Coordinated Control of Unmanned Aerial Vehicles

by

Peter Joseph Seiler

B.S. (University of Illinois at Urbana-Champaign) 1996
B.S. (University of Illinois at Urbana-Champaign) 1996

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering-Mechanical Engineering

in the

GRADUATE DIVISION
of the
UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor J.K. Hedrick, Chair
Professor Raja Sengupta
Professor Andrew Packard

Fall 2001

The dissertation of Peter Joseph Seiler is approved:

Chair Date

Date

Date

University of California, Berkeley

Fall 2001

Coordinated Control of Unmanned Aerial Vehicles

Peter Joseph Seiler

Abstract

Coordinated Control of Unmanned Aerial Vehicles

by

Peter Joseph Seiler

Doctor of Philosophy in Engineering-Mechanical Engineering

University of California, Berkeley

Professor J.K. Hedrick, Chair

This thesis considers the problem of coordinated control of unmanned aerial vehicles (UAVs). This problem has recently received significant attention in the controls community due to its numerous applications. To realize the benefits of advanced UAV systems, we must build hierarchically from simple to complex tasks. The focus of this thesis is one step in this process: formation flight. This problem is interesting in its own right and leads to several scientific questions.

Consider two possible distributed control architectures. First, each vehicle could use a control law that depends on measurements from all vehicles in the formation. This architecture allows us to design centralized controllers but requires the vehicles to communicate large amounts of information. Alternatively, we could design a distributed control architecture where each vehicle uses only sensor information about neighboring vehicles. This architecture does not require communication, but it may lead to disturbance propaga-

tion. Specifically, disturbances acting on one vehicle will propagate and, if amplified, may have a large effect on another vehicle. This amplification of disturbances is commonly called string instability. It is clear that formation flight requires basic research into the areas of distributed control, disturbance propagation, and control over communication channels.

This thesis makes advances in each of these areas with application to the formation flight problem. In a simple manner, we show that several strategies that do not require communication tend to be string unstable. This result motivates a control design procedure for formation flight that requires communicated leader information. We then determine how often this information must be communicated for acceptable control. We find theoretical bounds on network performance for a simple vehicle following problem. We also develop tools to determine how often information must be communicated in more general networked control systems. The main tool of interest is a computationally tractable method to find the closed loop performance as a function of packet loss rate. These tools require supporting results in the area of jump linear systems. Finally, we apply the control design procedure and the network analysis tool to the formation flight problem.

Professor J.K. Hedrick
Dissertation Committee Chair

Contents

List of Figures

Chapter 1

Introduction

In this thesis, we consider the problem of coordinated control of unmanned aerial vehicles (UAVs). This problem has recently received significant attention in the controls community due to its numerous applications. Some examples include firespotting [31], space science missions [77, 91], surveillance [6, 98], terrain mapping [98], and formation flight [65, 76]. In these applications, unmanned vehicles are used because they can outperform human pilots, they remove humans from dangerous situations, or because they perform repetitive tasks that can be automated.

Many of these tasks are complex and a systematic procedure is needed to realize the ultimate benefits of advanced UAV systems. A logical roadmap is to build hierarchically from simple to complex tasks. First, one should control a single UAV and be able to command the UAV to perform various useful tasks, e.g. travel to a given wavepoint. Next, one should coordinate several UAVs in a simple way. One simple coordinated control problem is formation flight. As we discuss below, this problem is interesting in its own

right and leads to several scientific questions. Finally, one should build up from simple coordinated tasks to the final complex task.

At Berkeley, the BEAR project (see [87] and references therein) has taken the first step of controlling a UAV. This thesis focuses on the second step of performing a simple coordinated vehicle maneuver. In particular, we concentrate on the problem of formation flight. While still at the beginning of the roadmap, it is an important step toward the ultimate goal of advanced UAV systems. Moreover, formation flight itself has many applications. For example, flying in formation can reduce fuel consumption by 30% [65]. However, this requires tight tracking to realize these fuel savings. Formation flying can also be used for airborne refueling and quick deployment of troops and vehicles [76]. Cooperating vehicles may also perform tasks typically done by large, independent platforms [92, 98]. Gains in flexibility and reliability are envisioned by replacing large platforms with smaller vehicles operating in a formation.

Even a simple coordinated task such as formation flight leads to several interesting scientific problems. A meeting sponsored by the Air Force Office of Scientific Research outlined several research needs in the area of advanced UAV systems [6]. Consider two possible distributed control architectures. First, each vehicle could use a control law that depends on measurements from all vehicles in the formation. This architecture allows us to design centralized controllers but requires the vehicles to communicate large amounts of information. Alternatively, we could design a distributed control architecture where each vehicle uses only sensor information about neighboring vehicles. This architecture does not require communication, but it may lead to disturbance propagation. Specifically,

disturbances acting on one vehicle will propagate and, if amplified, may have a large effect on another vehicle in the formation. This amplification of disturbances is commonly called string instability. It is clear that formation flight requires basic research into the areas of distributed control architectures, disturbance propagation and string stability, and control over noisy communication channels.

1.1 Thesis Overview

The problem of formation flight provides a backdrop for the basic investigation pursued in this thesis: What are the communication requirements for acceptable distributed control? This investigation leads to advances in the areas of distributed control architectures, disturbance propagation and string stability, and control over noisy communication channels. In this section we give an outline of this thesis and the path leading to these advances. All chapters give a review of previous work that is pertinent.

Chapter 2 gives an introduction to the coordinated vehicle control problem. Specifically, we introduce the problem of controlling a string of vehicles in the context of Automated Highway Systems (AHS). The possibility of disturbance propagation in vehicle strings has been known for some time. This chapter serves to review past work by offering new interpretations of this phenomenon. Specifically, we analyze several distributed control architectures that do not require communication. While the architectures under consideration only require sensors capable measuring distances to neighboring vehicles, we show that a large class are string unstable. In other words, errors amplify as they propagate through the vehicle string and hence these architectures are sensitive to disturbances. A

simple solution is to communicate lead vehicle information to all following vehicles. This solution has been previously used to design string stable control laws for AHS.

Chapter 3 generalizes the notion of string stability to multiple dimensions. In mathematical terms, a string of vehicles is a collection of single-input single-output systems with errors propagating along one direction in the string. For formation flight, we must consider multiple-input multiple-output systems as well as formations where errors propagate in multiple dimensions. We use the generalizations in this chapter to develop a simple formation control design procedure.

The formation controller proposed in Chapter 3 requires the lead vehicle in the formation to communicate its measurements to all followers in the mesh. The motivates an investigation into the effect of communication delays on closed loop systems with networks in the feedback loop. Such systems are termed Networked Control Systems (NCS). The study of NCS is relatively new and there is a need for both theoretical results and practical tools for such systems. The next two chapters of the thesis provide both theoretical results and practical tools for NCS.

Chapter 4 focuses on theoretical bounds for a simple networked control system. We pose a simple vehicle following problem involving a network. This system falls into the general class of Markov Jump Linear Systems (MJLS). The existence of a stabilizing controller for this problem is reduced to the existence of a stable estimator receiving measurements over a network. We then investigate this estimation problem and find theoretical bounds on the network performance (in terms of packet loss) for the existence of a stable estimator. These results are one step on the path to general controllability / observability

conditions for NCS.

In **Chapter 5** we develop tools for analysis and design of more general networked control systems. Again we note that the networked control systems under consideration fall into the class of Markov Jump Linear Systems (MJLS). First we present useful results on stability and H_∞ performance for MJLS. We derive a necessary and sufficient matrix inequality for a MJLS to achieve a given level of H_∞ performance. This yields a computationally tractable method to determine the performance of a networked control system. Then we show how this matrix inequality can be used to synthesize optimal controllers for networked systems. Finally we present several examples demonstrating the usefulness of the tools.

In **Chapter 6**, the tools developed in the previous chapters are applied to the problem of formation flight. We describe a linear model of a small-scale helicopter obtained by D. Shim [87] using time-domain system identification methods. This model is valid for hovering and low-velocity maneuvers. We also present a controller, designed by D. Shim [87], that is used for wavepoint navigation of a single UAV. This controller is used in to design a formation flight controller using the procedure detailed in Chapter 3 and thus results in a mesh stable design. This formation controller requires the UAVs to communicate information across a wireless network. Finally, we analyze the effect of communication delays using the results obtained in Chapter 5.

Chapter 7 presents conclusions and gives recommendations for future work. Several appendices follow the thesis and give supporting results.

1.2 Thesis Contributions

This thesis makes several contributions in the areas of distributed control and control of networked systems. Supporting results in this thesis also provide contributions in the area of Jump Linear Systems. These contributions are discussed below.

1. **Distributed Control:** The state of the art in distributed control is advanced by analyzing disturbance propagation in vehicle strings and formations.

   - *String Stability:* A simple vehicle following strategy is for each vehicle to use a radar keep a fixed distance behind the preceding vehicle. Many researchers have shown that this strategy is string unstable for specific control laws. In other words, errors may be amplified as they propagate through a vehicle string if this strategy is employed. In Chapter 2, we show in a simple manner that this strategy is string unstable for *any* linear controller. We also show that this strategy is sensitive to disturbances: a small disturbance acting on one vehicle can propagate through the string and have a large effect on another vehicle. One solution to this problem is to communicate lead vehicle information to all followers.

   - *Mesh Stability:* In Chapter 3, we extend the string stability results to formations of vehicles. This leads to a design procedure for mesh stable formation controllers. This is mostly a notational problem, but it does lead to some interesting ties with graph theory. These ties remain open for future work.

2. **Control of Networked Systems:** Theoretical results and practical tools for net-

worked control system are provided.

- *Theoretical bounds for Networked Control Systems:* In Chapter 4, we pose a simple vehicle following problem involving a network. For this problem, we characterize the effect of the network by a packet loss process. A necessary and sufficient condition on the packet loss rate is derived for the existence of a stabilizing controller for this vehicle following problem. We show that this condition provides a hard constraint on the network performance as measured in terms of bandwidth and packet loss rate.

- *Tools for Design and Analysis of Networked Control Systems:* In Chapter 5, we develop computationally efficient methods to solve problems related to control over wireless networks. A supporting result related to jump linear systems is a necessary and sufficient matrix inequality for a system to achieve a given level of H_∞ performance. We show how this matrix inequality can be used to synthesize optimal controllers for NCS. These tools are then applied to examples in vehicle control. In particular, we analyze the effect of packet losses on a formation flight controller.

3. **Jump Linear Systems:** In Appendix C, we derive a necessary and sufficient matrix inequality for a jump linear system to achieve a given level of H_∞ performance. This leads to two computationally tractable methods to determine the performance of a networked control system. The preferable algorithm for a particular problem depends on the state dimension of the plant.

Chapter 2

Error Propagation in Vehicle Strings

2.1 Introduction

The problem, in its most basic form, is to move a collection of vehicles from one point to another point. There are two simple solutions to this problem. First, we could design one centralized controller to move all vehicles to the desired destination. Second, we could decouple the problem by treating each vehicle independently. For this second approach, the control design for an individual vehicle can be posed in a classical form with x_i denoting the position of the i^{th} vehicle (Figure 2.1). It is sufficient to design a controller, $K(s)$, for the i^{th} vehicle, $H_i(s)$, and specify a reference trajectory, r_i, to move this vehicle to the destination. Little more than standard techniques are required for either of these two approaches.

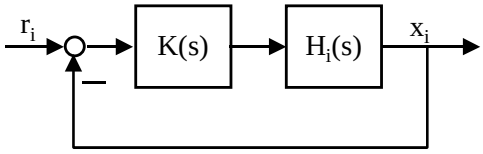

Figure 2.1: Standard Feedback Loop

The problem becomes interesting in applications where the vehicles must interact using incomplete information. One example is an Automated Highway System (AHS) [39]. In this application, the goal is to reduce traffic congestion by using closed loop control. To maximize the traffic throughput, the vehicles travel in closely spaced platoons (Figure 2.2). Centralized control is impractical for medium to large sized platoons. Thus a decentralized controller should be used. Furthermore, treating the vehicles independently is an unsafe approach because the inter-vehicle spacings are required to be small. A reasonable decentralized control strategy is for each vehicle to use a radar to keep a fixed distance behind the preceding vehicle. Figure 2.3 displays this control strategy. The reference trajectory for the $(i+1)^{th}$ vehicle is a fixed distance, δ_i, behind the preceding vehicle: $r_{i+1} = x_i - \delta_i$. In contrast to the decoupling strategy in Figure 2.1, the vehicles interact and their feedback loops are coupled. It is now possible for disturbances acting on one vehicle to propagate and affect other vehicles in the string. In fact, we show that for any control law, $K(s)$, it is possible for a small disturbance acting on one vehicle to have an arbitrarily large effect on another vehicle (Sections 2.4.1 and 2.5.2).

The possibility of disturbance propagation in vehicle strings has been known for some time. This chapter serves to review past work by offering new interpretations of

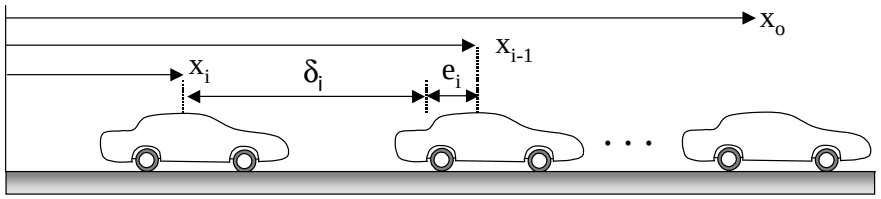

Figure 2.2: An AHS Platoon

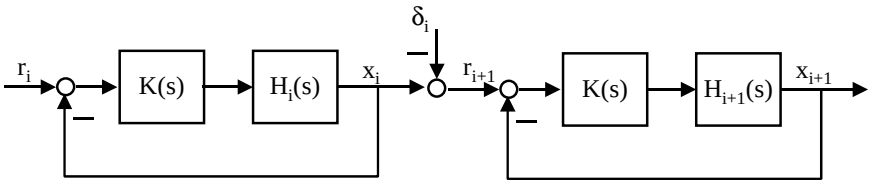

Figure 2.3: Coupled Feedback Loops

this phenomenon. Specifically, the remainder of this chapter is devoted to finding general properties of several decentralized control structures. This will also function as a bridge to the next chapter: Error Propagation in Vehicle Formations. In the next section, we survey the past work in this area. Then we formulate the problem and give a simple analysis of several control structures. We give a more detailed analysis of the control structures in Section 2.5. Finally we discuss the implications of these results on decentralized control design.

2.2 Related Work

The control of a string of vehicles has a long history of research. Early research [49, 57] was motivated by a high-speed ground transport for the Northeast United States. The research initially focused on centralized controllers for the string of vehicles. As stated

above, centralized designs for large platoons of vehicles are difficult to implement. Alternatively, each car could be treated individually using a moving cell reference [102]. This scheme has two drawbacks. It requires a method to communicate the reference trajectory to each vehicle. Furthermore, this scheme ignores the relative spacing between vehicles. It is important for vehicles to maintain a specified separation even if disturbances cause the vehicles to deviate from their moving cell reference. Thus if vehicles are closely spaced, a moving cell reference scheme may be unsafe.

Subsequent research focused on decentralized control designs [12, 16, 19, 68]. Chu's research [19] on the control of an infinite string of vehicles displays several key ideas. First, the vehicles in an actual platoon tend to have similar dynamics. A useful abstraction of this property is a string of identical vehicle models which are indexed by a discrete variable. Note that the dynamics have a spatial dependence (the vehicle index) and a time dependence. The string has the property of spatial invariance with respect to this discrete index, i.e. the dynamics of each vehicle are the same. Chu removed the spatial dependence using the bilateral Z-transform and then investigated the stability of the string for various decentralized control laws. He defined the closed loop string to be stable if the state of the infinite dimensional string remained bounded and converged asymptotically to zero for bounded initial conditions. An interesting result occurs when vehicles try to maintain a fixed distance behind their predecessor. If each vehicle uses only relative spacing error and a proportional control law, string stability cannot be achieved. A similar result was also shown via a transfer function analysis [68]. Specifically, Peppard used a PID controller and found the transfer function from one spacing error in the string to the next. If the controller

uses only relative spacing information, this transfer function has a gain greater than 1 at some frequency. Errors at this frequency will be amplified as they propagate through the string.

In the early 90s, renewed interest in AHS spurred further research on the control of vehicle strings [29, 38, 85, 88, 94, 95, 96]. Rigorous definitions of string stability and relations to error propagation transfer functions were obtained by Swaroop [94]. While Chu used the Z-transform to remove the spatial variable, Swaroop used the $\mathcal{L}_\infty$ norm to remove the temporal variable. He then used sliding control and investigated the propagation of errors in the spatial domain. He showed that string stability could not be achieved using only relative spacing information, but it could be achieved if the lead vehicle communicated its state information to all other cars in the string. Eyre, et.al. [29] gave interpretations of these results in terms of blocks connected with springs and dampers.

The research on vehicle strings can be generalized. Kamen [45] studied a class of systems whose inputs and outputs are functions of a time variable and a discrete spatial variable. This research on spatio-temporal systems was extended on several fronts by Bamieh, Paganini and Dahleh [5]. Their work further generalizes the problem to the case where the spatial variables form a Group and the dynamics are spatially invariant with respect to translations in the Group. For a class of optimal control problems, they show that the optimal controller has two nice properties. First, the optimal controller is spatially invariant. In terms of a vehicle string, this means that each vehicle uses the same control law. Second, the control gains have exponential rates of decay in space. For a vehicle string, this means the optimal control law depends heavily on the neighboring vehicles. Based on

these properties, spatial truncation of the optimal controller is suggested. The truncated controller is a decentralized approximation of the optimal centralized controller.

To summarize, we note that many definitions of "string stability" have occurred in previous literature. These definitions basically correspond to notions of scalability and disturbance damping. In particular, stability of an infinite string is related to the idea that we can add as many cars to a string without fear of instability. This scalability is reassuring if we do not have a priori bounds on the number of vehicles. Alternatively, the attenuation of propagating errors is related to the idea that disturbances acting on a vehicle should have small effect on other vehicles, i.e. they should be damped. We also note that many researchers have shown that "string stability" cannot be obtained when vehicles only use relative spacing information to maintain a constant distance behind their predecessor. All of these results have been for specific control laws such as proportional, PID, and sliding controllers.

In this chapter we derive disturbance attenuation properties for several decentralized control structures. In particular, we show in a simple manner that if vehicles only use relative spacing information, then we have "string instability" for *any* linear controller.

2.3 Problem Formulation

The problem is motivated by the control of an AHS platoon (Figure 2.2). The platoon is a string of $N+1$ vehicles. Let $x_0(t)$ denote the position of the lead car and $x_i(t)$ $(i = 1, \ldots, N)$ denote the position of the i^{th} follower in the string. Define the vehicle spacing errors as: $e_i(t) = x_{i-1}(t) - x_i(t) - \delta_i(t)$ $(i = 1, \ldots, N)$ where $\delta_i(t)$ is the desired

vehicle spacing. The goal is to force these spacing errors to zero and ensure that small disturbances acting on one vehicle cannot have a large effect on another vehicle. Before proceeding, we call attention to some of our assumptions:

Assumption 1: All the vehicles have the same model.
Assumption 2: The vehicle model is linear and SISO.
Assumption 3: All vehicles use the same control law.
Assumption 4: The desired spacing is a constant: $\delta_i(t) \equiv \delta$

We are more interested in performance at the platoon level rather than individual vehicle control. Thus Assumptions 1 and 2 are reasonable abstractions of the problem at this scale. Assumption 3 is a simplification for ease of implementation. However there is some theoretical justification for this decision. The work by Bamieh, et.al. [5] shows that the optimal controller for an infinite string of identical vehicles is spatially invariant, i.e. each vehicle uses the same controller. The use of a different controller for each vehicle will be discussed briefly in Section 2.4.1. Finally, there are a variety of other spacing laws and the constant spacing policy is chosen for this analysis. In particular, a weak version of string stability can be achieved if the the desired spacing is changed based on vehicle velocity. The reader is referred to [94] for a complete treatment.

Given any time-domain signal, $x(t)$, we denote its Laplace Transform, $\mathcal{L}\{x(t)\}$, by $X(s)$. Applying the assumptions, we can model each vehicle in the Laplace domain as (assuming the vehicles start from rest):

$$X_i(s) = H(s)U_i(s) + \frac{x_i(0)}{s} \qquad \text{for } i = 1, \ldots, N \tag{2.1}$$

where $H(s)$ has two poles at the origin and $x_i(0)$ is the initial position of the i^{th} vehicle. A simple point mass model for a car is $H(s) = \frac{1}{s^2}$ with the vehicle acceleration as the control

input. In general, $H(s)$ can include actuator dynamics. We assume that these actuator dynamics can be represented by a proper transfer function so that $H(s)$ has relative degree ≥ 2. The spacing error is given by $e_i(t) = x_{i-1}(t) - x_i(t) - \delta$. We assume the platoon starts with zero spacing errors and the leader starts at $x_0(0) = 0$. Hence, $x_i(0) = -i\delta$ for $i = 0, \ldots, N$.

2.4 Error Propagation

In this section we give a simple analysis of three decentralized control laws. We show that if vehicles use only relative spacing information, then some frequency content of the errors will be amplified as it propagates. In other words, this strategy is string unstable. This problem can be solved if each vehicle also uses leader information. We will make use of the following norm: $\|X(s)\|_\infty := \sup_{\omega \in \mathbb{R}} \bar{\sigma}\left(X(j\omega)\right)$. For a SISO system, this is just the peak magnitude on a Bode plot.

2.4.1 Predecessor Following

In this section we investigate the decentralized control law suggested at the beginning of this chapter. A linear control law based only on relative spacing error is given by:

$$U_i(s) = K(s)E_i(s) \tag{2.2}$$

Simple algebra gives the following relations:

$$\begin{aligned} E_1(s) =& X_0(s) - X_1(s) - \frac{\delta}{s} \overset{(a)}{=} X_0(s) - H(s)K(s)E_1(s) \\ E_i(s) =& X_{i-1}(s) - X_i(s) - \frac{\delta}{s} \overset{(a)}{=} H(s)K(s)(E_{i-1}(s) - E_i(s)) \qquad \text{for } i = 2, \ldots, N \end{aligned}$$

where equalities labeled (a) follow from Equations 2.1, 2.2 and the platoon initial conditions. We can obtain the spacing error dynamics from these relations:

$$E_1(s) = \frac{1}{1 + H(s)K(s)} X_0(s) := S(s)X_0(s) \tag{2.3}$$

$$E_i(s) = \frac{H(s)K(s)}{1 + H(s)K(s)} E_{i-1}(s) := T(s)E_{i-1}(s) \qquad \text{for } i = 2, \ldots, N \tag{2.4}$$

These equations show that the transfer function from $X_0(s)$ to $E_1(s)$ is the sensitivity function, $S(s)$. In a standard feedback loop, the sensitivity function is the transfer function from reference to error. Since $x_0(t)$ and $e_1(t)$ are the reference and error for the first follower, Equation 2.3 is not surprising. The transfer function from $E_{i-1}(s)$ to $E_i(s)$ is the complementary sensitivity function, $T(s)$.

There is a classical trade-off between making $|S(j\omega)|$ and $|T(j\omega)|$ small. Making $|S(j\omega)|$ small corresponds to disturbance rejection, reference tracking, and sensitivity to model variations. Making $|T(j\omega)|$ small corresponds to noise rejection and robustness to high frequency unmodeled dynamics. Since $S(s) + T(s) \equiv 1$, we cannot make $|S(j\omega)|$ and $|T(j\omega)|$ simultaneously small. Fortunately the competing objectives occur in different frequency regions. It is typically sufficient for $|S(j\omega)|$ to be small at low frequencies and $|T(j\omega)|$ to be small at high frequencies.

In the context of Equations 2.3 and 2.4, the $S(s)$ vs. $T(s)$ trade-off has the interpretation of limiting initial spacing error (making $|S(j\omega)|$ small) and limiting the propagation of errors (making $|T(j\omega)|$ small). We would like $|T(j\omega)| < 1$ at all frequencies so that propagating errors are attenuated. In this case, we cannot spread these competing objectives into different frequency bands.

In fact, it is not possible to attenuate propagating errors at all frequencies. Note

that if $K(s)$ stabilizes the closed loop, then $H(s)K(s)$ has two poles at $s = 0$. This follows since no unstable pole-zero cancellations between $H(s)$ and $K(s)$ is a requirement for closed loop stability. Thus $T(0) = 1$ and hence $\|T(s)\|_\infty \geq 1$. The next theorem implies that the inequality is strict: $\|T(s)\|_\infty > 1$. This is a simplified version of a theorem by Middleton and Goodwin [61, 53].

Theorem 2.1 *Suppose that $H(s)$ is a rational transfer function with at least two poles at the origin (i.e. two integrators). If the associated feedback system is stable, then the complementary sensitivity function must satisfy:*

$$\int_0^\infty \log|T(j\omega)|\frac{d\omega}{\omega^2} \geq 0 \tag{2.5}$$

where log *is the natural log.*

This theorem follows from Theorem 3.2 which is proved in Chapter 3. This integral relation is similar to the more common Bode Sensitivity integral. We note that $\log|T(j\omega)| > 0$ if $|T(j\omega)| > 1$ and $\log|T(j\omega)| < 0$ if $|T(j\omega)| < 1$. Therefore the integral implies that the area of error amplification is greater than or equal to the area of error attenuation. Since $H(s)$ is strictly proper, $|T(j\omega)| \to 0$ as $\omega \to \infty$. There is some frequency band of error attenuation (high frequencies) and hence there must be a frequency band of error amplification. Thus a simple consequence of this theorem is that for *any* stabilizing controller, there exists a frequency, ω, such that $|T(j\omega)| > 1$. We also note that time delays and right half plane zeros make the integral strictly positive. In this case, the area of error amplification exceeds the area of error attenuation. The interested reader is referred to [61, 53] for these generalizations.

Figure 2.4 shows an example of this result. The vehicle model is a double integrator with first order actuator dynamics and a lead controller is used to follow the preceding vehicle:

$$H(s) = \frac{1}{s^2(0.1s+1)} \tag{2.6}$$

$$K(s) = \frac{2s+1}{0.05s+1} \tag{2.7}$$

Figure 2.4 is a plot of $|T(j\omega)|$ and $|S(j\omega)|$. Note that the y-axis is a linear scale, not the log scale commonly used in Bode plots. As predicted by Theorem 2.1, there is a frequency such that $|T(j\omega)| > 1$. Specifically, $\|T(s)\|_\infty = 1.21$ and is achieved at $\omega_0 = 0.93$ $rads/sec$. Errors acting at this frequency will be amplified as they propagate.

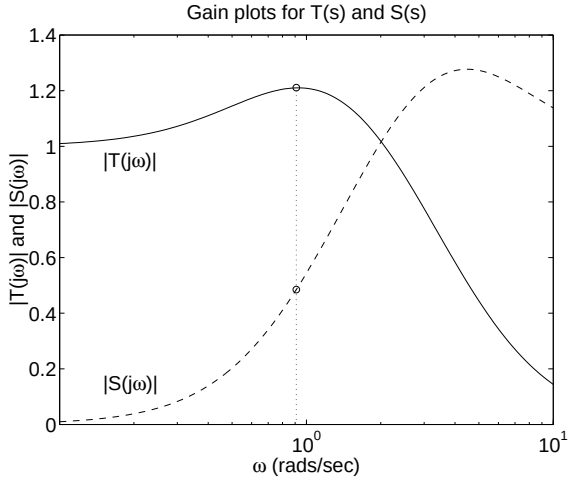

Figure 2.4: Plots of $|T(j\omega)|$ and $|S(j\omega)|$ for $H(s) = \frac{1}{s^2(0.1s+1)}$ and $K(s) = \frac{2s+1}{0.05s+1}$. $\|T(s)\|_\infty = 1.21$ and is achieved at $\omega_0 = 0.93 rads/sec$. At this frequency, $|S(j\omega_0)| = 0.50$.

We elaborate on this last statement. Consider a 6 car platoon ($N = 5$) starting from rest with initial conditions $x_i(0) = -i\delta$ for $i = 0, \ldots 5$. The desired spacing is $\delta = 5m$. The lead vehicle accelerates from rest to 20 m/s over 12 seconds using the following input:

$$U_0(s) = \frac{1}{s^2}\left[e^{-s} - e^{-3s} - e^{-11s} + e^{-13s}\right] \tag{2.8}$$

In the time domain, this corresponds to a trapezoidal input with peak acceleration of $2m/s^2$ (Left subplot of Figure 2.5). The lead vehicle motion, $X_0(s) = H(s)U_0(s)$, causes an initial spacing error, $E_1(s) = S(s)X_0(s)$. The right subplot of Figure 2.5 shows that $|E_1(j\omega)|$ has substantial low-frequency content. Figure 2.4 shows that $|T(j\omega)| > 1$ at low frequencies, so we expect low-frequency content to be amplified. The right subplot of Figure 2.5 confirms that low frequency content is amplified as it propagates from $E_1(s)$ to $E_5(s)$.

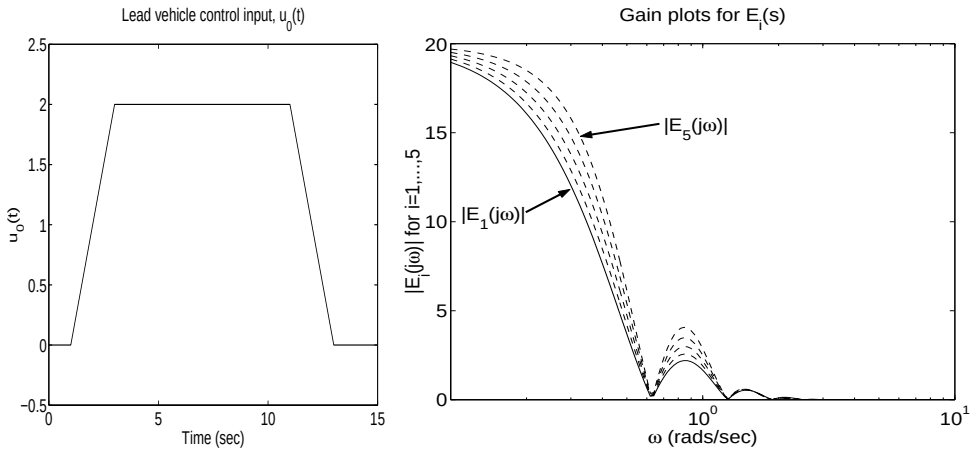

Figure 2.5: *Left:* Lead vehicle input, $u_0(t)$, in the time domain. *Right:* $|E_i(j\omega)|$ for all vehicles in the platoon. $|E_1(j\omega)|$ has substantial low-frequency content and this content is amplified as it propagates from $E_1(s)$ to $E_5(s)$.

This error amplification can also be interpreted in the time domain (Right subplot of Figure 2.6). In this example, the vehicles farthest from the leader experience the largest peak spacing error. We comment briefly on peak error amplification. Define the ∞-norm for a scalar time-domain signal as $\|y\|_\infty := \sup_\tau |y(\tau)|$. If $\|e_{i-1}\|_\infty$ is finite, then we can bound the i^{th} peak error by [89]:

$$\|e_i\|_\infty \leq \|t\|_1 \cdot \|e_{i-1}\|_\infty \tag{2.9}$$

where $t(\tau)$ is the impulse response of $T(s)$ and $\|t\|_1 = \int_0^\infty |t(\tau)|d\tau$. Peak errors can never

amplify by more than this bound, but it is possible for this bound to be achieved. For this example, the peak error is amplified by 1.1 from $i = 4$ to $i = 5$ (Figure 2.6). This amplification is less than $\|t\|_1 = 1.37$. If amplification of peak errors is a concern, then $\|t\|_1 < 1$ ensures that peak errors will be attenuated. Finally, we note that $\|t\|_1 \geq \|T(s)\|_\infty$ and hence $\|T(s)\|_\infty > 1$ implies that peak errors may amplify. Further results on peak error amplification can be found in [94].

The left subplot of Figure 2.6 shows that the vehicles farthest from the leader also use more control effort. This occurs because the control effort also propagates via $T(s)$: $U_i(s) = T(s)U_{i-1}(s)$. The same statements regarding amplification of peak control apply here. If more cars are added to the platoon, then either the actuators on the trailing cars will saturate or a collision will occur. Relaxing Assumption 3 in Section 2.3 does not improve the situation. Let each vehicle use a different controller, $K_i(s)$. The error propagation relation (Equation 2.4) becomes:

$$E_i(s) = \frac{H(s)K_{i-1}(s)}{1 + H(s)K_i(s)} E_{i-1}(s) \qquad \text{for } i = 2, \ldots, N$$

We can make errors attenuate by choosing the controllers to be progressively stronger. The control effort now propagates via the following relation:

$$U_i(s) = \frac{H(s)K_i(s)}{1 + H(s)K_i(s)} U_{i-1}(s)$$

Applying Theorem 2.1, the control effort will be amplified at some frequency for any stabilizing $K_i(s)$. The extra freedom in relaxing Assumption 3 allows us to attenuate errors, but does not solve the problem of actuator saturation on the trailing cars.

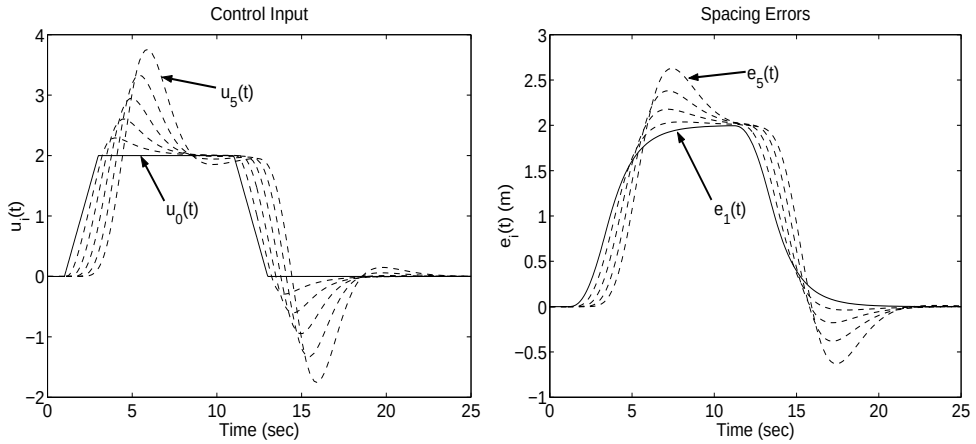

Figure 2.6: Time domain plots of predecessor following strategy. *Left:* Control input for all cars in the platoon. *Right:* Spacing error for all cars in the platoon.

2.4.2 Leader Following

In this section we analyze the decoupling strategy initially proposed in the chapter. In this case, each vehicle computes their control action based on the error with respect to the lead vehicle: $u_i(t) = k(t) * (x_0(t) - x_i(t) - i \cdot \delta)$ where $*$ denotes convolution. The control law in the Laplace domain is given by:

$$U_i(s) = K(s) \left(X_0(s) - X_i(s) - \frac{i\delta}{s} \right) \tag{2.10}$$

Intuitively this strategy is unsafe if the vehicles are closely spaced since it ignores the relative spacing between vehicles. This point will be discussed in Section 2.5. For now we ignore the issue of safety and show that this approach decouples the problem into N simple vehicle following problems.

As in Section 2.4.1, we can substitute the plant model (Equation 2.1) and the control law (Equation 2.10) into the error relation, $E_i(s) = X_{i-1}(s) - X_i(s) - \frac{\delta}{s}$, to obtain

the error dynamics:

$$E_1(s) = \frac{1}{1 + H(s)K(s)} X_0(s) = S(s)X_0(s) \tag{2.11}$$

$$(1 + H(s)K(s))E_i(s) = 0 \qquad \text{for } i = 2, \ldots, N \tag{2.12}$$

These equations again show that the transfer function from $X_0(s)$ to $E_1(s)$ is the sensitivity function, $S(s)$. They also show that all relative spacing errors, $E_i(s)$ $(i = 2, \ldots, N)$, will be zero. Each vehicle tries to maintain a fixed distance behind the leader and by assumption, each vehicle has the same model and controller. In theory, the vehicles perform identical maneuvers when tracking $x_0(t)$ (modulo the constant offset) and hence the relative spacings between the followers are always zero. In a certain sense, we can view this control structure as perfectly attenuating the propagating errors. In the next section, we combine the 'Predecessor Following' law with the 'Leader Following' law to obtain a decentralized control law which is both safe and also attenuates propagating errors.

2.4.3 Predecessor and Leader Following

In this section we combine the control laws used in the previous two sections. Each vehicle computes their control action as follows:

$$U_i(s) = K_p(s)E_i(s) + K_l(s)\left(X_0(s) - X_i(s) - \frac{i\delta}{s}\right) \tag{2.13}$$

This controller tries to keep the errors with respect to the preceding vehicle and with respect to the lead vehicle small. The leader motion is essentially the reference for the string. Intuitively, this control law gives each vehicle some preview information of this reference. In the predecessor following strategy, the i^{th} vehicle is not aware of the desired

platoon motion until the $(i-1)^{th}$ vehicle moves. The dynamics of all preceding vehicles contribute to this information lag. However, if each vehicle uses leader information, then the desired platoon motion is known immediately. Similar to the previous sections, we can obtain the error dynamics:

$$E_1(s) = \frac{1}{1 + H(s)\left(K_p(s) + K_l(s)\right)} X_0(s) := S_{lp}(s) X_0(s) \tag{2.14}$$

$$E_i(s) = \frac{H(s)K_p(s)}{1 + H(s)\left(K_p(s) + K_l(s)\right)} E_{i-1}(s) := T_{lp}(s) E_{i-1} \qquad \text{for } i = 2, \ldots, N \tag{2.15}$$

If $K_p(s) \equiv 0$ or $K_l(s) \equiv 0$ then these equations reduce to the corresponding equations in the previous sections. Note that $T_{lp}(0) = \frac{K_p(0)}{K_p(0)+K_l(0)}$ so we are free from the constraint $T(0) = 1$. For example, if we have access to the leader position, we can choose $K_l(s) = K_p(s)$ so that $T_{lp}(0) = 0.5$. More importantly, we can easily design $K_l(s)$ and $K_p(s)$ so that $\|T_{lp}(s)\|_\infty < 1$.

We compare this strategy to the predecessor following strategy described in Section 2.4.1. We use the same vehicle model given in Equation 2.6. Figure 2.7 shows $|T(j\omega)|$ using the predecessor following strategy again with $K(s) = \frac{2s+1}{0.05s+1}$. This figure also shows $|T_{lp}(j\omega)|$ using the control law in Equation 2.13 with $K_l(s) = K_p(s) = \frac{1}{2}K(s)$. With this choice, the total control effort with Equation 2.13 should be comparable to the control effort with Equation 2.2 using $K(s)$. Moreover, $T_{lp}(s) = \frac{1}{2}T(s)$, so the peak magnitude is dropped to $\|T_{lp}(s)\|_\infty = 0.605$. Thus all frequency content of propagating errors is attenuated.

Figure 2.8 shows the time responses for comparison. Note that the leader is also the preceding vehicle for the first follower. Therefore, Equations 2.2 and 2.13 are the same control law for the first follower. Consequently, $e_1(t)$ and $u_1(t)$ are the same in Figures 2.6 and 2.8. The right subplot shows that the spacing errors are attenuated as they propagate

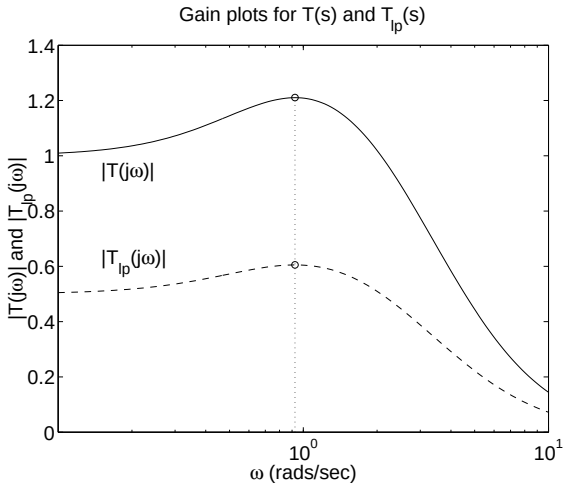

Figure 2.7: Plots of $|T(j\omega)|$ and $|T_{lp}(j\omega)|$ for $H(s) = \frac{1}{s^2(0.1s+1)}$, $K(s) = \frac{2s+1}{0.05s+1}$, and $K_l(s) = K_p(s) = \frac{s+1/2}{0.05s+1}$. $\|T(s)\|_\infty = 1.21$ and $\|T_{lp}(s)\|_\infty = 0.605$, both achieved at $\omega_0 = 0.93 rads/sec$.

down the chain. The left subplot shows that the control effort still grows, albeit slowly, as it propagates down the chain. For this control law, the i^{th} control effort is given by:

$$U_i(s) = \left[T_{lp}(s)^i + \frac{H(s)K_l(s)}{1 + H(s)(K_p(s) + K_l(s))} \sum_{k=0}^{i-1} T_{lp}(s)^k \right] U_0(s) \tag{2.16}$$

Although the control effort is increasing, it will not grow without bound since $\|T_{lp}(s)\|_\infty < 1$. This is in contrast to the predecessor following strategy where $U_i(s) = T(s)U_{i-1}(s)$ and $\|T(s)\|_\infty > 1$.

2.5 Sensitivity to Disturbances

In the previous section we examined error propagation for several decentralized control structures. In that analysis, the leader motion, $x_0(t)$, caused some initial spacing error, $e_1(t)$. We then examined the propagation of this error back through the string. In this section we want to further the analysis and determine the effect of disturbances acting

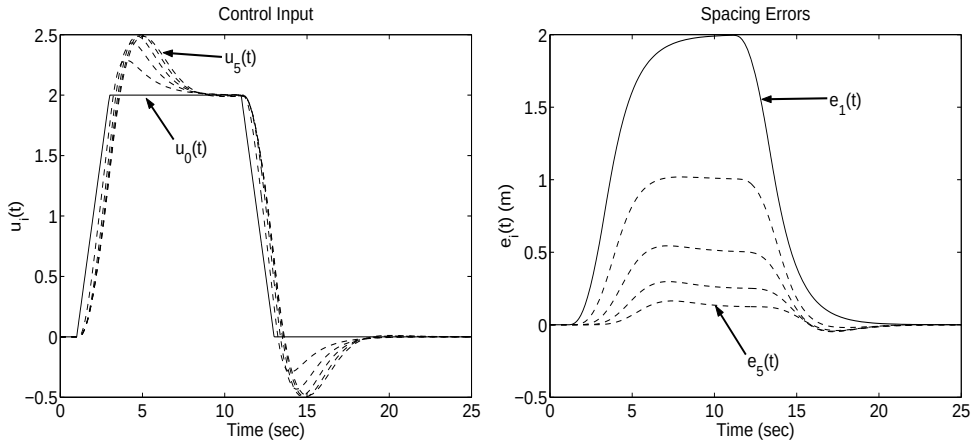

Figure 2.8: Time domain plots of predecessor and leader following strategy. *Left:* Control input for all cars in the platoon. *Right:* Spacing error for all cars in the platoon.

on each vehicle.

2.5.1 Single Vehicle

Let us first consider the control of a single vehicle with a disturbance at the input. Figure 2.9 is a standard control loop for a single vehicle, $H(s)$, with a disturbance, d_i, acting at the plant input. If $H(s)$ is a double integrator, then d_i is a force disturbance such as a wind gust. $K(s)$ is the designed controller and $r_i(t)$ is a reference trajectory from the initial point to the desired final point. The effect of the disturbance on the tracking error is shown in the following equation:

$$E_i(s) = S(s)R_i(s) - S(s)H(s)D_i(s) \tag{2.17}$$

where $S(s)$ is the complementary sensitivity function, $S(s) := \frac{1}{1+H(s)K(s)}$. Thus $-S(s)H(s)$ governs the effect of an input reflected disturbance on the tracking error.

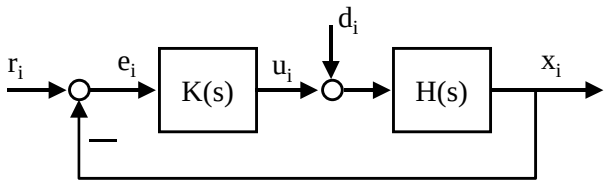

Figure 2.9: Feedback Loop with Disturbance at Plant Input

2.5.2 Look-ahead

Next consider an $N+1$ car platoon where each vehicle has an input reflected disturbance. As mentioned above, this disturbance is effectively a force disturbance acting on the vehicle. The vehicle model given in Equation 2.1 is modified to include this disturbance:

$$X_i(s) = H(s)\left(U_i(s) + D_i(s)\right) + \frac{-i\delta}{s} \qquad \text{for } i = 1, \ldots, N \tag{2.18}$$

In this relation, we have assumed the platoon initial conditions are given by $x_i(0) = -i\delta$. We will now derive the closed loop transfer function matrix from disturbances to errors when each vehicle uses preceding and lead vehicle information.

The i^{th} spacing error is given by $E_i(s) = X_{i-1}(s) - X_i(s) - \frac{\delta}{s}$. Using the vehicle model (Equation 2.18), we can write the spacing error dynamics for the platoon as:

$$\overline{E}(s) = P_{11}(s)\begin{bmatrix} X_0(s) \\ \overline{D}(s) \end{bmatrix} + P_{12}(s)\overline{U}(s) \tag{2.19}$$

where we have stacked the platoon errors, disturbances, and control inputs into N dimensional vectors and defined the necessary transfer function matrices:

$$\overline{E}(s) := [E_1(s) \ldots E_N(s)]^T, \quad \overline{D}(s) := [D_1(s) \ldots D_N(s)]^T, \quad \overline{U}(s) := [U_1(s) \ldots U_N(s)]^T$$

$$P_{11}(s) := \left[\begin{array}{c|cccc} 1 & -H(s) & & & \\ 0 & H(s) & -H(s) & & \\ \vdots & & \ddots & \ddots & \\ 0 & & & H(s) & -H(s) \end{array}\right] \quad P_{12}(s) := \begin{bmatrix} -H(s) & & & \\ H(s) & -H(s) & & \\ & \ddots & \ddots & \\ & & H(s) & -H(s) \end{bmatrix}$$

We assume each vehicle uses the control law given in Equation 2.13. This control law can be written in terms of the platoon spacing errors:

$$U_i(s) = K_p(s)E_i(s) + K_l(s)\left(\sum_{k=1}^{i} E_i(s)\right) \tag{2.20}$$

This form of the control law is strictly for convenience in the derivation that follows. In other words, the i^{th} control law, when implemented, would not require $E_1(s), \ldots, E_i(s)$. The vector of platoon inputs is given by:

$$\overline{U}(s) = \overline{K}(s)\overline{E}(s) \tag{2.21}$$

where:

$$\overline{K}(s) = \begin{bmatrix} K_l(s) + K_p(s) & & & \\ K_l(s) & K_l(s) + K_p(s) & & \\ \vdots & \ddots & \ddots & \\ K_l(s) & \cdots & K_l(s) & K_l(s) + K_p(s) \end{bmatrix}$$

We can eliminate $\overline{U}(s)$ from Equations 2.19 and 2.21 to obtain the closed loop equation from disturbances to errors:

$$\overline{E}(s) = \left[\left(I - P_{12}(s)\overline{K}(s)\right)^{-1} P_{11}(s)\right] \begin{bmatrix} X_0(s) \\ \overline{D}(s) \end{bmatrix} \tag{2.22}$$

Substitute for the matrices ($P_{11}(s)$, $P_{12}(s)$, $\overline{K}(s)$) and after some algebra:

$$\begin{bmatrix} E_1(s) \\ E_2(s) \\ \vdots \\ E_N(s) \end{bmatrix} = -S_{lp}(s)H(s) \times \begin{bmatrix} 1 & & & \\ (T_{lp}(s)-1) & 1 & & \\ \vdots & \ddots & \ddots & \\ (T_{lp}(s)-1)T_{lp}(s)^{N-2} & \cdots & (T_{lp}(s)-1) & 1 \end{bmatrix} \begin{bmatrix} D_1(s) \\ D_2(s) \\ \vdots \\ D_N(s) \end{bmatrix}$$
$$+ \begin{bmatrix} 1 \\ T_{lp}(s) \\ \vdots \\ T_{lp}(s)^{N-1} \end{bmatrix} \cdot S_{lp}(s)X_0(s) \tag{2.23}$$

$T_{lp}(s)$ and $S_{lp}(s)$ are as defined in Equations 2.14 and 2.15. There are several things to note in this relation. The transfer function vector from $X_0(s)$ to $\overline{E}(s)$ agrees with the analysis in the previous section (Equations 2.14 and 2.15). We can also find the transfer function from $U_0(s)$ to $U_i(s)$ by $\overline{U}(s) = \overline{K}(s)\overline{E}(s)$. This result agrees with Equation 2.16. Recall that the transfer function from disturbance to error for a single vehicle is $-S(s)H(s)$. Thus the first matrix on the right side would be an identity matrix if the vehicle dynamics were decoupled. The matrix shows how the coupling causes disturbances to propagates to other vehicles.

Two cases show the trade-off between disturbance propagation and safety. If we use only leader information then $T_{lp}(s) = 0$. There is no disturbance propagation in this case, but $D_i(s)$ affects $E_{i+1}(s)$ through $S_{lp}(s)H(s)$. On the other hand, if we use only preceding vehicle information, then the effect of $D_i(s)$ on $E_{i+1}(s)$ is through $-(T_{lp}(s)-1)S_{lp}(s)H(s)$. Typically, $(T_{lp}(s)-1) << 1$ at low frequencies, so the use of preceding vehicle information

reduces the effect of a disturbance on the spacing error. The price for this safety is that there always exists a frequency such that $|T_{lp}(j\omega)| > 1$. Hence disturbances may amplify as they propagate through the chain. For example, the effect of $D_1(s)$ on $E_k(s)$ is given by $-(T_{lp}(s)-1)T_{lp}^k(s)S_{lp}(s)H(s)$. If $|T_{lp}(\omega)| > 1$, then this effect is amplified geometrically for increasing k. The control law proposed in Section 2.4.3 provides a compromise to this trade-off.

This discussion of disturbance propagation is made rigorous in the ensuing theorem. In words, the theorem says that following the preceding vehicle is not a scalable algorithm. With this algorithm, as the platoon length tends to infinity, we can get an arbitrarily large gain from disturbances to errors. The theorem also states that the algorithm can be made scalable by including leader information. Then the gain from disturbances to errors will be uniformly bounded as the platoon length tends to infinity. The proof makes use of the following lemma:

Lemma 2.1 *Given any complex numbers, $a, b \in \mathbb{C}$, define the following sequence of matrices:*

$$X_N := \begin{bmatrix} 1 & & & & \\ b & 1 & & & \\ ab & b & 1 & & \\ \vdots & \ddots & \ddots & \ddots & \\ a^{N-2}b & \cdots & ab & b & 1 \end{bmatrix} \in \mathbb{C}^{N\times N} \tag{2.24}$$

If $|a| < 1$ then $\bar{\sigma}(X_N) \leq 1 + \frac{|b|}{1-|a|}$ for all N.

Lemma 2.1 is proved in Appendix A. We are now ready to state the disturbance propagation

theorem.

Theorem 2.2 *Assume $H(s)$ has 2 poles at the origin and the closed loop is stable. Let $\overline{T}_{de}(s) \in C^{N\times N}$ be the transfer function matrix from $\overline{D}(s)$ to $\overline{E}(s)$ in Equation 2.23. If $\|T_{lp}(s)\|_\infty > 1$, then given any $M > 0$ $\exists N$ such that $\|\overline{T}_{de}(s)\|_\infty \geq M$. If $\|T_{lp}(s)\|_\infty < 1$, then $\exists M > 0$ such that $\|\overline{T}_{de}(s)\|_\infty \leq M$ $\forall N$.*

Proof. For the first part of the theorem, there exists a frequency ω_0 such that $|T_{lp}(j\omega_0)| > 1$. Given any $M > 0$, choose N to satisfy the following inequality:

$$|T_{lp}(j\omega_0)|^{N-2} > \frac{M}{|S_{lp}(j\omega_0)H(j\omega_0)(T_{lp}(j\omega_0) - 1)|}$$

There is one technical subtlety in choosing N. The right side is infinite if $H(s)$ has a zero at $s = j\omega_0$ or $K(s)$ has a pole at $s = j\omega_0$. Since $H(s)$ and $K(s)$ have a finite number of poles and zeros, we can choose ω_0 such that $|T_{lp}(j\omega_0)| > 1$ and $|S_{lp}(j\omega_0)H(j\omega_0)| \neq 0$. Hence the right hand side of the inequality is finite and it is possible to choose N to satisfy the inequality. Let $e_1 \in \mathbb{R}^N$ be the first basis vector. By choice of N, $\bar{\sigma}\left(\overline{T}_{de}(j\omega_0)\right) \geq \|\overline{T}_{de}(j\omega_0)e_1\|_2 > M$. Hence $\|\overline{T}_{de}(s)\|_\infty \geq M$.

For the second part of the theorem, fix ω and define two complex numbers: $a := T_{lp}(j\omega)$ and $b := T_{lp}(j\omega) - 1$. Given these complex numbers, define the sequence of matrices, X_N, as in Equation 2.24. We can apply Lemma 2.1 to conclude that if $|a| < 1$ then $\bar{\sigma}(X_N) \leq 1 + \frac{|b|}{1-|a|}$ for all N. Therefore, if $|T(j\omega)| < 1$, then the gain from disturbance to error at the frequency ω can be upper bounded for all N:

$$\bar{\sigma}\left(\overline{T}_{de}(j\omega)\right) \leq |S_{lp}(j\omega)H(j\omega)| \cdot \left(1 + \frac{1 + |T_{lp}(j\omega)|}{1 - |T_{lp}(j\omega)|}\right) \tag{2.25}$$

By assumption, $\|T_{lp}(s)\|_\infty < 1$ and the closed loop is stable. Closed loop stability implies $\|S_{lp}(s)H(s)\|_\infty < \infty$. Equation 2.25 can now be applied to upper bound the peak gain from disturbances to errors uniformly in N:

$$\|\overline{T}_{de}(s)\|_\infty \leq \|S_{lp}(s)H(s)\|_\infty \cdot \left(1 + \frac{1 + \|T_{lp}(s)\|_\infty}{1 - \|T_{lp}(s)\|_\infty}\right) < \infty$$

■

As noted in Section 2.4.1, if we use the predecessor following strategy, then for any stabilizing, linear controller we have $\|T_{lp}(s)\|_\infty > 1$. From Theorem 2.2 we conclude that this strategy will always lack scalability because the gain from disturbances to errors grows without bound as the platoon length grows. However, if we use leader information, then it is possible to make $\|T_{lp}(s)\|_\infty < 1$. In this case, the theorem states that the algorithm is scalable because the gain from disturbances to errors is uniformly bounded as the platoon length grows.

The consequence of this theorem is displayed in Figure 2.10. The plot shows the disturbance to error gain as a function of frequency for strategies with (Right subplot) and without (Left subplot) leader information. N is the number of followers in the platoon. $H(s)$, $K(s)$, $K_l(s)$, and $K_p(s)$ are the same as those used to generate Figure 2.7. The curves labeled $N = 1$ show the disturbance to error gain for a single follower. They are the same in both subplots since the control action for the first follower is the same in both strategies. As shown in Section 2.5.1, the curve for $N = 1$ is given by $|S_{lp}(j\omega)H(j\omega)|$. The right subplot shows that the disturbance to error gain is relatively independent of vehicle size if leader information is used. The left subplot, on the other hand, shows that if the predecessor following strategy is used, then the platoon becomes sensitive to disturbances

as N grows.

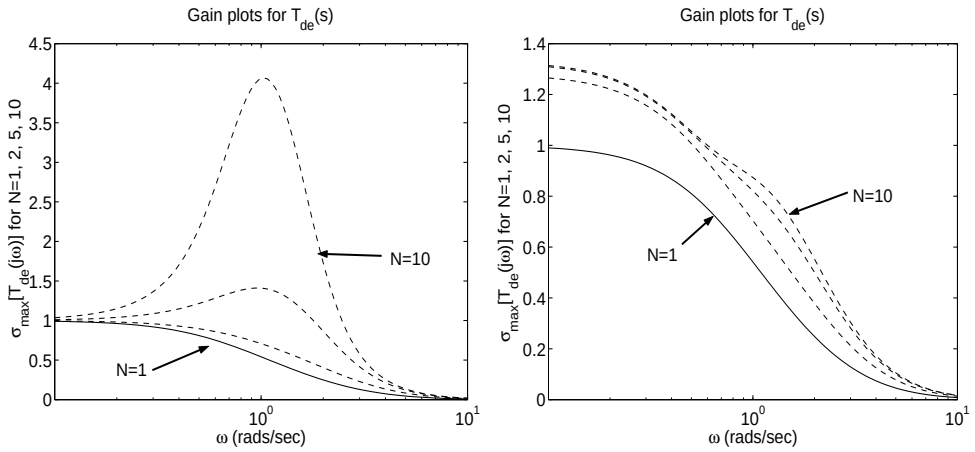

Figure 2.10: $\bar{\sigma}\left(\overline{T}_{de}(j\omega)\right)$ for $N = 1, 2, 5, 10$. *Left:* Predecessor following strategy . *Right:* Leader and predecessor following strategy.

2.5.3 Bidirectional

In the previous section, we showed that a vehicle following control law based only on relative spacing error is not scalable. The algorithm can be made scalable if all vehicles have knowledge of the lead vehicle motion. However, the latter algorithm requires a network to communicate this information to all vehicles while the former algorithm can be implemented with only on-board sensors, such as a radar. In this section, we try to construct a scalable control law that relies only on 'local' measurements, i.e. no communication between vehicles is necessary.

We consider platoon controllers which use relative spacing error with respect to adjacent vehicles. In this section, vehicles use a bidirectional controller:

$$U_i(s) = K_p(s)E_i(s) - K_f(s)E_{i+1}(s) \tag{2.26}$$

Since the last vehicle in the chain does not have a follower, it uses the control law: $U_N(s) =$

$K_p(s)E_N(s)$. P_{11}, P_{12} are as defined previously, but the controller matrix for the entire platoon, $\overline{K}(s)$, is given by:

$$\overline{K}(s) = \begin{bmatrix} K_p(s) & -K_f(s) & & \\ & \ddots & \ddots & \\ & & \ddots & -K_f(s) \\ & & & K_p(s) \end{bmatrix}$$

For this control structure, the closed loop equation from disturbances to errors is again given by Equation 2.22. We will focus on the effect of disturbances which is given by:

$$\begin{aligned} \overline{E}(s) &= \left[\left(I - P_{12}(s)\overline{K}(s) \right)^{-1} P_{12}(s) \right] \overline{D}(s) \\ &= \left(P_{12}^{-1} - \overline{K}(s) \right)^{-1} \overline{D}(s) \end{aligned} \tag{2.27}$$

where:

$$P_{12}^{-1}(s) = -\frac{1}{H(s)} \begin{bmatrix} 1 & & \\ \vdots & \ddots & \\ 1 & \cdots & 1 \end{bmatrix}$$

The next theorem shows that this strategy also fails to be scalable for a class of these bidirectional controllers.

Theorem 2.3 *Assume $H(s)$ has 2 poles at the origin and the closed loop is stable. Assume the bidirectional controller is symmetric: $K_p(s) = K_f(s)$ and $K_f(s)$ has no poles at $s = 0$. Let $\overline{T}_{de}(s) \in C^{N \times N}$ be the transfer function matrix from $\overline{D}(s)$ to $\overline{E}(s)$ in Equation 2.27. Given any $M > 0$ $\exists N$ such that $\|\overline{T}_{de}(s)\|_\infty \geq M$.*

Proof. Given the assumptions in the theorem, the disturbance to error transfer function at

$s = 0$ simplifies to:

$$\overline{E}(0) = -\frac{1}{K_f(0)} \begin{bmatrix} 1 & \cdots & 1 \\ & \ddots & \vdots \\ & & 1 \end{bmatrix} \overline{D}(0)$$

Let U_N be the $N \times N$ matrix with ones on the upper triangle and let e_N be the N^{th} basis vector. Then $\bar{\sigma}(U_N) \geq \|U_N e_N\| = \sqrt{N}$. Given any M, choose N such that $\frac{\sqrt{N}}{|K_f(0)|} > M$. For this N, $\|\overline{T}_{de}(s)\|_\infty \geq |\overline{T}_{de}(0)| > M$. ■

In the spirit of [29], we can interpret this result in terms of blocks and springs. Given the assumptions in Theorem 2.3, the platoon looks like a string of blocks connected by springs at steady state (Figure 2.11). The spring constants are given by $K_f(0)$. The problem with this strategy is that steady state disturbances are not attenuated. A simple free body diagram analysis on the second block reveals that the spring force must equal the disturbance force in steady state: $F_{s2} = d_2$. Thus the force on the first block is $d_1 + d_2$. In a general platoon containing N followers, the steady state disturbances pile up and the effect on the first vehicle is $d_1 + \cdots + d_N$. The first spacing error is then given by this force divided by the spring constant, $K_f(0)$. The problem illustrated in Theorem 2.3 is that a unit disturbance on the last vehicle, $d_N = 1$, is directly transferred to all other vehicles: $e_i = 1/K_f(0)$ for all i. Measured in the Euclidean norm, this one unit of disturbance causes $\frac{\sqrt{N}}{K_f(0)}$ units of error.

The left subplot of Figure 2.12 shows an example of the effect stated in Theorem 2.3. This plot shows the disturbance to error gain as a function of frequency when $K_f(s)$ has no poles at $s = 0$. The controller is given by $K_p(s) = K_f(s) = \frac{2s+1}{0.05s+1}$. As predicted by Theorem 2.3, the steady state gain grows as N increases. The right subplot

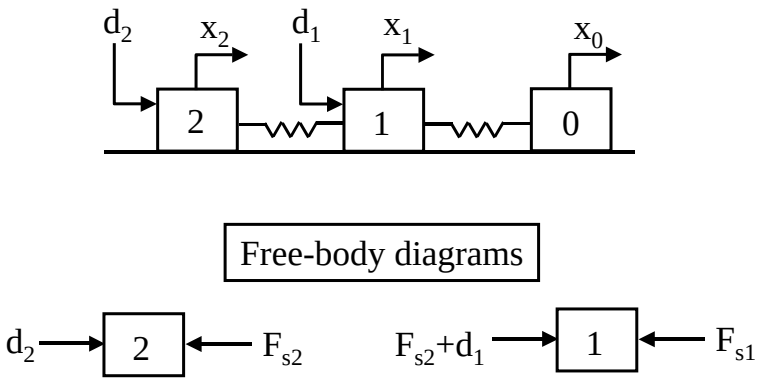

Figure 2.11: Block and spring interpretation of Theorem 2.3

shows the effect of adding an integrator to the control law: $K_p(s) = K_f(s) = \frac{2s^2+s+0.1}{s(0.05s+1)}$. The integrator causes the steady state gain to be 0 for all N. However, the peak gain from disturbances to errors changes greatly as vehicles are added to the platoon. In this example, the peak gain is actually greater for $N = 5$ than for $N = 10$. This behavior is in contrast to the predecessor/leader following strategy. Using that strategy, we can be assured that the disturbance to error gain is relatively independent of platoon size (Right subplot of Figure 2.10). Future work should further explore the properties of bidirectional controllers.

2.6 Implications for Decentralized Control Design

The key point of this chapter is that extending single vehicle designs to large platoons can lead to unintended problems. The most obvious design for vehicle following is to use a radar to follow the preceding vehicle. Section 2.4.1 shows that error amplification is always possible if this design is used for the control of a platoon. The addition of leader information can be used to solve this problem. Theorem 2.2 gives another interpretation of

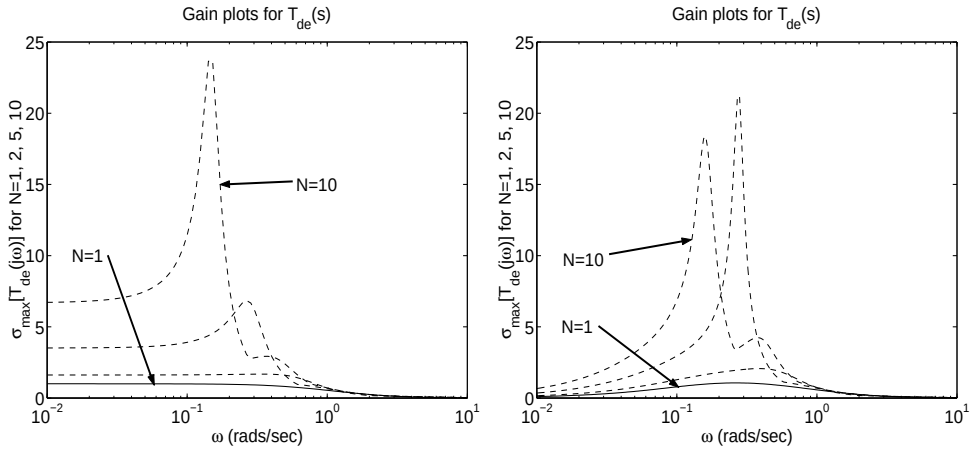

Figure 2.12: $\bar{\sigma}\left(\overline{T}_{de}(j\omega)\right)$ for $N = 1, 2, 5, 10$. *Left:* Bidirectional strategy: $K_f(s)$ with no poles at $s = 0$. *Right:* Bidirectional strategy: $K_f(s)$ with a pole at $s = 0$.

these results. The disturbance to error gain is relatively independent of platoon size when leader information is used. This algorithm is scalable in the sense that it does not become sensitive to disturbances for large platoons.

Theorem 2.3 states that a symmetric, bidirectional algorithm with no integrators fails to be scalable. While this does not imply that all bidirectional controllers lack scalability, the simple example in Section 2.5.3 demonstrates that the design of bidirectional controllers is not as straight forward as the controllers with leader information. Note that the optimal controllers designed in [5] have the property of spatial invariance and the connections between vehicles decay exponentially in space. It is suggested that the optimal controller should be truncated in space. This truncation yields a local decentralized controller that approximates the optimal controller. Unfortunately, approximating the optimal controller with a decentralized controller does not guarantee that the closed loop performance will be close to optimal. In the case of vehicle following, truncating down to a bidirectional controller may cause scalability problems.

In some cases, communication between vehicles is not possible. For example, an Adaptive Cruise Control (ACC) is a system which tracks a desired velocity similar to a standard automotive cruise control. An ACC has the additional capability to follow a slower moving vehicle using a radar. In this case, a control designer must be aware of the $S(s)$ versus $T(s)$ trade-off described in Section 2.4.1. Suppose that we want to improve tracking performance by increasing the low frequency gain. Figure 2.13 shows gain plots for $T_i(s)$ and $S_i(s)$ using two different controllers. The first controller, $K_1(s) = \frac{2s+1}{0.05s+1}$, is the same controller used to generate Figure 2.4. For the second controller, $K_1(s) = \frac{2s+2}{0.05s+1}$, the proportional gain is increased to improve tracking. As a result, $|S(j\omega)|$ is pushed down at low frequencies. However, this causes $\|T_1(s)\|_\infty = 1.21$ to pop up to $\|T_2(s)\|_\infty = 1.44$. Looze and Freudenberg give a detailed discussion of this phenomenon [53].

Figure 2.14 shows the time responses when each vehicle uses $K_2(s)$. In comparison to Figure 2.6, the right subplot of Figure 2.14 shows that $e_1(t)$ is smaller when $K_2(s)$ is used. Increasing the proportional gain improves the tracking. However, Figure 2.13 shows that this causes an increase in the error amplification (a larger peak in $|T_2(j\omega)|$). One manifestation of this error amplification is that the peak error is amplified by 1.18 from $i = 4$ to $i = 5$. Using the lower gain controller, $K_1(s)$, this peak amplification is only 1.10. For platoons longer than 6 vehicles, the loss in performance via error amplification will eventually outweigh the performance gain made in reducing e_1. Similarly, the control usage (Left subplot of Figure 2.14) is growing at a faster rate than in Figure 2.6. In fact, the fifth car uses much more control effort when $K_2(s)$ is used. The actuator on the rear vehicles will quickly become saturated if a high gain controller is used. Intuitively, using a

high gain controller makes it progressively more difficult for each following vehicle to track its predecessor. If we must use the predecessor following strategy, then a low gain controller should be used. We sacrifice some initial tracking performance (Increase $|S(j\omega)|$ at low frequencies) to improve the error amplification properties (Drop the peak of $|T(j\omega)|$).

The end result is that some communication between vehicles can vastly improve platoon performance and ease controller design. This leads to two interesting questions. What information should be communicated to obtain performance which is close to the optimal centralized performance? What is the effect of adding a network in the feedback loop of a large scale system? We have already shown that communicating leader information can lead to structural improvements in closed loop performance (i.e. no disturbance amplification). In subsequent chapters, we generalize this idea to vehicle formations and then focus on the effect of communicating information over a network.

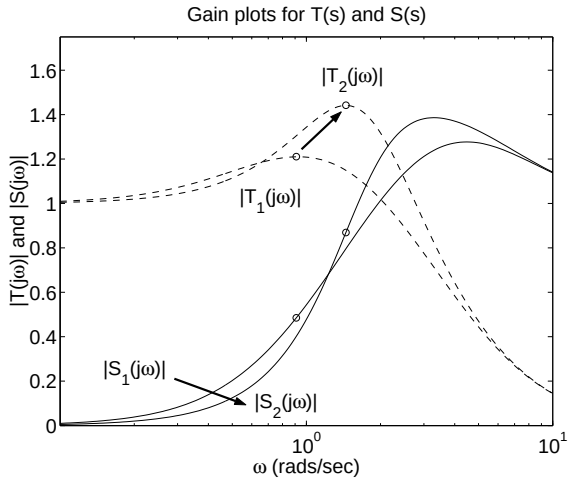

Figure 2.13: Plots of $|T_i(j\omega)|$ and $|S_i(j\omega)|$ for $K_1(s) = \frac{2s+1}{0.05s+1}$ and $K_2(s) = \frac{2s+2}{0.05s+1}$. Pushing $|S_1(j\omega)|$ down causes $\|T_1(s)\|_\infty = 1.21$ to pop up to $\|T_2(s)\|_\infty = 1.44$.

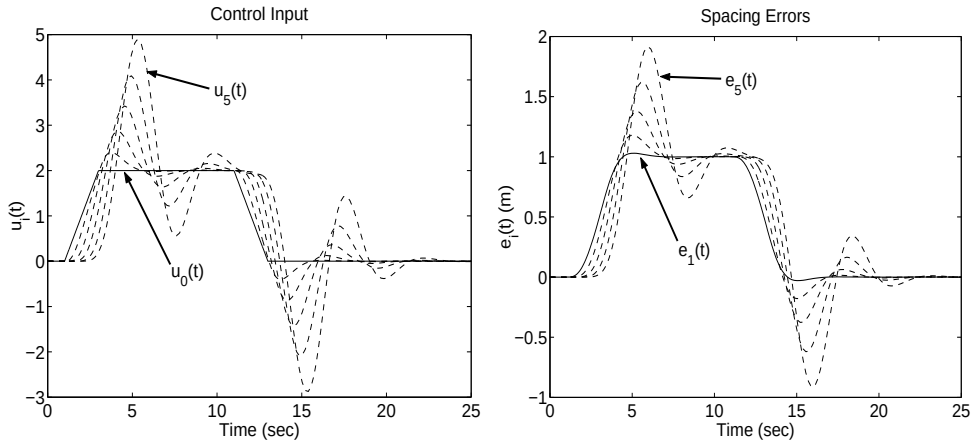

Figure 2.14: Time domain plots of predecessor following strategy with $K_2(s) = \frac{2s+2}{0.05s+1}$. *Left:* Control input for all cars in the platoon. *Right:* Spacing error for all cars in the platoon.

Chapter 3

Error Propagation in Vehicle Formations

3.1 Introduction

The control of vehicle formations has recently received significant attention in the controls community. Autonomous vehicles operate in a variety of environments such as space, air, water, and on the ground [80]. Applications for unmanned vehicles are numerous and they include firespotting [31], space science missions [77, 91], surveillance [6, 98], terrain mapping [98], and formation flight [65, 76]. In these applications, unmanned vehicles are used because they can outperform human pilots, they remove humans from dangerous situations, or because they perform repetitive tasks that can be automated.

The Air Force Scientific Advisory Board found that unmanned aerial vehicles (UAVs) are typically deployed in coordinated clusters rather than independent units [6].

Operating in formations offers benefits for a variety of tasks. For example, flying in formation can reduce fuel consumption by 30% [65]. However, this requires tight tracking to realize these benefits. Formation flying can also be used for airborne refueling and quick deployment of troops and vehicles [76]. Cooperating vehicles may also perform tasks typically done by large, independent platforms [92]. For example, high optical resolutions for astrophysical sources can be achieved with an extremely large aperture telescope or a long baseline spatial interferometer [91]. The same objective can be achieved by placing optical components aboard separate spacecraft. The use of cooperating vehicles permits flexibility in changing the baseline as well as ensuring vibrational isolation of the components. Similar gains in flexibility and reliability are envisioned by replacing large, independent satellites with clusters of cooperating microsatellites [98].

Along with the potential benefits comes the need for basic research into the control of vehicle formations. A meeting sponsored by the Air Force Office of Scientific Research outlined several research needs in this area [6]. Distributed control architectures, disturbance propagation and string stability, and control over noisy communication channels were specified as important components of the problem.

String instability refers to the amplification of disturbances as they propagate through a string of vehicles (see [94] and references therein). In this chapter, we focus on the generalization of this notion, termed mesh stability, for vehicle formations. In the next section, we survey related work on control of vehicle formations. Then we formulate the problem and in Section 3.4, we give an analysis of several control structures. We use these results to develop a simple formation control design procedure that is mesh stable.

3.2 Related Work

The numerous applications for vehicle formations, in particular formation flight, have driven research into this problem. The basic goal is to design a controller such that each vehicle maintains the formation and avoids collisions [35]. This basic goal leads to several interesting questions. How should the information topology (i.e. the sensing and communication capabilities of the formation) be chosen? More generally, how should the information topology and control law be designed for good performance? If centralized control is not possible, what is the minimum information necessary to ensure the attenuation of propagating disturbances? If we are interested in large formations, can we design a scalable control architecture? Below we outline some of the research that attempts to answer these questions.

The most natural way to represent the information topology is through directed graphs [30, 34, 58, 59, 62]. A directed graph consists of a set of vertices and a set of directed edges pointing from one vertex to another. The vertices represent the vehicles in the formation. The communication channels and sensing capabilities generate the edges of the graph. In general, these edges may be directed or bidirectional depending on the capabilities of the vehicles. Figure 3.1 shows the Blue Angels in Delta Formation (left subplot) and an information topology where each vehicle senses its predecessors (right subplot). The arrows are directed to show the flow of information through the formation. If the lead vehicle communicated to all followers, then this graph would have edges from the (1,1) vertex to all other vertices. An interesting observation is that the information topology is likely to change as vehicles move around. Mesbahi, using this observation as motivation, is working

on a dynamic extension of graph theory [58].

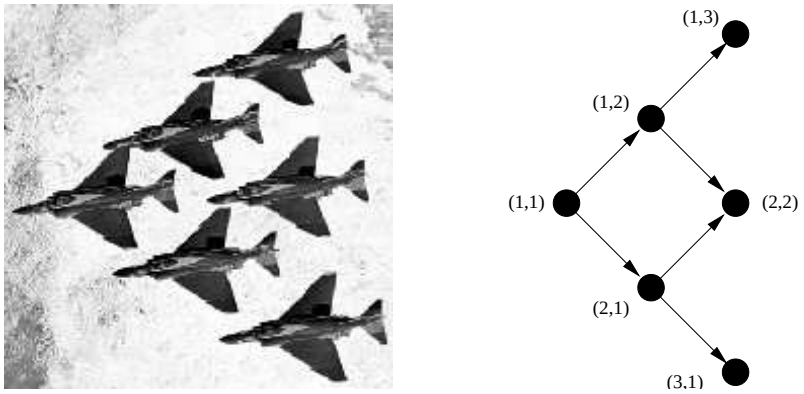

Figure 3.1: *Left:* Blue Angels in Delta formation *Right:* Graph representing a possible information topology for the Delta formation

Graphs are not only useful as a representation of the information topology, but also as a tool for control design. For example, Giulietti, et.al. give each edge of the graph a cost and they find the optimal topology with respect to these costs using the Dijkstra algorithm [34]. However, the assignment of the edge costs relative to their importance for control is a difficult and non-intuitive task. Fax and Murray use the Laplacian matrix of a graph to state a Nyquist-like stability criterion for a formation [30]. However, if the Laplacian condition shows a small stability margin, it is not clear if you need to change the information topology (keeping the controller fixed), change the controller (keeping the information topology fixed) or some combination of the two. Finally, Mesbahi has suggested the use of combinatorial optimization over valid graphs as a tool for control synthesis [59].

These are all useful tools but the effect of disturbance propagation is not directly considered. This is particularly important in tight formation flight for drag reduction. In this case, the vehicles are aerodynamically coupled due to vortices created by preceding vehicles [65]. These disturbances may propagate through the formation and, if amplified,

cause serious degradation in performance. Pachter showed that a controller that neglects aerodynamic coupling can maintain a tight formation during lead vehicle maneuvers [65]. The formation consisted of two F-16 class aircraft. One might try to extend this result to larger formations using the same control law with each vehicle treating its predecessor as the leader. In this chapter, we show that, for any stabilizing control law, disturbances will be amplified as they propagate through the formation. Several researchers have used communicated leader information to solve this problem [34, 55, 82].

Finally, we note that flocks of birds and schools of fish are able to perform formation maneuvers with apparent ease [22, 52, 69, 84, 103]. Moreover, some natural flocks appear to be scalable to any size as evidenced by 17 mile spawning runs formed by herring [84]. The amount of information used by each animal in the formation is debatable. In particular, the V formations of geese observed in nature [22] are not the optimal shape for drag reduction [52]. Furthermore, experimental observations show that the V-angle is correlated to the formation size [103] and wing-tip spacing is correlated to depth [22]. These observations support the hypothesis that the V formation is a trade-off between energy savings and visual contact of neighboring birds. In other research, Reynolds [75] developed a distributed behavior model to animate the motion of a bird flock. Each bird in the flock uses three simple rules listed in order of decreasing precedence: don't collide with neighbors, match velocity of neighbors, stay close to neighbors. A migratory urge is built into each bird so that the flock moves toward its migratory target. Natural flocks and schools point to the existence of a scalable, formation flight algorithm which may or may not require some global reference information.

3.3 Problem Formulation

The problem is motivated by advanced UAV systems. In particular, we focus on one step toward coordinating multiple UAVs: formation flight. In the previous chapter, the subsystems were connected so that information passed linearly along one direction. In this chapter, we consider information topologies where information naturally flows along two dimensions. Figure 3.2 (Left subplot) shows the follow-the-predecessor topology suggested in Figure 3.1. We also consider the information topology where the lead vehicle broadcasts his state information to all followers in addition to the predecessor sensing. The right subplot of Figure 3.2 shows this topology. The dashed edges from the (1,1) vertex to other vertices represent this broadcasted leader information.

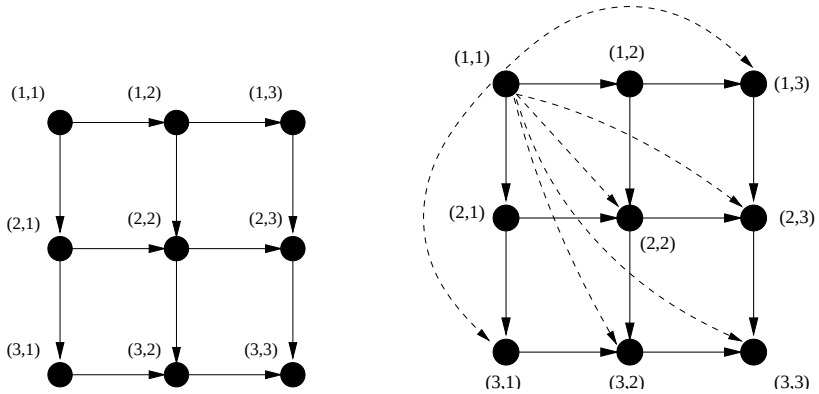

Figure 3.2: Formation Information Topologies. *Left:* Follow the predecessor *Right:* Follow the predecessor with leader information

Let $p_{i,j}(t)$ be an $n_p \times 1$ vector containing the outputs of the $(i,j)^{th}$ vehicle ($1 \leq i,j \leq N$). We assume that the outputs are translational or angular positions. For a UAV, $p_{i,j}$ could be a 4×1 vector containing the (x,y,z) positions and the heading, ψ. Define the

formation spacing errors as:

$$\epsilon^R_{i,j}(t) = p_{i,j-1}(t) - p_{i,j}(t) - \delta^R_{i,j}(t) \qquad \text{for } 1 \le i \le N,\ 1 < j \le N$$

$$\epsilon^C_{i,j}(t) = p_{i-1,j}(t) - p_{i,j}(t) - \delta^C_{i,j}(t) \qquad \text{for } 1 < i \le N,\ 1 \le j \le N \tag{3.1}$$

Figure 3.3 shows the specific error vectors for a point mass moving in a plane and $i = j = 2$. For this example, the output contains the $x - y$ position in a plane, $p_{i,j} = [x_{i,j};\ y_{i,j}]^T$. For simplicity, only one of the output vectors, $p_{2,1}$, is shown. The desired spacing vectors are given by $\delta^R_{2,2}(t) = [-L;\ 0]^T$ and $\delta^C_{2,2}(t) = [0;\ L]^T$. If all vehicles use these desired spacing vectors, the formation will have a grid-like shape similar in appearance to the information topology in Figure 3.2. Notice, however, that the formation is not required to have this rigid grid shape. By properly choosing $\delta^R_{i,j}$ and $\delta^C_{i,j}$ at each i, j we can obtain a myriad of spatial arrangements while maintaining the same subsystem communication/sensing dependencies. The $\epsilon^R_{i,j}$'s are the errors with respect to the 'left' neighbor along the row and the $\epsilon^C_{i,j}$'s are the errors with respect to the 'above' neighbor along the column. These errors are each vectors in $\mathbb{R}^{n_p}$.

The goal is to force these formation errors to zero and ensure that small disturbances acting on one vehicle cannot have a large effect on another vehicle. As in Chapter 2, we make some assumptions about the formation:

Assumption 1: All the vehicles have the same model.
Assumption 2: The vehicle model is linear.
Assumption 3: The vehicle has the same number of inputs and outputs.
Assumption 4: All vehicles use the same control law.
Assumption 5: The desired spacings are constants independent of time and formation index: $\delta^R_{i,j}(t) \equiv \delta^R$ and $\delta^C_{i,j}(t) \equiv \delta^C$.

Again, we are more interested in performance at the formation level rather than individual

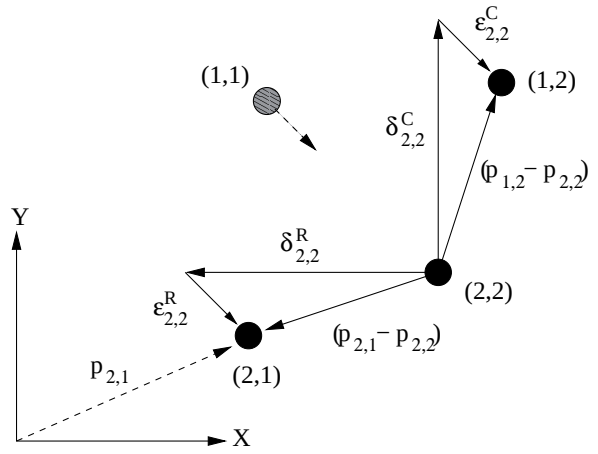

Figure 3.3: Formation Error Definitions

vehicle control. Thus Assumption 1 is a reasonable abstraction of the problem at this scale. Assumption 2 is not valid over a wide operating region because the dynamics are nonlinear. However, the results are valid for a vehicle linearized about an operating point. Alternatively, an inner loop controller could be used to regulate the UAV and make the dynamics approximately linear via feedback linearization. Assumption 3 implies that we have enough inputs to control all outputs. Thus we can force each output to track an independent reference trajectory. Assumption 4 a simplification for ease of implementation. As noted in Chapter 2, there is some theoretical justification for this decision. Finally, there are a variety of other spacing laws and the constant spacing policy is chosen for this analysis. As noted above, the formation can take on many arrangements by choosing the desired spacing vectors properly for each vehicle.

Given any time-domain signal, $x(t)$, we denote its Laplace Transform, $\mathcal{L}\{x(t)\}$, by $X(s)$. Applying the assumptions, we can model each vehicle in the Laplace domain as

(assuming the vehicles start from rest):

$$P_{i,j}(s) = H(s)U_{i,j}(s) + \frac{p_{i,j}(0)}{s} \qquad \text{for } 1 \leq i,j \leq N \tag{3.2}$$

where $H(s) = \frac{1}{s^2}\tilde{H}(s)$ and $\tilde{H}(s)$ is an $n_p \times n_p$ rational, proper transfer function matrix. $p_{i,j}(0)$ is the initial condition for the vehicle output. As mentioned above, the outputs are translational or angular positions. Thus $\tilde{H}(s)$ represents the actuator dynamics that generate forces and torques on the vehicle. These forces and torques are then integrated twice to obtain the translational or angular positions. A very simple two-dimensional point mass model for a UAV is $H(s) = \frac{1}{s^2}I_2$ with the vehicle accelerations in the x,y directions as the control inputs. In general, $\tilde{H}(s)$ includes the actuator dynamics and the outputs include the vehicle heading and altitude (z direction).

The spacing errors are given by $\epsilon^R_{i,j}(t) = p_{i,j-1}(t) - p_{i,j}(t) - \delta^R$ and $\epsilon^C_{i,j}(t) = p_{i-1,j}(t) - p_{i,j}(t) - \delta^C$. We assume $p_{1,1}(0) = 0$ and the platoon starts with zero spacing errors: $p_{i,j}(0) = -(i-1)\delta^C - (j-1)\delta^R$ for $1 \leq i,j \leq N$. We will find it useful to define an averaged error vector, $e_{i,j} = (\epsilon^R_{i,j} + \epsilon^C_{i,j})/2$, at each point in the mesh. For subsystems on the boundaries of the mesh, one of the two error terms, $\epsilon^R_{i,j}(t)$ or $\epsilon^C_{i,j}(t)$, will be undefined. In this case, the averaged error vector is defined as $e_{i,j} = \epsilon^R_{i,j}$ if $i = 1$ and $e_{i,j} = \epsilon^C_{i,j}$ if $j = 1$.

3.4 Error Propagation

In this section we give an analysis of decentralized control laws with the information topologies depicted in Figure 3.2. We show that if vehicles use a follow-the-predecessor information topology (Figure 3.2), then some frequency content of the errors will be amplified as it propagates. This problem can be solved if each vehicle also uses leader information.

We will make use of the following norm: $\|X(s)\|_\infty := \sup_{\omega\in\mathbb{R}} \bar{\sigma}\left(X(j\omega)\right)$. We also denote the spectral radius of A by $\rho[A]$.

3.4.1 Predecessor Following

A linear control law using only predecessor information (left subplot of Figure 3.2) in the form of the averaged error vector is given by:

$$U_{i,j}(s) = K(s)E_{i,j}(s) \tag{3.3}$$

Since $E_{i,j}(s) \in \mathbb{C}^{n_p\times 1}$ and $U_{i,j}(s) \in \mathbb{C}^{n_p\times 1}$, $K(s)$ is an $n_p \times n_p$ transfer function matrix. Simple algebra using the vehicle models (Equation 3.2), initial conditions, control laws (Equation 3.3), and error definitions gives the following relations:

$$E_{i,j}(s) = S_o(s)P_{1,1}(s) \qquad \text{for } (i,j) = (1,2) \text{ or } (2,1) \tag{3.4}$$

$$E_{i,1}(s) = T_o(s)E_{i-1,1}(s) \qquad \text{for } i > 2 \tag{3.5}$$

$$E_{1,j}(s) = T_o(s)E_{1,j-1}(s) \qquad \text{for } j > 2 \tag{3.6}$$

$$E_{i,j}(s) = T_o(s)\left(\frac{E_{i,j-1}(s) + E_{i-1,j}(s)}{2}\right) \qquad \text{for } 1 < i,j \leq N \tag{3.7}$$

where $S_o(s) := [I + H(s)K(s)]^{-1}$ and $T_o(s) := [I + H(s)K(s)]^{-1}H(s)K(s)$. Using terminology from [111], these are the output sensitivity and complementary sensitivity functions. Equation 3.4 shows that the transfer function matrix from $P_{1,1}(s)$ to $E_{1,2}(s)$ and $E_{2,1}$ is the output sensitivity function, $S_o(s)$. Equations 3.5 and 3.6 show that the output complementary sensitivity function governs the propagation of errors along the first column and row, respectively. Finally, Equation 3.7 shows that the internal errors of the mesh propagate, on average, via $T_o(s)$.

By Assumption 1 in Section 3.3, the formation is actually symmetrical. We can think of subsystems as lying on 'level sets' given by $L = i + j - 2$ (Figure 3.4). The $(1, 2)$ and $(2, 1)$ subsystems comprise the first level set and so on. Due to the formation symmetry, all subsystems on a given level set have the same error trajectories. For example, $E_{1,2}(s) = E_{2,1}(s)$ as verified by Equation 3.4. Induction can be used to prove the general result that $E_{i,j}(s) = E_{k,l}(s)$ if $i + j = k + l$. Define $E_L(s) := E_{i,j}$ for any (i, j) such that $L = i + j - 2$. $E_L(s)$ is well defined because all error trajectories on a given level set are the same. Using this notation, Equations 3.4-3.7 can be simplified:

$$E_1(s) = S_o(s)P_{1,1}(s) \tag{3.8}$$

$$E_L(s) = T_o(s)E_{L-1}(s) \qquad \text{for } L > 2 \tag{3.9}$$

Thus $S_o(s)$ governs the 'initial' spacing error generated by the leader motion, $P_{1,1}(s)$, and $T_o(s)$ governs the propagation of errors away from the leader along these level sets. We will use this level set notation in the remainder of the section with the understanding that all vehicles on a given level set have identical averaged spacing errors.

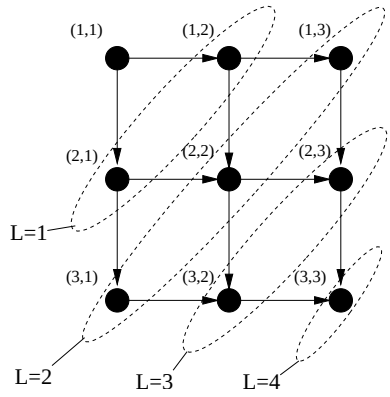

Figure 3.4: Formation level sets

We can obtain similar relations for the propagation of control effort. Below we only list the analogue for Equation 3.7, but the propagation along boundaries is similar:

$$U_{i,j}(s) = T_i(s)\left(\frac{U_{i,j-1}(s) + U_{i-1,j}(s)}{2}\right) \qquad \text{for } 1 < i, j \leq N \tag{3.10}$$

where $T_i(s) := [I + K(s)H(s)]^{-1}K(s)H(s)$ is the input complementary sensitivity function. Again, this can be simplified to $U_L(s) = T_i(s)U_{L-1}(s)$ using induction and the level set definition. Moreover, $U_1(s) = T_i(s)U_{1,1}(s)$ where $U_{1,1}(s)$ is the input to the leader.

There is a classical trade-off between making $\bar{\sigma}\left(S_o(j\omega)\right)$ and $\bar{\sigma}\left(T_o(j\omega)\right)$ small. Since $S_o(s) + T_o(s) \equiv I$, we cannot make $\bar{\sigma}\left(S_o(j\omega)\right)$ and $\bar{\sigma}\left(T_o(j\omega)\right)$ simultaneously small. Fortunately the competing objectives occur in different frequency regions. It is typically sufficient for $\bar{\sigma}\left(S_o(j\omega)\right)$ to be small at low frequencies and $\bar{\sigma}\left(T_o(j\omega)\right)$ to be small at high frequencies. In the context of Equations 3.8-3.9, the $S_o(s)$ vs. $T_o(s)$ trade-off has the interpretation of limiting initial spacing error (making $\bar{\sigma}\left(S_o(j\omega)\right)$ small) and limiting the propagation of errors (making $\bar{\sigma}\left(T_o(j\omega)\right)$ small). In this case we cannot spread these competing objectives into different frequency bands. In other words, we would like $\bar{\sigma}\left(T_o(j\omega)\right) < 1$ at all frequencies so that propagating errors are attenuated.

There are two versions of error amplification that can occur in the mesh. They are based on two conditions: $\bar{\sigma}\left(T_o(j\omega)\right) > 1$ and $\rho[T_o(j\omega)] > 1$. Consider the first condition. There exists a unit vectors, $v, w \in \mathbb{C}^{n_p}$, such that $T_o(j\omega)v = \bar{\sigma}\left(T_o(j\omega)\right)w$. If $E_{L-1}(j\omega) = v$ then $E_L(j\omega) = \bar{\sigma}\left(T_o(j\omega)\right)w$ and the error is amplified at this frequency. However, w is not necessarily aligned with v, so $E_L(j\omega)$ will not necessarily be amplified when it propagates to $E_{L+1}(j\omega)$. This leads to the second type of error amplification. Assume the second condition holds. There exists a unit vector $v \in \mathbb{C}^{n_p}$ such that $T_o(j\omega)v = \rho[T_o(j\omega)]v$. If

$E_{L-1}(j\omega) = v$, then $E_L(j\omega) = \rho[T_o(j\omega)]v$ and the error is amplified at this frequency. $E_L(j\omega)$ is now just a scaled version of v and it will again be amplified when it propagates: $E_{L+1}(j\omega) = \rho[T_o(j\omega)]^2 v$. The error is being geometrically amplified at this frequency and direction as it propagates away from the leader.

In summary, the two types of error propagation are the peak amplification between vehicles and the geometric amplification for a large mesh. The following theorem states that the spectral radius is the proper notion for error propagation in large systems. If Equation 3.9 is viewed as a dynamical system, the theorem is just a statement about its stability. This is a minor variation of Theorem 6.1 in [26] and the proof is omitted.

Theorem 3.1 *Given $A \in \mathbb{C}^{n_p \times n_p}$, $A^k v \to 0$ for all $v \in \mathbb{C}^{n_p}$ if and only if $\rho[A] < 1$. Moreover, if $\rho[A] > 1$ then there exists v such that $\|A^k v\|_2 \to \infty$.*

Thus to attenuate errors between all vehicles in the formation, we require $\bar{\sigma}\left(T_o(j\omega)\right) < 1$ at all frequencies. To ensure the eventual decay of all errors in the formation, we require $\rho[T_o(j\omega)] < 1$ at all frequencies. In fact, neither condition is possible. Note that if $K(s)$ stabilizes the closed loop, then it cannot have zeros at $s = 0$ because closed loop stability precludes unstable pole-zero cancellations. Thus $T_o(0) = I$ and hence DC errors are propagated without attenuation. Moreover, the next theorem implies that there is a frequency and a direction such that error amplification occurs. This theorem is a generalization of a SISO result by Middleton and Goodwin [61, 53]. It is similar to results obtained by Chen [14, 15].

Theorem 3.2 *Let $\tilde{L}(s)$ be a rational, proper transfer function and $L(s) = \frac{1}{s^2}\tilde{L}(s)$ be the open loop transfer function. Let $T(s) = [I + L(s)]^{-1}L(s)$ be the complementary sensitivity*

function. If the closed loop system is stable, then the complementary sensitivity function must satisfy:

$$\int_0^\infty \log \rho \left[T(j\omega)\right] \frac{d\omega}{\omega^2} \geq 0 \tag{3.11}$$

where log *is the natural* log.

Proof. This proof is similar to the proof of Theorem 4.1 in [15]. By assumption, $T(s)$ is an $n_p \times n_p$ matrix with entries that are analytic and bounded in the open right half plane. Boyd and Desoer [8] showed that $\log \rho \left[T(s)\right]$ is subharmonic and satisfies the Poisson Inequality for $x > 0$:

$$\log \rho \left[T(x)\right] \leq \frac{1}{\pi} \int_{-\infty}^{\infty} \log \rho \left[T(j\omega)\right] \frac{x d\omega}{x^2 + \omega^2} \tag{3.12}$$

Multiplying Equation 3.12 by $1/x$ and taking the limit of both sides as $x \to 0$ gives inequality (a) below:

$$\begin{aligned} \lim_{x \to 0} \frac{\log \rho \left[T(x)\right]}{x} &\overset{(a)}{\leq} \lim_{x \to 0} \frac{1}{\pi} \int_{-\infty}^{\infty} \log \rho \left[T(j\omega)\right] \frac{d\omega}{x^2 + \omega^2} \\ &\overset{(b)}{=} \frac{1}{\pi} \int_{-\infty}^{\infty} \log \rho \left[T(j\omega)\right] \frac{d\omega}{\omega^2} \\ &\overset{(c)}{=} \frac{2}{\pi} \int_{0}^{\infty} \log \rho \left[T(j\omega)\right] \frac{d\omega}{\omega^2} \end{aligned} \tag{3.13}$$

Equality (b) follows by applying the monotone convergence theorem [78] to the positive and negative parts of the integrand. Equality (c) follows from a conjugate symmetry property of $T(s)$: $\rho \left[T(-j\omega)\right] = \rho \left[T(j\omega)\right]$.

The proof is concluded by showing $\lim_{x \to 0} (1/x) \log \rho \left[T(x)\right] = 0$ and applying the end-to-end inequality in Equation 3.13. Since the open loop transfer function, $L(s)$, has two poles at $s = 0$, $T(x)$ can be expanded for sufficiently small x as:

$$T(x) = I + o(x) \text{ where } \lim_{x \to 0} \frac{o(x)}{x} = 0 \tag{3.14}$$

Thus $\rho\left[T(x)\right] = 1 + o(x)$ and $\log \rho\left[T(x)\right] = o(x)$ (where each $o(x)$ is, in general, a different function with the same limiting property as in Equation 3.14). We conclude that $\lim_{x\to 0}(1/x)\log \rho\left[T(x)\right] = 0$ as desired. ■

We note that $\log \rho[T(j\omega)] > 0$ if $\rho[T(j\omega)] > 1$ and $\log \rho[T(j\omega)] < 0$ if $\rho[T(j\omega)] < 1$. Therefore the integral implies that the area of error amplification is greater than or equal to the area of error attenuation when weighted by $\frac{1}{\omega^2}$. We can apply this theorem to $T_o(s)$ with $L(s) := \frac{1}{s^2}\tilde{H}(s)K(s)$. Since $L(s)$ is strictly proper, $\rho[T_o(j\omega)] \to 0$ as $\omega \to \infty$. There is some frequency band of error attenuation (high frequencies) and the theorem implies there must be a frequency band of error amplification. Thus for *any* stabilizing controller, there exists a frequency, ω, such that $\rho[T_o(j\omega)] > 1$. Moreover, $\bar{\sigma}\left(T_o(j\omega)\right) > 1$ because the maximum singular value upper bounds the spectral radius. Thus both types of error amplification may occur at this frequency and along particular directions. In general, there is an interval of frequencies and a set of directions (a subspace) such that amplification occurs.

Figure 3.5 shows a simple example of this result. The vehicle model is a 2×2 system with coupling between the outputs. A lead controller designed one loop at a time (ignoring the coupling) is used to follow the preceding vehicles:

$$H(s) = \frac{1}{s^2}\begin{bmatrix} \frac{1}{0.1s+1} & \frac{0.1}{s+1} \\ \frac{0.2}{2s+1} & \frac{1}{0.2s+1} \end{bmatrix} \tag{3.15}$$

$$K(s) = \begin{bmatrix} \frac{2s+1}{0.05s+1} & 0 \\ 0 & \frac{2s+1}{0.05s+1} \end{bmatrix} \tag{3.16}$$

Figure 3.5 is a plot of the spectral radius of $T_o(j\omega)$. As predicted by Theorem 3.2, there is a frequency such that $\rho[T_o(j\omega)] > 1$. Specifically, the peak is $\rho[T_o(j\omega)] = 1.30$ achieved at

$\omega = 1.29\ rad/sec$. The corresponding eigenvector is $[0.30 + 0.30i \;\; 0.90]^T$. Errors acting at this frequency and direction will be amplified as they propagate. For this particular example, $\underline{\sigma}\,(T_o(j\omega)) > 1$ for frequencies less than 2 rad/sec. Thus any frequency content below 2 rad/sec will be amplified independent of the direction: $\omega < 2\ rad/sec \Rightarrow \|E_L(j\omega)\| > \|E_{L-1}(j\omega)\|$.

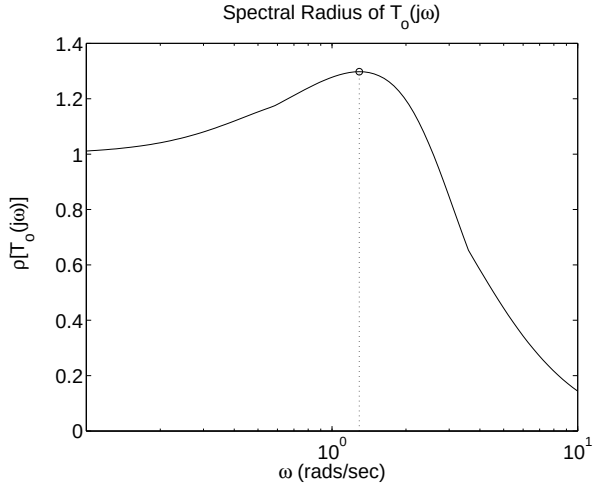

Figure 3.5: Plot of $\rho[T_o(j\omega)]$. Peak is $\rho[T_o(j\omega)] = 1.30$ achieved at $\omega = 1.29\ rad/sec$. The corresponding eigenvector is $[0.30 + 0.30i \;\; 0.90]^T$

We elaborate on this example. Consider a 9 vehicle formation starting from rest with the initial conditions $p_{i,j}(0) = -(i-1)\delta^C - (j-1)\delta^R$. The desired spacing vectors are given by $\delta^R = [-5;\ 0]^T$ and $\delta^C = [0;\ 5]^T$. The lead vehicle accelerates in the x direction from rest to 20 m/s over 12 seconds using the input $U_{1,1}(s) = [U_x(s) \;\; U_y(s)]^T$:

$$U_x(s) = \frac{1}{s^2}\left[e^{-s} - e^{-3s} - e^{-11s} + e^{-13s}\right]$$

$$U_y(s) = 0$$

In the time domain, $U_x(s)$ corresponds to a trapezoidal input with peak acceleration of $2m/s^2$ (Left subplot of Figure 3.6). The lead vehicle motion, $P_{1,1}(s) = H(s)U_{1,1}(s)$, causes

an initial spacing error, $E_1(s) = S_o(s)P_{1,1}(s)$. The right subplot of Figure 3.6 shows that $\|E_1(j\omega)\|$ has substantial low-frequency content. Figure 3.5 shows that $\rho[T_o(j\omega)] > 1$ at low frequencies, so we expect low-frequency content to be amplified along some direction. As noted above, we actually have $\underline{\sigma}\,(T_o(j\omega)) > 1$ for frequencies less than 2 rad/sec and so the error will be amplified along all directions at low frequencies. Consequently, $\|E_i(j\omega)\|$ will be amplified at low frequencies. The right subplot of Figure 3.6 confirms that low frequency content is amplified as it propagates from $E_1(s)$ to $E_4(s)$. In general, we can only conclude that the vector $E_1(j\omega)$ is amplified along a certain direction. However, the results demonstrated in this example will be typical for any loop transfer function with dominant dynamics on the diagonal and small cross-coupling terms (as with $H(s)K(s)$).

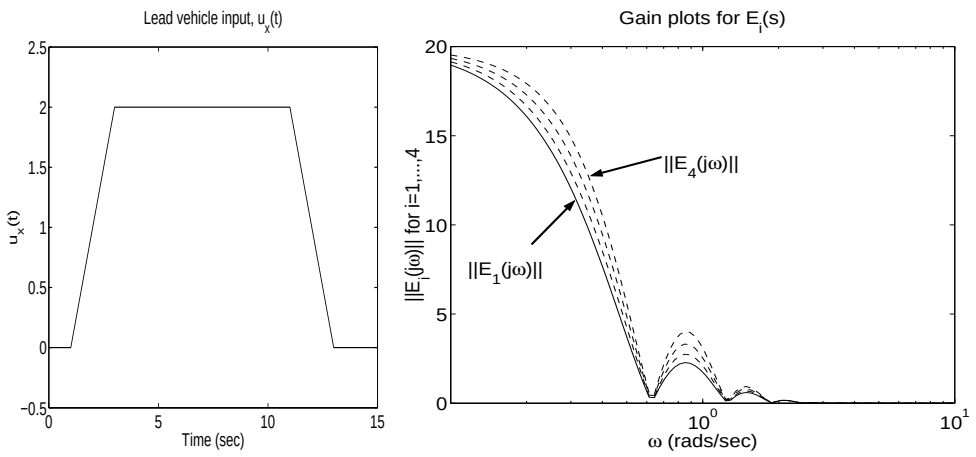

Figure 3.6: *Left:* Lead vehicle input, $u_x(t)$, in the time domain. *Right:* $\|E_i(j\omega)\|$ for all level sets. $\|E_1(j\omega)\|$ has substantial low-frequency content and this content is amplified as it propagates from $E_1(s)$ to $E_4(s)$.

This error amplification can also be viewed in the time domain (Right subplot of Figure 3.7). In this example, the vehicles farthest from the leader experience the largest spacing errors (measured in the Euclidean norm). The statements related to peak error amplification made in Section 2.4.1 can be generalized and are applicable here. The left

subplot of Figure 3.7 shows that the vehicles farthest from the leader also use more control effort. As noted above, the control effort propagates via $T_i(s)$. Theorem 3.2 can also be used to conclude the existence of a frequency, ω, such that $\rho[T_i(j\omega)] > 1$. The control effort will be amplified at this frequency and along the appropriate direction. If more vehicles are added to the formation, then either the actuators on the trailing vehicles will saturate or a collision will occur.

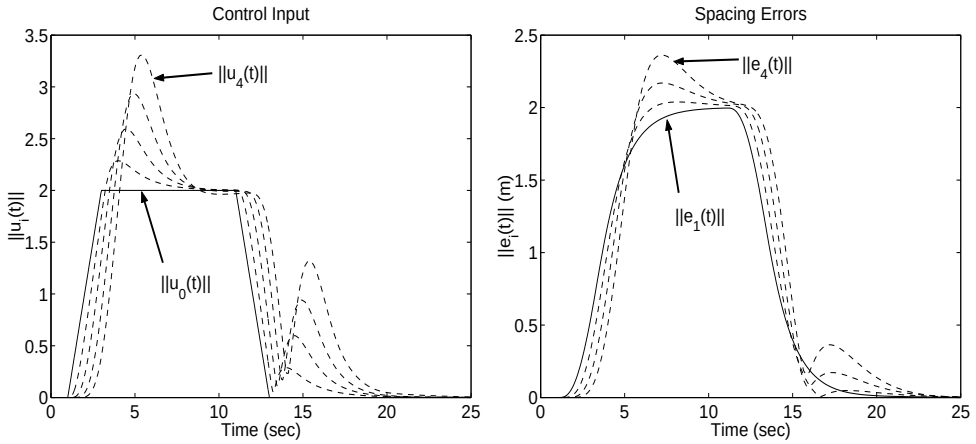

Figure 3.7: Time domain plots of predecessor following strategy. *Left:* Control input for all vehicles in the formation. *Right:* Spacing error for all vehicles in the formation.

3.4.2 Predecessor and Leader Following

A linear control law using only information from the lead and preceding vehicles (right subplot of Figure 3.2) is given by:

$$U_{i,j}(s) = K_p(s)E_{i,j}(s) + K_l(s)\left(P_{1,1}(s) - P_{i,j}(s) - \frac{(i-1)\delta^C + (j-1)\delta^R}{s}\right) \qquad (3.17)$$

This controller tries to keep the errors with respect to the preceding vehicles and with respect to the lead vehicle small. Intuitively, this control law gives each vehicle some

preview information. Similar to the previous section, we can obtain the error dynamics:

$$E_1(s) = S_{lp}(s)P_{1,1}(s) \tag{3.18}$$

$$E_L(s) = T_{lp}(s)E_{L-1}(s) \qquad \text{for } L > 2 \tag{3.19}$$

where $S_{lp}(s) := [I + H(s)(K_p(s) + K_l(s))]^{-1}$ and $T_{lp}(s) := [I + H(s)(K_p(s) + K_l(s))]^{-1} H(s)K_p(s)$. We have continued to use the level set notation introduced in the previous section. A similar induction proof can be used to show that this notation is still well defined. If $K_l(s) \equiv 0$ then these equations reduce to the corresponding equations in the previous section. Note that we are free from the constraint $T_o(0) = I$. In fact, $T_{lp}(0) = [K_p(0) + K_l(0)]^{-1}K_p(0)$. For example, if we have access to the leader position, we can choose $K_l(s) = K_p(s)$ so that $T_{lp}(0) = 0.5I$. More importantly, we can easily design $K_l(s)$ and $K_p(s)$ so that $\rho[T_{lp}(j\omega)] < 1$ for all ω or so that $\|T_{lp}(s)\|_\infty < 1$. As discussed in the previous section, the former condition ensures that errors eventually decay in the mesh while the latter condition implies that errors are attenuated between all vehicles in the formation.

We compare this strategy to the predecessor following strategy described in the previous section. We use the same vehicle model given in Equation 3.15. Figure 3.8 shows $\rho[T_o(j\omega)]$ using the predecessor following strategy again with $K(s)$ defined in Equation 3.16. This figure also shows $\rho[T_{lp}(j\omega)]$ using the control law in Equation 3.17 with $K_l(s) = K_p(s) = \frac{1}{2}K(s)$. With this choice, the total control effort with Equation 3.17 should be comparable to the control effort with Equation 3.3 using $K(s)$. Moreover, $T_{lp}(s) = \frac{1}{2}T_o(s)$, so the peak spectral radius is dropped to $\rho[T_{lp}(j\omega)] = 0.65$. Thus all frequency content of propagating errors is attenuated.

Figure 3.9 shows the time responses for comparison. Note that $\|e_1(t)\|$ and $\|u_1(t)\|$

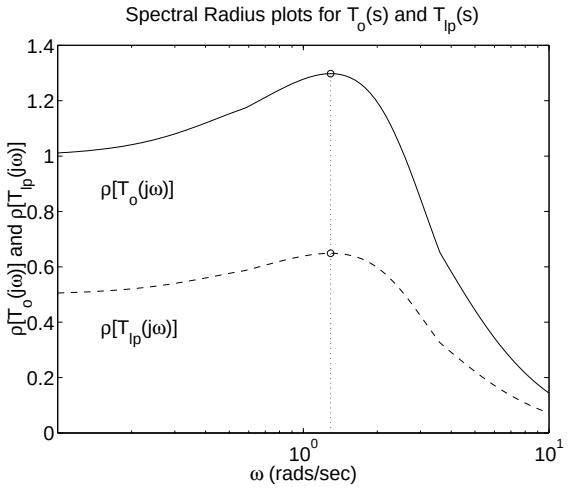

Figure 3.8: Plots of $\rho[T_o(j\omega)]$ and $\rho[T_{lp}(j\omega)]$. Peaks are $\rho[T_o(j\omega)] = 1.30$ and $\rho[T_{lp}(j\omega)] = 0.65$, both achieved at $\omega_0 = 1.29 rad/sec$.

are the same in Figures 3.7 and 3.9 because the leader is also the preceding vehicle for the first follower. The right subplot shows that the spacing errors are attenuated as they propagate down the chain. The left subplot shows that the control effort still grows, albeit slowly, as it propagates down the chain. The control effort on the L^{th} level set is given by:

$$U_L(s) = \left[T_{pi}(s)^{L-1} + T_{li} \sum_{k=0}^{L-2} T_{pi}(s)^k \right] U_{1,1}(s) \tag{3.20}$$

where we have defined $T_{li}(s) := [I + (K_p(s) + K_l(s))\, H(s)]^{-1} K_l(s) H(s)$ and $T_{pi}(s) := [I + (K_p(s) + K_l(s))\, H(s)]^{-1}\ K_p(s) H(s)$. Although the control effort is increasing, it will not grow without bound since $\rho[T_{li}(j\omega)] < 1$ for all ω (Lemma 2.1 in [26] can be applied to make this statement rigorous). This is in contrast to the predecessor following strategy where $U_L(s) = T_i(s) U_{L-1}(s)$ and $\rho[T_o(j\omega)] > 1$ for some ω.

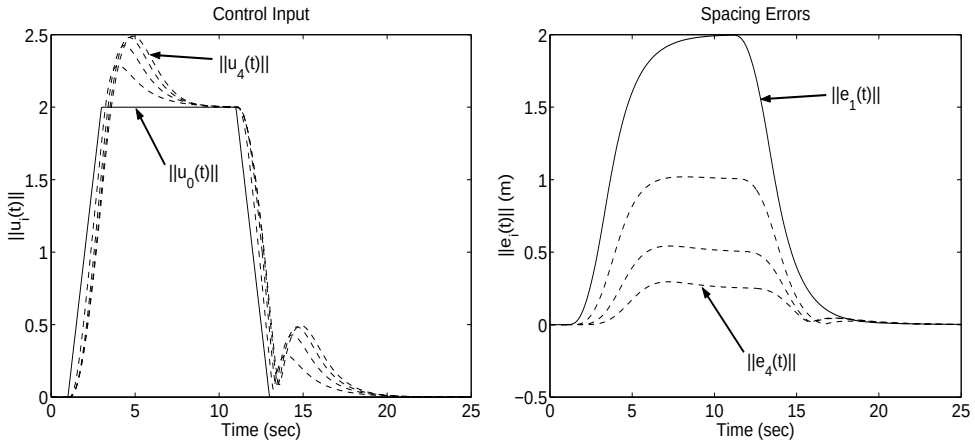

Figure 3.9: Time domain plots of predecessor and leader following strategy. *Left:* Control input for all vehicles in the formation. *Right:* Spacing error for all vehicles in the formation.

3.4.3 Disturbance Propagation

In this section we give a brief analysis of disturbance propagation in the formation. Let $D_{1,2}(s)$ be an input disturbance acting on vehicle $(1,2)$:

$$P_{1,2}(s) = H(s)[U_{1,2}(s) + D_{1,2}(s)] + \frac{p_{1,2}(0)}{s}$$

$E_{1,2}(s)$, $E_{1,3}(s)$, and $E_{2,2}(s)$ are functions of $P_{1,2}(s)$ and are affected directly by $D_{1,2}(s)$. If the control law with leader information (Equation 3.17) is used, then $E_{1,3}(s)$ can be expressed in terms of $P_{1,1}(s)$ and $D_{1,2}(s)$:

$$E_{1,3}(s) = T_{lp}(s)S_{lp}(s)P_{1,1}(s) - [T_{lp}(s) - I]S_{lp}(s)H(s)D_{1,2}(s) \tag{3.21}$$

This error propagates across the first row by $E_{1,j}(s) = T_{lp}(s)E_{1,j-1}(s)$. Using this relation and Equation 3.21, we can write $E_{1,j}(s)$ for $j \geq 3$:

$$E_{1,j}(s) = T_{lp}^{j-2}(s)S_{lp}(s)P_{1,1}(s) - T_{lp}^{j-3}(s)[T_{l,p}(s) - I]S_{lp}(s)H(s)D_{1,2}(s)$$

If leader information is used, then we can make $\|T_{l,p}(s)\|_\infty < 1$. More importantly, we can make $\rho[T_{l,p}(j\omega)] < 1$ for all ω. Repeating arguments from the previous sections (in

particular, we can apply Theorem 3.1), we conclude that the effect of $D_{1,2}(s)$ on $E_{1,j}(s)$ will tend to zero as j gets large. If leader information is not used, then $T_{lp}(s)$ reduces to $T_o(s)$. We showed in Section 3.4.1 that $\rho[T_o(j\omega)] > 1$ for some ω. Another application of Theorem 3.1 shows that the effect of the disturbance will grow geometrically if $D_{1,2}(j\omega)$ acts along a certain direction.

Disturbances propagating internally in the mesh have a similar effect in relation to $\rho[T_o(j\omega)]$. In summary, disturbances are propagated via $T_{lp}(s)$. Thus it is important for $\rho[T_{lp}(j\omega)] < 1$ at all frequencies to ensure that the effect of propagating disturbances does not grow geometrically.

3.5 Formation Control Design

We can use the results of the previous sections to motivate a simple formation controller design procedure.

1. First design a controller for a single vehicle: $P(s) = H(s)U(s)$. Let $R(s)$ denote the reference trajectory and choose $U(s) = K(s)[R(s) - P(s)]$. Design $K(s)$ so that $P(s)$ tracks the desired reference trajectories.

2. Plot $\bar{\sigma}\left(T_o(j\omega)\right)$ and find $\|T_o(s)\|_\infty$. Choose $\lambda < \frac{1}{\|T_o(s)\|_\infty}$.

3. For formation flight, use the control law given in Equation 3.17 with $K_p(s) := \lambda K(s)$ and $K_l(s) := (1 - \lambda)K(s)$. Essentially, we are using the control law designed in Step

1 to track the following reference:

$$R_{i,j}(s) = \lambda * \frac{1}{2}\left[(P_{i,j-1} - \delta^R) + (P_{i-1,j} - \delta^C)\right] + (1-\lambda) * \left[P_{1,1}(s) - \frac{(i-1)\delta^C + (j-1)\delta^R}{s}\right]$$

This is a combination of our desired position with respect to three vehicles: the leader and two predecessors. As a result of our design, $T_{lp}(s) = \lambda T_o(s)$. Hence $\|T_{lp}(s)\|_\infty < 1$ and $\rho[T_{lp}(j\omega)] < 1$ for all ω.

We can design and test the control law, $K(s)$, on an individual vehicle. Modifying this control law for formation flight as specified above ensures that we get performance that is similar to the individual vehicle case. Specifically, we have designed the controller such that propagating errors due to leader movement and disturbances will be attenuated. Thus disturbances acting on one vehicle should have a small effect on the performance of another vehicle. The cost inherent in this design is that we must broadcast leader information to all vehicles. In other words, the control law cannot be implemented with only the capability to sense neighboring vehicles. We will apply this design procedure in Chapter 6 to design a controller for a formation of UAVs. The broadcasted leader information is subject to communication network delays and in the next chapter we investigate the effect of these delays on control.

Chapter 4

Theoretical Bounds for Networked Control Systems

4.1 Introduction

A networked control system (NCS) is one in which a control loop is closed via a communication channel [101]. In the previous chapters, we showed examples in transportation systems and unmanned aerial vehicles where control is enhanced by communicated information. Other applications include flight control systems [73], manufacturing [109], automotive systems [70], and building automation [70]. In many of these applications, NCS are used not only to improve control performance, but also to increase flexibility and reduce long-term maintenance costs.

Several challenges are introduced when implementing networks in the feedback loop. In a standard sampled-data system, the sensor measurement arrives at the controller

at fixed sample instants. In theory, the controller acts on this measurement and the control action is fed back to the actuator instantaneously. In a NCS, the network introduces a time varying delay in the feedback loop due to processing (coding/decoding), waiting for access to the network, and transmission [51]. An artifact of this time-varying delay is the introduction of jitter [104]. For example, when a sensor takes a measurement, it may take some time to gain access to the network and communicate this measurement to the controller. Even if the sensor takes measurements at a fixed sample rate, the controller does not receive the measurements at fixed intervals. If this jitter is significant, then the feedback problem can no longer be modeled as synchronous. Finally, transmitted data can arrive with errors that corrupt the packet. Packets of data typically have redundant bits for error detection and correction. However, there will always be some probability of an undetected error or a detected error that cannot be corrected. These errors will either lead to longer delays or the use of incorrect information by the controller.

We have just discussed some aspects that make NCS unique from standard sampled data control. Before proceeding, we should discuss some of the aspects of control networks that differentiate them from typical data networks. A data network typically transmits large data packets at high data rates. They frequently have bursty traffic but rarely have hard real-time constraints [51, 70]. However, they usually have the constraint that the received data is an exact replication of the sent data. Retransmission of corrupted data is used to satisfy this requirement. Control networks, on the other hand, must transmit small packets frequently with real-time constraints [51, 100]. Moreover, for control it is sufficient for the received data to be a close approximation of the sent data [100]. Finally,

an interesting trade-off arises in NCS. Smaller sample times lead to better control because a discretized controller closely approximates a continuous time controller. The smaller sample time means larger network traffic as the sensor and controller signals must be transmitted more frequently. As a result, the smaller sample time may increase the communication delays [74, 110].

The study of NCS is relatively new and there is a need for both theoretical results and practical tools for such systems. With this in mind, this chapter will focus on theoretical bounds for a simple networked control system. In the following chapter, we will derive tools which have more immediate use for the design of NCS. The remainder of this chapter has the following structure: In the next section, we discuss related work on networked control systems. We then formulate a vehicle following problem involving a network. The existence of a stabilizing controller is reduced to the existence of a stabilizing estimator receiving measurements over a network. In Section 4.4 we investigate this estimation problem. We find theoretical bounds on the network performance (in terms of packet loss) for the existence of a stabilizing estimator.

4.2 Related Work

The issues raised in the introduction show that the study of NCS is interdisciplinary and includes communication, information theory, and control. One approach to NCS design is to treat the network conditions as given and synthesize a controller to account for the delays. Alternatively, the controller can be synthesized by neglecting the network and the network can be designed to minimize the probability of delays. These approaches

leave open the possibility of jointly designing the network and the controller. Some work of this nature has been undertaken by Xiao, et.al. [107]. However, 'joint design' does not have to imply that the controller and network are designed simultaneously in one step. In fact, the great success of control theory is that design does not rely on a complex and completely accurate model of the plant. Rather, 'joint design' may simply imply that the network should be designed with the intention of carrying control data and the controller should be designed to account for network induced delays. Below we discuss some work aimed at NCS design in the following order: network modeling, stability analysis, information theoretic approaches, controller design, and network design.

First, we will discuss modeling of networks for control design. Network models can vary greatly in complexity. For instance, wireless networks depend on complex radio wave propagation (e.g. multipath fading) that is not well understood [63]. These effects increase the bit error probability by several orders of magnitude over wired links. Moreover, the delay pattern for both wired and wireless networks depends greatly on the network protocol. As discussed below, Ethernet is nondeterministic so packets can have an arbitrarily long delay [51]. On the other hand, the delay can be treated as approximately constant for some protocols, e.g. token ring/bus [110]. Since the complexity of a controller depends on the complexity of a model, we would like a network model that is as simple as possible without sacrificing accuracy. For control, a model that ignores the complex issues underlying the data transmission and focuses on the delivery of the packet seems appropriate. Common techniques are to model the packet delivery characteristics with a jump system [13, 47, 81, 108, 110], a time-varying delay [28, 36, 64, 101, 99, 104], or an asynchronous dynamical

system [37, 110]. However, a scientific modeling of network characteristics for control has yet to be undertaken. Nguyen, et.al. [63] have proposed a trace-based approach to modeling wireless channels at the packet-delivery level. They obtain a simple two-state Markov model for the packet loss process and then evaluate the impact of errors on higher-layer network protocols. Stubbs and Dullerud are setting up an experimental facility with the capability to model and test both wireless and wired (Internet) networked control systems [93]. More work of this nature is required to obtain a variety of network models suitable for control.

Next we discuss stability analysis results for NCS. The basic assumption is that we have designed a controller that stabilizes the plant in the absence of the network. The goal is to determine if the closed loop will remain stable in the face of network delays. These approaches can be classified as deterministic or stochastic. Deterministic approaches rely on Lyapunov theory [37, 47, 99, 100, 101, 110]. One technique is to model the NCS as an asynchronous dynamical system with a rate constraint on packet transmission. The search for a Lyapunov function that decays on average can be posed as a convex optimization problem [37, 110]. Another approach is to assume the delays are bounded and treat them as time-varying perturbations on the nominal closed loop plant [99, 100, 101]. Using ideas from perturbation theory, bounds on the maximum delay are found to ensure stability. Since these bounds are only sufficient conditions for stability, they may restrict the acceptable network performance more than necessary. It is important to consider the time-varying nature of the delays if the jitter is significant. Suppose $\tau(t)$ is a time varying delay bounded by τ_{max}. It has been noted in the literature that an analysis based on the constant delay $\tau(t) = \tau_{max}$ can lead to optimistic results [74, 104]. Stochastic approaches try to prove a

version of stability such as mean square stability [108] or exponential mean-square stability [47]. For a class of jump systems, Ji, et.al. showed many of these notions of stochastic stability are equivalent and each implies almost sure stability [43].

Controller synthesis techniques are needed to complement the analysis tools presented above. The first step is to develop notions of stabilizability and detectability for NCS so that we can ensure that a stabilizing controller exists. Information theoretic tools have been applied to find bounds on the network rate that must be satisfied for stabilizing estimators and controllers to exist [97, 105, 106]. As an example, Tatikonda considers a discrete time estimation problem where the sensor measurement is sent across a network with a rate constraint of R bits per time step [97]. In general, the quantized measurement is coded, transmitted, decoded, and then used for estimation. For the deterministic case (no channel noise and no packet loss), Tatikonda has an elegant proof that the rate must be larger than a lower bound depending on the unstable eigenvalues of the plant [97]. Based on several assumptions, he constructs an encoder, decoder, and estimation scheme that achieves this lower bound. There are more general results on control and estimation over a network and they all give bounds on the network rate. However, it has been observed that information theoretic tools may not be useful in practice [100]. The results in [97, 105, 106], which are information theoretic in flavor, give lower bounds on how much useful data must be transmitted per time step or per second. The useful control data for an individual packet on a network is small relative to its header size. For example, Walsh et.al. note that control data is only 2% of the payload on an Ethernet network [100]. Thus decreasing the data size using information theoretic tools may not lead to noticeable results in practice.

Tools from control theory have also been applied to design controllers for networked control systems. An LQG style cost has been used to find optimal feedback gains for systems with network delays. [13]. The computation of these gains is a difficult task. If the data packets are time stamped, an observer can be used to compensate for delays in the transmission of sensor measurements [28, 64, 110]. Another approach is to model the pure time delay with a Pade approximation and use μ-synthesis to make the system robust to the delay [36, 104]. In wireless systems, two techniques have been employed by Eker, et.al. [28]. Corrupted packets can be retransmitted to guarantee the receipt of a correct packet. This increases the network delay, but as mentioned above an observer can be used to compensate for delays. Alternatively, corrupted packets can be accepted and an observer can be used to correct bit errors. This decreases network delay, but some bit errors will go uncorrected. Finally, the NCS can be modeled as a jump linear system and optimization can be used to synthesize a controller [108, 81]. This will be discussed further in the following chapter.

Finally, the requirements for acceptable feedback performance has led to numerous insights into the design of networks for control. A suitable network for real time control should be deterministic, i.e. it should transmit data within a bounded delay. It should also allow messages to be prioritized so that warnings and fault mode messages can be transmitted. Lian, et.al. compared the use of Ethernet, Token Bus, and CAN bus for control [50, 51]. The general conclusion is that Ethernet is unsuitable for control since it is not deterministic and does not support message prioritization. Walsh, et.al. have proposed a Try-Once-Discard (TOD) protocol using a Maximum-Error-First scheduler (MEF) [100, 101]. The dynamic scheduler grants the node with the largest error access to the

network. The remaining nodes discard their data and new data is used the following time. The data is discarded because newer data is generally better for control and hence queues are undesirable. Ray also noted that reducing queues improves control performance [73]. Ye, et.al. have proposed a protocol for wireless local area networks based on the definition of IEEE 802.11 [109]. Their protocol supports various priority levels and they also propose static and dynamics scheduling schemes. Finally, Lee, et.al. [48] have proposed a wireless token ring protocol for intelligent transportation systems. They develop a distributed medium access protocol that is robust against single node failures.

4.3 Vehicle Following Problem

4.3.1 Problem Formulation

In the previous chapters, we showed that communication improves control in vehicle following problems. In this section, we propose a simple vehicle following problem involving communication. Like the information theoretic approaches mentioned above, our goal is to find bounds on network performance that must be satisfied for a stabilizing controller to exist. Our approach differs in that the limiting factor of the network is the packet loss rate not the bit rate capacity.

The goal is to have one vehicle follow another vehicle using communicated information. Figure 4.1 shows a block diagram of the problem. P_1 is the model for the first vehicle. This vehicle communicates its position measurement, y_1, across a network to the second vehicle. The controller on the second vehicle has access to this communicated information, y_{1c}, as well as on-board measurements of its position, y_2. The goal is to synthesize

a controller so that y_2 tracks y_1 (modulo the desired spacings between the vehicles). In the absence of a network, Anderson and Moore term this the model following problem [2].

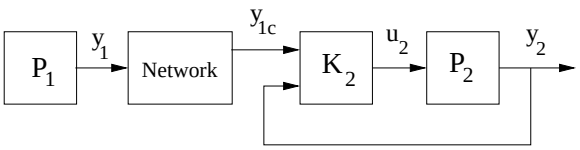

Figure 4.1: Vehicle Following Problem

We make several assumptions about the network, quantization, coding/decoding, and protocol. These assumptions are discussed below.

Assumption 1: Communication occurs over a wireless network.

Assumption 2: Quantization errors are small.

Assumption 3: Error detection is perfect.

Assumption 4: The protocol is such that jitter and transmission delays are negligible. Moreover, corrupted data is not retransmitted.

The first assumption is motivated by the use of wireless networks at California PATH (Partners for Advanced Transit and Highways) for automated highway systems [48, 54] and at the University of California-Berkeley for Unmanned Aerial Vehicle research [48, 87]. The implication is that packet errors are more likely than for wired systems. Empirical observations suggest that the packet loss process can be described by a two state Markov model [63]. We will use an even simpler two-state Markov model than developed in [63], but further work on channel modeling is needed. The second assumption implies that quantization errors can be modeled as noise [79] or as norm bounded disturbances [90]. Information theoretic tools give lower bounds on the number of bits needed to represent the sensor measurements. Instead of applying these results, we will assume that measurements are quantized to a reasonable precision at every sampling. The third assumption implies

that we can always detect packet errors but not necessarily correct them. In reality, error detection schemes are not perfect. However, observers and/or physically meaningful bounds can also be used to eliminate corrupted packets that contain unrealistic data. If a packet is corrupted, it is not used. The implication of the final assumption is that the NCS can be modeled as a synchronous process. That is, at every sample time the first vehicle sends a packet of data and the second vehicle receives this packet. The assumption that the transmission delays are negligible means that the sample time is large enough for the packet to be sent and received. If the packet is corrupted, the controller discards it and waits for the next packet. Moreover, if jitter causes the packet to be excessively delayed, the controller considers it a lost packet. The assumptions are valid for a deterministic protocol, such as token bus, that uses the Try-Once-Discard strategy. As an example, PATH uses such a protocol for the control of small platoons of vehicles [54]. This assumption can be relaxed to handle pure (i.e. constant) time delays.

If we apply the assumptions and use a simple packet-loss process for the network, then $y_1(k)$ and $y_{1c}(k)$ are related by:

$$y_{1c}(k) = \theta(k) y_1(k) \tag{4.1}$$

where $\theta(k)$ is a Bernoulli process given by $Pr[\theta(k) = 0] = p$ and $Pr[\theta(k) = 1] = 1 - p$. Thus p represents the probability that any given packet will be lost. The communication model given by Equation 4.1 simply states that a packet is either received ($\theta(k) = 1$) or lost ($\theta(k) = 0$). If a packet of data is corrupted or delayed, it is just discarded and considered to be lost. Experimental testing has shown that Equation 4.1 is a reasonable model for the packet delivery characteristics of a wireless link if $\theta(k)$ is governed by a Markov process [63].

In reality, the probability of a packet loss after receiving a correct packet is lower than after receiving a corrupt packet. In other words, the packet loss process of a wireless channel has a bursty characteristic. For our analysis, we use the simpler Bernoulli process so that the packet loss characteristics of the network can be specified by one parameter, p. The plants are modeled by the following discrete time systems:

$$\begin{aligned}
\mathbf{P_1}: \quad & x_1(k+1) = A_1 x_1(k) + w_1(k) \\
& y_1(k) = C_1 x_1(k) + v_1(k) \\
\mathbf{P_2}: \quad & x_2(k+1) = A_2 x_2(k) + B_2 u_2(k) + w_2(k) \\
& y_2(k) = C_2 x_2(k) + v_2(k)
\end{aligned}$$

where $w_i(k)$, $v_i(k)$ are zero mean, white, Gaussian noises. The initial conditions, $x_i(0)$, are also Gaussian and $(x_i(0),\ w_i(k),\ v_i(k))$ are all mutually independent. The relevant dimensions are: $y_1(k),\ y_2(k),\ u_2(k) \in \mathbb{R}^{n_p \times 1}$; $x_1(k) \in \mathbb{R}^{n_{x1} \times 1}$; $x_2(k) \in \mathbb{R}^{n_{x2} \times 1}$. As stated above, the goal is to design $\mathbf{K_2}$ so that $y_2(k)$ tracks $y_1(k)$ in a stable fashion. In the next subsection, this problem is posed in an LQG framework.

4.3.2 Optimal Control for Model Following

A reasonable performance index for the model following problem is given by:

$$J_N = E\left[\sum_{j=0}^{N-1} \left(y_2(j+1) - y_1(j+1)\right)^T Q \left(y_2(j+1) - y_1(j+1)\right) + u_2^T(j) R u_2(j)\right] \tag{4.2}$$

where the expectation is taken over all random quantities: $(x_i(0), w_i(k), v_i(k), \theta(k))$. Q is positive semi-definite (denoted $Q \geq 0$) and R is positive definite (denoted $R > 0$). As noted in [2], minimizing $y_2(j) - y_1(j)$ basically imposes n_p constraints on the state of $\mathbf{P_2}$.

We can add $n_{x2} - n_p$ further constraints on $x_2(k)$ without conflicting with the tracking objective. Anderson and Moore pose a method to add nonconflicting objectives to the cost in Equation 4.2 (see Chapter 4 of [2] for details). The end result is that the model following problem can be posed as an LQG regulator problem with the following cost:

$$J_N = E\left[\sum_{j=0}^{N-1} x^T(j+1)\hat{Q}x(j+1) + u_2^T(j)Ru_2(j)\right] \tag{4.3}$$

where R is given above and $\hat{Q} \geq 0$ if $Q \geq 0$. $x(k)$ is the state of the augmented plant which includes the packet loss model and the plant models, $\mathbf{P_1}$ and $\mathbf{P_2}$:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu_2(k) + w(k) \\ y(k) &= C_{\theta(k)}x(k) + D_{\theta(k)}v(k) \end{aligned} \tag{4.4}$$

where we have used the following notation:

$$x(k) := \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}, \quad y(k) := \begin{bmatrix} y_{1c}(k) \\ y_2(k) \end{bmatrix}, \quad w(k) := \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix}, \quad v(k) := \begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix}$$

$$A := \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}, \quad B := \begin{bmatrix} 0 \\ B_2 \end{bmatrix}, \quad C_{\theta(k)} := \begin{bmatrix} \theta(k)C_1 & 0 \\ 0 & C_2 \end{bmatrix}, \quad D_{\theta(k)} := \begin{bmatrix} \theta(k)I & 0 \\ 0 & I \end{bmatrix}$$

The mean/variance of the state initial condition and the variances of the noise processes are given by:

$$x_0 := E[x(0)] \qquad X_0 := E\left[(x(0) - x_0)(x(0) - x_0)^T\right] > 0$$

$$W := E[w(k)w(k)^T] > 0 \qquad V := E[v(k)v(k)^T] > 0$$

Note that plant in Equation 4.4 is a time varying system because the output matrices, $C_{\theta(k)}$ and $D_{\theta(k)}$, depend on the packet loss parameter, $\theta(k)$. Specifically, it is a stochastic hybrid

system because these matrices stochastically hop between two modes: $\theta(k) = 0$ when the communicated packet is dropped and $\theta(k) = 1$ when the packet is received. The probability of being in a given mode at time k depends on the underlying probability for $\theta(k)$. For example, the probability of being in mode given by $\theta(k) = 0$ is equal to p. This system falls into the class of discrete time Markovian Jump Linear Systems (MJLS) for which there is a large body of work.

Under our network assumptions, the controller has knowledge of $\theta(k)$, i.e. the controller knows when a packet has been dropped. Thus at time k, the controller has access to the following information:

$$\mathcal{I}_k := \{u_2(0), \ldots, u_2(k-1); y_{1c}(0), \ldots, y_{1c}(k); y_2(0), \ldots, y_2(k); \theta(0), \ldots, \theta(k)\} \tag{4.5}$$

The finite horizon LQG problem is to find the control input, $u_2(k)$, as a function of all the information available to the controller, $\mathcal{I}_k$, so that the quadratic cost (Equation 4.3) is minimized. As in the standard LQG problem, we impose no restriction of linearity or time-invariance on the controller. This is an instance of the Jump LQG problem solved by Chizeck and Ji and the next theorem follows from their solution [17, 42].

Theorem 4.1 *Define* $\hat{x}(k|j) := E[x(k)|\mathcal{I}_j]$. $\hat{x}(k|k)$ *is the optimal state estimate (in the*

sense of mean square error) which is given by a time varying Kalman filter:

$$\begin{aligned}
&\hat{x}(k|k) = \hat{x}(k|k-1) + F(k)[y(k) - C_{\theta(k)}\hat{x}(k|k-1)] \\
&\hat{x}(k|k-1) = A\hat{x}(k-1|k-1) + Bu_2(k-1) \\
&F(k) = M(k)C_{\theta(k)}^T \left[C_{\theta(k)}M(k)C_{\theta(k)}^T + D_{\theta(k)}VD_{\theta(k)}^T\right]^{-1} \\
&Z(k) = M(k) - M(k)C_{\theta(k)}^T \left[C_{\theta(k)}M(k)C_{\theta(k)}^T + D_{\theta(k)}VD_{\theta(k)}^T\right]^{-1} C_{\theta(k)}M(k) \\
&M(k+1) = AZ(k)A^T + W
\end{aligned} \tag{4.6}$$

The initial conditions for the filter are given by $\hat{x}(0|-1) = x_0$ *and* $M(0) = X_0$*. Given the information available to the controller,* $\mathcal{I}_k$*, the optimal control law that minimizes the cost function,* J_N*, is given by:*

$$u_2(k) = -L(k)\hat{x}(k|k) \tag{4.7}$$

for $k = 0, \ldots, N-1$ *where the feedback gains are computed recursively from a Riccati Difference Equation:*

$$\begin{aligned}
&L(k) = [R + B^T P(k+1)B]^{-1} B^T P(k+1)A \\
&P(k) = A^T P(k+1)A - A^T P(k+1)B[R + B^T P(k+1)B]^{-1} B^T P(k+1)A + \hat{Q}
\end{aligned} \tag{4.8}$$

The boundary condition for the recursion is $P(N) = \hat{Q}$*. The optimal cost is:*

$$\begin{aligned}
J_N^* =& x_0^T(P(0) - \hat{Q})x_0 + tr[(P(0) - \hat{Q})X_0] + \sum_{k=0}^{N-1} tr[P(k+1)W] \\
&+ \sum_{k=0}^{N-1} tr\left[A^T P(k+1)BL(k) \underset{\theta(0),\ldots,\theta(k)}{E}[Z(k)]\right]
\end{aligned} \tag{4.9}$$

where $tr[A]$ *is the trace of the matrix* A *and* $\underset{\theta(0),\ldots,\theta(k)}{E}[Z(k)]$ *is the expected value of* $Z(k)$ *taken over the packet loss parameters,* $\{\theta(0), \ldots, \theta(k)\}$*.*

The optimal controller, Equation 4.7, is linear but time-varying. Furthermore, it satisfies the separation property. The feedback gains are computed from a backward recursion of a Riccati Difference Equation (Equation 4.8). Note that the feedback gains are independent of the packet loss process. The state estimate is computed using a forward recursion of a Riccati Difference Equation (Equation 4.6). Since the controller has knowledge of the packet loss parameter, $\theta(k)$, the output matrices are time-varying but known. Thus the estimation problem is solved with a time-varying Kalman Filter. The filter gains and estimation error covariance, $Z(k)$, depend on the time-varying output matrices and hence depend on the packet loss process.

The optimal controller has a very simple structure which is shown in Figure 4.2. It is easy to show that the Kalman filter in Equation 4.6 decouples into two separate Kalman filters. The first filter computes an estimate of $x_1(k)$ from $y_{1c}(k)$ while the second filter estimates $x_2(k)$ from $y_2(k)$. As one would expect, the second filter is independent of the packet loss process. In other words, $Z(k) = \begin{bmatrix} Z_1(k) & 0 \\ 0 & Z_2(k) \end{bmatrix}$ where $Z_i(k)$ are the estimation error covariances for the appropriate filter. $Z_1(k)$ depends on the packet loss process, but $Z_2(k)$ does not. We can also block partition the feedback gain as $L(k) = \begin{bmatrix} L_1(k) \\ L_2(k) \end{bmatrix}$ and write the control law as $u_2(k) = -L_1(k)\hat{x}_1(k|k) - L_2(k)\hat{x}_2(k|k)$. Block partitioning the Riccati recursion (Equation 4.8) reveals that the feedback gain, $L_2(k)$, depends only on $\mathbf{P_2}$. Specifically, $L_2(k)$ is equal to the feedback gain for a regulator problem involving only $\mathbf{P_2}$. On the other hand, the feedforward gain, $L_1(k)$, depends on both $\mathbf{P_1}$ and $\mathbf{P_2}$. It is apparent from the Riccati recursion (Equation 4.8) that neither control gains depend on the packet loss process.

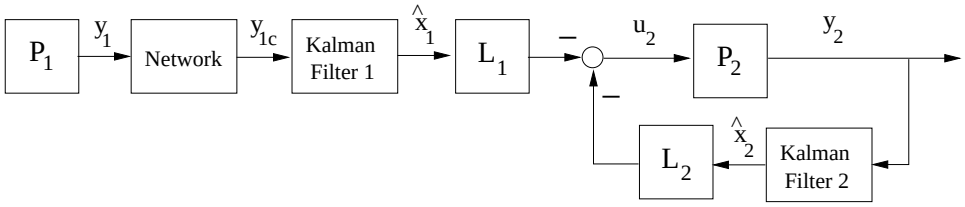

Figure 4.2: Structure of the Optimal Controller

The optimal cost is given in Equation 4.9. The terms on the first line are due to the control. In order, the three terms correspond to the cost due to the initial condition, the uncertain initial condition, and the plant noise. The term on the second line is the cost due to the estimation. It is evident from Equation 4.6 that $Z(k)$ depends on the packet loss process. As noted above, $Z(k)$ is actually block diagonal and only $Z_1(k)$ depends on the packet loss process. This is the only portion of the optimal cost that is dependent on the network performance.

It is important to emphasize that the optimal cost is a function of the packet loss rate, p. Let $J_N^*(p)$ denote the optimal cost given by Equation 4.9 with the functional dependence on the packet loss rate, p, made explicit. If we define the matrix function $f(k,p) := \underset{\theta(0),\ldots,\theta(k)}{E}[Z(k)]$, then we can write the optimal cost as:

$$J_N^*(p) = J_N^*(0) + \sum_{k=0}^{N-1} tr\left[A^T P(k+1)BL(k)\left(f(k,p) - f(k,0)\right)\right]$$

If p is larger (i.e. more packets are lost), then $\underset{\theta(0),\ldots,\theta(k)}{E}[Z_1(k)]$ will increase and add to the cost. In fact, it is straightforward to show that $f(k,p)$ is a monotonically nondecreasing function of p: $p_1 > p_2 \Rightarrow f(k,p_1) - f(k,p_2) \geq 0$, where $\geq$ denotes positive semidefiniteness. Therefore $J_N^*(p)$ is also a monotonically nondecreasing function of p. This raises the following question: For a given level of network performance, is the controller stabilizing?

Scaling the cost function does not change the optimal control law. Thus the optimal control law described in Theorem 4.1 also minimizes $\frac{J_N}{N}$ for any N. Let us rephrase the question: What conditions on the network performance ensure that the optimal cost, $\frac{J_N^*(p)}{N}$, remains finite as $N \to \infty$?

We assume that if $p = 0$, then the cost, $\frac{J_N^*(0)}{N}$, stays finite as $N \to \infty$. In other words, we can keep cost finite if there are no packet losses. As p increases from 0, the cost is increased by a term depending on $\underset{\theta(0),\ldots,\theta(k)}{E}[Z_1(k)]$. In the next section we find necessary and sufficient conditions on p to ensure that $\underset{\theta(0),\ldots,\theta(k)}{E}[Z_1(k)]$ and hence $\frac{J_N^*(p)}{N}$ stay bounded as $N \to \infty$.

4.4 Estimation over a Network

In this section we consider the problem of estimating the state of a plant using measurements obtained over a network, Figure 4.3. We switch some notation and partially review results from previous section to make the treatment of the estimation problem complete. The goal is to derive the optimal estimator given communicated measurements (Section 4.4.1) and find conditions on the network to ensure that the estimator is stable (Section 4.4.2) in the sense that the expected value of the error covariance stays bounded.

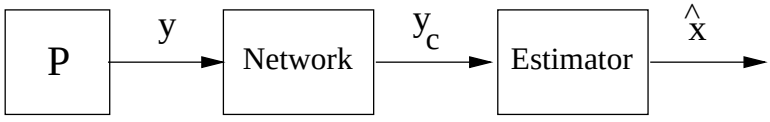

Figure 4.3: Estimation over a Network

4.4.1 Optimal Estimator

The plant, **P**, in Figure 4.3 is a discrete, linear, time-invariant system of the form:

$$\begin{aligned} x(k+1) &= Ax(k) + Bw(k) \\ y(k) &= Cx(k) + v(k) \end{aligned} \tag{4.10}$$

where $x(k) \in \mathbb{R}^{n_x}$ is the state and $y(k) \in \mathbb{R}^{n_y}$ is the measurement vector. $w(k) \in \mathbb{R}^{n_w}$ is the plant noise and $v(k) \in \mathbb{R}^{n_y}$ is the sensor noise. We assume that $v(k)$ and $w(k)$ are zero mean, white, Gaussian noises with variances $E[v(k)v(k)^T] = V > 0$ and $E[w(k)w(k)^T] = W > 0$, respectively. The state initial condition is also Gaussian with mean $E[x(0)] = x_0$ and variance $E\left[(x(0) - x_0)(x(0) - x_0)^T\right] = X_0 > 0$. Finally, $w(k)$, $v(k)$, and $x(0)$ are all mutually independent.

The measurement, $y(k)$, is obtained across a network. Thus, the estimator has access to the communicated $y(k)$, notated as $y_c(k)$. The network assumptions given in Section 4.3.1 are still in force. We again use the simple packet loss model for the network, so $y(k)$ and $y_c(k)$ are related by:

$$y_c(k) = \theta(k)y(k) \tag{4.11}$$

where $\theta(k)$ is a Bernoulli process given by $Pr[\theta(k) = 0] = p$ and $Pr[\theta(k) = 1] = 1 - p$. Again, p represents the probability that any given packet will be lost.

Given the plant, Equation 4.10, and the network model, Equation 4.11, we can form the combined plant viewed by the observer:

$$\begin{aligned} x(k+1) &= Ax(k) + Bw(k) \\ y_c(k) &= C_{\theta(k)}x(k) + D_{\theta(k)}v(k) \end{aligned} \tag{4.12}$$

where $C_{\theta(k)} = \theta(k)C$ and $D_{\theta(k)} = \theta(k)I$. We again assume that the estimator is aware of packet losses, $\theta(k)$. We now state the estimation problem.

Problem 4.1 *Given measurements* $\mathcal{Y}_k = \{y_c(0), \ldots, y_c(k)\}$ *and network observations* $\Theta_k = \{\theta(0), \ldots, \theta(k)\}$ *for the combined plant (Equation 4.12), find the state estimate,* $\hat{x}(k)$*, which minimizes:*

$$E\left[\|x(k) - \hat{x}(k)\|^2\right] \tag{4.13}$$

where the expectation is taken over the sensor and plant noises as well as the packet loss process.

The fact that we can observe the network performance, $\theta(k)$, is important. It implies that the combined plant, Equation 4.12, consists of a time-varying output equation where the estimator knows both $C_{\theta(k)}$ and $D_{\theta(k)}$. As a result, the optimal estimator is just the Kalman filter for a time-varying plant. The notation is given by:

$$\text{Optimal Estimate:} \qquad \hat{x}(k|j) = E[x(k)|\mathcal{Y}_j, \Theta_j] \tag{4.14}$$

$$\text{Estimation Error:} \qquad \tilde{x}(k|j) = x(k) - \hat{x}(k|j) \tag{4.15}$$

$$\text{Error Covariances:} \qquad Z(k) = E[\tilde{x}(k|k)\tilde{x}(k|k)^T] \tag{4.16}$$

$$M(k+1) = E[\tilde{x}(k+1|k)\tilde{x}(k+1|k)^T] \tag{4.17}$$

Theorem 4.2 *The optimal state estimate is given by the Kalman filter:*

$$\hat{x}(k|k) = \hat{x}(k|k-1) + F(k)[y_c(k) - C_{\theta(k)}\hat{x}(k|k-1)]$$

$$\hat{x}(k|k-1) = A\hat{x}(k-1|k-1) \tag{4.18}$$

where the observer gain and error covariances are found from the following recursions:

$$F(k) = M(k)C_{\theta(k)}^T \left[C_{\theta(k)}M(k)C_{\theta(k)}^T + D_{\theta(k)}VD_{\theta(k)}^T\right]^{-1}$$

$$Z(k) = M(k) - M(k)C_{\theta(k)}^T \left[C_{\theta(k)}M(k)C_{\theta(k)}^T + D_{\theta(k)}VD_{\theta(k)}^T\right]^{-1} C_{\theta(k)}M(k)$$

$$M(k+1) = AZ(k)A^T + BWB^T$$

The Kalman filter is started with the initial condition $\hat{x}(0|-1) = x_0$ *and initial error covariance given by* $M(0) = X_0$.

Proof. The combined plant, Equation 4.12, is a simple case of a jump linear system where the plant mode is known. While studying the Jump Linear Quadratic Gaussian problem, Chizeck and Ji noted that the optimal state estimate for such systems is obtained with a time varying Kalman filter [17, 42]. Kalman derived the optimal estimator for linear, discrete, time-varying plants in [44]. The forward Riccati Difference equation gives the estimation error covariance for a particular realization of the packet loss process. ■

4.4.2 Estimator Stability

The Kalman Filter gives the optimal state estimate in terms of the least squares cost given in Equation 4.13. This cost can be written as:

$$E\left[\|x(k) - \hat{x}(k)\|^2\right] = tr\left[\underset{\theta(0),\ldots,\theta(k)}{E}[Z(k)]\right] \tag{4.19}$$

where $tr[\cdot]$ denotes the trace of a matrix and the expectation on the right side is taken over the packet loss process. For the remainder of this section, the expectations are taken over the packet loss process, $\{\theta(0), \ldots, \theta(k)\}$. While the Kalman filter is optimal, it is not guaranteed to be stable in the sense of this cost staying bounded as $k \to \infty$. The goal of this

section is determine conditions on the network such that this cost stays bounded in time. For a symmetric, positive semidefinite matrix, the trace of matrix can be upper and lower bounded by its maximum singular value (denoted by $\bar{\sigma}(A)$ for a matrix A): $A = A^T \in R^{n\times n}$ and $A \geq 0 \Rightarrow \bar{\sigma}(A) \leq tr[A] \leq n\bar{\sigma}(A)$. Thus the cost in Equation 4.19 stays bounded if and only if $\bar{\sigma}(E[Z(k)])$ stays bounded. Moreover, $\bar{\sigma}(E[Z(k)])$ stays bounded if and only if $\bar{\sigma}(E[M(k)])$ stays bounded. For simplicity, we now focus on the propagation of $M(k)$ and the boundedness of $\bar{\sigma}(E[M(k)])$.

If $p = 0$, i.e there is no probability of packet loss, then the plant is time-invariant. $M(k)$ propagates forward via the standard Riccati Difference equation:

$$M(k+1) = AM(k)A^T + BWB^T - AM(k)C^T[CM(k)C^T + V]^{-1}CM(k)A^T \tag{4.20}$$

If (A,C) is detectable, then one can show that the variance, $M(k)$, is uniformly bounded in time [10]. That is, there exists a matrix U such that $M(k) \leq U$ for all k. Hence $\bar{\sigma}(M(k)) \leq \bar{\sigma}(U)$ for all k and the cost stays bounded in time. Conversely, if we drop all packets ($p = 1$), then the best estimate is obtained by propagation through our model of the plant. In this case $M(k)$ propagates forward as follows:

$$M(k+1) = AM(k)A^T + BWB^T \tag{4.21}$$

If the plant is unstable, then $M(k)$ will grow unbounded as it propagates in time. As a result, the cost will grow unbounded.

When $p \in (0, 1)$, the propagation of $M(k)$ is a stochastic process which depends

on $\theta(k)$:

$$M(k+1) = \begin{cases} AM(k)A^T + BWB^T & \text{if } \theta(k)=0 \\ AM(k)A^T + BWB^T - AM(k)C^T[CM(k)C^T + V]^{-1}CM(k)A^T & \text{if } \theta(k)=1 \end{cases} \tag{4.22}$$

The objective is to determine the largest packet loss rate such that $E[M(k)]$ stays bounded as $k \to \infty$. The following theorem gives a necessary condition for $E[M(k)]$ to stay bounded. This condition is also sufficient when C is nonsingular. C nonsingular corresponds to the case where we can observe the full state whenever a packet is received. Define the spectral radius of A as $\rho(A) := max|\lambda_i(A)|$ where $\lambda_i(A)$ are the eigenvalues of A.

Theorem 4.3 *We make the technical assumption that (A, B) is stabilizable, i.e. the plant noise actually disturbs the state dynamics. Given the plant (Equation 4.12) and Kalman filter (Equation 4.18), then uniform boundedness of $\bar{\sigma}\left(E[M(k)]\right)$ implies $p\rho(A)^2 < 1$. If C is nonsingular, then $p\rho(A)^2 < 1$ is sufficient for $\bar{\sigma}\left(E[M(k)]\right)$ to stay bounded as $k \to \infty$.*

Proof. First we show that $\bar{\sigma}\left(E[M(k)]\right)$ grows unbounded if $p\rho(A)^2 \geq 1$. From the propagation relation, Equation 4.22, we have:

$$\begin{aligned} E\left[M(k+1) \mid M(k) = M\right] &= p\left(AMA^T + BWB^T\right) \\ &\quad +(1-p)\left(AMA^T + BWB^T - AMC^T[CMC^T + V]^{-1}CMA^T\right) \end{aligned} \tag{4.23}$$

$AMA^T + BWB^T - AMC^T[CMC^T + V]^{-1}CMA^T$ is the error variance if $M(k) = M$ and we receive a packet, $\theta(k) = 1$. Since a variance matrix is positive semidefinite, we conclude the second term of Equation 4.23 is positive semidefinite. The same conclusion can be reached from simple linear algebra arguments. Either way, $E[M(k+1)|M(k) = M]$ can be lower

bounded as follows:

$$E\left[M(k+1) \mid M(k)=M\right] \geq p\left(AMA^T + BWB^T\right) \tag{4.24}$$

The importance of this lower bound is that it is a linear function of M and we can apply it inductively. Let v the left eigenvector of A which achieves $\rho(A)$ and is normalized to $\|v\| = 1$. Apply the lower bound to obtain:

$$\begin{aligned} E[v^T M(k+1)v] &\overset{(a)}{=} E\left[E[v^T M(k+1)v \mid M(k)]\right] \\ &\geq E[p\rho(A)^2 v^T M(k)v + pv^T BWB^T v] \\ &= p\rho(A)^2 E[v^T M(k)v] + pv^T BWB^T v \end{aligned} \tag{4.25}$$

On the right side of Equality (a), the inner expectation is taken over $\theta(k+1)$ while the outer expectation is taken over $\{\theta(0), \ldots, \theta(k)\}$. Since $p \in [0,1]$ and $\rho(A)^2 \geq \frac{1}{p}$ by assumption, the eigenvector v corresponds to an unstable mode of the plant. (A, B) stabilizable implies that $v^T B \neq 0$. The lower bound obtained in Equation 4.25 implies that $E[v^T M(k+1)v] \to \infty$ as $k \to \infty$ and hence $\bar{\sigma}\left(E[M(k)]\right)$ grows unbounded.

Next we show that if C is nonsingular and $p\rho(A)^2 < 1$, then $E[M(k)]$ is uniformly bounded. We return to the propagation relation in Equation 4.23 and try to upper bound the last term. To this end, we use properties of the maximum singular value to obtain the following bound:

$$\begin{aligned} \bar{\sigma}\left(M - MC^T[CMC^T + V]^{-1}CM\right) &\overset{(a)}{=} \bar{\sigma}\left(C^{-1}V(CMC^T + V)^{-1}CM\right) \\ &\overset{(b)}{\leq} \bar{\sigma}\left(C^{-1}V\right)\bar{\sigma}\left((CMC^T + V)^{-1}CMC^T\right)\bar{\sigma}\left(C^{-T}\right) \\ &\overset{(c)}{\leq} \bar{\sigma}\left(C^{-1}V\right)\bar{\sigma}\left(C^{-1}\right) \end{aligned}$$

Equality (a) follows because $M - MC^T[CMC^T + V]^{-1}CM = C^{-1}V(CMC^T + V)^{-1}CM$ when C is invertible. Inequality (b) follows from the submultiplicative property of the maximum singular value: $\bar{\sigma}(AB) \leq \bar{\sigma}(A)\bar{\sigma}(B)$ [40]. If two symmetric matrices satisfy $A > B \geq 0$ then $\bar{\sigma}\left(A^{-1}B\right) \leq 1$. Inequality (c) follows from this fact and $\bar{\sigma}\left(A^T\right) = \bar{\sigma}(A)$.

The final term of Equation 4.23 can be upper bounded using this result:

$$(1-p)A\left(M - MC^T[CMC^T + V]^{-1}CM\right)A^T \leq (1-p)\bar{\sigma}(A)^2\,\bar{\sigma}\left(C^{-1}V\right)\bar{\sigma}\left(C^{-1}\right)\cdot I \quad (4.26)$$

where I is an $n_x \times n_x$ identity matrix. The key point is that this term is upper bounded by a matrix which is independent of M. We can use this to bound $E[M(k+1)]$:

$$\begin{aligned} E[M(k+1)] &\overset{(a)}{=} E\left[E[M(k+1)|M(k)]\right] \\ &\overset{(b)}{\leq} E\left[pAM(k)A^T + BWB^T + (1-p)\bar{\sigma}(A)^2\,\bar{\sigma}\left(C^{-1}V\right)\bar{\sigma}\left(C^{-1}\right)\cdot I\right] \\ &\overset{(c)}{=} pAE[M(k)]A^T + BWB^T + (1-p)\bar{\sigma}(A)^2\,\bar{\sigma}\left(C^{-1}V\right)\bar{\sigma}\left(C^{-1}\right)\cdot I \end{aligned}$$

On the right side of Equality (a), the inner expectation is taken over $\theta(k+1)$ while the outer expectation is taken over $\{\theta(0), \ldots, \theta(k)\}$. Inequality (b) follows by applying the bound in Equation 4.26 to the propagation relation in Equation 4.23. Equality (c) follows because $M(k)$ is the only term depending on the packet loss process and the upper bound is linear in $M(k)$. This bound can be written more compactly as:

$$E[M(k+1)] \leq pAE[M(k)]A^T + Q$$

where Q is the appropriately defined positive definite matrix. If $p\rho(A)^2 < 1$, then we can use this upper bound inductively to show that $E[M(k+1)]$ is uniformly bounded as $k \to \infty$. From this we conclude that $\bar{\sigma}(E[M(k+1)])$ is uniformly bounded. ■

4.4.3 Generalized Estimator Stability Result

One might conclude that $p < \frac{1}{\rho(A)^2}$ is necessary and sufficient for boundedness of $\bar{\sigma}\left(E[M(k)]\right)$ even under the less restrictive assumption of (A,C) observable. This intuition is false as the following example shows. Consider a second order system, $\mathbf{P}$, in the form of Equation 4.10:

$$A = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$V = 1, \qquad W = 1, \qquad M(0) = \begin{bmatrix} m_{11}(0) & 0 \\ 0 & m_{22}(0) \end{bmatrix}$$

This plant satisfies the assumptions of (A, B) stabilizable and (A,C) observable. The condition in Theorem 4.3, if applied to this example, predicts that $E[M(k)]$ stays bounded if and only if $p < \frac{1}{\rho(A)^2} = 0.25$. We now show that this condition does not apply for this example.

We use the notation $M(2)|_{(\theta_0,\theta_1)}$ to denote the value of $M(2)$ propagated from $M(0)$ along the path (θ_0, θ_1). $M(2)$ is computed using the propagation relation in Equation 4.22. Using this notation, we can computed the expected value of $M(2)$ as:

$$E[M(2)] = p^2 M(2)|_{(0,0)} + p(1-p)\left(M(2)|_{(0,1)} + M(2)|_{(1,0)}\right) + (1-p)^2 M(2)|_{(1,1)}$$

$M(2)|_{(\theta_0,\theta_1)} \geq 0$ for any (θ_0, θ_1) since it is the estimation error variance for the given path.

Using this fact, we can lower bound $E[M(2)]$ by:

$$\begin{aligned} E[M(2)] &\overset{(a)}{\geq} p^2 M(2)|_{(0,0)} + p(1-p)M(2)|_{(1,0)} \\ &\overset{(b)}{\geq} pA^2M(0)(A^T)^2 - p(1-p)A^2M(0)C^T[CM(0)C^T+V]^{-1}CM(0)(A^T)^2 \\ &\overset{(c)}{=} \begin{bmatrix} \frac{16p^2m_{11}^2(0)+16pm_{11}(0)}{m_{11}(0)+1} & 0 \\ 0 & 16pm_{22}(0) \end{bmatrix} \end{aligned}$$

Inequality (a) follows because $M(2)|_{(1,1)} \geq 0$ and $M(2)|_{(1,0)} \geq 0$ as mentioned above. Inequality (b) follows from applying Equation 4.22 along the paths $(\theta_0, \theta_1) = (1,0)$, $(0,0)$ and then noting that $ABWB^TA^T$, BWB^T are positive semidefinite. Equality (c) follows from the given values of A, C, M(0), and V. Let $e_2 = [0\ 1]^T$ and apply this lower bound to obtain:

$$E[m_{22}(2)] = E[e_2^T M(2) e_2] \geq 4^2 p m_{22}(0)$$

Using induction, this inequality can be generalized:

$$E[m_{22}(2N)] = E[e_2^T M(2N) e_2] \geq 4^{2N} p^N m_{22}(0) = [16p]^N m_{22}(0)$$

If $p > \frac{1}{16} = .0625$ then $16p > 1$ and we can conclude that $E[M(k)]$ grows unbounded. Thus $p < \frac{1}{\rho(A)^2} = 0.25$ is not a necessary and sufficient condition for boundedness of $E[M(k)]$ when (A,C) is observable. It is clear that observability is lost along the path of where all packets are dropped ($\theta(k) = 0\ \forall k$). The problem displayed in this example is that observability is lost along other paths even if packets are received frequently. Along the path $(\theta_0, \theta_1, \ldots) = (1,0,1,0,\ldots)$, the output of the system (neglecting plant and sensor noise) is:

$$(y_c(0), y_c(1), y_c(2), y_c(3), \ldots) = (Cx(0), 0, CA^2x(0), 0, \ldots) \tag{4.27}$$

By construction, $A^2 = 4\left[\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix}\right]$ and in general A^{2k} is a multiple of the identity for all k. Thus the plant appears unobservable along this path since we can only observe the state space along the direction given by $C = [1\ 0]$. To ensure boundedness of $E[M(k)]$, we must ensure that the probability of an unobservable path occurring is sufficiently small.

Theorem 4.3 needs to be generalized to the case where (A, C) is observable. Let $A \in \mathbb{R}^{n\times n}$ and assume (A, C) is observable. If $rank[A^{i_1}C\ \ldots\ A^{i_n}C] = n$ for any collection of distinct, nonnegative integers, $\{i_1, \ldots, i_n\}$, then $p\rho(A)^2 < 1$ is still necessary and sufficient for boundedness of $\bar{\sigma}\left(E[M(k)]\right)$. It is not clear for what systems this 'strong observability' property is satisfied and hence this proof is omitted. A second method to extend Theorem 4.3 is to estimate the state prior to transmission across the network. In other words, (A, C) observable implies that we can design a stable observer to estimate x from y. We then transmit this estimate, $\hat{x}_1$, across the network. The estimator on the receiving end of the network constructs a second state estimate, $\hat{x}_2$, from the communicated information, $\hat{x}_{1c}$. If the first estimator is stable, then our goal is to make $\hat{x}_2$ estimate $\hat{x}_1$ using $\hat{x}_{1c}$. We have effectively turned the problem into an estimation problem of the form specified in Theorem 4.3.

4.5 Implications for NCS Design

The goal of this chapter was to derive bounds on the required network performance that are independent of a particular control law. For a simple vehicle following problem, we determined the level of network performance required for a stabilizing controller to exist. The network performance was measured in terms of packet loss rate, but we now show

that the condition inherently constrains network bandwidth and packet loss rate. Let us consider how this condition can be applied. Suppose the dynamics of vehicle 1 in Figure 4.1 are modeled by a continuous time system with state matrix $A_c \in \mathbb{R}^{n\times n}$. Assume that vehicle 1 can measure its full state, $y_1(t) = x_1(t)$, and the sampling interval is T seconds. The state matrix for the discrete time representation of vehicle 1 is given by $e^{A_c T}$. From the results contained in this chapter, we conclude that a stabilizing controller for the model following problem exists if and only if $p\rho(e^{A_c T})^2 < 1$. Now suppose that each state measurement is quantized to b bits. At the given sample rate we must send $R := \frac{bn}{T}$ bits per second of useful information. The actual rate will need to be higher to include the overhead of addressing, error checking bits, etc. Neglecting this fact, we can rewrite the constraint in terms of packet loss rate, p, and network bandwidth, R, as follows: $p\rho(e^{A_c(bn)/R})^2 < 1$.

A simple example illustrates this constraint. Consider the following continuous time state matrix:

$$A_c = \begin{bmatrix} 1.06 & -0.83 & 0.71 \\ 0.06 & 0.29 & 1.62 \\ -0.10 & -1.34 & -0.69 \end{bmatrix}$$

The eigenvalues of A_c are 1.04 and -0.19 ± 1.42. The state dimension is $n = 3$ and we assume that the state measurements are quantized with $b = 16$ bits. Under the assumption of full-state measurements (C nonsingular), a stabilizing controller exists if $p\rho(e^{A_c(48/R)})^2 < 1$. Figure 4.4 shows the feasible region for the existence of a stabilizing controller. Since the we are considering a randomly generated plant, the actual numerical results are not significant. However, the qualitative result that we seek is important. Specifically, this plot gives constraints on the network performance that must be satisfied for a stabilizing

controller to exist. Under the assumption of Gaussian noise, Theorem 4.1 gives the optimal controller for the LQG problem. Thus if this condition fails, then no controller, even if it is time-varying and/or nonlinear, can keep the LQG cost finite. This condition provides a hard constraint on network performance that must be satisfied for acceptable control.

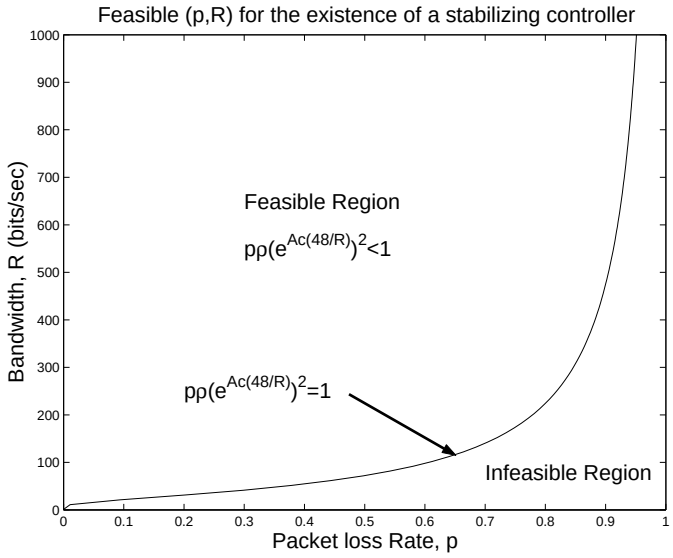

Figure 4.4: Feasible (p,R) region for the existence of a stabilizing controller

Conditions on the network performance that are independent of the control law are extremely general and useful results. The condition we have derived is one step in that direction, but it can certainly be extended in many ways. In the short term, Theorem 4.3 must be extended to find necessary and sufficient conditions for the existence of a stable estimator. If (A, C) is observable, Theorem 4.3 only gives a necessary condition on the network performance for the existence of a stable estimator. The next goal should be to include some measure of performance in this simple problem. Stabilization is a rather weak condition and this is clearly evidenced in Theorem 4.3. If the plant is stable, then

the condition is satisfied for any $p \in [0, 1]$. In other words, no information is required (i.e. we can lose all packets) to synthesize a stable estimator. Clearly some notion of control performance is required to strengthen the requirements on the network performance. Finally, the results evidenced by this simple vehicle following problem need to be extended to large scale, networked control systems.

Chapter 5

Controller Analysis and Synthesis for Networked Control Systems

5.1 Introduction

In the previous chapter, we described networked control systems (NCS) and discussed some related work in this area. We then gave a theoretical analysis for a simple vehicle following problem. Our ultimate vision is to coordinate large numbers of unmanned aerial vehicles (UAVs) to accomplish a complex task. The clusters of UAVs will communicate on an ad-hoc wireless network. Figure 5.1 shows that the nodes of the network may consist of the UAVs themselves, ground robots, and a satellite backbone. In contrast to the Automated Highway System problem, there may be malicious forces which try to jam communication links. To combat this possibility, the UAVs will be able to form alternative communication paths to the destination. As a concrete example, suppose that cluster C_1

is performing a task with cluster C_2 and they are exchanging information on link L_1. If the clusters sense that the performance of this wireless link is too poor, they may reroute packets up through the satellite backbone, which is assumed to be more reliable. Note that the backbone does not have sufficient resources to handle all information from all clusters. Thus it is important for clusters to route information through the satellite network only when control performance objectives cannot be met otherwise. Knowledge of the bounds on acceptable network performance is key to making this distributed agent system robust in hostile environments.

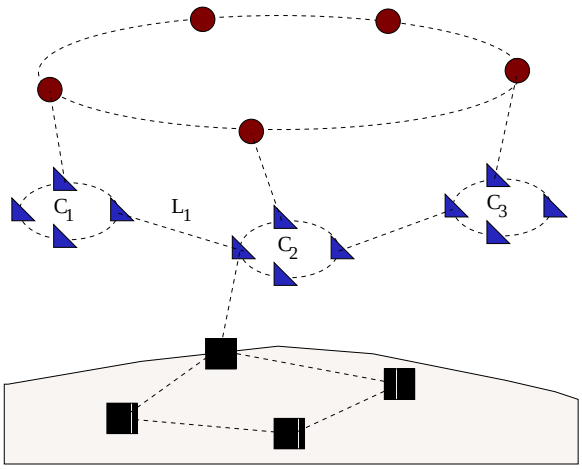

Figure 5.1: Coordinated UAV Network

In this chapter, we develop a computationally efficient method to solve problems related to formation flight of UAVs. Specifically, we develop a tractable condition for the analysis problem: Given an open loop model and a controller, determine how the closed-loop performance degrades as the probability of a packet loss increases. Using this condition, a control engineer can determine what probability of packet loss can be tolerated before the performance degradation is unacceptable. We also develop a tractable condition for the

centralized controller synthesis problem: Given an open loop plant and a level of network performance, synthesize a centralized controller that is optimal with respect to some cost function. This condition can be used to give the limits of closed loop performance for a given probability of packet loss. If a control performance objective is specified, this will then yield a hard bound that the network must satisfy. We will not develop a tractable condition for the decentralized controller synthesis problem: Given an open loop plant, a level of network performance, and structural constraints on the controller, synthesize a controller that is optimal with respect to some cost function. Such a condition would be extremely useful as it would allow us to synthesize decentralized controllers of various forms and investigate the limits of achievable performance. We could then determine how the cost of communicating additional information improves the closed loop performance. Unfortunately, this problem is extremely hard from a computational point of view. Hence we will leave this question unresolved for now and note that future research should look for tractable conditions which can approximately solve this problem.

As in the previous chapter, the networked control system consisting of UAVs in formation flight falls into the class of Markov Jump Linear Systems (MJLS). In the remainder of the chapter, we will develop an H_∞ control design method for discrete-time MJLS. First we give a review of past work on MJLS. Then we present useful results on stability and H_∞ performance for MJLS. We will derive a necessary and sufficient matrix inequality for a MJLS to achieve a given level of H_∞ performance. This gives us a computationally tractable method to determine the performance of a networked control system. In Section 5.6, we show how this matrix inequality can be used to synthesize optimal controllers. Finally we

present several examples demonstrating the usefulness of the tools. In the following chapter, we show that UAVs in formation flight can be modeled using the MJLS framework. We then apply the tools developed in this chapter to analyze the effect of packet losses on a formation flight controller.

5.2 Related Work

As noted in the previous chapter, a NCS can be modeled as a jump linear system [13, 47, 81, 108, 110]. The benefit of this modeling approach is that there is a large body of literature for this class of systems. In this section, we review some of the results available for Markovian Jump Linear Systems (MJLS).

Chizeck and Ji have obtained many results for discrete-time MJLS. Chizeck, et.al. extended Linear Quadratic control for MJLS [18]. Ji and Chizeck subsequently defined and developed algebraic conditions for controllability and observability of MJLS [41]. These ideas led to a solution of the Jump Linear Quadratic Gaussian control problem [17, 42].

Ji, et.al. have also shown that several notions of stability are equivalent for MJLS [43]. Specifically, mean square stability, stochastic stability, and exponential mean square stability are equivalent and any one implies almost sure stability. The authors refer to the equivalent notions stability as second-moment stability (SMS). It is interesting to note that stability of each mode is neither necessary nor sufficient for the system to be SMS. See [42] for several examples of this and other interesting properties of MJLS. Costa and Fragoso have established that mean square stability is equivalent to the existence of a solution to a Lyapunov equation [21]. The significance of this result is that the Lyapunov condition is

easily checkable. We will discuss these stability results in greater detail in Section 5.4.

Control design for MJLS has focused on using convex and nonconvex optimization for synthesis. For example, state feedback controllers for MJLS can be found by solving a set of linear matrix inequalities (LMIs) [7, 11, 20, 33, 72]. The benefit of this approach is that the problem can be efficiently solved by interior-point methods [9, 32] with a guarantee that the global optimum will be found. However, as noted in [108], the structure of the networked control problem results in an output feedback even if the problem was originally state feedback. It should be clear from the previous chapter that the output of the combined plant/network model viewed by the controller is not the full state if a packet is dropped. The output feedback problem for MJLS is quite complex when placed in the optimization framework. This problem has been attacked as a nonconvex optimization problem in [71] for the continuous time case and in [108] for the discrete time case. Unfortunately, these routines may converge to local extrema and do not guarantee convergence to the global optimum. Another approach is to use a congruence transformation to convert the problem into an LMI optimization problem. This approach was used in [24] to find mode-dependent dynamic output feedback controllers for continuous time MJLS. In this chapter, we take the latter approach to find LMI conditions for control design. The drawback is that the congruence transformation only converts the problem to an LMI for a restricted class of MJLS.

5.3 Markovian Jump Linear Systems

In this section, we give notation that will be used throughout the remainder of the chapter. Consider the following stochastic system, denoted $\mathcal{S}$:

$$\begin{bmatrix} x(k+1) \\ z(k) \end{bmatrix} = \begin{bmatrix} A_{\theta(k)} & B_{\theta(k)} \\ C_{\theta(k)} & D_{\theta(k)} \end{bmatrix} \begin{bmatrix} x(k) \\ w(k) \end{bmatrix} \tag{5.1}$$

where $x(k) \in \mathbb{R}^{n_x}$ is the state, $w(k) \in \mathbb{R}^{n_w}$ is the disturbance vector and $z(k) \in \mathbb{R}^{n_z}$ is the error vector. The state matrices are functions of a discrete-time Markov chain taking values in a finite set $\mathcal{N} = \{1, \ldots, N\}$. The Markov chain has a transition probability matrix $\mathcal{P} = [p_{ij}]$ where $p_{ij} = \Pr(\theta(k+1) = j \mid \theta(k) = i)$ subject to the restrictions $p_{ij} \geq 0$ and $\sum_{j=1}^{N} p_{ij} = 1$ for any $i \in \mathcal{N}$. When the plant is in mode $i \in \mathcal{N}$ (i.e. $\theta(k) = i$), we will use the following notation: $A_i := A_{\theta(k)}$, $B_i := B_{\theta(k)}$, $C_i := C_{\theta(k)}$, and $D_i := D_{\theta(k)}$. Plants of this form are called discrete-time Markovian jump linear systems. As a concrete example, the augmented plant for the vehicle following problem, Equation 4.4, is a MJLS with two modes: mode 1 if a packet is dropped and mode 2 if a packet is received. The transition probability matrix is $\mathcal{P} = \begin{bmatrix} p & 1-p \\ p & 1-p \end{bmatrix}$.

We also define ℓ_2 as the space of square summable (deterministic) sequences. That is, $w = (w(0), w(1), \ldots) \in \ell_2^n$ if $w(k) \in \mathbb{R}^n \ \forall k$ and $\|w\|_2 < \infty$, where the norm is defined by:

$$\|w\|_2 = \left(\sum_{k=0}^{\infty} w(k)^T w(k) \right)^{1/2} \tag{5.2}$$

If we input the deterministic sequence, w, into the MJLS given by Equation 5.1, then $x = (x(0), x(1), \ldots)$ and $z = (z(0), z(1), \ldots)$ are stochastic sequences depending on the sequence of Markov parameters, $\theta = (\theta(0), \theta(1), \ldots)$. The norm for the output sequence

will be given by:

$$\|z\|_{2E} = \left(E \left\{ \sum_{k=0}^{\infty} z(k)^T z(k) \right\} \right)^{1/2} \tag{5.3}$$

where the expectation is taken over the sequence of Markov parameters, $\{\theta(0), \ldots\}$. The notation $\|\cdot\|_{2E}$ is similarly used for any other sequence, such as x, depending on $\{\theta(0), \ldots\}$.

5.4 Stability Results

In this section, we review several useful results related to the stability of discrete-time jump linear systems. First we define several forms of stability for such systems [43].

Definition 1 *For the system given by (5.1) with $w \equiv 0$, the equilibrium point at the origin is:*

1. *mean-square stable if for every initial state (x_0, θ_0), $\lim_{k\to\infty} E\{\|x(k)\|^2 \mid x_0, \theta_0\} = 0$.*
2. *stochastically stable if for every initial state (x_0, θ_0), $E\left\{\sum_{k=0}^{\infty} \|x(k)\|^2 \mid x_0, \theta_0\right\} < \infty$. In other words, $\|x\|_{2E} < \infty$ for every initial state.*
3. *exponentially mean square stable if for every initial state (x_0, θ_0), there exists constants $0 < \alpha < 1$ and $\beta > 0$ such that $\forall k \geq 0$, $E\left\{\|x(k)\|^2 \mid x_0, \theta_0\right\} < \beta\alpha^k \|x_0\|^2$.*
4. *almost surely stable if for every initial state (x_0, θ_0), $\Pr\{\lim_{k\to\infty} \|x(k)\| = 0\} = 1$.*

We must determine which notion of stability we seek when dealing with the jump systems. Ji, et.al. showed that for System (5.1), the first three definitions of stability are actually equivalent [43]. The authors refer to the equivalent notions of mean-square, stochastic, and exponential mean square stability as second-moment stability (SMS). Moreover, SMS is sufficient but not necessary for almost sure stability. In the remainder of the

chapter, references to stability will be in the sense of second-moment stability. The major motivation for this choice is that computable conditions exist to check for SMS but not for almost-sure stability. Below we present necessary and sufficient matrix inequality conditions for SMS of the jump linear system. Theorem 5.1 was proved in [21] by showing equivalence of the matrix inequality conditions to mean-square stability.

Theorem 5.1 *System (5.1) is SMS if and only if there exists matrices* $G_i > 0$ *for* $i = 1, \ldots, N$ *that satisfy any of the following conditions:*

1. $G_i - A_i^T \left(\sum_{j=1}^N p_{ij} G_j \right) A_i > 0$ *for* $i = 1, \ldots, N$
2. $G_j - A_j \left(\sum_{i=1}^N p_{ij} G_i \right) A_j^T > 0$ *for* $j = 1, \ldots, N$
3. $G_i - \sum_{j=1}^N p_{ij} A_j^T G_j A_j > 0$ *for* $i = 1, \ldots, N$
4. $G_j - \sum_{i=1}^N p_{ij} A_i G_i A_i^T > 0$ *for* $j = 1, \ldots, N$

Thus SMS is equivalent to finding N positive definite matrices which satisfy N discrete Lyapunov conditions in a stochastic sense. We will actually apply a simplified version of this Theorem which is presented in [21]. For completeness, the following Theorem is proved in Appendix B using a stochastic Lyapunov function approach.

Theorem 5.2 *If* $p_{ij} = p_j$ *for all* $i, j \in \mathcal{N}$ *then System (5.1) is SMS if and only if there exists a matrix* $G > 0$ *such that:*

$$G - \sum_{j=1}^{N} p_j A_j^T G A_j > 0 \tag{5.4}$$

In words, if $\theta(k)$ is an independent process, we only need to find one positive definite matrix such that the associated Lyapunov function decreases on average at every step. It

is interesting to note that stability of each mode is neither necessary nor sufficient for the system to be SMS. See [42] for several examples of this and other properties of MJLS.

5.5 H_∞ Results

Next we give the definition of the H_∞ norm for discrete-time MJLS. We consider disturbances, $w \in \ell_2^{n_w}$, to the jump system given by Equation 5.1. The H_∞ norm of the system [20] is defined below.

Definition 2 *Assume the system, $\mathcal{S}$, given by Equation 5.1 is an SMS system. Let $x(0) = 0$ and define the H_∞ norm, denoted $\|\mathcal{S}\|_\infty$, as:*

$$\|\mathcal{S}\|_\infty := \sup_{0 \neq w \in \ell_2^{n_w}} \frac{\|z\|_{2E}}{\|w\|_2} \tag{5.5}$$

Below we derive a matrix inequality condition for a MJLS to achieve a given level of H_∞ performance. To derive this condition, we need a stochastic definition of controllability given by Ji and Chizeck [41]. The following is a simplification of their definition for the case where $p_{ij} = p_j$.

Definition 3 *Consider system, $\mathcal{S}$, given by Equation 5.1 and transition probabilities satisfying $p_{ij} = p_j$. For every initial state, x_0, and any choice of target value x_f, if $T_c = \inf_w\{k > 0 : \Pr(x(k) = x_f) > 0\}$ is finite, then the system is weakly controllable.*

Next we present a necessary and sufficient matrix inequality condition for a restricted class of MJLS to satisfy a given level of H_∞ performance. The weak controllability ensures that the disturbance can, in a stochastic sense, affect the system state. If the system is not weakly controllable, the matrix condition is still sufficient, but it may not be necessary.

Theorem 5.3 *Assume $p_{ij} = p_j$ for all $i, j \in \mathcal{N}$ and the system, $\mathcal{S}$, given by Equation 5.1 is weakly controllable. $\mathcal{S}$ is SMS and satisfies $\|\mathcal{S}\|_\infty < \gamma$ if and only if there exists a symmetric matrix $G > 0$ satisfying the following matrix inequality:*

$$\begin{bmatrix} G & 0 \\ 0 & \gamma^2 I \end{bmatrix} - \sum_{j=1}^{N} p_j \begin{bmatrix} A_j & B_j \\ C_j & D_j \end{bmatrix}^T \begin{bmatrix} G & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A_j & B_j \\ C_j & D_j \end{bmatrix} > 0 \tag{5.6}$$

Theorem 5.3 is proved in Appendix C. A stochastic Lyapunov approach is used to show sufficiency of the matrix inequality condition. Necessity follows by showing that $\|\mathcal{S}\|_\infty < \gamma$ implies that a related Riccati equation has a solution. The Riccati equality can be converted to an inequality by a perturbation argument which then leads to the matrix inequality given in Equation 5.6. We make several remarks concerning this H_∞ condition. We note that for the case of one mode ($N = 1$), the H_∞ condition given by Equation 5.6 reduces to the standard necessary and sufficient condition [66]. This condition also has an interpretation in terms of the coordinate transformation: $\tilde{x}(k) := G^{1/2}x(k)$. Multiplying the matrix condition on the left/right by $\left[\begin{smallmatrix} G^{-1/2} & 0 \\ 0 & \gamma^{-1}I \end{smallmatrix}\right]$ shows that the search for a suitable scaling matrix, G, is equivalent to a search for a single coordinate transformation (for all modes), $G^{1/2}$, which makes the plant satisfy a singular value condition, $\bar{\sigma}\left(\left[\begin{smallmatrix} I & 0 \\ 0 & \gamma^{-1} \end{smallmatrix}\right]\left[\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right]\left[\begin{smallmatrix} I & 0 \\ 0 & \gamma^{-1} \end{smallmatrix}\right]\right) < 1$, in a stochastic sense.

This Theorem can be tied to the condition for SMS. Using the proof of Theorem 5.2 in Appendix B, if the system is SMS then there exists $G > 0$ such that:

$$G - \sum_{j=1}^{N} p_j A_j^T G A_j = \sum_{j=1}^{N} p_j C_j^T C_j + I$$

For γ sufficiently large, G satisfies Equation 5.6. This can be seen by using Schur complements[1] to put Equation 5.6 in the equivalent form:

$$G - \sum_{j=1}^{N} p_j A_j^T G A_j > \tilde{W} \tag{5.7}$$

where:

$$\tilde{W} = \sum_{j=1}^{N} p_j C_j^T C_j + \left[\sum_{j=1}^{N} p_j \left(B_j^T G A_j + D_j^T C_j\right)\right]^T \left[\gamma^2 I - \sum_{j=1}^{N} p_j \left(B_j^T G B_j + D_j^T D_j\right)\right]^{-1} \cdot \left[\sum_{j=1}^{N} p_j \left(B_j^T G A_j + D_j^T C_j\right)\right]$$

If $\mathcal{S}$ is SMS then there exists $\gamma > 0$ and $G > 0$ which satisfy Equation 5.6. In other words, if a system is SMS, then there exists some finite $\gamma > 0$ such that $\|\mathcal{S}\|_\infty < \gamma$. By contraposition, if we cannot find $G > 0$ and $\gamma > 0$ to satisfy Equation 5.6, then the system is not SMS.

In the following section, we will use this matrix inequality condition for controller synthesis. It will turn out that the controller matrices will be embedded in the A_j, B_j, and C_j matrices. Unfortunately, Equation 5.6 contains quadratic and cross terms which are functions of these matrices. Our ultimate goal is to arrive at a linear matrix inequality (LMI) condition, i.e. a condition which is a linear function of unknown matrices. Before preceding, we will present an equivalent form of Equation 5.6 which is more suitable for this purpose.

Let $Z = G^{-1}$ and multiply Equation 5.6 on the left and right by the following congruence transformation: $\left[\begin{smallmatrix} Z & 0 \\ 0 & I \end{smallmatrix}\right]$. Since positive definiteness is invariant under congruence

[1] See [9]: $R > 0$ and $Q - SR^{-1}S^T > 0$ if and only if $\left[\begin{smallmatrix} Q & S \\ S^T & R \end{smallmatrix}\right] > 0$

transformations, Equation 5.6 is equivalent to:

$$\begin{bmatrix} Z & 0 \\ 0 & \gamma^2 I \end{bmatrix} - \sum_{j=1}^{N} p_j R_j^T \begin{bmatrix} Z^{-1} & 0 \\ 0 & I \end{bmatrix} R_j > 0$$

where $R_j = \left[\begin{smallmatrix} A_j Z & B_j \\ C_j Z & D_j \end{smallmatrix}\right]$. Use Schur complements to convert this inequality into the following equivalent condition where $(\bullet)^T$ denote entries which can be inferred from the symmetry of the matrix:

$$\begin{bmatrix} \begin{bmatrix} Z & 0 \\ 0 & \gamma^2 I \end{bmatrix} & (\bullet)^T & \cdots & (\bullet)^T \\ \sqrt{p_1}\begin{bmatrix} A_1 Z & B_1 \\ C_1 Z & D_1 \end{bmatrix} & \begin{bmatrix} Z & 0 \\ 0 & I \end{bmatrix} & \cdots & (\bullet)^T \\ \vdots & & \ddots & \vdots \\ \sqrt{p_N}\begin{bmatrix} A_N Z & B_N \\ C_N Z & D_N \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \cdots & \begin{bmatrix} Z & 0 \\ 0 & I \end{bmatrix} \end{bmatrix} > 0 \tag{5.8}$$

5.6 H_∞ Controller Synthesis

In this section we will apply Theorem 5.3 to derive an LMI condition for controller synthesis. We consider plants, $\mathcal{S}$, of the form:

$$\begin{bmatrix} x(k+1) \\ z(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} A_{\theta(k)} & B_{1,\theta(k)} & B_{2,\theta(k)} \\ C_{1,\theta(k)} & D_{11,\theta(k)} & D_{12,\theta(k)} \\ C_{2,\theta(k)} & D_{21,\theta(k)} & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ w(k) \\ u(k) \end{bmatrix} \tag{5.9}$$

where $x(k) \in \mathbb{R}^{n_x}$ is the state, $u(k) \in \mathbb{R}^{n_u}$ is the control input, $y(k) \in \mathbb{R}^{n_y}$ is the measurement vector, $w(k) \in \mathbb{R}^{n_w}$ is the disturbance vector and $z(k) \in \mathbb{R}^{n_z}$ is the error vector. We consider the case where $\theta(k) \in \mathcal{N} = \{1, 2\}$, i.e. the plant has two modes. This is done for

clarity of exposition and the results of this section can easily be extended to the general case where the plant has N modes. We also assume that the probability matrix for the Markov process satisfies the constraint $p_{ij} = p_j \ \forall i, j \in \mathcal{N}$.

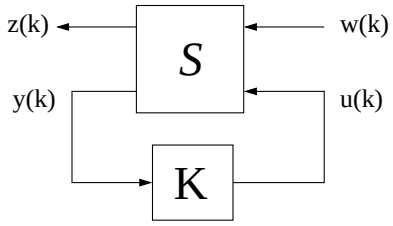

Figure 5.2: Feedback Loop for H_∞ Control Design

Our goal is to design a controller, K, for the feedback loop (Figure 5.2) which minimizes the closed loop gain from w to z. The gain from disturbances to errors is measured with the H_∞ norm. We assume that the controller has access to $\theta(k)$ and the output of the system, $y(k)$, but not the system state. The goal is to find an optimal H_∞ controller of the form:

$$\begin{aligned} x_c(k+1) &= A_{c,\theta(k)} x_c(k) + B_{c,\theta(k)} y(k) \\ u(k) &= C_{c,\theta(k)} x_c(k) \end{aligned} \tag{5.10}$$

where $x_c(k) \in \mathbb{R}^{n_c}$ is the controller state and the subscript c is used to denote the controller matrices/states. We will subsequently search for a controller of this form that gives the optimal H_∞ performance. We are not restricting the state dimension of the controller. The controller state dimension is only assumed to be finite, $n_c < \infty$. That is, we have not made any a priori assumptions on the state dimension of the controller. We will show in Theorem 5.4 that if a stabilizing controller giving a certain level of H_∞ performance exists, then we can always find a controller giving the same level of performance and with

state dimension equal to the state dimension of the original plant, $n_c = n_x$. On the other hand, it should be noted that we are restricting the search to finite dimensional, linear, time-varying controllers. More importantly, we are restricting the controller to depend only on the current plant mode, $\theta(k)$. Therefore, the controller is time-varying, but it can only hop between N modes depending on the current jump parameter $\theta(k)$. The LQG controller derived in the previous chapter was time-varying and used all past values of $\theta(k)$. This controller was time-varying and was not restricted to N modes.

With the controller structure above and the plant defined above, the closed loop becomes:

$$\begin{bmatrix} x_{cl}(k+1) \\ z(k) \end{bmatrix} = \begin{bmatrix} A_{cl,\theta(k)} & B_{cl,\theta(k)} \\ C_{cl,\theta(k)} & D_{cl,\theta(k)} \end{bmatrix} \begin{bmatrix} x_{cl}(k) \\ w(k) \end{bmatrix} \tag{5.11}$$

where:

$$A_{cl,\theta(k)} := \begin{bmatrix} A_{\theta(k)} & B_{2,\theta(k)}C_{c,\theta(k)} \\ B_{c,\theta(k)}C_{2,\theta(k)} & A_{c,\theta(k)} \end{bmatrix}$$

$$B_{cl,\theta(k)} := \begin{bmatrix} B_{1,\theta(k)} \\ B_{c,\theta(k)}D_{21,\theta(k)} \end{bmatrix}$$

$$C_{cl,\theta(k)} := \begin{bmatrix} C_{1,\theta(k)} & D_{12,\theta(k)}C_{c,\theta(k)} \end{bmatrix}$$

$$D_{cl,\theta(k)} := D_{11,\theta(k)}$$

The subscript 'cl' denotes the closed loop matrices. For the closed loop system, the transition probabilities satisfy the assumption given in Theorem 5.3: $p_{cl,ij} = p_{cl,j}$ for all i, j. Apply Theorem 5.3 to conclude that the closed loop system is SMS and has H_∞ gain less than γ if and only if there exists a matrix $0 < G \in \mathbb{R}^{(n_x+n_c)\times(n_x+n_c)}$ such that Equation 5.6 holds.

Or we can apply the equivalent condition given by Equation 5.8; the closed loop system has H_∞ gain less than γ if and only if there exists a matrix $0 < Z \in \mathbb{R}^{(n_x+n_c)\times(n_x+n_c)}$ such that:

$$\begin{bmatrix} \begin{bmatrix} Z & 0 \\ 0 & \gamma^2 I_{n_w} \end{bmatrix} & (\bullet)^T & (\bullet)^T \\ \sqrt{p_1}\begin{bmatrix} A_{cl,1}Z & B_{cl,1} \\ C_{cl,1}Z & D_{cl,1} \end{bmatrix} & \begin{bmatrix} Z & 0 \\ 0 & I_{n_z} \end{bmatrix} & (\bullet)^T \\ \sqrt{p_2}\begin{bmatrix} A_{cl,2}Z & B_{cl,2} \\ C_{cl,2}Z & D_{cl,2} \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} Z & 0 \\ 0 & I_{n_z} \end{bmatrix} \end{bmatrix} > 0 \tag{5.12}$$

A controller yielding closed loop H_∞ gain less than γ exists if and only if there exists $Z > 0$ and the appropriate controller matrices satisfying Equation 5.12. This is a bilinear matrix inequality since it is linear in the controller parameters (for a fixed scaling matrix Z) or in Z (for fixed controller matrices). The following theorem gives an equivalent linear matrix inequality condition.

Theorem 5.4 *There exists* $0 < Z = Z^T \in \mathbb{R}^{(n_x+n_c)\times(n_x+n_c)}$, $A_{ci} \in \mathbb{R}^{n_c\times n_c}$, $B_{ci} \in \mathbb{R}^{n_c\times n_y}$, *and* $C_{ci} \in \mathbb{R}^{n_u\times n_c}$ *for* $i \in \{1,2\}$ *such that Equation 5.12 holds if and only if there exists matrices* $Y = Y^T \in \mathbb{R}^{n_x\times n_x}$, $X = X^T \in \mathbb{R}^{n_x\times n_x}$, $L_i \in R^{n_x\times n_y}$, $F_i \in R^{n_u\times n_x}$, *and* $W_i \in R^{n_x\times n_x}$ *for* $i \in \{1,2\}$ *such that:*

$$\begin{bmatrix} R_{11} & R_{21}^T & R_{31}^T \\ R_{21} & R_{22} & 0 \\ R_{31} & 0 & R_{22} \end{bmatrix} > 0 \tag{5.13}$$

where the block matrices are defined as:

$$R_{11} := \begin{bmatrix} Y & I & 0 \\ I & X & 0 \\ 0 & 0 & \gamma^2 I \end{bmatrix}$$

$$R_{22} := \begin{bmatrix} Y & I & 0 \\ I & X & 0 \\ 0 & 0 & I \end{bmatrix}$$

$$R_{21} := \sqrt{p_1} \begin{bmatrix} YA_1+L_1(C_2)_1 & W_1 & Y(B_1)_1+L_1(D_{21})_1 \\ A_1 & A_1X+(B_2)_1F_1 & (B_1)_1 \\ (C_1)_1 & (C_1)_1X+(D_{12})_1F_1 & (D_{11})_1 \end{bmatrix}$$

$$R_{31} := \sqrt{p_2} \begin{bmatrix} YA_2+L_2(C_2)_2 & W_2 & Y(B_1)_2+L_2(D_{21})_2 \\ A_2 & A_2X+(B_2)_2F_2 & (B_1)_2 \\ (C_1)_2 & (C_1)_2X+(D_{12})_2F_2 & (D_{11})_2 \end{bmatrix}$$

The parenthesis are to distinguish $(B_1)_2$*, the second mode of* B_1*, and* $(B_2)_1$*, the first mode of* B_2*.*

Proof. ($\Rightarrow$) The proof uses a transformation motivated by the proof for the continuous time output feedback MJLS problem [24]. Assume Equation 5.12 holds and partition Z compatibly with A_{cl}:

$$Z = \begin{bmatrix} Z_1 & Z_2 \\ Z_2^T & Z_3 \end{bmatrix} \tag{5.14}$$

where $Z_1 \in \mathbb{R}^{n_x \times n_x}$, $Z_2 \in \mathbb{R}^{n_x \times n_c}$, $Z_3 \in \mathbb{R}^{n_c \times n_c}$. Without loss of generality, we assume $n_c \geq n_x$. If $n_c < n_x$, we can add stable, unobservable, uncontrollable modes to the controller without affecting closed loop performance. Under the assumption that $n_c \geq n_x$, if Z_2 is not full row rank, then we can find a full rank matrix arbitrarily close to Z_2 such that Equation 5.12 still holds. This follows since the set of full rank matrices of a given dimension is dense in the set of all matrices of the same dimension. Again without loss of generality, we assume that Z_2 is full row rank. Define the matrix $Y := (Z_1 - Z_2 Z_3^{-1} Z_2^T)^{-1}$ and note

that $Y > 0$ since $Z > 0$ (by Schur complements). Next, define the transformation:

$$T := \left[\begin{array}{ccc} Y & I_{n_x} & 0 \\ -Z_3^{-1}Z_2^T Y & 0 & 0 \\ \hline 0 & 0 & I_{n_w} \end{array}\right] \tag{5.15}$$

Since Z_2 is full row rank, T is full column rank. Positive definiteness is preserved under full column rank congruence transformations. That is, if $A > 0$ and B is full column rank, then $B^T AB > 0$. Thus we can multiply Equation 5.12 on the left by the congruence transformation $diag(T^T, T^T, T^T)$ and on the right by $diag(T, T, T)$ and the resulting matrix will still be positive definite. Letting T_{11} be the upper left block of T, we note the following block multiplications:

$$T_{11}^T Z T_{11} = \left[\begin{array}{cc} Y & I_{n_x} \\ I_{n_x} & Z_1 \end{array}\right]$$

$$T_{11}^T A_{cl,i} Z T_{11} = \left[\begin{array}{cc} YA_i + L_i(C_2)_i & W_i \\ A_i & A_i Z_1 + (B_2)_i F_i \end{array}\right]$$

$$C_{cl,i} Z T_{11} = \left[\begin{array}{cc} (C_1)_i & (C_1)_i Z_1 + (D_{12})_i F_i \end{array}\right]$$

$$T_{11}^T B_{cl,i} = \left[\begin{array}{c} Y(B_1)_i + L_i(D_{21})_i \\ (B_1)_i \end{array}\right]$$

where:

$$Y := (Z_1 - Z_2 Z_3^{-1} Z_2^T)^{-1}$$

$$X := Z_1$$

$$F_i := C_{ci} Z_2^T$$

$$L_i := -Y Z_2 Z_3^{-1} B_{ci}$$

$$W_i := Y A_i Z_1 + Y (B_2)_i F_i + L_i (C_2)_i Z_1 - Y Z_2 Z_3^{-1} A_{ci} Z_2^T$$

The congruence transformation shows that Equation 5.13 is satisfied with the $Y = Y^T$, $X = X^T$, L_i, F_i, and W_i for $i \in \{1, 2\}$ defined above. Furthermore, the dimensions of Z_1, Z_2, and Z_3 imply that $Y = Y^T$, $X = X^T$, L_i, F_i, and W_i for $i \in \{1, 2\}$ defined above have the dimensions given in the Theorem.

($\Leftarrow$) Assume that we have found $Y = Y^T$, $X = X^T$, L_i, F_i, and W_i for $i = 0, 1$ of the dimensions listed in the Theorem such that Equation 5.13 holds. Define the transformation:

$$\tilde{T} = \left[\begin{array}{cc|c} 0 & Y^{-1} & 0 \\ I_{n_x} & -I_{n_x} & 0 \\ \hline 0 & 0 & I_{n_w} \end{array}\right] \tag{5.16}$$

Multiply Equation 5.12 on the left by the congruence transformation $diag(\tilde{T}^T, \tilde{T}^T, \tilde{T}^T)$ and on the right by $diag(\tilde{T}, \tilde{T}, \tilde{T})$. After the appropriate matrix multiplications, we see that

Equation 5.12 is satisfied with the following scaling and controller matrices:

$$Z = \begin{bmatrix} X & Y^{-1} - X \\ Y^{-1} - X & X - Y^{-1} \end{bmatrix} \tag{5.17}$$

$$B_{ci} = Y^{-1} L_i$$

$$C_{ci} = F_i \left(Y^{-1} - X\right)^{-1}$$

$$A_{ci} = -Y^{-1} \left[Y A_i X + Y (B_2)_i F_i + L_i (C_2)_i X - W_i\right] \left(Y^{-1} - X\right)^{-1}$$

Also note that Condition 5.13 implies that Y and X are positive definite, hence we can apply the Schur complement lemma to show $Z > 0$. This reconstructed controller has state dimension $n_c = n_x$, i.e. $A_{ci} \in \mathbb{R}^{n_x \times n_x}$. ■

Before proceeding, we make several comments about this theorem. The constraint is a linear matrix inequality involving the scalar γ and matrices Y, X, L_i, F_i, W_i. The H_∞ optimal controller can be obtained by solving the semi-definite programming problem:

$$\min \gamma$$

subject to Equation 5.13

As noted previously, this problem can be efficiently solved by interior-point methods using freely available software [9, 32]. This problem is conceptually solved in two stages. A feasibility stage solves for matrices Y, X, L_i, F_i, W_i and the scalar γ (not necessarily the minimum) which satisfies the constraint, Equation 5.13. The optimization stage then searches for the minimal value of γ along with the matrices which satisfy the constraint for the this optimal value. If the optimization fails at the feasibility stage, then by the discussion in Section 5.5, the system is not SMS. If there is a feasible solution, the optimization will find

the minimal value of γ. The proof then gives a procedure for constructing the H_∞ optimal controller. A key point here is that the procedure generates a controller with dimension $n_c \leq n_x$. Thus, if there exists a controller (of any state dimension) which gives closed loop H_∞ norm less than γ, then there exists a controller with state dimension $n_c \leq n_x$ which achieves the same performance. In words, the algorithm gives the optimal controller of the form given in Equation 5.10. Finally, note that the proof given above can be easily extended to systems with more than 2 modes.

5.7 Numerical Examples

In this section, we show how the tools in this chapter can be used for analysis of and controller synthesis for networked control systems. The first example is a simple second order example. This example is not physically motivated, but it displays the structure of the optimal H_∞ controller and gives some comparison to the LQG results obtained in the previous chapter. Next, we synthesize an optimal controller for the vehicle following problem presented in the previous chapter. In the next chapter, we use the H_∞ matrix condition (Equation 5.6) to analyze the effect of communication delays on the formation flight performance.

5.7.1 Controller Synthesis for a Second Order System

We will use a randomly generated second order model in the form of Equation 5.9 to study the LMI condition of the previous section:

$$\begin{bmatrix} x(k+1) \\ z(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ \theta(k)C_2 & \theta(k)D_{21} & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ w(k) \\ u(k) \end{bmatrix}$$

where $x(k) \in \mathbb{R}^2$, $u(k) \in \mathbb{R}$, $y(k) \in \mathbb{R}$, $z(k) = [x^T(k)\ \epsilon_u u(k)]^T \in \mathbb{R}^3$ and $w(k) \in \mathbb{R}^3$. ϵ_u is a scalar used to weight the penalty we are putting on the control input. The state matrices are given by:

$$A = \begin{bmatrix} -0.2656 & -2.2023 \\ -1.1878 & 0.9863 \end{bmatrix} \qquad B_1 = \begin{bmatrix} \epsilon_d & 0 & 0 \\ 0 & \epsilon_d & 0 \end{bmatrix} \qquad B_2 = \begin{bmatrix} -0.5186 \\ 0.3274 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad D_{11} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad D_{12} = \begin{bmatrix} 0 \\ 0 \\ \epsilon_u \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 0.2341 & 0.0215 \end{bmatrix} \qquad D_{21} = \begin{bmatrix} 0 & 0 & \epsilon_n \end{bmatrix}$$

The error vector, $z(k)$, has entries to penalize each state and the control effort. The disturbance vector, $w(k)$, includes plant disturbances which inject into each state equation and a disturbance which injects into the measurement to model sensor noise. The scalars ϵ_d and ϵ_n can be used to weight the size of these disturbances. For this simple problem, we use $\epsilon_d = \epsilon_n = .1$ and $\epsilon_u = 1$. Finally, we note that the eigenvalues of this system are -1.3739 and 2.0946. Thus feedback is necessary to stabilize the system and we expect that a very high packet loss rate will make this impossible. In fact, we know from the previous chapter

(Theorem 4.3) that no second-moment stable estimator exists if $p \geq 1/\rho(A)^2 \approx 0.228$. The separation theorem (Theorem 4.3) that we applied for the vehicle following problem also holds for this control problem. Thus if $p \geq 0.228$, then no controller exists which stabilizes the closed loop system.

The LMI condition given in Theorem 5.4 is used to find the H_∞ optimal controller for each packet loss rate. Figure 5.3 shows the performance of the optimal H_∞ controller as a function of the packet loss rate, p. The H_∞ gain has been normalized to give a value of 1 for the nominal case of no packet losses, $p = 0$. For packet loss rates above $p^* = 0.121$, the LMI condition is infeasible. Thus we conclude that no stabilizing controller of the form given in Equation 5.10 exists above this critical packet loss rate. However, we cannot conclude that a stabilizing controller does not exist. A time-varying and/or nonlinear controller may exist to stabilize the system for $p \in (0.121, 0.228)$. Recall that the design procedure in Section 5.6 restricts the class of controllers and does not use the knowledge of $\theta(k)$ to the fullest extent possible. If we conjecture that $p < 0.228$ is a necessary and sufficient condition for stabilizability, then $(0.121, 0.228)$ represents the conservativeness in this control design procedure.

Both conditions, $p^* < 0.228$ or $p^* < 0.121$, give a hard constraint which the network must satisfy for the associated design procedure to return a stabilizing controller. A more practical constraint is given by the optimal performance vs. packet loss rate curve. Figure 5.3 shows that we can obtain reasonable performance, relative to the nominal case of $p = 0$, for packet loss rates under 0.1. However, the performance curve has a sharp knee around $p = .11$ which means that, although it may be possible to find a stabilizing

controller, the performance will be quite poor. For the LQG controller, we have yet to develop a tractable method to compute the optimal cost.

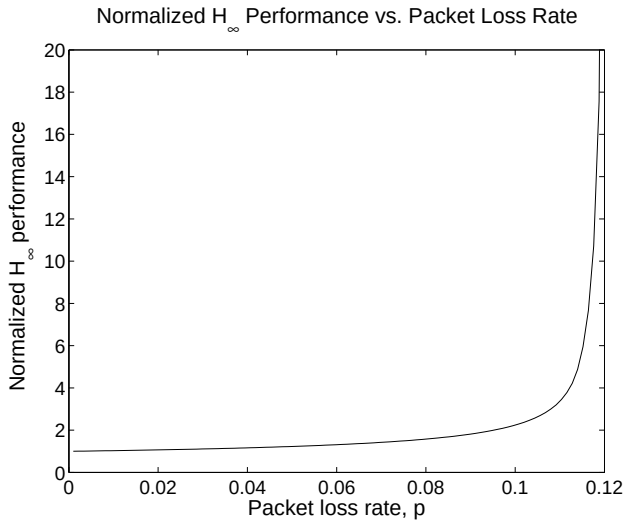

Figure 5.3: Normalized H_∞ Performance vs. Packet Loss Rate

We now investigate the structure of the controller produced by the H_∞ optimization. For $p = .12$, the following controller was found:

$$x_c(k+1) = A_{c,\theta(k)}x_c(k) + B_{c,\theta(k)}y(k)$$

$$u(k) = C_{c,\theta(k)}x_c(k) \tag{5.18}$$

where the controller matrices are:

$$A_{c0} = \begin{bmatrix} -0.0143 & -0.2359 \\ -1.3479 & -0.2549 \end{bmatrix} \quad B_{c0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad C_{c0} = \begin{bmatrix} 0.4815 & 3.7916 \end{bmatrix}$$

$$A_{c1} = \begin{bmatrix} -1.2813 & -0.3529 \\ 0.6178 & -0.0749 \end{bmatrix} \quad B_{c1} = \begin{bmatrix} -5.4482 \\ 8.3861 \end{bmatrix} \quad C_{c1} = \begin{bmatrix} 0.4814 & 3.7916 \end{bmatrix}$$

Controller 0 is used when the packet is dropped and controller 1 is used when the packet is received. It can be verified that the controller has an observer based structure where

B_{ci} are the observer gains and C_{ci} are the feedback gains: $A_{c0} \approx A - B_2C_{c0}$ and $A_{c1} \approx A + B_{c1}C_2 - B_2C_{c1}$. In other words, the controller state, $x_c(k)$, is an estimate of $-x(k)$. We note that $B_{c0} = [0\ 0]^T$ and B_{c1} is an observer gain making $A + B_{c1}C_2$ stable. This controller structure is very intuitive. When a packet is received the estimates of the original plant states are updated with the plant A and corrected with the observer gain B_{c1}. When a packet is dropped, the observer gain B_{c1} is set to zero and the estimates are updated using our knowledge of A, which is the best we can do when no new sensor information arrives. Finally, $C_{c0} \approx C_{c1}$. In words, once our best estimate of the state is obtained, there is no advantage to varying the feedback gain based on the loss or arrival of sensor information. The optimization has arrived at a controller that has a separation property.

We also remark that the observer gains are adjusted as the packet loss rate changes. For example, the observer gain for $p = 0.01$ is given by: $B_{c1} = \begin{bmatrix} -4.7403 & 7.4543 \end{bmatrix}^T$. This observer gain has decreased when compared to the gains for $p = 0.12$. The interpretation is that for higher packet loss rates, the observer must weight measurements more heavily. This makes sense because the time until another correct packet is received may be large if the packet loss rate is high.

5.7.2 Controller Synthesis for Vehicle Following

In this section, we synthesize a controller for the vehicle following problem described in the previous chapter. Figure 5.4 depicts the problem and notation for two cars. Let x_1 and x_2 denote the longitudinal positions of a leader and a follower vehicle, respectively. The goal is to have vehicle 2 follow a distance δ behind vehicle 1. In other words, the controller should regulate the spacing error, $e := x_1 - x_2 - \delta$, to zero. We will use the LMI

condition derived in Section 5.6 to design a controller assuming the vehicles are governed by linear dynamics for $i = 1, 2$:

$$\ddot{x}_i = a_i$$

$$\tau \dot{a}_i + a_i = u_i$$

where a_i is the acceleration of the vehicle and u_i is the desired acceleration. The first order dynamics between u_i and a_i are due to the throttle/brake actuator dynamics. The time constant for these dynamics is $\tau = 0.1$ seconds. We discretized these dynamics by sampling the output every T seconds and applying a first-order hold at the input. The notation for discrete time vehicle model is given by:

$$\bar{x}_i(k+1) = A_{d,i}\bar{x}_i(k) + B_{d,i}u_i(k) \qquad \text{for } i = 1, 2$$

where $\bar{x}_i(k) \in \mathbb{R}^3$ is the state of vehicle i.

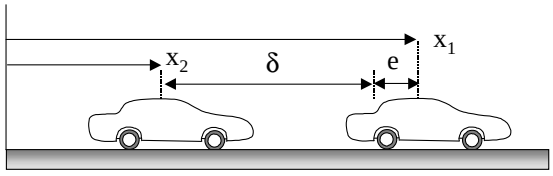

Figure 5.4: Vehicle Following Problem

We will assume that both vehicles can measure their full state: position, velocity and acceleration. Typically a radar is used to measure the vehicle spacing, but we will assume that every T seconds vehicle 1 communicates its state information across a wireless link. In this scenario, we desire a controller which uses the communicated state information to regulate the error to zero. The vehicle following problem can be placed in the form of

Equation 5.9:

$$\begin{bmatrix} x(k+1) \\ z(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_{2,\theta(k)} & D_{21,\theta(k)} & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ w(k) \\ u_2(k) \end{bmatrix} \tag{5.19}$$

where $x(k) := [\bar{x}_1(k)^T \ \bar{x}_2(k)^T]^T \in \mathbb{R}^6$ and $y(k) := [\bar{x}_{1c}(k)^T \ \bar{x}_2(k)^T]^T \in \mathbb{R}^6$. $\bar{x}_{1c}(k)$ is the state information communicated from vehicle 1 to the controller on-board vehicle 2. We again use the Bernoulli communication model (Equation 4.1):

$$\bar{x}_{1c}(k) = \theta(k)\bar{x}_1(k) \tag{5.20}$$

where $\theta(k)$ is a Bernoulli process given by $Pr[\theta(k) = 0] = p$ and $Pr[\theta(k) = 1] = 1 - p$. The error vector, $z(k)$, penalizes the spacing error and the weighted control effort: $z(k) := [\bar{x}_1(k)^T - \bar{x_2}(k)^T \ \epsilon_u u_2(k)] \in \mathbb{R}^2$. The weighting on the control effort is given by $\epsilon_u = 1.0$. The disturbance vector is given by: $w(k) := [u_1(k) \ d_2(k) \ n_1^T(k) \ n_2^T(k)]^T \in R^8$. This vector includes the lead vehicle acceleration ($u_1(k) \in \mathbb{R}$), a plant disturbance on vehicle 2 ($d_2(k) \in \mathbb{R}$), and sensor noises for all measurements ($n_1(k) \in \mathbb{R}^3$ and $n_2(k) \in \mathbb{R}^3$). The vehicle 2 plant disturbance is scaled by $\epsilon_d = 0.1$ and all the sensor noises are scaled by $\epsilon_n = 0.1$. Given all these definitions, the generalized plant matrices are given by:

$$A := \begin{bmatrix} A_{d,1} & 0 \\ 0 & A_{d,2} \end{bmatrix} \qquad B_1 := \begin{bmatrix} B_{d,1} & 0 & 0 \\ 0 & \epsilon_d B_{d,2} & 0 \end{bmatrix} \qquad B_2 := \begin{bmatrix} 0 \\ B_{d,2} \end{bmatrix}$$

$$C_1 := \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad D_{11} := 0 \qquad D_{12} := \begin{bmatrix} 0 \\ \epsilon_u \end{bmatrix}$$

$$C_{2,\theta(k)} = \begin{bmatrix} \theta(k)I & 0 \\ 0 & I \end{bmatrix} \qquad D_{21,\theta(k)} = \begin{bmatrix} 0 & \epsilon_n\theta(k)I & 0 \\ 0 & 0 & \epsilon_n I \end{bmatrix}$$

Note that no controller can be make this plant SMS because vehicle 1 is unstable (it has two poles at $z = 1$) and uncontrollable from $u_2(k)$. To solve this problem, we use a technique that is common to handle unstable weights in H_∞ design problems [56]. The model for vehicle 1 is altered slightly by moving the unstable poles just inside the unit disk.

Figure 5.5 shows the performance of the H_∞ optimal controller as a function of the packet loss rate, p. The performance has been normalized by the nominal performance at $p = 0$, so this curve shows the performance degradation caused by the packet losses. The model for vehicle 1 is marginally stable (although we approximate it with a stable plant). By the results in the previous chapter, it is possible to find a stabilizing controller for any $p \in [0, 1)$. Even though stabilizing controllers always exist, Figure 5.5 shows that performance rapidly degrades for packet loss rates above $p = 0.9$ when $T = 0.02$ seconds. It may seem surprising that we can obtain good performance relative to the nominal performance for packet loss rates up to 90%. Since we are assuming communication every T=20 milliseconds, a packet loss rate of 90% still implies that new information will arrive, on average, every 200 milliseconds. This update rate is still quite fast relative to the overall dynamics of the vehicle motion. Figure 5.5 also shows that the knee in the performance curve shifts left as the sample time increases. This shows that if we send information less often, then we must have a higher probability of receiving a correct packet in order to maintain good performance. As discussed in Section 4.5, the sample rate is tied to the required network bandwidth. Plotting these curves for several sample rates shows the trade-off between packet loss rate and network bandwidth.

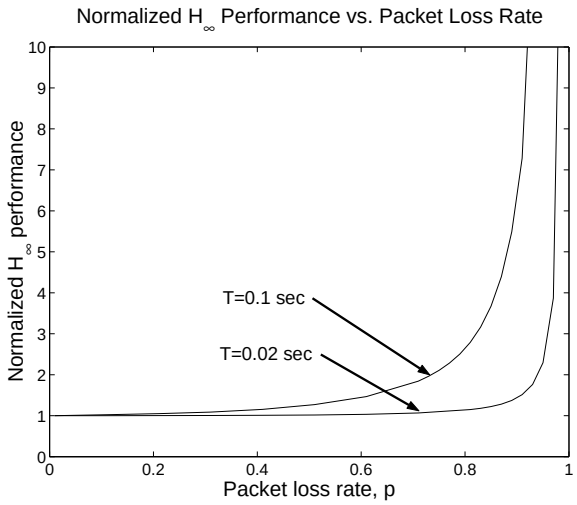

Figure 5.5: Normalized H_∞ performance vs. packet loss rate: Curve shows car following performance for the optimal controller

Chapter 6

Formation Flight of Unmanned Aerial Vehicles

6.1 Introduction

As mentioned in Chapter 3, cooperative control of unmanned vehicles has received significant interest due to a variety of applications. At the University of California, Berkeley, the goal of the BEAR project is to use autonomous vehicles to perform complex missions. Modeling of Unmanned Aerial Vehicles (UAVs), hover control, and wavepoint navigation have been completed [86, 87]. More complex tasks include using UAVs in pursuit-evasion games [46] and implementing a vision-based landing system [83]. The BEAR project has focused on rotorcraft-based platforms for their UAV research. The advantages of small-scale helicopters for UAV research include their agility and their vertical take off and landing capability. Moreover, the ability to hover is useful in many applications. Along with these

advantages come several characteristics that make small-scale helicopters difficult to model and control. Specifically, the dynamics are nonlinear, unstable, and subject to complex aerodynamic effects. These aspects make the benefits of small-scale helicopters difficult to obtain.

In the remainder of this chapter, we will describe a linear model of a small-scale helicopter obtained using time-domain system identification methods. This model is valid for hovering and low-velocity maneuvers. We will also present a controller, designed by D. Shim [87], that is used for wavepoint navigation of a single UAV. In Section 6.3, this controller is used in the design of a formation flight controller. The formation flight controller is constructed using the procedure detailed in Section 3.5 and results in a mesh stable design. This formation controller requires the UAVs to communicate information across a wireless network. In Section 6.4, we analyze the effect of communication packet losses using the results obtained in Chapter 5. In the final section, we present some conclusions of this work.

6.2 Modeling and Control of a Small-Scale Helicopter

In this section, we review some results for modeling and control of small-scale helicopters. In particular, we describe the work done by D. Shim as part of the BEAR project [87]. The BEAR fleet consists of several helicopters of various sizes, but we will focus on the modeling and control of a Yamaha R-50 (shown in Figure 6.1). More details on the results contained in this section can be found in [87] and the references therein.

As noted in [60], scaling down a helicopter increases its bandwidth and makes

Figure 6.1: Yamaha R-50 Helicopter

it harder to control. Thus stabilizer bars are common in small-scale helicopters to add damping to the pitch and roll dynamics. This stabilizer bar is a unique feature of small-scale helicopters and prevents the use of full size helicopter models. Mettler, et.al. have developed a parametric model for a Yamaha R-50 helicopter that includes the effect of this stabilizer bar [60]. The authors then used a frequency domain identification technique to fit the model parameters. Shim, et.al. used the model proposed in [60], but applied a time-domain system identification technique to find the best parameter values for a Kyosho small-scale helicopter [86] and a Yamaha R-50 [87]. These linear models are valid for hovering and low velocity maneuvers. The model of the Yamaha R-50 is given by:

$$\dot{x}_1 = A_1 x_1 + B_1 \bar{u} \tag{6.1}$$

where the state vector, x_1, and control input, $\bar{u}$, are given by:

$$x_1 := [u\ v\ p\ q\ \phi\ \theta\ a\ b\ w\ r\ r_{fb}]^T$$

$$\bar{u} := [\delta_{lat}\ \delta_{lon}\ \delta_{col}\ \delta_{ped}]^T$$

u, v, and w are the longitudinal, lateral, and vertical speeds (in feet/sec) in the helicopter

coordinate frame. p, q, and r are the roll, pitch, and yaw rates (in rads/sec) in the helicopter frame. ϕ and θ are the roll and pitch of the helicopter. a and b are the longitudinal and lateral rotor flapping angles. r_{fb} is a yaw rate feedback term in the dynamics. δ_{lat} and δ_{lon} correspond to the cyclic lateral and longitudinal control inputs, respectively. These two inputs basically control the lateral and longitudinal motions. δ_{col} and δ_{ped} are the collective and directional inputs. These inputs basically control the vertical motion and the helicopter heading.

Let $g := 32.2\ ft/sec^2$ be the gravitational constant. The state matrices are given by:

$$A_1 = \begin{bmatrix} -0.126 & 0 & 0 & 0 & 0 & -g & -g & 0 & 0 & 0 & 0 \\ 0 & -0.425 & 0 & 0 & g & 0 & 0 & g & 0 & 0 & 0 \\ -0.168 & 0.087 & 0 & 0 & 0 & 0 & 36.705 & 161.109 & 0 & 0 & 0 \\ -0.082 & -0.052 & 0 & 0 & 0 & 0 & 63.576 & -19.493 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -3.444 & 0.829 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0.361 & -3.444 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -38.995 & 9.640 & -0.760 & 8.423 & 0 \\ 0 & 0 & -1.33 & 0 & 0 & 0 & 0 & 0 & 0.057 & -5.511 & -44.873 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.816 & -11.021 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -0.842 & 2.823 & 0 & 0 \\ -2.409 & -0.351 & 0 & 0 \\ 0 & 0 & 70.504 & 0 \\ 0 & 0 & 23.626 & 44.873 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Note that the velocities and angular rates are given in the helicopter frame. To obtain positions in an earth-fixed frame, a nonlinear coordinate rotation must be used. For the purposes of this chapter, we assume that the UAV is sufficiently close to hovering conditions that this rotation can be neglected. Since the model obtained above is only valid for low speed maneuvers, this is not a severe limitation beyond the current assumptions. However, this assumption must clearly be relaxed before testing formation flight at significant cruising speeds. At any rate, given this assumption, the positions (x, y, z) and heading (ψ) in an

earth-fixed frame can be obtained by integrating the velocities (u, v, w) and yaw rate (r). Augmenting these integrators on to the state equations given in Equation 6.1 gives:

$$\dot{\bar{x}} = A\bar{x} + B\bar{u} \tag{6.2}$$

where $\bar{x} := [x_1^T \; x \; y \; z \; \psi]^T$ and $\bar{u}$ is defined above. The augmented state matrices are defined as:

$$A := \left[\begin{array}{c|c} A_1 & 0_{11\times 4} \\ \hline \begin{smallmatrix} 1&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0 \end{smallmatrix} & 0_{4\times 4} \end{array}\right] \qquad B := \left[\begin{array}{c} B \\ \hline 0_{4\times 4} \end{array}\right]$$

Next we discuss a controller proposed and experimentally tested by by D. Shim [87]. First, note that the helicopter has six degrees of freedom: three positions and three Euler angles. Since the helicopter has only four control inputs, it is an underactuated system. The four inputs will be used to control the three positions and the heading of the UAV. Physically, the helicopter moves longitudinally by pitching and laterally by rolling. Thus, the remaining two degrees of freedom, roll and pitch, are constrained by the helicopter motion. The helicopter dynamics can be roughly broken down into four subsystems: roll/lateral, pitch/longitudinal, yaw, and heave (vertical) dynamics. There is significant coupling between the channels, but Shim stabilizes each subsystem independently with a proportional-derivative (PD) controller. The justification is that this control structure is simple, intuitive, and has been effectively tested in experimental flights. The control law

designed and tested by Shim [87] is given by:

$$\begin{aligned}
\delta_{lat} &= -K_\phi \phi + K_v(v_d - v) + K_y(y_d - y) \\
\delta_{lon} &= -K_\theta \theta + K_u(u_d - u) + K_x(x_d - x) \\
\delta_{col} &= K_w(w_d - w) + K_z(z_d - z) \\
\delta_{ped} &= K_\psi(\psi_d - \psi)
\end{aligned} \tag{6.3}$$

The reference trajectories are denoted by the subscript 'd'. The gains used in the controller are given by: $K_\phi = -0.55$, $K_v = -0.02$, $K_y = -0.01$, $K_\theta = 0.55$, $K_u = -0.02$, $K_x = -0.01$, $K_w = 0.035$, $K_z = 0.12$, and $K_\psi = 1$.

A block diagram of the feedback system is shown in Figure 6.2. $C_a \in \mathbb{R}^{2\times 15}$ is defined so that $\left[\begin{smallmatrix}\phi\\ \theta\end{smallmatrix}\right] = C_a \bar{x}$. The inner loop stabilizes the attitude dynamics of the helicopter and the feedback gain is given by $K_a := \left[\begin{smallmatrix} K_\phi & 0 \\ 0 & K_\theta \\ 0 & 0 \\ 0 & 0 \end{smallmatrix}\right]$. We define the plant with this stabilizing inner loop as $H_{stab}(s) := C\left(sI - (A - BK_aC_a)\right)^{-1}B$. The outer loop attempts to track a desired reference trajectory. As noted above, we can only use the inputs to control four of the helicopter's six degrees of freedom. The output matrix, C, is defined to return the states that we will control: $[y\ x\ z\ \psi]^T$. $K(s)$ is a 4×4 transfer function matrix with PD controllers on the diagonal. Even though the PD controllers are not proper, they are realizable because we have access to the velocity measurements. Thus we can implement the PD controllers by feeding back the velocities and yaw rate. With this in mind, $K(s)$ is given by:

$$K(s) := \begin{bmatrix} K_v s + K_y & 0 & 0 & 0 \\ 0 & K_u s + K_x & 0 & 0 \\ 0 & 0 & K_w s + K_z & 0 \\ 0 & 0 & 0 & K_\psi \end{bmatrix} \tag{6.4}$$

Experimental test results of a Yamaha R-50 performing a low speed maneuver using this control law are shown in [87]. A simulation of a low speed maneuver is shown

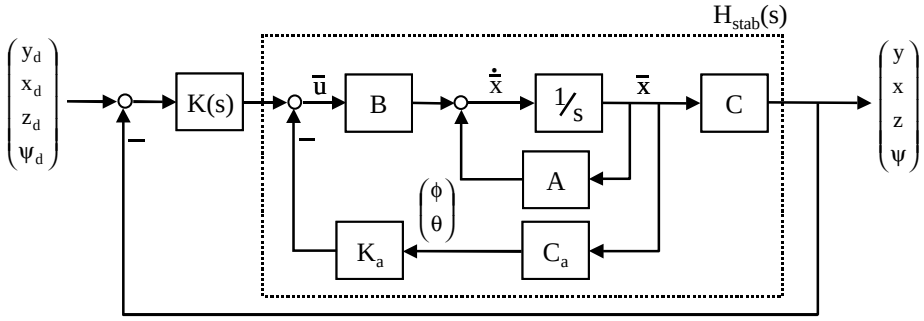

Figure 6.2: Feedback Block Diagram for UAV and Controller

in Figure 6.3. The reference trajectory, shown in the right subplot, starts from rest and accelerates up to a cruising speed of $3\frac{ft}{sec}$ in the x-direction. The desired heading is kept constant, $\psi_d \equiv 0$, as are y_d and z_d. As discussed above, the controller, $K(s)$, can be implemented as a static gain when the velocities and yaw rate are fed back. In this simulation, the plant is sampled every 100msec and the output of this static gain controller is zero-order held. The right subplot of Figure 6.3 shows the tracking response, $x(t)$, with this discretized implementation of the PD controller. The left subplot shows the cyclic longitudinal control effort, δ_{lon}. All other control inputs and state variables remain small and are not shown. In the next section, we use this controller to design a formation flight control law.

6.3 Formation Controller

In this section we design a control law to coordinate a mesh of UAVs. The mesh consists of 9 UAVs as depicted in Figure 6.4. Let $p_{i,j}(t) \in \mathbb{R}^4$ be the output of the ij^{th} vehicle: $P_{i,j}(s) := H_{stab}(s)U_{i,j}(s)$. As shown in Figure 6.2, $p_{i,j}(t) \in \mathbb{R}^4$ contains the three global positions and the heading of the ij^{th} vehicle. We use the spacing errors defined in

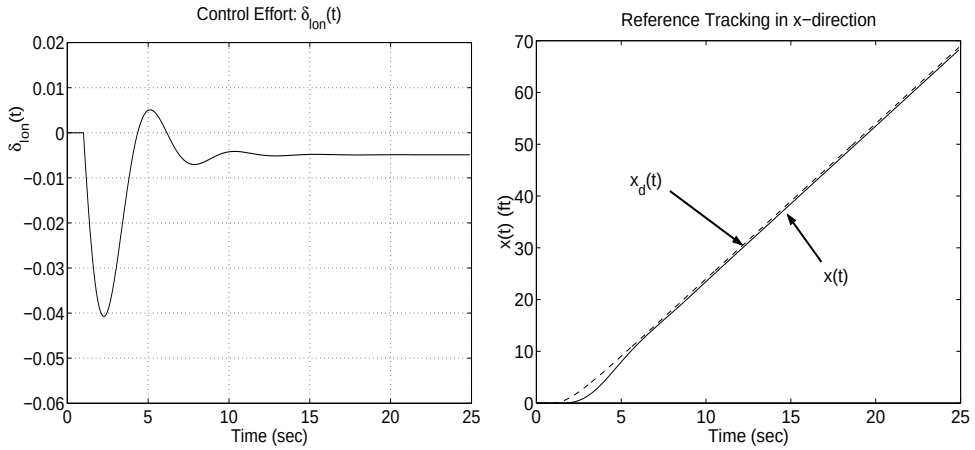

Figure 6.3: Time domain plots of reference tracking: *Left:* Control effort: Cyclic longitudinal input ($\delta_{lon}(t)$). *Right:* Reference ($x_d(t)$) and UAV ($x(t)$) trajectories in the x-direction.

Equation 3.1:

$$\epsilon^R_{i,j}(t) = p_{i,j-1}(t) - p_{i,j}(t) - \delta^R_{i,j}(t) \qquad \text{for } 1 \leq i \leq N,\ 1 < j \leq N$$

$$\epsilon^C_{i,j}(t) = p_{i-1,j}(t) - p_{i,j}(t) - \delta^C_{i,j}(t) \qquad \text{for } 1 < i \leq N,\ 1 \leq j \leq N$$

We also use the averaged spacing error, $e_{i,j} = (\epsilon^R_{i,j} + \epsilon^C_{i,j})/2$, defined in Chapter 3. As shown in Figure 6.4, the desired spacing vectors are given by: $\delta^R := \left[\, -10ft\ 0\ 0\ 0 \,\right]^T$ and $\delta^C := \left[\, 0\ 10ft\ 0\ 0 \,\right]^T$. The control law presented in the previous section (Equation 6.4), can be used as a formation controller:

$$U_{i,j}(s) = K(s)E_{i,j}(s) \tag{6.5}$$

From the theory developed in Chapter 3, we expect that this control law, which uses only preceding vehicle information, will perform poorly. Recall that the output complementary sensitivity function is defined as $T_o(s) := [I + H_{stab}(s)K(s)]^{-1}H(s)_{stab}K(s)$. Since $H_{stab}(s)$ is of the form $H_{stab}(s) = \frac{1}{s}\tilde{H}_{stab}(s)$, closed loop stability implies $T_o(0) = I$. Therefore, low frequency errors are not attenuated as they propagate through the mesh. Note

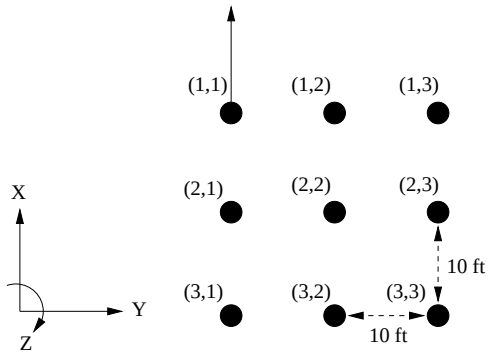

Figure 6.4: Formation of 9 UAVs

that the UAV has internal force feedbacks due to aerodynamic effects and hence $H_{stab}(s)$ does not have two free integrators. Thus Theorem 3.1 technically does not apply here. However, Figure 6.5 shows that error amplification will still occur. This figure shows plots of $\rho[T_o(j\omega)]$ and $\bar{\sigma}\left(T_o(j\omega)\right)$. $\bar{\sigma}\left(T_o(j\omega)\right) = 2.27$ at $\omega_0 = 2.44\frac{rads}{sec}$. This is the peak amplification from one vehicle to its neighbor. $\rho[T_o(j\omega)] = 1.77$ at $\omega_0 = 1.15\frac{rads}{sec}$. This is the peak geometric amplification in a large mesh. Moreover, the eigenvector that achieves the spectral radius is $[-0.08 + 0.32i;\ 0.94;\ 0.01 - 0.01i;\ 0]^T$. This eigenvector is almost aligned with the x-direction. In other words, the low frequency content of a desired trajectory of the following form: $(y_d(t), x_d(t), z_d(t), \psi_d(t)) = (const, x_d(t), const, const)$ will be amplified geometrically as it propagates through the mesh.

Figure 6.6 confirms our expectations. A mesh of nine UAVs was simulated with the lead vehicle tracking the reference depicted in Figure 6.3. This reference trajectory attempts to bring the mesh up to a cruising speed of $3\frac{ft}{sec}$ in the x-direction. The lead vehicle follows this reference trajectory using the control law in Equation 6.4. Each follower uses the control law in Equation 6.5 to maintain their position in the formation. All controllers use

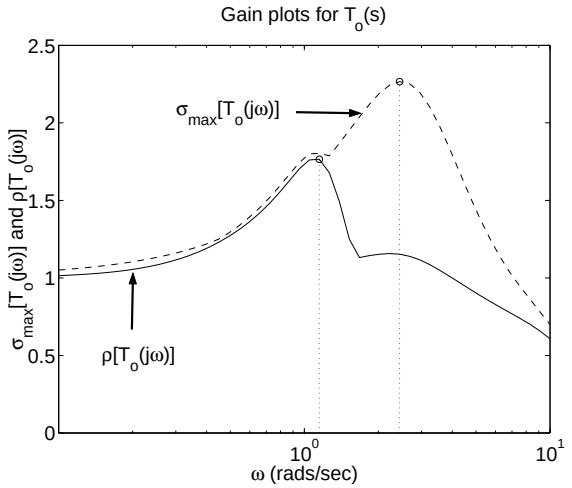

Figure 6.5: Plots of $\rho[T_o(j\omega)]$ and $\bar{\sigma}\left(T_o(j\omega)\right)$. Peaks are $\rho[T_o(j\omega)] = 1.77$ and $\bar{\sigma}\left(T_o(j\omega)\right) = 2.27$. achieved at $\omega_0 = 1.15\frac{rads}{sec}$ and $2.44\frac{rads}{sec}$, respectively. The eigenvector that achieves the spectral radius is $[-0.08 + 0.32i;\ 0.94;\ 0.01 - 0.01i;\ 0]^T$.

a sample time of 100msec. The right subplot of Figure 6.6 shows the spacing errors in the x-direction. These errors are larger for vehicles that are farther from the leader. The right subplot shows the cyclic longitudinal control input ($\delta_{lon}(t)$) for the mesh. Similarly the control effort is amplified for vehicles far from the leader.

To remove this amplification, we use the procedure described in Section 3.5. Figure 6.5 shows that $\|T_o(s)\|_\infty = 2.27$. We choose $\lambda := 0.4 < \frac{1}{2.27}$ and define a control law that depends on lead and preceding vehicle information:

$$U_{i,j}(s) = \underbrace{K_p(s)}_{:=\lambda K(s)} E_{i,j}(s) + \underbrace{K_l(s)}_{:=(1-\lambda)K(s)} \left(P_{1,1}(s) - P_{i,j}(s) - \frac{(i-1)\delta^C + (j-1)\delta^R}{s} \right) \quad (6.6)$$

Recall that $T_{lp}(s) := \left[I + H_{stab}(s)\left(K_p(s) + K_l(s)\right)\right]^{-1} H_{stab}(s)K_p(s)$. By construction, $T_{lp}(s) = \lambda T_o(s)$ and hence inequality (a) holds in the following relation:

$$\rho[T_{lp}(j\omega)] \leq \bar{\sigma}\left(T_{lp}(j\omega)\right) \overset{(a)}{\leq} \lambda \|T_o(s)\|_\infty \overset{(b)}{<} 1$$

Inequality (b) follows by our choice of λ. Using the control law in Equation 6.6, we can

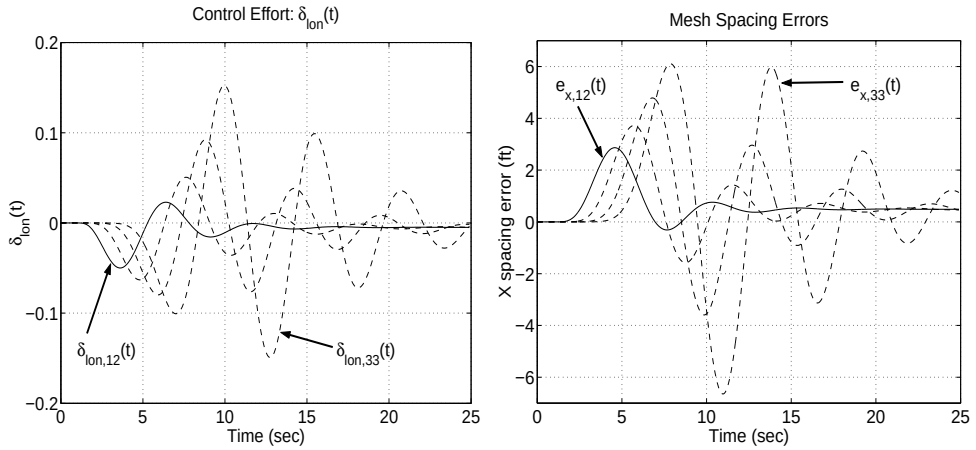

Figure 6.6: Time domain plots of predecessor following control law: *Left:* Cyclic longitudinal control input ($\delta_{lon}(t)$) for (1,2), (1,3), (2,3), and (3,3) vehicles. *Right:* (1,2), (1,3), (2,3), and (3,3) spacing errors in the x-direction

ensure that errors and control effort are attenuated as they propagate through the mesh. Figure 6.7 shows the simulation of the mesh when all followers use the control law in Equation 6.6. This figure shows that control effort and spacing errors are attenuated as we expect. Note that the axes are the same scale in Figures 6.6 and 6.7.

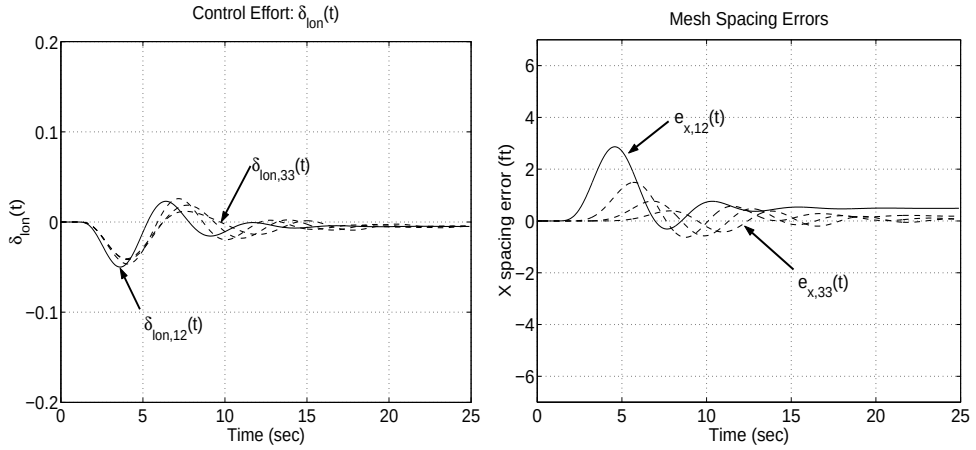

Figure 6.7: Time domain plots of predecessor following control law: *Left:* Cyclic longitudinal control input ($\delta_{lon}(t)$) for (1,2), (1,3), (2,3), and (3,3) vehicles. *Right:* (1,2), (1,3), (2,3), and (3,3) spacing errors in the x-direction

We could further decrease λ to enhance error attenuation. In the limit, $\lambda = 0$

decouples the vehicles so that each follow the leader independent of their neighbors. As explained in Chapter 2, this strategy is unsafe in a tight formation. Thus we should choose λ as large as possible to increase the coupling to the neighbors while still ensuring the error damping property (mesh stability) holds.

6.4 Analysis of Communication Packet Losses

The formation controller designed in the previous section (Equation 6.6) requires the lead UAV to communicate information to the followers. This information is transmitted across a wireless network and will be subject to communication packet losses. In this section, we analyze the effect of these packet losses using the results obtained in Chapter 5. Specifically, we will measure performance using the H_∞ norm and apply Theorem 5.3 to find the level of performance for a given packet loss rate.

Figure 6.8 shows the generalized plant for the formation of 9 UAVs. The only disturbance we consider is a velocity command, w, for the lead vehicle. Let $F_c(s) := \frac{1}{s+0.01}$ be an approximate integrator. We discretize $F_c(s)$ at a sample rate of $T_s = 100msec$ to obtain an approximate discrete-time integrator. $F(z) \in \mathbb{C}^{4\times 4}$ is a diagonal system with 4 copies of this approximate discrete time integrator on the diagonal. $w \in \ell_2^4$ is a velocity command for the lead vehicle and $r = F(z)w$ is the corresponding reference trajectory. The lead vehicle (i.e. the (1,1) vehicle) in the mesh uses the control law in Equation 6.4 to follow this reference trajectory. The remaining vehicles in the formation use the formation controller, Equation 6.6, to maintain their position in the mesh. The overbar notation in the figure denotes a stacking of all signals in the mesh: $\bar{u} := [u_{1,1}, \ldots, u_{3,3}]$

and $\bar{e} := [e_{1,2}, \ldots, e_{3,3}]$. The generalized error vector, z, in Figure 6.8 consist of all spacing errors and control efforts in the formation. We will measure performance as the H_∞ gain from generalized disturbances to generalized errors. In particular, we are measuring the gain from the desired velocity trajectory, w, to the resulting errors and control efforts of all vehicles in the mesh, z.

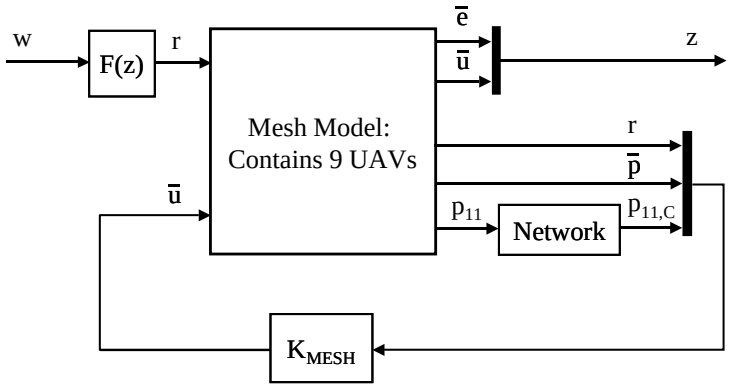

Figure 6.8: Generalized Plant for a Formation of 9 UAVs

We assume that the UAVs are able to sense their relative positions and range rates with respect to their neighbors. Each UAV has their own controller that depends on this sensed information as well as communicated leader information. The (1,2) and (2,1) followers do not require this communicated information because it is assumed they can sense the leader motion. Conceptually, all control laws can be stacked into a large formation control law, K_{MESH} (Figure 6.8). This large formation control law has access to the reference trajectory, r, all UAV measurements, p_{ij}, and the communicated leader information, $p_{11,C}$. We can then define K_{MESH} with PD controllers in the proper entries to represent the control actions of all UAVs. In a slight abuse of notation, we let $p_{ij} \in \mathbb{R}^8$,

consist of the three positions and the heading of the UAV as well as their derivatives. This allows us to write K_{MESH} as a constant matrix. This static feedback gain acts at a sample time of $T = 100msec$. We assume that every $100msec$ the leader communicates its measurements to the followers. All followers either receive a correct version or a corrupted version of the leader information. Thus the network model consists of two modes depending on whether the leader information is successfully transmitted or not.

To summarize, the closed loop plant, denoted $\mathcal{S}$, consists of a formation of 9 UAVs. For each packet loss rate, we measure the gain from the desired velocity trajectory to the resulting errors and control efforts of all vehicles in the mesh. According to Theorem 5.3, $\mathcal{S}$ is SMS and satisfies $\|\mathcal{S}\|_\infty < \gamma$ if and only if there exists a symmetric matrix $G > 0$ satisfying the following matrix inequality:

$$\begin{bmatrix} G & 0 \\ 0 & \gamma^2 I \end{bmatrix} - \sum_{j=1}^{2} p_j \begin{bmatrix} A_j & B_j \\ C_j & D_j \end{bmatrix}^T \begin{bmatrix} G & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A_j & B_j \\ C_j & D_j \end{bmatrix} > 0 \tag{6.7}$$

We can find the H_∞ gain for $\mathcal{S}$ by minimizing γ subject to this linear matrix inequality. Since $\mathcal{S}$ has large state dimension, the resulting semi-definite programming problem is time-consuming to solve. An alternative approach is given in Appendix C and is based on bisection and iterating a Riccati difference equation. We have only proved that this method gives a lower bound on the true gain, but we conjecture that it actually gives the true cost. This algorithm is much faster than solving the semi-definite program for problems where the system has large state dimension.

Figure 6.9 gives the H_∞ gain as a function of packet loss rate. There are several interesting aspects to this plot. For $p = 0$, the leader information is always received while for $p = 1$ the leader information is always lost. For these two particular values of p, the

network model reduces to a single mode. In this case, we can find the H_∞ gain using standard algorithms for discrete-time systems [4]. The two 'X' labels on the plot are the values returned by the μ-tools code. These values are within the the tolerances specified for the bisection algorithm given in Appendix C. The plot also shows that the performance rapidly degrades as the packet loss rate increases. Moreover, if the packet loss rate is greater than 0.50, then we are actually better off not using the information. The dashed line is at the performance level for the case where leader information is always lost ($p = 1$). This line makes it clear that if the packet loss rate is greater than 0.50, then the performance is worse than if $p = 1$. This does not necessarily say that the leader information is detrimental if $p \in (0.5, 1)$. However it does say that the leader information is detrimental given the control structure we have specified. Thus we must be careful when designing control laws for systems with wireless links in the feedback loop. Finally, we note that this curve was generated for a sample rate of $T = 100msec$. Similar to the discussion in Section 4.5, the network bandwidth (in bits/sec) is tied to the sample rate. Thus, we can generate several performance curves at different sample rates to find a trade-off between network bandwidth and packet loss rate.

6.5 Conclusions

The goal of this chapter was to demonstrate the ideas presented in the thesis on a simple cooperative control problem: formation flight. First, we tried to design a formation controller via simple extension of a wavepoint controller for a single UAV. This design failed to be mesh stable and resulted in errors amplifying as they propagated through the mesh.

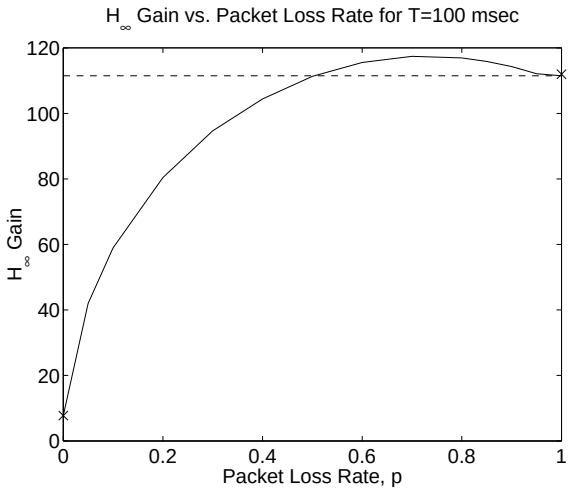

Figure 6.9: Normalized H_∞ performance vs. packet loss rate for a formation of 9 UAVs

Using the design procedure in Section 3, we were able to design a mesh stable formation control law. The cost of this design is that leader information must be communicated to the followers in the formation. In Section 6.4 we analyzed the effect of packet losses in communicating this lead vehicle information.

Chapter 7

Conclusions and Recommendations

This thesis considered a simple coordinated vehicle control problem: formation flight. The basic question in the design of a formation flight controller is: what should be communicated and how often should it be communicated? In a simple manner we showed that many strategies that do not require communication tend to be string or mesh unstable. That is, errors are amplified as they propagate and hence these strategies are sensitive to disturbances. This motivated a control design procedure for formation flight that required communicated leader information. We then determined how often this information must be communicated for acceptable control. The 'how often' is determined by the sample rate of the system as well as the packet loss characteristics of the network. In Chapter 4, we found theoretical bounds on network performance for a simple vehicle following problem. These bounds constrain the sample rate and packet loss rate. In Chapter 5, we developed tools to determine how often information must be communicated in more general networked control systems. The main tool of interest was a computationally tractable method to find

the closed loop performance (measured using the H_∞ gain) as a function of packet loss rate. Finally, we applied the control design procedure and the network analysis tool to the formation flight control problem.

The results presented in this thesis leave several topics open for future research. Three areas are discussed below.

- **Investigate different distributed control architectures:** In this thesis, we showed that some simple distributed control architectures are string / mesh unstable. We then offered one solution: the communication of leader information. A more general investigation of distributed control architectures is required. This would show the performance benefit gained as more information is communicated. As discussed in Chapter 3, one approach is to study the ties with graph theory. Another approach is to improve upon the results in Chapter 5. In Section 5.6, we turned the centralized control design problem for NCS into a convex optimization. Instead, one could solve this problem with various structural constraints on the controller. This will give intuition on the effect of extra communication.

- **Explore design aspects for networked control systems:** This is a broad heading and it encompasses several ideas for future work. First, a systematic procedure to model networks (both wired and wireless) for the purpose of control design should be pursued. A stochastic model of the packet delivery characteristics seems appropriate. A second area for future research involves coding for networked control systems. Current coding schemes try to minimize the probability of packet errors. For control, it is more important that the coding scheme keeps the difference between the received

and delivered packets small. There are also several ways to generalize the results in this thesis. At the end of Chapter 4, possible extensions of the theoretical results are given. Furthermore, the tools developed in Chapter 5 can be enhanced by extending the results to general Markov Jump Linear Systems. Moreover, the example given in Section 6.4 is of moderate size and it is computationally difficult to solve. Numerical algorithms for the analysis and design of large networked control systems should be developed. Finally, the joint design of the network and controller should be investigated. Some work of this nature has recently been done by Xiao, et.al. [107].

- **Experimentally verify the theoretical results:** The example presented in Chapter 6 verified the theoretical results to some extent, but these results should be experimentally verified. There is an experimental facility at Berkeley to perform a small-scale formation flight, but several steps will need to precede this experiment. First, the system modeling should be redone to obtain a linear model valid for cruise flight conditions. Second, simulations should be performed with the inclusion of the nonlinear rotation from body-fixed to earth-fixed coordinates. From this we can verify or disprove the assumption that this rotation can be neglected. At this point, a small-scale formation flight test can be performed. Unfortunately, the unmanned helicopters are an expensive testbed and are not ideal for testing distributed control algorithms. Another avenue for experimental verification of the theoretical results is the construction of a robot testbed. This is currently being pursued.

Bibliography

[1] H. Abou-Kandil, G. Freiling, and G. Jank. On the solution of discrete-time markovian jump linear quadratic control problems. *Automatica*, 31(5):765–768, 1995.

[2] B.D.O. Anderson and J.B. Moore. *Optimal Control: Linear Quadratic Methods*. Prentice Hall, 1990.

[3] Tamer Başar. Generalized Riccati equation in dynamic games. In Sergio Bittanti, Alan J. Laub, and Jan C. Willems, editors, *The Riccati Equation*, chapter 11, pages 293–333. Springer Verlag, 1991.

[4] G.J. Balas, J.C. Doyle, K. Glover, A. Packard, and R. Smith. *μ-Analysis and synthesis toolbox user's guide*. MathWorks, Natick, Mass., 1993.

[5] B. Bamieh, F. Paganini, and M. Dahleh. Distributed control of spatially-invariant systems. Submitted to IEEE Transactions on Automatic Control.

[6] S. Banda, J. Doyle, R. Murray, J. Paduano, J. Speyer, and G. Stein. Research needs in dynamics and control for uninhabited aerial vehicles. http://www.cds.caltech.edu/ murray/notes/uav-nov97.html, November 1997. Panel Report.

[7] El-Kebir Boukas and Peng Shi. H_∞ control for discrete-time linear systems with markovian jumping parameters. In *Proceedings of the 36th Conference on Decision and Control*, pages 4134–4139, December 1997.

[8] S. Boyd and C.A. Desoer. Subharmonic functions and performance bounds on linear time-invariant feedback systems. *IMA Journal of Mathematical Control and Information*, 2:153–170, 1985.

[9] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, 1994.

[10] P.E. Caines and D.Q. Mayne. On the discrete time matrix Riccati equation of optimal control. *International Journal of Control*, 12(5):785–794, 1970.

[11] Yong-Yan Cao and James Lam. Stochastic stabilizability and H_∞ control for discrete-time jump linear systems with time delay. *Journal of the Franklin Institute*, 336:1263–1281, 1999.

[12] R.J. Caudill and W.L. Garrard. Vehicle-follower longitudinal control for automated transit vehicles. *Journal of Dynamic Systems, Measurement, and Control*, pages 241–248, December 1977.

[13] H. Chan and Ü. Özgüner. Optimal control of systems over a communication network with queues via a jump system approach. In *Proceedings of the 4th IEEE Conference on Control Applications*, pages 1148–1153, September 1995.

[14] J. Chen. On logarithmic complementary sensitivity integrals for MIMO systems. In *Proceedings of the American Control Conference*, pages 3529–3530, June 1998.

[15] J. Chen. Logarithmic integrals, interpolation bounds, and performance limitations in MIMO feedback systems. *IEEE Transactions on Automatic Control*, 45(6):1098–1115, June 2000.

[16] H.Y. Chiu, G.B. Stupp, and S.J. Brown. Vehicle-follower control with variable-gains for short headway automated guideway transit systems. *Journal of Dynamic Systems, Measurement, and Control*, pages 183–189, September 1977.

[17] H.J. Chizeck and Y. Ji. Optimal quadratic control of jump linear systems with gaussian noise in discrete-time. In *Proceedings of the 27th Conference on Decision and Control*, pages 1989–1993, December 1988.

[18] H.J. Chizeck, A.S. Willsky, and D. Castanon. Discrete-time markovian-jump linear quadratic optimal control. *International Journal of Control*, 43(1):213–231, 1986.

[19] K. Chu. Decentralized control of high speed vehicular strings. *Transportation Science*, pages 361–384, 1974.

[20] O.L.V. Costa and Ricardo P. Marques. Mixed H_2/H_∞-control of discrete-time markovian jump linear systems. *IEEE Transactions on Automatic Control*, 43(1):95–100, January 1998.

[21] Oswaldo L.V. Costa and Marcelo D. Fragoso. Stability results for discrete-time linear systems with markovian jumping parameters. *Journal of Mathematical Analysis and Applications*, 179:154–178, 1993.

[22] C.J. Cutts and J.R. Speakman. Energy savings in formation flight of pink-footed geese. In *Journal of Experimental Biology*, volume 189, pages 251–261, 1994.

[23] M.A. Dahleh and I.J. Diaz-Bobillo. *Control of Uncertain Systems: A Linear Programming Approach*. Prentice Hall, 1995.

[24] D.P. de Farias, J.C. Geromel, J.B.R. do Val, and O.L.V. Costa. Output feedback control of markov jump linear systems in continuous-time. *IEEE Transactions on Automatic Control*, 45(5):944–949, May 2000.

[25] Carlos de Souza. On stabilizing properties of solutions of the Riccati difference equation. *IEEE Transactions on Automatic Control*, 34(12):1313–1316, December 1989.

[26] James W. Demmel. *Applied Numerical Linear Algebra*. SIAM, 1997.

[27] J.B.R. do Val, J.C. Geromel, and O.L.V. Costa. Uncoupled Riccati iterations for the linear quadratic control problem of discrete-time markov jump linear systems. *IEEE Transactions on Automatic Control*, 43(12):1727–1733, December 1998.

[28] J. Eker, A. Cervin, and A. Hörjel. Distributed wireless control using Bluetooth. Preprint, 2001.

[29] J. Eyre, D. Yanakiev, and I. Kanellakopoulos. A simplified framework for string stability analysis of automated vehicles. *Vehicle System Dynamics*, 30(5):375–405, November 1998.

[30] J.A. Fax and R.M. Murray. Graph Laplacians and vehicle formation stabilization.

Technical Report CDS 01-007, Control and Dynamical Systems, California Institute of Technology, 2001.

[31] FiRE: Uninhabited aerial vehicle first response experiment. http://geo.arc.nasa.gov/sge/UAVFiRE, February 2000.

[32] L. El Ghaoui, R. Nikoukhah, and F. Delebecque. *LMItool: A front-end for LMI optimization, user's guide*, February 1995. Available via anonymous ftp to `ftp.ensta.fr` under `/pub/elghaoui/lmitool`.

[33] Laurent El Ghaoui and Mustapha Ait Rami. Robust state-feedback stabilization of jump linear systems via LMIs. *International Journal of Robust and Nonlinear Control*, 6(9):1015–1022, 1996.

[34] F. Giulietti, L. Pollini, and M. Innocenti. Autonomous formation flight. *IEEE Control Systems Magazine*, 20(6):34–44, December 2000.

[35] D. N. Godbole. Control and coordination in uninhabited combat air vehicles. In *Proceedings of the American Control Conference*, pages 1487–1490, June 1999.

[36] F. Göktas, J.M. Smith, and R. Bajcsy. μ-Synthesis for distributed control systems with network-induced delays. In *IEEE Conference on Decision and Control*, pages 813–814, September 1996.

[37] A. Hassibi, S. Boyd, and J. How. Control of asynchronous dynamical systems with rate constraints on events. In *Proceedings of the 38th IEEE Conference on Decision and Control*, pages 1345–1351, December 1999.

[38] J.K. Hedrick and D. Swaroop. Dynamic coupling in vehicles under automatic control. In *13th IAVSD Symposium*, pages 209–220, August 1993.

[39] J.K. Hedrick, M. Tomizuka, and P. Varaiya. Control issues in automated highway systems. *IEEE Control Systems Magazine*, 14(6):21–32, December 1994.

[40] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, 1999.

[41] Y. Ji and H.J. Chizeck. Controllability, observability, and discrete-time markovian jump linear quadratic control. *International Journal of Control*, 48(2):481–498, 1988.

[42] Y. Ji and H.J. Chizeck. Jump linear quadratic gaussian control: Steady-state solution and testable conditions. *Control Theory and Advanced Technology*, 6(3):289–318, 1990.

[43] Y. Ji, H.J. Chizeck, X. Feng, and K.A. Loparo. Stability and control of discrete-time jump linear systems. *Control Theory and Advanced Technology*, 7(2):247–270, 1991.

[44] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35–45, March 1960.

[45] E.W. Kamen. Stabilization of linear spatially-distributed continuous-time and discrete-time systems. In N.K. Bose, editor, *Multidimensional Systems Theory*. Kluwer, 1985.

[46] H.J. Kim, R. Vidal, D.H. Shim, O. Shakernia, and S. Sastry. A hierarchical approach to probabilistic pursuit-evasion games with unmanned ground and aerial vehicles. In *IEEE Conference on Decision and Control*, December 2001.

[47] R. Krtolica, Ü. Özgüner, H. Chan, H. Göktas, J. Winkelman, and M. Liubakka. Stability of linear feedback systems with random communication delays. *International Journal of Control*, 59(4):925–953, 1994.

[48] D. Lee, R. Attias, A. Puri, R. Sengupta, S. Tripakis, and P. Varaiya. A wireless token ring protocol for intelligent transportation systems. In *IEEE 4th International Conference on Intelligent Transportation Systems*, August 2001.

[49] W.S. Levine and M. Athans. On the optimal error regulation of a string of moving vehicles. *IEEE Transactions on Automatic Control*, AC-11:355–361, July 1966.

[50] F. Lian, J.R. Moyne, and D. Tilbury. Control performance study of a networked machining cell. In *Proceedings of the American Control Conference*, pages 2337–2341, June 2000.

[51] F. Lian, J.R. Moyne, and D. Tilbury. Performance evaluation of control networks: Ethernet, ControllNet and DeviceNet. *IEEE Control Systems Magazine*, 21(1):66–83, February 2001.

[52] P.B.S. Lissaman and C.A. Shollenberger. Formation flight of birds. *Science*, 168:1003–1005, 1970.

[53] D.P. Looze and J.S. Freudenberg. Tradeoffs and limitations in feedback systems. In W.S. Levine, editor, *The Control Handbook*, chapter 31, pages 537–549. CRC Press, 1996.

[54] Sonia Mahal. Effects of communication delays on string stability in an AHS environment. Master's thesis, University of California at Berkeley, March 2000.

[55] R.K. Mehra, J.D. Bošković, and S. Li. Autonomous formation flying of multiple UCAVs under communication failure. In *Position Locations and Navigation Symposium*, pages 371–378. IEEE, March 2000.

[56] G. Meinsma. Unstable and nonproper weights in H_∞ control. *Automatica*, 31(11):1655–1658, 1995.

[57] S.M. Melzer and B.C. Kuo. Optimal regulation of systems described by a countably infinite number of objects. *Automatica*, 7:359–366, 1971.

[58] M. Mesbahi. On a dynamic extension of the theory of graphs. Preprint, September 2001.

[59] M. Mesbahi and F.Y. Hadaegh. Formation flying control of multiple spacecraft via graphs, matrix inequalities, and switching. *AIAA Journal of Guidance, Control and Dynamics*, 24(2):369–377, March 2001.

[60] B. Mettler, M.B. Tischler, and T. Kanade. System identification modeling of a small-scale unmanned rotorcraft for flight control design. To appear in the Journal of the American Helicopter Society.

[61] R.H. Middleton and G.C. Goodwin. *Digital Control and Estimation: A Unified Approach*. Prentice Hall, Englewood Cliffs, N.J., 1990.

[62] R.M. Murray and J.A. Fax. Information flow and cooperative control of vehicle formations. Technical Report CDS 01-011, Control and Dynamical Systems, California Institute of Technology, 2001.

[63] G.T. Nguyen, R.H. Katz, B. Noble, and M. Satyanarayanan. A trace-based approach for modeling wireless channel behavior. In *Proceedings of 1996 Winter Simulation Conference*, pages 597–604, 1996.

[64] J. Nilsson, B. Bernhardsson, and B. Wittenmark. Stochastic analysis and control of real-time systems with random time delays. *Automatica*, 34(1):57–64, 1998.

[65] M. Pachter, J.J. D'Azzo, and A.W. Proud. Tight formation flight control. *Journal of Guidance, Control, and Dynamics*, 24(2):246–254, March-April 2001.

[66] A. Packard and J. Doyle. The complex structured singular value. *Automatica*, 29(1):71–109, January 1993.

[67] Z. Pan and T. Başar. H_∞-control of markovian jump systems and solutions to associated piecewise-deterministic differential games. In G.J. Olsder, editor, *New Trends in Dynamic Games and Applications*, pages 61–94. Birkhäuser, 1995.

[68] Lloyd E. Peppard. String stability of relative-motion vehicle control systems. *IEEE Transactions on Automatic Control*, pages 579–581, October 1974.

[69] W.K. Potts. The chorus-line hypothesis of manoeuvre coordination in avian flocks. *Nature*, 309:344–345, May 1984.

[70] R. Raji. Smart networks for control. *IEEE Spectrum*, 31(6):49–55, June 1994.

[71] Mustapha Ait Rami, Jean-Pierre Folcher, Laurent El Ghaoui, Pascale Bendotti, and Clement-Marc Falinower. Control of jump linear systems: Application to the steam

generator water level. In *Proceedings of the 38th IEEE Conference on Decision and Control*, pages 4923–4928, December 1999.

[72] Mustapha Ait Rami and Laurent El Ghaoui. LMI optimization for nonstandard Riccati equations arising in stochastic control. *IEEE Transactions on Automatic Control*, 41(11):1666–1671, November 1996.

[73] A. Ray. Performance evaluation of medium access control protocols for distributed digital avionics. *ASME Journal of Dynamic Systems, Measurement and Control*, 109(4):370–377, December 1987.

[74] A. Ray. Introduction to networking for integrated control systems. *IEEE Control Systems Magazine*, 9(1):76–79, January 1989.

[75] Craig W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *Proceedings of the SIGGRAPH '87*, 1987.

[76] C. Richardson and M. Schoultz. Formation flight system design concept. In *Proceedings of the IEEE/AIAA 10th Digital Avionics Systems Conference*, pages 18–25, October 1991.

[77] A. Robertson, T. Corazzini, and J.P. How. Formation sensing and control technologies for a separated spacecraft interferometer. In *Proceedings of the American Control Conference*, pages 1574–1579, June 1998.

[78] H.L. Royden. *Real Analysis*. Macmillan Publishing Company, 1988.

[79] M.S. Santina, A.R. Stubberud, and G.H. Hostetter. Quantization effects. In W.S. Levine, editor, *The Control Handbook*, chapter 15, pages 301–311. CRC Press, 1996.

[80] D.A. Schoenwald. AUVs: In space, air, water, and on the ground. *IEEE Control Systems Magazine*, 20(6):15–18, December 2000.

[81] P. Seiler and R. Sengupta. Analysis of communication losses in vehicle control problems. In *Proceedings of the American Control Conference*, pages 1491–1496, June 2001.

[82] Pete Seiler, Aniruddha Pant, and J.K. Hedrick. Preliminary investigation of mesh stability for linear systems. In *Proceedings of the ASME: DSC Division*, volume 67, pages 359–364, 1999.

[83] O. Shakernia, Y. Ma, T.J. Koo, and S. Sastry. Landing an unmanned aerial vehicle: Vision based motion estimation and nonlinear control. *Asian Journal of Control*, 1(3):128–145, September 1999.

[84] E. Shaw. Fish in schools. *Natural History*, 84(8):40–46, 1975.

[85] Shahab Sheikholeslam and Charles A. Desoer. Longitudinal control of a platoon of vehicles. In *Proceedings of the American Control Conference*, volume 1, pages 291–297, May 1990.

[86] D.H. Shim, H.J. Kim, and S. Sastry. Control system design for rotorcraft-based unmanned aerial vehicles using time-domain system identification. In *IEEE International Conference on Control Applications*, 2000.

[87] H. Shim. *Hierarchical Flight Control System Synthesis for Rotorcraft-based Unmanned Aerial Vehicles*. PhD thesis, University of California at Berkeley, Fall 2000.

[88] B. Shu. Robust longitudinal control of vehicle platoons on intelligent highways. Master's thesis, University of Illinois at Urbana-Champaign, 1996.

[89] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. John Wiley and Sons, 1996.

[90] J.B. Slaughter. Quantization errors in digital control systems. *IEEE Transactions on Automatic Control*, 9:70–74, January 1964.

[91] R.V. Stachnik, K. Ashlin, and S. Hamilton. Space station-SAMSI: A spacecraft array for michelson spatial interferometry. *Bulletin of the American Astronomical Society*, 16(3):818–827, 1984.

[92] D.J. Stillwell and B.E. Bishop. Platoons of underwater vehicles. *IEEE Control Systems Magazine*, 20(6):34–44, December 2000.

[93] A. Stubbs and G.E. Dullerud. Networked control of distributed systems: A testbed. In *ASME International Mechanical Engineering Congress and Exposition*, November 2001.

[94] D. Swaroop. *String Stability of Interconnected Systems: An Application to Platooning in Automated Highway Systems*. PhD thesis, University of California at Berkeley, 1994.

[95] D. Swaroop and J.K. Hedrick. String stability of interconnected systems. *IEEE Transactions on Automatic Control*, 41(4):349–356, March 1996.

[96] D. Swaroop and J.K. Hedrick. Constant spacing strategies for platooning in automated highway systems. *Transactions of the ASME*, 121:462–470, September 1999.

[97] S. Tatikonda. *Control under communication constraints*. PhD thesis, Massachusetts Institute of Technology, September 2000.

[98] TechSat 21: Space missions using satellite clusters. http://www.vs.afrl.af.mil/Factsheets/techsat21.html, June 1998.

[99] G.C. Walsh, O. Beldiman, and L.G. Bushnell. Asymptotic behavior of nonlinear networked control systems. *IEEE Transactions on Automatic Control*, 46(7):1093–1097, 2001.

[100] G.C. Walsh and H. Ye. Scheduling of networked control systems. *IEEE Control Systems Magazine*, 21(1):57–65, February 2001.

[101] G.C. Walsh, H. Ye, and L. Bushnell. Stability analysis of networked control systems. In *Proceedings of the American Control Conference*, pages 2876–2880, June 1999.

[102] D.F. Wilkie. A moving cell control scheme for automated transit systems. *Transportation Science*, pages 347–364, 1970.

[103] T.C. Williams, T.J. Klonowski, and P. Berkeley. Angle of canada goose V flight formation measured by radar. *Auk*, 93:554–559, 1976.

[104] B. Wittenmark, J. Nillson, and M. Törngren. Timing problems in real-time control systems. In *Proceedings of the American Control Conference*, pages 2000–2004, June 1995.

[105] W.S. Wong and R.W. Brockett. Systems with finite communication bandwidth constraints-part I: State estimation problems. *IEEE Transactions on Automatic Control*, 42(9):1294–1299, September 1997.

[106] W.S. Wong and R.W. Brockett. Systems with finite communication bandwidth constraints-part II: Stabilization with limited information feedback. *IEEE Transactions on Automatic Control*, 44(5):1049–1053, May 1999.

[107] L. Xiao, M. Johansson, H. Hindi, S. Boyd, and A. Goldsmith. Joint optimization of communication rates and linear systems. Submitted to the IEEE Transactions on Automatic Control.

[108] Lin Xiao, Arash Hassibi, and Jonathan How. Control with random communication delays via a discrete-time jump system approach. In *Proceedings of the American Control Conference*, pages 2199–2204, June 2000.

[109] H. Ye, G.C. Walsh, and L. Bushnell. Wireless local area networks in the manufacturing industry. In *Proceedings of the American Control Conference*, pages 2363–2367, June 2000.

[110] W. Zhang, M.S. Branicky, and S.M. Phillips. Stability of networked control systems. *IEEE Control Systems Magazine*, 21(1):84–99, February 2001.

[111] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, 1996.

Appendix A

Matrix Sequence Lemma

In this appendix, we prove a lemma concerning sequences of matrices. This lemma is used to bound the maximum singular value of a transfer function matrix in Theorem 2.2.

Lemma A.1 *Given any complex numbers, $a, b \in \mathbb{C}$, define the following sequence of matrices:*

$$X_N := \begin{bmatrix} 1 & & & & \\ b & 1 & & & \\ ab & b & 1 & & \\ \vdots & \ddots & \ddots & \ddots & \\ a^{N-2}b & \cdots & ab & b & 1 \end{bmatrix} \in \mathbb{C}^{N \times N}$$

If $|a| < 1$ then $\bar{\sigma}(X_N) \leq 1 + \frac{|b|}{1-|a|}$ for all N.

Proof. For any matrix, $\rho(A) \leq \|A\|_1$ where $\rho(A)$ is the spectral radius and $\|A\|_1$ is the induced 1-norm. Applying this fact to $A = X_N^* X_N$ gives inequality (a) in the following

upper bound of $\bar{\sigma}(X_N)$ [40]:

$$\bar{\sigma}(X_N)^2 \overset{(a)}{\leq} \|X_N^* X_N\|_1 \leq \|X_N^*\|_1 \|X_N\|_1 = \|X_N\|_\infty \|X_N\|_1 \tag{A.1}$$

For all N, $\|X_N\|_1 = \|X_N\|_\infty = 1 + \frac{|b|(1-|a|^{N-1})}{1-|a|}$. If $|a| < 1$ then $\bar{\sigma}(X_N) \leq 1 + \frac{|b|}{1-|a|}$ $\forall N$.

This lemma can also be proved using an interesting dynamical system interpretation. Consider the following discrete time dynamical system:

$$\begin{bmatrix} x(k+1) \\ \\ y(k) \end{bmatrix} = \begin{bmatrix} a & b \\ \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ \\ u(k) \end{bmatrix}, \qquad x(1) = 0 \tag{A.2}$$

This discrete time system is intimately related to the sequence of matrices, X_N. Let u_N and y_N be vectors containing the input and output sequences for this discrete time system from time 1 up to time N. One can easily show that $y_N = X_N u_N$.

The transfer function for this system is $G(z) := \frac{b}{z-a} + 1$. Its H_∞ norm is given by:

$$\|G(z)\|_\infty := \sup_{0 \leq \omega \leq 2\pi} |G(e^{j\omega})|$$

This norm has an input-output gain interpretation. We define ℓ_2 as the set of all sequences, u, such that $\|u\|_2 := \sqrt{\left[\sum_{k=1}^{\infty} u_k^2\right]} < \infty$. The H_∞ norm for a discrete time system is equal to its induced $\ell_2 - \ell_2$ norm [23]. Therefore, if u and y are the input and output sequences and $u \in \ell_2$, then $\|y\|_2 \leq \|G(z)\|_\infty \|u\|_2$. Apply this result (inequality (a)) to any input sequence such that $u(k) = 0$ for $k > N$:

$$\|y_N\|_2 \leq \|y\|_2 \overset{(a)}{\leq} \|G(z)\|_\infty \|u\|_2 = \|G(z)\|_\infty \|u_N\|_2$$

It follows that for all N, $\bar{\sigma}(X_N) \leq \|G(z)\|_\infty$. Since $|a| < 1$, the discrete time system in Equation A.2 is stable and its H_∞ norm is finite. A simple upper bound on $\|G(s)\|_\infty$ gives

a uniform bound on $\bar{\sigma}(X_N)$:

$$\bar{\sigma}(X_N) \leq \|G(z)\|_\infty \leq 1 + \frac{|b|}{1-|a|} \tag{A.3}$$

■

In this appendix, we prove two lemmas related to the Riccati Difference Equation. These lemmas are used in proving theoretical bounds for networked control systems.

Lemma A.2 *Consider the following Riccati Difference Equation:*

$$\begin{aligned} M(k+1) &= AM(k)A^T + Q \\ &\quad -AM(k)C(k)^T[C(k)M(k)C(k)^T + V]^{-1}C(k)M(k)A^T \\ M(0) &= M \end{aligned} \tag{A.4}$$

where $A, M, Q \in \mathbb{R}^{n_x \times n_x}$, $C(k) \in \mathbb{R}^{n_y \times n_x}$, *and* $V \in \mathbb{R}^{n_y \times n_y}$. *Furthermore,* $V, M > 0$ *and* $Q \geq 0$. *Then* $\forall N$, *there exists* $\tilde{V}_N > 0$, $\tilde{Q}_N \geq 0$, *and* $\tilde{R}_N$ *such that* $M(N+1)$ *and* M *are related as follows:*

$$\begin{aligned} M(N+1) &= A^{N+1}M(A^T)^{N+1} + \tilde{Q}_N \\ &\quad - \left[O_N M(A^T)^{N+1} + \tilde{R}_N\right]^T \left[O_N M O_N^T + \tilde{V}_N\right]^{-1} \left[O_N M(A^T)^{N+1} + \tilde{R}_N\right] \end{aligned} \tag{A.5}$$

O_N *is the time-varying observability matrix (not necessarily full column rank) from time* 0 *to time* N*:*

$$O_N = \begin{bmatrix} C(0) \\ C(1)A \\ \vdots \\ C(N)A^N \end{bmatrix} \tag{A.6}$$

Moreover, O_N, $\tilde{V}_N$, $\tilde{Q}_N$, *and* $\tilde{R}_N$ *depend only on* A, Q, V *and* $C(0)$,...,$C(N)$.

Proof. The following equality is proved by induction:

$$M(k+1) = A^{k+1}M(A^T)^{k+1} + \sum_{j=0}^{k} A^j Q (A^T)^j - \left[\begin{array}{c} O_k M (A^T)^{k+1} + \sum_{j=1}^{k} O_{j,k} Q (A^T)^{k+1-j} \end{array} \right]^T \cdot$$
$$\left[\begin{array}{c} O_k M O_k^T + V_k + \sum_{j=1}^{k} O_{j,k} Q O_{j,k}^T \end{array} \right]^{-1} \left[\begin{array}{c} O_k M (A^T)^{k+1} + \sum_{j=1}^{k} O_{j,k} Q (A^T)^{k+1-j} \end{array} \right] \tag{A.7}$$

where V_k and $O_{j,k}$ $(k \geq j)$ are defined as:

$$V_k = \begin{bmatrix} V & & 0 \\ & \ddots & \\ 0 & & V \end{bmatrix} \in \mathbb{R}^{(k+1)n_y \times (k+1)n_y}, \qquad O_{j,k} = \begin{bmatrix} \left. \begin{matrix} 0_{n_y \times n_x} \\ \vdots \\ 0_{n_y \times n_x} \end{matrix} \right\} j \\ C(j) \\ \vdots \\ C(k)A^{k-j} \end{bmatrix} \in \mathbb{R}^{(k+1)n_y \times n_x}$$

$O_{j,k}$ is just the time-varying observability matrix from time j up to time k with an appropriate block of zeros. V_k is a block diagonal matrix with $k+1$ copies of V on the diagonal blocks. For example, $O_{0,N} = O_N$ and $V_0 = V$.

For $k = 0$, Equation A.7 follows directly from Equation A.4. Next make the induction assumption that Equation A.7 holds for $k = N-1$ and prove that it must hold for $k = N$ as well. $M(N+1)$ can be written in terms of M:

$$\begin{aligned} M(N+1) &\overset{(a)}{=} AM(N)A^T + Q \\ &\quad - AM(N)C(N)^T[C(N)M(N)C(N)^T + V]^{-1}C(N)M(N)A^T \\ &\overset{(b)}{=} A\left(S - L^T X^{-1} L\right) A^T + Q \\ &\quad - A\left[S - L^T X^{-1} L\right]^T C(N)^T Z^{-1} C(N) \left[S - L^T X^{-1} L\right]^T A^T \end{aligned}$$

where equality (a) follows from Equation A.4. Equality (b) follows from the induction assumption that Equation A.7 holds for $k = N-1$ and the notation $M(N) = S - L^T X^{-1} L$:

$$\begin{aligned}
S &\doteq A^N M (A^T)^N + \sum_{j=0}^{N-1} A^j Q (A^T)^j \\
X &\doteq O_{N-1} M O_{N-1}^T + V_{N-1} + \sum_{j=1}^{N-1} O_{j,N-1} Q O_{j,N-1}^T \\
L &\doteq O_{N-1} M (A^T)^N + \sum_{j=1}^{N-1} O_{j,N-1} Q (A^T)^{N-j} \\
Z &\doteq C(N) M(N) C(N)^T + V = C(N) \left(S - L^T X^{-1} L \right) C(N)^T + V
\end{aligned}$$

Grouping terms appropriately yields:

$$\begin{aligned}
M(N+1) =& ASA^T + Q \\
& - A \begin{bmatrix} L \\ C(N)S \end{bmatrix}^T \begin{bmatrix} X^{-1} + X^{-1} L C(N)^T Z^{-1} C(N) L^T X^{-1} & -X^{-1} L C(N)^T Z^{-1} \\ -Z^{-1} C(N) L^T X^{-1} & Z^{-1} \end{bmatrix} \begin{bmatrix} L \\ C(N)S \end{bmatrix} A^T \\
=& A^{N+1} M (A^T)^{N+1} + \left(\sum_{j=0}^{N} A^j Q (A^T)^j \right) \\
& - A \begin{bmatrix} L \\ C(N)S \end{bmatrix}^T \begin{bmatrix} X & LC(N)^T \\ C(N)L^T & C(N)SC(N)^T + V \end{bmatrix}^{-1} \begin{bmatrix} L \\ C(N)S \end{bmatrix} A^T
\end{aligned} \tag{A.8}$$

The second equality follows from a version of the Matrix Inversion Lemma [2]. Replace the temporary variables (S, X, L, Z) with their definitions given above to obtain the following two relations:

$$\begin{aligned}
\begin{bmatrix} L \\ C(N)S \end{bmatrix} A^T &= \begin{bmatrix} O_{N-1} M (A^T)^{N+1} \\ C(N) A^N M (A^T)^{N+1} \end{bmatrix} + \begin{bmatrix} \sum_{j=1}^{N-1} O_{j,N-1} Q (A^T)^{N-j} \\ C(N) \sum_{j=1}^{N-1} A^{N-j} Q (A^T)^{N-j} \end{bmatrix} A^T + \begin{bmatrix} 0 \\ C(N)Q \end{bmatrix} A^T \\
&= O_N M (A^T)^{N+1} + \sum_{j=1}^{N} O_{j,N} Q (A^T)^{N+1-j}
\end{aligned} \tag{A.9}$$

$$\begin{aligned}\begin{bmatrix} X & LC(N)^T \\ C(N)L^T & C(N)SC(N)^T+V \end{bmatrix} &= \begin{bmatrix} O_{N-1}MO_{N-1}^T & O_{N-1}M(A^T)^NC(N)^T \\ C(N)A^NMO_{N-1}^T & C(N)A^NM(A^T)^NC(N)^T \end{bmatrix} + \begin{bmatrix} V_{N-1} & 0 \\ 0 & V \end{bmatrix} \\ &+ \sum_{j=1}^{N-1} \begin{bmatrix} O_{j,N-1}QO_{j,N-1}^T & O_{j,N-1}Q(A^T)^{N-j}C(N)^T \\ C(N)A^{N-j}QO_{j,N-1}^T & C(N)A^{N-j}Q(A^T)^{N-j}C(N)^T \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & C(N)QC(N)^T \end{bmatrix} \quad \text{(A.10)} \\ &= O_NMO_N^T + V_N + \sum_{j=1}^{N} O_{j,N}QO_{j,N}^T\end{aligned}$$

Equations A.8, A.9, and A.10 show that Equation A.7 holds for $k = N$. By induction, Equation A.7 holds for all k.

Finally, Equation A.5 follows from Equation A.7 with the following defintions:

$$\begin{aligned}\tilde{Q}_N &= \sum_{j=0}^{N} A^jQ(A^T)^j \\ \tilde{R}_N &= \sum_{j=1}^{N} O_{j,N}Q(A^T)^{N+1-j} \\ \tilde{V}_N &= V_N + \sum_{j=1}^{N} O_{j,N}QO_{j,N}^T\end{aligned}$$

Note that $\tilde{Q}_N \geq 0$ and $\tilde{V}_N > 0$ since $Q \geq 0$ and $V > 0$. Furthermore, O_N, $\tilde{V}_N$, $\tilde{Q}_N$, and $\tilde{R}_N$ depend only on A, Q, V and $C(0),\ldots,C(N)$ as claimed. ■

Lemma A.3 *We again use the Riccati Difference Equation (Equation A.4) and the assumptions given in Lemma A.2. If O_N is full column rank (i.e. has rank = n_x) then there exists a matrix, U, which is independent of M such that $M(N+1) \leq U$.*

Proof. Lemma A.2 gives the relation between $M(N+1)$ and M (Equation A.5). Multiplying out the quadratic term, this relation can be written as:

$$\begin{aligned} M(N+1) &= \left(\tilde{Q}_N - \tilde{R}_N^T \left[O_N M O_N^T + \tilde{V}_N\right]^{-1} \tilde{R}_N\right) \\ &\quad + A^{N+1}\left(M - M O_N^T \left[O_N M O_N^T + \tilde{V}_N\right]^{-1} O_N M\right)(A^T)^{N+1} \\ &\quad - 2\tilde{R}_N^T \left[O_N M O_N^T + \tilde{V}_N\right]^{-1} O_N M (A^T)^{N+1} \end{aligned} \tag{A.11}$$

We upper bound the terms on each line of this relation. Since $V, M > 0$, the first line can be upper bounded by $\tilde{Q}_N$. Next we upper bound the term on the second line. Let $O_N^\dagger$ be the left inverse of O_N. The left inverse exists since O_N is full column rank by assumption. Use $O_N^\dagger$ to obtain the following matrix equality:

$$\begin{aligned} & M - M O_N^T \left[O_N M O_N^T + \tilde{V}_N\right]^{-1} O_N M \\ &\quad = \left\{O_N^\dagger \left[O_N M O_N^T + \tilde{V}_N\right] - M O_N^T\right\} \left[O_N M O_N^T + \tilde{V}_N\right]^{-1} O_N M \\ &\quad = O_N^\dagger \tilde{V}_N \left[O_N M O_N^T + \tilde{V}_N\right]^{-1} O_N M \end{aligned} \tag{A.12}$$

The term on the second line of Equation A.11 can now be upper bounded using this matrix equality and properties of the maximum singular value:

$$\begin{aligned} & \bar{\sigma}\left(A^{N+1}\left(M - M O_N^T \left[O_N M O_N^T + \tilde{V}_N\right]^{-1} O_N M\right)(A^T)^{N+1}\right) \\ &\quad \overset{(a)}{\leq} (\bar{\sigma}(A))^{2N+2}\, \bar{\sigma}\left(O_N^\dagger \tilde{V}_N \left[O_N M O_N^T + \tilde{V}_N\right]^{-1} O_N M\right) \\ &\quad \overset{(b)}{\leq} (\bar{\sigma}(A))^{2N+2}\, \bar{\sigma}\left(O_N^\dagger \tilde{V}_N\right) \bar{\sigma}\left(\left[O_N M O_N^T + \tilde{V}_N\right]^{-1} O_N M O_N^T\right) \bar{\sigma}\left(\left(O_N^\dagger\right)^T\right) \\ &\quad \overset{(c)}{\leq} (\bar{\sigma}(A))^{2N+2}\, \bar{\sigma}\left(O_N^\dagger \tilde{V}_N\right) \bar{\sigma}\left(O_N^\dagger\right) \end{aligned}$$

Inequality (a) and (b) follow from the submultiplicative property of the maximum singular value: $\bar{\sigma}(AB) \leq \bar{\sigma}(A)\,\bar{\sigma}(B)$ [40]. If two symmetric matrices satisfy $A > B \geq 0$ then

$\bar{\sigma}\left(A^{-1}B\right) \leq 1$. Inequality (c) follows from this fact and $\bar{\sigma}\left(A^T\right) = \bar{\sigma}\left(A\right)$.

Finally we upper bound the term on the third line of Equation A.11. Again we apply properties of the maximum singular value:

$$\begin{aligned}
&\bar{\sigma}\left(-2\tilde{R}_N^T\left[O_N M O_N^T + \tilde{V}_N\right]^{-1} O_N M (A^T)^{N+1}\right) \\
&\qquad \leq \quad 2\bar{\sigma}\left(\tilde{R}_N^T\right)\bar{\sigma}\left(\left[O_N M O_N^T + \tilde{V}_N\right]^{-1} O_N M O_N^T\right)\bar{\sigma}\left((O_N^\dagger)^T (A^T)^{N+1}\right) \\
&\qquad \leq \quad 2\bar{\sigma}\left(\tilde{R}_N\right)\bar{\sigma}\left(A^{N+1} O_N^\dagger\right)
\end{aligned}$$

Combining these three upper bounds, we conclude that $M(N) \leq U$ where U is defined by:

$$U = \tilde{Q}_N + \left(\bar{\sigma}\left(A\right)\right)^{2N+2}\bar{\sigma}\left(O_N^\dagger \tilde{V}_N\right)\bar{\sigma}\left(O_N^\dagger\right)\cdot I_{n_x} + 2\bar{\sigma}\left(\tilde{R}_N\right)\bar{\sigma}\left(A^{N+1}O_N^\dagger\right)\cdot I_{n_x}$$

I_{n_x} is a $n_x \times n_x$ identity matrix. Again, this upper bound depends only on N, A, Q, V and $C(0)$,...,$C(N)$. In other words, $M(N)$ is bounded by a matrix which is independent of M. In terms of the Kalman filter, once the observability matrix is full column rank, we can guarantee that the error variance is bounded by a matrix that is independent of the size of the initial error variance. ■

Appendix B

Matrix Inequality for Second Moment Stability

Theorem B.1 *We are given a jump linear system with notation defined in Section 5.3. If $p_{ij} = p_j$ for all $i, j \in \mathcal{N}$ then the system is SMS if and only if there exists a matrix $G > 0$ such that:*

$$G - \sum_{j=1}^{N} p_j A_j^T G A_j > 0 \tag{B.1}$$

Proof. This proof outline is derived from [42] with minor modifications to exploit the property: $p_{ij} = p_j$.

($\Leftarrow$) First we show sufficiency of Equation B.1. Suppose there exists $G > 0$ such that Equation B.1 holds. Define the stochastic Lyapunov function:

$$V(x(k)) = x(k)^T G x(k) \tag{B.2}$$

Note that since Equation B.1 holds, there exists $\epsilon > 0$ such that $\sum_{j=1}^{N} p_j A_j^T G A_j - G < -\epsilon I$. Therefore,

$$\underset{\theta(k)}{E}\left[V(x(k+1)) \mid x(k)\right] - V(x(k)) = x(k)^T \left(\sum_{j=1}^{N} p_j A_j^T G A_j - G\right) x(k) < -\epsilon x(k)^T x(k)$$

where $\underset{\theta(k)}{E}[\cdot]$ denotes the expectaction taken over $\theta(k)$. This gives inequality (a) in the following relations:

$$\begin{aligned}\underset{\theta(k)}{E}\left[V(x(k+1)) \mid x(k)\right] &\overset{(a)}{<} V(x(k)) - \epsilon x(k)^T x(k) \\ &\overset{(b)}{\leq} \left[1 - \frac{\epsilon}{\lambda_{max}(G)}\right] V(x(k)) := \alpha V(x(k)) \end{aligned} \tag{B.3}$$

where we can choose ϵ sufficiently small so that $\alpha := 1 - \frac{\epsilon}{\lambda_{max}(G)} > 0$. Inequality (b) follows because $z^T G z \leq \lambda_{max}(G) z^T z$ for any vector z. We will show by induction that the following statements are true $\forall n \in \mathbb{N}$:

$$\mathcal{P}_n : \quad \underset{\theta(k),\ldots,\theta(k+n-1)}{E}\left[V(x(k+n)) \mid x(k)\right] < \alpha^n V(x(k)) \;\; \forall k \tag{B.4}$$

$\mathcal{P}_1$ follows directly from Equation B.3. Next make the induction assumption that $\mathcal{P}_n$ is true and show this implies that $\mathcal{P}_{n+1}$ holds as well.

$$\begin{aligned}&\underset{\theta(k),\ldots,\theta(k+n)}{E}\left[V(x(k+n+1)) \mid x(k)\right] \\ &\qquad = \sum_{j=1}^{N} p_j \underset{\theta(k+1),\ldots,\theta(k+n)}{E}\left[V(x(k+n+1)) \mid x(k+1) = A_j x(k)\right] \\ &\qquad \overset{(a)}{<} \sum_{j=1}^{N} \alpha^n p_j V(x(k+1))\big|_{x(k+1)=A_j x(k)} \\ &\qquad = \alpha^n \underset{\theta(k)}{E}\left[V(x(k+1)) \mid x(k)\right] \\ &\qquad \overset{(b)}{<} \alpha^{n+1} V(x(k))\end{aligned}$$

where inequality (a) is due to the induction assumption, inequality (b) is due to Equation B.3, and the equalities are simple consequences of conditional probabilities. By induction, we conclude that $\mathcal{P}_n$ holds $\forall n \in \mathbb{N}$. Therefore,

$$\underset{\theta(0),\ldots,\theta(N-1)}{E}\left[\sum_{k=0}^{N} V(x(k)) \;\middle|\; x(0)\right] < \left(1+\alpha+\ldots+\alpha^N\right) V(x(0)) = \frac{1-\alpha^N}{1-\alpha} V(x(0))$$

Taking the limit as N tends to infinity gives inequality (b) below:

$$\begin{aligned}
&\lim_{N\to\infty} \underset{\theta(0),\ldots,\theta(N-1)}{E}\left[\sum_{k=0}^{N} x(k)^T x(k) \mid x(0)\right] \\
&\qquad \overset{(a)}{\leq} \lim_{N\to\infty} \frac{1}{\lambda_{min}(G)} \underset{\theta(0),\ldots,\theta(N-1)}{E}\left[\sum_{k=0}^{N} x(k)^T G x(k) \mid x(0)\right] \\
&\qquad \overset{(b)}{\leq} \frac{1}{\lambda_{min}(G)} \frac{1}{1-\alpha} x(0)^T G x(0) \\
&\qquad \overset{(c)}{=} \frac{\lambda_{max}(G)}{\epsilon \lambda_{min}(G)} x(0)^T G x(0) < \infty
\end{aligned}$$

Inequality (a) follows because $\lambda_{min}(G) z^T z \leq z^T G z$ for any vector z. Equality (c) follows by definition of α and the final cost is finite because $G > 0$ implies $\lambda_{min}(G) > 0$. We conclude the system is stochastically stable and hence SMS.

($\Rightarrow$) Next we prove that Equation B.1 is actually a necessary condition for SMS. We will show that if the system is stochastically stable then for any symmetric matrix $W > 0$, there exists a matrix $G > 0$ such that:

$$G - \sum_{j=1}^{N} p_j A_j^T G A_j = W > 0 \tag{B.5}$$

To begin the proof, let $W > 0$ be any symmetric matrix. First we will define a sequence of matrices. Given $k \in \mathbb{N}$, it is easy to show that the following function is quadratic and

positive definite:

$$f(v) := \underset{\theta(0),\ldots,\theta(k-1)}{E}\left[\sum_{t=0}^{k} x(t)^T W x(t) \mid x(0) = v\right]$$

In other words, there exists a sequence of symmetric, positive definite matrices, $G(k)$, such that for any vector v:

$$v^T G(k) v = \underset{\theta(0),\ldots,\theta(k-1)}{E}\left[\sum_{t=0}^{k} x(t)^T W x(t) \mid x(0) = v\right] \tag{B.6}$$

Given any vector v, $W > 0$ implies that the sequence $\{v^T G(k) v\}$ is monotonically nondecreasing as a function of k. Since the system is SMS, this sequence is bounded from above. Hence $\lim_{k\to\infty} v^T G(k) v$ exists for any vector v. Let e_i denote the i^{th} basis vector and define $g_{ii} := \lim_{k\to\infty} e_i^T G(k) e_i$. For $i \neq j$, define $g_{ij} := \frac{1}{2}\left[\lim_{k\to\infty}(e_i + e_j)^T G(k)(e_i + e_j) - g_{ii} - g_{jj}\right]$. By the symmetry of $G(k)$, $g_{ij} = g_{ji}$. Using these definitions, we define the limit matrix G whose entries are given by g_{ij}. By definition, this matrix satisfies: $G = \lim_{k\to\infty} G(k)$. Furthermore, G is symmetric because $G(k)$ is symmetric for all k. Moreover, G is positive definite because $G(k)$ is uniformly positive definite for all k.

Next we show that this limit matrix, G, is indeed a solution to Equation B.5:

$$\begin{aligned}
v^T G(k) v &= \underset{\theta(0),\ldots,\theta(k-1)}{E}\left[\sum_{t=0}^{k} x(t)^T W x(t) \mid x(0) = v\right] \\
&= v^T W v + \underset{\theta(0),\ldots,\theta(k-1)}{E}\left[\sum_{t=1}^{k} x(t)^T W x(t) \mid x(0) = v\right] \\
&= v^T W v + \sum_{j=1}^{N} p_j \underset{\theta(1),\ldots,\theta(k-1)}{E}\left[\sum_{t=1}^{k} x(t)^T W x(t) \mid x(1) = A_j v\right] \\
&= v^T W v + \sum_{j=1}^{N} p_j \underset{\theta(0),\ldots,\theta(k-2)}{E}\left[\sum_{t=0}^{k-1} x(t)^T W x(t) \mid x(0) = A_j v\right] \\
&= v^T W v + \sum_{j=1}^{N} p_j v^T A_j^T G(k-1) A_j v
\end{aligned}$$

The relations above follow from the definition of $G(k)$ and conditional probabilities. The end-to-end equality implies:

$$v^T \left[G(k) - \sum_{j=1}^{N} p_j A_j^T G(k-1) A_j - W \right] v = 0 \tag{B.7}$$

It is simple to show that if $Q = Q^T$ and $v^T Q v = 0$ for all vectors v, then $Q = 0$. Thus Equation B.7 implies:

$$G(k) - \sum_{j=1}^{N} p_j A_j^T G(k-1) A_j = W$$

Taking limits as $k \to \infty$, the argument above guarantees there exists a symmetric matrix $G > 0$ satisfying:

$$G - \sum_{j=1}^{N} p_j A_j^T G A_j = W > 0$$

which proves the necessity of Equation B.1. ■

Appendix C

Matrix Inequality for H_∞ Performance

First we give a brief outline of the proof. Sufficiency of the matrix condition follows by a stochastic Lyapunov argument similar to that used in the proof of Theorem 5.2. For necessity, we show that $\|\mathcal{S}\|_\infty < \gamma$ implies that a related Riccati equation has a solution. The Riccati equality can be turned in to a Riccati inequality by a perturbation argument which leads to the H_∞ matrix condition. This proof uses ideas from [67] which gives a dynamic game interpretation to the continuous time H_∞-control of jump linear systems. Reference [3] gives further information on generalized Riccati equations related to dynamic games.

Before proving the theorem, we need to introduce some notation and prove several lemmas. First define the cost function:

$$J(w, x_0) := \underset{\theta(0),\ldots}{E}\left[\sum_{k=0}^{\infty} z^T(k)z(k) - \gamma^2 w^T(k)w(k)\right] = \|z\|_{2E}^2 - \gamma^2\|w\|_2^2 \qquad \text{(C.1)}$$

where $w(k)$ and $z(k)$ are related by the jump system, $\mathcal{S}$, defined in Equation 5.1 with the initial condition $x(0) = x_0$. The expectation is taken over the jump process, $\{\theta(0), \ldots\}$. If $\|\mathcal{S}\|_\infty \leq \gamma$ then $J(w, 0) \leq 0$ for any $w \in \ell_2$ and $J(w, 0) = 0$ for $w = 0$. Conversely, if $\|\mathcal{S}\|_\infty > \gamma$ then there exists $w \in \ell_2$ such that $J(w, 0) > 0$. By scaling up this disturbance, we can make the cost arbitrarily large. Explicitly, choose $w_\alpha = \alpha w$ for $\alpha \in \mathbb{R}$ and by the linearity of the plant, $J(w_\alpha, 0) = \alpha^2 J(w, 0)$. This cost can be made arbitrarily large by letting $\alpha \to \infty$. If we define $J^* := \sup_{w \in \ell_2} J(w, 0)$, then from these arguments we conclude:

$$J^* = \begin{cases} 0 & \text{if } \|S\|_\infty \leq \gamma \\ +\infty & \text{if } \|S\|_\infty > \gamma \end{cases} \tag{C.2}$$

Given $T \geq 0$, define $G_T(k)$ for $0 \leq k \leq T$ by the following Riccati Difference Equation with the terminal condition $G_T(T) = 0$.

$$G_T(k) = \sum_{j=1}^{N} p_j \left(A_j^T G_T(k+1) A_j + C_j^T C_j \right) \tag{C.3}$$
$$+ \left[\sum_{j=1}^{N} p_j \left(B_j^T G_T(k+1) A_j + D_j^T C_j \right) \right]^T V_T^{-1}(k+1) \left[\sum_{j=1}^{N} p_j \left(B_j^T G_T(k+1) A_j + D_j^T C_j \right) \right]$$

where: $V_T(k+1) = \gamma^2 I - \sum_{j=1}^{N} p_j \left(B_j^T G_T(k+1) B_j + D_j^T D_j \right)$. Note that this recursion is well defined only if $V_T(k)$ is nonsingular for $1 \leq k \leq T$. In the deterministic LQ problem, a Riccati Difference Equation is used to generate optimal control inputs. Similarly, we use Equation 4.20 to generate optimal inputs (disturbances) that maximize $J(w, 0)$.

We now present several lemmas that will be used in the proof of the theorem. The first is a technical lemma concerning Riccati Difference Equations. The second lemma constructs a 'bad' disturbance from the solution of the Riccati Difference Equation. The remaining lemmas use this 'bad' disturbance to show that $\|S\|_\infty >$ or $\geq \gamma$ under various

conditions on the Riccati Difference Equation. The results are used to prove the necessity of matrix inequality. For the remainder of this appendix, assume $p_{ij} = p_j$ for all $i, j \in \mathcal{N}$ and the system, $\mathcal{S}$, given by Equation 5.1 is weakly controllable. The reader is encouraged to read the proof of Theorem C.1 prior to reading the lemmas.

Lemma C.1 *Given $T \geq 0$, let $G_i(k)$ $(i = 1, 2)$ be a solution of the following Riccati Difference Equation for $0 \leq k \leq T$:*

$$G_i(k) = \sum_{j=1}^{N} p_j \left(A_j^T G_i(k+1) A_j + C_j^T C_j\right) \tag{C.4}$$
$$+ \left[\sum_{j=1}^{N} p_j \left(B_j^T G_i(k+1) A_j + D_j^T C_j\right)\right]^T V_i^{-1}(k+1) \left[\sum_{j=1}^{N} p_j \left(B_j^T G_i(k+1) A_j + D_j^T C_j\right)\right]$$

where R_i is a symmetric matrix and $V_i(k+1) = R_i - \sum_{j=1}^{N} p_j \left(B_j^T G_i(k+1) B_j + D_j^T D_j\right)$. The Riccati Difference Equations have terminal conditions $G_1(T) \geq 0$ and $G_2(T) \geq 0$. Define $\tilde{G}(k) := G_2(k) - G_1(k)$ and $\tilde{R} := R_2 - R_1$. Then:

$$\tilde{G}(k) = \sum_{j=1}^{N} p_j \bar{A}_j^T(k+1) \tilde{G}(k+1) \bar{A}_j(k+1) - K^T(k+1) \tilde{R} K(k+1)$$
$$\left[-\tilde{R} K(k+1) + \sum_{j=1}^{N} p_j B_j^T \tilde{G}(k+1) \bar{A}_j(k+1)\right]^T V_2^{-1}(k+1) \cdot$$
$$\left[-\tilde{R} K(k+1) + \sum_{j=1}^{N} p_j B_j^T \tilde{G}(k+1) \bar{A}_j(k+1)\right]$$

where $K(k+1) := V_1^{-1}(k+1) \left[\sum_{j=1}^{N} p_j \left(B_j^T G_1(k+1) A_j + D_j^T C_j\right)\right]$ and $\bar{A}_j(k+1) := A_j + B_j K(k+1)$.

Proof. The proof is a simple, albeit algebraically intensive, extension of a result by C. de Souza (Lemma 3.1 in [25]). We now provide a sketch of the proof. For notational ease, we

make the following defintion for $i = 1, 2$:

$$X_i(k) := \sum_{j=1}^{N} p_j \left(B_j^T G_i(k) A_j + D_j^T C_j \right)$$

If we use this notation and Equation C.4, then we obtain a recursion for $\tilde{G}(k)$:

$$\begin{aligned} \tilde{G}(k) = & \sum_{j=1}^{N} p_j \left(A_j^T \tilde{G}(k+1) A_j \right) \\ & + X_2^T(k+1) V_2^{-1}(k+1) X_2(k+1) - X_1^T(k+1) V_2^{-1}(k+1) X_1(k+1) \\ & + X_1^T(k+1) V_2^{-1}(k+1) X_1(k+1) - X_1^T(k+1) V_1^{-1}(k+1) X_1(k+1) \end{aligned} \tag{C.5}$$

Note that we have added and subtracted the term $X_1^T(k+1)V_2^{-1}(k+1)X_1(k+1)$ to obtain this equality. Next we find some useful relations for $V_2^{-1}(k)$ and $V_1^{-1}(k)$. First define $\tilde{V}(k) := V_2(k) - V_1(k)$. We can apply the Matrix Inversion Lemma [2] to obtain equality (a) below:

$$V_2^{-1}(k) = \left[V_1(k) + \tilde{V}(k) \right]^{-1} \stackrel{(a)}{=} V_1^{-1}(k) - V_1^{-1}(k) \tilde{V}(k) \left[I + V_1^{-1}(k) \tilde{V}(k) \right]^{-1} V_1^{-1}(k)$$

Equation C.6 below follows from this relation and simple algebra. Equation C.7 follows from Equation C.6 because $V_i(k)$ is symmetric for $i = 1, 2$.

$$V_2^{-1}(k) = V_1^{-1}(k) \left[I - \tilde{V}(k) V_2^{-1}(k) \right] \tag{C.6}$$

$$V_2^{-1}(k) = \left[I - V_2^{-1}(k) \tilde{V}(k) \right] V_1^{-1}(k) \tag{C.7}$$

Next we use these relations to simplify several terms in Equation C.5. We start with the terms on the second line of Equation C.5. For notational ease, we supress the functional

dependence on $k+1$:

$$\begin{aligned} & X_2^T V_2^{-1} X_2 - X_1^T V_2^{-1} X_1 \\ & \quad \stackrel{(a)}{=} [X_2 - X_1]^T V_2^{-1} [X_2 - X_1] + [X_2 - X_1]^T V_2^{-1} X_1 + X_1^T V_2^{-1} [X_2 - X_1] \\ & \quad \stackrel{(b)}{=} [X_2 - X_1]^T V_2^{-1} [X_2 - X_1] + [X_2 - X_1]^T [I - V_2^{-1}\tilde{V}] K + K^T [I - \tilde{V} V_2^{-1}] [X_2 - X_1] \end{aligned} \tag{C.8}$$

where $K(k+1) = V_1^{-1}(k+1) X_1(k+1)$ as defined in the statement of the lemma. Equality (a) follows by completing the square. Equality (b) follows from the the definition of $K(k+1)$. and by applying Equations C.6-C.7. Next we simplify the terms on the third line of Equation C.5:

$$\begin{aligned} X_1^T V_2^{-1} X_1 - X_1^T V_1^{-1} X_1 & \stackrel{(a)}{=} -X_1^T V_1^{-1} X_1 + X_1^T V_1^{-1} [I - \tilde{V} V_2^{-1}] X_1 \\ & = -X_1^T V_1^{-1} \tilde{V} V_2^{-1} X_1 \\ & \stackrel{(b)}{=} -X_1^T V_1^{-1} \tilde{V} [V_1^{-1} - V_2^{-1} \tilde{V} V_1^{-1}] X_1 \\ & \stackrel{(c)}{=} -K^T \tilde{V} K + K^T \tilde{V} V_2^{-1} \tilde{V} K \end{aligned} \tag{C.9}$$

Equality (a) follows from Equation C.6 while equality (b) follows from Equation C.7. Equality (c) uses the definition of $K(k+1)$.

Now we can use Equations C.8 and C.9 to replace the terms in Equation C.5. This yields:

$$\begin{aligned} \tilde{G}(k) = & \sum_{j=1}^{N} p_j \left(A_j^T \tilde{G}(k+1) A_j \right) + [X_2 - X_1]^T V_2^{-1} [X_2 - X_1] \\ & + [X_2 - X_1]^T [I - V_2^{-1}\tilde{V}] K + K^T [I - \tilde{V} V_2^{-1}] [X_2 - X_1] - K^T \tilde{V} K + K^T \tilde{V} V_2^{-1} \tilde{V} K \end{aligned}$$

Again we have supressed the functional dependence on $k+1$ for most terms on the right

side of this relation. This can be rewritten as:

$$\tilde{G}(k) = \left[\sum_{j=1}^{N} p_j \left(A_j^T \tilde{G}(k+1) A_j\right) - K^T \tilde{V} K + [X_2 - X_1]^T K + K[X_2 - X_1]\right]$$
$$+ [X_2 - X_1 - \tilde{V}K]^T V_2^{-1} [X_2 - X_1 - \tilde{V}K] \tag{C.10}$$

Finally, we note that $\tilde{V}(k+1) = \tilde{R} - \sum_{j=1}^{N} p_j \left(B_j^T \tilde{G}(k+1) B_j\right)$ and $X_2(k+1) - X_1(k+1) = \sum_{j=1}^{N} p_j \left(B_j^T \tilde{G}(k+1) A_j\right)$. These relations allow us to finally reduce Equation C.10 to the form stated in the lemma:

$$\tilde{G}(k) = \sum_{j=1}^{N} p_j \bar{A}_j^T(k+1) \tilde{G}(k+1) \bar{A}_j(k+1) - K^T(k+1) \tilde{R} K(k+1)$$
$$\left[-\tilde{R}K(k+1) + \sum_{j=1}^{N} p_j B_j^T \tilde{G}(k+1) \bar{A}_j(k+1)\right]^T V_2^{-1}(k+1) \cdot$$
$$\left[-\tilde{R}K(k+1) + \sum_{j=1}^{N} p_j B_j^T \tilde{G}(k+1) \bar{A}_j(k+1)\right]$$

■

Lemma C.2 *Let $G_T(k)$ be the solution to the Riccati Difference Equation (C.3) with terminal condition $G_T(T) = 0$. Assume $V_T(k) > 0$ for $1 \leq k \leq T$ and define the following plant disturbance:*

$$\bar{w}(k) = \begin{cases} V_T(k+1)^{-1} \sum_{j=1}^{N} p_j \left(B_j^T G_T(k+1) A_j + D_j^T C_j\right) x(k) & 0 \leq k \leq T-1 \\ 0 & else \end{cases} \tag{C.11}$$

Then, $J(\bar{w}, x_0) \geq x_0^T G_T(0) x_0$.

Proof. First we state the trivial relation:

$$0 = x_0^T G_T(0) x_0 + \underset{\theta(0),\ldots,\theta(T-1)}{E} \left[\sum_{k=0}^{T-1} x^T(k+1) G_T(k+1) x(k+1) - x^T(k) G_T(k) x(k)\right]$$

Add this net zero quantity to $J(w, x_0)$ and use the system dynamics to substitute for $z(k)$ and $x(k+1)$:

$$J(w,x_0) = x_0^T G_T(0) x_0 + \underset{\theta(0),\ldots}{E}\left[\sum_{k=T}^{\infty} z^T(k) z(k)\right] + \sum_{k=0}^{T-1} \underset{\theta(0),\ldots,\theta(k)}{E}[*]$$
$$\geq x_0^T G_T(0) x_0 + \sum_{k=0}^{T-1} \underset{\theta(0),\ldots,\theta(k)}{E}[*] \tag{C.12}$$

where:

$$* = (C_{\theta(k)} x(k) + D_{\theta(k)} w(k))^T (C_{\theta(k)} x(k) + D_{\theta(k)} w(k)) - \gamma^2 w^T(k) w(k)$$
$$+ (A_{\theta(k)} x(k) + B_{\theta(k)} w(k))^T G_T(k+1)(A_{\theta(k)} x(k) + B_{\theta(k)} w(k)) - x^T(k) G_T(k) x(k)$$

Next, we use the assumption that $p_{ij} = p_j$. In other words, $\theta(k)$ is independent of $\theta(j)$ for $j < k$. Thus $\theta(k)$ is independent of $x(k)$ and $w(k)$ and we obtain:

$$\underset{\theta(0),\ldots,\theta(k)}{E}[*] \overset{(a)}{=} \sum_{j=1}^{N} p_j \underset{\theta(0),\ldots,\theta(k-1)}{E}\Big[(C_j x(k) + D_j w(k))^T (C_j x(k) + D_j w(k)) - \gamma^2 w^T(k) w(k)$$
$$+ (A_j x(k) + B_j w(k))^T G_T(k+1)(A_j x(k) + B_j w(k)) - x^T(k) G_T(k) x(k)\Big]$$
$$\overset{(b)}{=} - \underset{\theta(0),\ldots,\theta(k-1)}{E}\left[\left(w(k) - V_T^{-1}(k+1)\sum_{j=1}^{N} p_j \left(B_j^T G_T(k+1) A_j + D_j^T C_j\right) x(k)\right)^T\right.$$
$$\left. V_T(k+1)\left(w(k) - V_T^{-1}(k+1)\sum_{j=1}^{N} p_j \left(B_j^T G_T(k+1) A_j + D_j^T C_j\right) x(k)\right)\right]$$

As discussed above, equality (a) uses the independence of $\theta(k)$ and $(x(k), w(k))$. Equality (b) uses the Riccati Difference Equation (C.3) to substitute for $G_T(k)$ followed by a completion of the square. The lemma follows from Equation C.12 by noting that the choice of $\bar{w}(k)$ in Equation C.11 leads to $\underset{\theta(0),\ldots,\theta(k)}{E}[*] = 0$ for $0 \leq k \leq T-1$. ■

Lemma C.3 *Let $G_T(k)$ be a solution of the Riccati Difference Equation (C.3) with terminal condition $G_T(T) = 0$. If there exists $T \geq 0$ such that $V_T(k) > 0$ for $1 \leq k \leq T$ and $V_T(0)$ has a negative eigenvalue, then $\|S\|_\infty > \gamma$.*

Proof. Let r be the eigenvector, normalized to $\|r\| = 1$, and $\lambda < 0$ be the eigenvalue of $V_T(0)$. Note that $V_T(0)$ is symmetric so its eigenvalues are real and writing $\lambda < 0$ is sensible. For $\alpha \in \mathbb{R}$, define the disturbance:

$$w_\alpha(k) = \begin{cases} \alpha r & k = 0 \\ V_T(k+1)^{-1} \left[\sum_{j=1}^N p_j \left(B_j^T G_T(k+1) A_j + D_j^T C_j\right)\right] x(k) & 1 \leq k \leq T-1 \\ 0 & else \end{cases}$$

Applying this disturbance to the system with $x_0 = 0$ gives a lower bound on J^*:

$$\begin{aligned} J^* &:= \sup_{w \in \ell_2} J(w, 0) \geq J(w_\alpha, 0) \\ &= \mathop{E}_{\theta(0),\ldots} \left[w_\alpha^T(0) \left(D_{\theta(0)}^T D_{\theta(0)} - \gamma^2 I\right) w_\alpha(0) + \sum_{k=1}^{\infty} \left(z^T(k) z(k) - \gamma^2 w_\alpha^T(k) w_\alpha(k)\right) \right] \quad \text{(C.13)} \\ &= \alpha^2 \sum_{j=1}^N p_j r^T \left(D_j^T D_j - \gamma^2 I\right) r + \sum_{j=1}^N p_j J(w_\alpha, \alpha B_j r) \end{aligned}$$

The last equality follows from the independence of $\theta(k)$ and a slight abuse of notation concerning $J(\cdot, \cdot)$. By $J(w_\alpha, \alpha B_j r)$, we mean the cost function with the system starting at $\alpha B_j r$ and applying $w_\alpha(k)$ for $k \geq 1$. Apply Lemma C.2 to conclude $J(w_\alpha, \alpha B_j r) \geq \alpha^2 r^T B_j^T G_T(1) B_j r$. Substituting this result into Equation C.13 gives:

$$\begin{aligned} J^* &\geq -\alpha^2 r^T \left[\gamma^2 I - \sum_{j=1}^N p_j \left(B_j^T G_T(1) B_j + D_j^T D_j\right)\right] r \\ &= -\alpha^2 r^T V_T(0) r = (-\lambda)\alpha^2 > 0 \end{aligned}$$

This holds $\forall \alpha \in \mathbb{R}$. Take $\alpha \to \infty$ to show that $J^* = +\infty$. By Equation C.2, we conclude $\|S\|_\infty > \gamma$. ■

Lemma C.4 *Let $G_T(k)$ be a solution of the Riccati Difference Equation (C.3) with terminal condition $G_T(T) = 0$. If there exists $T \geq 0$ such that $V_T(k) > 0$ for $1 \leq k \leq T$ and $V_T(0)$ has an eigenvalue at zero, then $\|S\|_\infty \geq \gamma$.*

Proof. Using the notation of Lemma C.1, let $G_1(k)$ denote the solution of the Riccati Difference Equation (C.4) with $R_1 = \gamma^2 I$ and the terminal condition $G_1(T) = 0$. By assumption, $V_1(k) > 0$ for $1 \leq k \leq T$ and $V_1(0)$ has an eigenvalue at zero. Given $\epsilon > 0$, define $G_2(k)$ as the solution with $R_2 = (\gamma^2 - \epsilon)I$ and the terminal condition $G_2(T) = 0$. Thus $\tilde{G}(T) = 0$ and $\tilde{R} = -\epsilon I < 0$. If $\epsilon > 0$ is sufficiently small, then $V_2(k) > 0$ for $1 \leq k \leq T$. We can then use Lemma C.1 and induction to show that $\tilde{G}(k) \geq 0$ for $0 \leq k \leq T$. From $R_2 < R_1$ and $G_2(0) \geq G_1(0)$ it follows that $V_2(0) < V_1(0)$ and hence $V_2(0)$ has a negative eigenvalue. By Lemma C.3, $\|S\|_\infty > \gamma - \epsilon$. Since this holds for all $\epsilon > 0$ which are sufficiently small, we conclude that $\|S\|_\infty \geq \gamma$. ■

Lemma C.5 *Let $G_T(k)$ be a solution of the Riccati Difference Equation (C.3) with terminal condition $G_T(T) = 0$. Assume that for all $T \geq 0$, $V_T(k) > 0$ for $0 \leq k \leq T$. Also assume that the matrix sequence $G_T(0)$ is unbounded as $T \to \infty$. That is, there exists a sequence $\{T_j\}_{j=0}^{\infty}$ such that $\lambda_{max}(G_{T_j}(0)) \to \infty$ as $j \to \infty$. Then $\|S\|_\infty > \gamma$.*

Proof.

By assumption, there exists a sequence $\{T_j\}_{j=0}^{\infty}$ such that $\lambda_{max}(G_{T_j}(0)) \to \infty$ as $j \to \infty$. For each j, let r_j be the eigenvector associated with $\lambda_{max}(G_{T_j}(0))$ normalized to

$\|r_j\| = 1$. Then there exists an $r^* \in \mathbb{R}^{n_x}$ and a subsequence j_l such that $\lim_{j_l \to \infty} r_{j_l} = r^*$. Furthermore, $\lim_{j_l \to \infty} (r^*)^T G_{T_{j_l}} r^* = \infty$. To ease some of the notation below, we now refer to this subsequence as (G_{T_l}) with $(r^*)^T G_{T_l} r^* \to \infty$.

Now apply the assumption of weak controllability: There exists a time, T_c, and an input, $w_c(k)$, defined on $0 \leq k \leq T_c - 1$ such that $\Pr[x(T_c) = r^*] > 0$ when the system starts from the origin. We now construct a disturbance which can make the cost function arbitrarily large:

$$w_l(k) = \begin{cases} w_c(k) & 0 \leq k \leq T_c - 1 \\ V_{T_l}(k - T_c + 1)^{-1} \left[\sum_{j=1}^{N} p_j \left(B_j^T G_{T_l}(k - T_c + 1) A_j + D_j^T C_j \right) \right] x(k) & \\ & T_c \leq k \leq T_c + T_l - 1 \\ 0 & else \end{cases}$$

The first portion of the disturbance attempts to move the state from the origin to r^*. By construction, $x(T_c) = r^*$ with some positive probability and the second portion of the disturbance is able to make the cost function arbitrarily large. Mathematically, this argument is:

$$\begin{aligned} J^* \geq J(w_l, 0) &= \underset{\theta(0),\ldots,\theta(T_c-2)}{E} \left[\sum_{k=0}^{T_c-1} z^T(k) z(k) - \gamma^2 w^T(k) w(k) \right] \\ &\quad + \underset{\theta(0),\ldots}{E} \left[\sum_{k=T_c}^{\infty} z^T(k) z(k) - \gamma^2 w^T(k) w(k) \right] \\ &\overset{(a)}{\geq} \underset{\theta(0),\ldots,\theta(T_c-2)}{E} \left[\sum_{k=0}^{T_c-1} z^T(k) z(k) - \gamma^2 w^T(k) w(k) \right] + \Pr[x(T_c) = r^*] \cdot (r^*)^T G_{T_l} r^* \end{aligned}$$

Inequality (a) follows by the construction of $w_c(k)$ and by Lemma C.2. The first term on the second line is a fixed cost for all l. The second term can be made arbitrarily large as $l \to \infty$ by construction and thus $J^* = \infty$. By Equation C.2, we conclude $\|S\|_\infty > \gamma$. ■

We are now ready to prove the theorem using these lemmas.

Theorem C.1 *Assume $p_{ij} = p_j$ for all $i, j \in \mathcal{N}$ and the system, $\mathcal{S}$, given by Equation 5.1 is weakly controllable. $\mathcal{S}$ is SMS and satisfies $\|\mathcal{S}\|_\infty < \gamma$ if and only if there exists a symmetric matrix $G > 0$ satisfying the following matrix inequality:*

$$\begin{bmatrix} G & 0 \\ 0 & \gamma^2 I \end{bmatrix} - \sum_{j=1}^{N} p_j \begin{bmatrix} A_j & B_j \\ C_j & D_j \end{bmatrix}^T \begin{bmatrix} G & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A_j & B_j \\ C_j & D_j \end{bmatrix} > 0 \tag{C.14}$$

Proof.

($\Leftarrow$) First we show sufficiency of Equation C.14. Equation C.14 implies that the upper left block must also be positive definite. Thus:

$$G - \sum_{j=1}^{N} p_j A_j^T G A_j > \sum_{j=1}^{N} p_j C_j^T C_j \geq 0$$

By Theorem 5.2, we conclude that the system is SMS. Next define the function $V(x) := x^T G x$. Assuming zero initial conditions, $V(x(0)) = 0$ and hence:

$$\underset{\theta(0),\ldots,\theta(M)}{E} \left[\sum_{k=0}^{M} V(x(k+1)) - V(x(k)) \right] = \underset{\theta(0),\ldots,\theta(M)}{E} [V(x(M+1))] \geq 0 \tag{C.15}$$

Equation C.15 is used to show inequality (b) below. Definition (a) given below is the norm of a sequence truncated at time M.

$$\begin{aligned} \|z\|_M^2 &\overset{(a)}{:=} \underset{\theta(0),\ldots,\theta(M-1)}{E} \left[\sum_{k=0}^{M} z(k)^T z(k) \right] \\ &\overset{(b)}{\leq} \gamma^2 \|w\|_M^2 + \underset{\theta(0),\ldots,\theta(M)}{E} \left[\sum_{k=0}^{M} \left(z(k)^T z(k) - \gamma^2 w(k)^T w(k) \right) + \sum_{k=0}^{M} V(x(k+1)) - V(x(k)) \right] \\ &\overset{(c)}{=} \gamma^2 \|w\|_M^2 - \sum_{k=0}^{M} \underset{\theta(0),\ldots,\theta(k)}{E} \left[\begin{bmatrix} x(k)^T & w(k)^T \end{bmatrix} R_{\theta(k)} \begin{bmatrix} x(k) \\ w(k) \end{bmatrix} \right] \end{aligned}$$

Equality (c) follows by using the system dynamics to replace $z(k)$ and $x(k+1)$ in terms of $x(k)$ and $w(k)$. $R_{\theta(k)}$ is defined as:

$$R_{\theta(k)} := \begin{bmatrix} G & 0 \\ 0 & \gamma^2 I \end{bmatrix} - \begin{bmatrix} A_{\theta(k)} & B_{\theta(k)} \\ C_{\theta(k)} & D_{\theta(k)} \end{bmatrix}^T \begin{bmatrix} G & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{\theta(k)} & B_{\theta(k)} \\ C_{\theta(k)} & D_{\theta(k)} \end{bmatrix}$$

Finally, we use the assumption that $p_{ij} = p_j$. In other words, $\theta(k)$ is independent of $\theta(j)$ for $j < k$. Thus $\theta(k)$ is independent of $x(k)$ and $w(k)$ and we obtain:

$$\|z\|_M^2 \leq \gamma^2 \|w\|_M^2 - \sum_{k=0}^{M} \underset{\theta(0),\ldots,\theta(k-1)}{E} \left[\begin{bmatrix} x(k)^T & w(k)^T \end{bmatrix} \left(\sum_{j=1}^{N} p_j R_j \right) \begin{bmatrix} x(k) \\ w(k) \end{bmatrix} \right]$$

$\sum_{j=1}^{N} p_j R_j$ is in fact the left side of Equation C.14. Since this term is positive definite and $w(k) \neq 0$ for some k, there exists an $\epsilon > 0$ such that $\|z\|_M^2 < \gamma^2 \|w\|_M^2 - \epsilon$. Taking limits as $M \to \infty$ gives $\|z\|_{2E}^2 \leq \gamma^2 \|w\|_2^2 - \epsilon$ and hence for any $w \in \ell_2$, $\|z\|_{2E} < \gamma \|w\|_2$. To conclude, if there exists $G > 0$ satisfying Equation C.14, then the system is SMS and $\|\mathcal{S}\|_\infty < \gamma$.

($\Rightarrow$) Next we prove that Equation C.14 is actually a necessary condition. First we show that if $\|\mathcal{S}\|_\infty < \gamma$ then $\exists G \geq 0$ satisfing the following Riccati Equation:

$$\begin{aligned} G = & \sum_{j=1}^{N} p_j \left(A_j^T G A_j + C_j^T C_j \right) \\ & + \left[\sum_{j=1}^{N} p_j \left(B_j^T G A_j + D_j^T C_j \right) \right]^T V^{-1} \left[\sum_{j=1}^{N} p_j \left(B_j^T G A_j + D_j^T C_j \right) \right] \end{aligned} \tag{C.16}$$

where: $V = \gamma^2 I - \sum_{j=1}^{N} p_j \left(B_j^T G B_j + D_j^T D_j \right) > 0$.

Consider solutions, $G_T(k)$, to the Riccati Difference Equation (C.3) with terminal condition $G_T(T) = 0$. There are three possible cases:

Case 1: $\exists T \geq 0$ such that $V_T(0)$ is not well defined.

Case 2: $V_T(0)$ is well defined $\forall T \geq 0$ and $\exists T \geq 0$ such that $V_T(0) \not> 0$.
Case 3: $V_T(0)$ is well defined $\forall T \geq 0$ and $\forall T \geq 0$, $V_T(0) > 0$.

Let us discuss these cases. First, notice that the Riccati Difference Equation solutions satisfy the following time-shifting property: $G_{T-l}(k-l) = G_T(k)$ and $V_{T-l}(k-l) = V_T(k)$ for any integers l, k, T satisfying $0 \leq l \leq k \leq T$. Second, we recall that the Riccati Difference Equation is well defined for $0 \leq k \leq T$ only if $V_T(k)$ is nonsingular for $1 \leq k \leq T$. Consider Case 1 and let $T_{min} \geq 0$ be the smallest T such that $V_T(0)$ is not well defined. We can apply the time-shifting property to conclude that $V_{T_{min}}(k)$ is nonsingular for $1 \leq k \leq T_{min}$ and $V_{T_{min}}(0)$ has an eigenvalue at 0 (otherwise T_{min} would not be the minimum T). Therefore, we can rewrite Case 1 as:

Case 1: There exists $T \geq 0$ such that $V_T(k)$ is nonsingular for $1 \leq k \leq T$ and $V_T(0)$ has an eigenvalue at zero.

Similar applications of the time-shifting property allow us to equivalently rewrite the three cases given above as follows:

Case 1: There exists $T \geq 0$ such that $V_T(k) > 0$ for $1 \leq k \leq T$ and $V_T(0)$ has a negative eigenvalue.
Case 2: There exists $T \geq 0$ such that $V_T(k) > 0$ for $1 \leq k \leq T$ and $V_T(0)$ has an eigenvalue at zero.
Case 3(a): For all $T \geq 0$, $V_T(k) > 0$ for $0 \leq k \leq T$. Moreover, the matrix sequence $G_T(0)$ is unbounded.
Case 3(b): For all $T \geq 0$, $V_T(k) > 0$ for $0 \leq k \leq T$. Moreover, the matrix sequence $G_T(0)$ is bounded.

For Case 1, it follows from Lemma C.3 that $\|S\|_\infty > \gamma$. For Case 2, it follows from Lemma C.4 that $\|S\|_\infty \geq \gamma$. For Case 3(a), it follows from Lemma C.5 that $\|S\|_\infty > \gamma$. By the contraposition, $\|S\|_\infty < \gamma$ implies Case 3(b). We apply Case 3(b) to prove $\exists G \geq 0$ that satisfies the Riccati Equation (C.16).

First we show that the matrix sequence $G_T(0)$ is monotonically nondecreasing in T. If $V_T(k+1) > 0$ and $G_T(k+1) \geq 0$, then it is clear from the Riccati Difference Equation (C.3) that $G_T(k) \geq 0$. It follows from the terminal condition $G_T(T) = 0$ and induction that $G_T(k) \geq 0$ for $0 \leq k \leq T$. In particular, for any $T_2 \leq T$, $G_T(T_2) \geq 0$. Define a second solution to the Riccati Difference Equation, $G_{T_2}(k)$ for $0 \leq k \leq T_2$ with the terminal condition $G_{T_2}(T_2) = 0$. Define the difference $\tilde{G}(k) = G_T(k) - G_{T_2}(k)$ for $0 \leq k \leq T_2$ and note that $\tilde{G}(T_2) = G_T(T_2) - G_{T_2}(T_2) \geq 0$. Apply Lemma C.1 and induction to show $\tilde{G}(k) \geq 0$ for all $0 \leq k \leq T_2$. In particular, $T \geq T_2$ implies $G_T(0) \geq G_{T_2}(0) \geq 0$. Thus $G_T(0)$ is a bounded, monotonic matrix sequence. Consequently this sequence must have a limit, $G \geq 0$, and this limit matrix satisfies the Riccati Equation (C.16). Since $V_T(0) = \gamma^2 I - \sum_{j=1}^{N} p_j \left(B_j^T G_T(0) B_j + D_j^T D_j \right)$, it also has a well defined limit as $T \to \infty$ which we denote by V. We can show that $V \geq 0$ since $V_T(0) > 0$ for all T. Using an argument similar to the proof of Lemma C.4, we can show this inequality is strict, $V > 0$.

Define the perturbed plant, $\mathcal{S}_\epsilon$:

$$\begin{bmatrix} x(k+1) \\ z(k) \end{bmatrix} = \begin{bmatrix} A_{\theta(k)} & B_{\theta(k)} \\ C^\epsilon_{\theta(k)} & D^\epsilon_{\theta(k)} \end{bmatrix} \begin{bmatrix} x(k) \\ w(k) \end{bmatrix} \tag{C.17}$$

where the output equation matrices are given by:

$$C^\epsilon_{\theta(k)} := \begin{bmatrix} C_{\theta(k)} \\ \epsilon I_{n_x \times n_x} \end{bmatrix} \qquad D^\epsilon_{\theta(k)} := \begin{bmatrix} D_{\theta(k)} \\ 0_{n_x \times n_w} \end{bmatrix}$$

If a plant is SMS and has $\|\mathcal{S}\|_\infty < \gamma$ then $\exists \epsilon > 0$ such that $\mathcal{S}_\epsilon$ is also SMS and has

$\|\mathcal{S}_\epsilon\|_\infty < \gamma$. By the argument above, $\exists G_\epsilon \geq 0$ such that:

$$\begin{aligned} G_\epsilon &= \sum_{j=1}^{N} p_j \left(A_j^T G_\epsilon A_j + (C_j^\epsilon)^T C_j^\epsilon\right) + \\ &\quad \left[\sum_{j=1}^{N} p_j \left(B_j^T G_\epsilon A_j + (D_j^\epsilon)^T C_j^\epsilon\right)\right]^T V_\epsilon^{-1} \left[\sum_{j=1}^{N} p_j \left(B_j^T G_\epsilon A_j + (D_j^\epsilon)^T C_j^\epsilon\right)\right] \end{aligned} \tag{C.18}$$

where: $V_\epsilon = \gamma^2 I - \sum_{j=1}^{N} p_j \left(B_j^T G_\epsilon B_j + (D_j^\epsilon)^T D_j^\epsilon\right) > 0$. After multiplying out all the matrices we obtain:

$$G_\epsilon - \sum_{j=1}^{N} p_j \left(A_j^T G_\epsilon A_j + C_j^T C_j\right) - \left[\sum_{j=1}^{N} p_j \left(B_j^T G_\epsilon A_j + (D_j^T C_j\right)\right]^T V_\epsilon^{-1} \cdot \left[\sum_{j=1}^{N} p_j \left(B_j^T G_\epsilon A_j + (D_j^T C_j\right)\right] = \epsilon^2 I > 0$$

where, after multiplication, $V_\epsilon = \gamma^2 I - \sum_{j=1}^{N} p_j \left(B_j^T G_\epsilon B_j + D_j^T D_j\right) > 0$. Using Schur complements to this inequality:

$$\begin{bmatrix} G_\epsilon & 0 \\ 0 & \gamma^2 I \end{bmatrix} - \sum_{j=1}^{N} p_j \begin{bmatrix} A_j & B_j \\ C_j & D_j \end{bmatrix}^T \begin{bmatrix} G_\epsilon & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A_j & B_j \\ C_j & D_j \end{bmatrix} > 0 \tag{C.19}$$

The upper left block of this equation implies that $G_\epsilon > 0$. Thus $G_\epsilon > 0$ is a solution to Equation C.14. ■

Remark: The theorem gives a necessary and sufficient linear matrix inequality (LMI) for $\|\mathcal{S}\|_\infty < \gamma$. To find the H_∞ gain of a system, we can minimize γ subject to the LMI in Equation C.14. If $\mathcal{S}$ has state dimension n_x, then this requires solving a semi-definite programming problem with $1 + \frac{n_x(n_x+1)}{2}$ variables. For systems with large state dimension, this is a time consuming optimization to solve. An alternative technique is to iterate the Riccati Difference Equation (C.3) with the terminal condition $G_T(T) = 0$. Consider the following algorithm:

1: Given bounds on $\|\mathcal{S}\|_\infty$: γ_{min}, γ_{max}.

2: Choose $\gamma := \frac{\gamma_{min}+\gamma_{max}}{2}$.

3: Iterate the following Riccati Difference Equation forward starting from the initial condition $G(0) = 0$:

$$G(k+1) = \sum_{j=1}^{N} p_j \left(A_j^T G(k) A_j + C_j^T C_j\right) + \left[\sum_{j=1}^{N} p_j \left(B_j^T G(k) A_j + D_j^T C_j\right)\right]^T V^{-1}(k) \left[\sum_{j=1}^{N} p_j \left(B_j^T G(k) A_j + D_j^T C_j\right)\right]$$

where: $V(k) = \gamma^2 I - \sum_{j=1}^{N} p_j \left(B_j^T G(k) B_j + D_j^T D_j\right)$. It should be clear that this is just Equation C.3 with slightly different notation. In other words, $G(k)$ and $V(k)$ are the same as $G_k(0)$ and $V_k(0)$ in the proof above.

4a: If the matrix sequence $G(k)$ is unbounded or if $V(k)$ has a nonpositive eigenvalue for some $k \geq 0$, then set $\gamma_{min} = \gamma$ and then $\gamma := \frac{\gamma_{min}+\gamma_{max}}{2}$.

4b: If the matrix sequence converges to a solution of the Riccati Equation (C.16), then set $\gamma_{max} = \gamma$ and then $\gamma := \frac{\gamma_{min}+\gamma_{max}}{2}$.

5: Continue until $\gamma_{max} - \gamma_{min}$ is less than some desired tolerance. Set $\gamma := \frac{\gamma_{min}+\gamma_{max}}{2}$.

In the proof, we show that if the matrix sequence $G(k)$ is unbounded or if $V(k)$ has a nonpositive eigenvalue for some $k \geq 0$, then $\|\mathcal{S}\|_\infty \geq \gamma$. The only other possibility is for the matrix sequence to converge to a solution of the Riccati Equation (C.16). We have not proven that if there exists a solution to the Riccati Equation then $\|\mathcal{S}\|_\infty \leq \gamma$. However, this statement is true for a standard discrete time system (i.e. the single mode case). If this conjecture is true, then the algorithm above gives $\|S\|_\infty$ within the specified tolerance. If not, this algorithm still converges to a lower bound on $\|S\|_\infty$. The proof of this conjecture remains as future work. This algorithm is much faster than solving the semi-definite program when n_x is large. Iterating the Riccati Difference Equation has been previously employed to solve Generalized Riccati Equations stemming from the Jump Linear Quadratic Problem [1, 27].