24758152
11/11

A REDUCED ORDER CONTROLLER DESIGN METHOD BASED ON THE
YOULA PARAMETERIZATION OF ALL STABILIZING CONTROLLERS

A Dissertation Presented to
The Faculty of the Russ College of Engineering and Technology
Ohio University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

by
Russell David Glenn

November, 1995

11/11/95
OHIO UNIVERSITY
LIBRARY

Acknowledgements

I wish to thank Dr. Dennis Irwin, my research advisor, for his guidance and encouragement throughout the process of writing this dissertation. Special thanks also go to Dr. Doug Lawrence for his assistance in the development of the Linear Geometric Control theoretic interpretation of the reduced order controller design methods presented in Chapters 1 through 5. My most heartfelt gratitude goes out to my wife Ellen, without whose prayers, encouragement, and support I could not have persevered.

Table of Contents

Acknowledgements	iii
Table of Contents	iv
1 Introduction	1
2 Mathematical Background	11
2.1 System Representations	11
2.2 Controllability and Observability of a Realization	13
2.3 The Orthogonal Controllable Canonical Form	15
2.4 The Orthogonal Observable Canonical Form	16
2.5 The Youla Parameterization of All Stabilizing Controllers of G(s)	17
3 Reduced Order Controller Design	20
3.1 Strictly Proper Reduced Order Controllers of Order ν_o	23
3.2 Strictly Proper Reduced Order Controllers of Order ν_c	25
3.3 Reduced Order Controllers of Order Equal to $\min(\nu_o-1, \nu_c-1)$	28
3.4 Larger Reduced Order Controllers	30
4 A Proof of the Existence of Reduced Order Controllers	34
4.1 A Proof of the Existence of Strictly Proper Controllers of Order ν_o	35
4.2 A Computational Algorithm	43
4.3 Controllers of Order ν_o-1	46
4.4 An Analysis of Computational Complexity	50
4.5 Pole Placement Design Examples	53

5	Reduced Order H^∞ Controllers	66
5.1	Reducing the Order of H^∞ Controllers	67
5.2	The Normalized H^∞ Control Problem	72
5.3	A Reduced Order Normalized H^∞ Controller Design Example . .	77
5.4	The Mixed Sensitivity H^∞ Problem	83
5.5	A Reduced Order Mixed Sensitivity H^∞ Controller Design Example	92
6	A Geometric Interpretation	101
6.1	An Introduction to Linear Geometric Control Theory	102
6.2	Compensator Synthesis Using Controlled and Conditioned Invariant Subspaces	105
6.3	Connections Between the Geometric and Youla Parameterization Methods	108
6.4	Example Problems	116
6.5	Conclusion	123
7	Conclusions and Suggestions for Further Research	124
	References	128
	Appendix A Plant Models for Design Examples	131
	A.1 Pole Placement Design Examples	131
	A.2 The Normalized H^∞ Control Problem	132
	A.3 The Mixed Sensitivity H^∞ Design Problem	133
	Appendix B Matlab® M-file Listings	134
	B.1 Orthogonal Canonical Forms	134
	B.2 Sylvester Equation Solution Algorithms	140
	B.3 Pole Placement Example M-files	155
	B.4 The Normalized H^∞ Control Problem	165
	B.5 The Mixed Sensitivity H^∞ Control Problem	176

1 Introduction

The design of feedback controllers for complicated plants can be a very involved process. The controller must not only stabilize the plant, but must achieve desired closed loop system performance specifications and be realizable as a piece of electronic hardware, a software algorithm, or a combination of the two. Many controller design methods have been developed, but each has strengths and weaknesses; no particular method has evolved as the one preferred method for general controller design problems. When performing a controller design, the available design methods must be evaluated in order to determine which might yield an acceptable controller in the face of the constraints imposed by the problem at hand, and with a minimum amount of time and effort on the part of the designer. A brief summary of some popular controller design methods is presented.

Many of the early controller design methods are centered on frequency domain analysis of the design plant, and are most easily applied to single-input, single-output (SISO) systems. The traditional root locus and Nyquist Plot methods taught in many undergraduate controls courses (Phillips, 1991, Chapters 6-9) are examples of such

frequency domain analysis methods. Multi-input, multi-output (MIMO) examples include sequential loop closing, characteristic locus, and Nyquist Array methods (Maciejowski, 1989, Chapter 4). In these design approaches, the plant frequency response is examined and a compensator frequency response is determined which yields desired closed loop performance. The designer fabricates a compensator which approximates the desired frequency response, then performs a closed loop system analysis with the compensator in place to evaluate whether the design goals have been achieved. If not, the process is repeated. Advantages of the frequency response techniques are that, when successfully implemented, they typically yield low order stabilizing controllers and the designer has the freedom to specify the controller structure. Disadvantages include the fact that such designs can be more art than science, and they are often difficult to apply to complex MIMO systems.

In modern control approaches, compensators are designed based on a state-space realization of the design plant. In a state-space realization, continuous time control systems are modeled as a set of first order matrix linear differential equations

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du}\end{aligned}\quad (1.1)$$

In these equations, the matrix A contains relationships between the system state derivatives and the system states; the B matrix relates the system inputs to the state derivatives; matrix C describes the system outputs as linear combinations of the states, and matrix D characterizes any direct paths from system inputs to system outputs. Discrete time plants are similar except they are based on first order difference

equations rather than differential equations. Having access to the system states can provide insight into some internal system behavior. For example, asymptotic stability of a continuous time realization is guaranteed if and only if the eigenvalues of the system A matrix have negative real parts and discrete time realization stability is guaranteed if the eigenvalue magnitudes are less than one.

Several design methods have been developed based on state-space realizations. In eigenvalue placement techniques, linear combinations of the system states (or state estimates) are fed back into the system inputs to place the closed loop A matrix eigenvalues at arbitrary desired locations in the complex plane (Ogata, 1990, Chapter 10). Other state-space methods involve the determination of a state feedback which achieves optimization of certain cost functionals comprised of the system states, inputs, and outputs. Some examples are Linear Quadratic (LQ) and Linear Quadratic with Gaussian state estimator (LQG) methods. These methods have existed for some time, but have not proven to be popular in actual implementations (Maciejowski, 1989, Chapter 5). They are not very tolerant of errors or inaccuracies in the plant model, and often behavior which optimizes the cost function is either impractical or undesirable in an actual implementation. Also, it can be difficult to ascertain how changes in the cost function specifications affect closed loop frequency response behavior.

Some more recent controller design methods have combined state-space and frequency response design techniques. In frequency weighted Linear Quadratic Gaussian with Loop Transfer Recovery (LQG/LTR), the designer applies frequency dependent weighting functions in attempts to design stabilizing controllers which satisfy

the optimization constraint of LQ as well as provide desired closed loop frequency response characteristics. Some drawbacks of LQG/LTR are: (1) only certain types of frequency responses can be specified, (2) the method can break down if the plant contains nonminimum phase zeroes, and (3) the controller tends to cancel lightly damped open loop plant A matrix eigenvalues, rendering it intolerant to some model inaccuracies (Maciejowski, 1989, pp. 231-244).

Another approach which can combine state-space and frequency response design specifications is the H^∞ optimization technique (Francis, 1987). In the mixed sensitivity approach to H^∞ design, frequency dependent weighting functions are applied to certain system signals in order to set desired closed loop system performance and robustness specifications. The H^∞ algorithm then produces a stabilizing controller which meets these specifications, if such a controller exists. If a controller cannot be generated, the weighting functions must be redesigned and the procedure repeated. Although H^∞ is a powerful design method it has several disadvantages in application. When the H^∞ algorithm fails to produce a controller which meets a given set of specifications, it is unclear how to alter the specifications in order to increase the likelihood of obtaining a successful controller. Also, unless carefully designed, H^∞ controllers tend to cancel lightly damped plant eigenvalues, similar to LQG/LTR, resulting in controllers with poor stability robustness with respect to unstructured uncertainty. Although the author is unaware of any actual implementations of H^∞ controllers, research in this area continues to be intense.

The state-space design methods share in common the fact that they typically yield controllers with order greater than or equal to the number of states of the open loop plant. Controllers designed via eigenvalue placement or LQG methods have order equal to that of the open loop plant. H^∞ and LQG/LTR methods yield controllers with order equal to that of the open loop plant plus the order of the applied weighting functions. Such high order controllers are often impractical in actual implementation; an example is an H^∞ design performed for the Hubble Space Telescope (HST) for which the H^∞ controller was 85th order, but the HST flight computer could only implement a controller with a maximum order of 42 (Irwin, et. al., 1993). Clearly, when viewed in the context of implementation cost and effort, reduced order controllers (controllers with order less than that of the open loop design plant) are preferred over higher order implementations.

As control systems grow ever more complex, reduced order controller design remains an area of intense interest. Some methods of reduced order controller design attempt to directly design low order stabilizing controllers which achieve as much closed loop performance as possible within an imposed order constraint. Frequency response design techniques generally fall into this category, since they usually yield controllers with order lower than that of the design plant. Other direct methods are numerical techniques which iteratively change the free parameters of a controller realization to achieve desirable closed loop frequency response characteristics. These numerical methods typically treat the design problem as a constrained multiobjective optimization and use gradient search techniques to find a controller which best meets

the design criteria. Low order controller designs are possible because the designer can specify controller order prior to initiating the design process. Disadvantages include the requirement that the designer provide an initial stabilizing controller and the fact that determining constraint functions which yield satisfactory convergence properties for the algorithm can be difficult. Frazier (1993) and the references therein provide an in-depth study of iterative numerical controller design algorithms.

Other recent work involving iterative numerical techniques attempts to achieve approximate pole placement with compensators of fixed order (Battacharyya and Keel, 1990). The approach optimizes an objective function which is a weighted sum of (1) the condition number of a matrix with columns comprised of the closed loop A matrix eigenvectors and (2) the norm of $F-KC$, where F is the state feedback for desired pole placement and KC is the output feedback generated by a previous iteration of the algorithm. Although the algorithm can successfully yield low order stabilizing controllers for some design problems, there is no guarantee that if a suitable controller of given order exists that it will be found, nor is there any guarantee that the algorithm can generate a reduced order controller for all design problems.

Truncation methods generally involve putting a controller realization into a certain coordinate system in which it is possible to evaluate the contribution of individual states to overall controller performance (Moore, 1981), (Glover, 1984), (Mustafa, 1991). The states with the least significant contributions are discarded, resulting in a reduced order controller. There are established inequality bounds for some of these methods which, when satisfied, guarantee that the reduced order

controller provides closed loop stability (Anderson, 1989). However, the inequality bounds are very tight; experience has shown for some problems that omission of any state variables violates the bounds and usually results in instability of the closed loop system. In the HST controller design problem previously mentioned, deletion of as few as one or two states from the 85th order H^∞ controller results in closed loop instability (Irwin, et. al., 1993). Another disadvantage to truncation methods is that if the full order controller is optimal in some sense, optimality is usually lost if any states are omitted to reduce the controller order.

A variation of the truncation methods is to first reduce the order of the design plant, design a controller based on the reduced plant model, then close the loop with the full order plant to evaluate performance. The rationale is that since the most significant plant states are retained for the controller design, a controller that achieves desired specifications with the reduced order plant model should also perform well with the full order model. In practice, this approach suffers from the same drawbacks as controller truncation methods.

Other work in reduced order control has examined the problem of attempting to design low order controllers which achieve closed loop stability when the plant states are not directly available for feedback. Luenberger (1966) showed that a state observer of order equal to (v_o-1) , where v_o is the observability index of the system, can estimate any linear combination of the system state variables. Later work (Brasch and Pearson, 1970) proves that for any controllable and observable (i.e. minimal) plant, arbitrary eigenvalue placement is possible by using feedback which is such a linear combination

of the state variables, and it is also shown that a feedback compensator with order equal to the minimum of (v_c-1, v_o-1) can achieve arbitrary eigenvalue placement, where v_c is the controllability index of the system. In practice, these approaches are difficult to apply to large order control systems; (Murdoch, 1973) presents a simpler approach in which a linear combination of the state variables can be generated with an order v_o controller designed by solving a set of consistent linear equations. Another well known result (Kimura, 1975) is that a controller of order equal to $n-m-l+1$ can generate arbitrary eigenvalue placement for a minimal plant, where n is the number of system states, m is the number of system inputs, and l is the number of system outputs.

Linear geometric control theory is an area in which controller designs are undertaken by utilizing properties of certain subspaces of the full state-space. Schumacher (1980) presents a compensator synthesis procedure based on these methods, and the procedure can yield reduced order stabilizing controllers for some plants. Successful controller designs are dependent upon the ability of the designer to find subspaces which satisfy the properties dictated by the procedure. Several well known controller design methods can be interpreted within the context of the linear geometric compensator synthesis principle. Wonham (1979) and Basile and Marro (1991) provide detailed presentations of linear geometric control theory.

A recent paper (Gu, 1993) shows that certain parameters in the state-space realization of the Youla parameterization of all stabilizing controllers can be manipulated so that some of the controller states are rendered unobservable to the controller outputs. Thus, the controller order can be reduced while maintaining closed

loop stability. While the method is used to design low order stabilizing controllers for several example problems, research to date has yielded no method of guaranteeing the existence of stabilizing controllers with order less than $n-m-l+1$, where $G(s) = (A, B, C, D)$ is an n state, m input, l output realization.

The work presented in this dissertation shows that a new approach to the problem can always yield stabilizing controllers of order equal to the minimum of v_o-1 and v_c-1 , where v_o and v_c are the controllability and observability indices of the system realization, respectively. Furthermore, these controllers yield arbitrary closed loop pole placement. A computational algorithm is presented which calculates the controllers and the algorithm is superior to that of (Brasch and Pearson, 1970) in terms of numerical stability and computational complexity. Since the minimum of v_o and v_c is less than $n-m-l+1$ for many systems, this work presents a lower upper bound for the controller order necessary to stabilize a given realization using the Youla parameterization method.

Additional work illustrates that the method can be used to satisfy some of the necessary conditions for the existence of reduced order suboptimal H^∞ controllers for certain plants. Although additional conditions must be satisfied before the existence of a reduced order H^∞ controller is guaranteed, the method has been successfully implemented on several example problems and can be useful in ascertaining what level of closed loop H^∞ performance may be achieved with a reduced order controller.

Other work has revealed that the equations solved in the Youla parameterization method which render some of the controller states unobservable to the controller

outputs have a direct interpretation in linear geometric control theory. It is shown that if certain reduced order compensators can be fabricated by linear geometric control methods, it is implied that low order solutions to the equations of the Youla parameterization method exist and yield a reduced order stabilizing controller. Manipulation of certain equations shows that the two methods are equivalent.

The remainder of this document is now summarized. A brief presentation of mathematical background material is undertaken in Chapter 2. Chapter 3 summarizes the Youla parameterization method employed by (Gu, 1993) for the design of reduced order controllers and presents a new approach to the problem which yields controllers with order equal to the minimum of $v_o - 1$ and $v_c - 1$ that achieve arbitrary closed loop eigenvalue placement. Chapter 4 contains a proof that the approach of Chapter 3 is valid and presents a computationally efficient and numerically stable algorithm which solves the equations of the Youla parameterization method. Chapter 5 shows how the method can be applied to certain design plants to satisfy some of the necessary conditions for the existence of reduced order suboptimal H^∞ controllers. Reduced order suboptimal H^∞ controllers are designed for two example problems. Chapter 6 shows the relationships between the Youla parameterization method and the linear geometric compensator synthesis method of Schumacher (1980). Finally, Chapter 7 contains conclusions and suggestions for further research.

2 Mathematical Background

The work presented in this document deals with selecting parameters within the Youla parameterization of all stabilizing controllers to render some controller states unobservable to the controller outputs, allowing the order to be reduced. Working within this parameterization guarantees that the resulting reduced order controller will stabilize the closed loop system. Since the manipulations take place within state-space realizations in which certain canonical forms and associated properties are used extensively, an introduction to system representations, controllability, and observability are presented, along with the orthogonal controllable canonical form and the orthogonal observable canonical form. Also, because the Youla Parameterization is the foundation of this work, a brief introduction is included.

2.1 System Representations

In modern control approaches to control system analysis and design, a design plant $G(s)$ is typically represented by a set of first order linear differential equations, designated by four matrices A , B , C , and D as described in the introduction. In abbreviated notation, $G(s) = (A, B, C, D)$ denotes this state-space realization of $G(s)$.

The transfer function matrix representation of an m input, l output plant $G(s)$ is obtained from the state-space realization by the equation $G(s) = C(sI-A)^{-1}B+D$, yielding $G(s)$ represented as an m by l matrix of rational functions in the variable s . Thus, state-space realizations and transfer function matrices are equally valid representations of the plant, allowing the designer to perform analysis in whichever form is most amenable to the problem at hand.

Assuming the state-space realization of a design plant is given by

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}\tag{2.1}$$

a change of state variables can be accomplished with an invertible transformation matrix T by letting the transformed variable v satisfy $x = Tv$, then $T^{-1}x = v$, and Equation (2.1) becomes

$$\begin{aligned}T\dot{v} &= ATv + Bu \\ y &= CTv + Du\end{aligned}\tag{2.2}$$

implying that

$$\begin{aligned}\dot{v} &= T^{-1}ATv + T^{-1}Bu = A_v v + B_v u \\ y &= CTv + Du = C_v v + Du\end{aligned}\tag{2.3}$$

Thus, in the new coordinates, $A_v = T^{-1}AT$, $B_v = T^{-1}B$, and $C_v = CT$. Now, in these coordinates,

$$\begin{aligned}G(s) &= C_v(sI-A_v)^{-1}B_v + D = (CT)T^{-1}(sI-A)T(T^{-1}B) + D \\ &= C(sI-A)^{-1}B + D\end{aligned}\tag{2.4}$$

illustrating that the behavior of $G(s)$ is independent of the choice of coordinate system. The freedom to use state transformations allows the designer to choose coordinates which display certain system characteristics or are most amenable to the design problem at hand. Controllable canonical form, observable canonical form, and other canonical forms presented in most modern control textbooks can be obtained for any state-space realization through the use of an appropriate state transformation.

2.2 Controllability and Observability of a Realization

An n state realization $G(s) = (A, B, C, D)$ is fully state controllable if all of the system states can be driven to the origin in finite time by a suitable control input. A common test for controllability is to form the controllability test matrix given by

$$[B \ AB \ A^2B \ \dots \ A^{n-1}B] \quad (2.5)$$

and determine its rank. If the rank is n , the system is fully state controllable. Assuming a system is controllable, the controllability index is the smallest number $v_c \leq n$ yielding a rank n matrix

$$[B \ AB \ A^2B \ \dots \ A^{v_c-1}B] \quad (2.6)$$

Often, for a MIMO system, the controllability index is considerably lower than the system order.

A system is fully state observable if all of the system initial states can be determined by observing the system inputs and outputs for a finite time. If the observability test matrix

$$\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (2.7)$$

is full rank, the system is fully state observable. For an observable system, the observability index is the smallest number $v_o \leq n$ yielding a full rank matrix

$$\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ \vdots \\ CA^{v_o-1} \end{bmatrix}. \quad (2.8)$$

The observability index is often lower than the system order for a MIMO system. Throughout the remainder of this document, the term controllable is used to indicate full state controllability, and the term observable is used to indicate full state observability.

In order to design stabilizing controllers for $G(s)$, any unstable modes must lie in the controllable subspace of $G(s)$ (i.e. $G(s)$ is stabilizable) and they must not lie in the unobservable subspace of $G(s)$ (i.e. $G(s)$ is detectable). If any unstable mode does not satisfy these conditions the system cannot be stabilized (Maciejowski, 1989, p. 226). Furthermore, since the input/output behavior of a system is completely characterized by the controllable and observable part of a state-space realization representing it, the system can be assumed to be controllable and observable (i.e.

minimal) without loss of generality. Minimality is assumed throughout the remainder of this document. In order to simplify analysis it will also be assumed that the D matrix of the state-space realization of $G(s)$ is zero which can also be done without loss of generality (Maciejowski, 1989, pp. 305-306).

2.3 The Orthogonal Controllable Canonical Form

The orthogonal controllable canonical form (OCCF) of an n state m input system is presented in (Petkov, et. al., 1991) and has the structure

$$A_c = \begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} & \cdots & A_{1,v_c-1} & A_{1,v_c} \\ A_{2,1} & A_{2,2} & A_{2,3} & \cdots & A_{2,v_c-1} & A_{2,v_c} \\ 0 & A_{3,2} & A_{3,3} & \cdots & A_{3,v_c-1} & A_{3,v_c} \\ 0 & 0 & A_{4,3} & \cdots & A_{4,v_c-1} & A_{4,v_c} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A_{v_c,v_c-1} & A_{v_c,v_c} \end{bmatrix}, \quad B_c = \begin{bmatrix} B_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (2.9)$$

where v_c is the controllability index of the system, B_1 is $m_1 \times m$ and full row rank, $A_{i,i-1}$, $i = 1, 2, \dots, v_c$, are $m_i \times m_{i-1}$ and full row rank, and the numbers m_i satisfying

$$m \geq m_1 \geq m_2 \geq \cdots \geq m_{v_c} \quad m_1 + m_2 + \cdots + m_{v_c} = n \quad (2.10)$$

are the Kronecker indices of the pair (A, B) (Kailath, 1980, p. 396). C_c has no special properties in this realization. The OCCF is obtained through the use of orthogonal transformations, so it may be calculated in a numerically stable manner. The algorithm which calculates the OCCF yields the m_i , $i = 1, 2, 3, \dots, v_c$ and the controllability index v_c as outputs.

2.4 The Orthogonal Observable Canonical Form

Petkov, et. al., (1991) also presents the orthogonal observable canonical form of an n state l output system as

$$A_o = \begin{bmatrix} A_{1,1} & A_{1,2} & 0 & 0 & \dots & 0 \\ A_{2,1} & A_{2,2} & A_{2,3} & 0 & \dots & 0 \\ A_{3,1} & A_{3,2} & A_{3,3} & A_{3,4} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{v_o-1,1} & \dots & \dots & \dots & \dots & A_{v_o-1,v_o} \\ A_{v_o,1} & A_{v_o,2} & A_{v_o,3} & A_{v_o,4} & \vdots & A_{v_o,v_o} \end{bmatrix}, \quad (2.11)$$

$$C_o = [C_1 \ 0 \quad \dots \quad 0],$$

where v_o is the observability index of the system, C_1 is $l \times m_1$ and full column rank, and $A_{i,i+1}$, $i = 1, 2, \dots, v_o-1$, are $m_i \times m_{i+1}$ and full column rank, and the numbers m_i satisfying

$$l \geq m_1 \geq m_2 \geq \dots \geq m_{v_o} \quad m_1 + m_2 + \dots + m_{v_o} = n \quad (2.12)$$

are the Kronecker indices of the pair (C, A) (Kailath, 1980, p. 396). B_o has no special properties in these coordinates. Like OCCF, the OOCF is obtained in a numerically stable manner by orthogonal state transformations. The algorithm which calculates OOCF yields $m_1, m_2, \dots, m_{v_o}$ and the controllability index v_o as outputs.

An interesting relationship between OCCF and OOCF is that they are dual realizations. The OOCF can be obtained for a realization $G(s)=(A, B, C, D)$ by putting the dual realization $G_d(s)=(A^T, C^T, B^T, D^T)$ into OCCF, then dualizing the resulting matrices according to $A_o = A_{c_d}^T$, $B_o = C_{c_d}^T$, $C_o = B_{c_d}^T$, $D_o = D$, where A_{c_d} ,

B_{c_d} , and C_{c_d} are the matrices resulting from transforming the dual realization to OOCF. Thus, one algorithm can be used to calculate either canonical form. These canonical forms are used extensively in subsequent chapters. Algorithms to transform a general realization to OOCF or OCCF have been implemented in the Matlab® m-files Orthocf.m and Orthccf.m which are contained in Appendix B.

2.5 The Youla Parameterization of All Stabilizing Controllers of $G(s)$

Figure 2.1 displays a block diagram typically used to study the internal stability of a feedback control system, and is the block diagram upon which the Youla parameterization is based. It should be noted that negative feedback is employed.

Figure 2.1: Block Diagram for Stability Analysis

For any proper system $G(s)$ in transfer function matrix representation, there exist eight stable rational transfer function matrices satisfying the equations (Francis, 1987, p. 22):

$$G(s) = N(s)M^{-1}(s) = \tilde{M}^{-1}(s)\tilde{N}(s) \quad (2.13)$$

$$\begin{bmatrix} \tilde{X}(s) & -\tilde{Y}(s) \\ -\tilde{N}(s) & \tilde{M}(s) \end{bmatrix} \begin{bmatrix} M(s) & Y(s) \\ N(s) & X(s) \end{bmatrix} = I. \quad (2.14)$$

These equations together form a doubly coprime factorization of $G(s)$. State-space realizations for the coprime factors $N(s)$, $M(s)$, $X(s)$, and $Y(s)$ are obtained via a state feedback matrix F and an output injection matrix H such that the matrices $(A+BF)$ and $(A+HC)$ have stable eigenvalues. The factors are represented as (Nett, 1984):

$$\begin{aligned} N(s) &= (A+BF, \quad B, \quad C, \quad 0) \\ M(s) &= (A+BF, \quad B, \quad F, \quad I) \\ X(s) &= (A+BF, \quad -H, \quad C, \quad I) \\ Y(s) &= (A+BF, \quad -H, \quad F, \quad 0) \end{aligned} \quad (2.15)$$

The doubly coprime factorization is an integral component of the Youla parameterization of all stabilizing controllers of $G(s)$. Theorem 1 of (Francis, 1987) shows that the set of all stabilizing controllers $K(s)$ of the plant $G(s)$ is parameterized in terms of any stable rational transfer function matrix $Q(s)$ by the following equations:

$$\begin{aligned} K(s) &= -[Y(s) - M(s)Q(s)][X(s) - N(s)Q(s)]^{-1} \\ &= -[\tilde{X}(s) - Q(s)\tilde{N}(s)]^{-1}[\tilde{Y}(s) - Q(s)\tilde{M}(s)] \end{aligned} \quad (2.16)$$

Equation (2.16) is the Youla (or Q) parameterization of all stabilizing controllers of $G(s)$. A state-space realization of $K(s)$ may be obtained in the following manner: Let $Q(s)$ be represented by the state-space realization $Q(s) = (\hat{A}, \hat{B}, \hat{C}, \hat{D})$, then insert Equation (2.15) and the state-space realization of $Q(s)$ into the Youla parameterization equation

$$K(s) = -[Y(s) - M(s)Q(s)][X(s) - N(s)Q(s)]^{-1} . \quad (2.17)$$

Performing appropriate matrix operations yields a controller Youla parameterization state-space realization $K(s) = (A_k, B_k, C_k, D_k)$ given by the equations (Gu, 1993), (Maciejowski, 1989, p. 289)

$$\begin{aligned} A_k &= \begin{bmatrix} A+BF+HC+B\hat{D}C & B\hat{C} \\ \hat{B}C & \hat{A} \end{bmatrix} & B_k &= \begin{bmatrix} H+B\hat{D} \\ \hat{B} \end{bmatrix} \\ C_k &= [F+\hat{D}C \quad \hat{C}] & D_k &= [\hat{D}] \end{aligned} \quad (2.18)$$

It should be noted that although the Youla parameterization has been presented based on continuous time control systems, the results hold for discrete time systems as well.

$Q(s)$ is the free parameter in the Youla parameterization of all stabilizing controllers of $G(s)$; $K(s)$ stabilizes $G(s)$ with any $Q(s)$ for which $\hat{A}$ has all stable eigenvalues. The designer has complete freedom in choosing $Q(s)$, as long as it is stable, and it is guaranteed that the controller stabilizes the closed loop system. It would seem that the Youla parameterization is a very powerful design tool; however, in practice it is often difficult to determine a $Q(s)$ which yields a controller that meets any given set of design specifications. While in general the selection of $Q(s)$ to yield low order controllers satisfying closed loop design criteria is an open area of research, it will be shown in subsequent sections that for some MIMO systems, a judicious choice of $Q(s)$ can yield reduced order controllers.

3 Reduced Order Controller Design

Gu (1993) uses the state-space realization of the Youla Parameterization of all stabilizing controllers in Equation (2.18) to derive sufficient conditions for the existence of reduced order stabilizing controllers. Assuming $G(s)$ is a minimal, n state, m input, l output plant with zero D matrix it is shown by applying a state transformation to the controller $K(s)$ that if there exists a stable $Q(s) = (\hat{A}, \hat{B}, \hat{C}, \hat{D})$ such that some of the controller states can be rendered unobservable to the controller outputs, the controller order can be reduced by discarding the unobservable states. The reduced order controller is guaranteed to stabilize $G(s)$ since the manipulations take place within the Youla parameterization. A summary of the work follows.

Let the design plant $G(s) = (A, B, C, 0)$ and let $K(s) = (A_k, B_k, C_k, D_k)$ be the Youla parameterization of all stabilizing controllers in Equation (2.18), where matrices F and H are such that $(A+BF)$ and $(A+HC)$ are stable. Applying a state transformation

$$T = \begin{bmatrix} I & 0 \\ -X & I \end{bmatrix} \quad (3.1)$$

to $K(s)$, a realization $K(s) = (T^{-1}A_k T, T^{-1}B_k, C_k T, D_k)$ is obtained, defined by:

$$T^{-1}A_k T = \begin{bmatrix} K_{a11} & K_{a12} \\ K_{a21} & K_{a22} \end{bmatrix} \quad (3.2)$$

$$\begin{aligned} K_{a11} &= A + BF + HC + B\hat{D}C - B\hat{C}X & K_{a12} &= B\hat{C} \\ K_{a21} &= XA + XBF + XHC + XB\hat{D}C + \hat{B}C - XB\hat{C}X - \hat{A}X & K_{a22} &= XB\hat{C} + \hat{A} \end{aligned}$$

$$T^{-1}B_k = \begin{bmatrix} K_{b1} \\ K_{b2} \end{bmatrix} = \begin{bmatrix} H + B\hat{D} \\ XH + XB\hat{D} + \hat{B} \end{bmatrix} \quad (3.3)$$

$$C_k T = [K_{c1} \ K_{c2}] = [F + \hat{D}C - \hat{C}X \ \hat{C}] \quad (3.4)$$

$$D_k = [\hat{D}] . \quad (3.5)$$

If there exists a matrix X and stable $Q(s) = (\hat{A}, \hat{B}, \hat{C}, \hat{D})$ such that K_{c1} and K_{a21} both equal zero, then the order of the controller can be reduced to a dimension equal to the number of rows in X . Examination of the K_{c1} and K_{a21} equations reveals that in order for such an X and $Q(s)$ to exist, the following equations must be satisfied:

$$\hat{A}X = X(A + HC) + \hat{B}C , \quad (3.6)$$

$$F = \hat{C}X - \hat{D}C . \quad (3.7)$$

Equation (3.6) is a Sylvester equation, which crops up often in controls literature and has been the object of much study (Kucéra, 1974), (Tsui, 1987). Letting n_q denote the size of the square matrix $\hat{A}$ and letting n denote the size of $(A + HC)$, if there exists an n_q by n matrix X satisfying Equations (3.6) and (3.7), then a reduced order controller $K_r(s) = (A_r, B_r, C_r, D_r)$ of order n_q exists, where

$$\begin{aligned} A_r &= \hat{A} + XBC\hat{C} & C_r &= \hat{C} \\ B_r &= \hat{B} + XH + XB\hat{D} & D_r &= \hat{D} \end{aligned} \quad (3.8)$$

Referring to Figure 2.1, it is apparent that closing the loop with a controller $K_r(s) = (A_r, B_r, C_r, D_r)$ results in the following set of closed loop state equations:

$$\begin{bmatrix} \dot{x} \\ \dot{x}_c \end{bmatrix} = \begin{bmatrix} A - BD_rC & -BC_r \\ B_rC & A_r \end{bmatrix} \begin{bmatrix} x \\ x_c \end{bmatrix} + \begin{bmatrix} B & -BD_r \\ 0 & B_r \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}, \quad (3.9)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} C & 0 \\ D_rC & C_r \end{bmatrix} \begin{bmatrix} x \\ x_c \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & D_r \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}. \quad (3.10)$$

Inserting the expressions for the reduced order controller $K_r(s)$ from Equation (3.8) yields the closed loop A matrix

$$A_{cl} = \begin{bmatrix} A - B\hat{D}C & -B\hat{C} \\ \hat{B}C + XH + XB\hat{D}C & \hat{A} + XBC\hat{C} \end{bmatrix}. \quad (3.11)$$

Applying the state transformation of Equation (3.1) to A_{cl} and performing some algebraic manipulations reveals that

$$T^{-1}A_{cl}T = \begin{bmatrix} I & 0 \\ X & I \end{bmatrix} \begin{bmatrix} A - B\hat{D}C & -B\hat{C} \\ \hat{B}C + XH + XB\hat{D}C & \hat{A} + XBC\hat{C} \end{bmatrix} \begin{bmatrix} I & 0 \\ -X & I \end{bmatrix} = \begin{bmatrix} A + B(\hat{C}X - \hat{D}C) & -B\hat{C} \\ X(A + HC) - \hat{A}X & \hat{A} \end{bmatrix}. \quad (3.12)$$

Inserting the results of Equations (3.6) and (3.7) shows that in this coordinate system,

$$A_{cl} = \begin{bmatrix} A + BF & -B\hat{C} \\ 0 & \hat{A} \end{bmatrix}. \quad (3.13)$$

Hence, the eigenvalues of the closed loop system with the reduced order controller are the eigenvalues of $(A + BF)$ along with those of $\hat{A}$.

In light of these facts, the reduced order controller design becomes a problem of determining a low order stable $Q(s)$ of order n_q and an $n_q \times n$ matrix X which satisfy Equations (3.6) and (3.7), yielding an order n_q controller. The approach taken in (Gu, 1993) appears to be one of choosing a low order stable $\hat{A}$ and $\hat{B}$, solving Equation (3.6) for X , then searching for a stabilizing F and an associated $\hat{C}$ and $\hat{D}$ which satisfy Equation (3.7). The author is unaware of any general characterization of matrices F such that $(A+BF)$ is stable, which appears to limit the practical utility of the procedure. Gu (1993) indicates that research to date has yielded no guaranteed method of determining stable realizations $Q(s)$ and $n_q \times n$ matrices X which solve Equations (3.6) and (3.7) to yield controllers with order less than $n-m-l+1$.

3.1 Strictly Proper Reduced Order Controllers of Order v_o

In an attempt to create a general reduced order controller design procedure based on the work of (Gu, 1992), a fundamental change in the approach to the problem is employed. Rather than (1) choosing $\hat{A}$ and $\hat{B}$, (2) solving Equation (3.6) for X , and (3) searching for a stabilizing F that will satisfy Equation (3.7), an approach with greater utility is to first choose a stabilizing F , then examine Equations (3.7) and (3.6) to determine if there exists a stable realization $Q(s) = (\hat{A}, \hat{B}, \hat{C}, \hat{D})$ of order n_q and an $n_q \times n$ matrix X that satisfies them. Such an approach has been used to develop a design procedure which yields strictly proper reduced order stabilizing controllers with order equal to the plant observability index v_o for MIMO systems. Since many MIMO plants have observability indices which are much lower than the plant order, this method can often yield reduced order controllers.

As stated previously, it is assumed without loss of generality that the MIMO plant is minimal, and that the D matrix is zero. Any minimal MIMO plant can be stabilized by a rank 1 feedback matrix F , provided the A matrix for the plant is cyclic, i.e. the characteristic polynomial of the A matrix is equal to the minimal polynomial (Kailath, 1980, p. 659). If the A matrix is not cyclic, constant output feedback can be applied to yield a closed loop system that is cyclic; in almost all cases any arbitrary constant output feedback will suffice (Brasch and Pearson, 1970). The controller design is then performed on the plant with the constant output feedback in place. If the A matrix is cyclic or has been made cyclic by applying constant output feedback, a rank 1 matrix F can be determined which will yield arbitrary closed loop eigenvalue placement (Kailath, 1980, p. 504). Assuming such an F has been determined let it be represented by

$$F = \bar{q}^T \bar{f} = \begin{bmatrix} q_1 \\ \vdots \\ q_m \end{bmatrix} \begin{bmatrix} f_1 & \cdots & f_n \end{bmatrix} . \quad (3.14)$$

Rearrange Equation (3.6), as shown below, letting $\hat{A}$, $(A+HC)$, and C be specified and letting the x 's and b 's be unknowns.

$$[\hat{A}] \begin{bmatrix} f_1 & f_2 & \cdots & f_n \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & & \vdots \\ x_{v_o,1} & x_{v_o,2} & \cdots & x_{v_o,n} \end{bmatrix} = \begin{bmatrix} f_1 & f_2 & \cdots & f_n \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & & \vdots \\ x_{v_o,1} & x_{v_o,2} & \cdots & x_{v_o,n} \end{bmatrix} [A+HC] + \begin{bmatrix} b_{1,1} & \cdots & b_{1,l} \\ \vdots & & \vdots \\ b_{v_o,1} & \cdots & b_{v_o,l} \end{bmatrix} [C] \quad (3.15)$$

Equation (3.15) is merely a set of $v_o * n$ linear equations with a number of unknowns greater than or equal to the number of equations. It will be shown in a later

chapter that if the realization $G(s) = (A, B, C, 0)$ is in orthogonal observable canonical form and the stable matrix $\hat{A}$ is specified with a particular structure then a solution to the above set of equations is guaranteed to exist. Once the solution has been calculated, it is a simple matter to obtain the $m \times n$ matrix $\hat{C}$ and the $m \times l$ matrix $\hat{D}$ which solve Equation (3.7); the matrices are given by

$$\hat{C} = \begin{bmatrix} q_1 & 0 & \dots & 0 \\ q_2 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ q_m & 0 & \dots & 0 \end{bmatrix}, \quad \hat{D} = \begin{bmatrix} 0 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & 0 \end{bmatrix}. \quad (3.16)$$

Since Equations (3.15) and (3.16) are merely rewritten versions of Equations (3.6) and (3.7), the matrices $\hat{A}$, $\hat{B}$, $\hat{C}$, $\hat{D}$, and X which solve Equations (3.15) and (3.16) are also solutions to Equations (3.6) and (3.7). Once these matrices have been obtained, the strictly proper v_o order stabilizing controller is constructed according to Equation (3.8).

3.2 Strictly Proper Reduced Order Controllers of Order v_c

In some cases the controllability index v_c of a MIMO plant may be less than the observability index v_o ; clearly, in this event a controller of order v_c is preferred over a controller of order v_o . Strictly proper controllers of order v_c can be designed by a procedure which is a dualization of the procedure used to calculate controllers of order v_o .

Let the design plant be $G(s) = (A, B, C, 0)$ and let $K(s) = (A_k, B_k, C_k, D_k)$ be the Youla parameterization of all stabilizing controllers in Equation (2.18), where

matrices F and H are such that $(A+BF)$ and $(A+HC)$ are stable. Applying a state transformation

$$T = \begin{bmatrix} I & -X \\ 0 & I \end{bmatrix} \quad (3.17)$$

to $K(s)$, in the new coordinates a realization $K(s) = (T^{-1}A_k T, T^{-1}B_k, C_k T, D_k)$ is obtained, defined by:

$$T^{-1}A_k T = \begin{bmatrix} K_{a11} & K_{a12} \\ K_{a21} & K_{a22} \end{bmatrix} \quad (3.18)$$

$$K_{a11} = A + BF + HC + B\hat{D}C - X\hat{B}\hat{C} \quad K_{a12} = (A + BF)X + B\hat{C} - X\hat{A} + (H + B\hat{D} - X\hat{B})CX$$

$$K_{a21} = \hat{B}C \quad K_{a22} = \hat{A} + \hat{B}CX$$

$$T^{-1}B_k = \begin{bmatrix} K_{b1} \\ K_{b2} \end{bmatrix} = \begin{bmatrix} H + B\hat{D} - X\hat{B} \\ \hat{B} \end{bmatrix} \quad (3.19)$$

$$C_k T = [K_{c1} \ K_{c2}] = [F + \hat{D}C \quad FX + \hat{D}CX + \hat{C}] \quad (3.20)$$

$$D_k = [\hat{D}] . \quad (3.21)$$

Examination of the equations for K_{b1} and K_{a12} reveals that if there exist an $n \times n_q$ matrix X and a stable $Q(s) = (\hat{A}, \hat{B}, \hat{C}, \hat{D})$ of order n_q satisfying

$$X\hat{A} = (A + BF)X + B\hat{C} \quad (3.22)$$

and

$$H = X\hat{B} - B\hat{D}, \quad (3.23)$$

then an order n_q controller exists, given by

$$\begin{aligned} A_r &= \hat{A} + \hat{B}CX & C_r &= FX + \hat{D}CX + \hat{C} \\ B_r &= \hat{B} & D_r &= \hat{D} \end{aligned} \quad (3.24)$$

Now, if both sides of Equations (3.22) and (3.23) are transposed, the result is

$$\begin{aligned} \hat{A}^T X^T &= X^T (A^T + B^T F^T) + \hat{C}^T B^T \\ H^T &= \hat{B}^T X^T - \hat{D}^T B^T \end{aligned} \quad (3.25)$$

which are the Equations (3.6) and (3.7) that would result from calculating an order v_o controller for the dual system $G_d(s) = (A_d, B_d, C_d, 0)$ where $A_d = A^T$, $B_d = C^T$, and $C_d = B^T$ using a stabilizing state feedback $F_d = H^T$ and a stabilizing output injection $H_d = F^T$. Assuming a procedure exists to solve Equations (3.6) and (3.7) to design an order v_o controller for a realization $G(s) = (A, B, C, 0)$, an order v_c controller can be calculated by applying the algorithm to the dual plant realization in the following manner.

Given a realization $G(s) = (A, B, C, 0)$ and matrices F and H such that $(A+BF)$ and $(A+HC)$ are stable, form the dual realization and dual state feedback and output injection according to $A_d = A^T$, $B_d = C^T$, $C_d = B^T$, $F_d = H^T$, and $H_d = F^T$. This results in $(A_d + B_d F_d) = (A^T + C^T H^T) = (A + HC)^T$ and $(A_d + H_d C_d) = (A^T + F^T B^T) = (A + BF)^T$ both being stable. Since the controllability test matrix for $G(s) = (A, B, C, 0)$ is the transpose of the observability test matrix of the dual realization $G_d(s) = (A_d, B_d, C_d, 0)$, the observability index of the dual realization is equal to the controllability index of the original realization. Thus, controllers of order v_c for the realization $G(s) = (A, B, C, 0)$ can be calculated by: (1) finding a rank 1 matrix H such that $(A+HC)$ is stable, (2) finding a matrix F such that $(A+BF)$ is

stable, (3) forming the dual realization, dual state feedback and dual output injection as outlined above, (4) generating an order v_o stable $Q_d(s) = (\hat{A}_d, \hat{B}_d, \hat{C}_d, 0)$ and a $v_o \times n$ matrix X which solve Equations (3.15) and (3.16) for the dual realization, then (5) forming the reduced order controller according to Equation (3.24) using $\hat{A} = \hat{A}_d^T$, $\hat{B} = \hat{C}_d^T$, $\hat{C} = \hat{B}_d^T$, $\hat{D} = 0$, and $X = X_d^T$.

This procedure shows that designing strictly proper controllers of order v_c is simply a process of designing a strictly proper controller of order equal to the observability index of the dual realization. Thus, one set of tools which generates strictly proper controllers of order v_o for a realization can be used to design strictly proper controllers of order equal to the minimum of v_o and v_c .

3.3 Reduced Order Controllers of Order Equal to $\min(v_o-1, v_c-1)$

If it is not required that reduced order controllers be strictly proper, controllers of order v_o-1 can be designed by slightly altering the problem setup shown in Equation (3.15). Letting the m input, l output design plant be given by $G(s) = (A, B, C, 0)$ and assuming that C is full rank, the m_1 Kronecker index of the OOCF is equal to l . Supposing that

$$F = \bar{q}^T \bar{f} = \begin{bmatrix} q_1 \\ \vdots \\ q_m \end{bmatrix} \begin{bmatrix} f_1 & \dots & f_n \end{bmatrix} . \quad (3.26)$$

is such that $(A+BF)$ is stable, Equation (3.6) may be rewritten as Equation (3.27), where $\hat{A}$, $(A+HC)$, and C are specified and the x 's and b 's are unknowns. Equation

(3.27) differs from Equation (3.15) in that the first l elements in row 1 of X are allowed to be unknowns; in Equation (3.15) they were specified. Also, X has v_o-1 rows in Equation (3.27), whereas X has v_o rows in Equation (3.15). Equation (3.27)

$$[\hat{A}] \begin{bmatrix} x_{1,1} & \cdots & x_{1,l} & f_{l+1} & \cdots & f_n \\ x_{2,1} & \cdots & x_{2,l} & x_{2,l+1} & \cdots & x_{2,n} \\ \vdots & & \vdots & \vdots & & \vdots \\ x_{v_o-1,1} & \cdots & x_{v_o-1,l} & x_{v_o-1,l+1} & \cdots & x_{v_o-1,n} \end{bmatrix} =$$

$$\begin{bmatrix} x_{1,1} & \cdots & x_{1,l} & f_{l+1} & \cdots & f_n \\ x_{2,1} & \cdots & x_{2,l} & x_{2,l+1} & \cdots & x_{2,n} \\ \vdots & & \vdots & \vdots & & \vdots \\ x_{v_o-1,1} & \cdots & x_{v_o-1,l} & x_{v_o-1,l+1} & \cdots & x_{v_o-1,n} \end{bmatrix} [A+HC] + \begin{bmatrix} b_{1,1} & \cdots & b_{1,l} \\ \vdots & & \vdots \\ b_{v_o-1,1} & \cdots & b_{v_o-1,l} \end{bmatrix} [C] \quad (3.27)$$

is a set of $(v_o-1)*n$ equations with the number of unknowns greater than or equal to the number of equations. It will be shown in a subsequent chapter that if the design plant is in OOCF and $\hat{A}$ is specified with a certain structure, then a solution to Equation (3.27) is guaranteed to exist. After the matrices X and $\hat{B}$ are determined by solving Equation (3.27), matrices $\hat{C}$ and $\hat{D}$ which solve Equation (3.16) are given by

$$\hat{C} = \begin{bmatrix} q_1 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ q_m & 0 & \cdots & 0 \end{bmatrix}, \quad \hat{D} = \begin{bmatrix} q_1(x_{1,1}-f_1) & \cdots & q_1(x_{1,l}-f_l) \\ \vdots & & \vdots \\ q_m(x_{1,1}-f_1) & \cdots & q_m(x_{1,l}-f_l) \end{bmatrix}. \quad (3.28)$$

Once the matrices $\hat{A}$, $\hat{B}$, $\hat{C}$, and $\hat{D}$ have been obtained, the stabilizing controller of order v_o-1 is found according to Equation (3.8).

Controllers of order v_c-1 are designed using a dual procedure similar to that discussed in the previous section. Suppose a realization $G(s) = (A, B, C, 0)$ with

$v_c < v_o$ is given, along with a matrix F such that $(A+BF)$ is stable and a rank 1 matrix H such that $(A+HC)$ is stable. The dual realization $G_d(s) = (A_d, B_d, C_d, D_d)$, the dual state feedback F_d , and the dual output injection H_d are formed according to $A_d = A^T$, $B_d = C^T$, $C_d = B^T$, $F_d = H^T$, and $H_d = F^T$. Equations (3.27) and (3.28) are solved for this dual realization, yielding the dual solution matrix X_d and the dual Youla parameterization $Q_d(s) = (\hat{A}_d, \hat{B}_d, \hat{C}_d, \hat{D}_d)$. The reduced order controller is constructed according to Equation (3.24) using $\hat{A} = \hat{A}_d^T$, $\hat{B} = \hat{C}_d^T$, $\hat{C} = \hat{B}_d^T$, and $\hat{D} = 0$. Thus, an algorithm which solves Equations (3.27) and (3.28) can also be used to generate solutions to Equations (3.22) and (3.23), showing that one procedure can be used to fabricate reduced order controllers of order equal to the minimum of v_o-1 and v_c-1 .

It is clear that the design of strictly proper stabilizing controllers of order equal to the minimum of v_o and v_c can be accomplished with a common set of tools, and stabilizing controllers of order equal to the minimum of v_o-1 and v_c-1 can be designed with another common set of tools. Hence, all subsequent discussions will deal only with controllers of order v_o or v_o-1 with the understanding that controllers of order v_c and v_c-1 can be generated using the dual design procedures that have been presented.

3.4 Larger Reduced Order Controllers

Previous sections have shown how controllers of order v_o or v_o-1 can be generated for any minimal linear system realization $G(s) = (A, B, C, 0)$ under the restriction that a rank one state feedback matrix F is employed to create a stable matrix

$(A+BF)$. However, it may be that in some cases it is desirable to utilize a matrix F with rank greater than one; an example is the state feedback matrix generated by H^∞ design algorithms for MIMO systems. Under these circumstances, the design procedure previously described may still yield reduced order controllers in some situations. As an example, if the rank of F is $r > 1$ and $r \cdot v_o < n$ or $r \cdot (v_o - 1) < n$, a reduced order controller of order $r \cdot v_o$ or $r \cdot (v_o - 1)$ can be designed for the rank r feedback matrix F . Let the $m \times n$ matrix F be represented as $F = M\hat{F}$, where M is $m \times r$ and $\hat{F}$ is $r \times n$ and full row rank. The controller is designed by selecting r stable $v_o \times v_o$ or $(v_o - 1) \times (v_o - 1)$ matrices $\hat{A}_i$, $i=1, \dots, r$ and using the procedures of Section 3.1 or 3.3 to design r matrices $X_1, X_2, \dots, X_r$ of dimension $v_o \times n$ or $(v_o - 1) \times n$ satisfying

$$\begin{aligned}\hat{A}_1 X_1 &= X_1(A+HC) + \hat{B}_1 C \\ \hat{A}_2 X_2 &= X_2(A+HC) + \hat{B}_2 C \\ &\vdots \\ \hat{A}_r X_r &= X_r(A+HC) + \hat{B}_r C\end{aligned}\tag{3.29}$$

and

$$\hat{F} = \begin{bmatrix} \vec{f}_1 \\ \vdots \\ \vec{f}_r \end{bmatrix} = \begin{bmatrix} \hat{C}_1 X_1 + \hat{D}_1 C \\ \vdots \\ \hat{C}_r X_r + \hat{D}_r C \end{bmatrix}.\tag{3.30}$$

In Equation (3.30), the rows of $\hat{F}$ are denoted by

$$\hat{F} = \begin{bmatrix} \vec{f}_1 \\ \vdots \\ \vec{f}_r \end{bmatrix}.\tag{3.31}$$

The first row of X_i , $i=1, 2, \dots, r$ is the i_{th} row of the matrix $\hat{F}$ and the matrices $\hat{A}_i$ are stable. Also, the matrices $\hat{C}_1 = \hat{C}_2 = \dots = \hat{C}_r = [1 \ 0 \ \dots \ 0]$, and the matrices $\hat{D}_i$ are row vectors comprised of zeroes if strictly proper controllers are being designed or nonzero otherwise. Stacking the X_i matrices results in the matrix

$$\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_r \end{bmatrix} \quad (3.32)$$

which satisfies

$$\begin{bmatrix} \hat{A}_1 & 0 & \dots & 0 \\ 0 & \hat{A}_2 & \dots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \dots & 0 & \hat{A}_r \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_r \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_r \end{bmatrix} [A+HC] + \begin{bmatrix} \hat{B}_1 \\ \hat{B}_2 \\ \vdots \\ \hat{B}_r \end{bmatrix} C \quad (3.33)$$

and

$$F = M\hat{F} = M \begin{bmatrix} \vec{f}_1 \\ \vec{f}_2 \\ \vdots \\ \vec{f}_r \end{bmatrix} = M \left[\begin{bmatrix} \hat{C}_1 & 0 & \dots & 0 \\ 0 & \hat{C}_2 & \dots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \dots & 0 & \hat{C}_r \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_r \end{bmatrix} + \begin{bmatrix} \hat{D}_1 \\ \hat{D}_2 \\ \vdots \\ \hat{D}_r \end{bmatrix} C \right] \quad (3.34)$$

Thus, there exists an $(r*v_o) \times n$ (or $(r*(v_o-1)) \times n$) matrix X and a stable realization $Q(s) = (\hat{A}, \hat{B}, M\hat{C}, M\hat{D})$ satisfying Equations (3.6) and (3.7), indicating the existence of strictly proper stabilizing controllers of order $r*v_o$ and controllers with possibly nonzero D matrix of order $r*(v_o-1)$. This result is obtained by stacking the results of smaller problems, but it does not guarantee the existence of reduced order

controllers for all MIMO plants stabilized by a state feedback matrix F with rank $r > 1$. However, it can be used as a tool to attempt the design of reduced order controllers for MIMO plants where $r*v_o < n$ or $r*(v_o-1) < n$ and stabilization with rank one state feedback is not desirable. Furthermore, if the designer has the freedom to add linearly independent outputs to a system to reduce v_o , the method allows the evaluation of a tradeoff between adding sensors and reducing controller order. Similar arguments can be used to show that stabilizing controllers of order $r*v_c$ or $r*(v_c-1)$ can be designed using the methods of Section 3.3 or Section 3.4 where an output injection H of rank $r > 1$ must be used in a given implementation.

4 A Proof of the Existence of Reduced Order Controllers

Chapter 3 shows that if a minimal plant realization $G(s) = (A, B, C, 0)$ and a rank 1 matrix F are given such $(A+BF)$ is stable, the equations

$$\hat{A}X = X(A+HC) + \hat{B}C \quad (4.1)$$

and

$$F = \hat{C}X - \hat{D}C \quad (4.2)$$

can be written as Equation (3.15), a system of $v_o * n$ equations with a number of unknowns greater than or equal to the number of equations. Assuming Equation (3.15) can be solved and the matrix $\hat{A}$ is stable, the solution can be used to construct a strictly proper stabilizing controller of order v_o . Chapter 3 also shows that Equation (4.1) can be written as Equation (3.27), a set of $(v_o-1)*n$ equations with a number of unknowns greater than or equal to the number of equations. Assuming a solution exists and the matrix $\hat{A}$ is stable, the solution can be used to construct a stabilizing controller of order v_o-1 with possibly non-zero D matrix. Equation (3.8) shows how to construct the reduced order controllers based on the solutions to Equations (4.1) and (4.2), and duality arguments are used to show that the method can be used to fabricate strictly

proper controllers of order v_c or proper controllers of order $v_c - 1$. This chapter will prove that solutions to Equations (3.15) and (3.27) of Chapter 3 exist and presents efficient algorithms for calculating these solutions.

4.1 A Proof of the Existence of Strictly Proper Controllers of Order v_o

Suppose that $G(s) = (A, B, C, 0)$ is a minimal realization in OOCF, and the matrix

$$F = \bar{q}^T \bar{f} = \begin{bmatrix} q_1 \\ \vdots \\ q_m \end{bmatrix} \begin{bmatrix} f_1 & \dots & f_n \end{bmatrix} \quad (4.3)$$

is such that $(A + BF)$ is stable. As shown in Chapter 3, Equation (4.1) can be written as the system of equations

$$[\hat{A}] \begin{bmatrix} f_1 & f_2 & \dots & f_n \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \vdots & \vdots & & \vdots \\ x_{v_o,1} & x_{v_o,2} & \dots & x_{v_o,n} \end{bmatrix} = \begin{bmatrix} f_1 & f_2 & \dots & f_n \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \vdots & \vdots & & \vdots \\ x_{v_o,1} & x_{v_o,2} & \dots & x_{v_o,n} \end{bmatrix} [A + HC] + \begin{bmatrix} b_{1,1} & \dots & b_{1,l} \\ \vdots & & \vdots \\ b_{v_o,1} & \dots & b_{v_o,l} \end{bmatrix} [C] \quad (4.4)$$

This section will show that specifying the matrix $\hat{A}$ to be stable and tridiagonal with non-zero elements on the first superdiagonal implies the existence of matrices X and $\hat{B}$ which solve Equation (4.4), and Equation (4.2) is solved by matrices $\hat{C}$ and $\hat{D}$ given by

$$F = \hat{C}X + \hat{D}C = \begin{bmatrix} q_1 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ q_m & 0 & \dots & 0 \end{bmatrix} X + \begin{bmatrix} 0 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & 0 \end{bmatrix} C \quad (4.5)$$

In order to simplify notation, it will be understood that the notation v in subsequent discussions will represent v_o , the observability index of $G(s)$.

Let the matrix $\hat{A}$ be a tridiagonal matrix of dimension $v \times v$ with all the entries on the first superdiagonal non-zero, and the entries on the main diagonal and the first subdiagonal specified arbitrarily. $\hat{A}$ will have the following structure:

$$\hat{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & 0 & \cdots & 0 & 0 \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & 0 & 0 \\ 0 & a_{3,2} & a_{3,3} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & a_{v-1,v-1} & a_{v-1,v} \\ 0 & 0 & \cdots & \cdots & a_{v,v-1} & a_{v,v} \end{bmatrix}. \quad (4.6)$$

Since the designer has complete freedom in assigning the entries a_{ij} within the constraint that the entries $a_{k-1,k}$ must be nonzero, the eigenvalues of $\hat{A}$ can be specified arbitrarily by choosing appropriate matrix entries.

Assume the minimal m input, n state, l output design plant realization $G(s) = (A, B, C, 0)$ has a full rank C matrix and is in OOCF, i.e.

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} & 0 & 0 & \cdots & 0 \\ A_{2,1} & A_{2,2} & A_{2,3} & 0 & \cdots & 0 \\ A_{3,1} & A_{3,2} & A_{3,3} & A_{3,4} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{v-1,1} & \cdots & \cdots & \cdots & \cdots & A_{v-1,v} \\ A_{v,1} & A_{v,2} & A_{v,3} & A_{v,4} & \cdots & A_{v,v} \end{bmatrix}, \quad (4.7)$$

$$C = [C_1 \quad 0 \quad \cdots \quad 0],$$

where C_1 is $l \times l$ and full rank, the matrices $A_{i,i}$ are $m_i \times m_i$, and the matrices $A_{i,i+1}$ are $m_i \times m_{i+1}$ and full column rank. The numbers m_i , $i = 1, 2, \dots, v$ are the Kronecker indices of (C, A) which satisfy $l = m_1 \geq m_2 \geq \dots \geq m_v$ and $m_1 + m_2 + \dots + m_v = n$. Due to the structure of the C matrix, the realization $((A+HC), B, C, 0)$ is also in OOCF for any H . For notational convenience, let $(A+HC)$ be denoted as

$$(A+HC) = \begin{bmatrix} A_{1,1} & A_{1,2} & 0 & 0 & \dots & 0 \\ A_{2,1} & A_{2,2} & A_{2,3} & 0 & \dots & 0 \\ A_{3,1} & A_{3,2} & A_{3,3} & A_{3,4} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{v-1,1} & \dots & \dots & \dots & \dots & A_{v-1,v} \\ A_{v,1} & A_{v,2} & A_{v,3} & A_{v,4} & \dots & A_{v,v} \end{bmatrix}. \quad (4.8)$$

Inserting the expressions for $\hat{A}$ and $(A+HC)$ into Equation (4.4) results in the following equation:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & 0 & \dots & 0 & 0 \\ a_{2,1} & a_{2,2} & a_{2,3} & \dots & 0 & 0 \\ 0 & a_{3,2} & a_{3,3} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \dots & a_{v-1,v-1} & a_{v-1,v} \\ 0 & 0 & \dots & \dots & a_{v,v-1} & a_{v,v} \end{bmatrix} \begin{bmatrix} f_1 & f_2 & \dots & f_n \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \vdots & \vdots & & \vdots \\ x_{v,1} & x_{v,2} & \dots & x_{v,n} \end{bmatrix} = \quad (4.9)$$

$$\begin{bmatrix} f_1 & f_2 & \dots & f_n \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \vdots & \vdots & & \vdots \\ x_{v,1} & x_{v,2} & \dots & x_{v,n} \end{bmatrix} \begin{bmatrix} A_{1,1} & A_{1,2} & 0 & 0 & \dots & 0 \\ A_{2,1} & A_{2,2} & A_{2,3} & 0 & \dots & 0 \\ A_{3,1} & A_{3,2} & A_{3,3} & A_{3,4} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{v-1,1} & \dots & \dots & \dots & \dots & A_{v-1,v} \\ A_{v,1} & A_{v,2} & A_{v,3} & A_{v,4} & \dots & A_{v,v} \end{bmatrix} + \begin{bmatrix} b_{1,1} & \dots & b_{1,l} \\ \vdots & & \vdots \\ b_{v,1} & \dots & b_{v,l} \end{bmatrix} [C_1 \ 0 \ \dots \ 0].$$

Let the rows and columns of the matrix X be denoted as

$$X = \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vdots \\ \vec{x}_v \end{bmatrix} = \begin{bmatrix} (xc)_1 & (xc)_2 & \cdots & (xc)_n \end{bmatrix}, \quad (4.10)$$

where $\vec{x}_1 = [f_1 \ f_2 \ \cdots \ f_n]$, and let the columns of $(A+HC)$ be denoted by $(A+HC) = [(ahc)_1 \ (ahc)_2 \ \cdots \ (ahc)_n]$. Also, form a vector of variables $[s_1 \ s_2 \ \cdots \ s_v]$, where $s_1 = m_1$, $s_2 = m_1 + m_2, \dots$, $s_i = \sum_{k=1}^i m_k$. Since the matrix $C = [C_1 \ 0 \ \cdots \ 0]$ consists of zero entries in columns s_1+1 through n , the equations obtained by multiplying the first row of $\hat{A}$ by the columns s_1+1 through n of the matrix X in Equation (4.9) reveals that

$$x_{2j} = [\{\vec{x}_1 * (ahc)_j\} - a_{1,1} * x_{1,j}] / a_{1,2}, \quad j = s_1+1, s_1+2, \dots, n. \quad (4.11)$$

Thus, all but the first s_1 elements of $\vec{x}_2$ can be calculated in this manner. Since columns $(ahc)_{s_1+1}$ through $(ahc)_n$ of $(A+HC)$ have zeros as the first s_1 column entries and all but the first s_1 entries of $\vec{x}_2$ are known, it is apparent from multiplying the second row of $\hat{A}$ by columns s_2+1 through n of X in Equation (4.9) that

$$x_{3j} = [\{\vec{x}_2 * (ahc)_j\} - a_{2,1} * x_{1,j} - a_{2,2} * x_{2,j}] / a_{2,3}, \quad j = s_2+1, s_1+2, \dots, n. \quad (4.12)$$

Proceeding sequentially through the rows of X and following similar reasoning for each row reveals for the i th row that columns $(ahc)_{s_{i-1}+1}$ through $(ahc)_n$ of $(A+HC)$ have zeroes as the first s_{i-1} column entries and all but the first s_{i-1} elements of $\vec{x}_{i-1}$ have previously been calculated. Thus, the elements $x_{i,j}$, $j = s_{i-1}+1, \dots, n$ of the i th row of X , $i = 4, 5, \dots, v$ can be calculated according to

$$x_{ij} = [\{\bar{x}_{i-1} * (ahc)_j\} - a_{i-1,i-2} * x_{i-2,j} - a_{i-1,i-1} * x_{i-1,j}] / a_{i-1,i}, \quad j = s_{i-1} + 1, s_{i-1} + 2, \dots, n \quad (4.13)$$

These scalar equations have solved for the variables x_{ij} in the upper right hand portion of the matrix X ; at this point

$$X = \begin{bmatrix} f_1 & & & \dots & & f_n \\ * & \dots & * & x_{2,s_1+1} & & \dots & x_{2,n} \\ * & & \dots & & * & x_{3,s_2+1} & \dots & x_{3,n} \\ \vdots & & & & & \ddots & & \vdots \\ * & & \dots & & & * & x_{v,s_{v-1}+1} & \dots & x_{v,n} \end{bmatrix}, \quad (4.14)$$

where the f_i and x_{ij} indicate variables that have been specified or calculated, and the $*$'s indicate variables that are not yet known. The remaining elements of X can be calculated by working from lower right to upper left in the following manner.

Consider the last m_v columns of $(A+HC)$,

$$\begin{bmatrix} (ahc)_{s_{v-1}+1} & \dots & (ahc)_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ A_{v-1,v} \\ A_{v,v} \end{bmatrix}. \quad (4.15)$$

By the properties of the OOCF, $A_{v-1,v}$ is of dimension $m_{v-1} \times m_v$ and full column rank, and $A_{v,v}$ is of dimension $m_v \times m_v$. The equations obtained by multiplying the last row of $\hat{A}$ by the last m_v columns of X in Equation (4.9) are

$$\begin{bmatrix} 0 & \cdots & 0 & a_{v-1,v} & a_{v,v} \end{bmatrix} \begin{bmatrix} x_{1,s_{v-1}+1} & \cdots & x_{1,n} \\ x_{2,s_{v-1}+1} & \cdots & x_{2,n} \\ \vdots \\ x_{v,s_{v-1}+1} & \cdots & x_{v,n} \end{bmatrix} = \begin{bmatrix} x_{v,1} & \cdots & x_{v,n} \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ A_{v-1,v} \\ A_{v,v} \end{bmatrix}, \quad (4.16)$$

where

$$\begin{bmatrix} x_{1,s_{v-1}+1} & \cdots & x_{1,n} \\ x_{2,s_{v-1}+1} & \cdots & x_{2,n} \\ \vdots \\ x_{v,s_{v-1}+1} & \cdots & x_{v,n} \end{bmatrix} \quad (4.17)$$

is known because of the scalar calculations that have previously been performed.

Rearranging Equation (4.16) results in the matrix equation

$$\begin{bmatrix} x_{v,s_{v-2}+1} & \cdots & x_{v,s_{v-1}} \end{bmatrix} \begin{bmatrix} A_{v-1,v} \end{bmatrix} = \begin{bmatrix} 0 & \cdots & 0 & a_{v-1,v} & a_{v,v} \end{bmatrix} \begin{bmatrix} x_{1,s_{v-1}+1} & \cdots & x_{1,n} \\ x_{2,s_{v-1}+1} & \cdots & x_{2,n} \\ \vdots \\ x_{v,s_{v-1}+1} & \cdots & x_{v,n} \end{bmatrix} - \begin{bmatrix} x_{v,s_{v-1}+1} & \cdots & x_{v,n} \end{bmatrix} \begin{bmatrix} A_{v,v} \end{bmatrix}. \quad (4.18)$$

Since all quantities to the right of the equal sign are known and the matrix $A_{v-1,v}$ is either square or tall and full column rank, solutions for the vector $\begin{bmatrix} x_{v,s_{v-2}+1} & \cdots & x_{v,s_{v-1}} \end{bmatrix}$ are guaranteed to exist.

The equations generated by multiplying the last two rows of $\hat{A}$ by the $s_{v-2}+1$ through s_v columns of X in Equation (4.9) indicate that

$$\begin{bmatrix} x_{v-1,s_{v-3}+1} & \cdots & x_{v-1,s_{v-2}} \\ x_{v,s_{v-3}+1} & \cdots & x_{v,s_{v-2}} \end{bmatrix} [A_{v-2,v-1}] = \tag{4.19}$$

$$\begin{bmatrix} 0 & \cdots & 0 & a_{v-1,v-2} & a_{v-1,v-1} & a_{v-1,v} \\ 0 & \cdots & 0 & 0 & a_{v,v-1} & a_{v,v} \end{bmatrix} \begin{bmatrix} x_{1,s_{v-1}+1} & \cdots & x_{1,n} \\ x_{2,s_{v-1}+1} & \cdots & x_{2,n} \\ \vdots \\ x_{v,s_{v-1}+1} & \cdots & x_{v,n} \end{bmatrix} - \begin{bmatrix} x_{v-1,s_{v-2}+1} & \cdots & x_{v-1,n} \\ x_{v,s_{v-2}+1} & \cdots & x_{v,n} \end{bmatrix} \begin{bmatrix} A_{v-1,v-1} \\ A_{v,v-1} \end{bmatrix}.$$

Since all quantities to the right of the equal sign are known from previous calculations and because the matrix $A_{v-2,v-1}$ is either square or tall and full rank, solutions for the unknowns

$$\begin{bmatrix} x_{v-1,s_{v-3}+1} & \cdots & x_{v-1,s_{v-2}} \\ x_{v,s_{v-3}+1} & \cdots & x_{v,s_{v-2}} \end{bmatrix} \tag{4.20}$$

are guaranteed to exist.

Proceeding with similar reasoning from lower right to upper left yields solutions for the remaining unknown x_{ij} . Letting $\hat{A}_{r_i}$ denote the i th row of $\hat{A}$, the algorithm below implements this procedure:

for $j = v, v-1, \dots, 2$ % index of column groups in $(A+HC)$, counting down

for $i = v, v-1, \dots, j$ % row index, counting down from v to j

solve the equation

$$[x_{i,s_{j-2}+1} \cdots x_{i,s_{j-1}}] [A_{j-1,j}] = \hat{A}_{r_i} [(xc)_{s_{j-1}+1} \cdots (xc)_{s_j}] - [x_{i,s_{j-1}+1} \cdots x_{i,n}] \begin{bmatrix} A_{jj} \\ \vdots \\ A_{vj} \end{bmatrix} \tag{4.21}$$

end

end.

At each stage of the algorithm the quantities to the right of the equal sign are known, and the matrices $A_{j-1,j}$ are square or tall and full column rank, implying that solutions for the elements $[x_{i,s_{j-2}+1} \dots x_{i,s_{j-1}}]$ exist. Once the algorithm is completed, all the elements of X have been calculated.

The elements of $\hat{B}$ can be calculated in a manner similar to that used in the preceding algorithm. The s_1+1 through n columns of Equation (4.9) have been used to determine the elements of X so that

$$\hat{A} [(xc)_{s_1+1} \dots (xc)_n] = X [(ahc)_{s_1+1} \dots (ahc)_n] ; \quad (4.22)$$

hence, the elements of $\hat{B}$ are determined by the equations from the first $l = m_1 = s_1$ columns of Equation (4.9) given by

$$\hat{B}[C_1] = X [(ahc)_1 \dots (ahc)_{s_1}] - \hat{A} [(xc)_1 \dots (xc)_{s_1}] . \quad (4.23)$$

Since C_1 is square and full rank, a solution exists for $\hat{B}$.

This proof has used properties of the OOCF to show that a judicious choice of a stable matrix $\hat{A}$ and the use of any rank one matrix F such that $(A+BF)$ is stable imply the existence of a $v \times n$ matrix X of the form assumed for Equation (4.4) and a $v \times l$ matrix $\hat{B}$ that solve Equations (4.1) and (4.4). Furthermore, letting

$$\hat{C} = \begin{bmatrix} q_1 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ q_m & 0 & \dots & 0 \end{bmatrix} \quad (4.24)$$

and letting $\hat{D}$ be an $m \times l$ matrix of zeroes shows that the matrix X solves Equation (4.2) with this choice of $\hat{C}$ and $\hat{D}$. The designer has specified the matrix $\hat{A}$ to be

stable, so the matrices $\hat{A}$, $\hat{B}$, $\hat{C}$, $\hat{D}$, and X which solve Equations (4.1) and (4.2) can be used to construct a reduced order stabilizing controller according to Equation (3.8). Furthermore, with this controller, the closed loop eigenvalues are the eigenvalues of $\hat{A}$ along with those of $(A+BF)$ as shown in Chapter 3. Since $\hat{D}$ is a matrix of zeroes, the controller will be strictly proper.

4.2 A Computational Algorithm

The proof presented in the previous section can easily be implemented as a computational algorithm. It is assumed that the m input, n state, l output realization $G(s) = (A, B, C, 0)$ is minimal, has a full rank C matrix, and is in OOCF. Furthermore, it is assumed that a rank 1 state feedback matrix F is given such that $(A+BF)$ is stable and F has the structure indicated in Equation (4.3). The matrix $\hat{A}$ must be stable with the tridiagonal structure described previously.

Using the notation established in the previous section, a computational algorithm can be implemented in the following manner. First, solve for the upper right hand portion of X using the scalar equations presented in the previous section. Recalling that the first row of X is given by $\vec{x}_1 = [f_1 \ \dots \ f_n]$, the variables x_{2,s_1+1} through $x_{2,n}$ are calculated using the algorithm

$$\begin{aligned} & \text{for } j = s_1+1, s_1+2, \dots n \\ & \quad x_{2,j} = [\{\vec{x}_1 * (ahc)_j\} - a_{1,1} * x_{1,j}] / a_{1,2} \\ & \text{end.} \end{aligned}$$

In rows $\vec{x}_i$, $i = 3, 4, \dots n$, the unknowns $x_{i,s_1+1}, \dots, x_{i,n}$ are solved by the algorithm

for i = 3, 4, ... , v % row index
 for j = s_{i-1}+1, s_{i-1}+2, ... , n % column index

$$x_{i,j} = [\{\vec{x}_{i-1} * (ahc)_j\} - a_{i-1,i-2} * x_{i-2,j} - a_{i-1,i-1} * x_{i-1,j}] / a_{i-1,i}$$

 end
end.

At this point the upper right hand elements of X have been calculated. Solving for the elements in the lower left portion of X is accomplished by the algorithm below, which was presented in the previous section.

for j = v, v-1, ..., 2 % index of column groups in (A+HC), counting down
 for i = v, v-1, ... j % row index, counting down from v to j
 solve the equation

$$[x_{i,s_{j-2}+1} \quad \dots \quad x_{i,s_{j-1}}] [A_{j-1,j}] = \hat{A}_{r_i} [(xc)_{s_{j-1}+1} \quad \dots \quad (xc)_j] - [x_{i,s_{j-1}+1} \quad \dots \quad x_{i,n}] \begin{bmatrix} A_{j,j} \\ \vdots \\ A_{v,j} \end{bmatrix} \quad (4.25)$$

end
end.

The elements of $\hat{B}$ are determined by solving the equation

$$\hat{B}[C_1] = X [(ahc)_1 \quad \dots \quad (ahc)_{s_1}] - \hat{A} [(xc)_1 \quad \dots \quad (xc)_{s_1}] \quad (4.26)$$

The algorithm has been implemented as the Matlab® m-file entitled Solvsylv.m; a listing is presented in Appendix B. In the m-file, the method used to solve for a $1 \times m$ vector unknowns $\vec{x}$ in an equation of the form

$$\vec{x}A = \vec{b} \quad (4.27)$$

for a square or tall full column rank matrix A uses the singular value decomposition (SVD). The SVD of an $m \times r$ matrix A is given by

$$A = U \Sigma V^T = U \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_r \\ 0 & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} V^T, \quad (4.28)$$

where U is an orthogonal $m \times m$ matrix, Σ is of dimension $m \times r$, and V is an orthogonal matrix of dimension $r \times r$. Using the SVD, the equation $\bar{x}A = \bar{b}$ implies

$$\bar{x}U \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_r \\ 0 & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} = \bar{b}V. \quad (4.29)$$

Letting $[(xu)_1 \dots (xu)_m]$ denote the vector $\bar{x}U$, and letting the vector $\bar{b}V$ be denoted as $[(bv)_1 \dots (bv)_r]$ with $m \geq r$, the above equation becomes

$$[(xu)_1 \dots (xu)_m] \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_r \\ 0 & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} = [(bv)_1 \dots (bv)_r]. \quad (4.30)$$

It is clear that $[(xu)_1 \ (xu)_2 \ \dots \ (xu)_r] = [(bv)_1/\sigma_1 \ (bv)_2/\sigma_2 \ \dots \ (bv)_r/\sigma_r]$ and the variables $[(xu)_{r+1} \ \dots \ (xu)_m]$ may be assigned arbitrarily; in the m-file they are set equal to zero. Once the variables $\vec{x}U = [(xu)_1 \ \dots \ (xu)_m]$ have been calculated, the vector $\vec{x}$ is determined by

$$\vec{x} = [(xu)_1 \ \dots \ (xu)_m]U^T . \quad (4.31)$$

The algorithm has reduced the large problem of solving for the $v \times n$ unknowns in Equation (4.4) to a problem of solving a number of much smaller sets of equations through a "divide and conquer" technique. The algorithm is numerically efficient because all calculations involve either scalar equations or small matrix equations; no direct matrix inversions are required. Numerical stability is achieved since the OOCF is obtained using only orthogonal transformations, and all calculations in the algorithm utilize only scalar equations or the solution of small matrix equations using the SVD. An analysis of the computational complexity of the solution algorithm is presented in a subsequent section.

4.3 Controllers of Order v_o-1

Suppose that $G(s) = (A, B, C, 0)$ is a minimal realization in OOCF, and the matrix

$$F = \vec{q}^T \vec{f} = \begin{bmatrix} q_1 \\ \vdots \\ q_m \end{bmatrix} \begin{bmatrix} f_1 & \dots & f_n \end{bmatrix} \quad (4.32)$$

is such that $(A+BF)$ is stable. As shown in Chapter 3, Equation (4.1) can be written as the system of equations

$$[\hat{A}] \begin{bmatrix} x_{1,1} & \cdots & x_{1,l} & f_{l+1} & \cdots & f_n \\ x_{2,1} & \cdots & x_{2,l} & x_{2,l+1} & \cdots & x_{2,n} \\ \vdots & & \vdots & \vdots & & \vdots \\ x_{v_o-1,1} & \cdots & x_{v_o-1,l} & x_{v_o-1,l+1} & \cdots & x_{v_o-1,n} \end{bmatrix} = \quad (4.33)$$

$$\begin{bmatrix} x_{1,1} & \cdots & x_{1,l} & f_{l+1} & \cdots & f_n \\ x_{2,1} & \cdots & x_{2,l} & x_{2,l+1} & \cdots & x_{2,n} \\ \vdots & & \vdots & \vdots & & \vdots \\ x_{v_o-1,1} & \cdots & x_{v_o-1,l} & x_{v_o-1,l+1} & \cdots & x_{v_o-1,n} \end{bmatrix} [A+HC] + \begin{bmatrix} b_{1,1} & \cdots & b_{1,l} \\ \vdots & & \vdots \\ b_{v_o,1} & \cdots & b_{v_o,l} \end{bmatrix} [C] .$$

For notational convenience, v will be used to indicate the observability index of $G(s)$.

Let $A+HC$ be denoted as in Equation (4.8), and suppose that $\hat{A}$ is stable and of dimension $v-1 \times v-1$ with the structure given by Equation (4.6). Inserting the previously defined expressions for $\hat{A}$, $(A+HC)$ and C into Equation (4.33) results in

$$\begin{bmatrix} a_{1,1} & a_{1,2} & 0 & \cdots & 0 & 0 \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & 0 & 0 \\ 0 & a_{3,2} & a_{3,3} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & a_{v-2,v-2} & a_{v-2,v-1} \\ 0 & 0 & \cdots & \cdots & a_{v-1,v-2} & a_{v-1,v-1} \end{bmatrix} \begin{bmatrix} x_{1,1} & \cdots & x_{1,l} & f_{l+1} & \cdots & f_n \\ x_{2,1} & \cdots & x_{2,l} & x_{2,l+1} & \cdots & x_{2,n} \\ \vdots & & \vdots & \vdots & & \vdots \\ x_{v-1,1} & \cdots & x_{v-1,l} & x_{v-1,l+1} & \cdots & x_{v-1,n} \end{bmatrix} = \quad (4.34)$$

$$\begin{bmatrix} x_{1,1} & \cdots & x_{1,l} & f_{l+1} & \cdots & f_n \\ x_{2,1} & \cdots & x_{2,l} & x_{2,l+1} & \cdots & x_{2,n} \\ \vdots & & \vdots & \vdots & & \vdots \\ x_{v-1,1} & \cdots & x_{v-1,l} & x_{v-1,l+1} & \cdots & x_{v-1,n} \end{bmatrix} \begin{bmatrix} A_{1,1} & A_{1,2} & 0 & \cdots & 0 \\ A_{2,1} & A_{2,2} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{v-1,1} & \cdots & \cdots & \cdots & A_{v-1,v} \\ A_{v,1} & A_{v,2} & A_{v,3} & \cdots & A_{v,v} \end{bmatrix} +$$

$$\begin{bmatrix} b_{1,1} & \cdots & b_{1,l} \\ \vdots & & \vdots \\ b_{v-1,1} & \cdots & b_{v-1,l} \end{bmatrix} [C_1 \ 0 \ \cdots \ 0] .$$

where the x_{ij} and b_{ij} represent unknowns. The equation is similar to Equation (4.9); the differences are that in this equation the first l elements of $\vec{x}_1$ are unknowns and the matrix X is $(v-1) \times n$ whereas in Equation (4.9) all the entries of $\vec{x}_1$ are specified and X is of dimension $v \times n$.

Let the rows and columns of the matrix X be denoted as shown in Equation (4.10), and let the columns of $(A+HC)$ be denoted as

$$(A+HC) = [(ahc)_1 \ (ahc)_2 \ \dots \ (ahc)_n] \quad (4.35)$$

Once again, form a vector of variables $[s_1 \ s_2 \ \dots \ s_v]$, specified by $s_1 = m_1$, $s_2 = m_1 + m_2, \dots, s_i = \sum_{k=1}^i m_k$, where the m_i are the Kronecker indices of the OOCF.

Examination of the structural properties of $(A+HC)$ and C reveals that the variables

$[x_{2,s_2+1} \ \dots \ x_{2,n}]$ can be calculated according to

$$x_{2,j} = [\{\vec{x}_1 * (ahc)_j\} - a_{1,1} * x_{1,j}] / a_{1,2}, \quad j = s_2+1, s_2+2, \dots, n. \quad (4.36)$$

All but the first $s_2 = m_1 + m_2$ elements of $\vec{x}_2$ are obtained in this manner. In the rows $\vec{x}_i$, $2 < i \leq v-1$, the elements $[x_{i,s_i+1} \ \dots \ x_{i,n}]$ can be calculated by the following algorithm:

for $i = 3, 4, \dots, v-1$ % row index

for $j = s_i+1, s_i+2, \dots, n$ % column index

$$x_{i,j} = [\{\vec{x}_{i-1} * (ahc)_j\} - a_{i-1,i-2} * x_{i-2,j} - a_{i-1,i-1} * x_{i-1,j}] / a_{i-1,i} \quad (4.37)$$

end

end.

At this point, the upper right hand portion of X has been calculated; i.e.

$$X = \begin{bmatrix} * & \dots & * & f_{s_2+1} & \dots & f_n \\ * & & \dots & * & x_{2,s_2+1} & \dots & x_{2,n} \\ \vdots & & & & & \ddots & \vdots \\ * & & \dots & & * & x_{v,s_{v-1}+1} & \dots & x_{v,n} \end{bmatrix}, \quad (4.38)$$

where the f_i and $x_{i,j}$ indicate elements for which the values are known, and the $*$'s indicate unknown values. Letting $\hat{A}_{r_i}$ denote the i th row of $\hat{A}$, the remaining elements of X can be calculated using the algorithm

for $j = v, v-1, \dots, 2$ % index of column groups in $(A+HC)$, counting down
 for $i = v-1, v-2, \dots, j-1$ % row index, counting down from v to j
 solve the equation

$$\begin{bmatrix} x_{i,s_{j-2}+1} & \dots & x_{i,s_{j-1}} \end{bmatrix} \begin{bmatrix} A_{j-1,j} \end{bmatrix} = \hat{A}_{r_i} \begin{bmatrix} (xc)_{s_{j-1}+1} & \dots & (xc)_{s_j} \end{bmatrix} - \begin{bmatrix} x_{i,s_{j-1}+1} & \dots & x_{i,n} \end{bmatrix} \begin{bmatrix} A_{jj} \\ \vdots \\ A_{vj} \end{bmatrix} \quad (4.39)$$

end

end.

Since all quantities to the right of the equal sign are known and the matrices $A_{j-1,j}$ are square or tall and full column rank, the equation in the algorithm has a solution at each iteration.

Finally, the elements of the matrix $\hat{B}$ are determined by the first $s_1 = m_1$ columns of Equation (4.34) given by

$$\hat{B} \begin{bmatrix} C_1 \end{bmatrix} = X \begin{bmatrix} (ahc)_1 & \dots & (ahc)_{s_1} \end{bmatrix} - \hat{A} \begin{bmatrix} (xc)_1 & \dots & (xc)_{s_1} \end{bmatrix}. \quad (4.40)$$

Since C_1 is square and invertible, a solution exists for $\hat{B}$. Once X and $\hat{B}$ have been determined, the matrices $\hat{C}$ and $\hat{D}$ which solve Equation (4.2) are given by

$$\hat{C} = \begin{bmatrix} q_1 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ q_m & 0 & \dots & 0 \end{bmatrix}, \quad \hat{D} = -\left[(F\hat{C}X)_1 \ (F\hat{C}X)_2 \ \dots \ (F\hat{C}X)_{s_1}\right] C_1^{-1}, \quad (4.41)$$

where $(F\hat{C}X)_i$ denotes the i th column of the matrix $F-\hat{C}X$.

It is obvious that the solution procedure for $v-1$ order controllers is similar to the solution method used to design strictly proper v order controllers. In fact, the only differences are in the scalar equations used to calculate the variables x_{ij} , $1 < i < v$, $s_i+1 \leq j \leq n$ in the upper right hand portion of X and in the calculation of $\hat{D}$. The algorithms which solve for the elements in the lower left portion of X and $\hat{B}$ are identical in the two solution procedures. The algorithm for constructing controllers of order $v-1$ has been implemented in the Matlab® m-file titled Redsylyv.m which is contained in Appendix B.

4.4 An Analysis of Computational Complexity

One possible method of calculating a solution to Equation (4.9) would be to place the unknowns x_{ij} and b_{ij} into a column vector N with $v*n + v*l$ elements, then rewriting Equation (4.9) in the form

$$MN = P, \quad (4.42)$$

where M is of dimension $v*n$ by $(v-1)*n + v*l$ and full column rank, and P is a column vector with $(v-1)*n+v*l$ elements, where $v*l \geq n$. Obtaining a solution to

this equation is theoretically possible; it would require approximately $\frac{2}{3}*(v*n)^3$ computational operations (Petkov, et. al., 1991, p. 41).

The method presented in this chapter calculates the elements in the upper right portion of X by solving scalar equations; examination of Equation (4.13) reveals that approximately $2*(n+3)$ multiplications and additions are required to calculate each element. Thus, in calculating the upper right hand portion of X , term cost is approximately $2*(n+3)$ operations per term.

The other terms of X , along with the elements of $\hat{B}$, are calculated using an equation of the form

$$\begin{bmatrix} x_{i,s_{j-2}+1} & \dots & x_{i,s_{j-1}} \end{bmatrix} \begin{bmatrix} A_{j-1,j} \end{bmatrix} = \hat{A}_{r_i} \begin{bmatrix} (xc)_{s_{j-1}+1} & \dots & (xc)_{s_j} \end{bmatrix} - \begin{bmatrix} x_{i,s_{j-1}+1} & \dots & x_{i,n} \end{bmatrix} \begin{bmatrix} A_{jj} \\ \vdots \\ A_{vj} \end{bmatrix} \quad (4.43)$$

In order to simplify the analysis of computational complexity, it will be assumed that all $m_i = l$ since $m_i \leq l$, $\forall i \ni 1 \leq i \leq v$, and $A_{i-1,i}$ is of dimension m_{i-1} by m_i . Also, this assumption will yield a conservative figure for the number of computational operations required for this portion of the solution procedure. In the preceding equation, forming

$$\hat{A}_{r_i} \begin{bmatrix} (xc)_{s_{j-1}+1} & \dots & (xc)_{s_j} \end{bmatrix} \quad (4.44)$$

requires approximately $2*v*l$ multiplies and additions since $\hat{A}_{r_i}$ is a $1 \times v$ vector and $\begin{bmatrix} xc_{s_{j-1}+1} & \dots & xc_{s_j} \end{bmatrix}$ is assumed to be a $v \times l$ matrix. Forming

$$\begin{bmatrix} x_{i,s_{j-1}+1} & \dots & x_{i,n} \end{bmatrix} \begin{bmatrix} A_{jj} \\ \vdots \\ A_{vj} \end{bmatrix} \quad (4.45)$$

requires approximately $2*n*l$ or fewer multiplies and additions since the largest possible size of $[x_{i,s_{j-1}+1} \dots x_{i,n}]$ is $1 \times n$ when $j=1$ and the largest possible dimension of $[A_{j,j} \dots A_{v,j}]^T$ is $n \times l$. Thus, setting up the equation is assumed to require $2*(n+v)*l$ operations. Solving the equation once it has been set up requires approximately $\frac{3}{2}*l^3$ operations (Petkov, et.al., 1991, p. 41). Since there are approximately l terms in the vector $[x_{i,s_{j-2}+1} \dots x_{i,s_{j-1}}]$, the term cost can be defined as

$$\text{term cost} \cong \frac{1}{l} * (\text{setup cost} + \text{solution cost}).$$

Inserting the expressions for the setup cost and the solution cost reveals that

$$\text{term cost} \leq \frac{1}{l} * (2*(n+v)*l + \frac{3}{2}*l^3) = 2*(n+v) + \frac{3}{2}*l^2.$$

Now, the term cost for the upper right portion of X is approximately $2*(n+3)$ operations, while that of the upper left portion of X and $\hat{B}$ is $2*(n+v) + \frac{3}{2}*l^2$ operations. Assuming that $v > 3$, a conservative estimate is that the term cost for any unknown is $2*(n+v) + \frac{3}{2}*l^2$. Since a total of $(v-1)*n + v*l$ unknowns must be calculated, the total number of computational operations required to calculate a solution to Equation (4.9) is approximately $((v-1)*n + v*l)*(2*n*v + \frac{3}{2}*l^2)$.

In all probability, the design procedure presented in this dissertation would not be used unless the plant had the property that v was considerably less than n . Also, many MIMO plants are such that the number of plant states is significantly larger than

the number of outputs. In these circumstances it is reasonable to assume that $n+v \geq \frac{3}{2} * l^2$. Under this assumption, the computational cost can be bounded according to

$$\begin{aligned} \text{cost} &\cong ((v-1)*n + v*l)*(2*(n+v) + \frac{3}{2}*l^2) \\ &\leq v*(n+l)*(2*(n+v) + \frac{3}{2}*l^2) \\ &\leq v*(n+l)*3*(n+v) = v*3*(n^2 + (v+l)*n + l*v). \end{aligned}$$

Finally, assuming $(v+l)*n + l*v \leq n^2$, the computational cost is shown to be on the order of $6*v*n^2$ operations.

Comparing the computational cost of the vectorization solution method and the new method presented in this chapter reveals that the vectorization method requires approximately $\frac{2}{3}*(v*n)^3$ operations, while the new method requires fewer than $6*v*n^2$ operations for a plant realization where $v < n$ and $l < n$. Clearly, under these circumstances the new method is much more efficient in terms of computational effort. Since the procedure for calculating controllers of order $v-1$ is similar to that for calculating order v controllers, the computational complexity required for the order $v-1$ controllers calculations is approximately equal to that of the order v controllers.

4.5 Pole Placement Design Examples

In order to demonstrate the efficacy of the methods presented in this chapter for design of MIMO reduced order controllers, two pole placement examples based on a tenth order model of an actual control system are presented. The design plant for the examples is a tenth order model of the Image Motion Compensator (IMC) subsystem

of the ACES Flexible Structure Controller Test Facility at NASA Marshall Space Flight Center (NMSFC). Figure 4.1 is a diagram of the ACES facility.

Figure 4.1: The ACES Test Facility

The IMC subsystem consists of components 7 through 10 in the preceding figure; its purpose is to maintain laser position on the optical detector in the presence of flexible beam movement. The IMC is a discrete time control system with a sampling rate of 50 Hz. and is modeled by a ten state, two input, two output state-space realization obtained through research efforts at Ohio University (Medina, et. al., 1994). The plant model is tenth order with an observability index $\nu = 5$; the D matrix is assumed to be zero without loss of generality.

Since the reduced order controller design methods that have been presented achieve arbitrary closed loop pole placement, a good illustration of their capability is to design a full order observer which satisfies certain pole placement criteria, then use the new methods to design reduced order controllers which achieve similar pole

placement. For this example, the full order observer/controller compensator is tenth order whereas the reduced order controllers are of order $v = 5$ and $v-1 = 4$. The IMC plant lends itself to this example because the author has previous experience using it in pole placement designs; pole placement criteria for previous designs are the same criteria used for these examples. The design plant realization used for these examples is presented in Appendix A.

The first step is establishing the desired pole locations for the closed loop system. The design specification for this example is to place all poles such that their damping ratio ζ is greater than .5 and the natural frequency ω_n is greater than 10π radians per second. Although these are derived from time domain specifications, it is useful to study the complex frequency plane (s-plane) in order to determine allowable pole location regions which meet them. The shaded portion of Figure 4.2 displays the allowable region.

Figure 4.2: Allowable s-plane Pole Locations

Since it can be difficult to visualize a mapping of this shaded region into the z-plane, the approach used for z-plane pole selection is to first select allowable s-plane locations

then map them to the z -plane via the equation $z=e^{sT}$ where T is the sampling period of the system.

The s -plane pole locations to be placed via state feedback were chosen fairly close to the semicircular border of the shaded region in order to move the poles as little as possible while still meeting the design specifications. The chosen s -plane locations, along with their associated z -plane mappings, are

$$s = \begin{bmatrix} -16 \pm 28i \\ -17 \pm 28i \\ -18 \pm 28i \\ -19 \pm 28i \\ -20 \pm 28i \end{bmatrix} \quad z = \begin{bmatrix} 0.6152 \pm 0.3857i \\ 0.6031 \pm 0.3781i \\ 0.5911 \pm 0.3706i \\ 0.5794 \pm 0.3633i \\ 0.5679 \pm 0.3561i \end{bmatrix} . \quad (4.46)$$

The observer poles were chosen at locations farther to the left in the s -plane so that the observer state estimates will quickly converge to the system states in order to avoid significant errors due to incorrect state estimation. The observer s -plane pole locations and their corresponding z -plane mappings are given by

$$s = \begin{bmatrix} -26 \pm 28i \\ -27 \pm 28i \\ -27.1 \\ -28.9 \\ -29 \pm 28j \\ -30 \pm 28j \end{bmatrix} \quad z = \begin{bmatrix} 0.5037 \pm 0.3158i \\ 0.4937 \pm 0.3095i \\ 0.5816 \\ 0.5610 \\ 0.4744 \pm 0.2974i \\ 0.4650 \pm 0.2915i \end{bmatrix} . \quad (4.47)$$

In order to design a rank one feedback matrix F which achieves the desired state feedback, the method presented in (Kailath, 1980, pp. 503-504) was used. A vector $\vec{q} = [q_1 \ q_2]$ is chosen, then the realization $(A, B\vec{q}^T)$ is tested for controllability. If $(A, B\vec{q}^T)$ is controllable, any pole placement procedure may be used

to produce a vector $\vec{f} = [f_1 \ f_2 \ \dots \ f_n]$ which places the poles of $[A+B\vec{q}^T\vec{f}]$ at the desired pole locations. Clearly then, the matrix $F = \vec{q}^T\vec{f}$ places the eigenvalues of $(A+BF)$ at the same locations, since $(A+BF) = (A+B\vec{q}^T\vec{f})$. In this design the vector $\vec{q}$ was chosen as $\vec{q} = [1 \ 1]$. Using this vector and the set of desired z-plane state feedback poles the Matlab® pole placement algorithm Place.m yields the state feedback matrix

$$F = \begin{bmatrix} 1.514 & 0.112 & -1.352 & 0.280 & 0.794 & 1.147 & -1.899 & 1.640 & 0.412 & 0.493 \\ 1.514 & 0.112 & -1.352 & 0.280 & 0.794 & 1.147 & -1.899 & 1.640 & 0.412 & 0.493 \end{bmatrix}. \quad (4.48)$$

The output injection matrix H such that $(A+HC)$ has the desired observer pole locations is also obtained using Place.m; for this example the result is

$$H = \begin{bmatrix} 6.62 & 0.71 & 4.43 & -0.65 & 7.27 & 5.97 & -130.01 & 326.25 & -175.25 & -219.26 \\ -0.07 & -15.54 & 3.86 & -9.97 & -5.39 & 35.41 & -43.59 & 6.81 & -1190.57 & -559.78 \end{bmatrix}^T. \quad (4.49)$$

Once the state feedback F and the output injection H are obtained, a realization of the closed loop system with the observer/controller compensator is given by:

$$A_c = \begin{bmatrix} A & BF \\ -HC & A+BF+HC \end{bmatrix}, \quad B_c = \begin{bmatrix} B \\ B \end{bmatrix}, \quad C_c = [C \ 0], \quad D_c = [0]. \quad (4.50)$$

The eigenvalues of the closed loop observer/controller compensator are

$$\begin{bmatrix} 0.5610 \\ 0.5816 \\ 0.4650 \pm 0.2915i \\ 0.4744 \pm 0.2974i \\ 0.4937 \pm 0.3096i \\ 0.5037 \pm 0.3158i \\ 0.5683 \pm 0.3560i \\ 0.5783 \pm 0.3632i \\ 0.5923 \pm 0.3708i \\ 0.6024 \pm 0.3779i \\ 0.6153 \pm 0.3858i \end{bmatrix}, \quad (4.51)$$

which are the eigenvalues of $(A+BF)$ along with those of $(A+HC)$, as expected.

The observability index of the IMC model is $v = 5$, so a strictly proper fifth order controller can be designed using the method of Section 4.1 which meets the same pole placement criteria as the full order observer/controller compensator. The eigenvalues of the closed loop system will be those of $\hat{A}$ along with those of $(A+BF)$ as shown in Chapter 3. Let the state feedback matrix F be as shown previously, and let the matrix $\hat{A}$ have its eigenvalues equal to five of the eigenvalues of $(A+HC)$. Using an arbitrary choice of five of the eigenvalues of $(A+HC)$, an $\hat{A}$ meeting these specifications and subsequently used for design is given by

$$\hat{A} = \begin{bmatrix} 0.5816 & 1.0000 & 0 & 0 & 0 \\ 0 & 0.4937 & -0.3095 & 0 & 0 \\ 0 & 0.3095 & 0.4937 & 1.0000 & 0 \\ 0 & 0 & 0 & 0.5037 & -0.3158 \\ 0 & 0 & 0 & 0.3158 & 0.5037 \end{bmatrix}. \quad (4.52)$$

Since the eigenvalues of $\hat{A}$ were chosen as a subset of those of $(A+HC)$ and $(A+BF)$ is unchanged from the full order design, the reduced order controller should yield closed loop eigenvalues which are a subset of those in the full order design. Following the method of Section 4.1, let $\vec{f}$ equal the first row of X , and let $\hat{A}$ be as previously defined, then insert them into the equation

$$[\hat{A}] \begin{bmatrix} f_1 & f_2 & \cdots & f_n \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & & \vdots \\ x_{k,1} & x_{k,2} & \cdots & x_{k,n} \end{bmatrix} = \begin{bmatrix} f_1 & f_2 & \cdots & f_n \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & & \vdots \\ x_{k,1} & x_{k,2} & \cdots & x_{k,n} \end{bmatrix} [A+HC] + \begin{bmatrix} b_{1,1} & \cdots & b_{1,l} \\ \vdots & & \vdots \\ b_{k,1} & \cdots & b_{k,l} \end{bmatrix} [C], \quad (4.53)$$

where the realization $(A+HC, B, C, 0)$ is in OOCF. The m-file Solvsylv.m which implements the algorithm of Section 4.2 calculates an X and $\hat{B}$ which solve the Sylvester equation. The solution matrices are given by

$$X = \begin{bmatrix} 1.514 & 0.112 & -1.352 & 0.280 & 0.794 & 1.147 & -1.899 & 1.640 & 0.412 & 0.493 \\ -1.676 & -1.704 & 1.733 & -0.418 & 1.677 & -0.448 & 0.535 & -0.739 & -0.282 & -0.377 \\ -4.922 & 38.647 & 7.988 & -2.764 & 11.557 & 2.197 & -5.181 & 3.858 & 0.398 & 0.704 \\ 0.496 & -23.903 & -6.978 & -21.043 & 3.895 & -7.359 & 7.098 & -7.716 & -1.582 & -1.904 \\ -52.997 & 69.516 & 32.795 & 40.587 & -97.438 & -41.512 & 13.768 & -14.955 & -6.034 & -3.758 \end{bmatrix} \quad (4.54)$$

and

$$\hat{B} = \begin{bmatrix} 6.261 & -0.599 & 2.425 & 0.049 & 0.061 \\ 21.697 & -2.633 & 5.959 & 0.073 & 0.107 \end{bmatrix}^T \quad (4.55)$$

Examination of the equation $F = \hat{C}X - \hat{D}C$ reveals that

$$C_q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad D_q = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (4.56)$$

are solutions. The reduced order controller $K_r(s) = (A_r, B_r, C_r, D_r)$ is formed according to Equation (3.8) then the loop is closed with negative feedback and the eigenvalues are calculated. A comparison of the closed loop eigenvalues for the full order and reduced order controllers is

$$\lambda_{full} = \begin{bmatrix} 0.4650 \pm 0.2915i \\ 0.4744 \pm 0.2974i \\ 0.5610 \\ 0.5816 \\ 0.4937 \pm 0.3096i \\ 0.5037 \pm 0.3158i \\ 0.5683 \pm 0.3560i \\ 0.5783 \pm 0.3632i \\ 0.5923 \pm 0.3708i \\ 0.6024 \pm 0.3779i \\ 0.6153 \pm 0.3858i \end{bmatrix} \quad \lambda_s = \begin{bmatrix} 0.5816 \\ 0.4937 \pm 0.3095i \\ 0.5037 \pm 0.3158i \\ 0.5680 \pm 0.3561i \\ 0.5793 \pm 0.3632i \\ 0.5912 \pm 0.3706i \\ 0.6030 \pm 0.3781i \\ 0.6152 \pm 0.3857i \end{bmatrix} \quad (4.57)$$

The pole placement with the fifth order controller matches that of the full order controller to the third decimal place, and the closed loop poles obtained with the fifth order controller satisfy the ζ and ω_n specifications, so the design can be deemed to be successful.

The method of Section 4.3 was applied to the ACES plant to design a fourth order controller with nonzero D matrix which also satisfies the pole placement criteria. Suppose $\hat{A}$ is specified so that the eigenvalues match four of the eigenvalues of $(A+HC)$. Choosing four of the eigenvalues of $(A+HC)$ arbitrarily, an $\hat{A}$ meeting this criterion is

$$\hat{A} = \begin{bmatrix} 0.4937 & -0.3095 & 0 & 0 \\ 0.3095 & 0.4937 & 1.0000 & 0 \\ 0 & 0 & 0.5037 & -0.3158 \\ 0 & 0 & 0.3158 & 0.5037 \end{bmatrix}. \quad (4.58)$$

Based on the specified $\hat{A}$, the m-file Redsylyv.m which implements the procedure of Section 4.3 calculates a $4 \times n$ matrix X and a 4×2 matrix $\hat{B}$ which solve Equation (4.1); the matrices are

$$X = \begin{bmatrix} -1.010 & 9.826 & -1.352 & 0.280 & 0.794 & 1.148 & -1.899 & 1.640 & 0.412 & 0.493 \\ 9.484 & -17.639 & 8.468 & 17.227 & -5.644 & 1.122 & -1.192 & 1.922 & 0.795 & 1.079 \\ -12.532 & 39.930 & 25.967 & 29.195 & -18.347 & -4.694 & -4.698 & 3.519 & 0.340 & 0.647 \\ 50.308 & -63.199 & -44.052 & -35.744 & 75.080 & 45.133 & -14.959 & 15.122 & 4.927 & 5.872 \end{bmatrix}. \quad (4.59)$$

and

$$\hat{B} = \begin{bmatrix} 1.5438 & -2.5405 & -0.0344 & -0.0751 \\ 3.9023 & -5.1410 & -0.0462 & -0.1282 \end{bmatrix}^T. \quad (4.60)$$

Matrices $\hat{C}$ and $\hat{D}$ are calculated using Equation (4.41) resulting in

$$\hat{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad \hat{D} = \begin{bmatrix} -39.064 & -242.267 \\ -39.064 & -242.267 \end{bmatrix}. \quad (4.61)$$

The reduced order controller is formed according to Equation (3.8) and the loop is closed with negative feedback; the closed loop eigenvalues are

$$\lambda_4 = \begin{bmatrix} 0.4937 \pm 0.3095i \\ 0.5037 \pm 0.3158i \\ 0.5679 \pm 0.3561i \\ 0.5795 \pm 0.3632i \\ 0.5911 \pm 0.3706i \\ 0.6031 \pm 0.3781i \\ 0.6152 \pm 0.3857i \end{bmatrix} \quad (4.62)$$

which are a subset of the full order controller eigenvalues, as expected. Once again, the eigenvalues obtained with the reduced order controller are very close to those of the full order controller, and they satisfy the ζ and ω_n specifications.

Additional controller comparisons can be made through system closed loop frequency and time response data generated with each controller in place. Figures 4.3 through 4.6 display the singular value frequency responses for the open loop system, the closed loop system with full order controller, and closed loop system with the fifth and fourth order controllers, respectively. The controllers increase the closed loop DC gain, but if there is a problem it can be ameliorated by applying attenuation to the system inputs. Therefore, DC gains are not considered important in this discussion.

All of the controllers increase the damping of the open loop modes appearing at approximately 80 radians per second, but in some respects the reduced order controllers might be considered to exhibit behavior which is preferable to that of the full order controller. For example, in the 1 to 50 radian per second frequency regime, the reduced order controller responses are smoother than that of the full order controller; also, at high frequencies the minimum singular value response rolls off with

Figure 4.3: Open Loop Singular Value Frequency Response

Figure 4.4: Singular Value Response with Full Order Controller

Figure 4.5: Singular Value Response with Fifth Order Controller

Figure 4.6: Singular Value Response with Fourth Order Controller

the reduced order controllers, but not with the full order controller. Another observation is that closed loop bandwidth with the reduced order controllers is slightly greater than that with the full order controller. A drawback to the reduced order controllers is that they do not provide high frequency rolloff of the maximum singular value as the full order controller does. However, the gain is well below 0 dB in those frequency ranges, so this should not present a significant problem in actual implementation. Although the closed loop response of the reduced order controllers differs somewhat from that of the full order controller, it is apparent that if any DC gain problems are corrected, the reduced order controllers should yield acceptable closed loop performance. Other than differences in DC gains and slight differences in high frequency behavior, the responses of the fourth and fifth order controllers are very similar.

Closed loop step responses also display interesting behavior. Figures 4.7 through 4.10 display the normalized Y1 output response to a unit step applied to the X1 input for the open loop system and the closed loop systems with each controller in

place. Clearly, all controllers exhibit reduced settling time over the open loop system, but the reduced order controllers are better than the full order controller when percentage overshoot is the criterion for comparison. Although the fourth order controller exhibits larger amplitude oscillations than the fifth order controller, it is still less oscillatory in nature than the full order controller.

Figure 4.7: Normalized Open Loop Plant Output 1 Step Response

Figure 4.8: Normalized Output 1 Step Response with Full Order Controller

Figure 4.9: Normalized Output 1 Step Response with Fifth Order Controller

Figure 4.10: Normalized Output 1 Step Response with Fourth Order Controller

While it might be argued that these are not "good" controllers in some respects, it should be pointed out that they were designed with pole placement specifications as the only design criteria. Moreover, the poles were essentially chosen at random from the allowed region of Figure 4.2. It stands to reason that a more sophisticated design approach might yield better controllers. However, the emphasis here is that the design procedures yield reduced order controllers without great difficulty, and these controllers stabilize the closed loop IMC system.

5 Reduced Order H^∞ Controllers

Chapter 4 showed that stabilizing controllers of order equal to the minimum of (v_o, v_c) can be designed for any minimal plant realization through the use of pole placement techniques. A similar approach can be used to satisfy some of the necessary conditions for the existence of reduced order controllers which achieve closed loop infinity norm specifications when the matrices of the controller state-space realization satisfy certain rank conditions. These rank conditions are frequently encountered when the plant for which the H^∞ controller is designed has an unequal number of inputs and outputs. Such a situation may arise if a design plant naturally has an unequal number of inputs and outputs or the designer has the freedom to add linearly independent outputs to the system. The controller reduction method, while not guaranteed to yield reduced order H^∞ controllers, may serve as a tool to ascertain the closed loop infinity norm performance achievable with a reduced order controller and allow evaluation of the tradeoff between adding outputs to a system and controller order reduction.

5.1 Reducing the Order of H^∞ Controllers

The infinity norm of a linear system $G(s)$ is defined by $\|G\|_\infty = \sup_{\omega} \bar{\sigma}(G(j\omega))$, i.e. the infinity norm of $G(s)$ is its maximum frequency dependent singular value. Since

$$\underline{\sigma}(j\omega) \leq \frac{\|G(j\omega)u(j\omega)\|_2}{\|u(j\omega)\|_2} \leq \bar{\sigma}(j\omega) \quad , \quad (5.1)$$

the system gain defined according to

$$Gain = \frac{\|y(j\omega)\|_2}{\|u(j\omega)\|_2} = \frac{\|G(j\omega)u(j\omega)\|_2}{\|u(j\omega)\|_2} \quad (5.2)$$

is sandwiched between the maximum and minimum frequency dependent singular values of the plant. Thus, the infinity norm provides a bound on the maximum possible system gain (Maciejowski, 1988, p.79). The goal of H^∞ controller design is to construct a controller which causes the closed loop system to be internally stable and exhibit desirable infinity norm properties. The standard problem in H^∞ design is presented in (Francis, 1987); it is illustrated in the following figure.

Figure 5.1: Setup of the Standard H^∞ Controller Design Problem

The design plant $G(s)$ is partitioned as

$$G(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \quad (5.3)$$

representing the equations

$$\begin{aligned} z(s) &= G_{11}(s)w(s) + G_{12}(s)u(s) \\ y(s) &= G_{21}(s)w(s) + G_{22}(s)u(s) \quad . \\ u(s) &= K(s)y(s) \end{aligned} \tag{5.4}$$

A state-space realization for G is given by

$$G = \begin{bmatrix} [A] & [B_1 & B_2] \\ [C_1] & [D_{11} & D_{12}] \\ [C_2] & [D_{21} & D_{22}] \end{bmatrix} . \tag{5.5}$$

It will be assumed that all designs are performed on continuous time systems. If a design is to be undertaken on a discrete time system, it may be transformed into the w -plane by a bilinear transformation; then the design can be performed. The dependence on s will be suppressed throughout the remainder of this chapter for notational convenience, except where needed for clarity.

When a controller is connected from y to u as in Figure 5.1, the goal of the standard H^∞ problem is to produce an internally stabilizing controller K which minimizes the infinity norm of the transfer function T_{zw} defined by

$$z = T_{zw}w = [G_{11} + G_{12}K(I - G_{22})^{-1}G_{21}]w \quad . \tag{5.6}$$

A controller which is internally stabilizing and minimizes the infinity norm of T_{zw} is said to be H^∞ optimal.

A closely related problem is to find all stabilizing controllers K such that $\|T_{zw}\|_\infty < \gamma$, for some real valued γ greater than zero (Glover and Doyle, 1988). If γ is greater than the minimum achievable value γ_{opt} , such controllers exist, and the

set of all internally stabilizing controllers that satisfy $\|T_{zw}\|_\infty < \gamma$ is parameterized in terms of certain stable realizations $\Phi = (\hat{A}, \hat{B}, \hat{C}, \hat{D})$. Assume that the four block controller K has been obtained from the algorithm of Glover and Doyle (1988) with a state-space realization given by

$$K = \begin{bmatrix} \begin{bmatrix} A_k & B_{k1} & B_{k2} \end{bmatrix} \\ \begin{bmatrix} C_{k1} \\ C_{k2} \end{bmatrix} \end{bmatrix} \begin{bmatrix} D_{k11} & D_{k12} \\ D_{k21} & D_{k22} \end{bmatrix} . \quad (5.7)$$

Futhermore, assume that K is internally stabilizing and satisfies $\|T_{zw}\|_\infty < \gamma$ with $\Phi = 0$. Then, with any stable realization $\Phi = (\hat{A}, \hat{B}, \hat{C}, \hat{D})$ having infinity norm such that

$$\|\Phi\|_\infty = \|\hat{C}(sI - \hat{A})^{-1}\hat{B} + \hat{D}\|_\infty < \gamma , \quad (5.8)$$

the controller shown in Figure 5.2 is internally stabilizing and $\|T_{zw}\|_\infty < \gamma$ (Glover and Mustafa, 1991), (Glover and Doyle, 1988).

Figure 5.2: H^∞ Controller Parameterization

One drawback to H^∞ controllers is that the controller order is typically equal to the generalized plant order, and the controller parameterizations discussed above can yield even higher controller order. In cases where the plant order is very high, such controllers may not be practical. An example is the Hubble Space Telescope problem

mentioned previously where the plant was 58th order, but the highest controller order that could be accommodated by the flight computer was 42 (Irwin, et. al., 1993).

The parameterization of the H^∞ controllers can sometimes be used to reduce the controller order and still meet closed loop infinity norm requirements. Assume that K has been obtained from the algorithm of Glover and Doyle (1987) with a state-space realization given by Equation (5.8) and assume that $D_{k22} = 0$. This is a valid assumption for many H^∞ problems (Doyle, et. al., 1989). Also, suppose an order n_q realization for Φ is given by $\Phi = (\hat{A}, \hat{B}, \hat{C}, \hat{D})$. Connecting Φ to K as shown in Figure 5.2 results in the following closed loop controller realization:

$$K_{cl} = \begin{bmatrix} [A_{kcl}] & [B_{kcl}] \\ [C_{kcl}] & [D_{kcl}] \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} A_k + B_{k2} \hat{D} C_{k2} & B_{k2} \hat{C} \\ \hat{B} C_{k2} & \hat{A} \end{bmatrix} & \begin{bmatrix} B_{k1} + B_{k2} \hat{D} D_{k21} \\ \hat{B} D_{k21} \end{bmatrix} \\ \begin{bmatrix} C_{k1} + D_{k21} \hat{D} C_{k2} & D_{k12} \hat{C} \end{bmatrix} & \begin{bmatrix} D_{k11} + D_{k12} \hat{D} D_{k21} \end{bmatrix} \end{bmatrix}. \quad (5.9)$$

Assuming X is a matrix of dimension $n_q \times n$, the state transformation

$$T^{-1} = \begin{bmatrix} I & 0 \\ X & I \end{bmatrix}, \quad T = \begin{bmatrix} I & 0 \\ -X & I \end{bmatrix} \quad (5.10)$$

can be applied to K_{cl} with the result that the transformed realization for K_{cl} is

$$\begin{aligned} A_{kcl} &= \begin{bmatrix} A_k + B_{k2} \hat{D} C_{k2} - B_{k2} \hat{C} X & B_{k2} \hat{C} \\ X A_k + \hat{B} C_{k2} - \hat{A} X + X B_{k2} (\hat{D} C_{k2} - \hat{C} X) & \hat{A} + X B_{k2} \hat{C} \end{bmatrix}, \\ B_{kcl} &= \begin{bmatrix} B_{k1} + B_{k2} \hat{D} D_{k21} \\ X B_{k1} + X B_{k2} \hat{D} D_{k21} + \hat{B} D_{k21} \end{bmatrix}, \\ C_{kcl} &= \begin{bmatrix} C_{k1} + D_{k12} \hat{D} C_{k2} - D_{k12} \hat{C} X & D_{k12} \hat{C} \end{bmatrix}, \\ D_{kcl} &= \begin{bmatrix} D_{k11} + D_{k12} \hat{D} D_{k21} \end{bmatrix}. \end{aligned} \quad (5.11)$$

Suppose a square stable matrix $\hat{A}$ of dimension n_q and an $n_q \times n$ matrix X can be found along with matrices $\hat{B}$, $\hat{C}$, and $\hat{D}$ of appropriate dimension such that

$$C_{kl} = D_{kl2}\hat{C}X - D_{kl2}\hat{D}C_{k2} \quad (5.12)$$

and

$$\hat{A}X = X(A_k - B_{k2}C_{kl}) + \hat{B}C_{k2} \quad (5.13)$$

In this case, substitution of these two equations into Equation (5.11) reveals that

$$A_{kcl} = \begin{bmatrix} \hat{A} + B_{k2}\hat{D}C_{k2} - B_{k2}\hat{C}X & B_{k2}\hat{C} \\ 0 & \hat{A} + XB_{k2}\hat{C} \end{bmatrix} \quad (5.14)$$

and

$$C_{kcl} = \begin{bmatrix} 0 & D_{kl2}\hat{C} \end{bmatrix} \quad (5.15)$$

with the result that $n - n_q$ states of the controller realization are rendered unobservable.

If it is the case that $\Phi = (\hat{A}, \hat{B}, \hat{C}, \hat{D})$ satisfies $\|\Phi\|_\infty < \gamma$, the order n_q controller

$$\begin{aligned} A_r &= \hat{A} + XB_{k2}\hat{C} & B_r &= XB_{kl} + XB_{k2}\hat{D}D_{k2l} + \hat{B}D_{k2l} \\ C_r &= D_{kl2}\hat{C} & D_r &= D_{kl1} + D_{kl2}\hat{D}D_{k2l} \end{aligned} \quad (5.16)$$

guarantees that $\|T_{zw}\|_\infty < \gamma$.

These results are similar to those presented in Chapters 3 and 4 in that the solution to a Sylvester equation coupled with a standard linear equation render some of the controller states unobservable. A solution to these equations can be obtained using a method similar to that of Chapter 4. If $(A_k - B_{k2}C_{kl}, C_{k2})$ is a fully state observable realization, a state transformation can be applied to the controller realization so that the pair $(A_k - B_{k2}C_{kl}, C_{k2})$ is in OOCF with observability index v . While it

is not guaranteed that $(A_k - B_{k2}C_{k1}, C_{k2})$ is an observable realization, in all examples worked to date no unobservable realizations have been encountered. Furthermore, if the design plant has fewer inputs than outputs, some popular H^∞ problems have the property that the resulting controller matrix C_{k1} will have fewer rows than C_{k2} . Suppose C_{k1} has fewer rows or is lower in rank than C_{k2} . Letting r denote the rank of C_{k1} , the designer can then apply the algorithm of Chapters 3 and 4 to yield matrices $\hat{A}$, X , $\hat{B}$, $\hat{C}$, and $\hat{D} = 0$ which solve Equations (5.12) and (5.13), with the result that $n - (r \cdot v)$ states of the controller realization can be rendered unobservable. However, there is no assurance that the resulting $\Phi = (\hat{A}, \hat{B}, \hat{C}, \hat{D})$ satisfies $\|\Phi\|_\infty < \gamma$, so the existence of a reduced order controller satisfying the infinity norm requirements is not guaranteed. The designer has the freedom to specify $\hat{A}$ within the structural limitations presented in Chapter 4, so the infinity norm of Φ can be affected; however, it is unclear how to change $\hat{A}$ in order to reduce the infinity norm of Φ . In spite of these difficulties, successful results have been obtained for two different types of H^∞ design problems based on modified versions of the IMC plant model which was first presented in Chapter 4. The results of these examples are presented in the next two sections.

5.2 The Normalized H^∞ Control Problem

Figure 5.3: Block Diagram for the Normalized H^∞ Control Problem

A block diagram for the normalized H^∞ control problem is presented in Figure 5.3

The state equations for this system are given by

$$\begin{aligned}\dot{x} &= Ax + Bw_1 + Bu, \\ \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} &= \begin{bmatrix} Cx \\ u \end{bmatrix}, \\ y &= Cx + w_2.\end{aligned}\tag{5.17}$$

Letting $z = [z_1^T \ z_2^T]^T$ and $w = [w_1^T \ w_2^T]^T$, the closed loop transfer function T_{zw} is given by

$$T_{zw} = \begin{bmatrix} (I-GK)^{-1}G & (I-GK)^{-1}GK \\ K(I-GK)^{-1}G & K(I-GK)^{-1} \end{bmatrix}.\tag{5.18}$$

The four transfer functions of T_{zw} must each be stable in order to guarantee internal stability of a closed loop system (Maciejowski, 1988, p.p. 55-59). Furthermore, each of the four transfer functions of T_{zw} has a robustness interpretation. The transfer function $(I-GK)^{-1}G$ corresponds to additive uncertainty Δ_{11} on the controller K ; $(I-GK)^{-1}GK$ corresponds to output multiplicative uncertainty Δ_{12} on the plant G ; $K(I-GK)^{-1}G$ corresponds to input multiplicative uncertainty Δ_{21} on G ; and finally, $K(I-GK)^{-1}$ corresponds to additive uncertainty Δ_{22} on G . Block diagrams illustrating the four uncertainty types are presented in Figure 5.4. The upper bound on $\|T_{zw}\|_\infty$ indicates an upper bound on each of the four uncertainty types since

$$\left\| \begin{bmatrix} (I-GK)^{-1}G & (I-GK)^{-1}GK \\ K(I-GK)^{-1}G & K(I-GK)^{-1} \end{bmatrix} \right\|_\infty < \gamma\tag{5.19}$$

implies that

$$\begin{aligned}
& \| (I-GK)^{-1}G \|_{\infty} < \gamma \\
& \| (I-GK)^{-1}GK \|_{\infty} < \gamma \\
& \| K(I-GK)^{-1}G \|_{\infty} < \gamma \\
& \| K(I-GK)^{-1} \|_{\infty} < \gamma
\end{aligned} \tag{5.20}$$

The small gain theorem of (Zames, 1966) states that closed loop stability is maintained for any one stable Δ_{ij} satisfying

$$\|\Delta_{ij}\|_{\infty} < \gamma^{-1}, \quad i, j = 1, 2 \quad . \tag{5.21}$$

Thus, a lower value of γ implies greater closed loop tolerance of uncertainty. For the normalized H^{∞} problem, the goal of the H^{∞} optimal control problem is finding an internally stabilizing controller K such that the infinity norm of T_{zw} is minimized. The suboptimal H^{∞} control problem attempts to find an internally stabilizing controller K such that $\|T_{zw}\|_{\infty} < \gamma$, for some γ greater than the minimum achievable.

Figure 5.4: Uncertainty Types Covered by the Normalized H^{∞} Control Problem

A state-space realization of the standard plant or augmented plant P for the normalized H^∞ problem is given by

$$\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} [A] & [B_1 & B_2] \\ [C_1] & [D_{11} & D_{12}] \\ [C_2] & [D_{21} & D_{22}] \end{bmatrix} = \begin{bmatrix} [A] & [[B & 0] & [B]] \\ [C] & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & I \\ 0 & I & 0 \end{bmatrix} \end{bmatrix}, \quad (5.22)$$

where the plant $G = (A, B, C, D)$ is assumed to be minimal. For any γ greater than the minimal achievable value γ_{opt} , there exists a unique positive definite stabilizing solution W to the control Riccati equation

$$A^T W + W A - (1 - \gamma^{-2}) W B B^T W + C^T C = 0 \quad (5.23)$$

and there exists a unique positive definite solution Y to the filter algebraic Riccati equation

$$A Y + Y A^T - (1 - \gamma^{-2}) Y C^T C Y + B B^T = 0. \quad (5.24)$$

If the maximum eigenvalue of WY is less than γ^2 , a stabilizing controller which satisfies $\|T_{zw}\|_\infty < \gamma$ is given by the state-space realization (Mustafa and Doyle, 1991), (Doyle, et. al., 1989)

$$\begin{aligned} A_k &= A - (1 - \gamma^{-2}) Y C^T C - B B^T W Z, \\ [B_{k1} \ B_{k2}] &= [Y C^T \ B], \quad \begin{bmatrix} C_{k1} \\ C_{k2} \end{bmatrix} = \begin{bmatrix} -B^T W Z \\ -C Z \end{bmatrix}, \\ \begin{bmatrix} D_{k11} & D_{k12} \\ D_{k21} & D_{k22} \end{bmatrix} &= \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}, \end{aligned} \quad (5.25)$$

where $Z = (I - \gamma^{-2} Y W)^{-1}$.

Consider the equations which were derived in the previous section which, when solved, render some of the controller states unobservable. For this problem, Equation (5.12) becomes

$$-B^T WZ = \hat{C}X + \hat{D}CZ. \quad (5.26)$$

and Equation (5.13) becomes

$$\hat{A}X = X(A - (1 - \gamma^{-2})YC^T C) + \hat{B}CZ. \quad (5.27)$$

If the pair $([A - (1 - \gamma^{-2})YC^T C], CZ)$ is observable, a state transformation can be applied to the controller realization which puts $([A - (1 - \gamma^{-2})YC^T C], CZ)$ into OOCF. Although it is not certain that this can be done for all controllers obtained from normalized H^∞ design problems, in all examples worked to date the pair $([A - (1 - \gamma^{-2})YC^T C], CZ)$ has been observable.

Once the state transformation has been applied to the controller realization, a solution to Equations (5.26) and (5.27) can be obtained using the methods of Chapters 3 and 4. Let r denote the rank of $B^T WZ$, and let v denote the observability index of the pair $([A - (1 - \gamma^{-2})YC^T C], CZ)$. Assume without loss of generality that $B^T WZ$ is full row rank, and let $B^T WZ_{r_i}$ denote the i_{th} row of $B^T WZ$. Then choose stable matrices $\hat{A}_i$ of order v and apply the algorithm of Chapter 4 to determine matrices X_i , $\hat{B}_i$, $\hat{C}_i$, and $\hat{D}_i = 0$ of compatible dimensions which solve the equations

$$-B^T WZ_{r_i} = \hat{C}_i X_i \quad (5.28)$$

$$\hat{A}_i X_i = X_i(A - (1 - \gamma^{-2})YC^T C) + \hat{B}_i CZ$$

for the i_{th} row of $B^T WZ$, $i = 1, 2, \dots, r$. When the individual row solutions have been obtained, they can be combined as described in Section 3.4 to yield a solution to

Equations (5.26) and (5.27) which render $n-(r*v)$ states of the controller unobservable. If the overall solution $\Phi = (\hat{A}, \hat{B}, \hat{C}, \hat{D})$ satisfies $\|\Phi\|_{\infty} < \gamma$ and $r*v$ is less than n , a successful reduced order H^{∞} controller can be constructed according to Equation (5.16).

Clearly, the two main criteria in achieving a successful reduced order controller design are that $r*v < n$ and the γ condition is satisfied. If the design plant G has m inputs and l outputs with $m < l$ and v is approximately equal to $\frac{n}{l}$, then $B^T W Z$ has rank less than or equal to m and $r * v$ is less than n . This has been the case for all design examples attempted with plants having a greater number of outputs than inputs. If the γ condition is not satisfied, the problem must be repeated with different choices for the $\hat{A}_i$, a higher value of γ , or both. The use of an approach yielding nonzero matrices $\hat{D}_i$ has not been presented because in the author's experience such realizations have rarely satisfied the γ condition.

Another situation in which reduced order H^{∞} controller design may be possible is in the case where the plant has more inputs than outputs. The procedures detailed in this section and the previous section may be dualized in a manner similar to the dual Youla parameterization procedure of Chapter 3 to accommodate such a situation. A derivation of the dual procedure is straightforward and will not be presented.

5.3 A Reduced Order Normalized H^{∞} Controller Design Example

A reduced order normalized H^{∞} controller design was performed with a modified version of the IMC plant model of Chapter 4. The first modification combined the columns of the B matrix into a single column B matrix according to

$$B_{new} = [B_{c1} \ B_{c2}] \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (5.29)$$

to obtain a plant model which will result in a controller for which the rank of C_k is one, and the observability index of $([A-(1-\gamma^{-2})YC^TC], CZ)$ is less than n . In order to make the problem more interesting, the second modification was to alpha shift the A matrix to the right by 4.0 to destabilize it so the H^∞ controller must "work harder" to stabilize the closed loop system. The eigenvalues of the unstable modified plant are

$$\lambda = \begin{bmatrix} -8.752 \pm j113.29 \\ 1.739 \pm j94.56 \\ -51.46 \\ -48.66 \\ .5733 \pm 13.38 \\ -.1586 \pm j9.968 \end{bmatrix} . \quad (5.30)$$

The equations of the normalized H^∞ control problem were implemented in Matlab® for a SIMO plant; the m-file for the full order design is titled normhinf.m and the m-file for the reduced order design is titled rednhinf.m. The modified plant model is presented in Appendix A and the m-files are contained in Appendix B.

A tenth order controller was designed using the m-file normhinf.m of Appendix B; the minimum achievable γ was $\gamma = 26$. Closing the loop with this full order controller results in stability of each of the four transfer functions together which imply internal stability. Figures 5.5 through 5.10 display the singular value response of the open loop plant, the tenth order normalized H^∞ controller, and the four transfer functions previously discussed in this section.

Figure 5.5: Open Loop Plant Singular Value Response

Figure 5.6: Controller Singular Value Response

Figure 5.7: Singular Value Response of $(I-GK)^{-1}G$

Figure 5.8: Singular Value Response of $(I-GK)^{-1} GK$

Figure 5.9: Singular Value Response of $K(I-GK)^{-1} G$

Figure 5.10: Singular Value Response of $K(I-GK)^{-1}$

An internally stabilizing fifth order controller was designed using the m-file rednhinf.m of Appendix B. The lowest γ yielding successful controllers was $\gamma = 40$, but subsequent investigation revealed that increasing γ yields controllers with greater robustness to model uncertainty than those obtained with lower γ values, as evidenced by the responses of $(I-GK)^{-1}GK$, $K(I-GK)^{-1}G$, and $K(I-GK)^{-1}$. Figures 5.11 through 5.15 display the singular value responses of the controller generated with $\gamma = 1000$ and those of $(I-GK)^{-1}G$, $(I-GK)^{-1}GK$, $K(I-GK)^{-1}G$, and $K(I-GK)^{-1}$.

Figure 5.11: Fifth Order Controller Singular Value Response

Figure 5.12: Singular Value Response of $(I-GK)^{-1}G$

Figure 5.13: Singular Value Response of $(I-GK)^{-1}GK$

Figure 5.14: Singular Value Response of $K(I-GK)^{-1}G$

Figure 5.15: Singular Value Response of $K(I-GK)^{-1}$

Comparing the singular value plots from the two designs reveals that approximately 6 dB. of robustness to controller uncertainty was lost with the reduced order controller design as evidenced by the plots of $(I-GK)^{-1}G$, but the singular value responses of $(I-GK)^{-1}GK$, $K(I-GK)^{-1}G$, and $K(I-GK)^{-1}$ are lower in magnitude in the loop closed with the reduced order controller. Thus, closed loop robustness to the three types of plant uncertainty implicit in T_{zw} is greater with the reduced order controller than with the full order controller. Robustness to controller uncertainty seems to be of little importance since the designer oversees the controller implementation and should know the controller characteristics with a high degree of certainty. If this is truly the case, it appears the reduced order controller is more robust overall since it is more robust to plant uncertainty than the full order controller. Thus, it has been shown that performing a normalized H^∞ control problem with the modified IMC plant model by following the procedure outlined in Section 5.1 yields a reduced order normalized H^∞ controller with good stability properties.

5.4 The Mixed Sensitivity H^∞ Problem

Figure 5.16: Standard Plant for Mixed Sensitivity H^∞ Design

The mixed sensitivity approach to H^∞ controller design allows the designer to apply frequency dependent weighting functions to certain system outputs to specify closed loop performance and robustness requirements. A block diagram of the standard plant in the mixed sensitivity problem is shown in Figure 5.16. Straightforward manipulations based on the mixed sensitivity block diagram reveal that

$$\begin{aligned} e &= (I+GK)^{-1}w , \\ u &= K(I+GK)^{-1}w , \\ y_g &= GK(I+GK)^{-1}w . \end{aligned} \quad (5.31)$$

Defining the sensitivity function S and the complementary sensitivity function T according to $S = (I+GK)^{-1}$ and $T = GK(I+GK)^{-1} = I-S$, the preceding equations become

$$\begin{aligned} e &= Sw , \\ u &= KSw , \\ y_g &= Tw = (I-S)w . \end{aligned} \quad (5.32)$$

Now, T_{zw} in Figure 5.16 is given by

$$[T_{zw}] = \begin{bmatrix} W_1 S \\ W_2 KS \\ W_3 T \end{bmatrix} , \quad (5.33)$$

so finding a controller such that $\|T_{zw}\|_\infty < \gamma$ implies

$$\left\| \begin{bmatrix} W_1 S \\ W_2 KS \\ W_3 T \end{bmatrix} \right\|_\infty < \gamma . \quad (5.34)$$

If W_2 is chosen to be very small compared to W_1 and W_3 , the problem essentially simplifies to that of $\|T_{zw}\|_\infty < \gamma$ implying

$$\left\| \begin{bmatrix} W_1 S \\ W_3 T \end{bmatrix} \right\|_\infty < \gamma . \quad (5.35)$$

The ability to shape the singular value responses of S and T is useful because the singular value response of S and T provide measures of closed loop performance and robustness to model uncertainty. The fact that S provides a measure of closed loop performance is apparent when examining the block diagram of a typical feedback control system shown in Figure 5.17.

Figure 5.17: A Typical Feedback Control System

Clearly, in the figure $y = (I+GK)^{-1}d = Sd$, showing that the sensitivity function provides a measure of output disturbance attenuation; disturbances are attenuated where the singular value response of S is less than 1. The complementary sensitivity function is a measure of closed loop robustness to unstructured model uncertainty represented as multiplicative uncertainty at the plant output. In this type of uncertainty representation, the perturbed plant $G'(j\omega)$ is related to the nominal plant $G(j\omega)$ by

$$G'(j\omega) = (I+L(j\omega))G(j\omega) ; \quad (5.36)$$

the maximum singular value of the uncertainty $L(j\omega)$ is bounded by the positive scalar function $l_m(j\omega)$, where

$$\bar{\sigma}(L(j\omega)) < l_m(j\omega), \quad \forall \omega > 0. \quad (5.37)$$

According to (Doyle and Stein, 1981) the output multiplicative uncertainty representation is well suited to represent model uncertainty at high frequencies. The complementary sensitivity function is indicative of robustness to this type of uncertainty since the system will remain stable provided (Doyle and Stein, 1981)

$$\bar{\sigma}(GK(I+GK)^{-1}) = \bar{\sigma}(T) < \frac{1}{l_m(j\omega)}. \quad (5.38)$$

Thus, in the frequency ranges where $\bar{\sigma}(T)$ is small, the closed loop system is robust to output multiplicative model uncertainty.

Glover and Doyle (1988) present a method for designing H^∞ controllers for the standard problem based on the solution to two Riccati equations. If the mixed sensitivity problem is set up in an appropriate manner, the equations in that reference are greatly simplified. Using the block diagram of Figure 5.16, assume the plant D matrix is zero, and assume that the W_1 weighting function has zero D matrix as well. Furthermore, choose the W_2 weighting so that its D matrix is of the form $k*I$, where k is a scalar. The resulting transfer functions are

$$\begin{bmatrix} z_a \\ z_c \\ z_b \\ y \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} W_1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} -W_1 G \\ W_3 G \\ W_2 \end{bmatrix} \\ \begin{bmatrix} I \end{bmatrix} & \begin{bmatrix} -G \end{bmatrix} \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix}. \quad (5.39)$$

The ordering of z_a , z_b , and z_c yields a realization satisfying some of the requirements for the Glover-Doyle algorithm. Using the notation $G = (A_G, B_G, C_G, 0)$, a state-space realization of the standard plant or augmented plant is given by

$$\begin{bmatrix} [A] & [B_1 & B_2] \\ [C_1] & [D_{11} & D_{12}] \\ [C_2] & [D_{21} & D_{22}] \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} A_G & 0 & 0 & 0 \\ -B_{w_1}C_G & A_{w_1} & 0 & 0 \\ 0 & 0 & A_{w_2} & 0 \\ 0 & 0 & 0 & A_{w_3} \end{bmatrix} & \begin{bmatrix} 0 & B_G \\ B_{w_1} & 0 \\ 0 & B_{w_2} \\ 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & C_{w_1} & 0 & 0 \\ 0 & 0 & 0 & C_{w_3} \\ 0 & 0 & C_{w_2} & 0 \\ [-C_G & 0 & 0 & 0] \end{bmatrix} & \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ D_{w_2} \end{bmatrix} \\ [I] & [0] \end{bmatrix} \end{bmatrix} \quad (5.40)$$

Since $D_{w_2} = kI$, scaling the input u to yield $\tilde{u} = k*u$, the realization is given by Equation (5.41). It will be assumed that such a scaling has been performed; after the controller has been designed, the scaling should be reversed.

$$\begin{bmatrix} [A] & [B_1 & B_2] \\ [C_1] & [D_{11} & D_{12}] \\ [C_2] & [D_{21} & D_{22}] \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} A_G & 0 & 0 & 0 \\ -B_{w_1}C_G & A_{w_1} & 0 & 0 \\ 0 & 0 & A_{w_2} & 0 \\ 0 & 0 & 0 & A_{w_3} \end{bmatrix} & \begin{bmatrix} 0 & \frac{B_G}{k} \\ B_{w_1} & 0 \\ 0 & \frac{B_{w_2}}{k} \\ 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & C_{w_1} & 0 & 0 \\ 0 & 0 & 0 & C_{w_3} \\ 0 & 0 & C_{w_2} & 0 \\ [-C_G & 0 & 0 & 0] \end{bmatrix} & \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} \\ [I] & [0] \end{bmatrix} \end{bmatrix} \quad (5.41)$$

Also, assume w is a vector of m_1 inputs, u (or $\tilde{u}$) is a vector of m_2 inputs, z is a vector of p_1 outputs, and y is a vector of p_2 outputs. Since the plant G is assumed to be minimal and strictly proper, Equation (5.41) has the properties: (1) (A, B_2, C_2) is stabilizable and detectable; (2) the rank of $D_{12} = m_2$ and the rank of $D_{21} = p_2$; (3) $D_{12} = [0 \ I]^T$ and $D_{21} = [0 \ I]$; and (4) $D_{22} = 0$. These properties are four of the assumptions which must be satisfied in (Glover and Doyle, 1988). Two more assumptions which must be satisfied are that

$$\text{rank} \begin{pmatrix} A - j\omega * I & B_2 \\ C_1 & D_{12} \end{pmatrix} = n + m_2 \quad \forall \omega \in \mathbb{R} \quad (5.42)$$

and

$$\text{rank} \begin{pmatrix} A - j\omega * I & B_1 \\ C_2 & D_{21} \end{pmatrix} = n + p_2 \quad \forall \omega \in \mathbb{R} . \quad (5.43)$$

The rank conditions of Equations (5.42) and (5.43) are equivalent to the conditions that the realizations (A, B_2, C_1, D_{12}) and (A, B_1, C_2, D_{21}) have no transmission zeroes or pole-zero cancellations on the imaginary axis. These can be checked by plotting the singular value responses of the realizations (A, B_2, C_1, D_{12}) and (A, B_1, C_2, D_{21}) and verifying that none of the singular values approaches zero (Maciejowski, 1988, p.310). In the author's experience, careful selection of weighting functions usually alleviates any problems encountered in satisfying these requirements.

When the problem is set up as in Equation (5.41) and all of the assumptions are satisfied, the algorithm of (Glover and Doyle, 1988) becomes the following simplified algorithm. Define $D_{1.}$ and $D_{.1}$ according to

$$D_{1.} = \begin{bmatrix} D_{11} \\ D_{21} \end{bmatrix}, \quad D_{.1} = \begin{bmatrix} D_{11} & D_{12} \end{bmatrix}. \quad (5.44)$$

Next form R and $\tilde{R}$ according to

$$R = D_{1.}^T D_{1.} - \begin{bmatrix} \gamma^2 I_{m_1} & 0 \\ 0 & 0 \end{bmatrix}, \quad \tilde{R} = D_{.1} D_{.1}^T - \begin{bmatrix} \gamma^2 I_{p_1} & 0 \\ 0 & 0 \end{bmatrix}. \quad (5.45)$$

With the notation $B = [B_1 \ B_2]$ and $C = [C_1^T \ C_2^T]^T$, let W and Y be the solutions to the Riccati equations

$$\begin{aligned} & W[A - BR^{-1}D_{1.}^T C_1] + [A - BR^{-1}D_{1.}^T C_1]^T W \\ & - WBR^{-1}B^T W + C_1^T [I - D_{1.} R^{-1} D_{1.}^T] C_1 = 0 \end{aligned} \quad (5.46)$$

and

$$\begin{aligned} & Y[A - B_1 D_{.1}^T \tilde{R}^{-1} C]^T + [A - B_1 D_{.1}^T \tilde{R}^{-1} C] Y \\ & - YC^T \tilde{R}^{-1} C Y + B_1 [I - D_{.1}^T \tilde{R}^{-1} D_{.1}] B_1^T = 0. \end{aligned} \quad (5.47)$$

Form F and H according to

$$\begin{aligned} F &= -R^{-1}[D_{1.}^T C_1 + B^T W] = [F_{11} \ F_{12} \ F_2], \\ H &= -[B_1 D_{.1}^T + YC^T \tilde{R}^{-1}] = [H_{11} \ H_{12} \ H_2], \end{aligned} \quad (5.48)$$

where F_{11} has $m_1 - p_2$ rows, F_{12} has p_2 rows, F_2 has m_2 rows, H_{11} has $p_1 - m_2$ columns, H_{12} has m_2 columns, and H_2 has p_2 columns.

Now, if there exist positive definite solutions W and Y to the Riccati equations, and the maximum eigenvalue of (WY) is less than γ^2 , the set of all stabilizing controllers that satisfies $\|T_{zw}\|_\infty < \gamma$ is parameterized in terms of arbitrary stable Φ

satisfying $\|\Phi\|_\infty < \gamma$ according to the block diagram of Figure 5.2 (Glover and Doyle, 1988). Letting $Z = [I - \gamma^{-2}YW]^{-1}$, the controller matrices are given by

$$K = \begin{bmatrix} \begin{bmatrix} A_k & B_{k1} & B_{k2} \end{bmatrix} \\ \begin{bmatrix} C_{k1} \\ C_{k2} \end{bmatrix} \begin{bmatrix} D_{k11} & D_{k12} \\ D_{k21} & D_{k22} \end{bmatrix} \end{bmatrix}, \quad (5.49)$$

where $D_{k11} = 0$, $D_{k12} = I_{m_2}$, $D_{k21} = I_{p_2}$, $D_{k22} = 0$, and

$$\begin{aligned} B_{k2} &= B_2 + H_{12} \\ C_{k2} &= [C_2 + F_{12}]Z \\ B_{k1} &= -H_2 \\ C_{k1} &= F_2Z \\ A_k &= A + HC + B_{k2}C_{k1} \end{aligned} \quad (5.50)$$

As explained in Section 5.1, if the pair $(A_k - B_{k2}C_{k1}, C_{k2})$ is observable with observability index v and C_{k1} is of rank r such that $r \cdot v < n$, then a square stable matrix $\hat{A}$ of order $r \cdot v$ and an $r \cdot v \times n$ matrix X , along with matrices $\hat{B}$, $\hat{C}$, and $\hat{D}$ of compatible dimension can be found which render $n - (r \cdot v)$ of the controller states unobservable. If $\Phi = (\hat{A}, \hat{B}, \hat{C}, \hat{D})$ satisfies $\|\Phi\|_\infty < \gamma$, then a controller of order $r \cdot v$ can be constructed according to Equation (5.16) which guarantees $\|T_{zw}\|_\infty < \gamma$ for the closed loop system. For the mixed sensitivity problem setup described in this section, $A_k - B_{k2}C_{k1} = A + HC$ and $C_{k2} = [C_2 + F_{12}]Z$ so the pair $(A_k - B_{k2}C_{k1}, C_{k2})$ is $(A + HC, [C_2 + F_{12}]Z)$. If the design plant G has a number of outputs l greater than the number of inputs m , C_{k2} has maximum rank of l since $C_{k2} = [C_2 + F_{12}]Z$ and C_2 has l rows because $C_2 = [C_G \ 0 \ 0 \ 0]$. Also, $C_{k1} = F_2Z$ and F_2 has $m_2 = m$ rows, so if

C_{k2} is of full row rank l , then the rank of C_{kl} is less than that of C_{k2} . Section 5.1 showed that these conditions imply that it may be possible to design a reduced order controller that satisfies the H^∞ specifications; the procedure is applied to the mixed sensitivity problem in the following manner.

Assuming the pair $(A+HC, [C_2+F_{12}]Z)$ is observable, a state transformation is applied to the controller realization to put $(A+HC, [C_2+F_{12}]Z)$ in OOCF. Although observability of this pair is not guaranteed, no problems have been encountered in any example worked to date. It is assumed without loss of generality that C_{kl} is rank m and C_{k2} is rank l . Once the pair is in OOCF, matrices $\hat{A}_i$, $\hat{B}_i$, $\hat{C}_i$, $\hat{D}_i$, and X_i can be found using the method of Section 5.1 which satisfy the equations

$$\hat{A}_i X_i = X_i (A+HC) + \hat{B}_i [C_2+F_{12}]Z \quad (5.51)$$

and

$$C_{kl_i} = \hat{C}_i X_i + \hat{D}_i [C_2+F_{12}]Z, \quad (5.52)$$

where C_{kl_i} denotes the i_{th} row of C_{kl} , $i=1, 2, \dots, m$. If each of the realizations $\Phi_i = (\hat{A}_i, \hat{B}_i, \hat{C}_i, \hat{D}_i)$ satisfies $\|\Phi_i\|_\infty < \gamma$, these row solutions can be "stacked up" as presented in Section 3.4 to yield the stable matrix $\hat{A}$ and matrices X , $\hat{B}$, $\hat{C}$, $\hat{D}$, which can be combined according to Equation (5.16) to construct a controller of order $m \times v$ which meets the infinity norm constraint. If Φ_i cannot be found which satisfy $\|\Phi_i\|_\infty < \gamma$, the designer must try different $\hat{A}_i$ in an effort to reduce the infinity norm of Φ_i or repeat the entire design problem with a higher value of γ . It is difficult to predict how changes in $\hat{A}_i$ affect $\|\Phi_i\|_\infty$; this is an open area for further research. Clearly, this is not a "push of a button" process, but it has been applied successfully

to a design problem based on a modified version of the IMC plant model of Chapter 4. It should be noted that reduced order controllers may also be designed for many systems having a greater number of inputs than outputs. In this case, the procedure of this section and Section 5.1 can be dualized in a manner similar to that of the dual Youla parameterization approach shown in Chapter 3. The dualization is straightforward and will not be presented.

5.5 A Reduced Order Mixed Sensitivity H^∞ Controller Design Example

The design plant for the example problem is a tenth order SISO model taken as the input 1 to output 2 portion of a w-plane IMC model; Figure 5.18 displays the singular value response of this open loop plant.

Figure 5.18: Singular Value Response of SISO IMC Plant

The state-space matrices for the modified SISO IMC plant are contained in Appendix A. The design approach is to first design a full order controller based on the SISO plant model, then add a second output to the plant model which allows the design of a reduced order controller. It is assumed that the designer has the freedom to add

sensors to the plant to yield the second output. In this problem, the second output that was added is the output 1 that was part of the IMC plant prior to modification.

Figure 5.19: Closed Loop for IMC System

The block diagram used for this mixed sensitivity example is displayed in Figure 5.16. When the controller has been obtained, the loop is closed as shown in Figure 5.19, where d represents an output disturbance. It is clear from the block diagram that $y = (I+GK)^{-1}d = Sd$, showing that the sensitivity function gives an indication of the output response to the disturbance d . The weighting W_2 applied to the plant input in Figure 5.16 is $W_2 = .001$; this fulfills the rank requirements for the Glover-Doyle algorithm, yet does not place much weight on control energy or add order to the augmented plant. Since the goal of the example is an exploration of the level of performance and robustness to model uncertainty achievable with low order controllers, this is a reasonable choice. W_1 is chosen as a second order weighting specifying 40 dB. of attenuation to output disturbances at w-plane frequencies below .15 radians per second, and W_3 is chosen as a first order weighting specifying robustness to model uncertainty above 100 radians per second. The inverse weighting functions are displayed in Figure 5.20. A 13th order controller was designed which meets the specifications; results are shown in Figures 5.21 through 5.23.

Figure 5.20: W_1^{-1} and W_3^{-1} Weighting Functions

Figure 5.21: Singular Value Response of the Full Order Controller

Figure 5.22: Singular Value Response of S and $1/W_1$

Figure 5.23: Singular Value Response of T and $1/W_3$

In order to design the reduced order controller, a second output was added to the plant model as explained previously. The singular value response of the open loop plant remained unchanged. Performing a design with this 1 input 2 output plant results in a controller with 2 inputs and one output. In such a situation, GK has only one nonzero frequency dependent singular value since G and K each have only one nonzero frequency dependent singular value. In designing controllers for this SIMO plant, it was found that $\bar{\sigma}(S) = \bar{\sigma}(I+GK)^{-1} \geq 1$ at all frequencies, probably due to the rank deficiency in GK . However, if the designer is not concerned with the performance of the newly added output, a meaningful mixed sensitivity design can still be performed. Consider the following figure.

Figure 5.24: Plant With Added Output

It is clear that

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = (I+GK)^{-1} \begin{bmatrix} d \\ 0 \end{bmatrix} , \quad (5.53)$$

or

$$\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} (I+GK)^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} d , \quad (5.54)$$

implying

$$y_1 = \begin{bmatrix} 1 & 0 \end{bmatrix} (I+GK)^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} d . \quad (5.55)$$

Thus, in order to achieve attenuation in the response of y_1 to the input d , it is sufficient to design $\begin{bmatrix} 1 & 0 \end{bmatrix} (I+GK)^{-1}$ to obtain the desired attenuation. This may be accomplished by choosing the weighting $W_1 = [w_1 \ 0]$, where w_1 is a scalar weighting function; W_3 need not be changed. With these weightings and $\gamma=1$, assuming a successful controller is produced by the H^∞ algorithm, $\|W_1 S\|_\infty = \|w_1 \begin{bmatrix} 1 & 0 \end{bmatrix} S\|_\infty < 1$ implies that the designer can shape the singular value response of $\begin{bmatrix} 1 & 0 \end{bmatrix} S$.

A mixed sensitivity design was performed with $\gamma=1$ using the same weightings as those in the SISO design applied in the manner just described, yielding a successful 13th order controller. The behavior of this full order controller matched that of the SISO design exactly. The pair $(A+HC, [C_2+F_{12}]Z)$ was found to be observable, with observability index $\nu=7$; attempts were then made following the steps previously outlined in this section to obtain a 7th order $\Phi = (\hat{A}, \hat{B}, \hat{C}, \hat{D})$ that renders 6 of the controller states unobservable. After repeated attempts, a design with $\gamma=30$ yielded

a 7th order Φ that renders 6 of the controller states unobservable and satisfies $\|\Phi\|_{\infty} < 30$; a seventh order controller was constructed according to Equation (5.16). The following figures display the singular value responses of this reduced controller and that of the transfer function from d to y_1 given by $[1 \ 0]*S*[1 \ 0]^T$, along with the singular value response of T . The weighting functions displayed in the plots are those from the original SISO design with $\gamma=1$; they are included to aid in comparing the reduced order controller behavior to the original SISO design specifications.

Figure 5.25: Singular Value Response of Reduced Controller

Figure 5.26: Singular Value Response of $[1 \ 0] S [1 \ 0]^T$

Figure 5.27: Singular Value Response of T

The lightly damped mode in the controller response arises in the controller reduction process; the exact cause is unclear. The closed loop system meets the specification of 40 dB. of rejection of the disturbance d by the output y_1 below w-plane frequencies of .15 radians per second and robustness to model uncertainty above 100 radians per second. Although T crosses over the original W_3^{-1} weighting, this does not constitute a violation of the design constraints since γ is equal to 30 in the reduced order controller design. Comparing the results from the reduced order controller against those of the full order design reveals that both controllers meet the low frequency output disturbance attenuation specifications, but with the full order controller T begins to roll off at a lower frequency and with greater slope than the T obtained with the reduced controller. However, the reduced controller exhibits a lower peak in T than the full order controller.

Additional "brute force" robustness tests were also performed. The plant mode at approximately 96 radians per second was moved up and down 5 percent in frequency with the result that the closed loop remains stable with the full order controller, but

with the reduced controller, the loop remains stable when the mode is moved up in frequency, but is destabilized when the mode is moved to a lower frequency. Another test involved alpha shifting the design plant to the right in order to make it less stable, then closing the loop with each controller to test closed loop stability with the alpha shifted plant model. With an alpha shift of 1.8, the full order controller was destabilized, but the reduced order controller withstood an alpha shift of 3.8 without the closed loop becoming unstable. The alpha shift appears to model errors in the damping ratio of the mode at approximately 10 radians per second. This is apparent when comparing the singular value response of the plant with an alpha shift of 1.8 shown in Figure 5.28 to Figure 5.18, the singular value response of the unshifted plant. Clearly, one cannot say for certain which controller is most robust; the robustness determination depends on the type of robustness test being applied.

Figure 5.28: Singular Value Response of Alpha Shifted Plant

Both controllers appear to yield acceptable robustness and performance; it is up to the designer to decide between the two controllers based on the constraints of the problem at hand. The example has shown that adding an output to the design plant and

performing the steps detailed in this section can yield a reduced order mixed sensitivity H^∞ controller for the IMC system. Although such controllers are not guaranteed to exist, it has been shown that the reduced order controller design method of Chapters 3 and 4 can be used for problems more sophisticated than stabilization by pole placement techniques, and can be used as a tool to evaluate tradeoffs between plant complexity, controller complexity, and the performance and robustness achievable with reduced order controllers.

6 A Geometric Interpretation

Linear geometric control theory utilizes subspace manipulations for control systems analysis and design. Subspace properties are exploited to examine characteristics of the system plant $G(s)=(A, B, C, D)$ and to design controllers which satisfy certain closed loop performance requirements. Two examples of the types of subspaces which can be identified by these methods are the subspace of system controllable states and the subspace of unobservable states which are important in state feedback pole placement and observer design problems. Typical design goals might be to achieve closed loop pole placement or to design a compensator that achieves closed loop pole placement with disturbance decoupling. These problems and a variety of others can be approached by linear geometric methods. References (Basile and Marro, 1991) and (Wonham, 1979) provide in-depth presentations of linear geometric control theory and applications.

Schumacher (1980) develops a compensator synthesis procedure based on these methods and shows sufficient conditions for the existence of reduced order stabilizing controllers. Furthermore, Schumacher (1980) shows that a number of popular

controller design methods can be interpreted as special cases of the geometric design framework. A relationship exists between the reduced order controllers designed by the Youla parameterization method presented in previous chapters and the controllers designed via the linear geometric technique of (Schumacher, 1980). The Youla parameterization method has a direct interpretation as a special case of the linear geometric control method, and it can be shown that reduced order controllers designed via the two methods are equivalent. While this relationship may not prove to be a means to generate controllers of order lower than those previously obtained by either method, it does provide a link between the two approaches. The relationship between the two methods will be presented after a brief introduction into some fundamentals of linear geometric control theory.

6.1 An Introduction to Linear Geometric Control Theory

Since the foundation of linear geometric control theory is the manipulation of vector spaces, a brief introduction to vector space properties is in order. It is assumed that the reader is acquainted with the definition of a vector space; (Basile and Marro, 1991, p. 370) provides a detailed vector space definition. Letting $\bar{X}$ denote a vector space over the real numbers, a subset $\bar{V}$ of $\bar{X}$ is a subspace of $\bar{X}$ if any $v_1 \in \bar{V}$ and any $v_2 \in \bar{V}$ satisfy the property that $c_1 * v_1 + c_2 * v_2 \in \bar{V}$ for any real constants c_1 and c_2 . The notation for $\bar{V}$ being a subspace of $\bar{X}$ is $\bar{V} \subset \bar{X}$. A subspace of a vector space is itself a vector space.

A set of k vectors $\{ v_1, v_2, \dots, v_k \}$ belonging to a k dimensional subspace $\bar{V}$ is a basis of $\bar{V}$ if the vectors are linearly independent. Letting the basis vectors form the

columns of a matrix according to $V = [v_1 \ v_2 \ \dots \ v_k]$, it is apparent that $\bar{V}$ is equal to the image of V , denoted imV . Thus, the vector space $\bar{V}$ can be represented by the matrix V .

Supposing that $\bar{V} \subset \bar{X}$ and $\bar{W} \subset \bar{X}$, the subspace sum of $\bar{V}$ and $\bar{W}$ is defined as

$$\bar{V} + \bar{W} = \{ z : z = v + w, v \in \bar{V}, w \in \bar{W} \}. \quad (6.1)$$

The intersection of $\bar{V}$ with $\bar{W}$ is defined as

$$\bar{V} \cap \bar{W} = \{ z : z \in \bar{V} \text{ and } z \in \bar{W} \}. \quad (6.2)$$

The sum and intersection of subspaces of $\bar{X}$ are also subspaces of $\bar{X}$. Letting V and W be matrices such that $\bar{V} = imV$ and $\bar{W} = imW$, the subspace sum $\bar{V} + \bar{W}$ can be represented as $\bar{V} + \bar{W} = im[V \ W]$. If $\bar{V} \cap \bar{W} = 0$, the subspace sum of $\bar{V}$ and $\bar{W}$ is known as a direct sum and $\oplus$ is used instead of $+$ to denote the direct sum. In short, $\bar{V} \oplus \bar{W} = \bar{V} + \bar{W}$, where it is understood that $\bar{V} \cap \bar{W} = 0$.

There are several subspaces of the system state-space $\bar{X}$ which are utilized extensively in linear geometric control theory. The first to be discussed is the notion of an A -invariant subspace. Letting $\bar{X}$ represent the state-space, and letting A be a mapping $A: \bar{X} \rightarrow \bar{X}$, the subspace $\bar{V}$ is defined to be A -invariant if, for any $v \in \bar{V}$, $Av \in \bar{V}$. Thus, $\bar{V}$ is A -invariant if $A\bar{V} \subset \bar{V}$. If $\bar{V}$ is known to be A invariant, a change of coordinates can be introduced which generates a basis for $\bar{X}$ adapted to $\bar{V}$ in the following manner. Let V be matrix composed of linearly independent columns which span $\bar{V}$, and let W be matrix composed of columns which satisfy the property

that the columns of $[V \ W]$ span $\bar{X}$. Since $A\bar{V} \subset \bar{V}$ it is apparent that there exist matrices A_{11} , A_{12} , and A_{22} such that

$$A [V \ W] = [V \ W] \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} . \quad (6.3)$$

In this basis A_{11} is known as the map A restricted to $\bar{V}$, denoted $A|_{\bar{V}}$, and A_{22} is known as the map induced on the factor space $\bar{X}|\bar{V}$ by A , denoted $A|_{\bar{X}|\bar{V}}$. In this basis

$$\text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} = \bar{V} . \quad (6.4)$$

Other subspaces of interest include controlled and conditioned invariant subspaces. A controlled invariant subspace $\bar{V}$ contained in $\bar{X}$ satisfies the property

$$A\bar{V} \subset \bar{V} + \text{im} B . \quad (6.5)$$

Equivalently, there exists a matrix F (denoted a friend of $\bar{V}$) such that

$$(A + BF)\bar{V} \subset \bar{V} , \quad (6.6)$$

showing $\bar{V}$ is $(A + BF)$ invariant. Letting $\ker C$ denote the kernel of the matrix C , a conditioned invariant subspace satisfies the property that

$$A(\bar{V} \cap \ker C) \subset \bar{V} . \quad (6.7)$$

Equivalently, there exists a matrix G such that

$$(A + GC)\bar{V} \subset \bar{V} \quad (6.8)$$

showing that $\bar{V}$ is $(A + GC)$ invariant. Such a G will be termed a friend of $\bar{V}$ throughout the remainder of this document.

Controlled and conditioned invariant subspaces can play a significant role in systems analysis and design. For instance, there always exists a minimal A -invariant subspace containing $\text{im}B$ denoted $\langle A|B \rangle$ (Basile and Marro, 1991, p. 131). Since $\langle A|B \rangle$ is A -invariant, it is the case that $A\bar{V} \subset \bar{V} + \text{im}B$, so it is a controlled invariant subspace. If $\langle A|B \rangle = \bar{X}$, then $G(s)$ is fully state controllable and arbitrary closed loop eigenvalue placement may be accomplished. Also, if $\langle A|B \rangle \neq \bar{X}$, the system is stabilizable if and only if $\bar{X} \setminus \langle A|B \rangle$ contains only stable modes. There also exists a maximal A -invariant subspace $\bar{K}^*$ contained in $\ker C$ (Basile and Marro, 1991, p. 132). Since $\bar{K}^*$ is A -invariant and contained in $\ker C$, it is clear that $(A+GC)\bar{K}^* \subset \bar{K}^*$, showing that $\bar{K}^*$ is a conditioned invariant subspace. If $\bar{K}^* = \bar{0}$, $G(s)$ is fully observable. If $\bar{K}^* \neq \bar{0}$, the system is detectable if and only if $\bar{K}^*$ contains only stable modes. It will be shown in a subsequent section that reduced order stabilizing controllers may be constructed if a controlled invariant subspace and a conditioned invariant subspace can be found which satisfy certain properties.

6.2 Compensator Synthesis Using Controlled and Conditioned Invariant Subspaces

The following paragraphs summarize the compensator synthesis method of Schumacher (1980). Let $\bar{X}$, $\bar{U}$, and $\bar{Y}$ denote vector spaces of dimension n , m , and l respectively, and let A , B , and C be linear mappings such that $A: \bar{X} \rightarrow \bar{X}$, $B: \bar{U} \rightarrow \bar{X}$, and $C: \bar{X} \rightarrow \bar{Y}$. Suppose a system realization $G(s) = (A, B, C, 0)$ is given which satisfies

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \quad , \quad (6.9)$$

where $x(t) \in \bar{X}$, $y(t) \in \bar{Y}$, and $u(t) \in \bar{U}$. A compensator $K(s) = (N, M, L, K)$ satisfying

$$\begin{aligned}\dot{w}(t) &= Nw(t) + My(t) \\ u(t) &= Lw(t) + Ky(t)\end{aligned}\quad (6.10)$$

with $w(t) \in \bar{W}$ of dimension k equal to the dimension of $\bar{W}$ (denoted $\dim \bar{W}$) can be connected to this system. Combining the two sets of equations yields a new dynamic equation on the extended state-space $\bar{X} \oplus \bar{W}$:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{w}(t) \end{bmatrix} = \begin{bmatrix} A+BKC & BL \\ MC & N \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \end{bmatrix} = A_c \begin{bmatrix} x(t) \\ w(t) \end{bmatrix} . \quad (6.11)$$

The matrix A_c above will be referred to as the closed loop compensator matrix.

A (C, A, B) pair of subspaces of $\bar{X}$ is a pair of subspaces $(\bar{V}_1, \bar{V}_2)$ such that $\bar{V}_1$ is conditioned invariant, $\bar{V}_2$ is controlled invariant, and $\bar{V}_1 \subset \bar{V}_2$. Define

$$G(\bar{V}_1) = \{G: \bar{Y} \rightarrow \bar{X} \mid (A+GC)\bar{V}_1 \subset \bar{V}_1\} \quad (6.12)$$

and

$$F(\bar{V}_2) = \{F: \bar{X} \rightarrow \bar{U} \mid (A+BF)\bar{V}_2 \subset \bar{V}_2\} . \quad (6.13)$$

Finally, define

$$P(\bar{V}_1, \bar{V}_2) = \{(F, G) \in F(\bar{V}_2) \times G(\bar{V}_1) \mid \exists K: \bar{Y} \rightarrow \bar{U} \ni \bar{V}_1 \subset \ker(F-KC) \text{ and } \text{im}(G-BK) \subset \bar{V}_2\} . \quad (6.14)$$

For any (C, A, B) pair $(\bar{V}_1, \bar{V}_2)$ the set $P(\bar{V}_1, \bar{V}_2)$ is nonempty, and there exist matrices K , L , M , and N , along with an extension space $\bar{W}$ of dimension $\dim \bar{W} = \dim \bar{V}_2 - \dim \bar{V}_1$ such that: (1) there exists an A_c -invariant subspace $\bar{V}$ of $\bar{X} \oplus \bar{W}$ where A_c is the closed loop compensator matrix of Equation (6.11), (2) $\bar{V}_1$ is the intersection of $\bar{V}$ with $\bar{X}$, and (3) $\bar{V}_2$ is the projection of $\bar{V}$ into $\bar{X}$ (Schumacher, 1980). The matrices K , L , M , and N are constructed in the following manner: since $P(\bar{V}_1, \bar{V}_2)$ is nonempty, there exists a $K: \bar{Y} \rightarrow \bar{U}$ that satisfies the properties

$$\begin{aligned}\bar{V}_1 &\subset \ker(F-KC) \\ \text{im}(G-BK) &\subset \bar{V}_2\end{aligned}\quad (6.15)$$

Let R be a mapping $R: \bar{V}_2 \rightarrow \bar{W}$ such that $\ker R = \bar{V}_1$; also let $F_o = (F-KC)$ and $G_o = (G-BK)$. Since $\bar{V}_1 \subset \ker F_o$ there exists L satisfying $LR = F_o$, and because $\text{im}(G_o) \subset \bar{V}_2$ there exists M satisfying $M = -RG_o$. N can be obtained as a solution to the equation

$$NR = (A+GC+BF_o) \quad (6.16)$$

Although the equations yield the closed loop compensator matrix A_c of Equation (6.11) on the extended state-space $\bar{X} \oplus \bar{W}$, additional constraints must be met in order to guarantee that the closed loop system is stable. Theorem 4.4 of Schumacher (1980) states that a sufficient condition for the existence of stabilizing compensator of order k is the existence is a (C, A, B) pair $(\bar{V}_1, \bar{V}_2)$ with $\dim \bar{V}_2 - \dim \bar{V}_1 = k$ and a pair $(F, G) \in P(\bar{V}_1, \bar{V}_2)$ such that both $(A+BF)|_{\bar{V}_2}$ and $(A+GC)|_{\bar{X}|\bar{V}_1}$ are stable. Furthermore, the eigenvalues of the closed loop system are the eigenvalues of $(A+BF)|_{\bar{V}_2}$, along with those of $(A+GC)|_{\bar{X}|\bar{V}_1}$ (Schumacher, 1980). Thus, in the geometric control context, the problem of designing reduced order stabilizing controllers is a problem of finding a (C, A, B) pair $(\bar{V}_1, \bar{V}_2)$ that meets the requirements for closed loop stability and $\dim \bar{V}_2 - \dim \bar{V}_1 = k$ is a small number less than the plant order n .

The author is unaware of the existence of any tools which yield the largest conditioned invariant subspace contained within some other subspace, or the smallest controlled invariant subspace which contains some other subspace. The existence of such tools would aid in finding (C, A, B) pairs with $\dim \bar{V}_2 - \dim \bar{V}_1 = k$ a small

number, but the problem of finding (C, A, B) pairs having $(F, G) \in P(\bar{V}_1, \bar{V}_2)$ such that both $(A+BF) |_{\bar{V}_2}$ and $(A+GC) |_{\bar{X} | \bar{V}_1}$ are stable would still have to be addressed. In light of these facts, the general problem of designing reduced order stabilizing controllers using the geometric approach of Schumacher (1980) is largely an open issue. It will subsequently be shown, however, that the design method of Chapters 3 and 4 has a direct interpretation as a special case of the compensator synthesis procedure just described and also provides a tool for finding (C, A, B) pairs which satisfy the requirements for closed loop stability and yield compensators of order equal to the minimum of $v_o - 1$ and $v_c - 1$.

6.3 Connections Between the Geometric and Youla Parameterization Methods

Recalling Chapter 3, the equations which must be satisfied to construct a reduced order stabilizing controller using the Youla parameterization method are:

$$\begin{aligned} \hat{A}R &= R(A+HC) + \hat{B}C \\ F &= \hat{C}R - \hat{D}C \end{aligned} \quad , \quad (6.17)$$

where $\hat{A}$ must be stable, F is such that $(A+BF)$ is stable, and the notation has been switched from X to R in order to avoid confusion with the system state-space $\bar{X}$ in subsequent discussions. When an $n_q \times n$ matrix R exists which solves Equation (6.17) with some stable order n_q realization $Q(s) = (\hat{A}, \hat{B}, \hat{C}, \hat{D})$, a reduced order controller can be constructed according to

$$\begin{aligned} A_r &= \hat{A} + R\hat{B}\hat{C} & C_r &= \hat{C} \\ B_r &= \hat{B} + R\hat{H} + R\hat{B}\hat{D} & D_r &= \hat{D} \end{aligned} \quad . \quad (6.18)$$

Assuming a solution to Equation (6.17) exists where R is full rank and $(A+BF)$ and $\hat{A}$ are stable, it can be shown that: (1) $\ker R$ is a conditioned invariant subspace, (2) $\ker R$ is contained in $\ker(F+\hat{D}C)$, (3) there exists a matrix G and a basis for $\ker R$ such that in that basis $\hat{A} = (A+GC)|_{\bar{X}|\ker R}$, and (4) the matrices A_r , B_r , C_r , and D_r of Equation (6.18) are equivalent to the matrices N , $-M$, L , and $-K$, respectively, of the Schumacher method in the case where $\bar{V}_2 = \bar{X}$ and $\bar{V}_1 = \ker R$. Furthermore, given a (C, A, B) pair $(\bar{V}_1, \bar{V}_2)$ with $\bar{V}_2 = \bar{X}$ for which there exists an F and a G such that $(A+GC)|_{\bar{X}|\bar{V}_1}$ and $(A+BF)|_{\bar{X}}$ are stable, a solution of order $n - \dim \bar{V}_1$ to Equation (6.17) is implied. The statements are proved in the following theorems.

Theorem 6.1:

Assuming a full rank matrix R exists which solves Equation (6.17) with some $Q(s)=(\hat{A}, \hat{B}, \hat{C}, \hat{D})$, $\ker R$ is a conditioned invariant subspace.

Proof:

Since $Q(s) = (\hat{A}, \hat{B}, \hat{C}, \hat{D})$, the number of rows of $\hat{B}$ is equal to the number of rows of $\hat{A}$. $\hat{A}$ is square and multiplies R from the left in Equation (6.17), so the number of rows of $\hat{B}$ is equal to the number of rows of R . Because R is full rank, $\hat{B}$ lies in the column space of R , so there exists a matrix $\hat{H}$ such that $\hat{B} = R\hat{H}$. Then, from Equation (6.17)

$$\hat{A}R = R(A+HC) + \hat{B}C = R(A+(H+\hat{H})C) \quad (6.19)$$

Now, consider any vector $\tilde{r} \in \ker R$. Clearly,

$$R(A+(H+\hat{H})C)\bar{r} = \hat{A}R\bar{r} = 0 \quad (6.20)$$

implying that

$$(A+(H+\hat{H})C)\bar{r} \in \ker R \quad (6.21)$$

for any $\bar{r} \in \ker R$. Thus, $\ker R$ is a conditioned invariant subspace.

Theorem 6.2:

Assuming the conditions of Theorem 6.1, $\ker R \subset \ker(F+\hat{D}C)$.

Proof:

In Equation (6.17) $\hat{C}R = F + \hat{D}C$. Since there exists $\hat{C}$ which satisfies this equation, it is apparent that $\ker R \subset \ker(F+\hat{D}C)$.

Theorem 6.3:

Under the conditions of Theorem 6.1, there exists a matrix G and a basis for $\ker R$ such that in that basis, $\hat{A} = (A+GC) \big|_{\bar{x}|_{\ker R}}$.

Proof:

It was shown in Theorem 6.1 that there exists a matrix $\hat{H}$ such that

$$\hat{A}R = R(A+(H+\hat{H})C) . \quad (6.22)$$

Let P be a matrix comprised of a set of row vectors such that

$$\ker \begin{bmatrix} P \\ R \end{bmatrix} = 0 . \quad (6.23)$$

Denoting the number of rows of R as n_q , let the matrix $[S \ T]$ be such that

$$[S \ T] = \begin{bmatrix} P \\ R \end{bmatrix}^{-1}, \quad (6.24)$$

where the number of columns of S is $n - n_q$. Then

$$\begin{bmatrix} P \\ R \end{bmatrix} [S \ T] = I_n, \quad (6.25)$$

where I_n denotes an n by n identity matrix. Clearly,

$$\begin{bmatrix} P \\ R \end{bmatrix} S = \begin{bmatrix} I_{n-n_q} \\ 0 \end{bmatrix}, \quad (6.26)$$

illustrating that the columns of S form a basis for $\ker R$. It is obvious that there exist A_{11} and A_{12} such that

$$\begin{bmatrix} A_{11} & A_{12} \\ 0 & \hat{A} \end{bmatrix} \begin{bmatrix} P \\ R \end{bmatrix} = \begin{bmatrix} P \\ R \end{bmatrix} (A + (H + \hat{H})C), \quad (6.27)$$

implying

$$\begin{bmatrix} A_{11} & A_{12} \\ 0 & \hat{A} \end{bmatrix} = \begin{bmatrix} P \\ R \end{bmatrix} (A + (H + \hat{H})C) \begin{bmatrix} P \\ R \end{bmatrix}^{-1} = [S \ T]^{-1} (A + (H + \hat{H})C) [S \ T]. \quad (6.28)$$

Thus,

$$(A + (H + \hat{H})C) [S \ T] = [S \ T] \begin{bmatrix} A_{11} & A_{12} \\ 0 & \hat{A} \end{bmatrix}. \quad (6.29)$$

Since the columns of S are a basis for $\bar{V}_1$, it is clear that in this basis,

$\hat{A} = (A + (H + \hat{H})C)|_{\bar{X}|_{\ker R}}$. Letting $G = H + \hat{H}$, it is shown that there exists a

matrix G and a basis for $\ker R$ such that $\hat{A} = (A+GC) |_{\bar{x} | \ker R}$, completing the proof.

Theorem 6.4:

Assuming there exists a full rank matrix R which solves Equation (6.17) for some stable $Q(s)=(\hat{A}, \hat{B}, \hat{C}, \hat{D})$, and assuming F is such that $(A+BF)$ is stable, the controller matrices A_r , B_r , C_r , and D_r of Equation (6.18) are equivalent to the matrices N , $-M$, L , and $-K$, respectively, of the Schumacher method in the case where the (C, A, B) pair $(\bar{V}_1, \bar{V}_2)$ is given by $\bar{V}_2 = \bar{X}$ and $\bar{V}_1 = \ker R$. The minus signs occur because the Youla parameterization method assumes the controller is in a negative feedback configuration whereas a positive feedback path is assumed in the Schumacher derivation.

Proof:

Under the conditions assumed for Theorem 6.1, Theorems 6.1 through 6.3 have shown that $\bar{V}_1 = \ker R$ is a conditioned invariant subspace such that $\bar{V}_1 \subset \ker(F-\hat{D}C)$, and there exists $G = H+\hat{H}$ and a basis for $\ker R$ such that in that basis $\hat{A} = (A+GC) |_{\bar{x} | \bar{V}_1}$. $\hat{A}$ is stable by assumption, so the eigenvalues of $(A+GC) |_{\bar{x} | \bar{V}_1}$ are stable. Since $\bar{V}_2 = \bar{X}$, $\text{im}(G-BK) \subset \bar{V}_2$, and because F is such that $(A+BF)$ is stable, it is clear that $(A+BF) |_{\bar{V}_2}$ has stable eigenvalues. Thus, the (C, A, B) pair $(\bar{V}_1, \bar{V}_2)$ satisfies the sufficient conditions for the existence of a reduced order stabilizing controller.

By definition, R is a mapping from $\bar{X}$ onto the extension space $\bar{W}$ such that $\bar{V}_1 = \ker R$. The equation $F = \hat{C}R - \hat{D}C$ implies that $\hat{C}R = F + \hat{D}C$. Letting $K = -\hat{D} = -D_r$ and $L = \hat{C} = C_r$ satisfies the Schumacher (1980) equation $LR = F_o = F - KC$.

Defining $G_o = G - BK = G + B\hat{D}$, it is apparent that the equation $M = -RG_o$ must satisfy

$$M = -R(G + B\hat{D}) = -R(H + \hat{H}) - RB\hat{D} . \quad (6.30)$$

Since Theorem 6.1 shows that $\hat{B} = R\hat{H}$, the preceding equation becomes

$$M = -RH - \hat{B} - RB\hat{D} . \quad (6.31)$$

In Equation (6.18) the reduced order controller equation for B_r is $B_r = \hat{B} + RH + RB\hat{D}$, showing that $M = -B_r$.

Now consider the equation $A_r = \hat{A} + RB\hat{C}$. Clearly, this along with Equation (6.17) implies

$$A_r R = \hat{A}R + RB\hat{C}R = R(A + HC) + \hat{B}C + RB\hat{C}R . \quad (6.32)$$

In Theorems 6.1 through 6.3 it was shown that $\hat{B} = R\hat{H}$, $H + \hat{H} = G$, and $\hat{C}R = F - KC = F_o$, so the above equation becomes

$$A_r R = R(A + (H + \hat{H})C) + RB F_o = R(A + GC + BF_o) . \quad (6.33)$$

Clearly, letting $N = A_r$ satisfies the Schumacher (1980) equation

$$NR = R(A + GC + BF_o) . \quad (6.34)$$

This completes the proof that the Youla parameterization controller design method is equivalent to the Schumacher compensator design method using the (C, A, B) pair $(\bar{V}_1, \bar{V}_2)$ given by $\bar{V}_2 = \bar{X}$ and $\bar{V}_1 = \ker R$.

Theorem 6.5:

Assume there exists a (C, A, B) pair $(\bar{V}_1, \bar{V}_2)$ with $\bar{V}_2 = \bar{X}$ and matrices F, G and K such that $\bar{V}_1 \subset \ker(F-KC)$, $\text{im}(G-BK) \subset \bar{V}_2 = \bar{X}$, and $(A+BF)|_{\bar{X}}$ and $(A+GC)|_{\bar{X}|\bar{V}_1}$ are stable. Then there exist matrices $\hat{A}$, $\hat{B}$, $\hat{C}$, $\hat{D}$, R , and H that satisfy

$$\begin{aligned} \hat{A}R &= R(A+HC) + \hat{B}C \\ F &= \hat{C}R - \hat{D}C \end{aligned} \quad , \quad (6.35)$$

where $(A+HC)$ is stable, $\hat{A}$ is stable, and R is of dimension $n \times (n - \dim \bar{V}_1)$ and full rank.

Proof:

Let $\dim \bar{V}_1 = k$. Since $\bar{V}_1$ is a conditioned invariant subspace with $(A+GC)|_{\bar{X}|\bar{V}_1}$ stable, there exist matrices S and T such that the columns of S span $\bar{V}_1$ and $[S \ T]$ spans $\bar{X}$. Then

$$(A+GC) [S \ T] = [S \ T] \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \quad (6.36)$$

and A_{22} is stable. Let

$$\begin{bmatrix} P \\ R \end{bmatrix} = [S \ T]^{-1}, \quad (6.37)$$

where the number of rows of R equals $n-k$, the number of columns in T .

Since

$$\begin{bmatrix} P \\ R \end{bmatrix} S = \begin{bmatrix} I_k \\ 0 \end{bmatrix}, \quad (6.38)$$

it is clear that the columns of S span $\ker R$. Premultiplying and postmultiplying Equation (6.36) by $[S \ T]^{-1}$ yields

$$\begin{bmatrix} P \\ R \end{bmatrix} (A+GC) = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} P \\ R \end{bmatrix}, \quad (6.39)$$

so $A_{22}R = R(A+GC)$, and A_{22} is stable. Let H be a matrix such that $(A+HC)$ is stable. Then,

$$\begin{aligned} A_{22}R &= R(A+GC) = R(A+HC+GC-HC) \\ &= R(A+HC) + R(G-H)C \end{aligned} \quad (6.40)$$

Letting $\hat{A} = A_{22}$ and $\hat{B} = R(G-H)$, the equation becomes

$$\hat{A}R = R(A+HC) + \hat{B}C, \quad (6.41)$$

which is one of the equations that must be solved in the Youla parameterization design method.

Since $\bar{V}_1 \subset \ker(F-KC)$ and $\ker R = \bar{V}_1$, $\ker R \subset \ker(F-KC)$ implies that there exists a matrix $\hat{C}$ such that

$$\hat{C}R = F - KC. \quad (6.42)$$

Letting $\hat{D} = -K$ yields

$$F = \hat{C}R - \hat{D}C, \quad (6.43)$$

which is the second equation that must be solved in the Youla parameterization method. Since R is $k \times n$ and full rank, this implies that a stabilizing compensator of order $n-k$ can be fabricated using Equation (6.18).

Theorems 6.1 through 6.5 have shown the connections between the Youla parameterization design method and the linear geometric control compensator synthesis method of Schumacher (1980). A similar connection to the dual Youla parameterization approach can be made by deriving dual theorems to those of this section using arguments based on (C, A, B) pairs $(\bar{V}_1, \bar{V}_2)$ with $\bar{V}_1 = 0$, then relating the dual Youla parameterization equations to a controlled invariant subspace $\bar{V}_2$. Some contrived example problems will illustrate that judicious choices of (C, A, B) pairs of subspaces can lead to a low order solution to the Youla parameterization design Equation (6.17) and allow construction of a low order stabilizing controller.

6.4 Example Problems

Consider the system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + By(t) \\ y(t) &= Cx(t) \end{aligned} \quad , \quad (6.44)$$

where

$$A = \begin{bmatrix} -8 & 7 & -13 & 17 & -71.5 & -500 \\ -2 & -1 & 4 & 6 & 9 & 2 \\ 0 & -1 & .5 & 7 & 5 & 8 \\ 0 & 0 & 2 & -2 & 9 & 3 \\ 0 & 0 & 0 & -1 & -1 & 9 \\ 0 & 0 & 0 & 0 & -2 & -10 \end{bmatrix}, \quad B = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}^T. \quad (6.45)$$

A root locus plot of this system is given by Figure 6.1, and Figure 6.2 displays the complementary root locus plot. These plots show that constant output feedback cannot be used to stabilize this system. The realization is controllable and observable, but since it is a SISO realization the controllability index $\nu_c=6$ and the observability index $\nu_o=6$, so the methods of Chapter 3 will yield fifth order stabilizing controllers. It will be shown, however, that a (C, A, B) pair of subspaces can be found which will yield a first order stabilizing controller.

Figure 6.1: Root Locus

Figure 6.2: Complementary Root Locus

Letting $F = [0 \ 0 \ 0 \ 0 \ -101.5 \ -507]$, the eigenvalues of $(A+BF)$ are

$$\lambda(A+BF) = -.0429, -.0777, -7.97, -6.66, -7.6688 \pm j3.492 \quad (6.46)$$

Using $H = [15 \ 40 \ 17 \ 6 \ -2 \ .5]^T$, the eigenvalues of $(A+HC)$ are

$$\lambda(A+HC) = -.4530, -1.1281, -3.0406 \pm j3.5880, -7.6688 \pm j1.0825 \quad (6.47)$$

Now, letting $G = [0 \ 0 \ 0 \ 0 \ 0 \ 2]^T$,

$$(A+GC) = \begin{bmatrix} -8 & 7 & -13 & 17 & -71.5 & -500 \\ -2 & -1 & 4 & 6 & 9 & 2 \\ 0 & -1 & .5 & 7 & 5 & 8 \\ 0 & 0 & 2 & -2 & 9 & 3 \\ 0 & 0 & 0 & -1 & -1 & 9 \\ 0 & 0 & 0 & 0 & 0 & -8 \end{bmatrix} \quad (6.48)$$

Using $k = -101.5$, $[F-kC] = [0 \ 0 \ 0 \ 0 \ 0 \ -405.5]$. Then

$$\ker[F-kC] = \text{im} \begin{bmatrix} I_5 \\ 0 \end{bmatrix} \quad (6.49)$$

Denoting $\ker[F-kC]$ as $\bar{V}_1$, it is apparent that for the $(A+GC)$ of Equation (6.48), $(A+GC)\bar{V}_1 \subset \bar{V}_1$ and $-8 = (A+GC)|_{\bar{X}|\bar{V}_1}$. Thus, $\bar{V}_1$ is a conditioned invariant subspace such that $\dim \bar{V}_1 = 5$, $\bar{V}_1 \subset \ker[F-kC]$, and $(A+GC)|_{\bar{X}|\bar{V}_1}$ is stable. Then, by Theorem 6.5, there is a first order solution to the Youla parameterization equations (6.17). The solution will be constructed according to the theorem.

Construct $R: \bar{X} \rightarrow \bar{W}$ such that $\ker R = \bar{V}_1$. $R = [0 \ 0 \ 0 \ 0 \ 0 \ 1]$ is such a mapping. Also, let $\hat{A} = (A+GC)|_{\bar{X}|\bar{V}_1} = -8$. With the G and H previously defined, $(G-H) = [-15 \ -40 \ -17 \ -6 \ 2 \ 1.5]^T$. Letting $\hat{B} = R(G-H) = -1.5$, inserting these values into the equation

$$\hat{A}R = R(A+HC) + \hat{B}C \quad (6.50)$$

yields

$$\hat{A}R = -8*[0 \ 0 \ 0 \ 0 \ 0 \ 1] = [0 \ 0 \ 0 \ 0 \ 0 \ -8] , \quad (6.51)$$

and

$$\begin{aligned} R(A+HC) + \hat{B}C &= [0 \ 0 \ 0 \ 0 \ -1.5 \ -9.5] + 1.5*[0 \ 0 \ 0 \ 0 \ 1 \ 1] \\ &= [0 \ 0 \ 0 \ 0 \ 0 \ -8] \end{aligned} \quad (6.52)$$

showing that the equation is satisfied.

Letting $\hat{D} = -k = 101.5$, the equation $\hat{C}R = F + \hat{D}C$ implies

$$\begin{aligned} \hat{C}[0 \ 0 \ 0 \ 0 \ 0 \ 1] &= [0 \ 0 \ 0 \ 0 \ -101.5 \ -507] + 101.5*[0 \ 0 \ 0 \ 0 \ 1 \ 1] \\ &= [0 \ 0 \ 0 \ 0 \ 0 \ -405.5] \end{aligned} \quad (6.53)$$

showing that $\hat{C} = -405.5$ is the solution.

The reduced order controller is constructed as

$$\begin{aligned} A_r &= \hat{A} + R\hat{B}\hat{C} = -8 & C_r &= \hat{C} = -405.5 \\ B_r &= \hat{B} + RH + R\hat{B}\hat{D} = 2 & D_r &= \hat{D} = 101.5 \end{aligned} \quad (6.54)$$

Closing the loop with the reduced order controller yields closed loop eigenvalues of $-3.3735 \pm j3.3492$, -0.0429 , -0.0777 , -6.66 and -8 , which are the eigenvalues of $(A+BF)$ along with those of $\hat{A}$, as expected.

In a second example, a first order controller is designed which stabilizes a fifth order MIMO system. Consider the system in orthogonal observable canonical form

$$\begin{aligned} \dot{x}(t) &= Ax(t) + By(t) \\ y(t) &= Cx(t) \end{aligned} \quad (6.55)$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \quad (6.56)$$

The eigenvalues of the A matrix are $\lambda(A) = 1.0, .3090 \pm j.9511, -.8090 \pm j.5878$. This realization is fully observable with $v_o = 3$ and fully controllable with $v_c = 5$. The Youla parameterization method of Chapter 3 can be used to construct a second order controller that achieves arbitrary closed loop eigenvalue placement, but the purpose of the example is to illustrate that a (C, A, B) pair with appropriate stability properties can be found which will yield a first order stabilizing controller.

Using the matrix

$$F = \begin{bmatrix} -.2064 & -.1936 & -.8564 & -.1438 & .1389 \\ -.2064 & -.1936 & -.8564 & -.1438 & .1389 \end{bmatrix}, \quad (6.57)$$

the eigenvalues of $(A+BF)$ are

$$\lambda(A+BF) = -.0456 \pm j.0618, -.4238, -.2675 \pm j.0915. \quad (6.58)$$

Also, with

$$H = \begin{bmatrix} -.6653 & -.3343 & .0299 \\ -.3137 & -1.0234 & -.8024 \\ -.9324 & -.7975 & -1.5474 \\ -.9454 & -.4957 & -1.5479 \\ -1.8170 & -.7607 & -.8708 \end{bmatrix}, \quad (6.59)$$

the eigenvalues of $(A+HC)$ are

$$\lambda(A+HC) = -1.0, -.8090 \pm j.5878, -.3090 \pm j.9511. \quad (6.60)$$

Now, letting $R = [.1596 \quad -.1541 \quad .1489 \quad -.1438 \quad .1389]$ and $\hat{A} = [-1.0353]$,

$$\hat{A}R = [-.1652 \quad .1596 \quad -.1541 \quad .1489 \quad -.1438] \quad . \quad (6.61)$$

Using

$$G = \begin{bmatrix} -.19056 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (6.62)$$

it is easily verified that

$$R(A+GC) = [-.1652 \quad .1596 \quad -.1541 \quad .1489 \quad -.1438] \quad . \quad (6.63)$$

Thus, $\hat{A}R = R(A+GC)$, showing that $\ker R$ is a conditioned invariant subspace of dimension four.

Letting

$$K = \begin{bmatrix} -.3660 & -.0395 & -1.0053 \\ -.3660 & -.0395 & -1.0053 \end{bmatrix}, \quad (6.64)$$

the containment $\ker R \subset \ker(F-KC)$ is shown by

$$\begin{aligned} \ker R &= \ker[.1596 \quad -.1541 \quad .1489 \quad -.1438 \quad .1389] \subset \\ \ker \begin{bmatrix} .1596 & -.1541 & .1489 & -.1438 & .1389 \\ .1596 & -.1541 & .1489 & -.1438 & .1389 \end{bmatrix} &= \ker(F-KC) \quad . \end{aligned} \quad (6.65)$$

Letting $\bar{V}_1 = \ker R$ and $\bar{V}_2 = \bar{X}$, $\bar{V}_1$ is a conditioned invariant subspace contained in the controlled invariant subspace $\bar{V}_2$ with the following properties: (1) there exists a matrix F such that $(A+BF)|_{\bar{V}_2}$ is stable, (2) there exists a matrix G such that $(A+GC)|_{\bar{X}|\bar{V}_1}$ is stable, and there exists a matrix K such that $\bar{V}_1 \subset \ker(F-KC)$ and

$im(G-BK) \subset \bar{V}_2$. Thus, $\bar{V}_1$ and $\bar{V}_2$ constitute a (C, A, B) pair which satisfies all of the necessary conditions in (Schumacher, 1980) for the existence of a reduced order stabilizing compensator. The compensator can be constructed using the results of Theorem 6.5 and Equation (6.18). Letting $\hat{A} = (A+GC) |_{\bar{x} | \bar{v}_1} = -1.0353$ and letting $\hat{B} = R(G-H)$, the equation $\hat{A}R = R(A+GC)$ becomes

$$\hat{A}R = R(A+GC) = R(A+HC) + R(G-H)C = R(A+HC) + \hat{B}C \quad . \quad (6.66)$$

Also, letting $\hat{D} = -K$ and $\hat{C} = [1 \ 1]^T$,

$$\hat{C}R = \begin{bmatrix} .1596 & -.1541 & .1489 & -.1438 & .1389 \\ .1596 & -.1541 & .1489 & -.1438 & -.1389 \end{bmatrix} = F + \hat{D}C \quad , \quad (6.67)$$

so $F = \hat{C}R - \hat{D}C$. Since equations

$$\hat{A}R = R(A+HC) + \hat{B}C \quad (6.68)$$

and

$$F = \hat{C}R - \hat{D}C \quad (6.69)$$

have been satisfied, the reduced order controller is constructed according to

$$\begin{aligned} A_r &= \hat{A} + R\hat{B}\hat{C} = -1.0405 \\ B_r &= \hat{B} + RH + R\hat{B}\hat{D} = [-.3060 \quad -.0002 \quad -.0052] \\ C_r &= \hat{C} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad . \quad (6.70) \\ D_r &= \hat{D} = \begin{bmatrix} .3660 & .0395 & 1.0053 \\ .3660 & .0395 & 1.0053 \end{bmatrix} \end{aligned}$$

Closing the loop with the reduced order compensator results in closed loop eigenvalues at $-.4488$, $-.2724 \pm j.1323$, $-.0285 \pm j.0811$, and -1.0348 , which are not exactly the expected eigenvalues of $(A+BF)$ along with those of $\hat{A}$. This is due to the fact that all calculations in this example were rounded to four places to the right of the decimal point so that an interested reader may precisely duplicate the results presented herein. The importance of the example is that a first order stabilizing controller was obtained for a fifth order plant by judiciously choosing a (C, A, B) pair of subspaces having appropriate stability properties.

6.5 Conclusion

This chapter has shown the equivalence of the Youla parameterization controller design method and the compensator synthesis method of Schumacher (1980). Although this connection does not indicate the existence of controllers of lower order than were previously attainable through either method, it illustrates that tools developed within one method are directly applicable to the other, and allows the reduced order controller problem to be approached from either framework.

7 Conclusions and Suggestions for Further Research

The work presented in this dissertation has made several contributions to reduced order controller design. The work of (Gu, 1992) is based on the Youla parameterization of all stabilizing controllers and shows that if one can solve a Sylvester equation coupled with a standard linear equation, reduced order stabilizing controllers can be designed for a minimal, strictly proper design plant. However, no method is presented which can always yield such a reduced order stabilizing controller. The approach in (Gu, 1992) is one of solving a Sylvester equation, then searching for a stabilizing state feedback F in the row space of the Sylvester equation solution matrix. Since it is difficult to characterize stabilizing state feedback matrices, such a search procedure is essentially a trial and error process. Furthermore, the upper bound presented in the reference on the size of the achievable reduced order controllers is $n-m-l+1$ for an n state, m input, l output plant realization; this is not a significant reduction when the plant order is large.

Chapters 3 and 4 show that the reduced order controller design problem can be approached in a different manner to yield stabilizing controllers of order equal to the

minimum of $v_o - 1$ and $v_c - 1$, where v_o and v_c are the observability and controllability indices of the plant realization, respectively. Since these indices are often lower than the plant order for MIMO design plants, this presents a lower upper bound than that presented in (Gu, 1992) for many design problems. Chapter 4 presents a numerically stable and computationally efficient algorithm to calculate the controllers which does not require calculation of the controllability or observability test matrices of the design plant, nor are any matrix inversions required in the solution process. Since the algorithm uses a "divide and conquer" technique, the number of computational operations required to calculate the reduced order controllers is far less than the number of operations required by a direct approach. While the size of the controllers achieved through these method is equal to that of (Brasch and Pearson, 1970), the methods of Brasch and Pearson require the calculation of the controllability or observability test matrix of the design plant and the construction of a state transformation which must be applied to the design plant in order to obtain the controller. Since these test matrix calculations and state transformation inversions can be highly susceptible to computational errors, the method of Chapter 4 is a superior algorithm. Also, the new algorithm presents the designer with a fairly straightforward means of calculating reduced order stabilizing controllers; this may prove to be useful in conjunction with numerical optimization techniques that require the designer to supply an initial stabilizing controller which the numerical technique then attempts to optimize (Frazier, 1993).

It was shown in Chapter 4 that the designer must specify the matrix $\hat{A}$ as part of the design process and that the choice of $\hat{A}$ determines some of the closed loop eigenvalues. It may be the case, however, that the choice of $\hat{A}$ can be used to satisfy other closed loop design criteria, or perhaps some iterative technique may be applied to the elements of $\hat{A}$ in order to optimize the closed loop in some manner; these are areas into which future research should be directed. Furthermore, reference (Mohamed and Busch-Vishniac, 1995) uses the Youla parameterization to satisfy certain closed loop design requirements; since the reduced order controller design method is based on the Youla parameterization, it may be possible to satisfy the closed loop requirements with a reduced order controller. This is an area in which future research may prove productive. Another area of potential research is the application of nonlinear equation solution techniques to the set of nonlinear equations that result from Equations (4.1) and (4.2) if the upper right portion of $\hat{A}$ is allowed to consist of unknown elements rather than zeroes; perhaps such an investigation will yield tools which allow the design of stabilizing controllers of lower order than those obtained by the method that has been presented.

Chapter 5 presents the results of applying the algorithm of Chapter 4 to two types of H^∞ controller design problems. The two criteria which must be met in order to construct a reduced order controller that satisfies the infinity norm specifications are (1) a stable realization $\Phi = (\hat{A}, \hat{B}, \hat{C}, \hat{D})$ of order lower than the augmented plant order must be obtained which renders some of the controller states unobservable, and (2) $\|\Phi\|_\infty < \gamma$, where γ is the value specifying $\|T_{zw}\|_\infty < \gamma$ for the controller design

problem. As shown in Chapter 5, the algorithm of Chapter 4 can be applied to the full order H^∞ controller to yield Φ of low order which renders some of the controller states unobservable for some H^∞ design problems, but is difficult to ensure that $\|\Phi\|_\infty < \gamma$. Since $\hat{A}$ is specified during the controller reduction process, the designer can *affect* the infinity norm of Φ but it is unclear how to determine changes in $\hat{A}$ which will *reduce* $\|\Phi\|_\infty$. Clearly, this is another area ripe for further research.

Chapter 6 presents a linear geometric control interpretation of the Youla parameterization controller design method; it is shown that the Youla parameterization method is equivalent to the compensator synthesis method of (Schumacher, 1980). While the connection between the two methods does not imply the existence of controllers of lower order than those previously attained by either method, the connection shows that advances within either approach have a direct interpretation within the other framework. Since linear geometric control theory is a general mathematical treatment of control systems analysis and design, it may be possible to obtain additional controller reduction by geometric means, or obtain reduced order controllers that satisfy closed loop design requirements beyond that of stabilization. Additional research into reduced order controller design by geometric methods is suggested.

References

- Anderson, B.D.O., and Liu, Yi (1989), "Controller Reduction: Concepts and Approaches," *IEEE Transactions on Automatic Control*, vol. AC-34, no. 8, pp. 802-812
- Basile, G., and Marro, G. (1991), *Controlled and Conditioned Invariants in Linear System Theory*, Prentice Hall, Englewood Cliffs, New Jersey
- Bhattacharyya, S.P., and Keel, L.H. (1990), "State-Space Design of Low Order Stabilizers," *IEEE Transactions on Automatic Control*, vol. AC-35, no. 2, pp. 182-186
- Brasch, F.M., and Pearson, J.B. (1970), "Pole Placement Using Dynamic Compensators," *IEEE Transactions on Automatic Control*, vol. AC-15, no. 1, pp. 34-43
- Choi, B.W., Gu, D.W., and Postlethwaite, I. (1992), "Low Order Stabilizing Controllers and Pole Assignment," *Proceedings of the 1992 American Controls Conference*, Chicago, IL, USA, pp. 798-799
- Doyle, J.C., Glover, K., Khargonekar, P.P., and Francis, B.A. (1989), "State-Space Solutions to Standard H_2 and H^∞ Control Problems," *IEEE Transactions on Automatic Control*, vol. AC-34, no. 8, pp. 228-238
- Doyle, J.C. and Stein, G. (1981), "Multivariable Feedback Design: Concepts for a Classical/Modern Synthesis," *IEEE Transactions on Automatic Control*, vol. AC-26, no. 1, pp. 4-16
- Francis, B. A. (1987), *A Course in H^∞ Control Theory*, Lecture Notes in Control and Information Sciences, vol. 88, Springer-Verlag, Berlin
- Frazier, W.G. (1993), "Search Based Methods for Computer-Aided Controller Design Improvement," Ph.D. Dissertation, Ohio University, Athens, Ohio
- Glover, K. (1984), "All Optimal Hankel Norm Approximations of Linear Multivariable Systems and their L^∞ Error Bounds," *International Journal of Control*, Vol. 39, no. 6, pp. 1115-1193
- Glover, K. and Doyle, J.C. (1988), "State Space Formulae for all Stabilizing Controllers that Satisfy an H^∞ -Norm Bound and Relations to Risk Sensitivity," *Systems and Control Letters*, 11, pp. 167-172

- Gu, D.W., Choi, B.W., and Postlethwaite, I. (1993), "Low Order Stabilizing Controllers," *IEEE Transactions on Automatic Control*, vol. AC-38, no. 11, pp. 1713-1717
- Irwin, R.D., Glenn, R.D., Frazier, W.G., Lawrence, D.L., and Follett, R.F. (1993), "Controller Redesign for the Hubble Space Telescope," NASA Report NAG8-242
- Kailath, Thomas (1980), *Linear Systems*, Prentice-Hall, Englewood Cliffs, New Jersey
- Kimura, H. (1975), "Pole Assignment by Output Feedback," *IEEE Transactions on Automatic Control*, vol. AC-20, no. 4, pp. 509-516
- Kucéra, V. (1974), "The Matrix Equation $AX + XB = C$," *SIAM Journal of Control and Optimization*, vol. 26, no. 1, pp. 15-25
- Luenberger, D.G. (1966), "Observers for Multivariable Systems," *IEEE Transactions on Automatic Control*, vol. AC-11, no. 2, pp. 190-197
- Maciejowski, J.M. (1989), *Multivariable Feedback Design*, Addison-Wesley, Reading, Massachusetts.
- Medina, Enrique A., Irwin, R. Dennis, Mitchell, Jerrel R., and Bukley, Angelia P. (1994), "MIMO System Identification Using Frequency Response Data," *The Journal of the Astronautical Sciences*, Vol. 42, No. 1, pp. 113-129
- Mohamed, A.M., and Busch-Vishniac, I. (1995), "Imbalance Compensation and Automation Balancing in Magnetic Bearing Systems Using the Q-Parameterization Theory," *IEEE Transactions on Control Systems Technology*, vol. 3, no. 2, pp. 202-211
- Moore, B.C. (1981), "Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction," *IEEE Transactions on Automatic Control*, vol. AC-26, no. 1, pp. 17-32
- Murdoch, P. (1973), "Observer Design for a Linear Functional of the State Vector," *IEEE Transactions on Automatic Control*, vol. AC-18, no. 3, pp 308-310
- Mustafa, D. and Glover, K (1991), "Controller Reduction by H^∞ Balanced Truncation," *IEEE Transactions on Automatic Control*, vol. AC-36, no. 6, pp. 668-682

- Nett, C.N., Jacobsen, C.A. and Balas, M.J. (1984), "A Connection Between State-Space and Doubly Coprime Fractional Representations," *IEEE Transactions on Automatic Control*, vol. AC-29, no. 9, pp. 831-832
- Ogata, K. (1990), *Modern Control Engineering*, Prentice Hall, Englewood Cliffs, New Jersey
- Petkov, P.Hr., Christov, N.D., and Konstantinov, M.M. (1991), *Computational Methods for Linear Control Systems*, Prentice Hall International (UK) Ltd, Hertfordshire, UK
- Phillips, C.L. and Harbor, R.D. (1991), *Feedback Control Systems*, Prentice Hall, Englewood Cliffs, New Jersey
- Schumacher, J.M. (1980), "Compensator Synthesis Using (C,A,B) -Pairs," *IEEE Transactions on Automatic Control*, vol. AC-25, no. 6, pp. 1133-1138
- Tsui, C. (1987), "A Complete Analytical Solution to the Equation $TA-FT=LC$ and Its Applications," *IEEE Transactions on Automatic Control*, vol. AC-32, no. 8, pp. 742-744
- Wonham, W.M. (1979), *Linear Multivariable Control: A Geometric Approach*, Springer-Verlag, New York, New York
- Zames, G. (1966), "On the Input-Output Stability of Time Varying Nonlinear Feedback Systems - Part I," *IEEE Transactions on Automatic Control*, vol. AC-34, no. 8, pp. 228-238

Appendix A Plant Models for Design Examples

A.1 Pole Placement Design Examples

The design plant for the pole placement examples is a z-plane model of the IMC plant. The state-space realization is given by

$$A = \begin{bmatrix} 0.0937 & -0.4370 & 0.6671 & -0.1944 & 0.2469 & -0.1347 & 0.1157 & -0.0333 & 0.0013 & -0.0056 \\ 0.7021 & -0.2682 & -0.5245 & -0.2272 & -0.5619 & 0.1161 & -0.0900 & -0.0013 & -0.0454 & -0.0090 \\ -1.0292 & 0.1645 & 0.4290 & -0.4695 & -0.8356 & 0.1271 & -0.4552 & 0.0193 & 0.0365 & 0.0254 \\ 0.5235 & 0.2159 & -0.0602 & 1.0433 & -0.1888 & 0.2386 & -0.4618 & -0.0899 & -0.0691 & -0.0254 \\ 0.1255 & 0.9358 & 0.0744 & -0.5047 & 1.2978 & -0.6321 & 1.0224 & -0.1038 & 0.0782 & -0.0138 \\ 0.3009 & 2.0061 & 2.5145 & -1.3928 & 0.3769 & 0.2642 & 0.1362 & 0.0363 & 0.1580 & 0.0442 \\ 0.0944 & 0.9297 & 1.0937 & -0.2990 & 0.0687 & -0.0379 & 0.6185 & -0.0132 & 0.1094 & -0.0027 \\ 0.2082 & 3.9447 & 3.1167 & -2.5707 & 1.0704 & -2.3190 & 2.2671 & -0.1243 & 0.4006 & 0.1866 \\ 1.6399 & 6.4156 & 9.4476 & -1.0864 & 4.8004 & -1.9314 & 0.3695 & -0.2976 & 0.5837 & 0.3772 \\ 1.6953 & 6.0192 & 9.0817 & 1.3739 & 7.5273 & -1.7748 & 2.8622 & -0.7366 & 0.2919 & 0.1394 \end{bmatrix},$$

$$C = \begin{bmatrix} -0.0249 & -0.0133 & -0.0172 & 0.0140 & 0.0009 & -0.0236 & -0.0189 & 0.0066 & 0.0051 & -0.0018 \\ -0.0053 & 0.0035 & -0.0019 & -0.0129 & 0.0157 & 0.0147 & -0.0298 & -0.0071 & 0.0014 & -0.0000 \end{bmatrix},$$

$$B = \begin{bmatrix} -0.2656 & 0.2499 \\ 0.3585 & 0.4060 \\ 0.2432 & -1.2832 \\ 0.5228 & 0.9460 \\ -1.7149 & 0.7820 \\ -1.1242 & -2.1262 \\ -1.9101 & 0.3120 \\ -0.9458 & -5.9657 \\ -10.7943 & -15.1243 \\ -5.9245 & -45.6395 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

A.2 The Normalized H^∞ Control Problem

The plant used for the full order and reduced order normalized H^∞ controller design example is a modified version of the w-plane IMC plant model. This modified model is unstable. State-space matrices are

$$A = \begin{bmatrix} 11.857 & -49.613 & 57.361 & -2.768 & 27.563 & -6.497 & 6.871 & -2.650 & -3.297 & -0.055 \\ 55.104 & -5.676 & -45.996 & -42.448 & -50.214 & -7.662 & 6.306 & -6.680 & 0.497 & 0.122 \\ -85.255 & 35.137 & 15.999 & -43.967 & -24.943 & -10.851 & -5.266 & -6.340 & 5.438 & 0.187 \\ 30.881 & 61.410 & 31.939 & 22.143 & 23.303 & 6.490 & -18.391 & -6.434 & -1.397 & -0.400 \\ -2.793 & 85.086 & 40.258 & -16.214 & 49.623 & -37.553 & 56.125 & -5.277 & 4.765 & -0.385 \\ 70.948 & 104.252 & 140.677 & 16.290 & 73.635 & -2.718 & -14.036 & 15.376 & 4.602 & -0.648 \\ 14.337 & 38.333 & 53.074 & 21.351 & 26.376 & 8.259 & -13.728 & 2.218 & 6.838 & -3.249 \\ 39.644 & 498.865 & 356.095 & -138.058 & 198.391 & -193.934 & 188.038 & -79.183 & 39.140 & 20.526 \\ 379.706 & 268.715 & 607.632 & 277.089 & 429.964 & 39.285 & -158.759 & 58.915 & -53.750 & 42.460 \\ 509.574 & 56.718 & 714.876 & 596.120 & 820.714 & 54.490 & 85.939 & 17.267 & -15.157 & -61.008 \end{bmatrix},$$

$$C = \begin{bmatrix} -1.394 & -2.200 & -0.504 & 0.567 & -0.233 & -0.844 & -1.231 & 0.586 & 0.395 & -0.359 \\ -0.636 & 1.204 & 0.629 & -0.859 & 0.972 & 0.516 & -1.426 & -0.773 & 0.268 & -0.003 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.0914 \\ 0.1056 \\ -0.1533 \\ 0.1907 \\ -0.1535 \\ -0.6470 \\ -1.1688 \\ 8.5442 \\ -11.9138 \\ -78.6926 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

A.3 The Mixed Sensitivity H^∞ Design Problem

The SISO model used for the full order mixed sensitivity design consists of the input 1 to output 2 response of a w-plane version of the IMC plant. A realization is given by

$$A = \begin{bmatrix} 7.857 & -49.613 & 57.361 & -2.768 & 27.563 & -6.497 & 6.871 & -2.650 & -3.297 & -0.055 \\ 55.104 & -9.676 & -45.996 & -42.448 & -50.214 & -7.662 & 6.306 & -6.680 & 0.497 & 0.122 \\ -85.255 & 35.137 & 11.999 & -43.967 & -24.943 & -10.851 & -5.266 & -6.340 & 5.438 & 0.187 \\ 30.881 & 61.410 & 31.939 & 18.143 & 23.303 & 6.490 & -18.391 & -6.434 & -1.397 & -0.400 \\ -2.793 & 85.086 & 40.258 & -16.214 & 45.623 & -37.553 & 56.125 & -5.277 & 4.765 & -0.385 \\ 70.948 & 104.252 & 140.677 & 16.290 & 73.635 & -6.718 & -14.036 & 15.376 & 4.602 & -0.648 \\ 14.337 & 38.333 & 53.074 & 21.351 & 26.376 & 8.259 & -17.728 & 2.218 & 6.838 & -3.249 \\ 39.644 & 498.865 & 356.095 & -138.058 & 198.391 & -193.934 & 188.038 & -83.183 & 39.140 & 20.526 \\ 379.706 & 268.715 & 607.632 & 277.089 & 429.964 & 39.285 & -158.759 & 58.915 & -57.750 & 42.460 \\ 509.574 & 56.718 & 714.876 & 596.120 & 820.714 & 54.490 & 85.939 & 17.267 & -15.157 & -65.008 \end{bmatrix},$$

$$B = [0.1367 \ 0.0614 \ -0.1325 \ 0.0927 \ 0.0131 \ -0.4332 \ 0.3081 \ 1.8491 \ 0.1393 \ -80.0684]^T,$$

$$C = [-1.394 \ -2.200 \ -0.504 \ 0.567 \ -0.233 \ -0.844 \ -1.231 \ 0.586 \ 0.395 \ -0.359],$$

$$D = [0] .$$

The plant used for the reduced order design is a SIMO version of the IMC plant created by adding a second output to the SISO model. The A and B matrices are unchanged from the SISO model; the C and D matrices are given by

$$C = \begin{bmatrix} -1.394 & -2.200 & -0.504 & 0.567 & -0.233 & -0.844 & -1.231 & 0.586 & 0.395 & -0.359 \\ -0.636 & 1.204 & 0.629 & -0.859 & 0.972 & 0.516 & -1.426 & -0.773 & 0.268 & -0.003 \end{bmatrix},$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} .$$

Appendix B Matlab® M-file Listings

Listings of all the m-files which implement the algorithms and examples presented in this dissertation are contained in Appendix B. The comments within the individual files will instruct the interested reader on their use.

B.1 Orthogonal Canonical Forms

These m-files are used to place a state-space realization $G(s) = (A, B, C, D)$ into orthogonal observable or controllable canonical form. The algorithms which are implemented by these m-files are presented in (Petkov, et. al., 1991).

Orthccf.m places a realization $G(s) = (A, B, C, D)$ into orthogonal controllable canonical form (OCCF).

```
function [T,m,l,p,Ac,Bc,Cc,tol]=orthccf(A,B,C)

% function [T,m,l,p,Ac,Bc,Cc,tol]=orthccf(A,B,C)
%
% Implement the Orthogonal Transformation to Orthogonal Controllable Canonical
% Form (OCCF) via the staircase algorithm attributed to Van Dooren.
%
% Inputs: A, B, C - The A, B and C matrices of the system
%
% Outputs: T - The orthogonal state transformation to Controllable
%           Orthogonal Canonical Form
%
%           m - a 1 by n vector containing the Kronecker Indices of the
%           pair (A,C)
%
%           l - a scalar which equals the dimension of the controllable
%           part of the realization
%
%           p - a scalar equal to the controllability index of the system
```

```

%
%      Ac, Bc Cc - The A, B and C matrices after transformation.
%
%      tol      - tolerance used in rank determinations

% References: Petkov, P. Hr., Christov, N.D., and Konstantinov, M.M.,
%      Computational Methods for Linear Control Systems,
%      Prentice Hall, U.K., 1991, pp. 379-382
%
%      Maciejowsky, J.M., Multivariable Feedback Design,
%      Addison Wesley, 1989, pp.379-382.
%
% Russ Glenn 10-5-94

[rb,cb]=size(B);
[ra,ca]=size(A);

if ra ~= ca,
    disp('Warning, A matrix is not square')
    pause
else
    n=ra;
end

% Set up the tolerance to be used in rank determinations

afnorm=norm(A,'fro');
bfnorm=norm(B,'fro');
tol=(ra^2)*eps*max(afnorm,bfnorm);
disp(['tolerance is ',num2str(tol)])

% Initialize Variables

m=zeros(1,ra);      % Kronecker indices vector
l=0;                % sum of Kronecker indices
p=0;                % p is used as an index into the m vector and will equal
                    % the controllability index when algorithm is finished

W=B;                % W is "scratchpad memory" used in calculations
Bc=zeros(rb,cb);
T=eye(ra,ca) ;
nb=rb;              % nb and mb are indices into matrices during various
mb=cb;              % stages of this algorithm

```

```

% Begin Major Loop

% Processing stops when you have processed all of the controllable part of
% the realization

for k=1:n,

    % Get the QR factorization with column pivoting of the W Matrix

    [Q,R,E]= qr(W);

    % Determine Rank of R by looking at the diagonal entries

    rank=0;

    for j=1:min(nb,mb) ,

        if abs(R(j,j)) > tol ,
            rank=rank + 1 ;
        end

    end

    irnk=rank ;

    % If rank of W equal zero, you have finished processing the controllable
    % part of the system

    if irnk == 0,
        break
    end

    ni = 1 ; % ni keeps track of the previous sum of Kronecker Indices
    p=p+1 ;
    m(p) = irnk ; % Save the pth Kronecker Index
    l=l+irnk ; % Running sum of Kronecker Indices

    % Form the state transformation for the kth iteration of the process

    [qrows,qcols]=size(Q);

    if k == 1,
        Qk=Q;
    else

```



```

    Qk=[eye(n-nb),zeros(n-nb,qcols);zeros(qrows,n-nb),Q];
end

% Accumulate the transforms in T, Form the transformed A matrix for this
% iteration and write it over A.

T=T*Qk ;
A=Qk'*A*Qk ;

% If K=1, get Bc

if k == 1,
    Bc=Q'*W ;
end

% if irnk = nb, you have completed the process and are done.

if irnk==nb,
    break
end

% Extract the new W out of the transformed A matrix for the next iteration.

mb=irnk ;
nb=nb-irnk ;
W=A(l+1:l+nb,ni+1:ni+mb) ;
end

Ac=A ;
Cc=C*T ;

```

Orthocf.m places a realization $G(s) = (A, B, C, D)$ into orthogonal observable canonical form (OOCF).

```

function [T,m,l,p,Ao,Bo,Co,tol]=orthocf(A,B,C)

% function [T,m,l,p,Ao,Bo,Co,tol]=orthocf(A,B,C)
%
% Implement the Orthogonal Transformation to Orthogonal Observable Canonical
% Form (OOCF) via the staircase algorithm attributed to Van Dooren.
%
% Inputs: A, B, C - The A, B, and C matrices of the system

```

```

%
% Outputs: T - The orthogonal state transformation to Controllable
%           Orthogonal Canonical Form
%
%           m - a 1 by n vector containing the Kronecker Indices of the
%           pair (A,C)
%
%           l - a scalar which equals the dimension of the controllable
%           part of the realization
%
%           p - a scalar equal to the controllability index of the system
%
%           Ao, Bo, Co - The A, B, and C matrices after transformation.
%
%           tol - tolerance used in rank determinations

% References: Petkov, P. Hr., Christov, N.D., and Konstantinov, M.M.,
%           Computational Methods for Linear Control Systems,
%           Prentice Hall, U.K., 1991, pp. 379-382
%
%           Maciejowsky, J.M., Multivariable Feedback Design,
%           Addison Wesley, 1989, pp.379-382.
%
% Russ Glenn 1-7-95

% Just form the dual realization, put the dual realization into OCCF,
% Then 'undualize' the result

% Form Dual Realization

Adual=A';
Bdual=C';
Cdual=B';

% Put Dual realization into OCCF

[T,m,l,p,Adualc,Bdualc,Cdualc,tol]=orthccf(Adual,Bdual,Cdual);

% Now Undualize the results

Ao=Adualc';
Bo=Cdualc';
Co=Bdualc';
end

```

Makzeros.m takes the realization obtained as output of **orthocf.m** or **orthccf.m** and zeros out any matrix elements with absolute value lower than the tolerance value. The designer should compare the frequency response of the new orthogonal realization to that of the realization to which either **orthocf.m** or **orthccf.m** was applied. This will verify that no significant inaccuracies have been introduced by the coordinate change or the zeroing out of small magnitude matrix elements.

```
function Aout=makzeros(Ain,tol)

% function Aout=makzeros(Ain,tol)
%
% Scan the matrix Ain, and for any entry that absolute value below tol,
% set it equal to zero
%
% R. G. 10-6-94

[nr,nc]=size(Ain);
Aout=Ain ;
for i = 1:nr ,
    for j=1:nc ,
        if abs(Aout(i,j)) <= tol ,
            Aout(i,j) = 0;
        end
    end
end
```

B.2 Sylvester Equation Solution Algorithms

These files implement the Sylvester equation solution algorithms of Chapter 4.

Solvsylv.m implements the Sylvester equation solution method of Chapter 4, yielding Sylvester equation solutions with the matrix $\hat{A}$ of order ν , where ν is the observability index of $G = (A, B, C, D)$ in OOCF.

```
function [X,Bq]=solvsylv(Aq,AHCo,Co,f,m,nu)

% function [X,Bq]=solvsylv(Aq,AHCo,Co,f,m,nu)
%
% Solve the Sylvester equation  $Aq \cdot X = X \cdot [AHCo] + Bq \cdot Co$ , where the first
% row of the matrix X is specified by the designer and the components of
% Bq are unknowns. The solution is obtained by exploiting the structure of
% Aq, AHCo, and Co. SVD's are used to solve matrix equations. The resulting
% reduced order controller will be of order nu and strictly proper.
%
% Inputs:
% Aq is the Youla parameterization A matrix in a form with arbitrary diagonal
% elements, nonzero first superdiagonal elements, arbitrary first subdiagonal
% elements, and zeros elsewhere.
%
% AHCo, Co are the plant A+HC and C in Orthogonal Observable Canonical Form
%
% f is a 1 by n vector which generates desired pole placement
%
% m is a vector of the system Kronecker indices, nu is the observability index
%
% Outputs:
% X and Bq are the solution to the Sylvester equation
%
% Russ Glenn 6-23-95

% Orthogonal Observable Canonical Form of a realization Ao,Co is as follows:
%
%      +--
%      | A{1,1}  A{1,2}  0  . . . . .  0      |
%      |
```

```

%      | A{2,1}  A{2,2}  A{2,3}  0 . . . . .      . . . . . 0      |
%      |
%      | A{3,1}  A{3,2}  A{3,3}  A{3,4}  0 . .      . . . . . 0      |
%      |
%  Ao = | .              .              .              .              |
%      |
%      | .              .              .              .              |
%      |
%      | .              .              .              .              |
%      |
%      | A{nu-2,1} .      .      .      A{nu-2,nu-1}  0      |
%      |
%      | A{nu-1,1} .      .      .      .      A{nu-1,nu} |
%      |
%      | A{nu,1} .      .      .      .      A{nu,nu} |
%      +--
%
%      +--
%  Co = | C{1}  0      0      |
%      +--
%
%  Ao is n by n, and Co is l by n .
%
%  nu is observability index of the realization.
%
%  m=[m(1) m(2) . . . m(nu)] is a vector of the realization Kronecker indices.
%
%  and l >= m(1) >= m(2) >= . . . >= m(nu).
%
%  A{j,j} is of dimension m(j) by m(j)
%
%  A{j-1,j} is of dimension m(j-1) by m(j) and full column rank.
%
%  C(1) is l by m(1) and full column rank
%
%  Nothing can be said about the structure of Bo in this coordinate system.
%
%  The notation used in the subsequent comments is the notation from the
%  orthogonal observable canonical form presented above.
%
%  References on Orthogonal Observable Canonical Form:
%      Petkov, P. Hr., Christov, N.D., and Konstantinov, M.M.,
%      Computational Methods for Linear Control Systems,
%      Prentice Hall, U.K., 1991, pp. 379-382

```

```

%
%           Maciejowsky, J.M., Multivariable Feedback Design,
%           Addison Wesley, 1989, pp.379-382.

disp(' ')
% Check that Aq is in proper form.

[aqr,aqc]=size(Aq) ;
if aqr ~= aqc,
    disp('ERROR: Aq is not square')
    return
end

AA=Aq-diag(diag(Aq))-diag(diag(Aq,1),1)-diag(diag(Aq,-1),-1) ;
if max(max(abs(AA))) > 0
    disp('ERROR: Aq is not in proper form')
    return
end

% Form a vector of Kronecker Index sums.

msum=zeros(1,nu) ;
msum(1)=m(1) ;
for i= 2:nu ;
    msum(i)=m(i)+msum(i-1) ;
end

% Establish the size of the matrix AHCo

[n,n]=size(AHCo);

% Check that AHCo is in Orthogonal Observable Canonical Form .
% Check for zeros in the proper locations. Rank tests on the
%  $A_{j-1,j}^{\{j-1,j\}}$  blocks will be performed later in the program

for i=1:nu-1,
    if i==1,
        testzeros=AHCo(1:msum(1),msum(i+1)+1:n) ;
    else
        testzeros=AHCo(msum(i-1)+1:msum(i),msum(i+1)+1:n) ;
    end
end

```

```

if max(max(abs(testzeros))) > 0 ,
    disp('Error: AHCo is not in Orthogonal Observable Canonical Form!')
    return
end
end

% Check that Co is in Orthogonal Observable Canonical Form

[cor,coc]=size(Co);

testzeros=Co(:,m(1)+1:n) ;
if max(max(abs(testzeros))) > 0
    disp('Error: Co is not in Orthogonal Observable Canonical Form!')
    return
end

% Check that Co is full rank

rr=rank(Co) ;
if rr < min([cor,coc]),
    disp('Error: Co is not full rank!')
    return
end

% Initialize X. X is nu rows by n columns.

X=zeros(nu,n) ;
X(1,:)=f ;

% Calculate the Upper Right portion of X.

for i=2:nu , % Row index for X(i,j) Calculations
    for j= msum(i-1)+1:n , % Column index for X(i,j) Calculations

        if i-1 == 1,
            X(i,j)=(X(i-1,:)*AHCo(:,j)-Aq(i-1,i-1)*X(i-1,j))/Aq(i-1,i);
        else
            X(i,j)=(X(i-1,:)*AHCo(:,j)-Aq(i-1,i-1)*X(i-1,j)-...
                Aq(i-1,i-2)*X(i-2,j))/Aq(i-1,i);
        end
    end
end
end

```

```

end
% Now Work from lower Right to Upper Left to Calculate the Remaining
% Unknowns in X. Recall that all unknowns which have not yet been
% calculated are still initialized to zero at this point.

for j=nu:-1:2, % Index into Kronecker Index Sums vector

% create a variable to handle the case when j-2=0, since msum(j-2) is
% used in subsequent calculations, but msum(0) does not exist. When
% the index msum(j-2)+1 is used in matrix equation, it is desired that
% msum(j-2)+1=1 when j-2=0.

    if j-2 == 0,
        msumjminus2=0;
    else
        msumjminus2=msum(j-2);
    end

    for i=nu:-1:j, % Index of row in which calculations are occurring

% When working with the group of columns of AHCo comprised of
% the blocks  $A_{j,j}$  (notation borrowed from Orthogonal Observable
% Canonical Form), keep track of the matrix  $A_{j-1,j}$  in a variable called
% coltop, and the lower part of the columns in a variable called colbottom.
% Recall that everything above the  $A_{j-1,j}$  block is all zeroes.

        coltop=AHCo(msumjminus2+1:msum(j-1),msum(j-1)+1:msum(j)) ;
        colbottom=AHCo(msum(j-1)+1:n,msum(j-1)+1:msum(j)) ;

% Recalling that coltop=  $A_{j-1,j}$ , the equations to solve for the
% remaining groups of unknowns in X are of the form
%
%  $[X(i,msum(j-2)+1):msum(j-1)]*A_{j-1,j} =$ 
%  $Aq(i,:)*X(:,msum(j-1)+1:msum(j))$ 
%  $-[X(i,msum(j-1)+1:n)]*[A_{j,j} \dots A_{nu,j}]'$  Eq(1).
%
% where  $[X(i,msum(j-2)+1):msum(j-1)]$  are unknowns, and
% and  $[X(i,msum(j-1)+1:n)]$  have already been calculated.

% Let the right side of equation 1 be put into a variable called "right".

```



```

right=(Aq(i,:)*X(:,msum(j-1)+1:msum(j)))- ...
      X(i,msum(j-1)+1:n)*colbottom ;

% Then equation (1) is [X(i,msum(j-2)+1):msum(j-1)]*A{j-1,j} = right.
% Using SVD such that A{j-1,j}=UDV', we now have
% [X(i,msum(j-2)+1):msum(j-1)]*U*D= right*V. Solve for the components of
% [X(i,msum(j-2)+1):msum(j-1)]*U, then get [X(i,msum(j-2)+1):msum(j-1)] ,
% from [X(i,msum(j-2)+1):msum(j-1)]=[X(i,msum(j-2)+1):msum(j-1)]*U*U'.

% Find SVD of A{j-1,j}

[U,D,V]=svd(coltop);

% Note the matrices A{j-1,j} should be either square or tall
% and have full column rank in Orthogonal Observable Canonical Form.
% Check for these conditions.

[dr,dc]=size(D) ;
rr=rank(coltop) ;

if dc > dr,
    disp('ERROR: One of the A{j-1,j} is of incorrect dimension!')
    return
end

if rr < min([dr,dc]) ,
    disp('ERROR: One of the A{j-1,j} is not full column rank!')
    disp(' ')
    disp('Check that A matrix is in Orthogonal OCF. ')
    return
end

% Form a temporary variable XU= [[X(i,msum(j-2)+1):msum(j-1)]*U].

[xur,xuc]=size(X(i,msumjminus2+1:msum(j-1))*U) ;
XU=zeros(xur,xuc) ;

% The equation is now of the form XU*D=right*V. If there are more unknowns
% than equations, the "extra" unknowns will equal zero. Solve for the
% first "dc" unknowns of the group, let the rest be zero.

RSV=right*V;
for l=1:dc

```

```

        XU(l)=RSV(l)/D(l,l);
    end

% Now get the actual unknowns from the temporary vector XU.

    X(i,msumjminus2+1:msum(j-1))=XU*U' ;

end
end

% Solve for the unknowns in Bq. The necessary equation is of the form
% Bq*Co=Aq*X-X*AHCo. Use the SVD of Co such that the equation becomes
% Bq*U*D=[Aq*X-X*AHCo]*V. Solve for the elements of Bq*U, then extract
% Bq by the equation Bq=[Bq*U]*U'.

% Get the SVD of Co.

[U,D,V]=svd(Co) ;

% Set up a temporary variable called BqU.

bqur=nu;
[dr,dc]=size(D);
bquc=dr;
BqU=zeros(bqur,bquc);

% Now solve for the components of BqU.

right=(Aq*X-X*AHCo)*V ;

for j=1:dr ; % Column Index.

    BqU(:,j)=right(:,j)/D(j,j)    1148X
end

% Extract Bq from BqU.

Bq=BqU*U' ;

% display the maximum error in the solved Sylvester equation.

error=X*AHCo+Bq*Co-Aq*X ;

maxerr=max(max(abs(error))) ;

```

```

disp(' ')
disp('Sylvester equation solution is complete')
disp(' ')
disp(['Maximum Sylvester Equation Error is ',num2str(maxerr)]) ;

end

```

Redsylyv.m implements the Sylvester equation solution method of Chapter 4, yielding Sylvester equation solutions with the matrix $\hat{A}$ of order $\nu-1$, where ν is the observability index of $G = (A, B, C, D)$ in OOCF.

```

function [X,Bq]=redsylyv(Aq,AHCo,Co,f,m,nu)

% function [X,Bq]=redsylyv(Aq,AHCo,Co,f,m,nu)
%
% Solve the Sylvester equation  $Aq*X=X*[AHCo] + Bq*Co$ , where the last  $m(1)+1$ 
% elements of row 1 of X are specified by the designer and the components of
% Bq are unknowns. The solution is obtained by exploiting the structure of
% Aq, AHCo, and Co. SVD's are used to solve matrix equations. The resulting
% reduced order controller will be of order  $\nu-1$ .
%
% Inputs:
% Aq is the Youla parameterization A matrix in a form with arbitrary diagonal
% elements, nonzero first superdiagonal elements, arbitrary first subdiagonal
% elements, and zeros elsewhere.
%
% AHCo, Co are the plant  $A+HC$  and  $C$  in Orthogonal Observable Canonical Form
%
% f is a 1 by n vector which generates desired pole placement
%
% m is a vector of the system Kronecker indices, nu is the observability index
%
% Outputs:
% X and Bq are the solution to the Sylvester equation
%
% Russ Glenn 6-23-95

% It may be of interest to note that this file is identical to the m-file
% solvsylyv.m except for the size of X ( $\nu-1$  by n in this file vs. nu by n
% in solvsylyv.m) and in the index i in the for loops that calculate the

```

% the elements of X. Also Bq has nu-1 rows here vs. nu rows in solvsylv.m

% Orthogonal Observable Canonical Form of a realization Ao,Co is as follows:

```
%
%      +-
%      | A{1,1}  A{1,2}  0  . . . . . 0      |      +-
%      |
%      | A{2,1}  A{2,2}  A{2,3}  0 . . . . . 0      |
%      |
%      | A{3,1}  A{3,2}  A{3,3}  A{3,4}  0 . . . . . 0      |
%      |
% Ao = | .              .              .
%      |
%      | .              .              .
%      |
%      | .              .              .
%      |
%      | A{nu-2,1} . . . . . A{nu-2,nu-1}  0      |
%      |
%      | A{nu-1,1} . . . . . A{nu-1,nu} |
%      |
%      | A{nu,1} . . . . . A{nu,nu} |
%      +-
%      +-
%      +-
% Co = | C{1}  0      0      |
%      +-
%      +-
%
```

% Ao is n by n, and Co is l by n .

%

% nu is observability index of the realization.

%

% m=[m(1) m(2) . . . m(nu)] is a vector of the realization Kronecker indices.

%

% and $l \geq m(1) \geq m(2) \geq \dots \geq m(nu)$.

%

% A{j,j} is of dimension m(j) by m(j)

%

% A{j-1,j} is of dimension m(j-1) by m(j) and full column rank.

%

% C(1) is l by m(1) and full column rank

%

% Nothing can be said about the structure of Bo in this coordinate system.

%

```

% The notation used in the subsequent comments is the notation from the
% orthogonal observable canonical form presented above.
%
% References on Orthogonal Observable Canonical Form:
%       Petkov, P. Hr., Christov, N.D., and Konstantinov, M.M.,
%       Computational Methods for Linear Control Systems,
%       Prentice Hall, U.K., 1991, pp. 379-382
%
%       Maciejowsky, J.M., Multivariable Feedback Design,
%       Addison Wesley, 1989, pp.379-382.

disp(' ')
% Check that Aq is in proper form.

[aqr,aqc]=size(Aq) ;
if aqr ~= aqc,
    disp('ERROR: Aq is not square')
    return
end
onesvec=ones(1:aqr-1);

AA=Aq-diag(diag(Aq))-diag(diag(Aq,1),1)-diag(diag(Aq,-1),-1) ;
if max(max(abs(AA))) > 0
    disp('ERROR: Aq is not in proper form')
    return
end

% Form a vector of Kronecker Index sums.

msum=zeros(1:nu) ;
msum(1)=m(1) ;
for i= 2:nu ;
    msum(i)=m(i)+msum(i-1) ;
end

% Establish the size of the matrix AHCo

[n,n]=size(AHCo);

% Check that AHCo is in Orthogonal Observable Canonical Form .
% Check for zeros in the proper locations. Rank tests on the
%  $A\{j-1,j\}$  blocks will be performed later in the program

```

```

for i=1:nu-1,
    if i==1,
        testzeros=AHCo(1:msum(1),msum(i+1)+1:n)      ;
    else
        testzeros=AHCo(msum(i-1)+1:msum(i),msum(i+1)+1:n) ;
    end

    if max(max(abs(testzeros))) > 0 ,
        disp('Error: AHCo is not in Orthogonal Observable Canonical Form!')
        return
    end
end

```

% Check that Co is in Orthogonal Observable Canonical Form

```

[cor,coc]=size(Co);

testzeros=Co(:,m(1)+1:n)      ;
if max(max(abs(testzeros))) > 0
    disp('Error: Co is not in Orthogonal Observable Canonical Form!')
    return
end

```

% Check that Co is full rank

```

rr=rank(Co) ;
if rr < min([cor,coc]),
    disp('Error: Co is not full rank!')
    return
end

```

% Initialize X. X is nu-1 rows by n columns.

```

X=zeros(nu-1,n) ;
X(1,msum(1)+1:n)=f(msum(1)+1:n) ;
% Calculate the Upper Right portion of X.

```

```

for i=2:nu-1 ,           % Row index for X(i,j) Calculations
    for j= msum(i)+1:n , % Column index for X(i,j) Calculations

```

```

        if i-1 == 1,
            X(i,j)=(X(i-1,:)*AHCo(:,j)-Aq(i-1,i-1)*X(i-1,j))/Aq(i-1,i);
        else

```

```

        X(i,j)=(X(i-1,:)*AHCo(:,j)-Aq(i-1,i-1)*X(i-1,j))-...
            Aq(i-1,i-2)*X(i-2,j))/Aq(i-1,i);
    end

end

end

% Now Work from lower Right to Upper Left to Calculate the Remaining
% Unknowns in X. Recall that all unknowns which have not yet been
% calculated are still initialized to zero at this point.

for j=nu:-1:2, % Index into Kronecker Index Sums vector

% create a variable to handle the case when j-2=0, since msum(j-2) is
% used in subsequent calculations, but msum(0) does not exist.      When
% the index msum(j-2)+1 is used in matrix equation, it is desired that
% msum(j-2)+1=1 when j-2=0.

    if j-2 == 0,
        msumjminus2=0;
    else
        msumjminus2=msum(j-2);
    end

    for i=nu-1:-1:j-1 , % Index of row in which calculations are occurring

% When working with the group of columns of AHCo comprised of
% the blocks A{?,j} (notation borrowed from Orthogonal Observable
% Canonical Form) , keep track of the matrix A{j-1,j} in a variable called
% coltop, and the lower part of the columns in a variable called colbottom.
% Recall that everything above the A{j-1,j} block is all zeroes.

        coltop=AHCo(msumjminus2+1:msum(j-1),msum(j-1)+1:msum(j)) ;
        colbottom=AHCo(msum(j-1)+1:n,msum(j-1)+1:msum(j))      ;

% Recalling that coltop= A{j-1,j} , the equations to solve for the
% remaining groups of unknowns in X are of the form
%
% [X(i,msum(j-2)+1):msum(j-1)]*A{j-1,j}=
%     Aq(i,:)*X(:,msum(j-1)+1:msum(j))

```

```

%   -[X(i,msum(j-1)+1:n)]*[A{j,j} ... A{nu,j}]'   Eq(1).
%
% where [X(i,msum(j-2)+1):msum(j-1)] are unknowns, and
% and [X(i,msum(j-1)+1:n)] have already been calculated.

% Let the right side of equation 1 be put into a variable called "right".

right=(Aq(i,:)*X(:,msum(j-1)+1:msum(j)))- ...
      X(i,msum(j-1)+1:n)*colbottom ;

% Then equation (1) is [X(i,msum(j-2)+1):msum(j-1)]*A{j-1,j} = right.
% Using SVD such that A{j-1,j}=UDV', we now have
% [X(i,msum(j-2)+1):msum(j-1)]*U*D= right*V. Solve for the components of
% [X(i,msum(j-2)+1):msum(j-1)]*U, then get [X(i,msum(j-2)+1):msum(j-1)] ,
% from [X(i,msum(j-2)+1):msum(j-1)]=[X(i,msum(j-2)+1):msum(j-1)]*U*U'.

% Find SVD of A{j-1,j}

[U,D,V]=svd(coltop);

% Note the matrices A{j-1,j} should be either square or tall
% and have full column rank in Orthogonal Observable Canonical Form.
% Check for these conditions.

[dr,dc]=size(D) ;
rr=rank(coltop) ;

if dr > dc,
    disp('ERROR: One of the A{j-1,j} is of incorrect dimension!')
    return
end

if rr < min([dr,dc]) ,
    disp('ERROR: One of the A{j-1,j} is not full column rank!')
    disp(' ')
    disp('Check that A matrix is in Orthogonal OCF. ')
    return
end

% Form a temporary variable XU= [[X(i,msum(j-2)+1):msum(j-1)]*U].

[xur,xuc]=size(X(i,msumjminus2+1:msum(j-1))*U) ;
XU=zeros(xur,xuc) ;

```


% The equation is now of the form $XU*D=right*V$. If there are more unknowns
 % than equations, the "extra" unknowns will equal zero. Solve for the
 % first "dc" unknowns of the group, let the rest be zero.

```
RSV=right*V;
for l=1:dc
    XU(l)=RSV(l)/D(l,l);
end
```

% Now get the actual unknowns from the temporary vector XU.

```
X(i,msumjminus2+1:msum(j-1))=XU*U' ;
```

```
end
end
```

% Solve for the unknowns in Bq. The necessary equation is of the form
 % $Bq*Co=Aq*X-X*AHCo$. Use the SVD of Co such that the equation becomes
 % $Bq*U*D=[Aq*X-X*AHCo]*V$. Solve for the elements of Bq*U, then extract
 % Bq by the equation $Bq=[Bq*U]*U'$.

% Get the SVD of Co.

```
[U,D,V]=svd(Co) ;
```

% Set up a temporary variable called BqU.

```
bqur=nu-1;
[dr,dc]=size(D);
bquc=dr;
BqU=zeros(bqur,bquc);
```

% Now solve for the components of BqU.

```
right=(Aq*X-X*AHCo)*V ;

for j=1:dr ; % Column Index.

    BqU(:,j)=right(:,j)/D(j,j) ;
end
```

% Extract Bq from BqU.

```
Bq=BqU*U' ;
```

```

% display the maximum error in the solved Sylvester equation.

error=X*AHCo+Bq*Co-Aq*X ;

maxerr=max(max(abs(error))) ;

disp(' ')
disp('Sylvester equation solution is complete')
disp(' ')
disp(['Maximum Sylvester Equation Error is ',num2str(maxerr)]) ;

end

```

B.3 Pole Placement Example M-files

These files were written to perform the reduced order controller design which achieves arbitrary pole placement. The pole placement examples are presented in the last section of Chapter 4.

Order5.m generates a fifth order strictly proper stabilizing controller which achieves arbitrary closed loop pole placement for the tenth order IMC plant model.

```
% This m-file generates a test problem for the reduced order
% controller design algorithm where complex observer poles are specified.
% A full order observer is designed for a 10th order model of the
% the IMC subsystem of the ACES test facility at NASA Marshall Space Flight
% Center in Huntsville Alabama. A 5th order controller is then designed via
% the new method which achieves the same closed loop eigenvalue placement.
```

```
% Russell D. Glenn, 6-23-95
```

```
% System Sampling Frequency
```

```
Fs= 50 ;
```

```
Ts=1/Fs ;
```

```
% Load the realization
```

```
disp(' ')

```

```
disp('Loading the ACES Model in OOCF')

```

```
disp(' ')

```

```
load acsorth.mat
```

```
A=Ao; B=Bo; C=Co; D=Do;
```

```
disp('The eigenvalues of the open loop system are:')

```

```
disp(' ')

```

```
disp(eig(A))

```

```
disp(' ')

```

```
% Without loss of generality, assume D = 0.
```

```

[drows,dcols]=size(D)      ;
D=zeros(drows,dcols)      ;

[n,n]=size(A);
%%

%Now choose q for B*q, then check controllability of (A,B*q) %

disp(' ')
disp('Choose q, for B*q, then check controllability of A,B*q')
disp(' ')
q=ones(dcols,1)           ;
bq=B*q                     ;
ctg=dgram(A,bq)           ;
rnk=rank(ctg)              ;
%%
if rnk == n ,
    disp('Realization with B*q is controllable')
else
    disp('Warning: Realization with b*q is NOT controllable')
    disp(' ')
    disp('Press any key to continue, or CNTL-C to terminate')
    pause
end

%Place the Poles using Matlab's Place Routine      %

disp(' ')
disp('Placing state feedback poles')
disp(' ')
%    desired state feedback s-plane poles
p1=[-16+28*j -16-28*j -17+28*j -17-28*j -18+28*j -18-28*j] ;
p2=[-19+28*j -19-28*j -20+28*j -20-28*j] ;
p=[p1 p2]' ;

%    desired state feedback z-plane poles
deszpoles1=sort(exp(p*Ts)) ;
f=-place(A,bq,deszpoles1) ;
F=q*f ;

% Check that the poles went where you wanted them

disp(' ')
disp('The Desired and Actual z plane poles from rank 1 state feedback are:')

```

```

disp(' ')
disp('      Desired      Actual ')
disp(' ')
egcs=eig(A+B*F)                                ;
stateigcomp=[ deszpoles1 sort(egcs) ]          ;
disp(stateigcomp)
disp(' ')
pause(3)
%%
%%
%Observer Pole Placement Procedure - perform all of these %
%operations on the dual system to place the observer poles %
%%
%Place the Observer Poles                        %
disp(' ')
disp(' Placing the observer poles ')
disp(' ')

% Choose observer s plane pole locations

p1=[-26+28*j, -26-28*j, -27+28*j, -27-28*j, -27.1 ]    ;
p2=[ -28.9, -29+28*j, -29-28*j, -30+28*j, -30-28*j]    ;
p=[p1 p2]'                                             ;

% Convert to z plane pole locations

deszpoles2=sort(exp(p*Ts))                            ;

%Place the observer poles %
H=-place(A',C',deszpoles2)'    ;
egos=eig(A+H*C)    ;

disp(' ')
disp(' The desired and actual observer z-plane poles are:')
disp(' ')
disp('      Desired      Actual ')
disp(' ')
obseigcomp=[ deszpoles2 sort(egos)]                  ;
disp(obseigcomp)
%%
%%
pause(3)

%Form Closed Loop Observer/Controller and check the eigenvalues %

```

```

% [Kailath, 1980, p. 269]

ac=[A B*F; -H*C A+B*F+H*C] ;
bc=[B; B] ;
cc=[C D*F ] ;
dc=zeros(2,2) ;
fullcleigs=sort(eig(ac)) ;
% save fullcl.mat ac bc cc dc
% %
% Check the eigenvalues of the closed loop A matrix %
disp('Comparison of desired and actual closed loop poles ')
disp('with full order observer/controller compensator: ')
disp(' ')
disp('      Desired      Actual ')
disp(' ')
clcomp=[fullcleigs sort([egcs; egos])] ;
disp(clcomp)
pause(3)
%
% Now build a 5th order compensator which yields the same pole placement.
disp(' ')
disp(' ')
disp(' ')
disp('Building 5th order controller to yield the same pole placement')
%
% First create the matrix ahat
%
ahatpoles=deszpoles2(6:10) ;
ahp=ahatpoles;
ahat=[ahp(1) 1 0 0 0;
      0 real(ahp(2)) imag(ahp(2)) 0 0;
      0 imag(ahp(3)) real(ahp(3)) 1 0;
      0 0 0 real(ahp(4)) imag(ahp(4));
      0 0 0 imag(ahp(5)) real(ahp(5))];

% Construct the matrix (A+HC)

AHC=A+H*C ;

%
% Solve the Sylvester using Solvsylv.m, then build a 5th order controller.
%
disp(' ')
[XM,bhat]=solvsylv(ahat,AHC,C,f,m,nu) ;

```

```

disp(' ')

% Get chat and dhat according to the method in Chapter 3 of the Dissertation.

chat=[ q, zeros(dcols,nu-1) ]      ;
dhat=zeros(dcols,drows)           ;

% Build the reduced order controller

disp(' Building the reduced order controller ')
disp(' ')

ar=ahat + XM*B*chat      ;
br=bhat + XM*H + XM*B*dhat  ;
cr=chat                  ;
dr=dhat                  ;

% Close the loop and check the closed loop eigenvalues

[acr5,bcr5,ccr5,dcr5]=feedback(A,B,C,D,ar,br,cr,dr,-1);

disp('Closed loop eigenvalues with reduced order controller are ')
disp(' ')
redcleigs=sort(eig(acr5))  ;
disp(redcleigs)
% save closed5.mat acr5 bcr5 ccr5 dcr5
end

```

Order4.m generates a fourth order stabilizing controller with nonzero D matrix which achieves arbitrary closed loop pole placement for the tenth order IMC plant model. It is very similar to the m-file **order5.m**.

```

% This m-file generates a test problem for the reduced order
% controller design algorithm where complex observer poles are
% specified.
%
% A full order observer is designed for a 10th order model of the
% the IMC subsystem of the ACES test facility at NASA Marshall
% Space Flight Center in Huntsville Alabama. A 4th order controller
% is then designed via the new method which achieves the same

```

```

% closed loop eigenvalue placement.

% Russell D. Glenn, 6-23-95

% System Sampling Frequency
Fs= 50 ;
Ts=1/Fs ;

% Load the realization

disp(' ')
disp('Loading the ACES Model in OOCF')
disp(' ')
load acsorth.mat

A=Ao; B=Bo; C=Co; D=Do;

disp('The eigenvalues of the open loop system are:')
disp(' ')
disp(eig(A))
disp(' ')

% Without loss of generality, assume D = 0.
[drows,dcols]=size(D) ;
D=zeros(drows,dcols) ;

[n,n]=size(A);
%%

%Now choose q for B*q, then check controllability of (A,B*q) %

disp(' ')
disp('Choose q, for B*q, then check controllability of A,B*q')
disp(' ')
q=ones(dcols,1)
bq=B*q ;
ctg=dgram(A,bq) ;
rnk=rank(ctg) ;
%%
if rnk == n ,
    disp('Realization with B*q is controllable')
else
    disp('Warning: Realization with b*q is NOT controllable')

```



```

disp(' ')
disp('Press any key to continue, or CNTL-C to terminate')
pause
end

%Place the Poles using Matlab's Place Routine          %

disp(' ')
disp('Placing state feedback poles')
disp(' ')
%   desired state feedback s-plane poles
p1=[-16+28*j -16-28*j -17+28*j -17-28*j -18+28*j -18-28*j] ;
p2=[-19+28*j -19-28*j -20+28*j -20-28*j] ;
p=[p1 p2]' ;

%   desired state feedback z-plane poles
deszpoles1=sort(exp(p*Ts)) ;
f=-place(A,bq,deszpoles1) ;
F=q*f ;

% Check that the poles went where you wanted them

disp(' ')
disp('The Desired and Actual z plane poles from rank 1 state feedback are:')
disp(' ')
disp('      Desired      Actual ')
disp(' ')
egcs=eig(A+B*F) ;
stateigcomp=[ deszpoles1 sort(egcs) ] ;
disp(stateigcomp)
disp(' ')
pause(3)
%%
%%
%Observer Pole Placement Procedure - perform all of these %
%operations on the dual system to place the observer poles %
%%
%Place the Observer Poles %
disp(' ')
disp(' Placing the observer poles ')
disp(' ')

% Choose observer s plane pole locations

```

```

p1=[-26+28*j, -26-28*j, -27+28*j, -27-28*j, -27.1 ] ;
p2=[ -28.9, -29+28*j, -29-28*j, -30+28*j, -30-28*j] ;
p=[p1 p2]' ;

% Convert to z plane pole locations

deszpoles2=sort(exp(p*Ts)) ;

%Place the observer poles %
H=-place(A',C',deszpoles2)' ;
egos=eig(A+H*C) ;

disp(' ')
disp(' The desired and actual observer z-plane poles are:')
disp(' ')
disp('      Desired      Actual ')
disp(' ')
obseigcomp=[ deszpoles2 sort(egos)] ;
disp(obseigcomp)
%%
%%
pause(3)

%Form Closed Loop Observer/Controller and check the eigenvalues %
% [Kailath, 1980, p. 269]

ac=[A B*F; -H*C A+B*F+H*C] ;
bc=[B; B] ;
cc=[C D*F ] ;
dc=zeros(2,2) ;
fullcleigs=sort(eig(ac)) ;
% save fullcl.mat ac bc cc dc
%%
% Check the eigenvalues of the closed loop A matrix %
disp('Comparison of desired and actual closed loop poles ')
disp('with full order observer/controller compensator: ')
disp(' ')
disp('      Desired      Actual ')
disp(' ')
clcomp=[fullcleigs sort([egcs; egos])] ;
disp(clcomp)
pause(3)
%
% Now build a 4th order compensator which yields the same pole placement.

```

```

disp(' ')
disp(' ')
disp(' ')
disp('Building 4th order controller to yield the same pole placement')
%
% First create the matrix ahat
%
ahatpoles=deszpoles2(7:10) ;
ahp=ahatpoles

ahat=[ real(ahp(1)) imag(ahp(1)) 0 0 ;
       imag(ahp(2)) real(ahp(2)) 1 0;
       0 0 real(ahp(3)) imag(ahp(3));
       0 0 imag(ahp(4)) real(ahp(4))];

% Construct the matrix (A+HC)

AHC=A+H*C ;

%
% Solve the Sylvester equation using Redsylv.m,
% then build a 4th order controller.
%
disp(' ')
[XM,bhat]=redsylv(ahat,AHC,C,f,m,nu) ;
disp(' ')

% Get chat and dhat according to the method in Chapter 3 of the Dissertation.

chat=[ q, zeros(drows, nu-2) ] ;
diff=F-chat*XM ;
C1=C(1:m(1),1:m(1)) ;
dhat=-diff(:,1:m(1))*inv(C1) ;
disp('Max error in F=chat*X - dhat*C is ')
disp(max(max(abs(F-chat*XM+dhat*C))))

% Build the reduced order controller

disp(' ')
disp(' Building the reduced order controller ')
disp(' ')

ar=ahat+XM*B*chat ;
br=bhat+XM*H+XM*B*dhat ;

```

```
cr=chat          ;  
dr=dhat  
  
% Close the loop and check the closed loop eigenvalues  
  
[acr4,bcr4,ccr4,dcr4]=feedback(A,B,C,D,ar,br,cr,dr,-1);  
  
disp('Closed loop eigenvalues with reduced order controller are ')  
disp(' ')  
redcleigs=sort(eig(acr4))  ;  
disp(redcleigs)  
% save closed4.mat acr4 bcr4 ccr4 dcr4  
end
```

B.4 The Normalized H^∞ Control Problem

These files were written to perform the full order and reduced order normalized H^∞ controller design examples presented in Chapter 5.

Normhinf.m implements the equations necessary to implement full order normalized H^∞ controllers. The equations which are implemented are taken from (Mustafa and Glover, 1991).

```
function [Ak,Bk,Ck,Dk,X,Y,Z,gam]=normhinf(A,B,C,D,gam)

% function [Ak,Bk,Ck,Dk,X,Y,Z,gam]=normhinf(A,B,C,D,gam)
%
% Generate the Normalized Hoo controller for the normalized plant, shown
% below.
%
% Inputs: (A,B,C,D) - Plant Realization
%
%          gam - Max infinity norm for closed loop system
%
% Outputs: X,Y,Z,- Riccati Equation Solutions
%
%          gam - Max infinity norm for closed loop system
%
%          w1          +-----+          z1
%          -----> O-----> |      G(s)      |-----+
%                  ^ +      +-----+      |
%          u=z2 |      +-----+          y      \ /      w2
%              |      +-----+      K(s)      |<-----O<-----
%              +-----+          +
%
% Russ Glenn 1-25-95

% This controller guarantees that the infinity norm of the following transfer
% function matrix is less than gam.
%
%          +--          --+
```

```

%      | [inv(I-GK)]*G [inv(I-GK)]G*K |
%      |                               |
%      | K*[inv(I-GK)]*G K*[inv(I-GK)] |
%      +--                               --+

```

```

% Reference:

```

```

% Mustafa and Glover, Controller Reduction by Hoo Balanced Truncation,
% IEEE Transactions on Automatic Control, June, 1991 p. 668.

```

```

disp(' ')
gamsq=gam^2;

```

```

% Solve the Control Algebraic Riccati Equation:

```

```

X=are(A,(1-1/gamsq)*B*B',C'*C);

```

```

% Solve the Filter Algebraic Riccati Equation:

```

```

Y=are(A',(1-1/gamsq)*C'*C,B*B');

```

```

% Check Gamma Condition:

```

```

XY=X*Y;
if (max(eig(XY)) < gamsq) ,
    disp(' ')
    disp('Gamma Condition is Satisfied');
    disp(' ')
    disp([' Max Eig of X*Y is ', num2str(max(eig(XY)))])
else
    disp('ERROR: Gamma Condition is NOT Satisfied');
    disp(' ')
    disp([' Max Eig of X*Y is ', num2str(max(eig(XY)))])
    disp(' ')
    return
end

```

```

% Form an intermediate result Z.

```

```

[n,n]=size(A);
I=diag(ones(1,n));
Z=inv(I-(1/gamsq)*XY);

```

```

% Form the controller:

Ak = A - (1-1/gamsq)*Y*C'*C - B*B'*X*Z;
Bk = Y*C';
Ck = -B'*X*Z;
[dr,dc]=size(D');
Dk=zeros(dr,dc);

% Check that Controller stabilized the closed loop syste.

[ac1,bcl,ccl,dcl]=feedback(A,B,C,D,Ak,Bk,Ck,Dk,1);
if max(real(eig(ac1))) >= 0
    disp('Error! The closed loop is not stable!')
    disp(' ')
    disp('Terminating Program!!')
    disp(' ')
    break
end

disp(' ')
disp('The Closed Loop is Stable with Full Order Controller')
disp(' ')

save fullcont.mat Ak Bk Ck Dk gam
return

```

Fulltfms.m calculates the transfer functions used to evaluate the closed loop behavior with the full order normalized H^∞ controller.

```

% fulltfms.m
%
% This script file loads the plant matrices aw,bw,cw,dw from the
% file 1in2out.mat and the full order normalized Hoo controller
% matrices Ak,Bk,Ck,Dk from the file fullcont.mat The outputs
% are stored in a file called fulltfms.mat. The four realizations
% are (a11,b11,c11,d11), (a12,b12,c12,d12), (a21,b21,c21,d21) and
% (a22,b22,c22,d22) corresponding to the four transfer functions
% shown below.
%
% Russ Glenn 7-19-95
%

```

```

%          +--                                --+
%          | [inv(I-GK)]*G  [inv(I-GK)]G*K |
%          |                                |
%          | K*[inv(I-GK)]*G  K*[inv(I-GK)] |
%          +--                                --+

% load the plant realization and the controller realization

clear aw bw cw dw
load 1in2out.mat

clear ak bk ck dk
load fullcont.mat

% form a realization for GK

[agk,bgk,cgk,dgk]=series(Ak,Bk,Ck,Dk,aw,bw,cw,dw);

% form a realization of [I-GK] inverse

[as,bs,cs,ds]=feedback([],[],[],eye(size(dgk)*[1;0]),agk,bgk,cgk,dgk,1);

% form a realization for [inv(I-GK)]*G

[a11,b11,c11,d11]=series(aw,bw,cw,dw,as,bs,cs,ds);

% form a realization for [inv(I-GK)]*G*K

[a12,b12,c12,d12]=series(Ak,Bk,Ck,Dk,a11,b11,c11,d11);

% form a realization for K*[inv(I-GK)]*G

[a21,b21,c21,d21]=series(a11,b11,c11,d11,Ak,Bk,Ck,Dk);

% form a realization for K*[inv(I-GK)]

[a22,b22,c22,d22]=series(as,bs,cs,ds,Ak,Bk,Ck,Dk);

save fulltfs a11 b11 c11 d11 a12 b12 c12 d12 a21 b21 c21 d21 ...
           a22 b22 c22 d22 gam

```


Fullresp.m calculates the singular value responses used to evaluate the closed loop behavior with the full order normalized H^∞ controller.

```
% fullresp.m
%
% M-file to compute the singular value frequency responses of
% the transfer functions [inv(I-GK)]*G, [inv(I-GK)]*G*K,
% K*[inv(I-GK)]*G, and K*[inv(I-GK)] to evaluate the
% normalized Hoo controller design.
%
% Russ Glenn 7-19-95

fulltfms % Generate the appropriate transfer functions

load fulltfs.mat
load fullcont.mat
load 1in2out.mat

disp(['Gamma is ',num2str(gam)])

w=logspace(-2,4,150);
w(:);
f=w/(2*pi);
%
% Calculate the Singular Value Response of the Open Loop Plant

disp('Calculating the Singular Values of the Open Loop Plant')
[sigpl,w]=sigma(aw,bw,cw,dw,w);
sigpl1=sigpl(1,:);
disp('Plotting the Singular Values of the Open Loop Plant')
semilogx(w,20*log10(sigpl1));
title(['Singular Values of the Open Loop Plant'])
xlabel('W Plane Frequency, Rad/Sec.')
ylabel('Magnitude, db')
grid
axis([.01 10000 -40 40])
cmet=input('If you want to have a metafile of this plot, type 1 > > ')
if cmet==1
    print -dhpgl plant.hgl
end
cmet=0;
```

% Calculate the Singular Value Response of the Controller

```

disp('Calculating the Singular Values of the Controller')
[sigk,w]=sigma(Ak,Bk,Ck,Dk,w);
sigk1=sigk(1,:);
disp('Plotting the Singular Values of the Controller')
semilogx(w,20*log10(sigk1));
title(['Singular Values of the Controller, Gamma = ',num2str(gam)])
xlabel('W Plane Frequency, Rad/Sec.')
ylabel('Magnitude, db')
grid
axis([.01 10000 -40 40])
cmet=input('If you want to have a metafile of this plot, type 1 > > ')
if cmet==1
    print -dhpgl cont.hgl
end
cmet=0;

```

% Calculate the Singular Value Response $[\text{inv}(I-GK)]*G$.

```

disp('Calculating the Singular Values of  $[\text{inv}(I-GK)]*G$ ')
[sig11,w]=sigma(a11,b11,c11,d11,w);
sigkw1=sig11(1,:);
disp('Plotting the Singular Values of  $[\text{inv}(I-GK)]*G$ ')
semilogx(w,20*log10(sigkw1));
title(['Singular Values of  $[\text{inv}(I-GK)]*G$ , Gamma = ',num2str(gam)])
xlabel('W Plane Frequency, Rad/Sec.')
ylabel('Magnitude, db')
grid
axis([.01 10000 -40 40])
cmet=input('If you want to have a metafile of this plot, type 1 > > ')
if cmet==1
    print -dhpgl g11.hgl
end
cmet=0;

```

%

%

% Calculate the Singular Value Response $[\text{inv}(I-GK)]*G*K$.

```

disp('Calculating the Singular Values of  $[\text{inv}(I-GK)]*G*K$ ')
[sig12,w]=sigma(a12,b12,c12,d12,w) ;
sigiw1=sig12(1,:);
semilogx(w,20*log10(sigiw1));
title(['Singular Values of  $[\text{inv}(I-GK)]*G*K$ , Gamma = ',num2str(gam)])

```

```

xlabel('W Plane Frequency, Rad/Sec.')
ylabel('Magnitude, db')
grid
axis([.01 10000 -40 40])
cmet=input('If you want to have a metafile of this plot, type 1 > > ')
if cmet==1
    print -dhpgl g12.hgl
end
cmet=0;

%
% Calculate the Singular Value Response  $K*[inv(I-GK)]*G$ .
%

disp('Calculating the Singular Values of  $K*[inv(I-GK)]*G$ ')
[sig21,w]=sigma(a21,b21,c21,d21,w);
siggk1=sig21(1,:);
semilogx(w,20*log10(siggk1));
title(['Singular Values of  $K*[inv(I-GK)]*G$ , Gamma = ',num2str(gam)])
xlabel('W Plane Frequency, Rad/Sec.')
ylabel('Magnitude, db')
grid
axis([.01 10000 -40 40])
cmet=input('If you want to have a metafile of this plot, type 1 > > ')
if cmet==1
    print -dhpgl g21.hgl
end
cmet=0;

% Calculate the Singular Value Response  $K*[inv(I-GK)]$ .

disp('Calculating the Singular Values of  $K*[inv(I-GK)]$ ')
[sig22,w]=sigma(a22,b22,c22,d22,w);
sigd1yw1=sig22(1,:);
semilogx(w,20*log10(sigd1yw1));
title(['Singular Values of  $K*[inv(I-GK)]$ , Gamma = ',num2str(gam)])
xlabel('W Plane Frequency, Rad/Sec.')
ylabel('Magnitude, db')
grid
axis([.01 10000 -40 40])
cmet=input('If you want to have a metafile of this plot, type 1 > > ')
if cmet==1
    print -dhpgl g22.hgl
end
cmet=0, clear

```

Rednhinf.m implements the reduced order normalized H^∞ example problem. A fifth order controller is generated for the tenth order IMC plant.

```
function [Akr,Bkr,Ckr,Dkr,Aq,Bq,Cq,Dq]=rednhinf(A,B,C,D,gam)

% function [Akr,Bkr,Ckr,Dkr,Aq,Bq,Cq,Dq]=rednhinf(A,B,C,D,gam)
%
% Attempt to Generate a reduced order nomalized Hoo controller for the
% normalized plant, shown below.
%
% Inputs: (A,B,C,D) - Plant Realization
%
%          (gam) - Desired Gama Threshold
%
% Outputs: (Akr,Bkr,Ckr,Dkr) - Reduced Order Controller Realization
%
%          (Aq,Bq,Cq,Dq) - Controller Parameterization Realization
%
%          w1
%          -----> O-----> |      G(s)      |-----+
%                  ^ +      +-----+      |
%          u=z2 |      +-----+      y      \ /      w2
%              |      +-----+      | <-----O<-----
%              +-----+      +
%
% This controller guarantees that the infinity norm of the following transfer
% function matrix is less than Gamma, if the infinity norm of (Aq,Bq,Cq,Dq)
% is less than Gamma.
%
%          +-+
%          | [inv(I-GK)]*G  [inv(I-GK)]G*K |
%          |
%          | K*[inv(I-GK)]*G  K*[inv(I-GK)] |
%          +-+
%
%          +-+
%          | [inv(I-GK)]*G  [inv(I-GK)]G*K |
%          |
%          | K*[inv(I-GK)]*G  K*[inv(I-GK)] |
%          +-+
%
% Russ Glenn 7-19-95

% Get the Riccati Solutions
%
[ak,bk,ck,dk,X,Y,Z,gam]=normhinf(A,B,C,D,gam);
```

```

gamsq=gam^2;

% Put the realization (A - (1 - 1/gamsq)*Y*C'*C, -C*Z) into orthogonal
% observable canonical form. BB is a dummy variable in this case.

AA = A - (1-1/gamsq)*Y*C'*C;
[bbr,bbc]=size(B);
BB=zeros(bbr,bbc);
CC= -C*Z;
[T,m,l,nu,AHCo,Bo,Co,tol]=orthocf(AA,BB,CC);
AHCo=makzeros(AHCo,tol);
Co=makzeros(Co,tol);
disp(' ')
disp(['nu = ',num2str(nu)]);

% Now solve the Sylvester equation.

[nbr,nbc]=size(B);
f=B'*X*Z*T;
Cq=zeros(nbc,nu);
Cq(1,1)=-1;
Aq=buildaq(nu)
disp(' ')
disp('Solving Sylvester Equation')
disp(' ')
[Xs,Bq]=solvsylv(Aq,AHCo,Co,f,m,nu);

%
% Dq = 0
%

[ncr,ncc]=size(C);
Dq=zeros(nbc,ncr);
disp(['The Gamma Threshold is ',num2str(20*log10(gam)), ' dB'])
disp(' ')
disp('Calculating the S.V. response of (Aq,Bq,Cq,Dq)')
disp(' ')
[sv,w]=sigma(Aq,Bq,Cq,Dq);
semilogx(w,20*log10(sv));
title('Singular Value Response of (Aq,Bq,Cq,Dq)')
xlabel('Frequency, Rad/Sec.')
ylabel('Magnitude, db')
pause(2)
hinfnorm = max(max(abs(sv)));
disp(['The infinity norm of (Aq,Bq,Cq,Dq) is ',...
      num2str(20*log10(hinfnorm)), ' db'])

```

```

disp(' ')
if max(max(abs(sv))) >= gam
    disp('Infinity Norm of (Aq,Bq,Cq,Dq) not less than gam')
    disp('Raise gam or change the matrix Aq in buildaq.m and try again.')
    disp(' ')
    break
end

% Build the reduced order controller, assuming the Sylvester equation
% has been solved successfully.
%
disp('Constructing Reduced Order Controller')
disp(' ')
Akr=Xs*T'*B*Cq + Aq;
Bkr=Bq+Xs*(T'*Y*C' + T'*B*Dq) ;
Ckr=Cq;
Dkr=Dq ;
%
% Check for Stability of the Closed Loop System

[Acl,Bcl,Ccl,Dcl]=feedback(A,B,C,D,Akr,Bkr,Ckr,Dkr,1);
if max(real(eig(Acl))) >= 0
    disp('Error! The closed loop with reduced controller is not stable!!')
    disp(' ')
    disp('Terminating Program!!')
    disp(' ')
    break
end

disp(' ')
disp('Closed loop is stable with reduced controller')
disp(' ')

% The following lines were included to generate the parameterization of
% of Hoo controllers. Uncomment these lines if the parameterization is desired.
%
% [n,n]=size(A);
% T=eye(n);
% A11=T'*(A-(1-1/gamsq)*Y*C'*C-B*B'*X*Z-B*Dq*C*Z)*T ;
% A12=T'*B*Cq;
% A21=-Bq*C*Z*T;
% A22=Aq;

```

```

% Akr=[A11 A12;A21 A22];
% Bkr=[T'*Y*C' + T'*B*Dq ; Bq]
% Ckr=[-B'*X*Z*T-Dq*C*Z*T  Cq];
% Dkr=Dq;

save redcont.mat Akr Bkr Ckr Dkr gam
return

```

Buildaq.m builds the matrix $\hat{A}$ used in solving the Sylvester equation for the reduced order normalized H^∞ controller design example.

```

function Aq=buildaq(nu)

% function Aq=buildaq(nu)
%
% Quick and dirty way to build the A matrix for the controller
% parameterization.
%
% Russ Glenn 1-27-95

aqdiag = -0.5*[100 110 120 130 140 ];
Aq=1.0*(diag(aqdiag)+500*diag(ones(1,nu-1),1));

```

Redtfms.m calculates the transfer functions used to evaluate the closed loop behavior with the reduced order normalized H^∞ controller. The listing is exactly like fulltfms.m in this section, except the names of the .mat files loaded and saved have different names than those in fulltfms.m. A listing of redtfms.m will not be presented.

Redresp.m calculates the singular value responses used to evaluate the closed loop behavior with the reduced order normalized H^∞ controller. The listing is exactly like fullresp.m in this section, except the names of the .mat files loaded and saved have different names than those in fullresp.m. A listing of redresp.m will not be presented.

B.5 The Mixed Sensitivity H^∞ Control Problem

These files were written to perform the full order and reduced order mixed sensitivity H^∞ controller design examples presented in Chapter 5.

Raugsys.m builds the augmented plant as described in Chapter 5.

```
function [a,b1,b2,c1,c2,d11,d12,d21,d22] = raugsys(ag,...
    bg,cg,dg,aw1,bw1,cw1,dw1,aw2,bw2,cw2,dw2,aw3,bw3,cw3,dw3)

% function [a,b1,b2,c1,c2,d11,d12,d21,d22] = ..
%   raugsys(ag,bg,cg,dg,aw1,bw1,cw1,dw1,aw2,bw2,cw2,dw2,aw3,bw3,cw3,dw3)
%
%   Computes the state-space model of the augmented plant for reduced order
%   Hoo design with weighting strategy as follows:
%   plant = [ag bg cg dg]
%       W1 = [aw1 bw1 cw1 dw1] ---- on 'e', the error signal
%       W2 = [aw2 bw2 cw2 dw2] ---- on 'u', the control signal.
%       W3 = [aw3 bw3 cw3 dw3] ---- on 'y', the output signal
%
%   The augmented system P(s) := (A,B1,B2,C1,C2,D11,D12,D21,D22).
%
% This is a modified version of augss.m that comes with Matlab
% Robust Control System Toolbox. The order of the outputs is [y1a y1c y1b]'
% in this version, whereas it is [y1a y1b y1c]' in the Matlab version.
%
% This setup is used to investigate the performance achievable with reduced
% order H infinity controllers. Note dg and dw1 must be zero for
% This problem. dw2 must be a constant times the identity; dw3 is arbitrary.

% The problem setup is tailored to fit into the framework for Hoo design in
% the paper
%   Doyle, J. and Glover, K., "State-space formulae for all stabilizing
%   controllers that satisfy an H-infinity-norm bound and relations to
%   risk sensitivity", Systems and Control Letters, 1988

% Russ Glenn   July 6,1995

% Establish the sizes of the various realizations.
% x's are states, m's are outputs, n's are inputs
```



```

[xg,xg]=size(ag) ;
[mg,ng]=size(dg) ;

[xw1,xw1]=size(aw1) ;
[mw1,nw1]=size(dw1);

[xw2,xw2]=size(aw2) ;
[mw2,nw2]=size(dw2);

[xw3,xw3]=size(aw3) ;
[mw3,nw3]=size(dw3);

% Check for dimensional compatability

disp(' ')
disp('Checking for dimensional compatibility.')
disp(' ')

if nw1 ~= mg
    disp('There is an error in the system dimensions for nw1 or mg!')
    break
end

if nw3 ~= mg
    disp('There is an error in the system dimensions for nw3 or mg!')
    break
end

if nw2 ~= ng
    disp('There is an error in the system dimensions for nw2 or ng!')
    break
end

if nw2 ~= mw2
    disp('The dimensions of dw2 will violate Hoo rank conditions on D12!')
    break
end
disp('Dimensions are compatible.')
disp(' ')
%
% ----- A matrix:
%
a = [ag zeros(xg,xw1+xw2+xw3);-bw1*cg aw1 zeros(xw1,xw2+xw3);...
      zeros(xw2,xg+xw1) aw2 zeros(xw2,xw3);...
```

```

        bw3*cg zeros(xw3,xw1+xw2) aw3];
%
% ----- B matrix:
%
b1 = [zeros(xg,mg);bw1;zeros(xw2+xw3,mg)];
b2 = [bg;-bw1*dg;bw2;bw3*dg];
%
% ----- C matrix:
%
c1 = [-dw1*cg cw1 zeros(mw1,xw2+xw3);
      dw3*cg zeros(mw3,xw1+xw2) cw3 ;
      zeros(mw2,xg+xw1) cw2 zeros(mw2,xw3)];
c2 = [-cg zeros(mg,xw1+xw2+xw3)];
%
% ----- D matrix:
%
d11 = [dw1;zeros(mw2+mw3,mg)];
d12 = [-dw1*dg;dw3*dg;dw2];
d21 = eye(mg);
d22 = -dg;
%
disp('Augmented plant has been constructed.')
disp(' ')
% ----- End of RAUGSYS.M ---

```

Mixhinf.m implements the simplified Glover-Doyle algorithm equations presented in Chapter 5.

```

function [ak,bk1,bk2,ck1,ck2,dk11,dk12,dk21,dk22,gam]=...
        mixhinf(A,B1,B2,C1,C2,D11,D12,D21,D22,gam)

% [ak,bk1,bk2,ck1,ck2,dk11,dk12,dk21,dk22,gam]=...
%        mixhinf(A,B1,B2,C1,C2,D11,D12,D21,D22,gam)
%
% Generates a stabilizing Hoo controller such that the infinity norm
% of Ty1u1 of the closed loop system is less than gam, if such a controller
% exists. The assumed problem setup is a mixed sensitivity problem with W1
% weighting the error signal, W2 weighting the control energy, and W3 weighting
% the plant output.
%
% Requirements: Plant D matrix is zero.

```

```

%           D matrix for W1 is zero.
%           D matrix for W2 is a constant times the identity.
%
% A,B1, . . . D12,D22 are the augmented plant to be generated using the
% m-file raugsys.m written by Russ Glenn.
%
% ak,bk,ck,dk are the SS matrices of the Hoo controller.
%
% gam is gamma, the threshold for the Hoo design.

% This code is a simplified version of the procedure outlined in the paper
% Doyle, J. and Glover, K., "State-space formulae for all stabilizing
% controllers that satisfy an H-infinity-norm bound and relations to
% risk sensitivity", Systems and Control Letters, 1988

% See the m-file raugsys.m for more information.
%
% Russ Glenn   July 7,1995

% Check dimensional requirements.
%
% First build the huge realization. If this works, there are no gross
% dimensional errors.

bigreal = [A B1 B2; C1 D11 D12; C2 D21 D22];

% Assuming you got this far, clear it out to free up memory.

clear bigreal

% Find the numbers of inputs and outputs. The inputs are of dimension m1
% m2. The outputs are of dimension p1 and p2. The number of states is n.

[n,n]=size(A)    ;
[n,m1]=size(B1)  ;
[n,m2]=size(B2)  ;
[p1,n]=size(C1)  ;
[p2,n]=size(C2)  ;

disp(' ')

if p2 ~= m1
    disp('There is an error in the dimensions m1 and p2!')
    disp('Terminating program.')
```

```

        break
    end

% Check that D11=0, D22=0, D12=[0 k*I], and D21=[I]

    if max(max(abs(D11))) > 0 | max(max(abs(D22))) > 0
        disp('D11 or D22 is not zero!')
        disp('Terminating program.')
        break
    end

    [nn,nn]=size(D21);
    diff=D21-eye(nn) ;
    if max(max(abs(diff))) > 0,
        disp('D21 is not an identity matrix!')
        disp('Terminating program.')
        break
    end

    [d12r,d12c]=size(D12);
    if max(max(abs(D12(1:(d12r-d12c),:)))) > 0
        disp('D12 is not in the form [0 k*I] transpose')
        disp('Terminating program.')
        break
    end

    tempD12=D12(d12r-d12c+1:d12r,:);
    diff=tempD12-tempD12(1,1)*eye(d12c);
    if max(max(abs(diff))) > 0,
        disp('D12 is not in the form [0 k*I] transpose')
        disp('Terminating program.')
        break
    end

    disp('The dimensions are compatible and D matrices are in proper format.')

% It is assumed that D12 is in the form [0 I] transpose. If this is not
% the case, the system must be scaled. Since the input D12 is in the form
% [0 k*I] transpose, the u2 input will be scaled by  $\tilde{u}_2 = k*u_2$ , then the
% the matrices B2, D12 and D22 must be scaled by 1/k.

    disp(' ')
    disp('Scaling the U2 input')
    disp(' ')

```

```

k=D12(d12r,d12c);
D12=D12/k;
D22=D22/k;
B2=B2/k ;

disp(' ')
disp('Setting up Riccati Equations')
disp(' ')

D1dot=[D11 D12];
Ddot1=[D11;D21];

gamsq = gam*gam;

RR=zeros(m1+m2,m1+m2);
RR(1:m1,1:m1)=gamsq*eye(m1);
R=D1dot'*D1dot - RR;

RRT=zeros(p1+p2,p1+p2);
RRT(1:p1,1:p1)=gamsq*eye(p1);
RT=Ddot1*Ddot1'-RRT;

B=[B1 B2];
C=[C1;C2];

% Set up the state feedback Riccati Equation

disp(' ----- Solving State Feedback Riccati Equation -----')
disp(' ')

Xricmat=[A zeros(n,n);-C1'*C1 -A'] -([ B;-C1'*D1dot]*inv(R)*[D1dot'*C1 B']);

AXric=Xricmat(1:n,1:n)      ;
P=-Xricmat(1:n,n+1:2*n)  ;
Q=-Xricmat(n+1:2*n,1:n)  ;

wellposed='true' ;
% [X,X1,X2,Xerr,wellposed]=aresolv(AXric,Q,P,'eigen') dos version
[X1,X2,lamp,Xerr,wellposed,X]=aresolv(AXric,Q,P,'eigen') ;
save X.mat X
if wellposed == 'false'
    disp('No solution to the state feedback riccati exists')
    disp('Terminating program.')
    disp(' ')

```

```

        break
    else
        disp('A solution to the state feedback riccati exists')
        disp(' ')
    end

% Form the state feedback matrix F

F=-inv(R)*(D1dot'*C1 + B'*X) ;

% Check that A + BF is stable

abfeigs=eig(A + B*F) ;
if max(real(abfeigs)) > 0
    disp('A + B*F not stable, Riccati Solution is not stabilizing')
    disp('Terminating program.')
    disp(' ')
    break
else
    disp('A + B*F is stable, Riccati Solution is stabilizing')
    disp(' ')
end

% Set up the output injection Riccati Equation

disp(' ')
disp('----- Solving Output Injection Riccati Equation -----')
disp(' ')

Yricmat=[A' zeros(n,n);-B1*B1' -A] -([C';-B1*Ddot1']*inv(RT)*[Ddot1*B1' C]);

AYric= Yricmat(1:n,1:n) ;
P=-Yricmat(1:n,n+1:2*n) ;
Q=-Yricmat(n+1:2*n,1:n) ;

wellposed='true' ;
% [Y,Y1,Y2,Yerr,wellposed]=aresolv(AYric,Q,P,'eigen') ;
[Y1,Y2,lamp,Yerr,wellposed,Y]=aresolv(AYric,Q,P,'eigen') ;

if wellposed == 'false'
    disp('No solution to the output injection riccati exists')
    disp('Terminating program.')
    disp(' ')
    break
end

```

```

else
    disp('A solution to the output injection riccati exists')
    disp(' ')
end

% Form the output injection H

H = -(B1*Ddot1' + Y*C')*inv(RT);

% Check that A + HC is stable

ahceigs=eig(A+H*C)    ;
if max(real(ahceigs)) > 0
    disp('A+H*C not stable, Riccati Solution is not stabilizing.')
    disp('Terminating program.')
    disp(' ')
    break
else
    disp('A+H*C is stable, Riccati Solution is stabilizing')
    disp(' ')
end

disp(' ')
disp('----- Finished with Riccati Equation Solutions -----')
disp(' ')
disp(' ')
disp(' Check that max eigenvalue of X*Y is less than gamma squared ')
disp(' ')

XY = X*Y;
rho = max(eig(XY))    ;
if rho >= gamsq
    disp('Max eigenvalue of X*Y > gamma squared. Raise Gamma and try again')
    disp(' ')
    disp('Terminating program.')
    break
else
    disp(' Test passed')
    disp(' ')
    disp(' ')
end

disp(' Building the Controller ')
disp(' ')

```

```

% Partition F and H

F11 = F(1:m1-p2,:)      ;
F12 = F(m1-p2+1:m1-p2+p2,:) ;
F2  = F(m1+1:m1+m2,:)   ;

H11 = H(:,1:p1-m2)      ;
H12 = H(:,p1-m2+1:p1-m2+m2) ;
H2  = H(:,p1+1:p1+p2)   ;

% Construct Z = inv((- inv(gamsq)*X*Y))

Z=inv([eye(n)-(1/gamsq)*Y*X]) ;

% Construct the controller based on the assumption that D11 and D22 are
% zero. In this case, D1111, D1112, D1121 and D1122 are all zero.

dk11 = zeros(m2,p2)      ;
dk12 = eye(m2)           ;
dk21 = eye(p2)           ;
dk22 = zeros(p2,m2)      ;

bk2 = (B2 + H12)*dk12     ;
ck2 = -dk21*(C2 + F12)*Z  ;

bk1 = -H2                 ;
ck1 = F2*Z                ;

ak = A + H*C + bk2*inv(dk12)*ck1 ;

% Now close the loop and check the stability of the closed loop system.
% If this test is passed successfully, you have a successful controller
% design. Use the matlab LFTF command to build the closed loop system.
% I will build it the same way they did in Matlab's Hinf.m file.

akk=ak;
bkk=bk1;
ckk=ck1;
dkk=dk11;

sysp=[A B1 B2; C1 D11 D12; C2 D21 D22 ]      ;
dimp=[n, size(B1)*[0 1]', size(B2)*[0 1]', ...
      size(C1)*[1 0]', size(C2)*[1 0]' ]      ;

```



```

[ac1,bcl,ccl,dcl]=lftf(sysp,dimp,akk,bkk,ckk,dkk) ;

lamcl = eig(ac1)      ;

if max(real(lamcl)) > 0
% save Ty1u1 for further study
    disp('The closed loop system is unstable!')
    disp('Terminating Program!')
    disp(' ')
else
% save Ty1u1 for further study
    save ty1u1.mat ac1 bcl ccl dcl
    disp('The closed loop system is stable - the controller is successful!!')
    disp(' ')
    disp(' Congratulations! You have designed a successful H-infinity ')
    disp(' controller - a nontrivial task indeed!')
    disp(' ')
    disp(' ')
end

% Reverse the scaling done earlier

disp('Reversing the scaling')
disp(' ')

B2 = k*B2 ;
D12 = k*D12 ;
D22 = k*D22 ;

ck1 = ck1/k ;
dk11 = dk11/k ;
dk12 = dk12/k ;

disp('Finished with design')

```

Sisotest.m generates the full order (13th order) mixed sensitivity controller for the SISO IMC plant for the example presented in Chapter 5.

```

% Main file for rank SISO Mixed Sensitivity H-Infinity
% controller design for SISO plant.

```

```

%
% Russ Glenn 7-18-95
%
load sisoplnt.mat % SISO IMC w-plane model

w=logspace(-2,4,150);
save w.dat w
load w.dat -mat

% Now Design W1 and W3
%
sisow1w3
%
% Now augment the controller then run hinf.

% Use my new augmentation scheme, then try out my new H-inf algorithm.
% Russ Glenn 7-18-95

aw2=[]; bw2=[]; cw2=[];
dw2=.001;

[A,B1,B2,C1,C2,D11,D12,D21,D22]=raugsys(aiw,biw,ciw,diw,aw1,bw1,cw1,dw1,...
    aw2,bw2,cw2,dw2,aw3,bw3,cw3,dw3);
[b1r,b1c]=size(B1) ;
[b2r,b2c]=size(B2) ;

[c1r,c1c]=size(C1) ;
[c2r,c2c]=size(C2) ;

% Try out my new design algorithm! R.G. 7-18-95

[ak,bk1,bk2,ck1,ck2,dk11,dk12,dk21,dk22,gam]=mixhinf(A,B1,B2,...
    C1,C2,D11,D12,D21,D22,1);

acp=ak; bcp=bk1; ccp=ck1; dcp = dk11;

save sisocont.mat acp bcp ccp dcp
clear

```

SisoW1W3.m generates the W_1 and W_3 weighting functions for the SISO mixed sensitivity design example of Chapter 5. The values selected in the design were $w_{low}=0.15$, $w_{high}=30$, $atten=10$, $k_{low}=600$ and $w_{3high}=8000$.

```
% siso1w1w3.m
%
% A script file which generates the weighting functions for the
% SISO plant mixed sensitivity function problem.
%
% Russ Glenn 7-18-95

% First load in the frequency vector.
load w.dat -mat
tpi=2*pi;
wlow=input('Lower Frequency, Rad/Sec. > > ');
whigh=input('Upper Frequency, Rad/Sec. > > ');
atten=input('Input Low Frequency Attenuation Factor: > > ');
klow=input('Low Frequency Gain for 1/W3, > > ');
w3high=input('Input High Frequency Pole Location for w3, Rad/Sec. > > ');

% w1 second order and strictly proper

numw1=atten*(1/.032)*wlow^2 ;
denw1=[1 2*wlow wlow^2] ;

numw3=w3high*(1/(klow*wlow))*[1 wlow] ;
denw3=[1 w3high] ;

% SS realization of W1 , 2nd order and strictly proper
% W1 weights the error signal e.

[aw1,bw1,cw1,dw1]=tf2ss(numw1,denw1) ;

% SS realization of W3
% W3 only weights the plant output signal y1

[aw3,bw3,cw3,dw3]=tf2ss(numw3,denw3) ;
%
[magw1,phasew1,w]=bode(aw1,bw1,cw1,dw1,1,w) ;
[magw3,phasew3,w]=bode(aw3,bw3,cw3,dw3,1,w) ;
```

```

dbw1inv=-20*log10(magw1(:,1));dbw3inv=-20*log10(magw3(:,1));
[magw1,phasew1,w]=bode(aw1,bw1,cw1,dw1,1,w) ;
one=ones(1,length(w))';
dbw1w3=20*log10((one./magw1(:,1))+(one./magw3(:,1)));

semilogx(w, dbw1inv,w,dbw3inv,w,dbw1w3)
title('Frequency Response Plot of 1/W1, 1/W3, and 1/W1+1/W3')
ylabel('Magnitude, db')
xlabel('W-Plane Frequency, Rad/Sec.')
grid
axis([.01 10000 -60 60])
delete weights.hgl
print -dhpgl weights.hgl
save sisowts.dat dbw1inv dbw3inv dbw1w3
axis('auto')

```

Sisotfms.m calculates the transfer functions used to evaluate the closed loop behavior of the SISO system with the full order mixed sensitivity H^∞ controller.

```

% m-file to form the transfer functions of the
% system for h infinity design when the SISO
% plant is used.
%
% Russ Glenn 7-18-95.

load sisocont.mat
load sisopInt.mat

% Form the matrix GK
[agk,bgk,cgk,dgk]=series(acp,bcp,ccp,dcp,aiw,biw,ciw,diw);
%%
%          -1
% form S = [I + GK]
%agko=zeros(agk);, bgko=zeros(bgk);,
% cgko=zeros(cgk);
agko = []; bgko=[]; cgko = [];

[as,bs,cs,ds]=feedback(agko,bgko,cgko,eye(size(dgk)),agk,bgk,cgk,dgk,-1) ;
%
% form transfer function from disturbance input 1 to output.
%

```

```

[ad1,bd1,cd1,dd1]=feedback(aiw,biw,ciw,diw,acp,bcp,ccp,dcp,-1);
%
%          -1
% form T=[I+GK]  GK = I-S
%
at=as;,bt=-bs;,ct=cs;,dt=eye(size(ds))-ds;
%
save  sisotfs.dat agk bgk cgk dgk as bs cs ds at bt ct dt ad1 bd1 cd1 dd1
clear

```

Sisorsp.m calculates the singular value responses of the transfer functions used to evaluate the closed loop behavior of the SISO system with the full order mixed sensitivity H^∞ controller.

```

% M-file to compute the singular value frequency responses of
% the original plant, the sensitivity function, the Complementary
% Sensitivity function, and the transfer function from d1 to y.
% for the SISO H-infinity design.

```

```

% Russ Glenn 7-18-95

```

```

load sisocont.mat
load sisoplnt.mat
load sisotfs.dat -mat
load w.dat -mat
load sisowts.dat -mat

```

```

w=w(:);
f=w/(2*pi);
%
% Calculate the Frequency Response of the Controller.
disp('Calculating the Singular Values of the controller')
[sigkw,w]=sigma(acp,bcp,ccp,dcp,w);
[nr,nc]=size(sigkw);
sigkw1=sigkw(1,:);
disp('Plotting the Frequency Response the controller')
semilogx(w,20*log10(sigkw1));
title('Singular Value Response of Controller for SISO Plant')
xlabel('W Plane Frequency, Rad/Sec.')
ylabel('Magnitude, db')

```

```

grid
axis([.01 10000 -60 60])
cmet=input('If you want to have a metafile of this plot, type 1 > > ')
if cmet==1
%   meta conresp.met
    print -dhpgl conresp.hgl
end
cmet=0;
%
%
% Calculate the Frequency Response of the Open Loop Plant
disp('Calculating the Frequency Response of Open Loop Plant')
[siw,w]=sigma(aiw,biw,ciw,diw,w) ;
disp('Calculating the Singular Values of the Open Loop Plant')
sigw1=siw(1,:);
semilogx(w,20*log10(sigw1));
title('Singular Value Response of the Open Loop SISO Plant')
xlabel('W Plane Frequency, Rad/Sec.')
ylabel('Magnitude, db')
grid
axis([.01 10000 -60 60])
cmet=input('If you want to have a metafile of this plot, type 1 > > ')
if cmet==1
%   meta plntres.met
    print -dhpgl plntres.hgl
end
cmet=0;
%
% Calculate the frequency response of S
%
% First get frequency response of GK by multiplying the response of
% G by the response of K.

disp('Calculating the S.V. Response of GK')
[siggk,w]=sigma(agk,bgk,cgk,dgk,w);
siggk1=siggk(1,:);
semilogx(w,20*log10(siggk1));
title('Singular Values of GK with SISO Plant');
xlabel('W Plane Frequency, Rad/Sec.')
ylabel('Magnitude, db')
grid
axis([.01 10000 -60 60])
cmet=input('If you want to have a metafile of this plot, type 1 > > ')
if cmet==1

```

```

%      meta gkresp.met
      print -dhpgl gkresp.hgl
end
cmet=0;

% Frequency response of (i + gk)inverse
disp('Calculating the Singular Values of S')
[sigsw,w]=sigma(as,bs,cs,ds,w);
sigsw1=sigsw(1,:);
disp('Plotting the Frequency Response of S')
semilogx(w,20*log10(sigsw1),w,dbw1inv);
title('Singular Values of S and 1/W1 with SISO Plant')
xlabel('W Plane Frequency, Rad/Sec.')
ylabel('Magnitude, db')
grid
axis([.01 10000 -60 60])
cmet=input('If you want to have a metafile of this plot, type 1 > > ')
if cmet==1
%      meta sresp.met
      print -dhpgl sresp.hgl
end
cmet=0;

%
% Calculate the frequency response of T
%
disp('Calculating the Frequency Response of T')
% since T=I-S, subtract the response of S from I.
disp('Calculating the Singular Values of t')
[sigtw,w]=sigma(as,bs,-cs,eye(size(ds)*[1 0])'-ds,w);
sigtw1=sigtw(1,:);
disp('Plotting the Frequency Response of T')
semilogx(w,20*log10(sigtw1),w,dbw3inv);
title('Singular Values of T and 1/W3 with SISO Plant')
xlabel('W Plane Frequency, Rad/Sec.')
ylabel('Magnitude, db')
grid
axis([.01 10000 -60 60])
cmet=input('If you want to have a metafile of this plot, type 1 > > ')
if cmet==1
%      meta tresp.met
      print -dhpgl tresp.hgl
end
cmet=0;

```

```

% Calculate singular Value response of t.f. from d1 to y

disp('Calculating the Singular Values of t.f. from d1 to y')
[sigd1yw,w]=sigma(ad1,bd1,cd1,dd1,w);
sigd1yw1=sigd1yw(1,:);
disp('Plotting the Frequency Response of Td1y1')
semilogx(w,20*log10(sigd1yw1));
title('S.V. Response of d1 to y1 with SISO plant')
xlabel('W Plane Frequency, Rad/Sec.')
ylabel('Magnitude, db')
grid
axis([.01 10000 -60 60])
cmet=input('If you want to have a metafile of this plot, type 1 > > ')
if cmet==1
%   meta td1y1.met
    print -dhpgl td1y1.hgl
end
cmet=0;
axis('auto')
clear

```

Rnk1test.m implements the steps to design a 13th order controller for the 10th order one input two output IMC plant.

```

% Main file for rank 1 controller design for 1 input 2 output plant.
%
% Russ Glenn 7-18-95
%
load oneinput.mat % one input, 2 output IMC w-plane model
disp(diw)

w=logspace(-2,4,150);
save w.dat w
load w.dat -mat

% Now Design W1 and W3
%
rnk1w1w3
%
% Now augment the controller then run hinf.

```



```

% Use my new augmentation scheme, then try out my new H-inf algorithm.
% Russ Glenn 7-18-95

aw2=[]; bw2=[]; cw2=[];
dw2=.001;

[A,B1,B2,C1,C2,D11,D12,D21,D22]=raugsys(aiw,biw,ciw,diw,aw1,bw1,cw1,dw1,...
    aw2,bw2,cw2,dw2,aw3,bw3,cw3,dw3);
[b1r,b1c]=size(B1) ;
[b2r,b2c]=size(B2) ;

[c1r,c1c]=size(C1) ;
[c2r,c2c]=size(C2) ;

% Try out my new design algorithm! R.G. 7-18-95

[ak,bk1,bk2,ck1,ck2,dk11,dk12,dk21,dk22,gam]=mixhinf(A,B1,B2,...
    C1,C2,D11,D12,D21,D22,1);

acp=ak; bcp=bk1; ccp=ck1; dcp = dk11;

save rnk1cont.mat acp bcp ccp dcp gam
clear

```

Rnk1W1W3.m generates the W_1 and W_3 weighting functions for a mixed sensitivity design example with a 1 input 2 output version of the IMC plant. The values selected in the design were $w_{low}=.15$, $w_{high}=30$, $atten=10$, $k_{low}=600$ and $w_{3high}=8000$.

```

% rnk1w1w3.m
%
% A script file which generates the weighting functions for the
% 1 input 2 output plant mixed sensitivity function problem.
%
% Russ Glenn 7-18-95

% First load in the frequency vector.
load w.dat -mat
tpi=2*pi;

```

```

wlow=input('Lower Frequency, Rad/Sec. > > ');
whigh=input('Upper Frequency, Rad/Sec. > > ');
atten=input('Input Low Frequency Attenuation Factor: > > ');
klow=input('Low Frequency Gain for 1/W3, > > ');
w3high=input('Input High Frequency Pole Location for w3, Rad/Sec. > > ');

% w1 second order and strictly proper

numw1=atten*(1/.032)*wlow^2 ;
denw1=[1 2*wlow wlow^2] ;

numw3=w3high*(1/(klow*wlow))*[1 wlow] ;
denw3=[1 w3high] ;
% SS realization of W1 , 2nd order
%[aw1,bw1,cw1,dw1]=tf2ss([numw1; 0 0 0; 0 0 0;numw1],denw1,2,2) ;

% SS realization of W1 , 2nd order and strictly proper
% W1 only weights the error signal e1.

[aw1,bw1,cw1,dw1]=tf2ss(numw1,denw1) ;
[nw1,bcw1]=size(bw1);
bw1=[bw1 zeros(nw1,1)] ;
dw1=[dw1 0];

% SS realization of W3
% W3 only weights only the plant output signal y1
[aw3,bw3,cw3,dw3]=tf2ss(numw3,denw3) ;
[nw3,bcw3]=size(bw3);
bw3=[bw3 zeros(nw3,1)];
dw3=[dw3 0];
%
[magw1,phasew1,w]=bode(aw1,bw1,cw1,dw1,1,w) ;
[magw3,phasew3,w]=bode(aw3,bw3,cw3,dw3,1,w) ;
dbw1inv=-20*log10(magw1(:,1));,dbw3inv=-20*log10(magw3(:,1));
[magw1,phasew1,w]=bode(aw1,bw1,cw1,dw1,1,w) ;
one=ones(1,length(w))';
dbw1w3=20*log10((one./magw1(:,1))+(one./magw3(:,1)));

semilogx(w, dbw1inv,w,dbw3inv,w,dbw1w3)
title('Frequency Response Plot of 1/W1, 1/W3, and 1/W1+1/W3')
ylabel('Magnitude, db')
xlabel('W-Plane Frequency, Rad/Sec.')
grid

```

```

axis([.01 10000 -60 60])
delete weights.hgl
print -dhpgl weights.hgl
save rnk1wts.dat dbw1inv dbw3inv dbw1w3
axis('auto')

```

Rnk1tfms.m calculates the transfer functions used to evaluate the closed loop behavior of the system with the full order mixed sensitivity H^∞ controller designed for the 1 input 2 output plant.

```

% m-file to form the transfer functions of the
% system for h infinity design when the 1 input 2 output
% plant is used.
%
% Russ Glenn 7-18-95.

load rnk1cont.mat
load oneinput.mat

% Form the matrix GK
[agk,bgk,cgk,dgk]=series(acp,bcp,ccp,dcp,aiw,biw,ciw,diw);
% %
%          -1
% form S = [I + GK]
% agko=zeros(agk);, bgko=zeros(bgk);,
% cgko=zeros(cgk);
agko = [];, bgko=[];, cgko = [];

[as,bs,cs,ds]=feedback(agko,bgko,cgko,eye(size(dgk)),agk,bgk,cgk,dgk,-1);
%
% form transfer function from disturbance input 1 to output.
%
[ad1,bd1,cd1,dd1]=feedback(aiw,biw,ciw,diw,acp,bcp,ccp,dcp,-1);
%
%          -1
% form T=[I+GK]  GK = I-S
%
at=as;,,bt=-bs;,,ct=cs;,,dt=eye(size(ds))-ds;
%
save rnk1tfs.dat agk bgk cgk dgk as bs cs ds at bt ct dt ad1 bd1 cd1 dd1

```

```
clear
```

Rnk1rsp.m calculates the singular value responses of the transfer functions used to evaluate the closed loop behavior of the 1 input 2 output IMC plant with the full order mixed sensitivity H^∞ controller.

```
% M-file to compute the singular value frequency responses of
% the original plant, the sensitivity function, the Complementary
% Sensitivity function, and the transfer function from d1 to y.
% for the case where a 1 input 2 output plant is used.
%
% Russ Glenn 7-18-95

load rnk1cont.mat
load oneinput.mat
load rnk1tfs.dat -mat
load w.dat -mat
load rnk1wts.dat -mat

w=w(:);
f=w/(2*pi);
%
% Calculate the Frequency Response of the Controller.
disp('Calculating the Singular Values of the controller')
[sigkw,w]=sigma(acp,bcp,ccp,dcp,w);
[nr,nc]=size(sigkw);
sigkw1=sigkw(1,:);
disp('Plotting the Frequency Response the controller')
semilogx(w,20*log10(sigkw1));
title('Singular Value Response of Controller for SIMO Plant')
xlabel('W Plane Frequency, Rad/Sec.')
ylabel('Magnitude, db')
grid
axis([.01 10000 -60 60])
cmet=input('If you want to have a metafile of this plot, type 1 > >')
if cmet==1
%   meta conresp.met
    print -dhpgl conresp.hgl
end
cmet=0;
```

```

%
%
% Calculate the Frequency Response of the Open Loop Plant
disp('Calculating the Frequency Response of Open Loop Plant')
[signw,w]=sigma(aiw,biw,ciw,diw,w) ;
disp('Calculating the Singular Values of Open Loop SIMO Plant')
signw1=signw(1,:);
semilogx(w,20*log10(signw1));
title('Singular Value Response of the Open Loop SIMO Plant')
xlabel('W Plane Frequency, Rad/Sec.')
ylabel('Magnitude, db')
grid
axis([.01 10000 -60 60])
cmet=input('If you want to have a metafile of this plot, type 1 > > ')
if cmet==1
%   meta plntres.met
    print -dhppl plntres.hgl
end
cmet=0;
%
% Calculate the frequency response of S
%
% First get frequency response of GK by multiplying the response of
% G by the response of K.

disp('Calculating the S.V. Response of GK')
[siggk,w]=sigma(agk,bgk,cgk,dgk,w);
siggk1=siggk(1,:);
semilogx(w,20*log10(siggk1));
title('Singular Value Response of GK with SIMO Plant');
xlabel('W Plane Frequency, Rad/Sec.')
ylabel('Magnitude, db')
grid
axis([.01 10000 -60 60])
cmet=input('If you want to have a metafile of this plot, type 1 > > ')
if cmet==1
%   meta gkresp.met
    print -dhppl gkresp.hgl
end
cmet=0;

% Frequency response of (i+gk)inverse
disp('Calculating the Singular Values of [1 0]*S*[1 0]^T ')
[signsw,w]=sigma(as,bs*[1;0],[1 0]*cs,[1 0]*ds*[1 ; 0],w);

```

```

sigsw1=sigsw(1,:);
disp('Plotting the Frequency Response of S')
semilogx(w,20*log10(sigsw1),w,dbw1inv);
title('S. V. Response of [1 0]*S*[1 0]" and 1/W1 with SIMO Plant')
xlabel('W Plane Frequency, Rad/Sec.')
ylabel('Magnitude, db')
grid
axis([.01 10000 -60 60])
cmet=input('If you want to have a metafile of this plot, type 1 > > ')
if cmet==1
%   meta sresp.met
    print -dhpgl sresp.hgl
end
cmet=0;

%
% Calculate the frequency response of T
%
disp('Calculating the Frequency Response of T')
% since T=I-S, subtract the response of S from I.
disp('Calculating the Singular Values of t')
[sigtw,w]=sigma(as,bs,-cs,eye(size(ds)*[1 0]')-ds,w);
sigtw1=sigtw(1,:);
disp('Plotting the Frequency Response of T')
semilogx(w,20*log10(sigtw1),w,dbw3inv);
title('Singular Values of T and 1/W3 with SIMO Plant')
xlabel('W Plane Frequency, Rad/Sec.')
ylabel('Magnitude, db')
grid
axis([.01 10000 -60 60])
cmet=input('If you want to have a metafile of this plot, type 1 > > ')
if cmet==1
%   meta tresp.met
    print -dhpgl tresp.hgl
end
cmet=0;
% Calculate singular Value response of t.f. from d1 to y

disp('Calculating the Singular Values of t.f. from d1 to y')
[sigd1yw,w]=sigma(ad1,bd1,[1 0]*cd1,[1 0]*dd1,w);
sigd1yw1=sigd1yw(1,:);
disp('Plotting the Frequency Response of Td1y1')
semilogx(w,20*log10(sigd1yw1));
title('S.V. Response of d1 to y1 Transfer Function with SIMO Plant')

```

```

xlabel('W Plane Frequency, Rad/Sec.')
ylabel('Magnitude, db')
grid
axis([.01 10000 -60 60])
cmet=input('If you want to have a metafile of this plot, type 1 > >')
if cmet==1
%   meta td1y1.met
    print -dhpgl td1y1.hgl
end
cmet=0;
clear

```

Redtest.m implements the steps necessary to design a 7th order mixed sensitivity H^∞ controller for the 10th order 1 input 2 output IMC plant model. The Augmented plant model is 13th order in this problem.

```

% Main file for generating reduced order mixed sensitivity
% H-Infinity Controller for 1 input 2 output plant
%
% Russ Glenn 7-18-95
%
load oneinput.mat % one input, 2 output IMC w-plane model
disp(diw)

w=logspace(-2,4,150);
save w.dat w
load w.dat -mat

% Now Design W1 and W3
%
redw1w3
%
% Now augment the controller then run hinf.

% Use my new augmentation scheme, then try out my new H-inf algorithm.
% Russ Glenn 7-18-95

aw2=[]; bw2=[]; cw2=[];
dw2=.001;

```

```

[A,B1,B2,C1,C2,D11,D12,D21,D22]=raugsys(aiw,biw,ciw,diw,aw1,bw1,cw1,dw1,...
    aw2,bw2,cw2,dw2,aw3,bw3,cw3,dw3);
[b1r,b1c]=size(B1) ;
[b2r,b2c]=size(B2) ;

[c1r,c1c]=size(C1) ;
[c2r,c2c]=size(C2) ;

% Try out my new design algorithm! R.G. 7-18-95

gam=30;
[acp,bcp,ccp,dcp,gam]=redmix(A,B1,B2,C1,C2,D11,D12,D21,D22,gam);

save redcont.mat acp bcp ccp dcp gam
clear

```

Redmix.m implements the steps to design a 7th order mixed sensitivity H^∞ controller for the 10th order one input two output IMC plant.

```

function [akr,bkr,ckr,dkr,gam]=redmix(A,B1,B2,C1,C2,D11,D12,D21,D22,gam)

%
% function [akr,bkr,ckr,dkr,gam]=redmix(A,B1,B2,C1,C2,D11,D12,D21,D22,gam)
%
% Generates a stabilizing Hoo controller such that the infinity norm
% of Ty1u1 of the closed loop system is less than gam, if such a controller
% The function buildaq.m is then called and used in conjunction with
% the function solvsylv.m to solve a Sylvester equation necessary for
% controller reduction. Once the equation is solved and a solution
% Aq,Bq,Cq,Dq is generated, the S.V. response of (Aq,Bq,Cq,Dq) is plotted.
% The designer is then prompted whether to terminate or build the reduced
% order controller. If the controller is built, the controller S.V.
% response is displayed prior to saving the controller.

% The m-file redtfms.m generates the appropriate transfer functions to
% evaluate the reduced order controller, and the m-file redrsp.m plots
% the output results.

% Russ Glenn 7-18-95
%

```



```

% Requirements: Plant D matrix is zero.
%             D matrix for W1 is zero.
%             D matrix for W2 is a constant times the identity.
%
% A,B1, . . . D12,D22 are the augmented plant to be generated using the
% m-file raugsys.m written by Russ Glenn.
%
% akr,bkr,ckr,dkr are the SS matrices of the reduced order Hoo controller.
%
% gam is gamma, the threshold for the Hoo design.

% This code is a simplified version of the procedure outlined in the paper
% Doyle, J. and Glover, K., "State-space formulae for all stabilizing
% controllers that satisfy an H-infinity-norm bound and relations to
% risk sensitivity", Systems and Control Letters, 1988

% See the m-file raugsys.m for more information.
%
% Check dimensional requirements.
%
% First build the huge realization. If this works, there are no gross
% dimensional errors.

bigreal = [A B1 B2; C1 D11 D12; C2 D21 D22];

% Assuming you got this far, clear it out to free up memory.

clear bigreal

% Find the numbers of inputs and outputs. The inputs are of dimension m1
% m2. The outputs are of dimension p1 and p2. The number of states is n.

[n,n]=size(A)    ;
[n,m1]=size(B1)  ;
[n,m2]=size(B2)  ;
[p1,n]=size(C1)  ;
[p2,n]=size(C2)  ;

disp(' ')

% if p2 ~= m1
% disp('There is an error in the dimensions m1 and p2!')
% disp('Terminating program.')
% break

```

```

% end

% Check that D11=0, D22=0, D12=[0 k*I], and D21=[I]

if max(max(abs(D11))) > 0 | max(max(abs(D22))) > 0
    disp('D11 or D22 is not zero!')
    disp('Terminating program.')
    break
end

% [nn,nn]=size(D21);
% diff=D21-eye(nn) ;
% if max(max(abs(diff))) > 0,
%     disp('D21 is not an identity matrix!')
%     disp('Terminating program.')
%     break
% end

[d12r,d12c]=size(D12);
if max(max(abs(D12(1:(d12r-d12c),:)))) > 0
    disp('D12 is not in the form [0 k*I] transpose')
    disp('Terminating program.')
    break
end
tempD12=D12(d12r-d12c+1:d12r,:);
diff=tempD12-tempD12(1,1)*eye(d12c);
if max(max(abs(diff))) > 0,
    disp('D12 is not in the form [0 k*I] transpose')
    disp('Terminating program.')
    break
end

disp('The dimensions are compatible and D matrices are in proper format.')

% It is assumed that D12 is in the form [0 I] transpose. If this is not
% the case, the system must be scaled. Since the input D12 is in the form
% [0 k*I] transpose, the u2 input will be scaled by  $u2_{\text{tilde}}=k*u2$ , then the
% the matrices B2, D12 and D22 must be scaled by 1/k.

disp(' ')
disp('Scaling the U2 input')
disp(' ')

k=D12(d12r,d12c);

```

```

D12=D12/k;
D22=D22/k;
B2=B2/k ;

disp(' ')
disp('Setting up Riccatti Equations')
disp(' ')

D1dot=[D11 D12];
Ddot1=[D11;D21];

gamsq = gam*gam;

RR=zeros(m1+m2,m1+m2);
RR(1:m1,1:m1)=gamsq*eye(m1);
R=D1dot'*D1dot - RR;

RRT=zeros(p1+p2,p1+p2);
RRT(1:p1,1:p1)=gamsq*eye(p1);
RT=Ddot1*Ddot1'-RRT;

B=[B1 B2];
C=[C1;C2];

% Set up the state feedback Riccatti Equation

disp(' ----- Solving State Feedback Riccati Equation -----')
disp(' ')

Xricmat=[A zeros(n,n);-C1'*C1 -A'] -([ B;-C1'*D1dot]*inv(R)*[D1dot'*C1 B']);

AXric=Xricmat(1:n,1:n) ;
P=-Xricmat(1:n,n+1:2*n) ;
Q=-Xricmat(n+1:2*n,1:n) ;

wellposed='true' ;
% [X,X1,X2,Xerr,wellposed]=aresolv(AXric,Q,P,'eigen') dos version
[X1,X2,lamp,Xerr,wellposed,X]=aresolv(AXric,Q,P,'eigen') ;
if wellposed == 'false'
    disp('No solution to the state feedback riccati exists')
    disp('Terminating program.')
    disp(' ')
    break
else

```

```

        disp('A solution to the state feedback riccati exists')
        disp(' ')
    end

% Form the state feedback matrix F

F=-inv(R)*(D1dot'*C1 + B'*X) ;

% Check that A + BF is stable

abfeigs=eig(A + B*F) ;
if max(real(abfeigs)) > 0
    disp('A + B*F not stable, Riccati Solution is not stabilizing')
    disp('Terminating program.')
    disp(' ')
    break
else
    disp('A + B*F is stable, Riccati Solution is stabilizing')
    disp(' ')
end

% Set up the output injection Riccati Equation

disp(' ')
disp('----- Solving Output Injection Riccati Equation -----')
disp(' ')

Yricmat=[A' zeros(n,n);-B1*B1' -A] -([C';-B1*Ddot1']*inv(RT)*[Ddot1*B1' C]);

AYric= Yricmat(1:n,1:n) ;
P=-Yricmat(1:n,n+1:2*n) ;
Q=-Yricmat(n+1:2*n,1:n) ;

wellposed='true' ;
% [Y,Y1,Y2,Yerr,wellposed]=aresolv(AYric,Q,P,'eigen') ;
[Y1,Y2,lamp,Yerr,wellposed,Y]=aresolv(AYric,Q,P,'eigen') ;

if wellposed == 'false'
    disp('No solution to the output injection riccati exists')
    disp('Terminating program.')
    disp(' ')
    break
else
    disp('A solution to the output injection riccati exists')

```

```

        disp(' ')
    end

% Form the output injection H

    H = -(B1*Ddot1' + Y*C')*inv(RT);

% Check that A+HC is stable

    ahceigs=eig(A+H*C)    ;
    if max(real(ahceigs)) > 0
        disp('A+H*C not stable, Riccati Solution is not stabilizing.')
        disp('Terminating program.')
        disp(' ')
        break
    else
        disp('A+H*C is stable, Riccati Solution is stabilizing')
        disp(' ')
    end

    disp(' ')
    disp('----- Finished with Riccati Equation Solutions -----')
    disp(' ')
    disp(' ')
    disp(' Check that max eigenvalue of X*Y is less than gamma squared ')
    disp(' ')

    XY = X*Y;
    rho = max(eig(XY))    ;
    if rho >= gamsq
        disp('Max eigenvalue of X*Y > gamma squared. Raise Gamma and try again')
        disp(' ')
        disp('Terminating program.')
        break
    else
        disp(' Test passed')
        disp(' ')
        disp(' ')
    end

    disp(' Building the Controller ')
    disp(' ')

% Partition F and H

```

```

F11 = F(1:m1-p2,:)      ;
F12 = F(m1-p2+1:m1-p2+p2,:) ;
F2  = F(m1+1:m1+m2,:)   ;

H11 = H(:,1:p1-m2)      ;
H12 = H(:,p1-m2+1:p1-m2+m2) ;
H2  = H(:,p1+1:p1+p2)   ;

% Construct Z = inv((- inv(gamsq)*X*Y))

Z=inv([eye(n)-(1/gamsq)*Y*X]) ;

% Construct the controller based on the assumption that D11 and D22 are
% zero. In this case, D1111, D1112, D1121 and D1122 are all zero.

dk11 = zeros(m2,p2)      ;
dk12 = eye(m2)           ;
dk21 = eye(p2)           ;
dk22 = zeros(p2,m2)      ;

bk2 = (B2 + H12)*dk12    ;
ck2 = -dk21*(C2 + F12)*Z ;

bk1 = -H2                ;
ck1 = F2*Z               ;

ak = A + H*C + bk2*inv(dk12)*ck1 ;

% Now try to build a reduced order controller.

disp('Attempting to build a reduced order controller.')
disp(' ')

% Check rank of ck1

c1rnk=rank(ck1);

if c1rnk ~= 1
    disp('ck1 is not rank 1. Terminating Program')
    break
end

% Determine the transformation to put the realization

```

```
% (ak-bk2*ck1,ck2) into orthogonal observable canonical
% form.
```

```
AA = ak-bk2*ck1 ;
BB = zeros(n,n) ; % Dummy variable.
CC = ck2 ;
```

```
[T,m,l,nu,AAo,BBo,CCo,tol]=orthocf(AA,BB,CC);
AAo=makzeros(AAo,tol);
CCo = makzeros(CCo,tol);
disp(['nu equals ',num2str(nu)'])
save orthreal AAo CCo
```

```
% Apply the transformation to the controller matrices;
```

```
ak=T'*A*T ;
bk1 = T'*bk1 ;
bk2 = T'*bk2 ;
ck1 = ck1*T ;
ck2 = ck2*T ;
```

```
% Solve the Sylvester Equation
```

```
aaeigs=eig(AAo);
Aq = buildaq(nu);
[XX,Bq]=solvsylv(Aq,AAo,CCo,ck1,m,nu);
pause(2)
[nrx,ncx]=size(XX);
Cq = [ 1 zeros(1,nrx-1)]
Dq = zeros(m2,p2)
save youla Aq Bq Cq Dq XX aaeigs AAo CCo
```

```
% Examine the SV response of Aq,Bq,Cq,Dq
```

```
sigma(Aq,Bq,Cq,Dq);
grid
title('S.V. Response of (Aq,Bq,Cq,Dq)')
continue = input('Enter a 1 to continue > >')
```

```
if continue ~= 1
    disp('Terminating Program')
    disp(' ')
    break
end
```

```

% build the reduced order controller

akk=Aq+XX*bk2*Cq;
bkk = XX*(bk1 + bk2*Dq) + Bq;
ckk = Cq;
dkk=Dq ;

% Now close the loop and check the stability of the closed loop system.
% If this test is passed successfully, you have a successful controller
% design. Use the matlab LFTF command to build the closed loop system.
% I will build it the same way they did in Matlab's Hinf.m file.

sysp=[A B1 B2; C1 D11 D12; C2 D21 D22 ]      ;
dimp=[n, size(B1)*[0 1]', size(B2)*[0 1]', ...
      size(C1)*[1 0]', size(C2)*[1 0]' ]      ;

[ac1,bcl,ccl,dcl]=lftf(sysp,dimp,akk,bkk,ckk,dkk) ;

lamcl = eig(ac1)      ;

if max(real(lamcl)) > 0
% save Tylul for further study
    disp('The closed loop system is unstable!')
    disp('Terminating Program!')
    disp(' ')
else
% save Tylul for further study
    save tylul.mat ac1 bcl ccl dcl
    disp('The closed loop system is stable - the controller is successful!!')
    disp(' ')
    disp(' Congratulations! You have designed a successful H-infinity ')
    disp(' controller - a nontrivial task indeed!')
    disp(' ')
    disp(' ')
end

% Reverse the scaling done earlier

disp('Reversing the scaling')
disp(' ')

B2 = k*B2 ;
D12 = k*D12 ;
D22 = k*D22 ;

```



```

ckr = ckk/k ;
dkr = dkk/k ;

% Rename the other 2 controller matrices

akr=akk;
bkr=bkk;
sigma(akr,bkr,ckr,dkr);
title('S.V. Response of Reduced Order Controller')
disp('Finished with design')

```

BuildAq.m builds the matrix $\hat{A}$ used in the Sylvester equation solution which is a part of the reduced order controller design process.

```

function Aq=buildaq(nu)

% function Aq=buildaq(nu)
%
% Quick and dirty way to build the A matrix for the controller
% parameterization for the reduced order mixed sensitivity H∞
% example problem.
%
% Russ Glenn 7-17-95

Aq = [-8000 1000 0 0 0 0 0 ;
      0 -478.75 500 0 0 0 0 ;
      0 0 -330.15 500 0 0 0;
      0 0 0 -55.458 500 0 0 ;
      0 0 0 0 -56.658 500 0;
      0 0 0 0 0 -.15 500;
      0 0 0 0 0 0 -.15 ];

```

RedW1W3.m generates the W_1 and W_3 weighting functions for used in the reduced order controller design example. The values selected in the design were wlow=.15, whigh=30, atten=10, klow=600 and w3high =8000. This file is identical to

RankW1W3.m presented previously except for the names of the files that are loaded and saved. A listing of RedW1W3.m will not be presented.

Redtfms.m calculates the transfer functions used to evaluate the closed loop behavior of the system with the reduced order mixed sensitivity H^∞ controller designed for the 1 input 2 output plant. The file is identical to Rnk1tfms.m presented above except for the names of files that are loaded and saved. A listing of Redtfms.m is not presented.

Redrsp.m calculates the singular value responses of the transfer functions used to evaluate the closed loop behavior of the 1 input 2 output IMC plant with the reduced order mixed sensitivity H^∞ controller. The file is identical to Rnk1rsp.m presented above except for the names of files that are loaded and saved. A listing of Redrsp.m is not presented.

GLENN, RUSSELL DAVID, Ph.D. November, 1995

Electrical and Computer Engineering

A Reduced Order Controller Design Method Based on the Youla Parameterization of All Stabilizing Controllers (210 pp.)

Director of Dissertation: Dr. R. Dennis Irwin

As control system complexity has increased, design of reduced order controllers (controllers with lower order than the design plant) has become an area of great interest. A new method of reduced order controller design based on the Youla parameterization of all stabilizing controllers is presented. In this method, a reduced order controller can be designed if a set of matrices which solves a Sylvester equation coupled with a standard linear equation can be found. A numerically stable and computationally efficient algorithm is developed which generates the solution and yields controllers with order equal to the minimum of $\nu_o - 1$ and $\nu_c - 1$, where ν_o and ν_c represent the observability and controllability indices, respectively, of the design plant. Because ν_o and ν_c are lower than the plant order for many multi-input, multi-output systems, reduced order controller designs are possible. Design examples are presented based on a control systems test fixture at NASA Marshall Space Flight Center.

The algorithm can also be applied to H^∞ controller design problems to yield reduced order H^∞ controllers for some systems. Examples of successful reduced order controller designs are presented for a normalized H^∞ control problem and a mixed sensitivity H^∞ problem.