];
veca=[];
%------------------------------------------
% Pre process the data
%------------------------------------------
getdata % call getdata.m with M,C,K,D,ed,F
Minv=inv(M);
[rowD,columD]=size(D);
if isempty(F)==1, F=ones(2*rowD,1);, end % if F undefined make it ones
if isempty(Fvec)==1, Fvec=zeros(2*rowD,1);, end %if Fvec undefined make it zero
eigsort;
%------------------------------------------
% Process the data
%------------------------------------------
%initialize the matrix x=[p11 p21 p22...pnn r11 r21 r22...rnn]'
%init. to Gp=Gr=I
s=1;
for i=1:columD
  for j=1:i
    if i==j, x0(s)=1;
    else x0(s)=0;
    end
    s=s+1;
  end
end
```

```
plotinit(2*rowD,ed)
% Plot the open loop eigenvalues
oltemp=[[zeros(rowD,rowD);eye(rowD)]';[-Minv*K';-Minv*C']'];
ol=eig(oltemp)
for index=1:size(ol,1)
  plot(real(ol(index)),imag(ol(index)),'xb')
end
% Initialize fmins routine
x0=[x0 x0]; % 2 x0 for Gp and Gr equal Identity
%x0=zeros(1,columD^2+columD);
x=fmins(routine,x0,[],[],C,K,D,ed,F,vecd,Fvec,Minv); %minimize the cost function
%----------------------------------------------------------------
% Post process the data
%----------------------------------------------------------------
% Create and print pos definite gain maticies
[dummy,colx]=size(x); %note m is both p and r
sizeG=(-1+sqrt(1+4*colx))/2;
s=1;
Lp=zeros(sizeG,sizeG);
Lr=zeros(sizeG,sizeG);
for ind=1:sizeG
  for j=1:ind
    Lp(ind,j)=x(s);
    Lr(ind,j)=x(s+colx/2);
    s=s+1;
  end
end
Gp=Lp*Lp', detGp=det(Gp)
Gr=Lr*Lr', detGr=det(Gr)
% Create pos definite gain maticies and CL A matrix
Ktil=-Minv*[K+D*Gp*D'];
Ctil=-Minv*[C+D*Gr*D'];
A=[[zeros(rowD,rowD);eye(rowD)]';[Ktil';Ctil']'];
% Calculate achievable eigenvalues and vectors
[vecat,eatemp]=eig(A);
for ind=1:size(eatemp,1)
  ea(ind,1)=eatemp(ind,ind);
end
% sort eigenvalues and corresponding eigenvectors
[ea,index]=sort(ea);
for ind=1:size(eatemp,1)
  vecatmp(:,ind)=vecat(:,index(ind));
end
veca=vecatmp;
% Plot final eigenvalues
for count=1:2*rowD
  plot(real(ea(count)),imag(ea(count)),'og')
end
title('Gradient Search For Achievable Poles')
```

```
xlabel('real')
ylabel('imaginary')
hold off
% print final results
ed
ea
for iii=1:2:size(ea,1)
  nat=abs(ea(iii))
  gamma=-real(ea(iii))/abs(ea(iii))
end
vecd
veca
```

A.2 GETDATA.M

```
%------------------------------------------------------------
%
% GET DATA INITIALIZATION
%
% Author: Adam G. Harris Date: 31 March 1999
%
% This input file must be created before running the
% eigenstructure assignment algorithm.
%
% routine = 'value' uses eigenvalue cost function
% 'structure' uses eigenstructure cost function
% M = mass matrix
% C = damping matrix
% K = stiffness matrix
% D = control matrix
% ed = desired eigenvalues
% F = eignvalue weighting
% vecd = desired eigenvectors
% Fvec = eigenvector weighting
%
%------------------------------------------------------------
routine='value';
M=[1 0;0 1];
C=[0 0;0 0];
K=[2 -1;-1 1];
D=[1;1];
ed=[-1+1i -1-1i 0+2i 0-2i]';
F=[1 1 1 1];
vecd=[
0.3947 + 0.0976i 0.3947 - 0.0976i 0.2631 + 0.1151i 0.2631 - 0.1151i
0.4011 - 0.0849i 0.4011 + 0.0849i 0.0551 - 0.1599i 0.0551 + 0.1599i
-0.2970 - 0.4923i -0.2970 + 0.4923i -0.2960 - 0.7566i -0.2960 + 0.7566i
-0.4859 - 0.3162i -0.4859 + 0.3162i -0.4301 + 0.2096i -0.4301 - 0.2096i];
Fvec=[0 0 0 0];
% routine may be given 'value' or 'structure' depending on whether the user
% desires a cost function of just the eigenvalues or a cost function with
% both eigenvalues and eigenvectors. Note: using 'value' is equivalent to using
% 'structure' and setting Fvec=[zeros], however 'value' will run much quicker
```

A.3 EIGSORT.M

```
%--------------------------------------------------------------
%
% EIGENSTRUCTURE SORTING
%
% Author: Adam G. Harris Date: 31 March 1999
%
% This routine sorts the eigenvalues, eigenvectors, and weighting
% matricies for the eigenstructure assignment algorithm.
%
% ed = desired eigenvalues
% F = eigenvalue weighting
% vecd = desired eigenvectors
% Fvec = eigenvector weighting
%
%--------------------------------------------------------------
%sort eigenvalues
[count,dumb]=size(ed);
[ed,index]=sort(ed);
Ftmp=zeros(count,1);
Fvectmp=zeros(count,1);
for k=1:count
  Ftmp(k)=F(index(k));
  vecdtmp(:,k)=vecd(:,index(k))/norm(vecd(:,index(k))); %norm to 1
  Fvectmp(k)=Fvec(index(k));
end
F=diag(Ftmp); %diagonalize into a square matrix
vecd=vecdtmp;
Fvec=diag(Fvectmp); %diagonalize into a square matrix
```

A.4 PLOTINIT.M

```
%-----------------------------------------------------------
%
% PLOT INITIALIZATION
%
% Author: Adam G. Harris Date: 31 March 1999
%
% This function initializes the plot for the ear.m routine.
%
% plotinit(n,ed)
%
% n = number of eigenvalues
% ed = desired eigenvalues
%
%-----------------------------------------------------------
function plotinit(n,ed)
% used to set up the plotting feature
axis('square')
e=sort(ed);
if abs(real(e(n)))>abs(imag(e(n))), axisize=ceil(abs(real(e(n))))-0.5;
else axisize=ceil(abs(imag(e(n))))-0.5;
end
axis([-axisize axisize -axisize axisize])
plot([0 0],[-axisize axisize],'-b',[-axisize axisize],[0 0],'-b')
hold on
for i=1:size(e,1)
  plot(real(e(i)),imag(e(i)),'*c')
end
```

A.5 VALUE.M

```
%------------------------------------------------------------------
%
% VALUE
%
% Author: Adam G. Harris Date: 31 March 1999
%
% This function is used by fmins to minimize the cost function for
% the eigenvalues only.
%
% [Jtemp]=value(x,C,K,D,ed,F,vecd,Fvec,Minv)
%
% x = input gain matricies reduced to vector form
% C = damping matrix
% K = stiffness matrix
% D = control matrix
% ed = desired eigenvalues
% F = eignvalue weighting
% vecd = desired eigenvectors (not used in this routine)
% Fvec = eigenvector weighting (not used in this routine)
% Minv = inverse of the mass matrix
%
%------------------------------------------------------------------
function Jtemp=value(x,C,K,D,ed,F,vecd,Fvec,Minv)
[rowK,dummy]=size(K);
% x=[p11 p21...pnn r11 r21...rnn]'
% form initial gain matrix lower triangular portion of Cholesky factorization
[dummy,colX]=size(x); %note colX is both p and r
sizeG=(-1+sqrt(1+4*colX))/2;
s=1;
Lp=zeros(sizeG,sizeG);
Lr=zeros(sizeG,sizeG);
for ind=1:sizeG
  for j=1:ind
    Lp(ind,j)=x(s);
    Lr(ind,j)=x(s+colX/2);
    s=s+1;
  end
end
% Create pos definite gain maticies and CL A matrix
Gp=Lp*Lp';
Gr=Lr*Lr';
Ktil=-Minv*[K+D*Gp*D'];
Ctil=-Minv*[C+D*Gr*D'];
A=[[zeros(rowK,rowK);eye(rowK)]';[Ktil';Ctil']'];
% Calculate achievable eigenvalues
eatemp=eig(A);
% sort eigenvalues
ea=sort(eatemp);
```

```
% Calculate Cost Function for all eigenvalues
Jtemp=(ed-ea)'*F*(ed-ea)
for index=1:size(ea,1)
  plot(real(ea(index)),imag(ea(index)),'.r')
end
```

A.6 STRUCTURE.M

```
%--------------------------------------------------------------------------
%
% STRUCTURE
%
% Author: Adam G. Harris Date: 31 March 1999
%
% This function is used by fmins to minimize the cost function for both
% the eigenvalues and eignevectors.
%
% [J]=structure(x,C,K,D,ed,F,vecd,Fvec,Minv)
%
% x = input gain matricies reduced to vector form
% C = damping matrix
% K = stiffness matrix
% D = control matrix
% ed = desired eigenvalues
% F = eignvalue weighting
% vecd = desired eigenvectors
% Fvec = eigenvector weighting
% Minv = inverse of the mass matrix
%
%--------------------------------------------------------------------------
function J=structure(x,C,K,D,ed,F,vecd,Fvec,Minv)
[rowK,dummy]=size(K);
% x=[p11 p21...pnn r11 r21...rnn]'
% form initial gain matrix lower triangular portion of Cholesky factorization
[dummy,colX]=size(x);%note colX is both p and r
rowG=(-1+sqrt(1+4*colX))/2;
s=1;
Lp=zeros(rowG,rowG);
Lr=zeros(rowG,rowG);
for ind=1:rowG
  for j=1:ind
    Lp(ind,j)=x(s);
    Lr(ind,j)=x(s+colX/2);
    s=s+1;
  end
end
% Create pos definite gain maticies and CL A matrix
Gp=Lp*Lp';
Gr=Lr*Lr';
Ktil=-Minv*[K+D*Gp*D'];
Ctil=-Minv*[C+D*Gr*D'];
A=[[zeros(rowK,rowK);eye(rowK)]';[Ktil';Ctil']'];
% Calculate achievable eigenvalues and vectors
[vecat,eatemp]=eig(A);
numeig=size(eatemp,1);
% Change from a diagonal matrix to a vector
```

```
for ind=1:numeig
  ea(ind,1)=eatemp(ind,ind);
end
% sort eigenvalues and corresponding eigenvectors
[ea,index]=sort(ea);
% Calculate Cost Function for all eigenvalues
Jtemp=(ed-ea)'*F*(ed-ea);
for ind=1:numeig
  vecatmp(:,ind)=vecat(:,index(ind));
end
veca=vecatmp; % remember that MatLab normalizes e-vecs to one
Jvec=0.0;
for i=1:numeig
  Jvec=Jvec+(vecd(:,i)-veca(:,i))'*Fvec(i,i)*(vecd(:,i)-veca(:,i));
end
%----------------------------------------
% calculte J and plot
%----------------------------------------
J=Jtemp+Jvec;
for i=1:size(ea,1)
  plot(real(ea(i)),imag(ea(i)),'.r')
end
```

Bibliography

[1] Huckabone, Thomas C. *An Algorithm for Robust Eigenstructure Assignment Using the Linear Quadratic Regulator*. MS thesis, Air Force Institute of Technology, Wright Patterson AFB, OH, December 1991.

[2] Junkins, John L., editor. *Mechanics and Control of Large Flexible Structures, vol. 129*. Progress in Astronautics and Aeronautics. Washington, DC: American Institute of Aeronautics and Astronautics, Inc., 1990.

[3] Junkins, John L. and Youdan Kim. *Introduction to Dynamics and Control of Flexible Structures*. American Institute of Aeronautics and Astronautics, Inc., 1993.

[4] Kim, Y. and J.L. Junkins. "A Measure of Controllability for Actuator Placement," *Journal of Guidance, Control, and Dynamics, vol. 14*(No. 5):895–902 (Sept.-Oct. 1991).

[5] Lee, W. C. *Effects of Colocation and Non-Colocation of Sensors and Actuators on Flexible Structures*. MS thesis, Air Force Institute of Technology, Wright Patterson AFB, OH, March 1991.

[6] Meirovitch, Leonard. *Analytical Methods in Vibrations*. New York: Macmillan Publishing Co., Inc., 1967.

[7] Natsiavas, S. and J. L. Beck. "Almost Classically Damped Linear Discrete Systems," *Proceedings of the 12th International Modal Analysis Conference, vol. 2*:1656–1661 (1994).

[8] Robinson, J. D. *A Linear Quadratic Regulator Weight Selection Algorithm for Robust Pole Assignment*. MS thesis, Air Force Institute of Technology, Wright Patterson AFB, OH, Month 1990.

[9] Soloman, Dempsey D. *Helicopter Flight Control System Design Using the Linear Quadratic Regulator for Robust Eigenstructure Assignment*. MS thesis, Air Force Institute of Technology, Wright Patterson AFB, OH, December 1992.

[10] Strang, Gilbert. *Linear Algebra and its Applications*. Orlando, Florida: Harcourt Brace and Company, 1988.

[11] Wilkinson, J. H. *The Algebraic Eigenvalue Problem*. Oxford: Clarendon Press, 1965.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1999	Master's Thesis

4. TITLE AND SUBTITLE
Robust Eigenstructure Assignment Using Positive Definite Output Feedback Control

5. FUNDING NUMBERS

6. AUTHOR(S)
Adam G. Harris, 1Lt, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GAE/ENY/99M-6

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Capt Richard G. Cobb
Tech Sat Program Manager
AFRL/VSOC
Kirtland AFB, NM
(505) 853-3330

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Dr Bradley S. Liebst, 255-3636 x4636, bradley.liebst@afit.af.mil

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
This research investigates a method of eigenstructure assignment for the classes of controllers known as positive definite output feedback control. The research began with a closed loop system with output feedback control that utilizes a collocated actuator-sensor pair. It is shown that this system is robust because it is globally stable for positive definite feedback. Analytical analysis is then completed to determine gains that minimize a cost function that is the error between the desired and achievable (assuming positive definite output feedback) eigenstructure. Examples are given at the end of the thesis to validate the computer code and theory results.

14. SUBJECT TERMS
Eigenstructure Assignment; Eigenvalue; Eigenvector; Positive Definite; Output Feedback; Control Theory; Optimization; Linear Systems

15. NUMBER OF PAGES
94

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL