Application of Robust Control and Gain Scheduling to Missile Autopilot Design

Final Report

T. E. Bullock
S. L. Fields

September 1, 1998

U.S. Army Research Office

Contract No: DAAH04-94-G-0276

Department of Electrical and Computer Engineering

UNIVERSITY OF FLORIDA

APPROVED FOR PUBLIC RELEASE;

DISTRIBUTION UNLIMITED.

THE VIEWS, OPINIONS, AND OR FINDINGS CONTAINED IN THIS REPORT ARE THOSE OF THE AUTHORS AND SHOULD NOT BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION, POLICY, OR DECISION, UNLESS SO DESIGNATED BY OTHER DOCUMENTATION.

SF 298 MASTER COPY   KEEP THIS COPY FOR REPRODUCTION PURPOSES

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

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

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 1, 1998	3. REPORT TYPE AND DATES COVERED final report, 8/1/94- 1/31/96

4. TITLE AND SUBTITLE
Application of Robust Control and Gain Scheduling to Missile Autopilot Design

5. FUNDING NUMBERS
DAAH04-94-G-0276

6. AUTHOR(S)
T. E. Bullock and S. L. Fields

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)
Department of Electrical and Computer Engineering
University of Florida
Gainesville, FL 32611

8. PERFORMING ORGANIZATION REPORT NUMBER
#1. final

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
ARO 32865.1-MA

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

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

12 b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The problem of gain scheduling LPV systems is studied and applied to missile autopilot design. Necessary and sufficient conditions for determining the Quadratic stabilizing of LPV systems are given. A stablizing LPV controller is found if one exists. The necessary and sufficient conditions are realistically computable with reduced dimension and with only the elements of the Lyapunov matrix P as the unknown variables.

The quadratic Stability results are then extended to the more general case of General Lyapunov stability. The derivation is the same as for Quadratic stability, except there is an additional term that is a function of the derivative of the Lyapunov function. An algorithm is presented that will solve these new conditions.

Finally two methods for finding minimal/near minimal realizations for 2-D systems is presented. The first method requires extracting any greatest common divisors out of a nonminimal realization and then returning the realization back to state-space form. This method does not extend to N-D systems. However the second method, the System Equivalent Based method, can be extended to N-D systems. This method begins with the LPV system description being converted to Rosenbreock's system matrix form. Then the system matrix form is systematically manipulated until it is in state-space form.

14. SUBJECT TERMS
Missile autopilots, Gain scheduling, Robust control

15. NUMBER IF PAGES
114

16. PRICE CODE

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

Application of Robust Control and Gain Scheduling to Missile Autopilot Design

Final Report

T. E. Bullock
S. L. Fields
September 1, 1998

U.S. Army Research Office

Contract No: DAAH04-94-G-0276

Department of Electrical and Computer Engineering

UNIVERSITY OF FLORIDA

APPROVED FOR PUBLIC RELEASE;

DISTRIBUTION UNLIMITED.

THE VIEWS, OPINIONS, AND OR FINDINGS CONTAINED IN THIS REPORT ARE THOSE OF THE AUTHORS AND SHOULD NOT BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION, POLICY, OR DECISION, UNLESS SO DESIGNATED BY OTHER DOCUMENTATION.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
Introduction

One of the many interesting problems related to missile guidance and control, which has applications in other areas as well, is the technique known as gain scheduling. Frequently theoretical results are far ahead of applications, the so called "theory–practice gap". The practice of gain scheduling seems to reverse this idea as the practice is ahead the theory. The current state of the theory and application of gain scheduling is summarized in the review papers by Rugh [1] and by Shamma [2].

Gain scheduling arose naturally in practice by engineers who were versed in the classical textbook control system designs and were faced with the problem of designing a controller for a system which was not really a linear time invariant system. The method converts a non-linear possibly time varying system to a linear system by first linearizing about a present operating point. For example, an autopilot would be designed for given flight conditions of say, 10,000 feet altitude, flying at Mach 1.8 and an angle of attack of 10 degrees. A controller is designed for a number of operating points across the flight envelope. The actual controller would then be a function of the usual feedback variables for the constant operating condition (linearized) design and the operating conditions.

For example, to control a non-linear system

$$\frac{dx(t)}{dt} = f(x(t), u(t), \theta(t))$$

we design a controller for the linearized plant

$$\frac{d\,(\delta x)}{dt} = \frac{\partial f(x_0, u_0, \theta)}{\partial x}\delta x + \frac{\partial f(x_0, u_0, \theta)}{\partial u}\delta u$$

in the form

$$\delta u(t) = k(x_0, u_0, \theta)\delta x.$$

Then we attempt to control the non-linear plant with the (non-linear) controller

$$u(t) = u_0 + k(x_0, u_0, \theta)(x - x_0).$$

The nominal values, u_0, θ, x_0 are taken as constant or time varying by updating at discrete sampling times.

As specified, the controller is designed as if the linearization parameters, x_0, u_0, and θ are constants, so called "frozen coefficients". In order for this procedure to stabilize the system, practitioners of gain scheduling have developed two "rules-of-thumb" in the design of gain scheduled controllers.

1. The scheduling variables should capture the plant's nonlinearities; and

2. the scheduling variables should be slowly varying.

To illustrate the problem with the "frozen coefficient" approach consider the time varying linear system

$$\frac{dx(t)}{dt} = A(t)x(t).$$

If $A(t)$ were constant, asymptotic stability requires the eigenvalues of A to have negative real parts. However, a time varying system may be unstable even if $A(t)$ has eigenvalues with negative real parts. Stability may be assured if $\|\dot{A}(t)\|$ is sufficiently small, a requirement which may not hold in applications. Another weaker sufficient condition is that the symmetric part of $A(t)$, $\frac{1}{2}(A(t) + A^T(t))$, has eigenvalues with negative real parts for all t. This concept is useful in extending gain scheduling to systems which are not slowly varying and is discussed later in this report.

If the usual "frozen coefficient" approach is taken, one of the difficulties in applications is the test to determine if the perturbations in the parameters are sufficiently small and the time variation is sufficiently slow. In spite of the lack of theoretical basis, gain scheduling is more or less a standard technique for the control of non-linear systems.

Recently, Shamma and Athans [2, 3] have begun to provide the theoretical foundation for gain scheduling and to quantify these "rules-of-thumb." They have used Linear Parameter-Varying (LPV) systems as an underlying framework for nonlinear gain scheduled systems. An LPV system is a finite dimensional linear system whose dynamics depend continuously on a time-varying exogenous parameter vector $\theta(t)$. The value of $\theta(t)$ is not known a priori but can be measured on-line. The only a priori knowledge on $\theta(t)$ is typically a knowledge of its range and an upper bound on its time rate of change. This type of system can be used to describe systems ranging from missiles whose exogenous parameters include velocity, altitude and angle-of-attack to chemical processes whose exogenous parameters might include temperature and pressure. Several other researchers have considered the control of LPV systems. Shahruz and Behtash [4] studied LPV systems with arbitrarily fast parameters. Their work presented a method for stabilizing a time–varying linear system by gain scheduling without the slowly varying constraint. They suggested selection of the gain matrix $K(\theta)$, which is a function of the parameter θ such that the left hand side of

$$A(\theta) - B(\theta)K(\theta) + (A(\theta) - B(\theta)K(\theta))^T = -Q(\theta) \tag{1.1}$$

is negative definite. Based on this equation, they present an algorithm for computing the gain matrix $K(\theta)$ and provide a necessary and sufficient condition (i.e. $\tilde{A}_{22} < 0$ where $\tilde{A}_{22}$ is defined in [4]) to show when it is possible to compute a controller $K(\theta)$ for a fixed θ. This condition must be satisfied in order to use their method to gain schedule.

Rugh [5] has commented that Shahruz and Behtash's algorithm is too restrictive. The algorithm, stated in Theorem 3.8 of [4], gives necessary and sufficient conditions not for stability in general but for stability with $V(x,t) = x(t)^T x(t)$ as a Lyapunov function for the system.

For quadratic stability of a linear parameter-varying system, a more general single quadratic Lyapunov function $V(x,t) = x(t)^T P x(t)$ with $P > 0$ can be introduced with corresponding changes to (1.1). Note that P is constant for all θ. The reason quadratic stability is desirable is that it gives a strong form of robust stability with respect to time varying parameters [6].

In this report we extend Shahruz and Behtash's work to include the more general condition of quadratic stability. A new necessary and sufficient condition for quadratic stability of LPV systems will be derived. This condition based only on the open loop system parameters and P will be expressed as a Linear Matrix Inequality (LMI) which allows for efficient computation when checking this condition. This condition allows the computation of the closed loop quadratic Lyapunov function without knowledge of the controller. A couple of examples are provided to demonstrate this new theory and the problem of conventional gain scheduling via "frozen coefficient" controllers. A geometric interpretation of this new condition will be provided and for second order systems a method for graphically representing the set of all valid quadratic Lyapunov functions for the closed loop system will be given. From the P that is computed, a stabilizing controller will be derived. This stabilizing controller and the resulting scheduling law is easy to implement, requiring only a few matrices to be stored in memory. We then discuss how our general condition for quadratic stabilizability relates to previous and concurrent work. Barmish [7] derived an equivalent $\tilde{A}_{22}$ condition for uncertain linear systems with time-varying uncertainties, but the resulting controller was nonlinear and only for SISO systems.

The works of Khargonekar, Zhou, Bernussou and others are briefly discussed. Concurrent work by Becker and Packard is briefly discussed as well. In our discussion, we show how our solution to the $\tilde{A}_{22}$ condition provides a solution to their theorem of quadratically stabilizing an LPV plant via output feedback for which they did not provide a solution.

After discussing quadratic stability we then extend our results to the even more general case of General Lyapunov Stability. Quadratic stability requires that a single Lyapunov function be valid over the entire parameter space. This requirement may be too restrictive if it is known a priori that the parameters will not vary infinetly fast. If the parameter variations are bounded and their rates of change are known then their may exist a time-varying Lyapunov function which is valid over the entire parameter space. We derive necessary and sufficient conditions for General Lyapunov Stability. We discuss how to compute this time-varying Lyapunov function. In fact it is just a natural extension of quadratic stability. There is an extra term in the derivation that is a function of $\dot{P}$.

Also we extend our results of Quadratic Stability to include $\mathcal{L}_2$ performance. We show, for strictly proper systems, how to determine if a controller can be constructed such that the closed-loop system has an induced $\mathcal{L}_2$ norm less than a specified value. A missile autopilot example is provided.

In Chapter 3, we review the work by Apkarian and Gahinet [8]. Their work provides an H_∞ based gain scheduled controller for LPV systems. The parameters are treated as uncertainties and are pulled out into one large uncertainty block. Collecting the parameters in this manner allows the powerful methods of H_∞ theory to be used. The controller is assumed to have an LFT dependence on the parameters. The Small Gain Theorem and the Scale Bounded Real Lemma are used to solve this problem. Scaling is introduced to reduce the conservatism of the Small Gain Theorem but the fact that parameters are real is not exploited. This method provides a nice way to design a controller that satisfies an $\mathcal{L}_2$ performance criteria. During our review we provide a corrected proof of their main theorem. This corrected proof is due to Spillman [9] and Apkarian and Gahinet [10]. In addition, we provide a clear explanation on how to construct a controller. A discussion of numerical issues related to solving for a controller is given as well.

Pulling out the parameters into one large "uncertainty block" requires that the plant has a linear fractional dependence on the uncertainty, i.e. the plant is a rational function of the parameters. It is important that this linear fractional dependence be realized in an efficient manner otherwise the size of the matrices get large resulting in computational problems. This type of realization is directly related to the problem of N-D realization theory. In Chapter 4 we review the status of N-D realization theory. In fact N-D realization theory has not progressed very far due the lack of a minimality test. We review some general methods for obtaining a realization but these do not provide the minimal realization. The method by Belcastro [11] does provide a small realization. The others provide ways of further reducing the realization that is derived.

Kung et.al. [12, 13] do provide a minimality test for 2-D systems by computing greatest common divisors. The problem that Kung et.al. faced was that when the Greatest Common Divisor (GCD) is removed the system no longer was in "state-space" form and they did not know how to return the system back to "state-space" form. In this chapter, we extend the work of Rosenbrock [14] on system equivalence to N-D systems and use it to return a 2-D system back into "state-space" form. We provide a GCD Based algorithm for taking a nonminimal 2-D realization, extracting the GCD and then returning it back into "state-space" form such that the resulting realization is minimal/near minimal. A second 2-D method, the System Equivalent Based Algorithm, does not require a realization but begins with the plant description

P. System equivalent operations are used to transform the realization to a minimal/near minimal "state-space" form. This second method could easily be extended to N-D systems.

CHAPTER 2
Lyapunov Based Gain Scheduling

In this chapter, we investigate gain scheduling based on guaranteeing that the resulting gain scheduled system satisfies a Lyapunov function. From the Lyapunov function both stability and performance can be guaranteed over the entire parameter space. In Section 2.1, we investigate gain scheduling from the standpoint of guaranteeing quadratic stability. A geometric interpretation is then given. A discussion is also provided on the connection of this result with previous work in other related areas. Examples are provided which show the practical implementation of this method of gain scheduling. The next section (Section 2.2) extends the quadratic stability result to general Lyapunov stability. Methods for checking this condition and designing a controller are discussed. The final section discusses how to add $\mathcal{L}_2$-performance to this new gain scheduled controller design method. A missile autopilot design example is provided.

For the purposes of this chapter, the following LPV system is considered.

$$\begin{bmatrix} \dot{x}(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A(\theta(t)) & B(\theta(t)) \\ C(\theta(t)) & D(\theta(t)) \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \tag{2.1}$$

where for all $t > 0$, $x(t) \in \mathcal{R}^n$, $u(t) \in \mathcal{R}^{n_u}$, $y(t) \in \mathcal{R}^{n_y}$, and the real matrices $A(\theta) \in \mathcal{R}^{n \times n}$, $B(\theta) \in \mathcal{R}^{n \times n_u}$, $C(\theta) \in \mathcal{R}^{n_y \times n}$, $D(\theta) \in \mathcal{R}^{n_y \times n_u}$ are all functions of the parameter vector $\theta(t)$. Let this LPV system be denoted as $\mathcal{G}(\theta, A, B, C, D)$ or more concisely $\mathcal{G}_\theta$ if the particular state-space matrices are clear from the context.

$$\theta(t) = (\theta_1(t), \ldots, \theta_{n_p}(t)) \in R^{n_p} \tag{2.2}$$

The parameter variations are not known a priori but are known to belong to some set.

$$\theta_i(t) \in \mathbf{\Omega} \tag{2.3}$$

In addition the system matrices $A(\theta(t))$, $B(\theta(t))$, $C(\theta(t))$ and $D(\theta(t))$ must be bounded for all $\theta(t) \in \mathbf{\Omega}$. With these definitions for the LPV system and its parameters, we can now investigate quadratic stabilization.

2.1 Quadratic Stabilization

Quadratic stability can be formally defined as follows.

Definition 2.1 (Quadratic Stability). *The LPV system given in (2.1) is quadratically stable over the parameter space if there exist a $P \in \mathcal{R}^{n \times n}$, $P = P^T > 0$ such that for all $\theta \in \mathbf{\Omega}$*

$$PA(\theta) + A^T(\theta)P < 0 \tag{2.4}$$

This means that there exists a single quadratic Lyapunov function, $V(x,t) = x^T(t)Px(t)$, such that its derivative is negative definite over the entire parameter space. If we could design a gain scheduled controller such that a single quadratic Lyapunov function exists for the closed-loop system then scheduling on slowly varying parameters would no longer be a restriction. In fact the parameters could vary infinitely fast and the system would remain stable.

To do this, we must first determine if quadratic stability is possible and then design a controller that will achieve it. Shahruz and Behtash in [4] were close in designing such a controller but they only considered the Lyapunov function represented by $P = I$. Restricting a priori to this Lyapunov function is conservative in the sense that for $P = I$ the derivative of the Lyapunov function may not remain negative definite over the entire parameter space but there may exist another $P \neq I$ which has $\dot{V}(x,t) < 0$.

The following theorem, which is a generalization of the result in [4] gives necessary and sufficient conditions for determining quadratic stabilizability for an LPV system. As part of the result, a gain scheduled controller is provided.

Theorem 2.2 (LPV Quadratic Stabilizability). *The LPV system defined in (2.1) is quadratically stable if and only if*

$$\tilde{A}_{22}(\theta) = U_2^T(\theta)(A(\theta)W + WA^T(\theta))U_2(\theta) < 0 \quad \forall\,\theta \in \Omega \tag{2.5}$$

where $W \in \mathcal{R}^{n\times n} < 0$ and the range of U_2 is the null space of $B(\theta)$.

Proof. ($\Longrightarrow$) Suppose the system is stabilizable then there exists a state feedback $K(\theta)$ such that the closed-loop system

$$\dot{x}(t) = (A(\theta) - B(\theta)K(\theta))x(t) + B(\theta)r(t) \tag{2.6}$$

is stable. The Lyapunov function corresponding to (2.6) would be

$$V(x,t) = x(t)^T P x(t) \tag{2.7}$$

$$\begin{aligned} \dot{V}(x,\theta,t) &= -x^T(t)Q(t)x(t) \\ &= x^T(t)\left(PA(\theta) - PB(\theta)K(\theta) + A^T(\theta)P - K^T(\theta)B^T(\theta)P\right)x(t). \end{aligned} \tag{2.8}$$

Therefore

$$P(A(\theta) - B(\theta)K(\theta)) + (A(\theta) - B(\theta)K(\theta))^T P = -Q(\theta) \tag{2.9}$$

is negative definite. An approach is taken similar to Shahruz's approach but makes no restriction on P other than it has to be positive definite. Even though the closed-loop Lyapunov function is a function of the controller $K(\theta)$, we will transform this Lyapunov function in two parts. In one part the controller $K(\theta)$ has complete control over and the other part the controller has absolutely no control over. It is this second part that is of interest. If there exists a matrix $P > 0$ that will make this uncontrollable part negative definite, then the system is quadratically stabilizable and if not then it is not possible to make the system quadratically stable with an LPV feedback control.

Proceeding with the proof, we begin by expanding (2.9). Note that we will drop the notation which shows functional dependence on θ to make the equations easier to read.

$$PA - PBK + A^TP - K^TB^TP = -Q. \tag{2.10}$$

Next take the singular value decomposition of $B = U\Sigma V^T$ and substitute this back in for B yielding

$$PA - PU\Sigma V^TK + A^TP - K^TV\Sigma^TU^TP = -Q. \tag{2.11}$$

Recall that U is orthogonal and P is positive definite, therefore $(PU)^{-1}= U^TP^{-1}$ exists. It will be convenient to introduce the following substitution

$$W = P^{-1}. \tag{2.12}$$

which makes $(PU)^{-1} = U^TW$. Multiplying (2.11) on the left by U^TW yields

$$U^TA - \Sigma V^TK + U^TWA^TP - U^TWK^TV\Sigma^TU^TP = -U^TWQ. \tag{2.13}$$

Multiplying on the right by WU yields

$$U^TAWU - \Sigma V^TKWU + U^TWA^TU - U^TWK^TV\Sigma^T = -U^TWQWU. \tag{2.14}$$

Let

$$\tilde{A} = U^T(AW + WA^T)U \tag{2.15}$$

$$L = V^T KWU \tag{2.16}$$

$$\tilde{Q} = U^T WQWU. \tag{2.17}$$

Then rewriting (2.14) yields

$$\tilde{A} - \Sigma L - L^T \Sigma^T = -\tilde{Q}. \tag{2.18}$$

Now partition U and Σ as

$$U = [U_1 \;\; U_2] \;\; , \;\; \Sigma = \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix}. \tag{2.19}$$

where $U_1 \in \mathcal{R}^{n \times n_u}$, $U_2 \in \mathcal{R}^{n \times (n-n_u)}$ and $\Sigma_1 = \text{diag}[\sigma_{11}, \ldots, \sigma_{n_u n_u}] \in \mathcal{R}^{n_u \times n_u}$. This will partition $\tilde{A}$ as follows

$$\tilde{A} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{12}^T & \tilde{A}_{22} \end{bmatrix} \tag{2.20}$$

where

$$\begin{aligned} \tilde{A}_{11} &:= U_1^T(AW + WA^T)U_1 = \tilde{A}_{11}^T \in \mathcal{R}^{n_u \times n_u} \\ \tilde{A}_{12} &:= U_1^T(AW + WA^T)U_2 \in \mathcal{R}^{n_u \times (n-n_u)} \\ \tilde{A}_{22} &:= U_2^T(AW + WA^T)U_2 = \tilde{A}_{22}^T \in \mathcal{R}^{(n-n_u) \times (n-n_u)} \end{aligned} \tag{2.21}$$

In addition, partition L as

$$L(\theta) = [L_{11}(\theta) \;\; L_{12}(\theta)]. \tag{2.22}$$

Substituting $\tilde{A}(\theta)$, Σ and L from (2.20), (2.19) and (2.22) into (2.18),then we obtain

$$\begin{bmatrix} \tilde{A}_{11}(\theta) - (\Sigma_1(\theta)L_{11}(\theta) + L_{11}^T(\theta)\Sigma_1(\theta)) & \tilde{A}_{12}(\theta) - \Sigma_1(\theta)L_{12}(\theta) \\ \tilde{A}_{12}^T(\theta) - L_{12}^T(\theta)\Sigma_1(\theta) & \tilde{A}_{22}(\theta) \end{bmatrix} = -\tilde{Q}(\theta). \quad (2.23)$$

Since $U(\theta)$ and W are nonsingular, the matrix $-\tilde{Q}(\theta) = -U^T(\theta)WQ(\theta)WU(\theta)$ is negative definite if and only if $-Q(\theta)$ is negative definite which is our original assumption. Multiplying (2.23) from the left and right by $[0\ x^T]$ and $[0\ x^T]^T$, respectively, where x is any vector in $\mathcal{R}^{n-n_u}$, we conclude that $\tilde{A}_{22}(\theta)$ is negative definite.

($\Longleftarrow$) Suppose there exists a $W > 0$ such that $\tilde{A}_{22}(\theta)$ in (2.23) is negative definite. From (2.16), we have free choice of L. Therefore looking at (2.23), if you choose the elements of $L_{11}(\theta) = [l_{ij}(\theta)] \in \mathcal{R}^{n_u \times n_u}$ as

$$l_{ij} = \frac{\tilde{a}_{ij} - \sigma_{jj} l_{ji}}{\sigma_{ii}}, \quad l_{ji} = \text{ your choice } \ i = 1, \dots n_u - 1, \ j = i+1, \dots n_u \qquad (2.24)$$

$$l_{ii} = \frac{\tilde{a}_{ii} + d_{ii}}{2\sigma_{ii}}, \ i = 1, \dots n_u \qquad (2.25)$$

where $d_{ii} > 0$ are arbitrary real numbers for all $i = 1, \dots n_u$ and $\tilde{a}_{ij}$ are the elements of $\tilde{A}$ and, in addition, choose the matrix $L_{12}(\theta) \in \mathcal{R}^{n_u \times n - n_u}$ to be

$$L_{12}(\theta) = \Sigma_1^{-1}(\theta) U_1^T(\theta)(A(\theta)W + WA^T(\theta))U_2(\theta), \qquad (2.26)$$

then we obtain from (2.23) that $-\tilde{Q}(\theta) = \text{diag}[-d_{11}, \dots, -d_{n_u n_u}, \ \tilde{A}_{22}(\theta)]$ which is negative definite. □

Lemma 2.3. *If there exists a* $W > 0$ *such that* $\tilde{A}_{22}(\theta) < 0$, *then a stabilizing controller,* $K(\theta)$, *can be chosen as*

$$K(\theta) = V(\theta)L(\theta)U^T(\theta)P \qquad (2.27)$$

where $P = W^{-1}$ *and* $L(\theta) = [L_{11}(\theta)L_{12}(\theta)]$ *with the elements of* $L_{11}(\theta) = [l_{ij}(\theta)] \in \mathcal{R}^{n_u \times n_u}$ *are*

$$l_{ij} = \begin{cases} \frac{\tilde{a}_{ij}-\sigma_{jj}}{\sigma_{ii}} & i < j, \\ \frac{\tilde{a}_{ii}+d_{ii}}{2\sigma_{ii}} & i = j \\ 1 & i > j \end{cases} \tag{2.28}$$

$$l_{ij} = \frac{\tilde{a}_{ij} - \sigma_{jj} l_{ji}}{\sigma_{ii}}, \;\; l_{ji} = \text{ your choice } \; i = 1, \ldots n_u - 1, \; j = i+1, \ldots n_u \tag{2.29}$$

$$l_{ii} = \frac{\tilde{a}_{ii} + d_{ii}}{2\sigma_{ii}}, \; i = 1, \ldots n_u \tag{2.30}$$

where $d_{ii} > 0$ *are arbitrary real numbers for all* $i = 1, \ldots n_u$ *and* $\tilde{a}_{ij}$ *are the elements of* $\tilde{A}$*. The matrix* $L_{12}(\theta) \in \mathcal{R}^{n_u \times n - n_u}$ *is*

$$L_{12}(\theta) = \Sigma_1^{-1}(\theta) U_1^T(\theta)(A(\theta)W + WA^T(\theta))U_2(\theta). \tag{2.31}$$

Proof. Assuming that $W > 0$ exists such that $\tilde{A}_{22}(\theta) < 0$, then from the proof of Theorem 2.2 we see that the particular choice of $L(\theta) = \begin{bmatrix} L_{11}(\theta) & L_{12}(\theta) \end{bmatrix}$ from (2.28) and (2.31) will make the system stabilizable. Recall (2.16)

$$L(\theta) = V^T(\theta)K(\theta)WU(\theta).$$

Solving for $K(\theta)$ yields

$$K(\theta) = V(\theta)L(\theta)U^T(\theta)P$$

which is our stated controller. □

The controller $K(\theta)$ places the controllable poles of $\tilde{Q}(t)$ which indirectly places the poles of $Q(t)$. Performance can be introduced by proper choice of $\tilde{Q}(t)$. This will be discussed further in Section 2.3. The $\tilde{A}_{22}$ condition is an LMI. LMI's are convex and allow for convex optimization techniques to be used to solve them. Since (2.5) is a continuous function of time and must be satisfied at all times, this problem represents a convex feasibility problem with an infinite number of constraints. For practical reasons the control engineer will have to resort to gridding the parameter space and solve approximations to the actual condition (2.5). The disadvantage is that as the number of parameters increase the number of constraints corresponding to each grid point will increase exponentially. The limit on the number of parameters will be determined by the computational capability that is available to the control engineer and the time frame allowed for the controller design. Fortunately once the $\tilde{A}_{22}$ condition has been solved, the scheduling law is simple to implement and does not require storing a controller for each operating condition (i.e. grid point) and then having to interpolate between controllers as the system moves around the parameter space. The controller given in (2.27) only requires storage of a few matrices and knowledge of the current operating conditions in order to be implemented.

Two examples are now presented that illustrate a problem that can be encountered when gain scheduling with frozen point controllers but can be overcome by the Lyapunov based gain scheduling method.

Example 2.4. *Given the open loop LPV plant*

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 1 \\ -(1+.5\theta) & -.2 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} x \\ -1 &\leq \theta \leq 1, \end{aligned} \tag{2.32}$$

and a state feedback matrix $K = \begin{bmatrix} k_1 & k_2 \end{bmatrix}$ *with*

$$\begin{aligned} k_1 &= \frac{1}{2}\theta + 0.02 \\ k_2 &= .1, \end{aligned} \tag{2.33}$$

the closed-loop poles will be in left half-plane for all values of θ *as shown in Figure 2.1. Even though the poles are in the left half-plane the response of the system with initial condition* $x_0 = \left[\begin{smallmatrix} 1 \\ -1 \end{smallmatrix}\right]$ *and with* $\theta = \cos(2t)$ *is unstable as shown in Figure 2.2.*

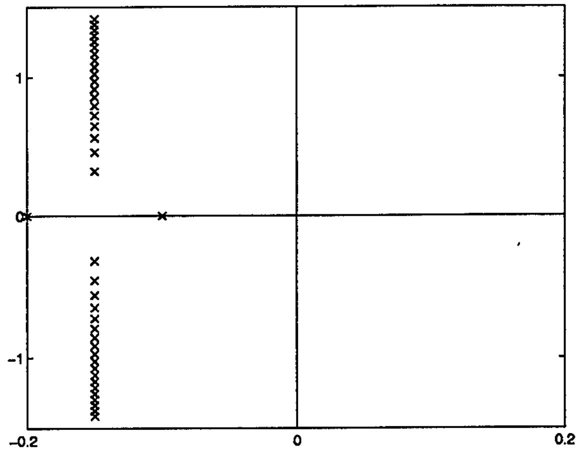

Figure 2.1: The closed-loop pole locations for $-1 \le \theta \le 1$ using the frozen point controller.

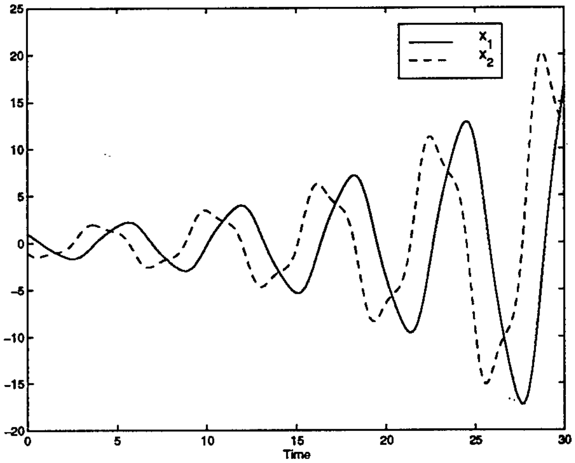

Figure 2.2: The response of the system with $x_0 = [1\ -1]^T$ and $\theta = \cos(2t)$ using the frozen point gain scheduled controller.

Now to design a Lyapunov based controller. First determine a Lyapunov P that satisfies the $\tilde{A}_{22}$ *condition by solving* (2.5) *with an LMI solver. For this example the system was gridded over 20 evenly spaced points over the parameter space. SDPSOL [15] was used and the following P was obtained.*

$$P = \begin{bmatrix} 1.1250 & 0.3750 \\ 0.3750 & 1.1250 \end{bmatrix} \tag{2.34}$$

The controller is designed from Lemma 2.3

$$K(\theta) = V(\theta)L(\theta)U^T(\theta)P. \tag{2.35}$$

Using (2.28) and (2.31) to compute L and with $d_{11} = 10$, *the matrix L can be computed. Figure 2.3 shows the closed-loop response using the Lyapunov based controller (2.27) with inititial condition* $x_0 = \left[\begin{smallmatrix} 1 \\ -1 \end{smallmatrix}\right]$ *and with* $\theta = \cos(2t)$. *The Lyapunov based controller drives the states to zero and does not go unstable since the controller was designed such that the closed-loop system always satisfies the Lyapunov Stability Condition under any parameter trajectory.*

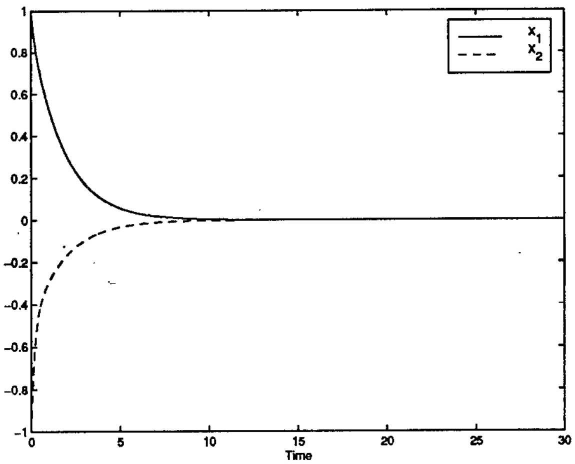

Figure 2.3: The response of the system using the Lyapunov based gain scheduled controller with inititial condition $x_0 = [1\ -1]^T$ and with $\theta = \cos(2t)$.

□

Example 2.5. *In this example consider the open-loop plant*

$$\dot{x} = \begin{bmatrix} -1 + 1.5\cos^2(t) & 1 - 1.5\sin(t)\cos(t) \\ 1 & -1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \tag{2.36}$$

where t is our parameter. We need only consider $0 \leq t \leq 2\pi$ when designing controllers. For the frozen point controller let

$$K = \begin{bmatrix} 2 + 1.5\sin(t) & -\sin^2(t) \end{bmatrix}. \tag{2.37}$$

With this controller the resulting closed-loop system A matrix is

$$A_{cl} = \begin{bmatrix} -1 + 1.5\cos^2(t) & 1 - 1.5\sin(t)\cos(t) \\ -1 - 1.5\sin(t)\cos(t) & -1 + 1.5\sin^2(t) \end{bmatrix} \tag{2.38}$$

which has closed-loop poles located at $-0.25 \pm j0.66144$ for all time. This closed-loop matrix is Vidyasagar's example of a time-varying plant which has constant left half-plane poles and yet is unstable. Figure 2.4 demonstrates this instability in the response of the system with inititial condition $x_0 = [\begin{smallmatrix}1\\1\end{smallmatrix}]$.

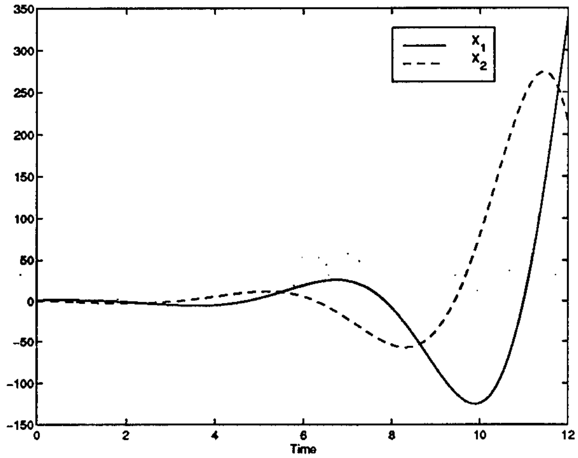

Figure 2.4: The esponse of the system using the frozen point controller with inititial condition $x_0 = [1\ 1]^T$.

Now to design a Lyapunov based gain scheduled controller (2.27). First determine a Lyapunov P that satisfies the $\tilde{A}_{22}$ condition. One such P is

$$P = \begin{bmatrix} 9.0046 & 8.4899 \\ 8.4899 & 9.0046 \end{bmatrix} \qquad (2.39)$$

The controller is designed from Lemma 2.3

$$K(\theta) = V(\theta)L(\theta)U^T(\theta)P. \qquad (2.40)$$

Using (2.28) and (2.31) to compute L and with $d_{11} = 10$, the matrix L can be computed. The response of the resulting closed-loop system using the Lyapunov based controller with inititial condition $x_0 = \left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$ is shown in Figure 2.5. Under the same parameter trajectory, the Lyapunov based gain scheduled design is stable where as the frozen point design is not.

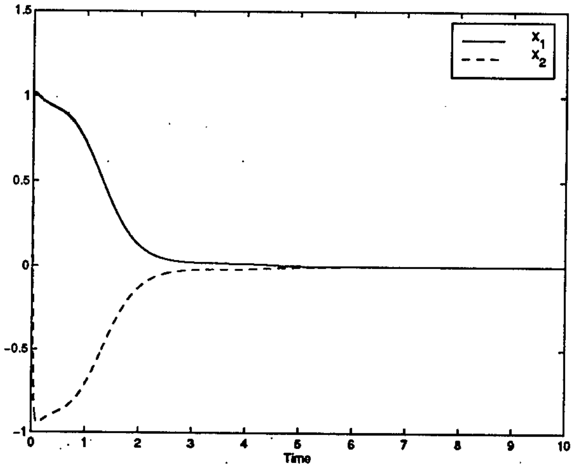

Figure 2.5: The response of the system using the Lyapunov based controller with inititial condition $x_0 = [1\ 1]^T$.

□

To gain a better understanding of this new gain scheduling law, a geometric interpretation of the $\tilde{A}_{22}$ condition is given in the next section.

2.1.1 Geometric Interpretation

The submatrix $\tilde{A}_{22}$ represents the part of the closed-loop Lyapunov function for which the controller has no control. Therefore, if it is not negative definite, the derivative of the Lyapunov function will not be negative definite and quadratic stability can not be attained. In this section, as we investigate the meaning of $\tilde{A}_{22} < 0$, we will be able to derive the set of all positive definite W's that satisfy $\tilde{A}_{22} < 0$. This in turn also specifies the set of all positive definite P's which satisfy the Lyapunov equation (2.9) because $P = W^{-1}$. We will be able to specify this set without specification of $K(\theta)$. Note that $\tilde{A}_{22}$ is only a function of the open loop system matrices $A(\theta)$ and $B(\theta)$.

The $\tilde{A}_{22}$ condition (2.5) is

$$U_2^T(AW + WA^T)U_2 < 0. \tag{2.41}$$

And since a quadratic constraint depends only on the symmetric part of the matrix, this is equivalent to

$$z^T\left(U_2^T AWU_2\right)z < 0 \quad \forall z. \tag{2.42}$$

Let $v = U_2 z$, i.e. v is a vector in the range of U_2. Substituting this in gives

$$v^T AWv < 0 \quad \forall v = U_2 z. \tag{2.43}$$

Hence the quadratic condition in (2.42) can be replaced by another quadratic condition in a larger space subject to a linear side condition. In order to get a geometric interpretation, define $\dot{v} = A^T v$ and substitute this back into our equation. Therefore

$$\dot{v}^T Wv < 0 \quad \forall v = U_2 z. \tag{2.44}$$

From (2.44), we can determine its geometrical interpretation for the second order case. Equation (2.44) is the dot product of two vectors: $\dot{v}$ and Wv. Wv is the

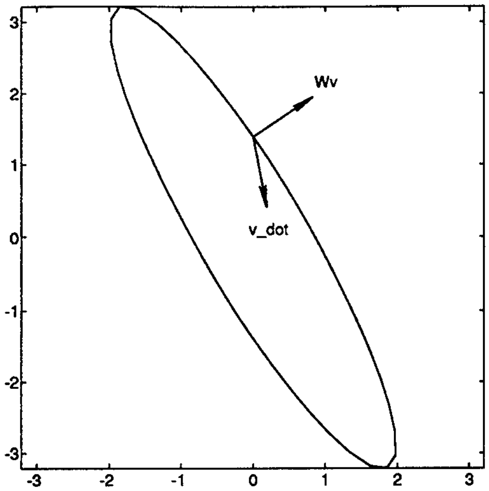

Figure 2.6: An example of what Wv and $\dot{v}$ might look like plotted on the ellipse described by W.

vector, located at the point $(v1, v2)$, that is perpendicular to the ellipse drawn by W and passing through the point $(v1, v2)$. This is illustrated in Figure 2.6. The vector $\dot{v} = A^T v$ is the derivative of v at $(v1, v2)$. If the dot product, $\dot{v}^T W v$, is negative then the angle between $\dot{v}$ and Wv is greater than 90°. This means that $\dot{v}$ points inside the ellipse at the location $(v1, v2)$.

To summarize the meaning of (2.5) for the second order case, we have that if you can draw an ellipse described by W such that at the point of intersection between the ellipse and the line described by $v = U_2 z$ the resulting $\dot{v}$ at this intersection points inside the ellipse, then you can say that the system is stabilizable. In other words, the positive definite matrix W which describes this ellipse satisfies (2.5) and therefore it also satisfies the Lyapunov equation (2.9) for some K. Figure 2.7 gives an example picture of a valid ellipse for a given second order system. Remember that the ellipses are drawn in the phase plane plot of A^T and not A. (This is why the variable was labelled v instead of x the state).

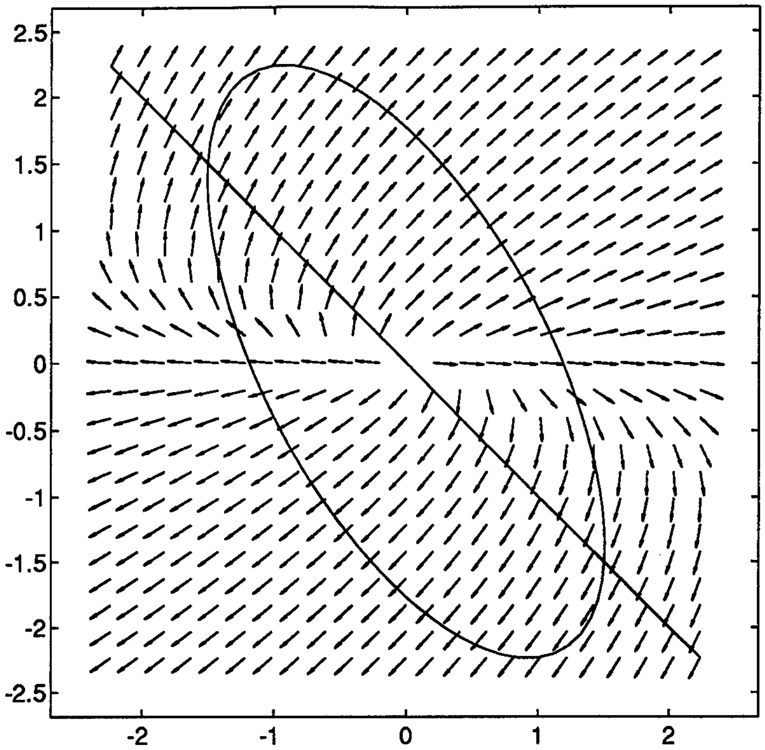

Figure 2.7: An example of a valid ellipse for a given second order system.

For the n^{th} order system this means that if the velocity vector calculated from $\dot{v} = A^T v$ points inside an ellipsoid W on the intersection with $v = U_2 z$ (i.e. the range of U_2), then the system is stabilizable.

It is instructive to consider the set of all positive definite matrices which satisfy the $\tilde{A}_{22}$ condition for a fixed θ. For the second order case we can take $W_{11} = 1$ without loss of generality. The condition $W > 0$ implies $W_{11} > 0$, $W_{22} > 0$ and $W_{11}W_{22} - W_{12}^2 > 0$. Therefore, we have

$$-\sqrt{W_{22}} < W_{12} < \sqrt{W_{22}}. \tag{2.45}$$

By manipulation, the $\tilde{A}_{22}$ condition in this case reduces to

$$aW_{12} < bW_{22} + c \tag{2.46}$$

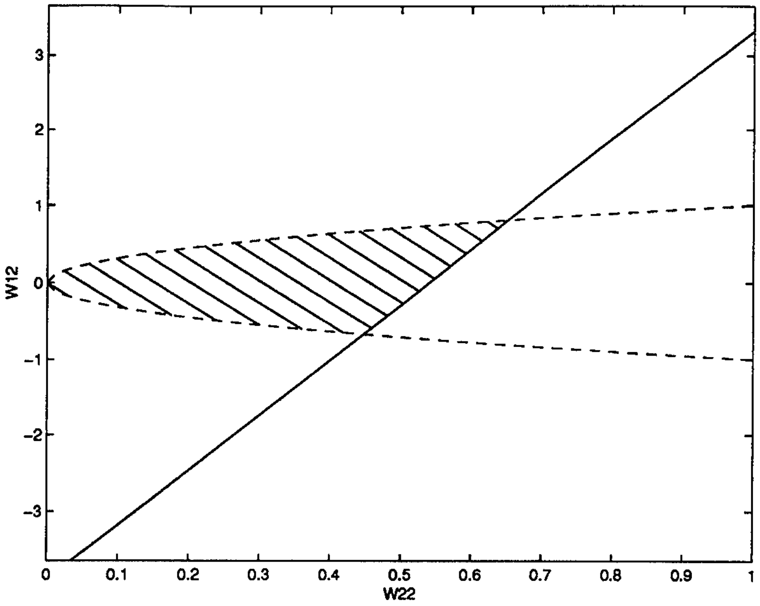

Figure 2.8: An example of the set of all $W > 0$ which satisfy the $\tilde{A}_{22}$ condition for a given second order system.

where

$$\begin{aligned} a &= (u_{2_1}^2 A_{11} + u_{2_1} u_{2_2}(A_{12} + A_{21}) + u_{2_2}^2 A_{22}) \\ b &= (A_{11} u_{2_1} u_{2_2} + A_{21} u_{2_2}^2) \\ c &= (A_{21} u_{2_1}^2 + A_{22} u_{2_1} u_{2_2}) \end{aligned} \qquad (2.47)$$

The shaded region in Figure 2.8 represents a typical plot of the set of W's which satisfy the $\tilde{A}_{22}$ condition for a second order LTI plant. This region is bounded on the left by the dashed line corresponding to the positive definite condition on W as given in (2.45) and on the right by the solid straight line corresponding to the $\tilde{A}_{22}$ condition as given in (2.46).

While Figure 2.8 represents an LTI system, Figure 2.9 represents an LPV system. In particular, it is the phase plane plot of the LPV plant given in (2.38) with $B = \begin{bmatrix} \cos(t + \pi/18) \\ -\sin(t + \pi/18) \end{bmatrix}$. One hundred grid points were taken. Each straight line represents a frozen coefficient plant at the associated grid point. Looking closely at

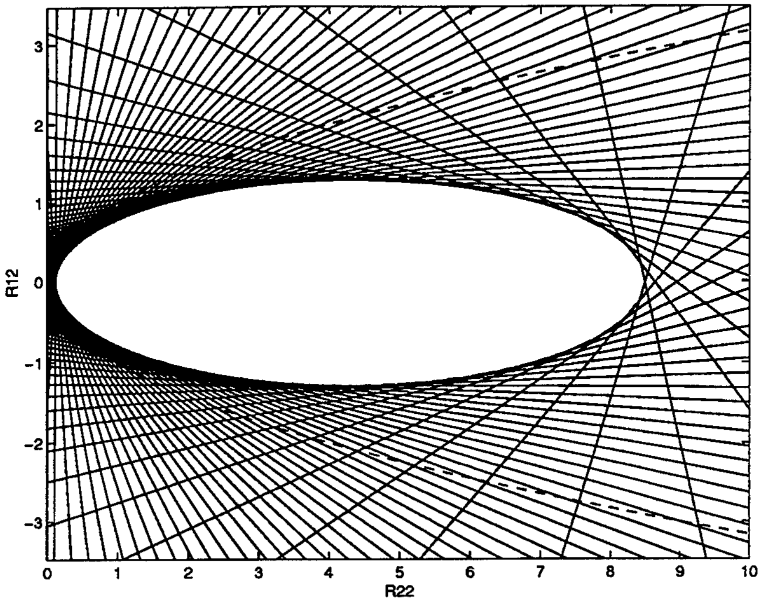

Figure 2.9: The set of all $W > 0$ which satisfy the $\tilde{A}_{22}$ condition for Vidyasagar's example.

the inequalities that are associated with each straight line to determine the valid Lyapunov region and then taking the intersection of all these regions results in the intersection being the clear oval shape in the middle of the plot. Therefore, this oval shape represents the set of all valid Lyapunov functions for this LPV plant. By choosing a W inside this region, the control designer can stabilize this plant.

2.1.2 Relation to Previous Work

Quadratic Stabilizability of Uncertain Systems

When the $\tilde{A}_{22}$ condition was first derived, we did not realize that the stability that we were guaranteeing was quadratic stability. This fact did not become evident until we discovered the concurrent work of Becker et al. [6] and Packard et al. [16] which investigated quadratic stabilization of LPV systems. These and related works will be discussed later. A literature survey was then conducted on quadratic stabilization of linear systems. As it turns out, the condition $\tilde{A}_{22} < 0$ is fundamental in determining quadratic stabilization of linear systems. Our $\tilde{A}_{22}$ condition is an extension to LPV systems of the work conducted by Barmish [7] on uncertain linear systems with time-varying uncertainties. In particular, he considered the system

$$\dot{x} = A(r(t))x + B(s(t))u \tag{2.48}$$

$$\theta(t) = \begin{bmatrix} r(t) \\ s(t) \end{bmatrix} \in \mathcal{R}^p, \tag{2.49}$$

where $r(t) \in \mathcal{R}^p_r$ represents the model parameter uncertainty and $s(t) \in \mathcal{R}^p_s$ represents the input connection parameter uncertainty. The only difference between this and our LPV definition is that Barmish does not make the assumption that the parameters can be measured on-line. Before Barmish's work much of the work on satisfying quadratic stabilization centered around the solution of so called matching conditions [17, 18]. Barmish enlarged considerably on these conditions and provided necessary and sufficient conditions for quadratic stabilizability. In fact his conditions are identical to $\tilde{A}_{22} < 0$ for systems with time-varying uncertainties just stated in a different manner. The problem is that the resulting controller is a nonlinear function of the states and is only valid for single input systems.

Upon Barmish's discovery of these necessary and sufficient conditions for quadratic stabilizability, many researchers focused their efforts on determining the conditions needed to allow for quadratic stabilization of uncertain systems via linear control. This type of stabilization was referred to as linear quadratic stabilization.

Zhou and Khargonekar [19] pursuing the goal of linear quadratic stabilization focused their attention on systems with norm bounded time-varying uncertainties entering both the state matrix, A, and the input matrix, B. They described their system as follows.

$$\dot{x} = [A + \Delta A]x + [B + \Delta B]u \tag{2.50}$$

$$[\Delta A \; \Delta B] = DF(t)E \tag{2.51}$$

where A, B, D, E are known constant matrices and $F(t) \in \mathbf{F} \subset \mathcal{R}^{p \times q}$ is the modeling or parameter uncertainty. The set $\mathbf{F}$ is assumed to be Lebesgue measurable and is defined as

$$\mathbf{F} := \{F(t) \mid F^T(t)F(t) \leq I\}. \tag{2.52}$$

They showed that linear quadratic stabilization for this class of uncertain systems is equivalent to the existence of a positive definite matrix solution to a Riccati equation. An LTI controller is given in terms of the solution to this Riccati equation.

Khargonekar and Rotea in [20] show that quadratic stabilizability is equivalent to linear quadratic stabilizability for systems with norm bounded uncertainties. In particular, they consider the system

$$\dot{x}(t) = Ax(t) + Bu(t) + Dw(t) \tag{2.53}$$

$$e(t) = E_1 x(t) + E_2 u(t) \tag{2.54}$$

$$w(t) = \Delta(t)e(t) \tag{2.55}$$

where $w(t) \in \mathcal{R}^k$, $e(t) \in \mathcal{R}^p$ and the real matrices A, B, D, E_1 and E_2 are known and of appropriate dimensions. The matrix valued function $\Delta(t)$ is assumed to be Lebesgue measurable and is defined as

$$\Delta(t) \in \mathbf{U} := \{U \in \mathcal{R}^{k \times l} \mid \|U\| \leq 1\}. \tag{2.56}$$

They show that quadratic stabilization of this system is equivalent to the solution of a Riccati equation. From this solution a stabilizing LTI controller can be constructed.

It is interesting to note that by adding the assumption that uncertainties (or parameters) can be measured on-line then our $\tilde{A}_{22}$ condition is necessary and sufficient to guarantee linear quadratic stabilization. In fact, for this case, quadratic stabilization and linear quadratic stabilization are equivalent.

Bernussou et al. in [21] made different assumptions on the uncertainty in order to guarantee linear quadratic stabilization. He considered the uncertain linear system

$$\dot{x} = Ax + Bu \tag{2.57}$$

where A and B are matrices belonging to uncertainty domains $\mathcal{D}_A$ and $\mathcal{D}_B$ defined by

$$\mathcal{D}_A := \{A \mid A = \sum_{i=1}^{N} \lambda_i A_i, \lambda_i \geq 0, \sum_{i=1}^{N} \lambda_i = 1\} \tag{2.58}$$

$$\mathcal{D}_B := \{B \mid B = \sum_{j=1}^{N} \mu_j B_j, \mu_j \geq 0, \sum_{j=1}^{N} \mu_j = 1\}. \tag{2.59}$$

This corresponds to the case of linear systems where the coefficients are known up to a certain precision defined by bounding them from above and below. This is commonly known as interval plants. In this case the uncertainties are not time-varying. The necessary and sufficient condition derived is a form of $\tilde{A}_{22} < 0$ for interval plants.

$$A_i W + W A_i^T - B_j R - R^T B_j^T < 0 \quad \forall i \, \forall j \tag{2.60}$$

These conditions are LMI's and therefore can be efficiently checked. These conditions not only are for determining the feasibility of a constant Lyapunov function (i.e. quadratic stability) but also for a constant controller. By requiring a fixed controller, the conditions require additional variables (i.e. R) that have to be determined. Now the condition given in (2.60) can be used to determine if a constant controller could be used to quadratically stabilize an LPV system. If this condition were satisfied then gain scheduling would not be needed.

Becker and Packard in [22] and summarized in [23] extend Bernussou's work to a different class of uncertainties. They provide necessary and sufficient conditions for quadratically stabilizing uncertain systems via LTI state-feedback where the uncertainty in the A and B matrices can depend on both a correlated uncertainty and an uncorrelated uncertainty. The uncorrelated uncertainty is a special form of linear fractional uncertainty on real parameters. The correlated uncertainties lie within a prescribed convex polytope while the uncorrelated uncertainty lie within the unit cube having dimension equal to the number of uncorrelated uncertainties. The necessary and sufficient conditions are essentially (2.60) checked at all of the vertices of the uncertainty space.

Concurrent Work in LPV Gain Scheduling

Research in the area of LPV gain scheduling has become a topic of very intense interest as it is applicable to so many areas especially to the area of missile autopilot design. In particular, contemporary with this research, has been the work conducted by Becker, Packard and others who have collaborated with them in their research. Several important papers in this area for which Becker and Packard were primary authors are [6, 16, 24, 22, 23] along with Becker's dissertation [25] which is a summary for most of this work.

In the first part of [6], Becker et al. provide necessary and sufficient conditions for quadratic LPV stabilization via output feedback with a dynamic controller. The LPV plant under consideration is

$$\begin{bmatrix} \dot{x}(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A(\theta) & B(\theta) \\ C(\theta) & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \tag{2.61}$$

The authors state that this plant is quadratically stabilizable via output feedback if and only if the following two conditions hold:

1. there exists a $P_K \in \mathcal{R}^{n \times n}$,$P_K = P_K^T > 0$, and a continuous, bounded function $F : \mathcal{R}^k \rightarrow \mathcal{R}^{n_u \times n}$ such that for all $\theta(t)$ within our parameter set,

$$[A(\theta) - B(\theta)F(\theta)]^T P_K + P_K[A(\theta) - B(\theta)K(\theta)] < 0, \tag{2.62}$$

2. there exists a $P_L \in \mathcal{R}^{n \times n}$,$P_L = P_L^T > 0$, and a continuous, bounded function $L : \mathcal{R}^k \rightarrow \mathcal{R}^{n \times n_y}$ such that for all $\theta(t)$ within our parameter set,

$$[A(\theta) - L(\theta)C(\theta)]^T P_L + P_L[A(\theta) - L(\theta)C(\theta)] < 0. \tag{2.63}$$

If the functions $F(\theta)$ and $L(\theta)$ exist as in the above conditions then

$$K(\theta) = \begin{bmatrix} A(\theta) + B(\theta)F(\theta) - L(\theta)C(\theta) & -L(\theta) \\ F(\theta) & 0 \end{bmatrix} \tag{2.64}$$

is the quadratically stabilizing parameter dependent output-feedback controller expressed in compact matrix form. Further more they proceed to parameterize all stabilizing output-feedback controllers. What Becker et al. do not show in this paper or in Becker's dissertation [25] is how to determine if $F(\theta)$ and $L(\theta)$ exist and if they do exist then how to compute them. Our Theorem 2.2 provides the answer to both the existence and computation of the functions.

In the second part of [6], the authors add $\mathcal{L}_2$ performance to the quadratic stabilization problem. The LPV plant considered is as follows.

$$\begin{bmatrix} \dot{x}(t) \\ q_1(t) \\ q_2(t) \\ y(t) \end{bmatrix} = \left[\begin{array}{c|ccc} A(\theta) & B_1(\theta) & 0 & B_2(\theta) \\ \hline C_1(\theta) & 0 & 0 & 0 \\ 0 & 0 & 0 & I \\ C_2(\theta) & 0 & I & 0 \end{array} \right] \begin{bmatrix} x(t) \\ w_1(t) \\ w_2(t) \\ u(t) \end{bmatrix} = \tag{2.65}$$

where θ is the vector of parameters as defined in (2.2), $x(t) \in \mathcal{R}^n$, $[w_1^T(t),\ w_2^T(t)]^T \in \mathcal{R}^{n_w}$ is the disturbance input, $[q_1^T(t),\ q_2^T(t)]^T \in \mathcal{R}^{n_q}$ is the performance output, $u(t) \in \mathcal{R}^{n_u}$ and $y(t) \in \mathcal{R}^{n_y}$ for all $t > 0$. Some assumptions were made in order to get this form of an LPV plant that are restrictive. In Becker's dissertation [25] these assumptions are relaxed.

Becker et al. state that the open-loop system given in (2.65) is quadratically stabilizable and satisfies an $\mathcal{L}_2$ performance criteria if and only if there exist matrices $X_{11} > 0 \in \mathcal{R}^{n \times n}$ and $Y_{11} > 0 \in \mathcal{R}^{n \times n}$ such that for all θ in the parameter space

$$\begin{gathered}
\begin{bmatrix} A(\theta)Y_{11} + Y_{11}A^T(\theta) - B_2(\theta)B_2^T(\theta) & Y_{11}C_1^T(\theta) & B_1(\theta) \\ C_1(\theta)Y_{11} & -I & 0 \\ B_1^T(\theta) & 0 & -I \end{bmatrix} < 0 \\
\begin{bmatrix} A^T(\theta)X_{11} + X_{11}A(\theta) - C_2^T(\theta)C_2(\theta) & X_{11}B_1(\theta) & C_1^T(\theta) \\ B_1^T(\theta)X_{11} & -I & 0 \\ C_1(\theta) & 0 & -I \end{bmatrix} < 0 \\
\begin{bmatrix} X_{11} & I_n \\ I_n & Y_{11} \end{bmatrix} \geq 0.
\end{gathered} \tag{2.66}$$

Given that these conditions are feasible, the authors then provide formulas to construct an LPV controller. The formulas given are quite involved and are rather awkward. In the second part of [16], Packard et al. improve upon these controller formulas. In this reference, Packard et al. apply the controller formulas from [26] to this LPV problem and obtain controller formulas which are much easier to use. We did discover some typographical errors when comparing the formulas presented in [16] with those of [26]. The solution of the LMI's given in (2.66) require gridding the parameter space and solving an approximation to (2.66). As long as the number of parameters is not too large then this gridding of the space should not be too burdensome computationally but as the number of parameters grow the number of grid points grow exponentially.

In Becker's dissertation he pulls together the works from [6],[24] and [16]. In addition to this he provides necessary and sufficient conditions for quadratic stabilizability for an LPV system which are given in the following Lemma.

Lemma 2.6. *[25] The open-loop system given in (2.1) is quadratically stabilizable over the parameter space if there exists a* $W > 0$ *and a continuous, bounded function* $R : \mathcal{R}^{n_p} \times \mathcal{R} \rightarrow \mathcal{R}^{n_u \times n}$ *such that*

$$A(\theta)W + WA^T(\theta) + B_2(\theta)R(\theta) + R^T(\theta)B_2(\theta) < 0 \tag{2.67}$$

for all θ *in the parameter space.*

Unfortunately Becker does not indicate how to compute the function $R(\theta)$. One method would be to grid the space and have a different variable R_i at each grid point. This would not be practical though due to the extremely large number of variables that would have to be determined. Fortunately the $\tilde{A}_{22}$ condition that we developed determines stabilizability without having to compute a stabilizing controller. It uses not only a reduced number of variables but an LMI that is of reduced dimensions.

2.2 General Lyapunov Stability

Quadratic stability may be too conservative in that it requires a single quadratic Lyapunov function to be satisfied over the entire parameter space. The implication of this single quadratic Lyapunov function is that the stability of the LPV system is not dependent on the time rate of change of the parameters. In fact the parameters can change infinitely fast and the system will remain stable as long as we have knowledge of the current values of the parameters. A less conservative form of stability would be General Lyapunov Stability. General Lyapunov Stability does not hold the Lyapunov

Function constant as the system changes but allows it to vary with the system. In this case, however, knowledge of the maximum time rate of change of the parameters will be necessary to insure stability.

$$|\dot{\theta}_i| \leq \dot{\theta}_{i_{max}}, \quad 1 \leq i \leq n_p. \tag{2.68}$$

The following theorem gives necessary and sufficient conditions for general Lyapunov stability.

Theorem 2.7 (General Lyapunov Stability). *The LPV system defined in (2.1) is stable (in the sense of General Lyapunov Stability) if and only if there exists a differentiable matrix function $P(t) > 0$ with $W(t) = P^{-1}(t)$ such that for all $\theta(t)$*

$$\tilde{A}_{22}(\theta, t) = U_2^T(\theta)(A(\theta)W(t) + W(t)A^T(\theta) + \underbrace{\dot{W}(t)}_{new\ term})U_2(\theta) < 0 \tag{2.69}$$

where the range of $U_2(\theta)$ is the null space of $B^T(\theta)$.

Proof. ($\Longrightarrow$)Given our LPV system as described in (2.1) with parameters defined by (2.2) and having bounds on there time rate of change given by (2.68), for necessity we assume there exists a controller $K(\theta)$ such that the closed-loop system

$$\dot{x}(t) = (A(\theta) - B(\theta)K(\theta))x(t) + B(\theta)r(t) \tag{2.70}$$

is stable. The general Lyapunov function corresponding to (2.70) would be

$$\begin{aligned} V(x, t) &= x(t)^T P(t) x(t) && (2.71) \\ \dot{V}(x, \theta, t) &= -x^T(t)Q(t)x(t) \\ &= x^T(t)(P(t)A(\theta) - P(t)B(\theta)K(\theta) + A^T(\theta)P(t) - \\ &\quad K^T(\theta)B^T(\theta)P(t) + \underbrace{\dot{P}(t)}_{\text{new term}})x(t). && (2.72) \end{aligned}$$

Therefore

$$P(t)A(\theta) - P(t)B(\theta)K(\theta) + A^T(\theta)P(t) - K^T(\theta)B^T(\theta)P(t) + \dot{P}(t) = -Q(t) \quad (2.73)$$

is negative definite.

Proceeding with the proof in the same manner as for the proof of Theorem 2.2 but, in addition, carry along the "new term", results in (2.73) becoming

$$\tilde{A}(\theta) - \Sigma(\theta)L(\theta) - L^T(\theta)\Sigma^T(\theta) = -\tilde{Q}(t). \quad (2.74)$$

where

$$\tilde{A}(\theta) = U^T(\theta)\left(A(\theta)W(t) + W(t)A^T(\theta) + \underbrace{\dot{W}}_{\text{new term}}\right)U(\theta) \quad (2.75)$$

$$L(\theta) = V^T(\theta)K(\theta)W(t)U(\theta) \quad (2.76)$$

$$\tilde{Q}(t) = U^T(\theta)W(t)Q(t)W(t)U(\theta). \quad (2.77)$$

Now partition $U(\theta)$ and $\Sigma(\theta)$ as

$$U(\theta) = [U_1(\theta) \;\; U_2(\theta)], \quad \Sigma(\theta) = \begin{bmatrix} \Sigma_1(\theta) \\ 0 \end{bmatrix}. \quad (2.78)$$

where $U_1(\theta) \in \mathcal{R}^{n\times n_u}$, $U_2(\theta) \in \mathcal{R}^{n\times(n-n_u)}$, $\Sigma_1(\theta) = \text{diag}[\sigma_{11}(\theta),\dots,\sigma_{n_u n_u}(\theta)] \in \mathcal{R}^{n_u\times n_u}$. This will partition $\tilde{A}(\theta)$ as follows

$$\tilde{A}(\theta) = \begin{bmatrix} \tilde{A}_{11}(\theta) & \tilde{A}_{12}(\theta) \\ \tilde{A}_{12}^T(\theta) & \tilde{A}_{22}(\theta) \end{bmatrix} \quad (2.79)$$

where

$$\begin{aligned}
\tilde{A}_{11}(\theta) &:= U_1^T(\theta)\left(A(\theta)W(t) + W(t)A^T(\theta) + \underbrace{\dot{W}}_{\text{new term}}\right)U_1(\theta) = \tilde{A}_{11}^T(\theta) \in \mathcal{R}^{n_u \times n_u} \\
\tilde{A}_{12}(\theta) &:= U_1^T(\theta)\left(A(\theta)W(t) + W(t)A^T(\theta) + \underbrace{\dot{W}}_{\text{new term}}\right)U_2(\theta) \in \mathcal{R}^{n_u \times (n-n_u)} \\
\tilde{A}_{22}(\theta) &:= U_2^T(\theta)\left(A(\theta)W(t) + W(t)A^T(\theta) + \underbrace{\dot{W}}_{\text{new term}}\right)U_2(\theta) \\
&= \tilde{A}_{22}^T(\theta) \in \mathcal{R}^{(n-n_u) \times (n-n_u)}
\end{aligned} \tag{2.80}$$

Partition $L(\theta)$ as

$$L(\theta) = [L_{11}(\theta) \;\; L_{12}(\theta)], \tag{2.81}$$

Substituting $\tilde{A}(\theta)$, $\Sigma(\theta)$ and $L(\theta)$ from (2.79), (2.78) and (2.81) into (2.74), we obtain

$$\begin{bmatrix} \tilde{A}_{11}(\theta) - (\Sigma_1(\theta)L_{11}(\theta) + L_{11}^T(\theta)\Sigma_1(\theta)) & \tilde{A}_{12}(\theta) - \Sigma_1(\theta)L_{12}(\theta) \\ \tilde{A}_{12}^T(\theta) - L_{12}^T(\theta)\Sigma_1(\theta) & \tilde{A}_{22}(\theta) \end{bmatrix} = -\tilde{Q}(t). \tag{2.82}$$

Since $U(\theta)$ and $W(t)$ are nonsingular, $\tilde{Q}(t) < 0$ if and only if $Q(t) < 0$ which is our assumption. Multiplying (2.82) from the left and on the right by $[0\ x^T]$ and $[0\ x^T]^T$, respectively where x is any vector in $\mathcal{R}^{n \times n_{n_u}}$, we conclude that $\tilde{A}_{22}$ is negative definite.

($\Longleftarrow$) Now suppose that there exists a differentiable function $W(t) > 0$ such that $\tilde{A}_{22}(\theta) < 0$, from (2.76), we have free choice of $L(\theta)$. Therefore looking at (2.82), if you choose the elements of $L_{11}(\theta) = [l_{ij}(\theta)] \in \mathcal{R}^{n_u \times n_u}$ as

$$l_{ij}(\theta) = \frac{\tilde{a}_{ij}(\theta) - \sigma_{jj}(\theta)l_{ji}(\theta)}{\sigma_{ii}(\theta)}, \;\; l_{ji}(\theta) = \text{ your choice } \; i = 1, \ldots n_u - 1, \; j = i+1, \ldots n_u \tag{2.83}$$

$$l_{ii}(\theta) = \frac{\tilde{a}_{ii}(\theta) + d_{ii}}{2\sigma_{ii}(\theta)}, \; i = 1, \ldots n_u \tag{2.84}$$

where $dii > 0$ are arbitrary real numbers for all $i = 1, \ldots n_u$, and $\tilde{a}_{ij}$ are the elements of $\tilde{A}$ and, in addition, choose the matrix $L_{12}(\theta) \in \mathcal{R}^{n_u \times n - n_u}$ to be

$$L_{12}(\theta) = \Sigma_1^{-1}(\theta)U_1^T(A(\theta)W(t) + W(t)A(\theta)^T + \underbrace{\dot{W}}_{\text{new term}})U_2(\theta), \tag{2.85}$$

then we obtain $-\tilde{Q}(t) = \text{diag}[-d_{11}, \ldots - d_{n_u n_u}, \ \tilde{A}_{22}(\theta)]$ which is negative definite. □

Lemma 2.8. *If there exists a differentiable matrix $W(t) > 0$ with $P(t) = W^{-1}(t)$ such that $\tilde{A}_{22}(\theta, t) < 0$, then a stabilizing controller, $K(\theta)$, can be chosen as*

$$K(\theta) = V(\theta)L(\theta)]U(\theta)^T P(t) \tag{2.86}$$

where $L(\theta) = [L_{11}(\theta)L_{12}(\theta)]$ with the elements of $L_{11}(\theta) = [l_{ij}(\theta)] \in \mathcal{R}^{n_u \times n_u}$ are

$$l_{ij}(\theta) = \begin{cases} \frac{\tilde{a}_{ij}(\theta) - \sigma_{jj}(\theta)}{\sigma_{ii}(\theta)} & i < j, \\ \frac{\tilde{a}_{ii}(\theta) + d_{ii}(\theta)}{2\sigma_{ii}(\theta)} & i = j \\ 1 & i > j \end{cases} \tag{2.87}$$

where $dii > 0$ are arbitrary real numbers for all $i = 1, \ldots n_u$ and $\tilde{a}_{ij}$ are the elements of $\tilde{A}$. The matrix $L_{12}(\theta) \in \mathcal{R}^{n_u \times n - n_u}$ is

$$L_{12}(\theta) = \Sigma_1^{-1}(\theta)U_1^T(A(\theta)W(t) + W(t)A(\theta)^T + \underbrace{\dot{W}}_{\textit{new term}})U_2(\theta). \tag{2.88}$$

Proof. Assuming that a differentiable $W(t) > 0$ exists such that $\tilde{A}_{22}(\theta, t) < 0$, then from the proof of Theorem 2.7 we see that the particular choice of $L(\theta) = \begin{bmatrix} L_{11}(\theta) & L_{12}(\theta) \end{bmatrix}$ from (2.87) and (2.88) will make the system stabilizable. Recall (2.76)

$$L(\theta) = V^T(\theta)K(\theta)W(t)U(\theta).$$

Solving for $K(\theta)$ yields

$$K(\theta) = V(\theta)L(\theta)U^T(\theta)P(t)$$

which is our stated controller. □

With the addition of the "new term", (2.69) becomes rather difficult to solve since it is a function of both W and $\dot{W}$. In order to solve (2.69) we will again have to result to solving an approximation by girding the parameter space and solving for several single quadratic Lyapunov functions such that each Lyapunov function covers a region of the parameter space. The intention is to solve for as few Lyapunov functions as possible. The most desirable situation would be to find a single quadratic Lyapunov function that is valid over the entire parameter space. If this is the case then (2.69) reduces to the single quadratic Lyapunov case (2.5). The controller $K(\theta)$ given in (2.86) also reduces to the single quadratic Lyapunov controller of (2.27). For the case where a single quadratic Lyapunov function does not exist, we will divide the parameter space into several regions with a single quadratic Lyapunov function for each region. The controller will then be constructed as before except that the Lyapunov function used to construct the controller will change depending on the location of the LPV plant in the parameter space. There is a paradigm shift here. This is analogous to the classical gain scheduled method of solving for frozen coefficient controllers and then interpolating the controllers as the parameters change, instead, now we will solve for frozen region Lyapunov functions and interpolate these Lyapunov functions to get the current Lyapunov function which will then be used to compute the controller gains as given in (2.86). The following is an algorithm to solve (2.69).

Algorithm 2.9.

1. *Determine the feasibility of (2.69) with a single quadratic Lyapunov function (i.e. assume $\dot{W} = 0$). If feasible then implement the controller as given by (2.86) with a fixed W.*

2. *If infeasible, divide the parameter space into two regions. Then determine the feasibility of (2.69) with two W's. For example, determine the feasibility of*

$$U_2^T(\theta)(A(\theta)W_1 + W_1A^T(\theta) + \Delta)U_2(\theta) < 0 \quad \forall\theta \in \textit{Region 1}$$
$$U_2^T(\theta)(A(\theta)W_2 + W_2A^T(\theta) + \Delta)U_2(\theta) < 0 \quad \forall\theta \in \textit{Region 2}$$

   with $W_1, W_2 > 0$ where Δ is a positive definite matrix that allows for $\dot{W} \neq 0$. If feasible, proceed to the next step. Otherwise, continue dividing the parameter space, thereby adding more frozen region Lyapunov functions, until there is a solution.

3. *Upon finding a solution, it becomes necessary to check (2.69). This check requires knowledge of $\dot{W}$. The value of $\dot{W}$ can be approximated as follows,*

$$\dot{W} = \frac{W_{current} - W_{previous}}{\Delta t} \tag{2.89}$$

   where $W_{current}$ is the current interpolated value of $W(t)$ and $W_{previous}$ is the previous value of $W(t)$. Unfortunately, $W_{previous}$ may posses many different values. However, from the knowledge of the bounds on $\dot{\theta}$ as given in (2.68), we can determine the regions in the parameter space that can reach our current location in one Δt. From these regions we can compute the various values of $\dot{W}$, allowing us to check (2.69) at each grid point. If (2.69) is satisfied at each grid point then the controller (2.86) can be implemented with the interpolated values of $W(t)$.

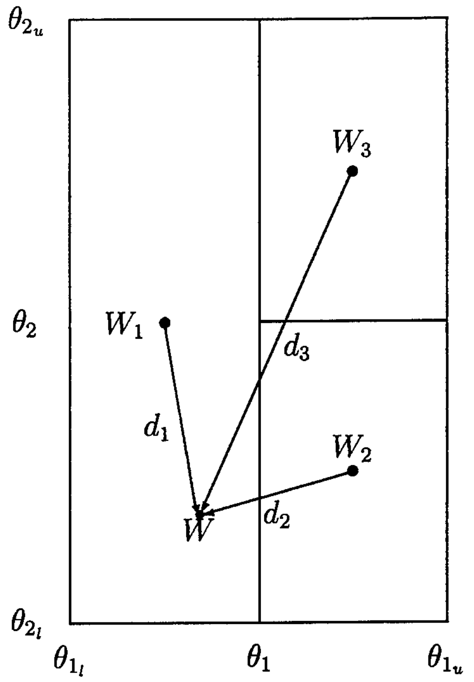

Figure 2.10: An example on how to interpolate the Lyapunov Functions.

An advantage of Algorithm 2.9 is that it initially tries to find a single quadratic Lyapunov function and then if this is not possible it naturally converts to the General Lyapunov case.

One method of interpolating the W's is to consider the distance from the current location in the parameter space to the center of each region that can be reached from this location in one Δt. It is assumed that the center of each region will have as its Lyapunov function the W that is associated with that region. As an example, consider the two parameter space as depicted in Figure 2.10. The current W can be interpolated as

$$W = \left(\frac{1}{1+\frac{d_1}{d_2}+\frac{d_1}{d_3}}\right)W_1 + \left(\frac{1}{1+\frac{d_2}{d_1}+\frac{d_2}{d_3}}\right)W_2 + \left(\frac{1}{1+\frac{d_3}{d_1}+\frac{d_3}{d_2}}\right)W_3 \qquad (2.90)$$

with d_1, d_2 and d_3 as shown in Figure 2.10. To insure that this W is a valid Lyapunov function we need to check

$$U_2^T(\theta)(A(\theta)W + WA^T(\theta) + \Delta)U_2(\theta) < 0 \qquad (2.91)$$

where Δ is the same as the one used when computing the frozen Lyapunov regions. If it is not satisfied then solve the following LMI's

$$\begin{aligned} U_2^T(\theta)(A(\theta)(W+\Delta W)+(W+\Delta W)A^T(\theta)+\Delta)U_2(\theta) &< 0 \\ W+\Delta W &> 0 \end{aligned} \tag{2.92}$$

with objective function

$$f_{obj} = trace(\Delta W) \tag{2.93}$$

The objective function is used to keep the new W as close to the original interpolated W as possible. Thereafter,

$$W_{new} = W + \Delta W. \tag{2.94}$$

If Step 3 of Algorithm 2.9 fails, then before coming to the conclusion that the system is uncontrollable the control engineer can try the following. Recall that when we solved for the current regional W_i that we used a fixed Δ for the entire parameter space. Try solving (2.91) with a new Δ for the offending locations. This may mean setting up a new region inside an existing region. If a larger Δ is possible, it will allow for a larger $\dot{W}$.

The concept now is not to grid the space with frozen point controllers but, instead, to grid the space with frozen point Lyapunov functions. These Lyapunov functions can be stored in tables as previously the controllers were stored and then interpolated as the plant changes with the parameters. The controller scheduling law is given by (2.86). This method of scheduling will guarantee general Lyapunov stability.

So far only stability has been investigated, now the addition of performance will be investigated.

2.3 Performance

In this section the concept of guaranteeing performance via state-feedback subject to quadratic stabilization is discussed. Several methods will be explored including minimizing a cost function, minimizing the output energy and finally guaranteeing $\mathcal{L}_2$-performance for a strictly proper system. A missile autopilot design example will be given to demonstrate guaranteeing $\mathcal{L}_2$

2.3.1 Minimizing a Cost Function

One method of introducing performance into the design of LPV systems is to design a controller such that the resulting closed-loop system minimizes a given cost function. The cost function that we will use is the standard cost function used in control system design

$$J = \int_0^\infty (x^T Q(\theta) x + u^T R(\theta) u) dt \tag{2.95}$$

where $Q = L^T L \geq 0$ and $R(\theta) > 0$ for all θ in the parameter space. By allowing the cost function to change with the parameters, allows the control designer to set different performance objectives depending on where in the parameter space the system is located. For example, a missile autopilot designer for a ballistic missile will want to reduce performance at apogee when the dynamic pressure is low and will want increased performance during the end-game when the dynamic pressure has come back up. Thereom 2.10 describes how to design such a controller.

Theorem 2.10. *Given the LPV system as described by* (2.1) *with the magnitude of the parameters being bounded and initial condition x_0, there exists a controller*

$$K = R^{-1} B^T(\theta) P \tag{2.96}$$

that minimizes the cost function $J = \int_0^\infty (x^T Q(\theta) x + u^T R(\theta) u) dt$ subject to quadratic stabilization if and only if there exists a $W = P^{-1} > 0$ such that the following LMI *problem is solvable for all θ in the parameter space.*

$$\text{maximize} \quad x_0^T(W)x_0$$

subject to

$$\begin{bmatrix} A(\theta)W + WA^T(\theta) - B(\theta)R^{-1}(\theta)B^T(\theta) & WL^T(\theta) \\ L(\theta)W & -\mathrm{I} \end{bmatrix} \leq 0 \tag{2.97}$$

$$W > 0$$

Proof. The idea is to bound the cost function with a Lyapunov function. Suppose there exists a quadratic Lyapunov function $V(t) = x^T(t)Px(t)$ such that

$$P > 0, \quad \frac{d}{dt}V(x(t)) \leq -(x^TQx + u^TRu) \tag{2.98}$$

then integrating both sides from $t = 0$ to $t = T$ yields

$$V(x(T)) - V(x_0) \leq -\int_0^T (x^TQx + u^TRu)dt. \tag{2.99}$$

Since $V(x(T)) > 0$, we have

$$V(x_0) \geq \int_0^T (x^TQx + u^TRu)dt. \tag{2.100}$$

This equation shows that the cost function can be bounded from above by $x_0^TPx_0$. From (2.98), we can derive an equivalent condition that will be used to prove the theorem. Expanding out $\frac{d}{dt}V(x)$ with $u = -Kx$, (2.98) becomes

$$x^T(A - BK)^TPx + x^TP(A - BK)x \leq -(x^TQx + x^TK^TRKx)$$

Then collecting terms it becomes

$$x^T((A - BK)^TP + P(A - BK) + Q + K^TRK)x \leq 0. \tag{2.101}$$

Therefore, if there exists a P such that

$$P > 0, \quad (A - BK)^T P + P(A - BK) + Q + K^T RK \leq 0, \tag{2.102}$$

then an upper bound on the cost function is $x_0^T P x_0$. Equivalently, if $W = P^{-1}$, then (2.102) becomes

$$W > 0, \quad W(A - BK)^T + (A - BK)W + WQW + WK^T RKW \leq 0 \tag{2.103}$$

and the corresponding upper bound on the cost function is $x_0^T W^{-1} x_0$.

The existance of a controller, K, subject to quadratic stabilization that minimizes the given cost function is equivalent to (2.103) having a solution such that $x_0^T W x_0$ is maximized. Letting $Y = KW$, the QMI in (2.103) becomes

$$WA^T + AW - Y^T B^T - BY + WQW + Y^T RY \leq 0 \tag{2.104}$$

$\Longleftrightarrow$ (Letting $R = S^T S$ and by completing the square)

$$WA^T + AW + WQW - BR^{-1}B^T + (SY - (S^T)^{-1}B^T)^T (SY - (S^T)^{-1}B^T) \leq 0 \tag{2.105}$$

$\Longleftrightarrow$ (Since $Y = KW = R^{-1}B^T$, the completed square term = 0)

$$WA^T + AW + WQW - BR^{-1}B^T \leq 0 \tag{2.106}$$

$\Longleftrightarrow$ (Via Schur Complements)

$$\begin{bmatrix} AW + WA^T - BR^{-1}B^T & WL^T \\ LW & -\mathrm{I} \end{bmatrix} \leq 0 \tag{2.107}$$

which completes the proof. □

As an example to demonstrate Theorem 2.10, a controller will be designed for the open-loop system from Example 2.5 such that a given cost function is minimized.

Example 2.11. *Recall the open-loop LPV plant from Example 2.5*

$$\dot{x} = \begin{bmatrix} -1 + 1.5\cos^2(t) & 1 - 1.5\sin(t)\cos(t) \\ 1 & -1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \tag{2.108}$$

with initial condition $x_0 = \left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$. *We desire to design a controller that minimizes the cost function*

$$J = \int_0^\infty (x^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x + u^T(.1)u)dt. \tag{2.109}$$

Gridding the parameter space as before into 20 points and solving (2.97) *for* $W = P^{-1}$ *yields*

$$P = \begin{bmatrix} 9.59073 & 1.94165 \\ 1.94165 & 0.78845 \end{bmatrix}. \tag{2.110}$$

Figure 2.11 shows the response of the system to the initial conditions. This controller quickly drives the states to zero yielding a cost of

$$J = 4.28$$

As a point of comparison if we applied this same cost function to the closed-loop system constructed using the Lyapunov based controller from Example 2.5, the cost would be

$$J = 11.89$$

Therefore, as expected the cost is lower for our controller designed with performance taken into account.

□

Another method of interest in integrating performance into the Lyapunov based gain scheduling is to minimize the output energy. The next section covers this method.

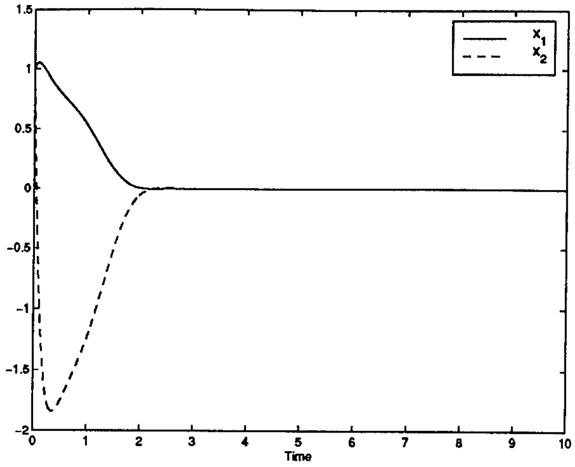

Figure 2.11: The response of the system using the Performance based Lyapunov controller with inititial condition $x_0 = [1\ 1]^T$.

2.3.2 Minimizing the Output Energy

In this section we are interested in bounding the output energy of a system. To do this we extend the results taken from Boyd et al. [27] for LTI systems, where $D_{yu}^T C_y = 0$, to the LPV systems with no constraints on $D_{yu}^T C_y$.

Given a certain initial state, x_0, consider bounding the output energy of the closed-loop system

$$\begin{aligned} \dot{x} &= A_{cl}(\theta)x \\ y &= C_{cl_y}(\theta)x \end{aligned} \tag{2.111}$$

where the output energy is

$$E_{output} = \int y^T(t)y(t)dt. \tag{2.112}$$

Suppose there exists a quadratic Lyapunov function $V(t) = x^T(t)Px(t)$ such that

$$P > 0, \quad \frac{d}{dt}V(x(t)) < -y^T(t)y(t) \tag{2.113}$$

then integrating both sides from $t = 0$ to $t = T$ yields

$$V(x(T)) - V(x_0) < -\int_0^T y^T(t)y(t)dt. \tag{2.114}$$

Since $V(x(T)) > 0$, we have

$$V(x_0) > \int_0^T y^T(t)y(t)dt. \tag{2.115}$$

This equation shows that the output energy can be bounded from above by $x_0^T P x_0$. From (2.113), we can derive an equivalent condition that will be used to solve the LPV output energy minimization problem. Expanding out $\frac{d}{dt}V(x)$ and inserting (2.111) into (2.113) yields

$$x^T A_{cl}^T P x + x^T P A_{cl} x < -x^T C_{cl_y}^T C_{cl_y} x \tag{2.116}$$

Then collecting terms

$$x^T (A_{cl}^T P + P A_{cl} + C_{cl_y}^T C_{cl_y}) x < 0. \tag{2.117}$$

Therefore, if there exists a P such that

$$P > 0, \quad A_{cl}^T P + P A_{cl} + C_{cl_y}^T C_{cl_y} < 0, \tag{2.118}$$

then an upper bound on the output energy is $x_0^T P x_0$. Equivalently, if $W = P^{-1}$, then (2.118) becomes

$$W > 0, \quad W A_{cl}^T + A_{cl} W + W C_{cl_y}^T C_{cl_y} W < 0. \tag{2.119}$$

Condition (2.119) is equivalent via Schur complements to

$$W > 0, \quad \begin{bmatrix} A_{cl} W + W A_{cl}^T & W C_{cl_y}^T \\ C_{cl_y} W & -I \end{bmatrix} < 0 \tag{2.120}$$

Therefore, if W satisfies (2.120) then an upper bound on the output energy is $x_0^T W^{-1} x_0$.

Now take this analysis result and apply it to designing an LPV controller, $K(\theta)$, for the LPV system

$$\begin{aligned} \dot{x} &= A(\theta)x + B_u(\theta)u \\ y &= C_y(\theta)x + D_{yu}(\theta)u \end{aligned} \tag{2.121}$$

with $u = K(\theta)x$ such that we minimize the output energy of the resulting closed-loop system

$$\begin{aligned} \dot{x} &= (A(\theta) + B_u(\theta)K(\theta))x \\ y &= (C_y(\theta) + D_{yu}(\theta)K(\theta))x \end{aligned} \tag{2.122}$$

The following theorem describes how to do this.

Theorem 2.12. *Given an initial condition x_0 for the LPV system in (2.121) and that $(A(\theta), B(\theta), C_y(\theta))$ is minimal; $D_{yu}^T(\theta)D_{yu}(\theta)$ is invertible for all θ in the parameter space and $(\mathrm{I} - D_{yu}(D_{yu}^T D_{yu})^{-1} D_{yu}^T) = E^T E$, there exists a controller*

$$K(\theta) = -(D_{yu}^T D_{yu})^{-1}(B_u^T W^{-1} + D_{yu}^T C_y) \tag{2.123}$$

that minimizes the output energy $\int_0^\infty y^T y dt$ subject to Quadratic stabilization if and only if there exists a $W > 0$ such that the following Quadratic Matrix Inequality

(QMI) problem is solvable for all θ in the parameter space.

$$\begin{aligned} &\text{maximize} \quad x_0^T W x_0 \\ &\text{subject to} \end{aligned}$$

$$\begin{bmatrix} \begin{matrix} AW + WA^T - B_u(D_{yu}^T D_{yu})^{-1}B_u^T \\ \quad - WC_y^T D_{yu}(D_{yu}^T D_{yu})^{-1}B_u^T \\ \quad - B_u(D_{yu}^T D_{yu})^{-1}D_{yu}^T C_y W \end{matrix} & WC_y^T E^T \\ EC_y W & -\mathrm{I} \end{bmatrix} < 0 \tag{2.124}$$

$$W > 0$$

Proof. The existance of a controller, K,for the given system that minimizes the output energy subject to quadratic stabilization is equivalent to (2.120) having a solution such that $x_0^T W x_0$ is maximized. Substituting A_{cl}, B_{cl}, C_{cl} from (2.122) into (2.120) and letting $Y = KW$ the matrix constraint becomes

$$\begin{bmatrix} AW + WA^T + B_u Y + Y^T B_u^T & (C_y W + D_{yu} Y)^T \\ (C_y W + D_{yu} Y) & -I \end{bmatrix} < 0, \tag{2.125}$$

$\Longleftrightarrow$ (Via Schur complements)

$$AW + WA^T + B_u Y + Y^T B_u^T + (C_y W + D_{yu} Y)^T (C_y W + D_{yu} Y) < 0 \tag{2.126}$$

$\Longleftrightarrow$ (Multiplying out and collecting terms)

$$\begin{aligned} &AW + WA^T + WC_y^T C_y W + Y^T D_{yu}^T D_{yu} Y \\ &\qquad + (WC_y^T D_{yu} + B_u) Y + Y^T (B_u^T + D_{yu}^T C_y W) < 0 \end{aligned} \tag{2.127}$$

$\Longleftrightarrow$ (By completing the square)

$$\begin{aligned} &AW + WA^T + WC_y^T C_y W - (B_u^T + D_{yu}^T C_y W)^T (D_{yu}^T D_{yu})^{-1} (B_u^T + D_{yu}^T C_y W) \\ &\qquad + (D_{yu} Y + D_{yu}(D_{yu}^T D_{yu})^{-1}(B_u^T + D_{yu}^T C_y W))^T \\ &\qquad\qquad (D_{yu} Y + D_{yu}(D_{yu}^T D_{yu})^{-1}(B_u^T + D_{yu}^T C_y W)) < 0 \end{aligned} \tag{2.128}$$

$\Longleftrightarrow$ (Since $Y = KW = -K(\theta) = -(D_{yu}^T D_{yu})^{-1}(B_u^T + D_{yu}^T C_y W))$, the completed square term = 0)

$$AW + WA^T + WC_y^T C_y W - (B_u^T + D_{yu}^T C_y W)^T (D_{yu}^T D_{yu})^{-1} (B_u^T + D_{yu}^T C_y W) \tag{2.129}$$

$\Longleftrightarrow$ (Expanding and collecting terms)

$$\begin{aligned} &AW + WA^T - B_u(D_{yu}^T D_{yu})^{-1} B_u^T - WC_y^T D_{yu}(D_{yu}^T D_{yu})^{-1} B_u^T \\ &\quad - B_u(D_{yu}^T D_{yu})^{-1} D_{yu}^T C_y W + WC_y^T(\mathrm{I} - D_{yu}(D_{yu}^T D_{yu})^{-1} D_{yu}^T) C_y W < 0 \end{aligned} \tag{2.130}$$

$\Longleftrightarrow$ (Via Schur Complements and letting $(\mathrm{I} - D_{yu}(D_{yu}^T D_{yu})^{-1} D_{yu}^T) = E^T E$ since it is always positive semidefinite.)

$$\begin{bmatrix} AW + WA^T - B_u(D_{yu}^T D_{yu})^{-1} B_u^T & \\ \quad - WC_y^T D_{yu}(D_{yu}^T D_{yu})^{-1} B_u^T & WC_y^T E^T \\ \quad - B_u(D_{yu}^T D_{yu})^{-1} D_{yu}^T C_y W & \\ EC_y W & -\mathrm{I} \end{bmatrix} < 0 \tag{2.131}$$

which completes the proof. □

Finally we will look at adding $\mathcal{L}_2$ performance in the next section.

2.3.3 Guaranteeing $\mathcal{L}_2$-Performance

$\mathcal{L}_2$-Performance criteria have become a very popular way to specify performance. As we begin to investigate designing Lyapunov based gain scheduled controllers utilizing $\mathcal{L}_2$-performance criteria it is necessary to define the induced $\mathcal{L}_2$-norm and $\mathcal{L}_2$-performance for LPV systems. The following definition and $\mathcal{L}_2$-performance Lemma are taken from [25].

Definition 2.13 (Induced $\mathcal{L}_2$-Norm for an LPV System [25]). *Given a quadratically stable LPV system $\mathcal{G}_\theta$ with input u and output y, for zero initial conditions,* *define*

$$\|\mathcal{G}_\theta\|_{ind,2} := \sup_{\theta \in \substack{\text{parameter} \\ \text{space}}} \; \sup_{\substack{\|u\| \neq 0 \\ u \in \mathcal{L}_2}} \frac{\|y\|_2}{\|u\|_2} \tag{2.132}$$

to be the induced $\mathcal{L}_2$-norm.

Therefore, the induced $\mathcal{L}_2$-norm for an LPV system is the largest input to output induced $\mathcal{L}_2$-norm over the set of all causal linear operators described by the LPV system. Now $\mathcal{L}_2$-performance can be quantified as follows.

Lemma 2.14 ($\mathcal{L}_2$-Performance Lemma [25]). *Consider the strictly proper LPV system $\mathcal{G}(\theta,\ A_{cl},\ B_{cl},\ C_{cl},\ 0)$. If there exists a $P > 0 \in \mathcal{R}^{n \times n}$, such that for all θ in the parameter space,*

$$A_{cl}^T(\theta)P + PA_{cl}(\theta) + \frac{1}{\gamma^2}C_{cl}^T(\theta)C_{cl}(\theta) + PB_{cl}(\theta)B_{cl}^T(\theta)P < 0, \tag{2.133}$$

then

1. *The function $A_{cl}(\theta)$ is quadratically stable over the parameter space.*

2. *There exists a $\beta < \gamma$ such that $\|\mathcal{G}_\theta\|_{ind,2} < \beta$*

Proof. See [25]. □

Lemma 2.14 is an analysis tool. That is given an LPV system and that (2.133) is satisfied, we can say that the system is quadratically stable and has a certain $\mathcal{L}_2$-performance associated with it. As control designers, we wish to design controllers for LPV systems such that we can guarantee $\mathcal{L}_2$-performance over the parameter space. To guarantee $\mathcal{L}_2$-performance for a strictly proper system we have the following theorem.

Theorem 2.15. *Consider the open-loop LPV system $\mathcal{G}(\theta, A, B, C, 0)$. There exists a state-feedback controller $K(\theta)$, such that the resulting closed-loop system $\mathcal{G}_{cl}(\theta,\ A_{cl},\ B_{cl},\ C_{cl}, 0)$ is quadratically stable and has an induced $\mathcal{L}_2$-norm*

$$\|\mathcal{G}_{\theta_{cl}}\|_{ind,2} < \gamma \tag{2.134}$$

if there exists a $W > 0$ such that

$$\begin{bmatrix} U_2^T(\theta)(A(\theta)W + WA^T(\theta))U_2^T(\theta) & \frac{1}{\gamma}U_2^T(\theta)WC^T(\theta) \\ \frac{1}{\gamma}C(\theta)WU_2(\theta) & -I \end{bmatrix} < 0, \tag{2.135}$$

where the range of $U_2(\theta)$ is the null space of $B(\theta)$.

Proof. Assume that there exists $W > 0$ such that (2.135) holds. The strictly proper open-loop system under consideration is given by

$$\begin{aligned} \dot{x} &= A(\theta)x + B(\theta)u \\ y &= C(\theta)x \end{aligned} \tag{2.136}$$

The state-feedback control law $u = r - K(\theta)x$ results in the following closed-loop system.

$$\begin{aligned} \dot{x} &= \underbrace{(A(\theta) - B(\theta)K(\theta))}_{A_{cl}}x + \underbrace{B(\theta)}_{B_{cl}}r \\ y &= \underbrace{C(\theta)}_{C_{cl}}x. \end{aligned} \tag{2.137}$$

Then to insure $\mathcal{L}_2$-performance and quadratic stability of the closed-loop system as stated in Lemma 2.14 we need to find a $P > 0$ and $K(\theta)$ such that

$$(A - BK)^T P + P(A - BK) + \frac{1}{\gamma^2}C^T C + PBB^T P < 0. \tag{2.138}$$

Note that for simplicity the specific dependence on θ has been dropped from the notation. As was done previously, substitute in for B its singular value decomposition,

$U\Sigma V^T$, and pre- and post-multiply (2.138) by U^TW and WU where $W = P^{-1}$. This will yield

$$\tilde{A} - \Sigma L - L^T\Sigma^T + \tilde{Q}_{desired} < 0 \tag{2.139}$$

where

$$\tilde{A} = U^T(AW + WA^T)U \tag{2.140}$$

$$L = V^TKWU \tag{2.141}$$

$$\tilde{Q}_{desired} = \frac{1}{\gamma^2}U^TWC^TCWU + \Sigma\Sigma^T. \tag{2.142}$$

Partition $L = [L_{11},\ L_{12}]$, $U = [U_1 \quad U_2]$, $\Sigma = \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix}$, where $U_1 \in \mathcal{R}^{n \times n_u}$, $U_2 \in \mathcal{R}^{n\times(n-n_u)}$ and $\Sigma_1 = \text{diag}[\sigma_{11}, \cdots, \sigma_{n_un_u}] \in \mathcal{R}^{n_u \times n_u}$. This will partition $\tilde{A}$ as follows

$$\tilde{A} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{12}^T & \tilde{A}_{22} \end{bmatrix} \tag{2.143}$$

where

$$\begin{aligned} \tilde{A}_{11} &:= U_1^T(AW + WA^T)U_1 = \tilde{A}_{11}^T \in \mathcal{R}^{n_u \times n_u} \\ \tilde{A}_{12} &:= U_1^T(AW + WA^T)U_2 \in \mathcal{R}^{n_u \times (n-n_u)} \\ \tilde{A}_{22} &:= U_2^T(AW + WA^T)U_2 = \tilde{A}_{22}^T \in \mathcal{R}^{(n-n_u)\times(n-n_u)}. \end{aligned} \tag{2.144}$$

Then (2.139) becomes

$$\begin{bmatrix} \begin{array}{c} \tilde{A}_{11} - (\Sigma_1 L_{11} + L_{11}^T\Sigma_1^T) \\ + \frac{1}{\gamma^2}U_1^TWC^TCWU_1 + \Sigma_1\Sigma_1^T \end{array} & \tilde{A}_{12} - \Sigma_1 L_{12} + \frac{1}{\gamma^2}U_1^TWC^TCWU_2 \\ \tilde{A}_{12}^T - L_{12}^T\Sigma_1 + \frac{1}{\gamma^2}U_2^TWC^TCWU_1 & \tilde{A}_{22} + \frac{1}{\gamma^2}U_2^TWC^TCWU_2 \end{bmatrix} < 0. \tag{2.145}$$

From (2.141),we have free choice of L. Therefore if you choose the elements of $L_{11}(\theta) = [l_{ij}(\theta)] \in \mathcal{R}^{n_u \times n_u}$ to be

$$l_{ij} = \frac{\tilde{a}_{ij} - \sigma_{jj} l_{ji} + \tilde{q}_{desired_{ij}}}{\sigma_{ii}}, \quad l_{ji} = \text{ your choice } \; i = 1, \ldots n_u - 1, \; j = i+1, \ldots n_u \tag{2.146}$$

$$l_{ii} = \frac{\tilde{a}_{ii} + \tilde{q}_{desired_{ij}} + d_{ii}}{2\sigma_{ii}}, \; i = 1, \ldots n_u \tag{2.147}$$

where $d_{ii} > 0$ are arbitrary real numbers for all $i = 1, \ldots n_u$, $\tilde{A} = [\tilde{a}_{ij}]$,and $\tilde{Q}_{desired} = [\tilde{q}_{desired_{ij}}(\theta)]$. In addition, choose the matrix $L_{12}(\theta) \in \mathcal{R}^{n_u \times n - n_u}$ to be

$$L_{12}(\theta) = \Sigma_1^{-1}(\theta)\{U_1^T(\theta)(A(\theta)W + WA^T(\theta))U_2(\theta) + \tilde{Q}_{desired}(1\colon n_u, n_u + 1\colon n)\} \tag{2.148}$$

then equation (2.145) becomes

$$\begin{bmatrix} -d_{11} & & & & \\ & -d_{22} & & & \\ & & \ddots & & \\ & & & -d_{n_u n_u} & \\ & & & & U_2^T(AW + WA^T)U_2 + \frac{1}{\gamma^2} U_2^T W C^T C W U_2 \end{bmatrix} < 0$$

which is negative definite if and only if

$$U_2^T(AW + WA^T)U_2 + \frac{1}{\gamma^2} U_2^T W C^T C W U_2 < 0.$$

From the Schur complement this matrix is negative definite if and only if

$$\begin{bmatrix} U_2^T(AW + WA^T)U_2^T & \frac{1}{\gamma} U_2^T W C^T \\ \frac{1}{\gamma} C W U_2 & -I \end{bmatrix} < 0. \tag{2.149}$$

Since we originally assumed that there existed a $W > 0$ such that this matrix was negative definite and recalling that $P = W^{-1}$, Lemma 2.14 is satisfied thereby guaranteeing that the closed-loop system will be quadratically stable and have an induced $\mathcal{L}_2$-norm less than γ. □

Lemma 2.16. *If there exists a $W > 0$ with $P = W^{-1}$ such that (2.135) is satisfied, then a stabilizing controller, $K(\theta)$, that yields a closed-loop system with an induced $\mathcal{L}_2$-norm*

$$\|\mathcal{G}_{\theta_{cl}}\|_{ind,2} < \gamma, \tag{2.150}$$

can be chosen as

$$K(\theta) = V(\theta)L(\theta)U^T(\theta)P \tag{2.151}$$

where $L(\theta) = [L_{11}(\theta)L_{12}(\theta)]$ with the elements of elements of $L_{11}(\theta) = [l_{ij}(\theta)] \in \mathcal{R}^{n_u \times n_u}$ are

$$l_{ij} = \frac{\tilde{a}_{ij} - \sigma_{jj}l_{ji} + \tilde{q}_{desired_{ij}}}{\sigma_{ii}}, \;\; l_{ji} = \text{ your choice } \; i = 1, \ldots n_u - 1, \; j = i+1, \ldots n_u \tag{2.152}$$

$$l_{ii} = \frac{\tilde{a}_{ii} + \tilde{q}_{desired_{ij}} + d_{ii}}{2\sigma_{ii}}, \; i = 1, \ldots n_u \tag{2.153}$$

and where $d_{ii} > 0$ are arbitrary real numbers for all $i = 1, \ldots n_u$, $\tilde{A} = [\tilde{a}_{ij}]$,and $\tilde{Q}_{desired} = [\tilde{q}_{desired_{ij}}(\theta)]$. The matrix $L_{12}(\theta) \in \mathcal{R}^{n_u \times n - n_u}$ is defined as

$$L_{12}(\theta) = \Sigma_1^{-1}(\theta)\{U_1^T(\theta)(A(\theta)W + WA^T(\theta))U_2(\theta) + \tilde{Q}_{desired}(1\colon n_u, n_u + 1\colon n)\}. \tag{2.154}$$

Proof. Assuming that $W > 0$ exists such that (2.135)is satisfied, then from the proof of Theorem 2.15 we see that the particular choice of $L(\theta) = \begin{bmatrix} L_{11}(\theta) & L_{12}(\theta) \end{bmatrix}$ from

(2.152), (2.153) and (2.154) will result in a quadratically stable closed-loop system having an induced $\mathcal{L}_2$-norm$< \gamma$. Recall (2.141)

$$L(\theta) = V^T(\theta)K(\theta)WU(\theta).$$

Solving for $K(\theta)$ yields

$$K(\theta) = V(\theta)L(\theta)U^T(\theta)P$$

which is our stated controller. □

As an example of designing a controller to guarantee $\mathcal{L}_2$-performance we have the following autopilot example.

Example 2.17. *Consider the autopilot example taken from [28] which we have modified. The modifications to this example are that we allow the speed and altitude to vary instead of being held constant. The nonlinear missile dynamics are as follows*

$$\dot{\alpha} = \frac{fg\cos(\frac{\alpha}{f})Z}{WtV} + q \tag{2.155}$$

$$\dot{q} = \frac{fm}{I_{yy}} \tag{2.156}$$

where

d = reference diameter, .75 ft

f = radian-to-degrees conversion, $\frac{180}{\pi}$

g = acceleration of gravity, 32.2 $\frac{\text{ft}}{s^2}$

I_{yy} = pitch moment of Inertia, 182.5 slug-ft^2

m = $C_m QSd$ = pitch moment, ft-lb

Q = dynamic pressure $\frac{\rho V^2}{2}$

ρ = $-2.0777\text{E-}18\ alt^3 + 6.3675\text{E-}13\ alt^2 - 6.6022\text{E-}08\ alt + 2.3465\ \text{E-}03 =$
a polynomial aproximation for air density

q = pitch rate, $\frac{\text{deg}}{s}$

S = reference area, .44 ft^2

alt = altitude, $10,000 - 30,000$ft

V = velocity, $2,000-4,000\frac{\text{ft}}{s}$
Wt= weight, 450 lb
Z = C_zQS = normal force, lb
α = angle-of-attack, ±20 deg
η_z = normal acceleration in g

The normal force and pitch moment aerodynamic coefficients are approximated by

$$C_z = \phi_{z\alpha}\alpha + b_z\delta \tag{2.157}$$

$$C_m = \phi_{m\alpha}\alpha + b_m\delta \tag{2.158}$$

where

b_z = -0.034
b_m = -0.206
δ = *fin deflection, deg*
$\phi_{z\alpha}$ = $.000103\alpha^2 - .00945|\alpha| - .170$
$\phi_{m\alpha}$ = $.000215\alpha^2 - .00195|\alpha| - .051$

The actuators are modeled as a second order system with transfer function

$$\delta(s) = \frac{\omega_a^2}{s^2 + 1.4\omega_a s + \omega_a^2}\delta_c(s) \tag{2.159}$$

where

δ_c = *commanded fin deflection, deg*
ω_a = *actuator bandwidth, 150* $\frac{rad}{s}$

The autopilot is required to control normal acceleration.

$$\eta_z = \frac{Z}{Wt} \tag{2.160}$$

The acceleration is typically expressed in g's which is the reason for dividing the normal force by the missile's weight and not mass. Now normal acceleration is not one of the states of the system but is a function of the fin deflection, δ, and angle-of-attack,

α. In order to design a state feedback Lyapunov based controller that has performance goals of controlling acceleration, we need to have acceleration or something directly related to it as one of the states. To accomplish this we augment an integrator to the acceleration error which can then be fed back to control the fin deflection. This augmentation is shown in the block diagram for the missile dynamic equations in Figure 2.12 as state $\eta_{z_{error}integ}$. The state-space plant description for the autopilot is

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\omega_a^2 & -1.4\omega & 0 & 0 & 0 \\ C_1 b_z & 0 & C_1\phi_{z\alpha} & 1 & 0 \\ C_{2b_m} & 0 & C_2\phi_{m\alpha} & 0 & 0 \\ -\frac{QSB_z}{Wt} & 0 & -\frac{\phi_{z\alpha}QS}{Wt} & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & \omega_a^2 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} u \qquad (2.161)$$

$$y = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ \frac{QSB_z}{Wt} & 0 & \frac{\phi_{z\alpha}QS}{Wt} & 0 & 0 \end{bmatrix} x \qquad (2.162)$$

To begin the design of a Lyapunov based gain scheduled controller with an $\mathcal{L}_2$ performance objective, we solve (2.135) with $C(\theta) = (0\,0\,0\,0\,30)$ and $\gamma = 1$ for W , yielding

$$W = \begin{bmatrix} 3.1923E{+}00 & -2.9917E{+}02 & 5.4020E{-}02 & 2.9604E{+}00 & -1.1890E{-}02 \\ -2.9917E{+}02 & 1.4702E{+}05 & -4.2091E{+}00 & 1.7026E{+}02 & -1.9795E{-}01 \\ 5.4020E{-}02 & -4.2091E{+}00 & 2.6663E{-}02 & -2.4629E{-}01 & -6.2319E{-}03 \\ 2.9604E{+}00 & 1.7026E{+}02 & -2.4629E{-}01 & 1.2872E{+}01 & -2.4599E{-}02 \\ -1.1890E{-}02 & -1.9795E{-}01 & -6.2319E{-}03 & -2.4599E{-}02 & 3.9060E{-}03 \end{bmatrix}, \qquad (2.163)$$

which can then be used to solve for the controller

$$K(\theta) = V(\theta)L(\theta)U^T(\theta)P \qquad (2.164)$$

Using (2.152), (2.153) and (2.154) and the free choice elements of L_{11} to be 1, the matrix L can be computed. A unit step response for the nominal closed-loop system

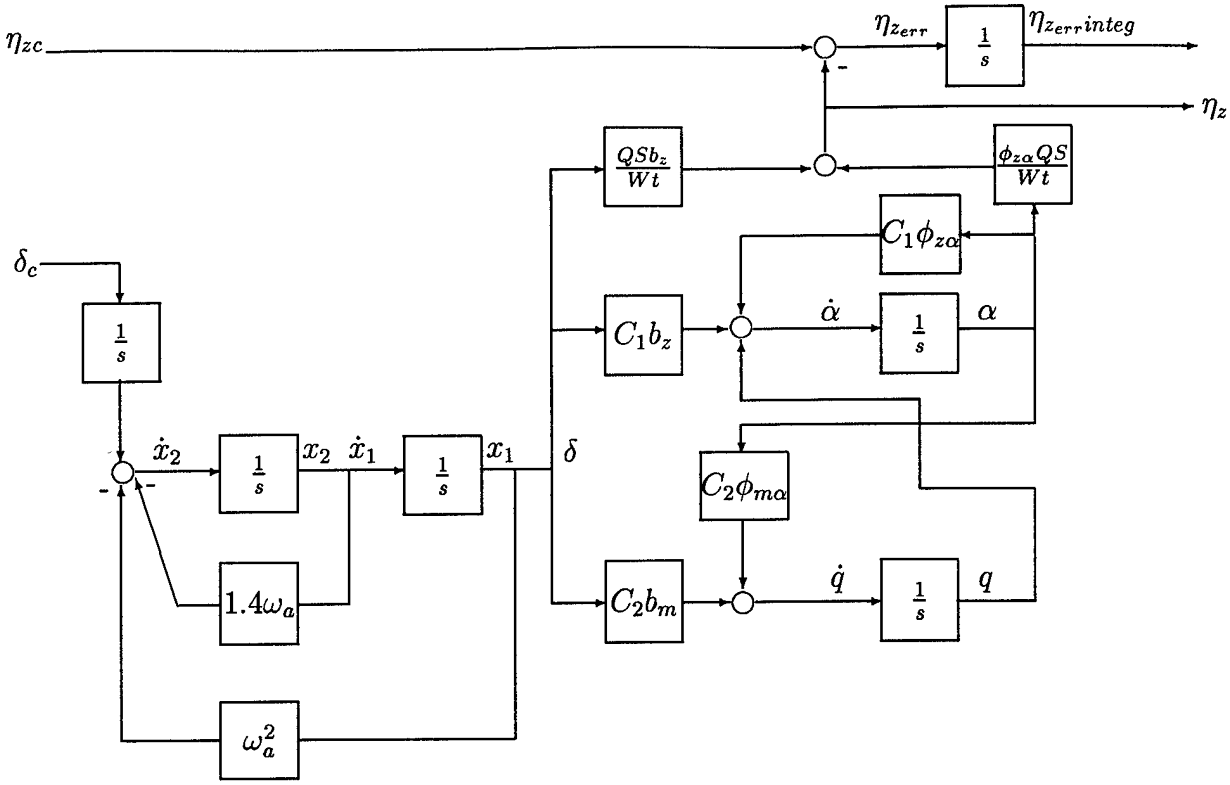

Figure 2.12: Autopilot block diagram.

is given in Figure 2.13. Nominal in the since that $V = 3000\frac{\text{ft}}{s}$ *and alt* $= 20000$ *ft during the step response. Figure 2.14 shows the response of the autopilot at the extremes of the parameter variations. The performance varies between the extremes because we held the Lyapunov function P to be constant which forces the Lyapunov Q to vary and performance is related to this Q. Our optimization on the inequality (2.135), pushes the worst case system to satisfy the* $\mathcal{L}_2$ *performance criteria almost exactly while the other systems do much better, as opposed to all systems satisfying the criteria exactly.*

Figure 2.15 shows the unit step response of the same autopilot with both V and alt varying wildly.

$$alt = 20000 + 10000\cos(6t) \tag{2.165}$$

$$V = 3000 + 1000\cos(12t) \tag{2.166}$$

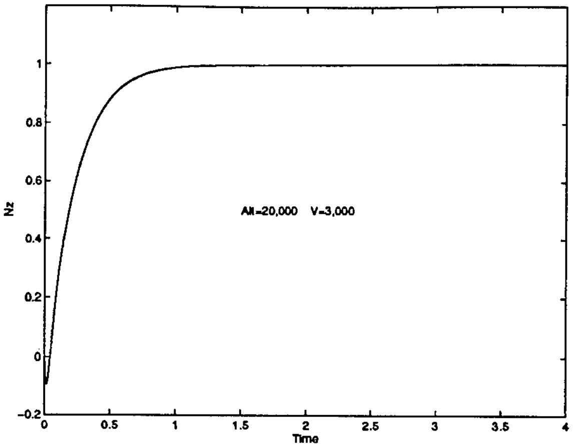

Figure 2.13: Unit step response of the autopilot with the nominal system.

The autopilot remains stable even under these unrealistic parameter variations. Even though the output acceleration η_z *varies quite a bit during this four second interval, its mean value is* 1.0093.

□

Another method of LPV gain scheduling is presented in the next Chapter. This method is H_∞ based.

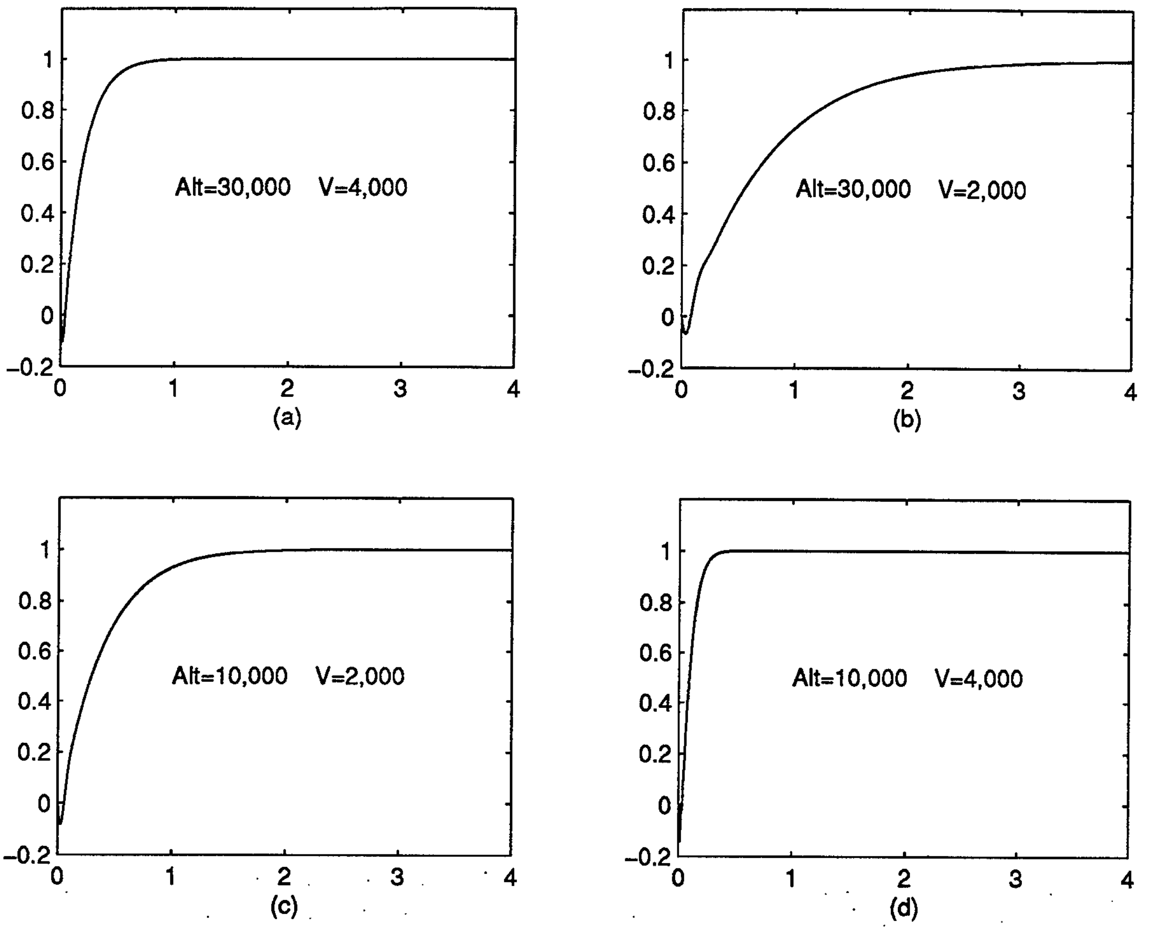

Figure 2.14: Unit step responses of the autopilot at various points in the parameter envelope.

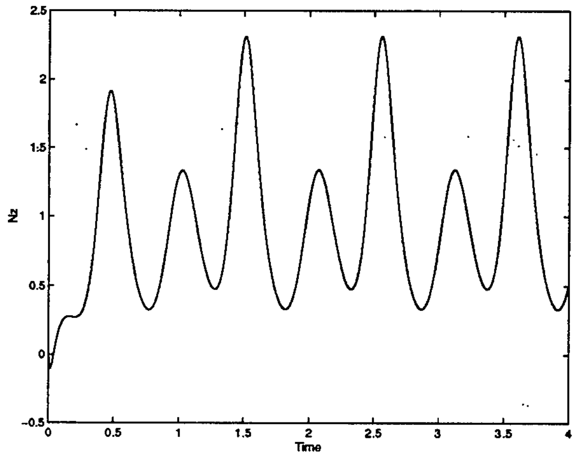

Figure 2.15: Unit step response of the autopilot with the missile dynamics varying rapidly.

CHAPTER 3
H_∞ Based Gain Scheduling

An alternative to the single quadratic Lyapunov approach for designing gain scheduled controllers for LPV plants which also has a firm theoretical basis has been proposed by Packard and Becker [16], Packard [29], and Apkarian and Gahinet [8]. There are two central ideas behind all three of these papers. The first is to treat the time-varying parameters as uncertainties by pulling them out into one large "uncertainty" block. The resulting LTI portion of the plant has a linear fractional dependence on this uncertainty block. By collecting the time-varying portion into an uncertainty block, the powerful methods of H_∞ control theory can be used to design a controller. The second central idea in these papers is in the assumption that the controller also has a linear fractional dependence on the parameters. This controller structure allows the controller to have knowledge of the parameters. In [16] and [29], the discrete version of the LPV system is discussed and sufficient conditions for designing an H_∞ gain scheduled controller are given. Apkarian and Gahinet in [8] discuss both the discrete case and the continuous case under the framework of the scaled bounded real lemma and provide sufficient conditions for designing an H_∞ gain scheduled controller. The conditions are only sufficient due to the fact that the small gain theorem is used which does not take into account the realness of the parameters as did the single quadratic Lyapunov approach. In the sequel, only the continuous time case will be presented. The material in this chapter is found primarily in [8] and is included as background for the present application.

3.1 H_∞ Gain Scheduling Problem

We begin by specifically defining the LPV plant and the parameters. Let $\theta = (\theta_1, \cdots, \theta_{n_\theta}) \in \mathcal{R}^{n_\theta}$ be the measured parameters. The fact that the parameters vary with time is assumed but is dropped from the notation. The plant is assumed to have a linear fractional dependence on the parameters. Therefore, the plant can be represented by the upper LFT

$$\begin{bmatrix} q \\ y \end{bmatrix} = F_u(P(s), \Theta) \begin{bmatrix} w \\ u \end{bmatrix} \tag{3.1}$$

where

$$\Theta = \text{blockdiag}\left(\theta_1 I_{r1}, \cdots, \theta_{n_\theta} I_{r\theta}\right), \tag{3.2}$$

$w \in \mathcal{R}^{p_1}$ is the disturbance input, $q \in \mathcal{R}^{p_1}$ is the controlled output, $u \in \mathcal{R}^{m_2}$ is the controlled input, and $y \in \mathcal{R}^{p_2}$ is the measured output. If the plant can not be represented as an LFT, approximation methods may be used to fit the parameter dependence to the LFT model. Equation (3.2) describes the structure of the set of parameters. In other words, it tells how many times each parameter or parameter block is repeated. Realizing a general LPV plant that has rational dependence on the parameters may require using multiple copies of each parameter. Various methods by which to accomplish this using as few copies as possible will be discussed in Chapter 4. The upper LFT of the plant with the parameters is depicted in Figure 3.1. Writing the feedback equations for the diagram in Figure 3.1 yields

$$\begin{bmatrix} q_\theta(s) \\ q(s) \\ y(s) \end{bmatrix} = \overbrace{\begin{bmatrix} P_{\theta\theta}(s) & P_{\theta 1}(s) & P_{\theta 2}(s) \\ P_{1\theta}(s) & P_{11}(s) & P_{12}(s) \\ P_{2\theta}(s) & P_{21}(s) & P_{22}(s) \end{bmatrix}}^{P(s)} \begin{bmatrix} w_\theta(s) \\ w(s) \\ u(s) \end{bmatrix} \tag{3.3}$$

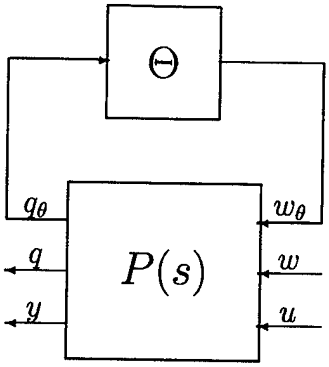

Figure 3.1: A diagram of the LPV plant with an upper linear fractional dependence on the parameters.

$$w_\theta = \Theta q_\theta \tag{3.4}$$

The controller is assumed to have a linear fractional dependence on Θ.

$$u = F_l(K(s), \Theta)y \tag{3.5}$$

This form for the controller is not the most general one but allows the gain scheduled problem to be converted into a standard H_∞ problem. Connecting the gain scheduled controller in (3.5) with the LPV plant in (3.1) as depicted in Figure 3.2 yields

$$T(P, K, \Theta) = F_l(F_u(P, \Theta), F_l(K, \Theta)) \tag{3.6}$$

With the plant, parameters and controller defined, it is now necessary to define the problem. The precise problem definition is influenced by the method used in solving it. In the problem at hand, it is desired to design an H_∞ gain scheduled controller, and in H_∞ theory the small gain theorem plays a prominent role. A particular form of the small gain theorem can be stated as follows

Theorem 3.1 (Small Gain Theorem). *Let $M(s)$ represent the transfer function of a linear time invariant system and $\Delta(t)$ be a stable time-varying perturbation.*

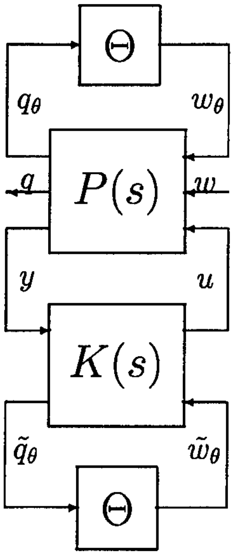

Figure 3.2: A diagram of the interconnection of the LPV plant with the gain scheduled controller.

The closed-loop system shown in Figure 3.3 is $\mathcal{L}_2$-stable for all Δ with $\|\Delta\| < \frac{1}{\gamma}$ if $\|M(s)\| < \gamma$.

Proof. See pg 45 [30]. □

Essentially, what this means is that a feedback loop constructed of stable operators will remain stable if the product of the operators is less than unity. In terms of Nyquist, it means that the magnitude of the Nyquist plot is never larger than unity, thereby making an encirclement of the -1 point impossible. This approach is somewhat conservative because it assumes arbitrary bounded parameter variations. Taking into account parameter variations having a certain structure by use of similarity scalings reduces the conservatism, however, constraining the parameters to be real will not be taken into account.

Taking the small gain theorem into consideration, our problem definition becomes

Find a control structure $K(s)$ such that the LPV controller $F_l(K(s), \Theta)$ satisfies

i) the closed loop system $T(P, K, \Theta)$ is internally stable for all bounded parameter trajectories $\theta(t)$ such that $\gamma^2\Theta^T\Theta \leq 1$;

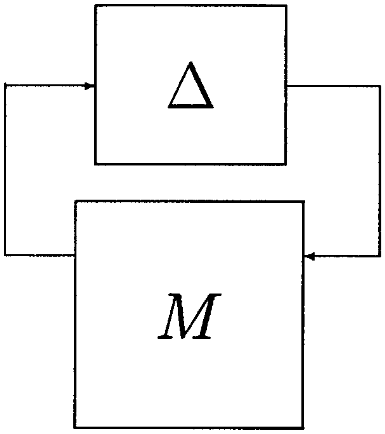

Figure 3.3: Typical plant interconnection for use with the small gain theorem.

ii) the induced $\mathcal{L}_2$-norm of the operator $T(P, K, \Theta)$ satisfies

$$\max_{\|\Theta\|_\infty \leq \frac{1}{\gamma}} \|T(P, K, \Theta)\|_\infty < \gamma. \tag{3.7}$$

The first condition i) can be assumed without loss of generality. It is just a matter of redefining the parameters to satisfy this condition. The condition ii) is directly from the small gain theorem. We want to find a controller structure $K(s)$ such that the feedback loop gain is less than unity. Redefining the parameters to satisfy i) allows for clear interpretation of knowing what the desired γ should be. In order for the system to remain stable over the entire parameter space, we must have $\|T(P, K, \Theta)\|_\infty < 1$, in other words, we must find a $K(s)$ that results in $\gamma < 1$.

H_∞ theory is set up to find a controller $K(s)$ that will stabilize the plant depicted in Figure 3.1 and not to find a controller of the form $F_l(K(s), \Theta)$. Therefore, we must find a way to put our problem into that form. Collecting the parameters into one repeated "uncertainty" block, $\begin{bmatrix} \Theta & 0 \\ 0 & \Theta \end{bmatrix}$, and redrawing Figure 3.2 as Figure 3.4 results in the standard H_∞ form. This block repeated structure, $\begin{bmatrix} \Theta & 0 \\ 0 & \Theta \end{bmatrix}$, will be

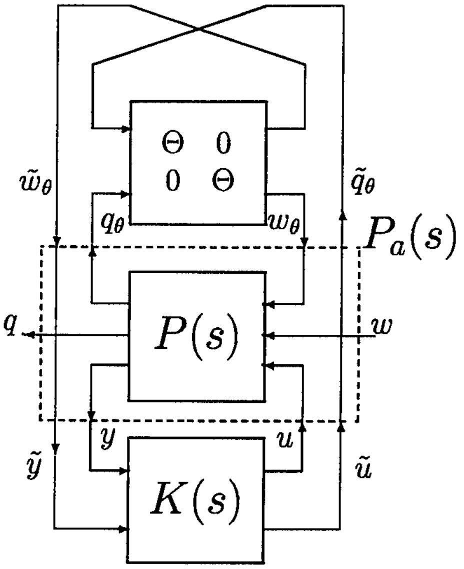

Figure 3.4: A diagram of the interconnection of the LPV plant with the gain scheduled controller.

denoted $\Theta \oplus \Theta$. Figure 3.4 is represented mathematically as

$$T(P,K,\Theta) = F_u\left(F_l(P_a,K), \begin{bmatrix} \Theta & 0 \\ 0 & \Theta \end{bmatrix}\right) \tag{3.8}$$

where P_a is the augmented plant and is defined in the following equation.

$$\left[\begin{array}{c} \tilde{q}_\theta \\ \hline q_\theta \\ q \\ y \\ \hline \tilde{y} \end{array}\right] = \overbrace{\left[\begin{array}{c|c|c} 0 & 0 & I_r \\ 0 & P(s) & 0 \\ I_r & 0 & 0 \end{array}\right]}^{P_a} \left[\begin{array}{c} \tilde{w}_\theta \\ \hline w_\theta \\ w \\ u \\ \hline \tilde{u} \end{array}\right] \tag{3.9}$$

In order to reduce the conservatism as mentioned earlier, similarity scaling will be introduced. The similarity scalings reduce the conservatism due to the fact that the H_∞ norm is not invariant under similarity scalings. Therefore, in general, there exists an $L \neq I$ such that the H_∞ norm in (3.12) is minimized. Specifically consider the set of positive definite similarity scalings associated with the structure Θ in (3.2)

$$L_{\Theta} = \{L > 0 \colon L\Theta = \Theta L, \forall\Theta\} \subset \mathcal{R}^{r\times r} \tag{3.10}$$

with

$$r = \sum_{i=1}^{n_\theta} r_i.$$

Positive definiteness of the scalings is assumed without loss of generality due to the polar decomposition theorem (see for example [31]) and the fact that the H_∞ norm is invariant with respect to unitary matrices. Positive definiteness insures the invertibility of the scaling L with the use of a convex constraint. For the repeated block structure, $\Theta \oplus \Theta$, the scaling set becomes

$$L_{\Theta\oplus\Theta} = \left\{ \begin{bmatrix} L_1 & L_2 \\ L_2^T & L_3 \end{bmatrix} > 0 \colon L_1, L_3 \in L_{\Theta}\text{and } L_2\Theta = \Theta L_2, \forall\Theta \right\}. \tag{3.11}$$

Suppose there is a scaling matrix L such that (3.11) holds. Without loss of generality we can insert blocks as shown in Figure 3.5(a). Using (3.11), the top two L and L^{-1} blocks can be dropped so that the diagram can now be drawn as shown in Figure 3.5(b) which must be optimized over all controllers $K(s)$ and scaling matrices L such that (3.10) holds.

The following theorem, by the use of the small gain theorem, formally states a sufficient condition for solvability of the H_∞ gain scheduled control problem.

Theorem 3.2 ([8]). *Consider the open loop system $P_a(s)$ with uncertainty structure $\Theta \oplus \Theta$ along with its associated similarity scalings $L_{\Theta\oplus\Theta}$. If there exist a scaling matrix $L \in L_{\Theta\oplus\Theta}$ and an LTI controller $K(s)$ such that the nominal closed-loop*

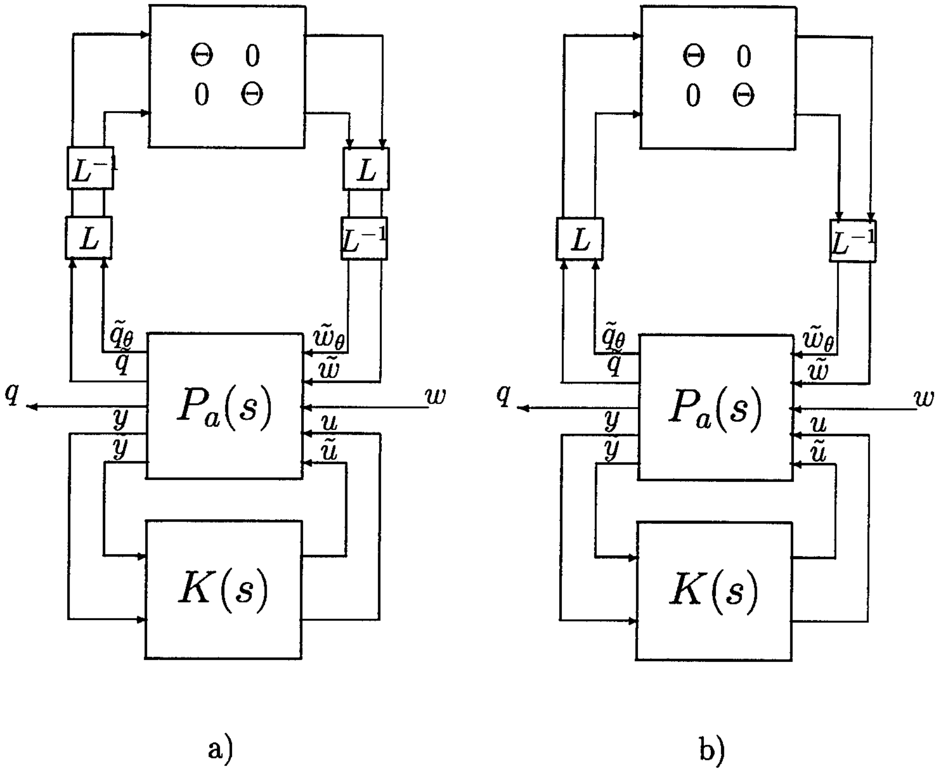

Figure 3.5: Insertion of scaling matrices into the plant.

system $F_l(P_a(s), K(s))$ is internally stable and satisfies

$$\left\| \begin{bmatrix} L^{\frac{1}{2}} & 0 \\ 0 & I \end{bmatrix} F_l(P_a(s), K(s)) \begin{bmatrix} L^{-\frac{1}{2}} & 0 \\ 0 & I \end{bmatrix} \right\|_\infty < \gamma \tag{3.12}$$

then $F_l(K(s), \Theta)$ is a γ-suboptimal gain scheduled H_∞ controller.

Proof. The proof is a straightforward application of the small gain theorem. □

Equation (3.12) is the scaled small gain condition. Pictorially (3.12) can be represented as shown in Figure 3.6. The square roots are introduced here and at this point are unnecessary. There necessity comes about from the Scaled Bounded Real Lemma which will be used to solve (3.12)

Theorem 3.2 is a particular case of the general scaled H_∞ problem, which will be discussed in the next section, but before proceeding it is necessary that state-space realizations for the plant and the controller be defined. The purpose for this is that

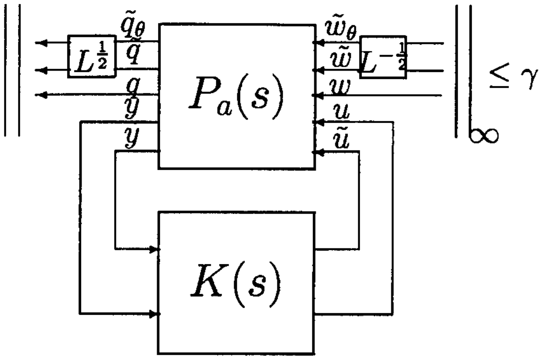

Figure 3.6: A diagram of the scaled small gain condition.

very efficient state-space based LMI algorithms will be developed to solve for the controller structure $K(s)$. Therefore consider some state-space realization (ideally minimal in terms of the states and the parameters) of the LTI plant $P(s)$.

$$P(s) = \begin{bmatrix} D_{\theta\theta} & D_{\theta 1} & D_{\theta 2} \\ D_{1\theta} & D_{11} & D_{12} \\ D_{2\theta} & D_{21} & D_{22} \end{bmatrix} + \begin{bmatrix} C_\theta \\ C_1 \\ C_2 \end{bmatrix} (sI - A)^{-1}(B_\theta \; B_1 \; B_2) \qquad (3.13)$$

where the partitioning is conformable to (3.3). The problem dimensions are given by

$$A \in \mathcal{R}^{n\times n}, \quad D_{\theta\theta} \in \mathcal{R}^{r\times r}, \quad D_{11} \in \mathcal{R}^{p_1\times p_1}, \quad D_{22} \in \mathcal{R}^{p_2\times m_2}. \qquad (3.14)$$

The following assumptions are made concerning the realization of the plant $P(s)$.

A1) $(A,\ B_2,\ C_2)$ is stabilizable and detectable;

A2) $D_{22} = 0$.

The first assumption is a standard assumption and is necessary and sufficient to allow stabilization of the plant by dynamic output feedback. The second assumption is made without loss of generality but greatly simplifies the calculations. It should be noted that both the parameter set Θ and the plant transfer function from the disturbance w to the controlled output q have been considered to be square. If this is not the case then by augmenting the plant with rows and/or columns of zeros the plant transfer function can be made square. This assumption on the parameters and plant transfer functions simplifies the notation and the manipulations of the scaling matrices.

A state-space realization for the plant $P_a(s)$ can be derived from (3.13) by augmenting it with the parameters due to the controller.

$$P_a(s) = \begin{bmatrix} 0 & 0 & 0 & 0 & I_r \\ 0 & D_{\theta\theta} & D_{\theta 1} & D_{\theta 2} & 0 \\ 0 & D_{1\theta} & D_{11} & D_{12} & 0 \\ 0 & D_{2\theta} & D_{21} & D_{22} & 0 \\ I_r & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ C_\theta \\ C_1 \\ C_2 \\ 0 \end{bmatrix} (sI - A)^{-1}(0\ B_\theta\ B_1\ B_2\ 0) \tag{3.15}$$

The state-space realization for the controller $K(s)$ is

$$K(s) = \begin{bmatrix} D_{K11} & D_{K1\theta} \\ D_{K\theta 1} & D_{K\theta\theta} \end{bmatrix} + \begin{bmatrix} C_{K1} \\ C_{K\theta} \end{bmatrix} (sI - A_K)^{-1}(B_{K1}\ B_{K\theta}), \tag{3.16}$$

with

$$A_K \in \mathcal{R}^{k \times k}.$$

Note that the order k of $K(s)$ is arbitrary at this point.

3.2 General Scaled H_∞ Problem

In the last section, the H_∞ gain scheduled problem was set up to be a particular case of the general scaled H_∞ problem. By use of the scaled bounded real lemma, the H_∞ constraints can be turned into LMI constraints which can now be efficiently solved [15, 32].

Lemma 3.3 (Scaled Bounded Real Lemma [8]). *Consider a parameter structure Θ, the associated scaling set L_Θ defined in (3.10), and a square continuous-time* *transfer function $T(s)$ with realization $T(s) = D_{cl} + C_{cl}(sI - A_{cl})^{-1}B_{cl}$. The following statements are equivalent.*

i) A_{cl} is stable and there exists $L \in L_\Theta$ such that

$$\|L^{\frac{1}{2}}(D_{cl} + C_{cl}(sI - A_{cl})^{-1}B_{cl})L^{-\frac{1}{2}}\|_\infty < \gamma. \tag{3.17}$$

ii) There exist positive definite solutions X and $L \in L_\Theta$ to the matrix inequality

$$\begin{bmatrix} A_{cl}^T X + XA_{cl} & XB_{cl} & C_{cl}^T \\ B_{cl}^T X & -\gamma L & D_{cl}^T \\ C_{cl} & D_{cl} & -\gamma L^{-1} \end{bmatrix} < 0 \tag{3.18}$$

Proof. See [33] and references therein. □

The scaled bounded real lemma is used here to design an H_∞ gain scheduled controller as the Lyapunov equation was used to design a gain scheduled controller in Chapter 2. The scaled bounded real lemma is an analysis tool. Given a state-space realization of a system, one can check whether or not the system is stable and satisfies a certain H_∞ constraint. For our purposes the state-space system, $T(s)$ will be the state-space realization for the closed-loop system. As was done previously, necessary and sufficient conditions can be derived in terms of only the open-loop system and X. Let the open-loop system be denoted $G(s)$ and have a state-space realization

$$G(s) = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} + \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} (sI - A)^{-1}(B_1 \; B_2) \tag{3.19}$$

In (3.19) the matrix A refers to the open-loop system where A_{cl} in (3.17) refers to the complete closed-loop system including the controller.

Theorem 3.4 (Theorem 4.1 in [8]). *With $G(s)$, Θ and L_Θ defined as above, Let $\mathcal{K}_R$ and $\mathcal{K}_S$ denote the bases of the null spaces of $(B_2^T \; D_{12}^T \; 0_{m_2 \times p_1})$ and $(C_2 \; D_{21} \; 0_{p_2 \times p_1})$,* *respectively. With this notation, the suboptimal scaled H_∞ problem is solvable if and only if there exist pairs of symmetric matrices $(R, S) \in \mathcal{R}^{n\times n}$ and $(L, J) \in \mathcal{R}^{r\times r}$ such that*

$$\mathcal{K}_R^T \begin{bmatrix} AR + RA^T & RC_1^T & B_1 \\ C_1R & -\gamma J & D_{11} \\ B_1^T & D_{11}^T & -\gamma L \end{bmatrix} \mathcal{K}_R < 0 \tag{3.20}$$

$$\mathcal{K}_S^T \begin{bmatrix} A^TS + SA & SB_1 & C_1^T \\ B_1^TS & -\gamma L & D_{11}^T \\ C_1 & D_{11} & -\gamma J \end{bmatrix} \mathcal{K}_S < 0 \tag{3.21}$$

$$\begin{bmatrix} R & I \\ I & S \end{bmatrix} \geq 0 \tag{3.22}$$

$$L \in L_\Theta, J \in L_\Theta, LJ = I. \tag{3.23}$$

Moreover, there exists suboptimal controllers of order k if and only if (3.20)-(3.23) hold for some quadruple (R, S, L, J) where R, S further satisfy the rank constraint

$$\mathit{rank}(I - RS) \leq k. \tag{3.24}$$

Proof. See Apkarian and Gahinet [8]. □

In Theorem 3.4 the two matrices R and S have replaced the X matrix as follows

$$X = \begin{bmatrix} S & N \\ N^T & E \end{bmatrix}, \quad X^{-1} = \begin{bmatrix} R & M \\ M^T & F \end{bmatrix}, \tag{3.25}$$

with $(N, M) \in \mathcal{R}^{n\times k}$, $(E, F) \in \mathcal{R}^{k\times k}$ and $(S, R) \in \mathcal{R}^{n\times n}$. Due to this substitution and the introduction of the nullspaces $\mathcal{K}_R$ and $\mathcal{K}_S$, the LMI (3.18) reduces to the two

LMI's (3.20) and (3.21). Note that the LMI (3.18) contains the controller parameters through the A_{cl} but the new LMI's (3.20) and (3.21) involve the open-loop A matrix. The relationships given in (3.25) are used to find X and hence the controller parameters.

The three LMI constraints (3.20)-(3.22) are all convex. The structure constraints $L,\ J \in L_\Theta$ are also convex, but the constraint, $LJ = I$, is strongly non-convex. Developing methods and algorithms to solve (3.20)-(3.23) with the non-convex $LJ = I$ constraint remains an important and open area of research.

Theorem 3.4 is considered to be the general scaled H_∞ problem since it reduces to the standard H_∞ problem when Θ becomes an arbitrary unstructured uncertainty block (i.e. $\Theta \in \mathcal{C}^{r\times r}$). In this case the scaling set becomes $L_\Theta = \{l * I_r : l \in \mathcal{R}, l > 0\}$ and $L,\ J$ can be set to the identity without loss of generality. This will then immediately satisfy the $LJ = I$ constraint leaving the other convex constraints which can efficiently be solved via various LMI solver utilities such as SDPSOL [15] or LMI-LAB [32]. This alternative method of solving the classical H_∞ problem avoids the problem with imaginary axis zeros and the rank deficiencies in D_{12} and D_{21}. For a detailed discussion see [33].

3.3 Solution of the H_∞ Gain Scheduled Problem

Fortunately, the particular structure of $P_a(s)$, which provides the controller knowledge of the parameters, replaces the strongly non-convex constraint $LJ = I$ with a convex constraint. The following theorem states the necessary and sufficient conditions for solvability of the H_∞ gain scheduled problem.

Theorem 3.5 (Theorem 5.1 in [8] and corrected in [10]). *Consider an LPV plant given by the LFT interconnection (3.1) where $P(s)$ is a proper continuous-time LTI plant with minimal realization (3.13), and Θ is the parameter set given by (3.2). The corresponding scaling set for Θ is denoted L_Θ and defined by (3.10). Finally,* *assume A1)-A2) and let* $\mathcal{N}_R$ *and* $\mathcal{N}_S$ *denote the arbitrary bases of the null spaces of* $(B_2^T,\ D_{\theta 2}^T,\ D_{12}^T,\ 0_{m_2\times(r+p_1)})$ *and* $(C_2,\ D_{2\theta},\ D_{21},\ 0_{p_2\times(r+p_1)})$, *respectively. In addition, let*

$$\hat{B}_1 = (B_\theta\, B_1),\ \ \hat{C}_1 = \begin{bmatrix} C_\theta \\ C_1 \end{bmatrix},\ \ \hat{D}_{11} = \begin{bmatrix} D_{\theta\theta} & D_{\theta 1} \\ D_{1\theta} & D_{11} \end{bmatrix}. \tag{3.26}$$

With this notation and assumptions the gain scheduled H_∞ *problem is solvable if there exist pairs of symmetric matrices* $(R,S) \in \mathcal{R}^{n\times n}$ *and* $(L_3, J_3) \in \mathcal{R}^{r\times r}$ *such that*

$$\mathcal{N}_R^T \begin{bmatrix} AR + RA^T & R\hat{C}_1^T & \hat{B}_1 \begin{bmatrix} J_3 & 0 \\ 0 & I \end{bmatrix} \\ \hat{C}_1 R & -\gamma \begin{bmatrix} J_3 & 0 \\ 0 & I \end{bmatrix} & \hat{D}_{11} \begin{bmatrix} J_3 & 0 \\ 0 & I \end{bmatrix} \\ \begin{bmatrix} J_3 & 0 \\ 0 & I \end{bmatrix} \hat{B}_1^T & \begin{bmatrix} J_3 & 0 \\ 0 & I \end{bmatrix} \hat{D}_{11}^T & -\gamma \begin{bmatrix} J_3 & 0 \\ 0 & I \end{bmatrix} \end{bmatrix} \mathcal{N}_R < 0 \tag{3.27}$$

$$\mathcal{N}_S^T \begin{bmatrix} A^T S + SA & S\hat{B}_1 & \hat{C}_1^T \begin{bmatrix} L_3 & 0 \\ 0 & I \end{bmatrix} \\ \hat{B}_1^T S & -\gamma \begin{bmatrix} L_3 & 0 \\ 0 & I \end{bmatrix} & \hat{D}_{11}^T \begin{bmatrix} L_3 & 0 \\ 0 & I \end{bmatrix} \\ \begin{bmatrix} L_3 & 0 \\ 0 & I \end{bmatrix} \hat{C}_1 & \begin{bmatrix} L_3 & 0 \\ 0 & I \end{bmatrix} \hat{D}_{11} & -\gamma \begin{bmatrix} L_3 & 0 \\ 0 & I \end{bmatrix} \end{bmatrix} \mathcal{N}_S < 0 \tag{3.28}$$

$$\begin{bmatrix} R & I \\ I & S \end{bmatrix} \geq 0, \tag{3.29}$$

$$L_3,\ J_3 \in L_\Theta, \quad \begin{bmatrix} L_3 & I \\ I & J_3 \end{bmatrix} \geq 0 \tag{3.30}$$

Proof. The proof as well as the original statement of this theorem as given in [8] are not correct and are incomplete, but after personal communication with Capt. M.S. Spillman, [9], the following proof is given. The proof begins by substituting our particular plant structure into the necessary and sufficient conditions given in Theorem 3.4. These new conditions will be shown to be equivalent to the conditions (3.27)-(3.30). Therefore, the following substitutions are made. Note $X \rightarrow Y$ reads replace the matrix X in the state-space realization of $G(s)$ with Y.

$$C_1 \rightarrow \begin{bmatrix} 0 \\ C_\theta \\ C_1 \end{bmatrix}, \quad C_2 \rightarrow \begin{bmatrix} C_2 \\ 0 \end{bmatrix}, \quad B_1 \rightarrow (0\ \ B_\theta\ \ B_1), \quad B_2 \rightarrow (B_2\ \ 0) \tag{3.31}$$

$$D_{11} \rightarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & D_{\theta\theta} & D_{\theta 1} \\ 0 & D_{1\theta} & D_{11} \end{bmatrix}, \quad D_{12} \rightarrow \begin{bmatrix} 0 & I_r \\ D_{\theta 2} & 0 \\ D_{12} & 0 \end{bmatrix} \tag{3.32}$$

$$D_{21} \rightarrow \begin{bmatrix} 0 & D_{2\theta} & D_{21} \\ I & 0 & 0 \end{bmatrix}, \quad D_{22} \rightarrow \begin{bmatrix} D_{22} & 0 \\ 0 & 0 \end{bmatrix} \tag{3.33}$$

The A matrix remains unchanged and the nullspaces $\mathcal{N}_R$ and $\mathcal{N}_S$ become (B_2^T, $D_{\theta 2}^T$, D_{12}^T, 0) and (C_2, $D_{2\theta}$, D_{21}, 0), respectively. Upon making these substitutions, (3.20) becomes

$$\mathcal{N}_R^T \begin{bmatrix} AR + RA^T & R\begin{bmatrix} 0 & C_\theta^T & C_1^T \end{bmatrix} & \begin{bmatrix} 0 & B_\theta & B_1 \end{bmatrix} \\ \begin{bmatrix} 0 \\ C_\theta \\ C_1 \end{bmatrix} R & -\gamma \begin{bmatrix} J_1 & J_2 & 0 \\ J_2^T & J_3 & 0 \\ 0 & 0 & I \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & D_{\theta\theta} & D_{\theta 1} \\ 0 & D_{1\theta} & D_{11} \end{bmatrix} \\ \begin{bmatrix} 0 \\ B_\theta^T \\ B_1^T \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & D_{\theta\theta}^T & D_{1\theta}^T \\ 0 & D_{\theta 1}^T & D_{11}^T \end{bmatrix} & -\gamma \begin{bmatrix} L_1 & L_2 & 0 \\ L_2^T & L_3 & 0 \\ 0 & 0 & I \end{bmatrix} \end{bmatrix} \mathcal{N}_R < 0 \qquad (3.34)$$

where $\mathcal{N}_R = \begin{bmatrix} P_1 & 0 \\ 0 & 0 \\ P_2 & 0 \\ P_3 & 0 \\ 0 & \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \end{bmatrix}$ is a basis of the null space of $\begin{bmatrix} B_2^T & 0 & D_{\theta 2}^T & D_{12}^T & 0 \\ 0 & I_r & 0 & 0 & 0 \end{bmatrix}$.

Let $P = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix}$ and $\tilde{A} = \begin{bmatrix} AR + RA^T & RC_\theta^T & RC_1^T \\ C_\theta R & -\gamma J_3 & 0 \\ C_1 R & 0 & -\gamma I \end{bmatrix}$. Now multiplying (3.34) out and deleting the block row and block column corresponding to the block zero row of $\mathcal{N}_R$ yields

$$\begin{bmatrix} P^T \tilde{A} P & P^T \begin{bmatrix} 0 & B_\theta & B_1 \\ 0 & D_{\theta\theta} & D_{\theta 1} \\ 0 & D_{1\theta} & D_{11} \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ B_\theta^T & D_{\theta\theta}^T & D_{1\theta}^T \\ B_1^T & D_{\theta 1}^T & D_{11}^T \end{bmatrix} P & -\gamma \begin{bmatrix} L_1 & L_2 & 0 \\ L_2^T & L_3 & 0 \\ 0 & 0 & I \end{bmatrix} \end{bmatrix} < 0 \qquad (3.35)$$

Taking the Schur complement of (3.35) yields

$$P^T\tilde{A}P+\frac{1}{\gamma}P^T\begin{bmatrix}0 & B_\theta & B_1\\ 0 & D_{\theta\theta} & D_{\theta 1}\\ 0 & D_{1\theta} & D_{11}\end{bmatrix}\begin{bmatrix}J_1 & J_2 & 0\\ J_2^T & J_3 & 0\\ 0 & 0 & I\end{bmatrix}\begin{bmatrix}0 & 0 & 0\\ B_\theta^T & D_{\theta\theta}^T & D_{1\theta}^T\\ B_1^T & D_{\theta 1}^T & D_{11}^T\end{bmatrix}P<0 \qquad (3.36)$$

which is equivalent to

$$P^T\tilde{A}P+\frac{1}{\gamma}P^T\begin{bmatrix}B_\theta & B_1\\ D_{\theta\theta} & D_{\theta 1}\\ D_{1\theta} & D_{11}\end{bmatrix}\begin{bmatrix}J_3 & 0\\ 0 & I\end{bmatrix}\begin{bmatrix}B_\theta^T & D_{\theta\theta}^T & D_{1\theta}^T\\ B_1^T & D_{\theta 1}^T & D_{11}^T\end{bmatrix}P<0 \qquad (3.37)$$

Equation (3.37) is equivalent to

$$P^T\tilde{A}P+\frac{1}{\gamma}P^T\begin{bmatrix}B_\theta & B_1\\ D_{\theta\theta} & D_{\theta 1}\\ D_{1\theta} & D_{11}\end{bmatrix}\begin{bmatrix}J_3 & 0\\ 0 & I\end{bmatrix}\begin{bmatrix}J_3 & 0\\ 0 & I\end{bmatrix}^{-1}\begin{bmatrix}J_3 & 0\\ 0 & I\end{bmatrix}\begin{bmatrix}B_\theta^T & D_{\theta\theta}^T & D_{1\theta}^T\\ B_1^T & D_{\theta 1}^T & D_{11}^T\end{bmatrix}P<0 \qquad (3.38)$$

which is the Schur complement of (3.27).

The constraint given in (3.21) can be shown to be equivalent to (3.28) in exactly the same manner. The third constraint given in (3.22) is already exactly (3.29). The last constraint $LJ=I$ can be converted to an LMI constraint by use of matrix dilation. Note that $L1$, $L2$, $J1$ and $J2$ no longer appear in (3.27)-(3.29). This fact makes them arbitrary. This arbitrariness coupled with matrix dilation converts $LJ=I$ into

$$\begin{bmatrix}L_3 & I\\ I & J_3\end{bmatrix}\geq 0$$

which is the last constraint given in (3.30). Therefore, under the particular structure of the H_∞ gain scheduled problem the four constraints given in (3.20)-(3.23) have been shown to be equivalent to (3.27)-(3.30) which are all convex. □

These LMI constraints provide only solvability conditions for an H_∞ gain scheduled controller. The next step is to take the quadruple (R, S, L_3, J_3) and build the controller $K(s)$.

3.4 Computation of the Controller

The idea in constructing a controller is to first construct an X from R and S and an L, which in turn also specifies J, from L_3 and J_3. Once these are constructed, the scaled bounded real lemma can be used to solve for the controller, which is now an LMI in terms of the controller. Apkarian and Gahinet [8] showed that all of the controller parameters can be gathered into one matrix, Ω defined in (3.41), and the scaled bounded real lemma could be rewritten as

$$\Psi + \begin{bmatrix} X & 0 \\ 0 & I \end{bmatrix} \mathcal{P}^T \Omega \mathcal{Q} + \mathcal{Q}^T \Omega^T \mathcal{P} \begin{bmatrix} X & 0 \\ 0 & I \end{bmatrix} < 0 \tag{3.39}$$

where Ψ, $\mathcal{P}$, Ω and $\mathcal{Q}$ are given by

$$\Psi := \begin{bmatrix} A_0^T X + X A_0 & X B_0 & C_0^T \\ B_0^T X & -\gamma \mathcal{L} & \mathcal{D}_{11}^T \\ C_0 & \mathcal{D}_{11} & \gamma \mathcal{J} \end{bmatrix}, \quad \mathcal{P} := (\mathcal{B}^T,\ 0,\ \mathcal{D}_{12}^T). \tag{3.40}$$

$$\Omega := \begin{bmatrix} A_K & B_{K1} & B_{K\theta} \\ C_{K1} & D_{K11} & B_{K1\theta} \\ C_{K\theta} & D_{K\theta 1} & B_{K\theta\theta} \end{bmatrix} \in \mathcal{R}^{(k+m_2+r)\times(k+p_2+r)}, \quad \mathcal{Q} := (\mathcal{C},\ \mathcal{D}_{21}, 0). \tag{3.41}$$

and where

$$A_0 = \begin{bmatrix} A & 0 \\ 0 & 0_{k\times k} \end{bmatrix}, \qquad B_0 = \begin{bmatrix} 0 & B_\theta & B_1 \\ 0_{k\times r} & 0 & 0 \end{bmatrix},$$

$$\mathcal{B} = \begin{bmatrix} 0 & B_2 & 0 \\ I_k & 0 & 0_{k\times r} \end{bmatrix}, \qquad C_0 = \begin{bmatrix} 0 & 0_{r\times k} \\ C_\theta & 0 \\ C_1 & 0 \end{bmatrix},$$

$$\mathcal{D}_{11} = \begin{bmatrix} 0_{r\times r} & 0 & 0 \\ 0 & D_{\theta\theta} & D_{\theta 1} \\ 0 & D_{1\theta} & D_{11} \end{bmatrix}, \qquad \mathcal{D}_{12} = \begin{bmatrix} 0_{r\times k} & 0 & I_r \\ 0 & D_{\theta 2} & 0 \\ 0 & D_{12} & 0 \end{bmatrix},$$

$$\mathcal{C} = \begin{bmatrix} 0 & I_k \\ C_2 & 0 \\ 0 & 0_{r\times k} \end{bmatrix}, \qquad \mathcal{D}_{21} = \begin{bmatrix} 0_{k\times r} & 0 & 0 \\ 0 & D_{2\theta} & D_{21} \\ I_r & 0 & 0 \end{bmatrix},$$

$$\mathcal{L} = \begin{bmatrix} L & 0 \\ 0 & I_{p1} \end{bmatrix}, \qquad \mathcal{J} = \mathcal{L}^{-1}. \tag{3.42}$$

In (3.39) the unknowns are Ω and certain elements of X, $\mathcal{L}$ and $\mathcal{J}$, the following algorithm shows how (3.39) may be used to construct a controller.

First we consider finding a solution X from (3.25) given S and R. From (3.25)

$$MN^T = I - RS. \tag{3.43}$$

Using the singular value decomposition of $I - RS$,

$$I - RS = \underbrace{U\Sigma^{\frac{1}{2}}}_{M}\underbrace{\Sigma^{\frac{1}{2}}V^T}_{N^T} = MN^T. \tag{3.44}$$

By identification of terms, we have matrices M and N^T and

$$k = \operatorname{rank}(I - RS). \tag{3.45}$$

Again from (3.25) since M is full rank, we have

$$E = -RN^T M(M^T M)^{-1} \tag{3.46}$$

which completely determines X.

We also need L which may be found from L_3 and J_3 in a similar manner. From (3.11)

$$\begin{aligned} L &:= \begin{bmatrix} L_1 & L_2 \\ L_2^T & L_3 \end{bmatrix}, \\ J &:= \begin{bmatrix} J_1 & J_2 \\ J_2^T & J_3 \end{bmatrix} = L^{-1}. \end{aligned} \tag{3.47}$$

It follows that

$$I - L_3 J_3 = L_2^T J_2 \tag{3.48}$$

and J_2 and L_2 may be found from the singular value decomposition of $I - J_3 L_3$. Note that in general J_3 is a block diagonal matrix where the number and size of blocks is determined by the number of parameters and how many times each parameter is copied. Therefore, compute the singular value decomposition of each diagonal block and then take these decompositions and build L_2 and J_2. For example, let $U_i \Sigma_i V_i^T$ be the singular value decomposition of $(I - L_{3_i} J_{3_i})$ which corresponds to the i^{th} block of $(I - L_3 J_3)$, then L_{2_i} and J_{2_i} can be computed as

$$\underbrace{U_i \Sigma_i^{\frac{1}{2}}}_{L_{2_i}^T} \underbrace{\Sigma_i^{\frac{1}{2}} V_i^T}_{J_{2_i}} = \text{svd}(I - L_{3_i} J_{3_i}).$$

Therefore $L_2 = \text{diag}(L_{2_1}, \ldots, L_{2_{n_\theta}})$ and $J_2 = \text{diag}(J_{2_1}, \ldots, J_{2_{n_\theta}})$. Then since

$$\begin{aligned} L_1 J_2 &= -L_2 J_3, \\ L_1 &= -L_2 J_3 J_2^T (J_2 J_2^T)^{-1} \end{aligned} \tag{3.49}$$

which completes the determination of L.

The controller parameters Ω are the only remaining unknowns in (3.39) so that (3.39) may be solved as an LMI for Ω.

There are various other explicit methods of determining the controllers. These methods are described in [34, 33, 35]. In particular in [35], efficient and numerically reliable formulas are presented for both the full-order and reduced-order cases. These controllers are simple extensions to the standard H_∞ central controllers. The advantage of these controllers are in their efficiency and numerical reliability. If there are no other constraints on the controller or the closed-loop properties of the system, then these explicit formulas can be used. If on the other hand further constraints that can be expressed as LMI's are placed on the controller, then the above method to compute the controller should be used. Computing the controller as the solution to an LMI, offers the flexibility of adding more constraints to the controller. The disadvantage is that it is much more computationally intensive and tends to be numerically ill-conditioned. Section 3.6 addresses methods of reducing the numerical problems related to the above method for solving for the controller. There is an assumption that the LFT realization of the controller always exists but in fact it may not. Both Apkarian and Gahinet [8] and Packard [36] discuss this situation and provide work arounds to this problem.

3.5 Comparison of H_∞ and LPV Controllers

It is important to also know whether or not an H_∞ gain scheduled controller provides better performance than a single LTI robust controller. If it does not, then there is no reason to go to all the extra trouble to gain schedule. It is not clear that the H_∞ gain scheduled controller will take into account the knowledge of the parameters, since H_∞ theory is predicated on reducing the affects of system uncertainty and the system parameters are treated as uncertainties.

To see how the H_∞ gain scheduling problem does in fact provide better performance, first consider the LTI robust control problem. Given the plant realization $P(s)$ in (3.13), find a parameter independent LTI controller $K(s)$ such that

$$\max_{\|\Theta\|_\infty \le \frac{1}{\gamma}} \|F_u(F_l(P(s), K(s)), \Theta)\|_\infty < \gamma \tag{3.50}$$

By applying the small gain theorem, a sufficient condition for solvability is the existence of a scaling matrix $L \in L_\Theta$ and of an LTI controller $K(s)$ such that

$$\left\| \begin{bmatrix} L^{\frac{1}{2}} & 0 \\ 0 & I \end{bmatrix} F_l(P(s), K(s)) \begin{bmatrix} L^{-\frac{1}{2}} & 0 \\ 0 & I \end{bmatrix} \right\|_\infty < \gamma \tag{3.51}$$

Recall that Theorem 3.4 addressed this problem and provides the following solvability conditions.

Theorem 3.6. *Assuming A1)-A2), problem (3.51) is solvable if and only if there exists a quadruple* (R, S, L_3, J_3) *satisfying (3.27)-(3.30) together with*

$$L_3 J_3 = I_r \tag{3.52}$$

Proof. Same as the proof of Theorem 3.4 except replace the scalings L and J with $\begin{bmatrix} L_3 & 0 \\ 0 & I_{p_1} \end{bmatrix}$. $\begin{bmatrix} J_3 & 0 \\ 0 & I_{p_1} \end{bmatrix}$, respectively. □

Comparing Theorems 3.5 and 3.6, the only difference is in the condition placed on L_3 and J_3. Theorem 3.5 states that L_3 and J_3 must satisfy (3.30) which is equivalent to

$$L_3 > 0, \quad J_3 > 0, \quad \lambda_{min}(L_3 J_3) \ge 1. \tag{3.53}$$

The interpretation of this is that the set of admissible scaling matrices is any positive scalings L_3, J_3 such that

$$\lambda_i(L_3 J_3) \geq 1, i = 1, \cdots, r, \tag{3.54}$$

while the set of admissible scaling matrices for robust LTI controller synthesis is any positive scaling L_3, J_3 such that

$$\lambda_i(L_3 J_3) = 1, i = 1, \cdots, r. \tag{3.55}$$

Clearly, the robust LTI set described by (3.55) is smaller than the set described by (3.54). In fact, (3.55) is the boundary set of the set (3.54). Since the set of scalings used for H_∞ gain scheduling is larger than that for robust LTI control, it will typically provide controllers with better performance.

3.6 Numerical Issues of Controller Computation

A discussion on some of the numerical issues related to solving for an H_∞ gain scheduled controller would be helpful, since one of the major objectives is to provide control engineers with practical methods on designing gain scheduled controllers. As mentioned earlier powerful toolboxes have been developed to solve LMI equations. Two packages, in particular, are compatible with MATLAB. One is LMI-LAB [32], which is a MATLAB toolbox. The other is SDP Solver [15] developed by Professor S. Boyd at Stanford. This package is extremely easy to use. SDP Solver accepts as input and writes as output `.mat` files. Therefore, it can easily be used inside MATLAB m-files.

Although there are very powerful and efficient LMI solver packages available, our particular problem tends to be ill-conditioned and care must be taken to avoid numerical problems. Even though one of the advantages of the LMI formulation of the H_∞ problem is that it relaxes the constraint that D_{12} and D_{21} must be full column and row rank respectively, for practical controller solutions, these matrices must be full rank with reasonable values. Otherwise, controllers with large poles will result since it does not believe there is any measurement noise. The observer portion of the controller acts as a differentiator, there by resulting in an extremely fast controller which is not practically implementable.

As one iterates solving the necessary and sufficient conditions stated in Theorem 3.5 to find γ that is close to optimal, the quadruple (R, S, J_3, L_3) tends to have large condition numbers. In fact, solving Theorem 3.5 for large values of γ will result in an ill-conditioned quadruple (R, S, J_3, L_3) if precautions are not taken. The reason is because the LMI solver stops as soon as all of the LMI conditions are satisfied. This will occur right on the boundary for at least one of the matrices in the quadruple. For example, say that

$$\begin{bmatrix} R & I \\ I & S \end{bmatrix} \geq 0$$

is the active constraint. Implied by this LMI constraint are two more LMI constraints: $R \geq 0$ and $S \geq 0$. Again assume that $R \geq 0$ is the active constraint, then the LMI solver will stop as soon as it has found an $R \geq 0$. This R will be near singular since it will be extremely close to the boundary. This near singular R will result in an X that is even more ill-conditioned due to the use of matrix dilation to compute it. This ill-conditioned X may result in the control designer not being able to solve for Ω in (3.39) due to numerical problems.

To mitigate this problem the LMI solver needs to search for a quadruple that is not ill-conditioned and yet satisfies all of the constraints. This will require more than just adding a lower bound to (3.29) and (3.30) because the condition number of a matrix is computed as the ratio of the maximum singular value over the minimum singular value. Adding a lower bound will not necessarily effect this ratio since the maximum singular value is not also constrained. The method by which to constrain the maximum singular value is to use the objective function or, in terms of Linear Quadratic control, the cost function. If the objective function is

$$F_{obj}(R, S, J_3, L_3) = \text{Trace} \begin{bmatrix} R & & & \\ & S & & \\ & & J_3 & \\ & & & L_3 \end{bmatrix} \tag{3.56}$$

then the LMI solver will solve for a quadruple that minimizes this trace. Since the trace of a matrix is the sum of the eigenvalues of that matrix, this will minimize the maximum eigenvalue which for positive definite matrices also minimizes the maximum singular value.

Another problem that arises when solving these LMI constraints is that the solver package will return an error indicating that the unknown variables are linearly dependent. One method to eliminate this problem is to begin with a large guess for γ and then continue reducing γ until it has reached its optimum value with in a tolerance or until $\gamma \leq 1$ assuming that the parameters were scaled to have norm bound less than 1. On the first iteration with the large γ the initial guess for Ω is not crucial but on successive iterations use the previous Ω for the initial guess. In addition, one or more elements of Ω may have to be constrained. This can be accomplished by imposing equality constraints on several elements of Ω. The idea is to constrain as few elements as possible in order to find a quadruple. As γ is significantly reduced and good initial guesses for Ω are obtained, the number of elements of Ω that are constrained can generally be reduced. When constraining elements of Ω, constrain them to have the value of the corresponding elements of Ω from the previous successful iteration. Since it is desired to have all of the elements of Ω to be free and not to be constrained, it is necessary to vary which elements are constrained. This has the affect of allowing all of the variables to vary over several iterations but in any one particular iteration several elements may be constrained.

CHAPTER 4
N-D Realization

The problem of realizing a linear parameter-varying (LPV) system, whose state space model is a rational function of the elements, as an LFT is directly connected to N-D system realization. N-D system realization has proven to be a difficult area of research especially with respect to finding the minimal realization. In this chapter, we begin by illustrating how realization theory may be used to derive a representation for a LPV system.

The LPV plant description in terms of state variables is given (see chapter 2) as

$$\begin{bmatrix} \dot{x}(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A(\theta) & B(\theta) \\ C(\theta) & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \tag{4.1}$$

In the following we will consider $u(t)$ as a general input which includes the disturbance inputs (w in chapter 2). We will also consider that $y(t)$ includes the usual $y(t)$ outputs and the controlled or performance outputs $q(t)$.

Assuming θ is constant, we may derive a transfer function representation from (1) as

$$Y(s) = C(\theta)(sI - A(\theta))^{-1}B(\theta)U(s) \tag{4.2}$$

where the transfer function

$$C(\theta)(sI - A(\theta))^{-1}B(\theta) \tag{4.3}$$

is a rational function of s. Extending this idea to the functional dependence of the matrices $A(\theta)$, $B(\theta)$, and $C(\theta)$ on θ we consider the relation

$$q = F(\theta)p \tag{4.4}$$

and ask when we can write

$$F(\theta) = C_F(\Theta^{-1} - A_F)^{-1}B_F + D_F \tag{4.5}$$

where now we take Θ as a matrix whose elements are linear functions of the parameters θ. By a slight abuse of notation, henceforth θ (without the boldface) will be used to refer to the *matrix* (previously Θ) of parameters. If $F(\theta)$ can be written as (4.5) exactly or approximately, then (4.4) can be rewritten as

$$q = C_F(\theta^{-1} - A_F)^{-1}B_F p + D_F p \tag{4.6}$$

Define

$$r \;=\; (\theta^{-1} - A_F)^{-1}B_F p \tag{4.7}$$

which may be written as

$$\theta^{-1}r \;=\; A_F r + B_F p \tag{4.8}$$

Further define w by the relation $r = \theta w$ which leads to the set of equations

$$q \;=\; C_F r + D_F p \tag{4.9}$$

$$w \;=\; A_F r + B_F p \tag{4.10}$$

$$r \;=\; \theta w \tag{4.11}$$

Hence (4.9), (4.10), and (4.11), which are linear in θ, may be used to replace (4.4) with the assumption that $F(\theta)$ is a rational function of θ.

Applying this idea to the matrices in (4.1) is simplified by the definition

$$F(\theta) = \begin{bmatrix} A(\theta) & B(\theta) \\ C(\theta) & 0 \end{bmatrix} \tag{4.12}$$

Then

$$\begin{bmatrix} \dot{x}(t) \\ y(t) \end{bmatrix} = F(\theta) \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \tag{4.13}$$

becomes

$$\begin{bmatrix} \dot{x}(t) \\ y(t) \end{bmatrix} = C_F r + D_F \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \tag{4.14}$$

$$w = A_F r + B_F \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} \tag{4.15}$$

$$r = \theta w \tag{4.16}$$

In general the matrix θ may be a linear function of many parameters. Therefore, assuming (4.5) holds, we seek matrices (A_F, B_F, C_F, D_F) given the matrix function $F(\theta)$. Comparison of (4.5) with the usual formula for a transfer function of a L.T.I system with matrices (A, B, C, D)

$$C(sI - A)^{-1}B + D$$

reveals that sI has been replaced by θ^{-1}. There are two problems to consider here. The first is to extend realizations to the case where θ^{-1} is not a constant times the identity. The second is to consider the selection of "small" matrices which work, *i.e.* a minimal realization, primarily due to the large increase in computational requirements with the dimension of θ. In the following sections this chapter discusses non-minimal realizations and then turns to minimal-realizations.

4.1 Non-Minimal Realizations

One possible realization of (4.4) is to realize each element of the matrix $F(\theta)$ separately. To illustrate the idea, we begin with a few simple examples.

To realize the scalar function

$$F(a,b,c) = ab + a^2 + c \tag{4.17}$$

by inspection we construct the following linear equations:

$$w = ap \tag{4.18}$$

$$q = bw + aw + cp \tag{4.19}$$

Then the input p and output q in (4.18) and (4.19) realize the relationship in (4.17).

For a more involved example, suppose we wish to realize the multinomial matrix

$$\begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} ab + a^2 + c & ac + b \\ ab + c & b^2 + a \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{4.20}$$

The problem of finding a minimal realization here is equivalent to computing p and q from x and y with the minimum number of multiplications. As a start, we might search for a common factor in the rows and then search for a common factor in the columns. We note there are none. However, we can search for common factors in groupings of terms. The term

$$(ab + c)x$$

appears in computing p and q so it needs to be computed only once. Furthermore, the term ax appears twice. Hence terms in the first column may be computed with

$$\alpha = a \times x \tag{4.21}$$

$$q\_first = \alpha \times b + c \times x \tag{4.22}$$

$$p\_first = q + \alpha \times a \tag{4.23}$$

where the temporary variable α has been introduced. This representation has four multiplications in place of the nine multiplications which would result with the original equations. The second column terms may be found and added to the other terms by

$$\beta = b \times y \tag{4.24}$$

$$\gamma = a \times y \tag{4.25}$$

$$p = p\_first + c \times \gamma + \beta \tag{4.26}$$

$$q = q\_first + b \times \beta + \gamma \tag{4.27}$$

with the new temporary variables β and γ. Viewed in this light, the reduction of calculations involved in the non-minimal realizations is related to a problem in optimizing compilers. An optimizing compiler attempts to find a minimal sequence of operations to evaluate a given expression by recognizing and exploiting common terms and by the clever introduction of temporary variables. It might prove fruitful to see if the techniques used by optimizing compilers could lead to algorithms for minimal realizations.

It is clear to see how we can realize any $F(\theta)$ which is a multinomial in θ. If $F(\theta)$ is a scalar rational function of θ,

$$F(\theta) = \frac{n(\theta)}{d(\theta)} \tag{4.28}$$

where $n(\theta)$ is a multinomial in θ and $d(\theta)$ is of the form $1 - D(\theta)$ where $D(\theta)$ is a multinomial in θ, then a simple technique works.

We may realize $n(\theta)$ and $D(\theta) = 1 - d(\theta)$ as before and then construct $F(\theta)$ by the feedback formula as follows:

$$q = e \tag{4.29}$$

$$e = n(\theta)p - D(\theta)q \tag{4.30}$$

Currently minimal realizations exist for the following special cases: a) 2-D transfer functions with separable numerator [37], [13]; b) 2-D transfer functions with separable denominator [38], [13]; c) 2-D all-pole and all-zero [39], d) 2-D, 3-D, and N-D systems that can be expanded into a continued fraction expansion (CFE) [40], [41], [42], [43], [44], [45]. It should be noted that all of these special case methods apply to the single-input single-output case (SISO) systems. For a missile autopilot design and for many other practical systems a multi-input multi-output (MIMO) method is required.

In [46] Koonar and Sohal present a more general method for 2-D realization using Markov parameters and a moments of impulse response matrix. In [47] Eydgahi provides a counter example to Koonar and Sohal's method. After searching the literature, no response has been discovered to this counter example. Another limitation to Koonar and Sohal's method is that it requires that the total degree of the numerator polynomial should at most be equal to that of the denominator polynomial. This is a drawback for our case since, in many instances, linear multi-parameter-systems are curve fitted with N-D polynomials. This results in LPV systems whose elements are multivariate polynomials in the parameters, for which Koonar and Sohal's method would not be valid. Recently, general nonminimal realization methods have been developed for both SISO and MIMO N-D systems.

4.2 General Nonminimal Realization Methods

Lambrecht's paper [48] presents a general method by which to compute an LFT system description of a linear system with parametric uncertainties. The entries of the state-space matrices are bounded and can be given as real-rational N-D polynomials in the parameters. His method begins by modeling the most basic element as an LFT, i.e. an individual term of an N-D polynomial. Once the most basic element is modeled they then exploit the fact that parallel combinations of LFTs, cascading of LFTs, and inverses of LFTs, all result in LFTs. From these properties, they develop one LFT for the entire system. This LFT will be quite large since this method does not take into account any redundancy. They then reduce this large LFT by treating the parameters individually as states and then extracting out the locally "uncontrollable" and "unobservable" parameters. This will reduce the LFT substantially but it is not clear how close the resulting LFT will be to minimal.

Cheng and DeMoor [49], [50] provide an algorithm to systematically solve the N-D realization problem. They do not provide a minimal realization but a systematic method to obtain a realization. The authors do suggest how to reduce the order of the realization which requires the solution of a set of matrix equations such that the solution matrix has minimal rank. Unfortunately, it is still an open question on how to find the minimal rank solution to this set of equations.

Cheng and DeMoor consider the following N-D realization problem. Given a multivariate rational matrix $F(\theta) \in \mathcal{R}^{n\times m}$, find an LFT with block structure $b_s = [r_1, r_2, \ldots, r_q]$ with r_i being an integer and a coefficient matrix $L \in \mathcal{R}^{(n+r)\times(m+r)}$, $r = \sum_{i=1}^{q} r_i$, partitioned as

$$L = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} \tag{4.31}$$

such that

$$F(\theta) = L_{22} + L_{21}\theta(I - \theta L_{11})^{-1}L_{12} \tag{4.32}$$

Cheng and DeMoor assumed the structure for θ was diagonal

$$\theta = \mathrm{diag}(\theta_1 I_{r_1} \dots \theta_q I_{r_q}) \tag{4.33}$$

The elements of $F(\theta)$ are further assumed to be of the following form

$$[F(\theta)]_{ij} = c_{ij} + f_{ij}(\theta) \tag{4.34}$$

with c_{ij} being constant and $f_{ij}(\theta)$ rational multinomials of the form

$$\frac{k(\theta)}{1 + l(\theta)},\ k(0) = l(0) = 0. \tag{4.35}$$

and $k(\theta)$ and $l(\theta)$ are multinomials in the elements of θ. Given the form (4.34), $F(\theta)$ can be written as

$$F(\theta) = F_0 + \sum_{i=1}^{l} f_i(\theta)F_i \tag{4.36}$$

where F_0 is a constant matrix with entries c_{ij}. The functions $f_i(\theta)$, $i = 1, \dots m$ are formed from the elements $f_{ij}(\theta)$ of $F(\theta)$ such that for each pair i, j there is a k such that $f_k(\theta) = f_{ij}(\theta)$. Finally F_i is the constant coefficient matrix that tells how $f_i(\theta)$ enters $F(\theta)$.

The algorithm for constructing a realization starts by first realizing (4.36) which is a linear function of the $f_i(\theta)$'s. That is we find a coefficient matrix L_a so

$$F(\theta) = L_{a_{22}} + L_{a_{21}}\Delta(f)(I - L_{a_{11}}\Delta(f))^{-1}L_{a_{12}} \tag{4.37}$$

where

$$L_a = \begin{bmatrix} L_{a_{11}} & L_{a_{12}} \\ L_{a_{21}} & L_{a_{22}} \end{bmatrix}, \; b_{s_a} = [r_{a_1}, r_{a_2}, \ldots, r_{a_q}] \tag{4.38}$$

$$\Delta(f) = \mathrm{diag}(f_1(\theta)I_{r_{a_1}} \ldots f_q(\theta)I_{r_{a_q}}) \tag{4.39}$$

Safanov and Athans' Internal Feedback Loop ([51]) parameter representation is used to find L_a. A realization with $L_{a_{11}} = 0$ can be easily found by first factoring each of the F_i matrices in (4.36) as

$$F_i = U_i \Sigma_i V_i^T$$

by singular value decomposition with Σ_i a $r_{a_i} \times r_{a_i}$ positive definite diagonal matrix. Then if

$$\begin{aligned} U &= [U_1\Sigma_1^{\frac{1}{2}}, U_2\Sigma_2^{\frac{1}{2}}, \ldots, U_l\Sigma_l^{\frac{1}{2}}] \\ V^T &= [\Sigma_1^{\frac{1}{2}}V_1^T, \Sigma_1^{\frac{1}{2}}V_2^T, \ldots, \Sigma_1^{\frac{1}{2}}V_l^T] \\ F(\theta) &= F_0 + U\Delta(f)V^T \end{aligned} \tag{4.40}$$

Comparison with (4.37) gives

$$\begin{aligned} L_{a_{22}} &= F_0 \\ L_{a_{21}} &= U \\ L_{a_{12}} &= V^T \\ L_{a_{11}} &= 0 \end{aligned} \tag{4.41}$$

The next step involves finding a realization for $\Delta(f)$ as

$$\Delta(f) = L_{b_{22}} + L_{b_{21}}\Delta(\theta)(I - L_{b_{11}}\Delta(\theta))^{-1}L_{b_{12}}$$

Assume that each (scalar) rational function $f_i(\theta)$ has a realization L_{f_i}. Then the realization L_b above can be obtained by combining these realizations. The bookkeeping for this procedure is a bit tedious but possible. By the assumption $f_i(0) = 0$, $\Delta(f)_{\theta=0} = 0$ so that $L_{b_{22}} = 0$. The final answer becomes

$$L = \begin{bmatrix} L_{b_{11}} & L_{b_{12}} L_{a_{12}} \\ L_{a_{21}} L_{b_{21}} & L_{a_{22}} \end{bmatrix}$$

It is clear that this realization is by no means minimal, however Cheng and DeMoor give some suggestions for reducing the complexity.

In a recent paper, Belcastro et. al. [11] present an overview of a matrix-based method for low-order realization of a plant that has a multivariate rational functional dependence on uncertain parameters. The full details of this method are given in Belcastro's dissertation [52]. The claim of the authors is that this method is efficient and produces a lower order realization than other methods that use reduction techniques such as removing unobservable and uncontrollable modes.

Belcastro, et. al., used the same assumptions and problem formulation presented earlier with the exception that the $F(\theta)$ matrix was given by

$$F(\theta) = \begin{bmatrix} A(\theta) & B(\theta) \\ C(\theta) & D(\theta) \end{bmatrix}$$

and the parameter vector θ was defined in terms of a nominal parameter value θ_0 as

$$\theta = \theta_0 + \delta$$

Although Belcastro, et.al. do not propose to find a minimal solution, they attempt to find a realization with a small dimension. Expanding the assumed form for $F(\theta)$ in terms of the nominal parameters θ_0 and a perturbation δ yields the following LFT formulation for $F(\theta)$.

$$F(\theta) = P_{22} + P_{21}(I - \theta P_{11})^{-1}\Delta P_{12} = F(\theta_0) + F_\Delta(\delta) \tag{4.42}$$

where

$$F_\Delta(\delta) = L_{21}(I - \Delta(\delta)L_{11})^{-1}\Delta(\delta)L_{12} \tag{4.43}$$

If the state matrices $A(\theta)$, $B(\theta)$, $C(\theta)$ and $D(\theta)$ are multinomial functions of θ. straightforward algebra can be used to determine the matrices in $F_\Delta(\delta)$. From (4.42) we have

$$\begin{aligned} F(\theta) &= \begin{bmatrix} A(\theta) & B(\theta) \\ C(\theta) & D(\theta) \end{bmatrix} && (4.44) \\ &= \begin{bmatrix} A(\theta_0+\delta) & B(\theta_0+\delta) \\ C(\theta_0+\delta) & D(\theta_0+\delta) \end{bmatrix} && (4.45) \end{aligned}$$

which may be expanded to separate the nominal and perturbed parts of the system.

In order to equate (4.43) with the perturbed part of $F(\theta)$ from (4.45), $F_\Delta(\delta)$ must be a multinomial function of θ which implies that the expension of $F_\Delta(\delta)$ terminates or $\Delta(\delta)L_{11}$ is nilpotent with index of nilpotency r,

$$(\Delta(\delta)L_{11})^{r+1} = 0$$

and

$$(I - \Delta L_{11})^{-1} = I + (\Delta L_{11}) + (\Delta L_{11})^2 + \ldots (\Delta L_{11})^r \tag{4.46}$$

However if the state matrices are rational multinomial functions the expansion becomes more difficult. In her dissertation [52], Belcastro proposed first representing the system in a matrix fraction description MFD and presents a scheme for the factorization of a rational $S_\Delta(\delta)$ as a matrix fraction.

$$S_{\Delta}(\delta) = S_{N_{\Delta}}(\delta) S_{D_{\Delta}}(\delta)^{-1} = \tilde{S}_{D_{\Delta}}(\delta)^{-1} \tilde{S}_{N_{\Delta}}(\delta), \tag{4.47}$$

where $S_{N_{\Delta}}(\delta)$, $S_{D_{\Delta}}(\delta)$ $\tilde{S}_{D_{\Delta}}(\delta)$ and $\tilde{S}_{N_{\Delta}}(\delta)$ are all multinomial matrices.

By substituting (4.46) into (4.43), and then comparing the result with the $S_{\Delta}(\delta)$ computed from the MFD and then broken out such that like terms can be computed, results in a set of equations that can then be used to solve for L_{11}, L_{12} and L_{21}. If a solution does not exist then the order of the offending δ_i is increased by 1 until a solution is found. By increasing the order of Δ only when it is necessary to obtain a solution, ensures that a low-order realization is found. It is not clear, however, whether or not the solution takes into account any redundancy or possible cancellations between $S_{N_{\Delta}}$ and $S_{D_{\Delta}}$.

4.3 Minimal/Near Minimal 2-D Realization

Currently two methods of realization show promise in providing minimal/near minimal 2-D realization. Realization methods for 1-D systems have, as an underlying principle, the fact that a minimal realization is both observable and controllable. Therefore one can take a nominal 1-D system, test for controllability and observability and then obtain a minimal realization by throwing away the uncontrollable and unobservable states. Kung et.al. [13] showed that the notions of controllability and observability, although extending to 2-D systems, do not provide a test for minimality. In fact Kung et.al. provide an example of a controllable and observable system that is not minimal. In their paper, they investigate several ideas on determining if a system is minimal. In the end they extend Rosenbrock's minimality test to 2-D systems which will also hold for N-D systems as well. This result states that a plant

$$\begin{aligned} P &= C\Theta \left[I - A\Theta\right]^{-1} B \\ &= C\left[\Theta^{-1} - A\right]^{-1} B, \end{aligned} \tag{4.48}$$

where Θ is as defined previously, is minimal if and only if $[\Theta^{-1} - A], B$ are left coprime and $C, [\Theta^{-1} - A]$ are right co-prime.

There are two difficulties with this definition of minimality. One is that it allows the A, B and C matrices to have complex gains and the other is that Youla and Gnavi [53] state that certain decomposition techniques, which have proven basic for 1-D and 2-D systems, are no longer applicable for 3-D and larger systems. Therefore finding techniques to compute greatest common divisors (GCD) for 3-D and larger systems will be difficult. For 2-D systems though Morf et. al. [12] have developed methods to compute greatest common right (left) divisors (GCRD/GCLD). We have implemented Morf et.al's GCD extraction method in MAPLE©. The difficulty that Kung et.al. had in implementing the extraction of GCDs to reduce non-minimal state descriptions to minimal ones, was that after the GCD had been extracted the plant description was no longer in "state space" form (i.e. parameters only on the diagonal) and they did not know how to return it to "state space" form.

We discovered that Rosenbrock in [14] introduced the concepts of the system matrix and system equivalence for 1-D systems. Rosenbrock defined for 1-D systems a system matrix by writing the Laplace transformed state equations in the following form.

$$\underbrace{\begin{bmatrix} T(s) & U(s) \\ -V(s) & W(s) \end{bmatrix}}_{\mathcal{S}} \begin{bmatrix} -x(s) \\ u(s) \end{bmatrix} = \begin{bmatrix} 0 \\ y(s) \end{bmatrix} \tag{4.49}$$

where $\mathcal{S}$ is defined as the system matrix. Note that the dimension of $T(s)$ is $n \times n$ Two system matrices are system equivalent if and only if they give rise to the same transfer function matrix. The relation of system equivalence is generated by the following operations on system matrices.

(i) Multiply any one of the first n rows(columns) by a rational function not identically zero.

(ii) Add a multiple, by a rational function, of any one of the first n rows(columns) to any other row(column).

(iii) Interchange any two among the first n rows(columns).

(iv) Add a row and a column to $\mathcal{S}$ to form $\begin{bmatrix} 1 & 0 \\ 0 & \mathcal{S} \end{bmatrix}$.

System equivalent operations can be used to return the system matrix back into state-space form after the GCD has been extracted. These concepts and operations extend readily to N-D systems by replacing s with Θ^{-1}. The purpose for replacing s with Θ^{-1} has to do with the nature of LFTs. For a linear system, the integrators (s^{-1}) enter in a linear fractional manner and for our LPV system, Θ and not Θ^{-1} enters in a linear fractional manner. Therefore s and Θ^{-1} are similar. Compare (4.48) with $C\left[sI - A\right]^{-1} B$.

4.3.1 GCD Based Algorithm

By performing system equivalent operations, one can put the system matrix back into state-space form. Below is an algorithm to obtain a 2-D minimal/near minimal realization.

Algorithm 4.1 (GCD Based). .

1. *Obtain a nonminimal description of the system.*

2. *Using [12] 2-D GCD extraction method, extract the GCLD of* $[\Theta^{-1} - A]$, B. *If these are already coprime then extract the GCRD of* C, $[\Theta^{-1} - A]$. *After extraction the system matrix is no longer in "state space" form. Its new general form is notated as* $\mathcal{S} = \begin{bmatrix} T & U \\ -V & W \end{bmatrix}$.

3. *Use system equivalent operations to return the T matrix back into "state space" form (i.e. parameters only on the diagonal). This will yield a new system matrix to be* $\mathcal{S} = \begin{bmatrix} T_1 & U_1 \\ -V_1 & W_1 \end{bmatrix}$.

4. *Solve* $V_s T_1^{-1} U_s - V_1 T_1^{-1} U_1 - W_1 = 0$ *for* V_s *and* U_s *such that they contain only real gains. Note that* T_1 *is already in "state space" form while* U_1, V_1 *and* W_1 *are most likely not.*

5. *If no real gain solution exists then introduce another parameter and expand the* T_1 *matrix up one dimension.*

6. *Repeat steps 4 and 5 until a real gain solution is obtained. The real gain solution will be the minimal/near minimal solution that is desired.*

A difficulty is that the GCD extraction process, in general, reduces the system too far since it allows for complex gains. For example, there are 2-D systems that are real gain minimal but, when Rosenbrock's minimality test is applied, are not found to be minimal. The reason is that there exist a complex gained system that is of smaller order than the real gain system. This is a serious problem in the search for minimal realization algorithms for N-D systems. The above algorithm manages this problem in step 5 by expanding up another dimension. The key here is on intelligently expanding so that you can get step 4 to solve with a real gain solution with as few of parameters as possible. Work is currently being conducted on how to best expand the system.

4.3.2 System Equivalent Based Algorithm

The second method, which shows promise for the general case, is to begin with the plant P in the lower right corner of Rosenbrock's system description matrix

$$\begin{bmatrix} I & 0 \\ 0 & P \end{bmatrix}$$

The idea is to move P over to the left using system equivalent operations.

$$\begin{bmatrix} I & I \\ -P & 0 \end{bmatrix}$$

From here we can multiply the column corresponding to the last column of P by the least common multiple of the denominators of P and then expand it up until all parameters are only on the diagonal of T.

$$\begin{bmatrix} T & U \\ -V & W \end{bmatrix}$$

Once T only has parameters on the diagonal it is time to begin eliminating the parameters out of U and V. At this point we can use step 4 of the GCD based algorithm to see if there exists a real gain solution. If there is not then we continue expanding until a real gain solution exists. The following is an algorithm for 2-D systems that implements this method.

Algorithm 4.2 (System Equivalent Based). .

1. *Let* $H(z,w) = \begin{bmatrix} A(z,w) & B(z,w) \\ C(z,w) & D(z,w) \end{bmatrix}$ *where* z *and* w *are the parameters.* $H(z,w)$ *must have a constant term in its collective denominator if it does not then create one by a change of variable.*

2. *Substitute* $z = \frac{1}{z_i}$ *and* $w = \frac{1}{w_i}$ *into* $H(z,w)$.

3. *Place $H(z_i, w_i)$ into the W position of Rosenbrocks system matrix.*

4. *By use of system equivalent operations move $H(z_i, w_i)$ over one position to the left.*

5. *Clear the fraction of the last column of $H(z_i, w_i)$ by multiplying the corresponding column by $d(z_i, w_i)$ which is the least common multiple of denominators of $H(z_i, w_i)$. Now the T matrix contains the least common multiple of the denominators.*

6. *Expand the T matrix up until it is in "state space" form. This can be done by moving $d(z_i, w_i)$ over to the left one space and at the same time depositing a z_i in its place on diagonal. Then by column multiplication eliminate all of the z_i's from the $d(z_i, w_i)$. This will place a new $d_1(z_i, w_i)$ with the order of z_i reduced by one on the diagonal above. Continue depositing z_i's until they are completely eliminated and then by the same process place w_i's on the diagonal until they are eliminated.*

7. *Solve $V_s T^{-1} U_s - V T^{-1} U - W = 0$ for V_s and U_s such that they contain only real gains. Note that T is already in "state space" form while U, V and W are most likely not.*

8. *If no real gain solution exists then introduce another parameter and expand the T_1 matrix up one dimension.*

9. *Repeat steps 7 and 8 until a real gain solution is obtained. The real gain solution will be the minimal/near minimal solution that is desired.*

Due to the special structure that this algorithm creates by placing 1's on the super-diagonal, it is not completely general for finding the minimal realization but can find near minimal realizations for all systems.

The idea of starting with $\begin{bmatrix} A(\Theta^{-1}) & B(\Theta^{-1}) \\ C(\Theta^{-1}) & D(\Theta^{-1}) \end{bmatrix}$ matrix in the lower right of the system matrix and then expanding up can easily be extended to N-D systems. The above algorithm will always put T in "state space" form for 2-D systems but for N-D systems, in general, it will not.

Both of these methods work on both MIMO and SISO systems. The second method shows promise on obtaining near minimal realization for general N-D systems.

CHAPTER 5
Conclusion

In this report, we focused on the problem of gain scheduling LPV systems. In particular we developed necessary and sufficient conditions for determining the Quadratic stabilizability of LPV systems and provide a stabilizing LPV controller if one exists. Our necessary and sufficient conditions are realistically computable with reduced dimension and with only the elements of the Lyapunov P as the unknown variables where as the one given by Becker in [25] is of full dimension and requires an additional set of variables the size of the controller for each grid point. Examples were given that demonstrated problems that can be encountered with "frozen coefficient" gain scheduling which can be overcome by our method. We investigated the geometric interpretation of the $\tilde{A}_{22}$ condition and, for the second order case, developed a method by which to graphically display the set of all stabilizing Lyapunov functions for an LPV system. Furthermore our solution to the $\tilde{A}_{22}$ condition is usefull in solving Becker and Packards [6] output stabilization problem for which they set up but did not provide a solution.

We extended our results on Quadratic stability to the more general case of General Lyapunov stability. The derivation was the same as for Quadratic stability, except there was an additional term that was a function of $\dot{P}$. We developed a method by which to solve these new conditons. In fact, when Quadratic stabilizability fails, we showed that it was necessary to split the parameter space into regions of frozen Lyapunov functions. After determining all of the necessary frozen Lyapunov functions, these functions are then "stitched" together such that General Lyapunov stability holds. This requires knowledge of the rate of change of the parameter variations. In order to properly stitch these Lyapunov functions together, $\dot{P}$ cannot change so fast that the necessary and sufficient conditions for General Lyapunov stability no longer hold. This fact is taken into account when first determining the frozen regions. Now, instead of the clasical frozen point controllers, we have frozen region Lyapunov functions. From these Lyapunov functions we showed how to construct a stabilizing controller.

In addition, we add performance to our Quadratic stability results for strictly proper systems by guaranteeing that the closed loop system has an induced $\mathcal{L}_2$ norm less than a specified value if a modified $\tilde{A}_{22}$ condition holds. A missile autopilot example was provided.

Another method of gain scheduling LPV systems is via H_∞ based gains scheduling as developed by Apkarian and Gahinet [8]. We reviewed their work and discussed how to construct a controller and how to mitigate the inherent numerical problems associated in solving the LMI's. This method of LPV stabilization does not require gridding the parameter space but also does not take into account the realness of the parameters as does our Lyapunov based method of gain scheduling. It also requires that a realization of the plant with a linear fractional dependence on the parameters be provided.

In the final chapter we discussed methods of finding linear fractional realizations. These realizations are related to N-D realizations for which minimal realizations are very difficult to find. We developed two methods for finding minimal/near minimal realizations for 2-D systems based on the work of Kung et.al. [12, 13] and Rosenbrock [14]. The first method required extracting any greatest common divisors out of a nonminimal realization and then returning the realization back to "state-space" form. This method does not extend to N-D systems since the techniques used to compute the GCD for 2-D sytems does not extend to N-D systems [53]. However the second method, the System Equivalent Based method, can be extended to N-D systems. This method begins with the LPV system description being converted to Rosenbrock's system matrix form. Then the system matrix form is systematically manipulated until it is in "state-space" form.

This report provides the control engineer with a design method of gain scheduling LPV systems that relaxes the requirement of scheduling on slowly varying parameters. Missile systems have been approximated as LPV systems in [54] [55] and [28] to list a few examples. We demonstrated in Section 2.3 that this work can now be applied to the missile autopilot problem. In addition, there are many other succesful applications of classical gain scheduling such as aircraft control systems, turbojet engines, ship steering, pH control, combustion control and fuel-air control in an automobile engine (see for example, [56, 57, 58]) for which our method of Lyapunov based gains scheduling could be applied.

REFERENCES

[1] Wilson J. Rugh. Analytical framework for gain scheduling. *IEEE Control Systems Magazine*, pages 79–84, January 1991.

[2] Jeff S. Shamma and Michael Athans. Gain scheduling: Potential hazards and possible remedies. *IEEE Control Systems Magazine*, 12(3):101–107, June 1992.

[3] Jeff S. Shamma and Michael Athans. Analysis of gain scheduled control of nonlinear plants. *IEEE Transactions on Automatic Control*, 35(8):898–907, August 1990.

[4] S.M. Shahruz and S. Behtash. Design of controllers for linear parameter–varying systems by the gain scheduling technique. *Journal of Mathematical Analysis and Applications*, 168:195–217, 1992.

[5] Di Guo and Wilson J. Rugh. An approach to gain scheduling on fast variables. In *Proceedings of the 31st Conference on Decision and Control*, pages 759–763, December 1992.

[6] Gregory Becker, Andy Packard Doug Philbrick, and Gary Balas. Control of parametrically–dependent linear systems: A single quadratic lyapunov approach. In *American Control Conference*, pages 2795–2799, June 1993.

[7] B.R. Barmish. Necessary and sufficient conditions for quadratic stabilizability of an uncertain system. *Journal of Optimization Theory and Applications*, 46(4):399–408, August 1985.

[8] Pierre Apkarian and Pascal Gahinet. A convex characterization of gain–scheduled h_∞ controllers. *IEEE Transactions on Automatic Control*, 40(5):853–864, May 1995.

[9] Mark Spillman. Proof of apkarian and gahinet's theorem. Personal Comunication, July 1996.

[10] Pierre Apkarian and Pascal Gahinet. Erratum to "a convex characterization of gain-scheduled h_∞ controllers". *IEEE Trans. Auto. Cont.*, 40(9):1681, sept 1995.

[11] Christine Belcastro, B.C. Chang, and Robert Fischl. A matrix approach to low-order uncertainty modelling of real parameters. In *13th World Congress of IFAX*, page ??, July 1996.

[12] Bernard Levy Martin Morf and Sun-Yuan Kung. New results in 2-d systems theory, part i: 2-d polynomial matrices, factorization, and coprimeness. *Proc. of the IEEE*, 65(6):861–872, june 1977.

[13] Martin Morf Sun-Yuan Kung, Bernard Levy and Thomas Kailath. New results in 2-d systems theory, part ii: 2-d state space models – realization and the notions of controllability, observability and minimality. *Proc. of the IEEE*, 65(6):945–961, june 1977.

[14] H. H. Rosenbrock. *State-space and Multivariable Theory.* Wiley, 1970.

[15] Stephen Boyd and Shaopo Wu. sdpsol - a parser / solver for semidefinite programs with matrix structure, user's guide. User's Guide obtained from Internet, Nov 1995.

[16] Andy Packard, Greg Becker, Doug Philbrick, and Gary Balas. Control of parameter–dependent systems: Applications to h^{∞} gain–scheduling. In *1st IEEE Regional Conference on Aerospace Control Systems*, pages 329–333, May 1993.

[17] G. Leitman. Guaranteed ultimate boundedness for a case of uncertain linear dynamic systems. *IEEE Transactions on Automatic Control*, 21(6):1109–1110, 1978.

[18] J. S. Thorp and B. R. Barmish. On guaranteed stability of uncertain linear systems via linear control. *J. Optim. Theory Appl.*, 35(4):559–579, 1981.

[19] Kemin Zhou and Pramod P. Khargonekar. Robust stabilization of linear systems with norm–bounded time–varying uncertainty. *System & Control Letters*, 10:17–20, 1988.

[20] Pramod P. Khargonekar and Mario A. Rotea. Stabilization of uncertain systems with norm bounded uncertainty using control lyapunov functions. In *Proceedings of the 27th Conference on Decision and Control*, pages 503–507A, December 1988.

[21] J. Bernussou, P.L.D. Peres, and J.C. Geromel. A linear programming oriented procedure for quadratic stabilization of uncertain systems. *Systems and Control Letters*, 13:65–72, 1989.

[22] Andy Packard and Gregory Becker. State feedback solutions to quadratic stabilization problems. In *Twenty-Eighth Annual Allerton Conference on Communication, Control, and Computing*, pages 768–769, October 1990.

[23] Andy Packard, Kemin Zhou, Pradeep Pandey, and Greg Becker. A collection of robust control problems leading to lmi's. In *Proceedings of the 30th Conference on Decision and Control*, pages 1245–1250, December 1991.

[24] Andy Packard and Gregory Becker. Quadratic stabilization of parametrically–dependent linear systems using parametrically–dependent linear, dynamic feedback. *Advances in Robust and Nonlinear Control Systems*, DSC-Vol 43,:29–36, 1992.

[25] Gregory Becker. *Quadratic Stability and Performance of Linear Parameter Dependent Systems.* PhD thesis, University of California at Berkeley, 1993.

[26] Mitsuji Sampei, Tsutomi Mita, and Matsuroh Nakamichi. An algebraic approach to h_∞ output feedback control problems. *System & Control Letters*, 14:13–24, 1990.

[27] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, chapter 6 and 7. SIAM, 1994.

[28] Jeff S. Shamma and James R. Cloutier. Gain–scheduled missile autopilot design using linear parameter varying transformations. *Journal of Guidance, Control, and Dynamics*, 16(2):256–263, March-April 1993.

[29] Andy Packard. Gain scheduling via linear fractional transformations. *System & Control Letters*, 22:79–92, 1994.

[30] Munther A. Dahleh and Ignacio J. Diaz-Bobillo. *Control of Uncertain Systems A Linear Programing Approach.* Prentice Hall, ?

[31] F. R. Gantmacher. *The Theory of Matrices.* Chelsea Publishing Company, New York, N.Y., 1960.

[32] P. Gahinet, A. Nemirovskii, A. J. Laub, and M. Chilali. *LMI Control Toolbox.* MathWorks, 1994.

[33] Pascal Gahinet and Pierre Apkarian. A linear matrix inequality approach to h_∞ control. *International Journal of Control*, pages 421–448, 1994.

[34] Pascal Gahinet. A convex parametrization of h_∞ suboptimal controllers. In *Proceedings of the 31st Conference on Decision and Control*, pages 937–942, December 1992.

[35] Pascal Gahinet. Explicit controller formulas for lmi-based H_∞ synthesis. In *Proceedings of the American Control Conference*, pages 2396–2400, June 1994.

[36] Andy Packard. Gain scheduling via linear fractional transformations. *System & Control Letters*, 22:74–92, 1994.

[37] Guoxiang Gu, Jorge Aravena, and Kemin Zhou. On minimal realization of 2-d systems. *IEEE Transactions on Circuits and Systems*, 38(10):1228–1233, October 1991.

[38] T. Hinamoto and Maekawa. Canonic form state-space realization of 2-d transfer functions having separable denominator. *Int J. Syst. Sci.*, 13:177–184, 1982.

[39] S.J. Varoufakis, P.N. Paraskevopoulos, and G.E. Antoniou. On the minimal state-space realizations of all-pole and all-zero 2-d systems. *IEEE Transactions on Circuits and Systems*, CAS-34(3):289–292, March 1987.

[40] G. Subba Rao, P. Karivaratharajan, and K.P. Rajappan. On realization of two–dimensional digital–filter structures. *IEEE Transactions on Circuit and Systems*, CAS-23(7):479, 1976.

[41] S.K. Mitra, A.D. Sagar, and N.A. Pendergrass. Realizations of two–dimensional recursive digital filters. *IEEE Transactions on Circuit and Systems*, CAS-22(3):177–184, March 1975.

[42] G.E. Antoniou, S.J. Varoufakis, and P.N. Paraskevopoulos. State–space realization of 2-d sytems via continued fraction expansion. *IEEE Transactions on Circuits and Systems*, CAS-33(9):926–930, September 1986.

[43] P.N. Paraskevopoulos, G.E. Antoniou, and S.J. Varoufakis. Minimal state space realisation of 3-d systems. *IEE Proceedings*, 135, Pt. G(2):65–70, April 1988.

[44] G.E. Antoniou. Minimal state space realization of n–dimensional systems via continued fraction expansions. *Control–Theory and Advanced Technology*, 7(1):129–145, March 1991.

[45] S.H. Mentzelopoulou and N.J. Theodorou. N-dimensional minimal state–space realization. *IEEE Transactions on Circuits and Systems*, 38(3):340–343, March 1991.

[46] S.S. Koonar and J.S. Sohal. Realization of two-variable transfer functions. *Proceedings of the IEEE*, 72(6):743–745, June 1984.

[47] Ali Eydgahi. A counter example to "realization of two-variable transfer functions". *Proceedings of the IEEE*, 74(1):219, January 1986.

[48] Paul Lambrechts, Jan Terlouw, and Samir Bennani Maarten Steinbuch. Parametric uncertainty modeling using lfts. In *Proceedings of the American Control Conference*, pages 267–272, June 1993.

[49] Yi Cheng and Bart De Moor. A multidimensional realization algorithm for parametric uncertainty modeling problems. In *Proceedings of the 32nd Conference on Decision and Control*, pages 3022–3023, December 1993.

[50] Yi Cheng and Bart De Moor. A multidimensional realization algorithm for parametric uncertainty modelling and multiparameter margin problems. *International Journal of Control*, 60(5):789–807, 1994.

[51] M.G. Safanov and M. Athans. Gain and phase margins for multiloop lqg regulators. *IEEE Transactions on Automatic Control*, 22(4):173–179, April 1977.

[52] Christine Belcastro. *Uncertainty Modeling of Real Parameter Variations for Robust Control Applications.* PhD thesis, Drexel University, 1994.

[53] D. C. Youla and B. Gnavi. Notes on n-dimensional system theory. *IEEE Trans. Circuits and Systems*, CAS-26(2):105–111, feb 1979.

[54] Jeff S. Shamma and James R. Cloutier. A linear parameter varying approach to gain scheduled missile autopilot design. In *American Control Conference*, pages 1317–1321, 1992.

[55] Jeff S. Shamma and James R. Cloutier. Trajectory scheduled missile autopilot design. In *1st IEEE Conference on Control Applications*, pages 237–242, Dayton, OH, September 13-16 1992.

[56] J. B. Dempster and J. I. Arnold. Flight test and evaluation of an advanced stability augmentation system for b-52 aircraft. *AIAA Journal of Aircraft*, 6:343–348, 1969.

[57] J. L. Peczkowski and M. K. Sain. Synthesis of system responses: A nonlinear multivariable control design approach. In *Proceedings of the American Control Conference*, pages 1322–1329, Boston, MA, 1985.

[58] K. J. Astrom and B. Wittenmark. *Adaptive Control.* Addison-Wesley, New York, 1989.