An H_∞ Approach to Networked Control

Pete Seiler Raja Sengupta

Abstract

In this paper we study the effect of a network in the feedback loop of a control system. We consider a simple packet-loss model for the network and note that results for discrete-time linear systems with Markovian jumping parameters can be applied. We measure performance using an H_∞ norm and compute this norm via a necessary and sufficient matrix inequality condition. Finally we apply these results to study the effect of communication losses on vehicle control. These results can be used to give specifications on network performance that are necessary to achieve the given control objectives.

Index Terms

Networked Control, H_∞ Norm, Wireless networks.

I. Introduction

A networked control system is one in which a control loop is closed via a communication channel. The use of a network will lead to intermittent losses of the communicated information. These losses will deteriorate the performance and may even cause the system to go unstable. We consider the effect of random losses due to a communication network in the feedback loop. The approach taken is motivated by vehicle control problems arising in Automated Highway Systems and coordination of unmanned aerial vehicles (See [16], [14] and references therein). We briefly describe these problems. The vehicles in an Automated Highway Systems can use a radar to sense the relative spacing from their predecessor. It has been noted that tracking performance improves if all following vehicles also use the state of

Pete Seiler is with the Department of Mechanical Engineering, University of Illinois, Urbana-Champaign, Urbana IL, 61801, USA, Phone: (217) 244-3101 , email: pseiler@vehicle.me.berkeley.edu

Raja Sengupta is with the Department of Civil Engineering, University of California, Berkeley, Berkeley CA, 94720, USA, email: raja@PATH.berkeley.edu

the lead car for control [36]. The lead car information is sent over a wireless network and it would be useful to analyze the effect of packet losses on tracking performance. For the coordination of unmanned aerial vehicles (UAVs), the vision is to coordinate large numbers of UAVs to accomplish a complex task. These UAVs will communicate on an ad hoc network formed by the UAVs themselves. In this scenario, malicious forces may try to jam communication links. If a UAV senses that the current network performance is too poor for control, then the network can be reconfigured. Knowledge of the bounds on acceptable network performance is key to making this distributed agent system robust in hostile environments.

In this paper, we develop a computationally efficient method to solve problems related to networked control. Specifically, we develop a condition to compute the degradation in closed-loop performance as the probability of a packet loss increases. The performance is measured using an H_∞ cost. Using this condition, a control engineer can determine the probability of packet loss that can be tolerated before the performance degradation is unacceptable. We also develop a tractable condition to synthesize an optimal centralized controller for a networked system. We again use an H_∞ cost as the performance criterion. The optimal cost can be used to give the limits of closed loop performance for a given probability of packet loss. For a given control specification, this yields a hard bound on the network performance.

Before proceeding, we should mention several caveats to the synthesis and analysis tools that are presented in this paper. First, all tools rely on semi-definite programming optimization [3]. The statement that these tools are “computationally efficient” means that small to medium size problems can be solved exactly using freely available software [35]. However, semidefinite programming is currently not mature enough to claim that large scale networked control problems can be solved with the methods introduced in this paper. Moreover, the problem grows exponentially with the number of network links. Second, the decentralized networked control problem is not solved. It seems likely that controllers for most networked systems will be decentralized because it will be costly to communicate all information to all controllers. This enforces additional structural constraints on the controller. Unfortunately, finding H_∞-optimal controllers with additional structural constraints placed on the controller is difficult problem even for time-invariant plants. However, our results on centralized controller design can be viewed as providing the limits of possible performance.

In the remainder of the paper we develop these H_∞ analysis and synthesis tools for networked systems. In the next section, we briefly review work that is related to this paper. An important point of this review is that jump systems are commonly used to model the packet delivery characteristics of a network. This is the motivation for the theoretical results on discrete-time Markovian Jump Linear Systems (MJLS) given in Section III. We will derive a necessary and sufficient matrix inequality for a MJLS to achieve a given level of H_∞ performance. We then show how this matrix inequality can be used to synthesize optimal controllers. Finally we present several examples demonstrating the usefulness of the tools in Section IV.

II. Related Work

Models for wireless networks can vary greatly in complexity. One reason is that complex radio wave propagation, such as multipath fading, is not well understood [25]. These effects increase the probability of bit error by several orders of magnitude over wired links. Moreover, delay patterns depend greatly on the network protocol. For example, Ethernet is nondeterministic so packets can have an arbitrarily long delay [22]. On the other hand, the delay can be treated as approximately constant for some protocols, e.g. token bus [38], [21].

Since the complexity of a controller depends on the complexity of the plant 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 a packet seems appropriate. A common technique is to model the packet delivery characteristics with a discrete-time jump system [6], [20], [31], [37], [38]. However, a scientific modeling of network characteristics for control has yet to be undertaken. Nguyen, et.al. [25] have proposed a trace-based approach to modeling wireless channels at the packet-delivery level. They obtain a Markov model for the packet loss process and then evaluate the impact of errors on higher-layer network protocols. Additionally, Stubbs and Dullerud are setting up an experimental facility with the capability to model and test wireless networked control systems [34]. More work of this nature is required to obtain a network models suitable for control.

A benefit of using jump systems to model wireless networks is that there is a large body of literature for this class of systems. We briefly review these results here and leave further details until Section III. Chizeck and Ji obtained many results for Markovian Jump Linear Systems (MJLS) while solving the Jump Linear Quadratic Gaussian control problem [8], [7], [17], [18]. Ji, et al. [19] then showed equivalence of several notions of stability while Costa and Fragoso [10] derived Lyapunov conditions for stability of a MJLS. Finally, we note that several authors have developed bounded real lemmas for MJLS [1], [12], [5]. These results show that satisfying a particular matrix inequality condition is sufficient for the MJLS to have H_∞ norm less than one. It was recently proved that this condition is also necessary [29], [30]. Thus the matrix inequality condition provides a tight characterization of the input-output gain of a jump system.

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) [1], [5], [9], [13], [28]. The benefit of this approach is that the problem can be efficiently solved by interior-point methods [3], [35] with a guarantee that the global optimum will be found. However, as noted in [37], the structure of the networked control problem results in an output feedback even if the problem was originally state feedback. 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 [27] for the continuous time case and in [37] 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 [11] to find mode-dependent dynamic output feedback controllers for continuous time MJLS. In this paper, 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.

III. Markovian Jump Linear Systems

Consider the following stochastic system, denoted P:

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

where $x(k) \in \mathbb{R}^{n_x}$ is the state, $d(k) \in \mathbb{R}^{n_d}$ 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 transition probabilities $p_{ij} = \Pr(\theta(k+1) = j \mid \theta(k) = i)$ which are subject to the restrictions $p_{ij} \geq 0$ and $\sum_{j=1}^{N} p_{ij} = 1$ for any $i \in \mathcal{N}$. The plant initial conditions are given by specifying $\theta(0)$ and $x(0)$. 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 (MJLS).

We will work with sequences, $x := \{x(k)\}_{k=0}^{\infty}$, that depend on the sequence of Markov parameters, $\Theta = \{\theta(k)\}_{k=0}^{\infty}$. For notation, define $\Theta_k := \{\theta(1), \ldots, \theta(k)\}$. We define ℓ_2 as the space of square summable (stochastic) sequences:

$$\ell_2^n := \left\{ \{x(k)\}_{k=0}^{\infty} \; : \; \forall k \; x(k) \in \mathbb{R}^n \text{ is a random variable depending on } \Theta_k \text{ and } \|x\|_2 < \infty \right\}$$

where the ℓ_2-norm is defined by $\|x\|_2^2 := \sum_{k=0}^{\infty} \underset{\Theta_{k-1}}{E} \left[x(k)^T x(k) \right]$. Note that Θ_k does not contain $\theta(0)$ because this is assumed to be given as part of the plant initial conditions.

A. Stability of a MJLS

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 [19].

Definition 1: For the system given by (1) with $d \equiv 0$, the equilibrium point at $x = 0$ is:

1. Mean-square stable if for every initial state (x_0, θ_0), $\lim_{k\to\infty} \underset{\Theta_{k-1}}{E} [\|x(k)\|^2 \mid x_0, \theta_0] = 0$.
2. Stochastically stable if for every initial state (x_0, θ_0), $\sum_{k=0}^{\infty} \underset{\Theta_{k-1}}{E} [\|x(k)\|^2 \mid x_0, \theta_0] < \infty$. In other words, $\|x\|_2 < \infty$ 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 $\forall k \geq 0$, $\underset{\Theta_{k-1}}{E} [\|x(k)\|^2 \mid x_0, \theta_0] < \beta \alpha^k \|x_0\|^2$.
4. Almost surely stable if for every initial state (x_0, θ_0), $\Pr [\lim_{k\to\infty} \|x(k)\| = 0] = 1$.

Ji, et.al. showed that the first three definitions of stability are actually equivalent for a MJLS [19]. They 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 paper, references to stability will be in the sense of second-moment stability. The major motivation for this choice is that straightforward necessary and sufficient conditions exist to check for SMS but not for almost-sure stability. Below we present a necessary and sufficient condition for SMS of the jump linear system.

Theorem 1: System (1) is SMS if and only if there exist matrices $\{G_i\} > 0$ that satisfy:

$$A_i^T \left[\sum_{j=1}^{N} p_{ij} G_j \right] A_i - G_i < 0 \qquad i = 1, \ldots, N$$

In the theorem, $\{G_i\} > 0$ means $G_i > 0 \; \forall i \in \mathcal{N}$. We will use similar notation whenever all matrices in the set satisfy a given condition. Proofs of Theorem 1 can be found in [18] and [10]. The theorem states that SMS is equivalent to finding N positive definite matrices which satisfy N coupled, discrete Lyapunov equations. It is interesting to note that stability of each mode is neither necessary nor sufficient for the system to be SMS [18]. We also note that the condition simplifies under an additional assumption on the Markov process, $\theta(k)$ [10]:

Theorem 2: If $p_{ij} = p_j$ for all $i, j \in \mathcal{N}$ then System (1) is SMS if and only if there exists a matrix $G > 0$ such that:

$$\sum_{j=1}^{N} p_j A_j^T G A_j - G < 0 \tag{2}$$

Thus 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.

B. H_∞ Norm of a MJLS

Next we give the definition of the H_∞ norm [9] for discrete-time MJLS.

Definition 2: Assume that P is an SMS system. Let $x(0) = 0$ and define the H_∞ norm, denoted $\|P\|_\infty$, as:

$$\|P\|_\infty := \sup_{\theta(0) \in \mathcal{N}} \; \sup_{0 \neq d \in \ell_2^{n_d}} \frac{\|z\|_2}{\|d\|_2} \tag{3}$$

The H_∞ norm is defined as the maximum input-output gain with the appropriately defined 2-norms. To derive the bounded real lemma, we need a definition of controllability for a MJLS.

Definition 3: The system, P, is weakly controllable if for every initial state/mode, (x_0, θ_0), and any final state/mode, (x_f, θ_f), there exists a finite time T_c and an input $d_c(k)$ such that $Pr[x(T_c) = x_f \text{ and } \theta(T_c) = \theta_f] > 0$.

This version of weak controllability is motivated by the definition given by Ji and Chizeck [17]. The weak controllability assumption in the bounded real lemma ensures that the disturbance can affect the system state. If the system is not weakly controllable, the LMI condition is still sufficient, but it may not be necessary.

Theorem 3 (Bounded Real Lemma) Assume the system, P, is weakly controllable. P is SMS and $\|P\|_\infty < \gamma$ if and only if there exist matrices $\{G_i\} > 0$ that satisfy $\{R_i\} < 0$ where:

$$R_i := \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}^T \begin{bmatrix} \sum_{j=1}^N p_{ij} G_j & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix} - \begin{bmatrix} G_i & 0 \\ 0 & \gamma^2 I \end{bmatrix} \qquad i \in \mathcal{N} \qquad (4)$$

The sufficiency of Equation 4 was proved in [1], [12], [5] using a stochastic Lyapunov approach. The necessity of this condition was recently proved in [30]. 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 4. For the case of one mode ($N = 1$), the H_∞ condition given by Equation 4 reduces to the standard necessary and sufficient condition for discrete-time systems [26]. Finally we note that this condition also simplifies under an additional assumption on the Markov chain.

Theorem 4: Assume $p_{ij} = p_j$ for all $i, j \in \mathcal{N}$ and the system, $\mathcal{P}$, is weakly controllable. $\mathcal{P}$ is SMS and $\|\mathcal{P}\|_\infty < \gamma$ if and only if there exists a symmetric matrix $G > 0$ that satisfies the following matrix inequality:

$$\sum_{j=1}^N p_j \begin{bmatrix} A_j & B_j \\ C_j & D_j \end{bmatrix}^T \begin{bmatrix} G & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A_j & B_j \\ C_j & D_j \end{bmatrix} - \begin{bmatrix} G & 0 \\ 0 & \gamma^2 I \end{bmatrix} < 0 \qquad (5)$$

The necessity of this simplified condition was proved in [29]. In the following section, we will use this matrix inequality condition for controller synthesis. It will turn out that the controller matrices are embedded in the A_j, B_j, and C_j matrices. Unfortunately, Equation 5 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 which is more suitable for this purpose.

Let $Z = G^{-1}$ and multiply Equation 5 on the left and right by the following congruence transformation: $\left[\begin{smallmatrix} Z & 0 \\ 0 & I \end{smallmatrix}\right]$. Since definiteness is invariant under congruence transformations, Equation 5 is equivalent to:

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

Use the Schur complement lemma [3] to convert this inequality into the following equivalent condition where $(\bullet)^T$ denote entries which can be inferred from the symmetry of the matrix:

$$\begin{bmatrix} \begin{bmatrix} Z & 0 \\ 0 & \gamma^2 I \end{bmatrix} & (\bullet)^T & \cdots & (\bullet)^T \\ \sqrt{p_1} \begin{bmatrix} A_1 Z & B_1 \\ C_1 Z & D_1 \end{bmatrix} & \begin{bmatrix} Z & 0 \\ 0 & I \end{bmatrix} & \cdots & (\bullet)^T \\ \vdots & & \ddots & \vdots \\ \sqrt{p_N} \begin{bmatrix} A_N Z & B_N \\ C_N Z & D_N \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \cdots & \begin{bmatrix} Z & 0 \\ 0 & I \end{bmatrix} \end{bmatrix} > 0 \qquad (6)$$

C. H_∞ Controller Synthesis

In this section we will apply Theorem 4 to derive an LMI condition for controller synthesis. We consider plants, $\mathcal{P}$, of the form:

$$\begin{bmatrix} x(k+1) \\ z(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} A_{\theta(k)} & B_{1,\theta(k)} & B_{2,\theta(k)} \\ C_{1,\theta(k)} & D_{11,\theta(k)} & D_{12,\theta(k)} \\ C_{2,\theta(k)} & D_{21,\theta(k)} & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ d(k) \\ u(k) \end{bmatrix} \tag{7}$$

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, $d(k) \in \mathbb{R}^{n_d}$ 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 $p_{ij} = p_j \; \forall i, j \in \mathcal{N}$.

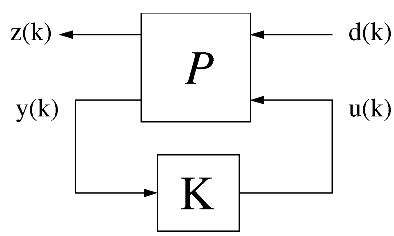

Fig. 1. Feedback Loop for H_∞ Control Design

Our goal is to design a controller, K, for the feedback loop (Figure 1) which minimizes the closed loop gain from d 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:

$$\begin{aligned} x_c(k+1) &= A_{c,\theta(k)} x_c(k) + B_{c,\theta(k)} y(k) \\ u(k) &= C_{c,\theta(k)} x_c(k) \end{aligned} \tag{8}$$

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 do not make restrict the state dimension of the controller. The controller state dimension is only assumed to be finite, $n_c < \infty$. We will show in Theorem 5 that if a stabilizing controller achieving a certain level of H_∞ performance exists, then we can always find a controller achieving the same level of performance and with state dimension equal to $n_c = n_x$. On the other hand, it should be noted that we are restricting the search to controllers that 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 $\theta(k)$. A larger class of controllers would use all past values of $\theta(k)$.

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

$$\begin{bmatrix} x_{cl}(k+1) \\ z(k) \end{bmatrix} = \begin{bmatrix} A_{cl,\theta(k)} & B_{cl,\theta(k)} \\ C_{cl,\theta(k)} & D_{cl,\theta(k)} \end{bmatrix} \begin{bmatrix} x_{cl}(k) \\ d(k) \end{bmatrix} \tag{9}$$

where:

$$\begin{aligned} A_{cl,\theta(k)} &:= \begin{bmatrix} A_{\theta(k)} & B_{2,\theta(k)} C_{c,\theta(k)} \\ B_{c,\theta(k)} C_{2,\theta(k)} & A_{c,\theta(k)} \end{bmatrix} \\ B_{cl,\theta(k)} &:= \begin{bmatrix} B_{1,\theta(k)} \\ B_{c,\theta(k)} D_{21,\theta(k)} \end{bmatrix} \\ C_{cl,\theta(k)} &:= \begin{bmatrix} C_{1,\theta(k)} & D_{12,\theta(k)} C_{c,\theta(k)} \end{bmatrix} \\ D_{cl,\theta(k)} &:= D_{11,\theta(k)} \end{aligned}$$

The subscript 'cl' denotes the closed loop matrices. For the closed loop system, the transition probabilities satisfy the assumption given in Theorem 4: $p_{cl,ij} = p_{cl,j}$ for all i, j. Apply Theorem 4 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 holds. Or we can apply the equivalent condition given by Equation 6; 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:

$$\begin{bmatrix} \begin{bmatrix} Z & 0 \\ 0 & \gamma^2 I_{n_w} \end{bmatrix} & (\bullet)^T & (\bullet)^T \\ \sqrt{p_1}\begin{bmatrix} A_{cl,1}Z & B_{cl,1} \\ C_{cl,1}Z & D_{cl,1} \end{bmatrix} & \begin{bmatrix} Z & 0 \\ 0 & I_{n_z} \end{bmatrix} & (\bullet)^T \\ \sqrt{p_2}\begin{bmatrix} A_{cl,2}Z & B_{cl,2} \\ C_{cl,2}Z & D_{cl,2} \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} Z & 0 \\ 0 & I_{n_z} \end{bmatrix} \end{bmatrix} > 0 \tag{10}$$

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 10. 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: 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 10 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:

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

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 [11]. Assume Equation 10 holds and partition Z compatibly with A_{cl}:

$$Z = \begin{bmatrix} Z_1 & Z_2 \\ Z_2^T & Z_3 \end{bmatrix} \tag{12}$$

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 10 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:

$$T := \left[\begin{array}{cc|c} Y & I_{n_x} & 0 \\ -Z_3^{-1} Z_2^T Y & 0 & 0 \\ \hline 0 & 0 & I_{n_w} \end{array}\right] \tag{13}$$

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 10 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:

$$
\begin{aligned}
T_{11}^T Z T_{11} &= \begin{bmatrix} Y & I_{n_x} \\ I_{n_x} & Z_1 \end{bmatrix} \\
T_{11}^T A_{cl,i} Z T_{11} &= \begin{bmatrix} YA_i + L_i(C_2)_i & W_i \\ A_i & A_i Z_1 + (B_2)_i F_i \end{bmatrix} \\
C_{cl,i} Z T_{11} &= \begin{bmatrix} (C_1)_i & (C_1)_i Z_1 + (D_{12})_i F_i \end{bmatrix} \\
T_{11}^T B_{cl,i} &= \begin{bmatrix} Y(B_1)_i + L_i(D_{21})_i \\ (B_1)_i \end{bmatrix}
\end{aligned}
$$

where:

$$
\begin{aligned}
Y &:= (Z_1 - Z_2 Z_3^{-1} Z_2^T)^{-1} \\
X &:= Z_1 \\
F_i &:= C_{ci} Z_2^T \\
L_i &:= -Y Z_2 Z_3^{-1} B_{ci} \\
W_i &:= Y A_i Z_1 + Y(B_2)_i F_i + L_i (C_2)_i Z_1 - Y Z_2 Z_3^{-1} A_{ci} Z_2^T
\end{aligned}
$$

The congruence transformation shows that Equation 11 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 11 holds. Define the transformation:

$$
\tilde{T} = \left[\begin{array}{cc|c} 0 & Y^{-1} & 0 \\ I_{n_x} & -I_{n_x} & 0 \\ \hline 0 & 0 & I_{n_w} \end{array}\right] \tag{14}
$$

Multiply Equation 10 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 10 is satisfied with the following scaling and controller matrices:

$$\begin{aligned} Z &= \begin{bmatrix} X & Y^{-1} - X \\ Y^{-1} - X & X - Y^{-1} \end{bmatrix} \qquad (15) \\ B_{ci} &= Y^{-1} L_i \\ C_{ci} &= F_i \left(Y^{-1} - X\right)^{-1} \\ A_{ci} &= -Y^{-1} \left[Y A_i X + Y (B_2)_i F_i + L_i (C_2)_i X - W_i \right] \left(Y^{-1} - X\right)^{-1} \end{aligned}$$

Also note that Condition 11 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:

$$\min \gamma$$
$$\text{subject to Equation 11}$$

As noted previously, this problem can be efficiently solved by interior-point methods using freely available software [3], [35]. 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 $n_c \le n_x$. 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 $n_c \le n_x$ which achieves the same performance. In words, the algorithm gives the optimal controller of the form given in Equation 8. Finally, note that the proof given above can be easily extended to systems with more than 2 modes.

IV. Numerical Examples

In this section, we show how the tools in this paper can be used for analysis of and controller synthesis for networked control systems. First we analyze the closed loop peformance of a 5 car platoon using a controller designed at California Partners for Advanced Transit and Highways (PATH) [16]. This analysis is performed for two different assumptions on the network. Then we synthesize an optimal controller for a vehicle following problem.

A. Analysis of Platoon Controller

The platoon (Figure 2) is a string of 5 vehicles. Let x_0 denote the position of the lead car and x_i $(i = 1, \ldots, 4)$ denote the position of the i^{th} follower in the platoon. The reference trajectory for the lead vehicle is denoted as r_0 and the tracking error for the lead vehicle is $e_0 = r_0 - x_0$. Define the vehicle spacing errors as: $e_i = x_{i-1} - x_i - \delta_i$ $(i = 1, \ldots, 4)$ where δ_i is the desired vehicle spacing. The control objective is to force all tracking errors to zero.

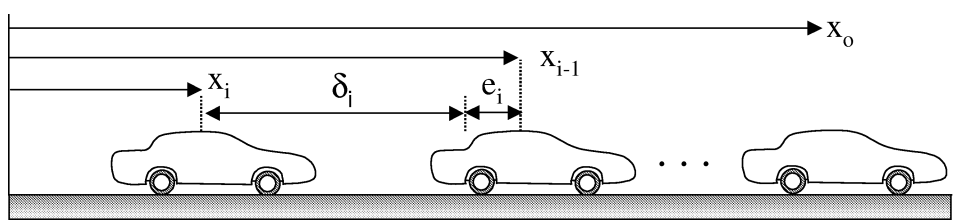

Fig. 2. A platoon of vehicles

The dynamics for an individual vehicle are nonlinear due to wind drag, engine dynamics, and brake dynamics. However, the vehicles at California Partners for Advanced Transit and Highways (PATH) [16] use a two-layered controller: the upper layer computes a desired acceleration and the lower layer computes throttle / brake pressure commands to track this acceleration profile. After linearization by the lower layer controller, a reasonable model for the vehicle dynamics is a double integrator plus a first-order lag:

$$\frac{d}{dt}\begin{bmatrix} x_i(t) \\ v_i(t) \\ a_i(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau} \end{bmatrix}\begin{bmatrix} x_i(t) \\ v_i(t) \\ a_i(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\tau} \end{bmatrix} u_i(t) \tag{16}$$

where $x_i(t)$, $v_i(t)$, and $a_i(t)$ are the position, velocity and acceleration of the i^{th} vehicle. $\tau = 100$ milliseconds is the time constant of the first-order lag. The upper layer runs at a sample time of $T_s = 20$ milliseconds and a zero-order hold is used on the control input. After discretization, the open loop model for the entire platoon can be written as:

$$x_{OL}(k+1) = A_{OL}\ x_{OL}(k) + B_{OL,1}\ r_0(k) + B_{OL,2}\ u(k) \tag{17}$$

where $x_{OL}(k) := \begin{bmatrix} x_0(k) & v_0(k) & a_0(k) & \dots & x_4(k) & v_4(k) & a_4(k) \end{bmatrix}^T$ is the discrete state vector and $u(k) = [u_0(k) \dots u_4(k)]^T$ is the vector of all control inputs.

At each sample time, the upper layer controller computes the desired acceleration as:

$$\begin{aligned} u_0(k) &= k_d * \dot{e}_0(k) + k_p * e_0(k) \\ u_i(k) &= \lambda * a_0(k) + (1-\lambda) * a_{i-1}(k) + k_0 * (v_0(k) - v_i(k)) + k_d * \dot{e}_i(k) + k_p * e_i(k) \\ & \qquad i = 1, \dots, 4 \end{aligned} \tag{18}$$

Note that each follower is coupled to the preceding vehicle by the spacing error, range rate, and acceleration terms. The controller also makes use of range rate relative to the leader and the lead vehicle acceleration. These extra terms are included for string stability [15], which is a property ensuring error waves do not amplify as they propagate through this decentralized system. The spacing error, e_i, range rate, $\dot{e}_i$, and vehicle velocity, v_i, are computed by on-board sensors but the remaining terms must be communicated.

We make several assumptions about the network, quantization, coding/decoding, and protocol involved in communicating this information. These assumptions are motivated by the use of wireless networks at PATH [21], [23] and at the University of California, Berkeley for Unmanned Aerial Vehicle research [21], [32]. First, we assume that quantization errors are small and that errors introduced by quantization can be modeled as norm bounded disturbances [33]. Second, we assume that we can always detect packet errors, although we may not be able to correct them. In reality, error detection schemes are not perfect. However, observers and/or physically meaningful bounds can be used to eliminate corrupted packets that contain unrealistic data. If a packet is corrupted, it is not used. Finally, we assume that the network protocol is such that jitter and transmission delays are negligible and corrupted data is not retransmitted.

The implication of the final assumption is that the networked system can be modeled as a synchronous process. That is, at every sample time a vehicle communicates its measurement. 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. This final assumption is valid for a deterministic protocol, such as token bus, that uses a Try-Once-Discard strategy. PATH uses such a protocol for the control of small platoons of vehicles [23].

Let $y_c(k)$ and $\hat{y}_c(k)$ denote a single measurement that is sent and received, respectively. If we apply these assumptions then a simple packet-loss model for the network is given by:

$$\hat{y}(k) = \begin{cases} y(k) & \text{if } \theta(k) = R \\ \emptyset & \text{if } \theta(k) = L \end{cases} \tag{19}$$

where $\emptyset$ denotes a corrupted (i.e. useless) packet of information. $\theta(k) = L$ denotes a corrupted packet was received and discarded while $\theta(k) = R$ denotes an error-free packet was received. This is known as an erasure model for a network. $\theta(k)$ is a random process that governs the packet delivery characteristics of the network. For many wireless networks, 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. A two state Markov process can be used to model this bursty packet loss process (Figure 3): $p_{i,j} = Pr[\theta(k+1) = j \mid \theta(k) = i]$ for $i, j \in \{L, R\}$. The bursty nature is modeled with $p_{L,L} > p_{R,L}$.

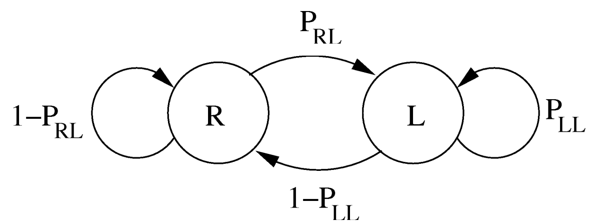

Fig. 3. Two state Markov Network Model

In principle, we need to model 7 channels: the leader must communicate its acceleration and velocity to cars 2 through 4 and cars 0 through 3 must communicate their acceleration to the following vehicle. Each of these channels can be modeled as an erasure channel with underlying packet loss process, $\theta_i(k)$. Each $\theta_i(k) \in \{L, R\}$, so the network can be in any one of $2^7 = 128$ states at each sample time. As noted in the introduction, the number of network states grows exponentially in the number of links. For this example, however, we will make the simplifying assumption that all communicated measurements in the platoon are simultaneously corrupted or received. Based on this simplifying assumption, the network has only 2 possible states and it can be modeled as:

$$\hat{y}_c(k) = \begin{cases} y_c(k) & \text{if } \theta(k) = R \\ \emptyset & \text{if } \theta(k) = L \end{cases} \tag{20}$$

where $y_c(k)$ is the vector of communicated measurements available for feedback, $y_c(k) = \begin{bmatrix} v_0 & a_0 & a_1 & a_2 & a_3 \end{bmatrix}^T$. The measurements available from on-board sensors are:

$$y_o(k) = \begin{bmatrix} e_0 & \dots & e_4 & \dot{e}_0 & \dots & \dot{e}_4 & v_1 & \dots & v_4 \end{bmatrix}^T$$

If the measurements are communicated without error ($\theta(k) = R$), the controller in Equation 18 can be written as $u(k) = K_1 y(k)$ where the measurement vector for the platoon takes the form:

$$y(k) := \begin{bmatrix} y_o(k) \\ y_c(k) \end{bmatrix} = \begin{bmatrix} C_{OL,1} \\ C_{OL,2} \end{bmatrix} x_{OL}(k) + \begin{bmatrix} D_{OL,1} \\ D_{OL,2} \end{bmatrix} r_0(k)$$

If the measurements are received with an error ($\theta(k) = L$), then each controller discards the corrupted packet and implements the modified control action that uses only measurements from on-board sensors: $u_i(k) = k_d * \dot{e}_i(k) + k_p * e_i(k)$ for $i = 0, \ldots, 4$. In this case, the controller can be written as $u(k) = K_0 y(k)$ with the appropriate gain matrix K_0. To summarize, each follower uses the following switching logic:

$$u_i(k) = \begin{cases} \lambda * a_0(k) + (1-\lambda) * a_{i-1}(k) + k_0 * (v_0(k) - v_i(k)) + k_d * \dot{e}_i(k) + k_p * e_i(k) & \text{if } \theta(k) = R \\ k_d * \dot{e}_i(k) + k_p * e_i(k) & \text{if } \theta(k) = L \end{cases} \tag{21}$$

The control gains are $\lambda = 0.1$, $k_0 = 0.07$, $k_d = 1.33$, and $k_p = 0.49$. Note that this control law switches based on the current value of $\theta(k)$ and does not depend on past values of $\theta(k)$.

The closed-loop system consisting of the open loop plant, the network, and the controller is depicted on the left side of Figure 4. The primary platoon disturbance is the lead vehicle reference trajectory, so we define the generalized disturbance vector as $d(k) := r_0(k)$. Our goal is to analyze how the spacing errors are affected by packet losses, so we define the generalized error vector as $z(k) := [e_0(k) \ \ldots \ e_4(k)]^T$. In a more complicated analysis problem, we could include wind gusts and sensors noises in $d(k)$ and control effort in $z(k)$. In any case, the closed loop is just a jump linear system as depicted on the right side of Figure 4. Given this formulation, we can apply the bounded real lemmas (Theorem 3 and 4) to analyze the platoon performance.

First, we assume that $\theta(k)$ is a Bernoulli process: $p_{R,L} = p_{L,L} := p_L$ and $p_{R,R} = p_{L,R} := 1 - p_L$. Thus the probability that any given packet will be lost, p_L, is independent of the past. Figure 5 shows the H_∞ gain from disturbance to errors as a function of the packet

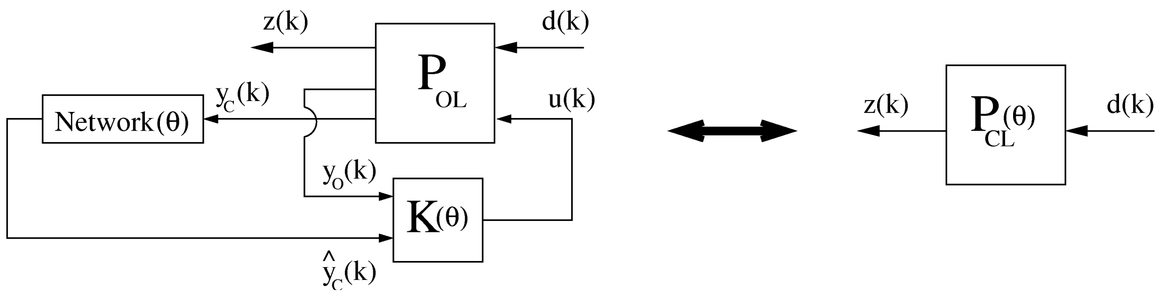

Fig. 4. The closed loop is a jump system

loss rate, p_L. This plot was computed using the simplified version of the bounded real lemma, Theorem 4. This plot shows there is a minor degradation in performance until $p_L = 0.5$ at which point there is a sharp knee in the curve. Depending on the control objectives, one could make a case that packet loss rates up $p_L = .50$ are acceptable. We should note that reasonable performance is obtained at relatively high packet loss rates because the spacing errors and range rates are obtained from on-board measurements and are not subject to network delays. Thus each vehicle controller has access to these key measurements at all times. The objectives at PATH require very tight spacing control, so one would have to determine how much performance can be sacrificed. Figure 5 would then lead to a performance constraint on the network.

We should also comment on the "computational efficiency" of these semidefinite programming optimizations. The entire plot of 21 points was generated in 66 seconds on a 700 MHz processor. At $p_L = 0$, the packets are always received and the closed loop plant actually takes on only one mode. Similarly, when $p_L = 1$, all packets are lost and the closed loop again takes on one mode. In both these cases, the closed loop is a time-invariant plant and we can apply standard techniques to compute the H_∞ gain. The bold x's at $p_L = 0$ and $p_L = 1$ are the gains computed using the MATLAB function `dhfnorm`. The values agree with the gains computed using Equation 5. For comparison, the data points at $p_L = 0$ and $p_L = 1$ took 3 seconds to compute using the bounded real lemma, but only 0.05 seconds with `dhfnorm`. The algorithm used by `dhfnorm` relies on eigenvalue decompositions of a Hamiltonian matrix and bisection [2], [4]. It seems unlikely that this Hamiltonian-based approach, which is based on frequency-domain arguments, can be extended to time-varying jump systems. However, this comparison demonstrates that there is room for improving the computational efficiency of our optimization-based approach.

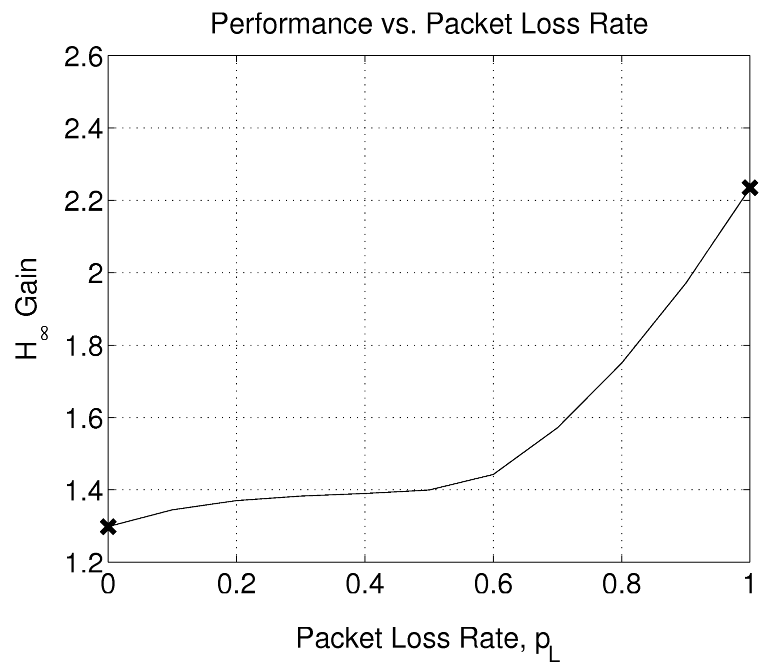

Fig. 5. Platoon Performance versus Packet Loss Rate: Bernoulli Channel

Next, we consider the two state Markov model for the network. This model is characterized by the loss rate after receiving an error-free packet, $p_{R,L}$, and the loss rate after receiving a corrupted packet, $p_{L,L}$. Figure 6 shows the H_∞ gain from disturbance to errors as a function of these two loss rates. This plot was computed using the general version of the bounded real lemma, Theorem 3. This plot consists of 121 points and was computed in 31 minutes on a 700 MHz processor. The Bernoulli channel corresponds to the case that $p_{R,L} = p_{L,L}$. Thus the dark solid line and bold x's running along the diagonal correspond to the same data as in Figure 5. This plot has an interesting peak when $p_{L,L} \approx 1$ and $p_{R,L} \approx 0$. This peak means that performance is actually worse for the a bursty channel than if the communicated measurements are always corrupted. Recall that the platoon controller (Equation 21) only uses the current value of $\theta(k)$ and ignores past values. For the bursty channel, the present is highly correlated with the past. An interpretation of this peak is that the controller, which neglects this correlation, is not suitable for a bursty channel. This analysis leads to many interesting conjectures on how controllers should be designed for such channels.

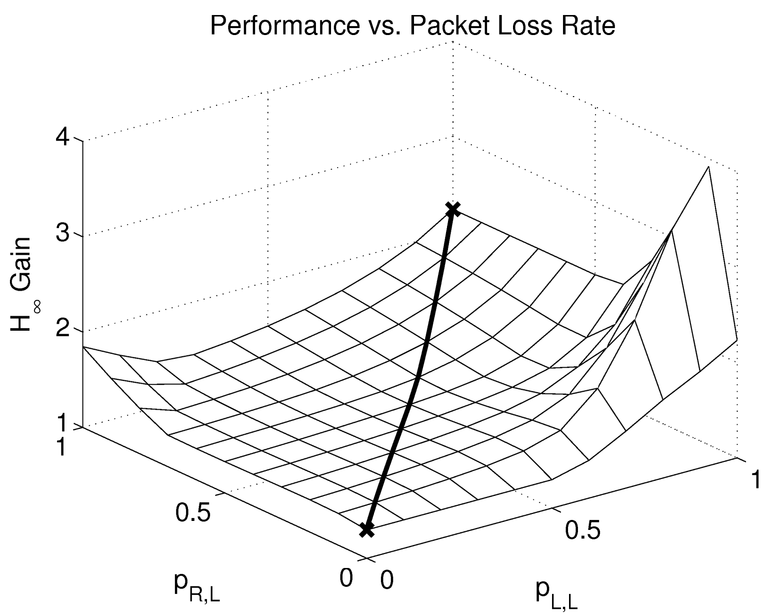

Fig. 6. Platoon Performance versus Packet Loss Rate: Two-state Markov Channel

B. *Controller Synthesis for Vehicle Following*

In this section, we synthesize a controller for a vehicle following problem. We use the same notation presented in the previous section except that we consider a two-car platoon. The goal is to have vehicle 1 follow a distance δ behind vehicle 0. In other words, the controller should regulate the spacing error, $e := x_0 - x_1 - \delta$, to zero. We will use the LMI condition derived in Section III to design a controller assuming the vehicles are governed by the linear dynamics in Equation 16. We again discretized these dynamics by sampling the output every T_s seconds and applying a first-order hold at the input.

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_s seconds vehicle 0 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 7:

$$\begin{bmatrix} x_{OL}(k+1) \\ z(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_{2,\theta(k)} & D_{21,\theta(k)} & 0 \end{bmatrix} \begin{bmatrix} x_{OL}(k) \\ d(k) \\ u_2(k) \end{bmatrix} \tag{22}$$

where $x_{OL}(k) := \begin{bmatrix} x_0(k) & v_0(k) & a_0(k) & x_1(k) & v_1(k) & a_1(k) \end{bmatrix}^T \in \mathbb{R}^6$ is the discrete state vector. The measurement vector available to car 1 is $y(k) = \begin{bmatrix} y_o(k) \\ \hat{y}_c(k) \end{bmatrix}$ where $y_o(k)$ is the state of vehicle 1 and $\hat{y}_c(k)$ is the state vector of vehicle 0 that is communicated to the controller on vehicle 1. We again use the Bernoulli communication model (Equation 20).

For this problem, the error vector, $z(k)$, penalizes the spacing error and the weighted control effort: $z(k) := [e(k)\ \epsilon_u u_1(k)]^T \in \mathbb{R}^2$. The weighting on the control effort is given by $\epsilon_u = 1.0$. The disturbance vector is given by: $d(k) := [u_0(k)\ d_1(k)\ n_0^T(k)\ n_1^T(k)]^T \in R^8$. This vector includes the lead vehicle acceleration ($u_0(k) \in \mathbb{R}$), a plant disturbance on vehicle 1 ($d_1(k) \in \mathbb{R}$), and sensor noises for all measurements ($n_0(k) \in \mathbb{R}^3$ and $n_1(k) \in \mathbb{R}^3$). The vehicle 1 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 0 is unstable (it has two poles at $z = 1$) and uncontrollable from $u_1(k)$. To solve this problem, we use a technique that is common to handle unstable weights in H_∞ design problems [24]. The model for vehicle 0 is altered slightly by moving the unstable poles just inside the unit disk.

Figure 5 shows the performance of the H_∞ optimal controller as a function of the packet loss rate, p_L. The performance has been normalized by the nominal performance at $p_L = 0$, 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). Figure 5 shows that performance rapidly degrades for packet loss rates above $p_L = 0.9$ when $T_s = 20$ milliseconds. 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 $T_s = 20$ 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 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. Since the sample rate is tied to network bandwidth, plotting these curves for several sample rates shows the trade-off between packet loss rate and network bandwidth.

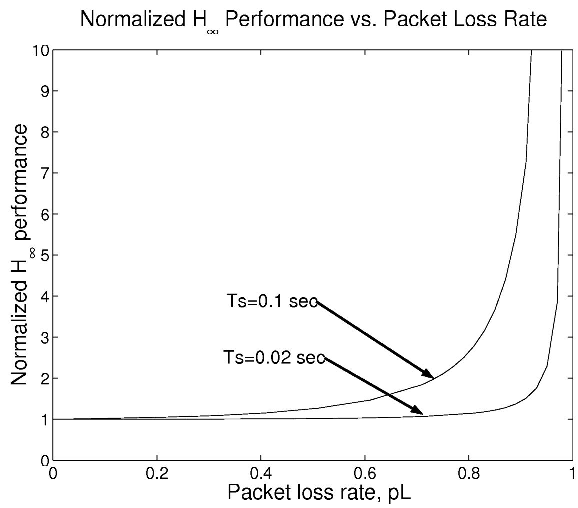

Fig. 7. Normalized H_∞ performance vs. packet loss rate: Curve shows car following performance for the optimal controller

V. Conclusions

In this paper we studied the effect of a network in the feedback loop of a control system. We measured control performance using the H_∞ gain from disturbances to errors. This gain is computable via a necessary and sufficient matrix inequality. These results can be used to give specifications on network performance that are necessary to achieve the given control objectives. Two problems related to the computational cost of this method need further exploration. First, the number of Markov states needed to model the network grows exponentially with the number of channels. Second, computing the gain using the matrix inequality condition is costly for moderate to large systems.

References

[1] 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.

[2] S. Boyd, V. Balakrishnan, and P. Kabamba. A bisection method for computing the H_∞ norm of a transfer matrix and related problems. *Mathematics of Control, Signals, and Systems*, 2(3):207–219, 1989.

[3] 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.

[4] N.A. Bruinsma and M. Steinbuch. A fast algorithm to compute the H_∞-norm of a transfer function matrix. *Systems and Control Letters*, 14(4):287–293, April 1990.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] C.E. de Souza and M.D. Fragoso. H_∞ filtering for discrete-time systems with markovian jumping parameters. In *Proceedings of the 36th CDC*, pages 2181–2186, 1997.

[13] 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.

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

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

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

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

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

[25] 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.

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

[27] 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.

[28] 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.

[29] P. Seiler. *Coordinated Control of Unmanned Aerial Vehicles*. PhD thesis, University of California, Berkeley, Fall 2001.

[30] P. Seiler and R. Sengupta. A bounded real lemma for jump systems. Sumitted to IEEE Transactions on Automatic Control.

[31] 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.

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

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

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

[35] J. Sturm. SeDuMi version 1.05. http://fewcal.kub.nl/sturm/software/sedumi.html, 2001.

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

[37] 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.

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