Towards Receding Horizon Networked Control ⋆

Vijay Gupta [a], Sachin Adlakha [b], Bruno Sinopoli [b] and Andrea Goldsmith [b]

[a] *Institute for Systems Research, University of Maryland, College Park, MD 20742, USA*

[b] *Department of Electrical Engineering, Stanford University, Stanford, CA, USA*

Abstract

This paper deals with the design of control systems over communication channels that stochastically drop packets. To deal with the data loss in the channel between the controller and the actuator, the use of actuation buffers and a receding horizon control strategy is proposed for the Linear Quadratic Control (LQG) problem. Optimal control inputs are designed via a separation principle and the stability and performance is analyzed. The theoretical results are illustrated with the help of simulations that show the improvement in overall performance as the actuator buffer length is increased.

1 Introduction and Motivation

Recently, there has been a lot of interest in systems where the plant and the controller communicate over imperfect communication links or networks . Such systems, that are sometimes generically referred to as Networked Control Systems (NCS), are expected to become more important as applications of decentralized estimation and control mature. Various researchers have looked at understanding and counteracting specific effects introduced by the communication links in control loops.

In this work, we look at the systems communicating over links that randomly drop packets. The nominal system is shown in Figure 1. A discrete-time plant is being controlled remotely by a controller. Both the sensor-controller and the controller-actuator channels drop packets stochastically. Preliminary work in this area has largely concentrated on system stability when the sensor-controller channel drops packets in an independent and identically distributed (i.i.d.) fashion, as in [22,25]. Performance of such systems as a function of packet loss rate was analyzed, for example, by Seiler in [22] and by Ling and Lemmon in [15] assuming specific dropout models. To compensate for the data loss, several approaches have been proposed by Nilsson [20], Hadjicostis and Touri [12], Ling and Lemmon [15] and

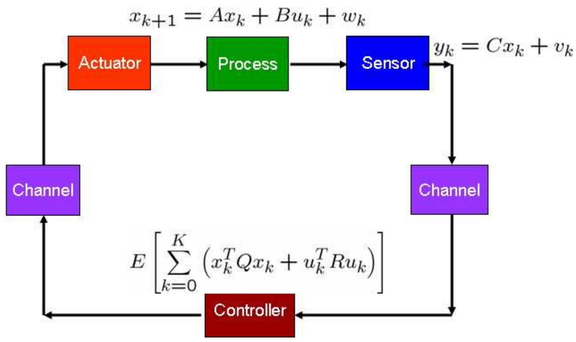

Fig. 1. System setup considered in the problem.

Montestruque and Antsaklis [18]. The works of Azimi-Sadjadi [1], Schenato et al. [23] and Imer et al. [13] who looked at controller structures to minimize quadratic costs for systems in which both sensor-controller and controller-actuator channels are present are also relevant. Finally, we would like to mention the stability and performance analysis done for the related problem of optimal estimation across a packet-dropping link that was considered by Sinopoli et al. in [24] for the case of one sensor with packet drops occurring in an i.i.d. fashion and by Gupta et al in [7] for multiple sensors and for more general packet drop models.

This work takes a more general approach to the control of networked control systems. It has often been recognized that a typical network / communication data packet has much more space for carrying information than is required by a traditional control loop. For instance, as mentioned in [11], the minimum size of an Ethernet data packet is 72 bytes, while a typical data point will only consume 2 bytes. Many other examples

⋆ This paper was not presented at any IFAC meeting. Corresponding author V. Gupta.

Email addresses: vijay@cds.caltech.edu (Vijay Gupta), adlakha@stanford.edu (Sachin Adlakha), sinopoli@stanford.edu (Bruno Sinopoli), andrea@ee.stanford.edu (Andrea Goldsmith).

are given in Lian et al. [16]. Moreover, many devices used in networked control systems have some processing and memory capabilities to communicate across wireless channels or networks. One can pre-process information prior to transmission and/or transmit extra data to improve the performance of a networked control system by countering the effects introduced by communication channels, such as, packet delays, losses, etc. In Gupta et al. [10] it was shown that pre-processing (or encoding) information before transmission over the communication link can indeed yield huge improvements in terms of stability and performance. That work focused on the sensor-controller channel and identified simple recursive encoding strategies that were optimal. The benefits incurred become even more apparent when data needs to be transmitted over *networks* instead of links, as shown in [8], or when multiple sensors are present [9,10].

In this work, we study the benefit of encoding data over the controller-actuator channel. We analyze the system for the case of an actuator that does not have access to any plant information and when the information processing strategy is to transmit a trajectory of future control inputs by the controller, similar to the practice in a Receding Horizon control strategy. Thus the work can also be seen to be related to the work on model predictive control of networked control systems. The idea of placing buffers at the actuator end and transmitting many control inputs at every time step was proposed by Graham et al. in [6] but no theoretical analysis was carried out. Montestruque and Antsaklis [17] proposed using a model of the plant at the controller to reduce transmission frequency from the sensors while retaining stability. Georgiev and Tilbury [11] considered the problem of designing an H_2-optimal controller that transmits multiple data points in the same network packet for a continuous-time plant so that transmission frequency is reduced through a better use of the data packet. However, effects such as packet drops or delays were not considered. Kawka and Alleyne [14] also proposed transmitting multiple control values per packet to deal with packet drops; however the analysis was limited to calculating the stability and no control synthesis method was proposed. This idea was also proposed by Naghshtabrizi and Hespanha [21] who studied the stability of such a system using delay differential equations and proposed a numerical algorithm to synthesize stabilizing controllers.

In this paper, we consider a discrete-time system that is communicating with a controller over links that drop packets in an i.i.d. or a bursty (according to a Markov model) fashion. Using a separation principle, we first solve for the optimal control trajectory that should be transmitted. This complements the practice heuristically carried out in many implementations of networked control systems, e.g., [5]. We then carry out the stability and performance analysis for the system. Simulation examples presented in the paper illustrate the benefits of using the proposed encoding strategy. We see that, with our algorithm there is an increase in the performance of the system for the same packet drop probabilities. Alternatively, we can use our algorithm to decrease the transmission rate for the same performance level.

The paper is organized as follows. We begin by formulating the problem in the next section. In Section 3, we prove a separation principle and show how to choose the optimal control inputs. Using tools from [7], we then carry out a stability and performance analysis of the system. In Section 4 we illustrate the results using a few examples.

2 Problem Formulation

Consider the system setup shown in Figure 1. The process evolves in discrete-time as [1]

$$x_{k+1} = Ax_k + Bu_k + w_k, \tag{1}$$

where $x_k \in \mathbf{R}^m$ is the process state, $u_k \in \mathbf{R}^n$ is the control input and w_k is the process noise assumed white, Gaussian, zero mean and with covariance matrix $R_w > 0$. The state of the process is observed by a sensor with observations of the form

$$y_k = Cx_k + v_k, \tag{2}$$

where v_k is the measurement noise again assumed white, Gaussian, zero mean and with covariance $R_v > 0$. Further, the noises w_k and v_k are assumed independent of each other. We assume that the pairs (A, B) and $(A, R_v^{1/2})$ are controllable and the pairs (A, C) and $(A, R_w^{1/2})$ are observable.

The channels (or networks) present in the control loop are modeled using the packet erasure model. Thus a channel takes in as input a data packet consisting of a vector of real numbers. We assume enough bits available so that quantization error is subsumed by the measurement and process noise. [2] At every time step, the channel erases the data packet stochastically. We assume that erasures can be modeled as a Markov chain, which can model bursty errors as typically seen in real wireless channels and which include the i.i.d Bernoulli drops as a special case. Even though our analysis can be extended to a Markov chain with an arbitrary number of states, for ease of exposition, we will cover only the Gilbert-Elliott channel model [19]. The Gilbert-Elliott channel (Figure 2) is a simple two state channel where the loss process is

[1] Even though all the arguments in the paper can be carried over for the case of time-varying systems, we consider the time-invariant version for ease of exposition.

[2] Once again, the large size of a typical data packet makes this assumption reasonable.

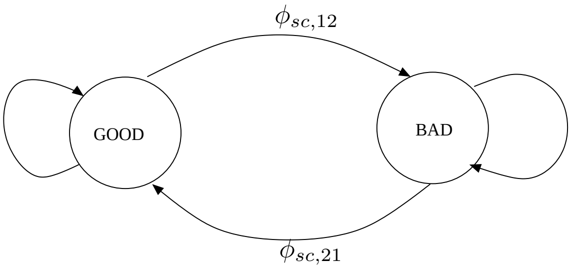

Fig. 2. Gilbert-Elliott Channel Model

determined by the current state of a discrete time stationary binary Markov Process. The Markov chain represents the channel state transition. The two states are defined as a *good state* and a *bad state*, denoted by G or 1 and B or 2, respectively. The packet loss probability in the *good state* is much less that the packet loss probability in the *bad state*. However in this paper we assume that no packets are lost in the *good state* while all packets are lost in the *bad state*. The results presented below however can be easily generalized. The error process depends on the underlying state of the Markov process. Following the notation of [19], we denote the state process as $\{s_l\}_{l=0}^{\infty}$, where each $s_l \in \{G, B\}$. The state process is a stationary first order Markov process, that is

$$\mathbb{P}[s_l|\mathbf{s_{l-1}}] = \mathbb{P}[s_l|s_{l-1}] \tag{3}$$

where $\mathbf{s_{l-1}} = (s_{l-1}, s_{l-2}, \ldots, s_1)$. For the sensor-controller channel the transition probabilities for the chain are given as

$$\begin{aligned} \phi_{sc,21} &= \mathbb{P}[s_{l,sc} = G|s_{(l-1),sc} = B], \\ \phi_{sc,12} &= \mathbb{P}[s_{l,sc} = B|s_{(l-1),sc} = G] \end{aligned} \tag{4}$$

The initial distribution of the state process is taken to be the stationary distribution of the Markov chain and is given as

$$\begin{aligned} \mathbb{P}[s_0 = G] &= \frac{\phi_{sc,21}}{\phi_{sc,21} + \phi_{sc,12}}, \\ \mathbb{P}[s_0 = B] &= \frac{\phi_{sc,12}}{\phi_{sc,21} + \phi_{sc,12}} \end{aligned} \tag{5}$$

A similar Markov chain can be defined for the controller-actuator channel. The corresponding state transition probabilities are given by the transition probability matrix $\Phi_{ca} = [\phi_{ca,ij}]_{2\times 2}$. Note that the results of this paper can be extended trivially to the case where the drops in the sensor-controller channel are correlated with the drops in the controller-actuator channel. In that case, we only have one Markov chain with four states. However, for ease of exposition, we restrict ourselves to the case where the drops in the sensor-controller channel are independent of the drops in the controller-actuator channel.

As in [19] we define *channel memory* m as

$$m \triangleq 1 - \phi_{sc,21} - \phi_{sc,12}. \tag{6}$$

When $m = 0$, the channel is memoryless and the losses are independent. When $m > 0$, the channel has persistent memory, i.e., the probability of remaining in any given state is higher that its steady state probability.

The measurements are processed and transmitted over a sensor-controller channel to the controller. For the sensor-controller channel, we use the optimal strategy identified in [10]. At time step k, the controller calculates $N+1$ control inputs [3] $u_k^k, u_{k+1}^k, \cdots, u_{k+N}^k$ to minimize the finite-horizon cost function

$$J_K = \sum_{k=0}^{K} E\left[x_k^T Q x_k + u_k^T R u_k\right] + E\left[x_{K+1}^T P_{K+1} x_{K+1}\right]. \tag{7}$$

The control inputs are transmitted over a controller-actuator channel to the actuator. The actuator maintains a buffer of length $N+1$ control inputs. If the actuator receives a packet, it erases all the information stored in the buffer previously and stores all the control inputs contained in the latest packet in the buffer. If it does not receive a packet, it does not change the buffer contents. To choose the control input at time step k, the actuator carries out two actions:

(1) It picks the control input u_k^j for any $j \leq k$ if it exists in the buffer.
(2) If such an input does not exist, it arbitrarily picks the value 0 [4].

Thus the control input u_{k+i}^k calculated by the controller at time step k corresponds to the optimal control input to be applied at time step $k+i$ that the controller can calculate at time step k. The actuator applies the appropriate control input to the process according to equation (1). We assume that an acknowledgment is available to the controller at every time step whether or not the packet it transmitted at that time step was received at the actuator end. Thus the controller knows the control input that the actuator applies at all times. Apart from the system matrices and the cost function, the control inputs u_i^k are constrained to be a function of the messages received over the sensor-controller channel up until time k and the control inputs applied to the system up until time $k-1$.

The length $N+1$ of the trajectory transmitted by the controller is the largest length allowed by the data packet

[3] N in some sense corresponds to the horizon over which the trajectory is calculated in receding horizon control.

[4] We use the value 0 for simplicity. It may also correspond to any 'fail-safe' mode input

length (subject to our quantization related assumptions) and by the length of the buffer at the actuator. As discussed above, in most communication protocols, the packet length is more than required for transmitting a single control input. Thus it is usually possible to append future control packets to every transmission. Nevertheless, the maximum value of N will depend on the specific protocol being used. Moreover, more control inputs communicated translates to more power expended. Note that since the controller transmits only one packet at every time step, there is no benefit to be incurred by, e.g., transmitting the information u_k^k multiple times.

Two questions arise naturally.

(1) How to design the control inputs $\{u_k^j\}$?
(2) How does the performance of the system vary as a function of N?

We now proceed to answer these questions.

3 Analysis

3.1 Markov Chain Model

We will find it convenient to define two more Markov chains. When the controller transmits N future control inputs (thus the packet contains u_k^k, u_{k+1}^k, $\cdots$, u_{k+N}^k at time step k), there are $N+2$ states in the Markov chains. The first $N+1$ states correspond to the last packet being received at the actuator t time steps ago, for values of $t = 0, 1, \cdots, N$ while the $N+2$-th state corresponds to the last packet being received t time steps ago, where $t \geq N+1$. For the first Markov chain, the (i, j)-th element of the transition probability matrix Φ_{for} is defined as

$$\phi_{for,ij} = \text{Prob}\left(\text{state } j \text{ at time } k+1 | \text{state } i \text{ at time } k\right).$$

The matrix Φ_{for} can then be calculated as

$$\Phi_{for} = \begin{bmatrix} \phi_{ca,11} & \phi_{ca,12} & 0 & 0 & \cdots & 0 \\ \phi_{ca,21} & 0 & \phi_{ca,22} & 0 & \cdots & 0 \\ \phi_{ca,21} & 0 & 0 & \phi_{ca,22} & 0 & \vdots \\ \vdots & \vdots & & \ddots & \ddots & \ddots \\ \phi_{ca,21} & 0 & \cdots & & 0 & \phi_{ca,22} \end{bmatrix}.$$

For simplicity, we assume that none of the elements $\phi_{ca,ij}$ are zero, thus the Markov chain described by Φ_{for} reaches a unique stationary distribution. Using this Markov chain, we can define another secondary Markov chain that corresponds to the system evolving *backwards* in time. Thus the (i, j)-th element of the transition probability matrix Φ_{back} is defined as

$$\phi_{back,ij} = \text{Prob}\left(\text{state } j \text{ at time } k | \text{state } i \text{ at time } k+1\right).$$

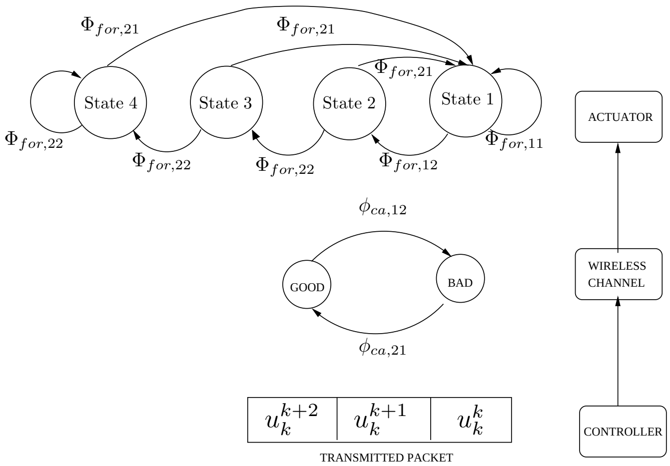

Fig. 3. Pictorial Representation of Receding Horizon Networked Control. Here the controller transmits packets with $N = 2$ future control input. Thus the Markov chain at the actuator has 4 states. *State 1* corresponds to packet being received at the actuator. In this state, the actuator applies the current input and flushes the buffer. In *state 2* and *state 3*, packet is not received at the controller and control input from previously received packet is applied. *State 4* corresponds to the case where the packet is not received and control input of 0 is applied.

The matrix Φ_{back} can be easily calculated using Φ_{for}. From now on, unless otherwise stated, the word Markov chain in the analysis will refer to this Markov chain with transition probability matrix Φ_{back}. For ease of notation, we will denote $\phi_{back,ij}$ as q_{ij}. Also, we will denote the probability of being in state j at time step k by π_k^j. Using the Markov chain model, a pictorial representation of our scheme is presented in figure 3. Here the controller transmits a packet with $N = 2$ future control inputs. The channel drops packets according to a Markov chain. At the actuator, there are 4 states of a Markov chain. State 1 corresponds to the case where the channel is in *good* state and hence the packet is received at the actuator. The actuator applies the latest control input and flushes the buffer. States 2, 3 and 4 correspond to the case where the packet is lost by the channel. In state 2 and 3, the actuator applies the control input from previously received packet. In state 4, a control input of 0 is applied since there have been 3 consecutive packet losses. In the next subsection, we use this Markov chain model to analyze the cost function and derive the optimal control values.

3.2 A Closer Look at the Cost Function

We begin by extracting the terms in the cost function dependent on x_K and u_K. We can write them as

$$T_K = E\left[x_K^T Q x_K + u_K^T R u_K + x_{K+1}^T P_{K+1} x_{K+1}\right]. \quad (8)$$

We can condition T_K on the event that the Markov state is in the state i at time step K. Let us denote this event by $m_K = i$. Thus

$$T_K = \sum_{i=1}^{N+2} \pi_K^i T_K^i, \tag{9}$$

where we have defined

$$T_K^i = E\left[x_K^T Q x_K + u_K^T R u_K + x_{K+1}^T P_{K+1}^i x_{K+1} | m_K = i\right], \tag{10}$$

and represented the quantity P_{K+1} entering the i-th term in the summation as P_{K+1}^i. The state i determines the control input u_K. For $i = 1, \cdots, N+1$, the actuator applies the control input retrieved from the buffer and hence

$$u_K = u_K^{K-i+1}.$$

For the state $i = N+2$, the actuator buffer is empty and hence the control input applied is $u_K = 0$.

To see what the terms u_K^{K-i+1} should be, let us isolate the corresponding term from the summation. We can write the term T_k^i as

$$\begin{aligned}
T_k^i &= E\left[x_K^T Q x_K + u_K^T R u_K + x_{K+1}^T P_{K+1}^i x_{K+1} | m_K = i\right] \\
&= E\Big[x_K^T Q x_K + u_K^T R u_K + (A x_K + B u_K^{K-i+1} + w_K)^T \\
&\quad P_{K+1}^i (A x_K + B u_K^{K-i+1} + w_K)\Big] \\
&= E\Big[(u_K^{K-i+1} + (S^i)^{-1} B^T P_{K+1}^i A x_K)^T S^i \\
&\quad (u_K^{K-i+1} + (S^i)^{-1} B^T P_{K+1}^i A x_K) + w_K^T P_{K+1}^i w_K + \\
&\quad x_K^T (Q + A^T P_{K+1}^i A - P_{K+1}^i B (S^i)^{-1} B^T P_{K+1}^i) x_K\Big],
\end{aligned}$$

where we have used the fact that the noise w_K is zero mean and have denoted

$$S^i = R + B^T P_{K+1}^i B.$$

Thus it is apparent that u_K^{K-i+1} should be chosen so as to minimize the mean squared error

$$E\Big[(u_K^{K-i+1} + (S^i)^{-1} B^T P_{K+1}^i A x_K)^T S^i (u_K^{K-i+1} + (S^i)^{-1} B^T P_{K+1}^i A x_K)\Big]. \tag{11}$$

Thus at time step $K - i + 1$, the controller should calculate the minimum mean squared estimate of x_K and then multiply it by the matrix $(S^i)^{-1} B^T P_{K+1}^i A$ to determine u_K^{K-i+1}. Let us denote the corresponding error covariance incurred by Δ_K^i. Note that while calculating u_K^{K-i+1}, the controller knows all the control inputs applied until time step $K-i$. Further, if the input u_K^{K-i+1} is used at time step K, the controller knows that no packet was transferred over the controller-actuator channel successfully after time step $K-i+1$. Hence it can also determine the control inputs applied from time $K-i+1$ until time $K-1$. Thus the controller knows all the previous control inputs while estimating x_K. Hence, Δ_K^i is independent of all previous control inputs.

With the optimizing choice of u_K^{K-i+1}, the term T_K^i becomes

$$T_K^i = \Delta_K^i + E\Big[w_K^T P_{K+1}^i w_K + x_K^T (Q + A^T P_{K+1}^i A - A^T P_{K+1}^i B (S^i)^{-1} B^T P_{K+1}^i A) x_K\Big]. \tag{12}$$

For ease of notation, for the values of $i = 1, \cdots, N+1$, let us define an operation $f^i(.)$ as

$$f^i(X) = Q + A^T X A - A^T X B (R + B^T X B)^{-1} B^T X A. \tag{13}$$

Thus

$$T_K^i = \Delta_K^i + E\left[w_K^T P_{K+1}^i w_K + x_K^T f^i\left(P_{K+1}^i\right) x_K\right]. \tag{14}$$

This form of T_K^i holds for $i = 1, \cdots, N+1$. For $i = N+2$, $u_K = 0$ and

$$\begin{aligned}
T_K^{N+2} &= E\left[x_K^T Q x_K + (A x_K + w_K)^T P_{K+1}^i (A x_K + w_K)\right] \\
&= E\left[w_K^T P_{K+1}^i w_K + x_K^T (Q + A^T P_{K+1}^i A) x_K\right].
\end{aligned}$$

Thus for the optimizing choice of control inputs at time step $K+1$, we can finally write

$$\begin{aligned}
T_K &= \sum_{i=1}^{N+2} \pi_K^i T_K^i \\
&= \sum_{i=1}^{N+2} \pi_K^i E\left[w_K^T P_{K+1}^i w_K\right] + \sum_{i=1}^{N+2} \pi_K^i \Delta_K^i \\
&\quad + \sum_{i=1}^{N+2} \pi^i(K) E\left[x_K^T f^i\left(P_{K+1}^i\right) x_K\right],
\end{aligned} \tag{15}$$

where we have defined

$$\Delta_K^{N+2} = 0 \qquad f^{N+2}(X) = Q + A^T X A.$$

The cost function after choosing the control inputs at time K optimally can thus be rewritten as

$$J_K = \sum_{i=1}^{N+2} \pi^i(K) E\left[x_K^T f^i\left(P_{K+1}^i\right) x_K\right] + \sum_{i=1}^{N+2} \pi_K^i \Delta_K^i + \sum_{i=1}^{N+2} \pi_K^i E\left[w_K^T P_{K+1}^i w_K\right] + \sum_{k=0}^{K-1} E\left[x_k^T Q x_k + u_k^T R u_k\right]. \tag{16}$$

The third summation involves only the noise terms and thus cannot be affected by the choice of the control inputs. The second summation involves the estimation error covariance incurred while calculating u_K, and as explained earlier, that term is also independent of all previous control input choices. Thus to optimally choose all the control inputs from time 0 to time $K-1$, we only need to consider the first and the fourth summations. Let us take a closer look at the first summation and denote it by Γ_K. We have

$$
\begin{aligned}
\Gamma_K &= \sum_{i=1}^{N+2} \pi^i(K) E\left[x_K^T f^i\left(P_{K+1}^i\right) x_K\right] \\
&= \sum_{i=1}^{N+2} \sum_{j=1}^{N+2} \pi_K^i q_{ij} E\left[x_K^T f^i\left(P_{K+1}^i\right) x_K | m_{K-1}=j\right] \\
&= \sum_{j=1}^{N+2} \sum_{i=1}^{N+2} \pi_K^i q_{ij} E\left[x_K^T f^i\left(P_{K+1}^i\right) x_K | m_{K-1}=j\right] \\
&= \sum_{j=1}^{N+2} E\left[x_K^T \left(\sum_{i=1}^{N+2} \pi_K^i q_{ij} f^i\left(P_{K+1}^i\right)\right) x_K | m_{K-1}=j\right].
\end{aligned}
$$

Let us define the matrices P_K^j through the equations

$$
\pi_{K-1}^i P_K^j = \sum_{i=1}^{N+2} \pi_K^i q_{ij} f^i\left(P_{K+1}^i\right).
$$

Thus

$$
\begin{aligned}
\Gamma_K &= \sum_{j=1}^{N+2} E\left[x_K^T \left(\pi_{K-1}^j P_K^j\right) x_K | m_{K-1}=j\right] \\
&= \sum_{j=1}^{N+2} \pi_{K-1}^j E\left[x_K^T P_K^j x_K | m_{K-1}=j\right]. \qquad (17)
\end{aligned}
$$

We can ignore the second and the third terms in equation (16) since these summations do not play any role in further minimization. We can thus write the cost function as

$$
\begin{aligned}
\tilde{J}_K &= \sum_{k=0}^{K-1} E\left[x_k^T Q x_k + u_k^T R u_k\right] \\
&+ \sum_{j=1}^{N+2} \pi_{K-1}^j E\left[x_K^T P_K^j x_K | m_{K-1}=j\right], \qquad (18)
\end{aligned}
$$

We can again extract the term T_{K-1} in a form similar to the one in (9) since our argument does not not rely on the time index K. Thus we can carry out a similar argument to evaluate the optimal control inputs at all time steps and the resulting cost function. All that remains is to specify the terms P_k^j for any time k. For these terms, we can identify the recursions according to which these terms evolve. These terms evolve *backwards* in time according to the following coupled equations:

$$
\pi_{k-1}^i P_k^j = \sum_{i=1}^{N+2} \pi_k^i q_{ij} f^i\left(P_{k+1}^i\right), \qquad (19)
$$

where π_k^j is the probability of being in state j at time k, q_{ij} is the transition probability of being in state j at time step $k-1$ given that the state at time state k was i and the operators $f^i(.)$ have been defined earlier. The initial values are $P_{K+1}^i = P_{K+1}, \forall i$.

We have, in effect, proved a separation principle in the problem setting we are considering.

Proposition 1 (Separation Principle) *Consider the problem setting described in Section 2. The optimal value of the control input u_k^{k-i+1}, i.e., the control input corresponding to time k included in the packet transmitted by the controller at time step $k-i+1$ is given by*

$$
u_k^{k-i+1} = \left(R + B^T P_{k+1}^i B\right)^{-1} B^T P_{k+1}^i A \hat{x}_{k|k-i+1},
$$

where $\hat{x}_{k|k-i+1}$ is the minimum mean squared estimate of the state x_k that the controller can calculate given all the received measurements till time step $k-i+1$ and the control inputs until time step $k-1$ and the terms P_k^i evolve according to the coupled recursions (19).

3.3 Stability of the Cost Function

We can consider the optimal cost as the time horizon K becomes larger. For the infinite time horizon problem, we will consider the cost

$$
J_\infty = \lim_{K\to\infty} \frac{1}{K} J_K.
$$

If this cost is bounded,we will say that the system is stable. Looking at the analysis in Section 3.2, there are two reasons that the cost can grow unbounded:

(1) The terms Δ_k^i grow unbounded. This is a function of the reliability of the sensor-controller channel.
(2) The terms P_0^j grow unbounded. This is a function of the reliability of the controller-actuator channel.

We will now consider these two effects.

Let us begin with the terms Δ_k^i. For $i = 1, \cdots, N+1$, the terms Δ_k^i represent the estimation error covariance incurred when the control input for time step k is calculated at time $k-i+1$. Also, by definition, $\Delta_k^{N+2} = 0$. The only term whose stability we need to consider is the term Δ_k^1. The other error terms Δ_k^i's for $i > 1$ are affine functions of the term Δ_k^1 and cannot grow unbounded as long as Δ_k^1 is finite. Intuitively this makes sense since the fundamental quantity being estimated at any time $k-i+1$ is the state of the system at all time steps from $k-i+1$ to k. Since the control input has no effect on the estimation error covariance, the error covariance involved in estimating the state cannot grow unbounded as long as N is finite. Since, by assumption the pair (A, C) is observable, the only way for the error covariance Δ_k^1 to grow unbounded is if the sensor-controller channel loses measurements at a high enough rate. The problem of considering the stability of Δ_k^1 is thus identical to the problem of stability of the estimate of a dynamic process over a packet dropping channel. If the sensor were transmitting measurements, the stability and performance could be upper and lower bounded using techniques in, e.g., [24]. If we are doing the optimal pre-processing of information prior to transmission, the analysis of [10] is applicable.

For the terms P_k^j, we need to look at the recursions (19). The behavior of these equations can be analyzed using results from [4,7]. We thus have the following result.

Proposition 2 (Stability Conditions) *The system is stable if and only if both of the terms Δ_k^i's and P_k^j's are stable.*

(1) A necessary and sufficient condition for the stability of the terms Δ_k^i is that the probability of two consecutive drops $\phi_{sc,22}$ and the spectral radius of matrix A should satisfy the relation

$$\phi_{sc,22} \mid \rho(A) \mid^2 < 1. \tag{20}$$

(2) Assume that the Markov chain transition probability matrix Φ_{back} is such that the states reach a stationary probability distribution with the probability of being in the j-th state as π^j. Further assume that all π^j's are strictly positive. If there exist $N+2$ positive definite matrices X^1, X^2, $\cdots$, X^{N+2} and $(N+2)^2$ matrices $K^{1,1}$, $K^{1,2}$, $\cdots$, $K^{1,N+2}$, $K^{2,1}$, $\cdots$, $K^{N+2,N+2}$ such that

$$\pi^j X^j > \sum_{i=1}^{N+2} \Big((A^T + K^{ij} B^T) X^i (A^T + K^{ij} B^T)^* + R_w + K^{ij} R_v (K^{ij})^* \Big) q_{ij} \pi^i,$$

then the terms P_j^k's converge for all initial conditions $X_{K+1}^i > 0$ and the limit $\bar{X}^j$ is the unique positive semi-definite solution of the equation

$$\pi^j X^j = \sum_{i=1}^{N+2} f^j \left(X^i \right) q_{ij} \pi^i. \tag{21}$$

Let the probability of choosing the $N+2$-th state consecutively be $q_{N+2,N+2}$. Denote $\rho(A)$ as the spectral radius of matrix A. Then a sufficient condition for the expected estimate error covariance to diverge from at least one initial value is given by

$$q_{N+2,N+2} |\rho(A)|^2 > 1. \tag{22}$$

Note that both (20) and (22) are independent of the value of N.

3.4 Performance Analysis

We can also analyze the performance of the system in terms of the value of the cost function achieved. For the sake of illustration, we state the results for the infinite-horizon case in which we are interested in the cost J_∞ when the packet drops over the sensor-controller channels happen in an independent and identically distributed fashion with probability p at every time step. Assuming that the stability conditions are satisfied, the terms Δ_k^i and P_0^i will reach a steady state. Denote the steady state values of the terms as Δ^i and P^i respectively. Using the results from [10] and [7], we have the following result.

Proposition 3 (Performance Analysis) *The term Δ^1 satisfies the Lyapunov equation*

$$\Delta^1 = \sqrt{p} A \Delta^1 A^* \sqrt{p} + R_w + (1-p) A \Delta^\star A^*,$$

where $\Delta^\star$ denotes the steady-state error covariance incurred when estimating x_k given all the measurements $\{y_j\}_{j=0}^{k-1}$ and the past control inputs applied to the system. The terms Δ^i for $i = 2, \cdots, N+1$ can also be bounded using the relation

$$\Delta^i = A \Delta^{i-1} A^* + R_w.$$

Finally the term Δ^{N+2} is zero by definition.

Similarly, call the stationary probability of the Markov chain Φ_{back} being in state i as π^i. Then the terms P^i are found from the coupled equations

$$\pi^i P^j = \sum_{i=1}^{N+2} \pi^i q_{ij} f^i \left(P^i \right),$$

where the operator $f^i(.)$ has been defined already.

The cost function J_∞ can be calculated as

$$J_\infty = \sum_{i=1}^{N+2} \pi^i \, \mathit{Trace} \left(P^i R_w \right) + \sum_{i=1}^{N+2} \pi^i \Delta^i.$$

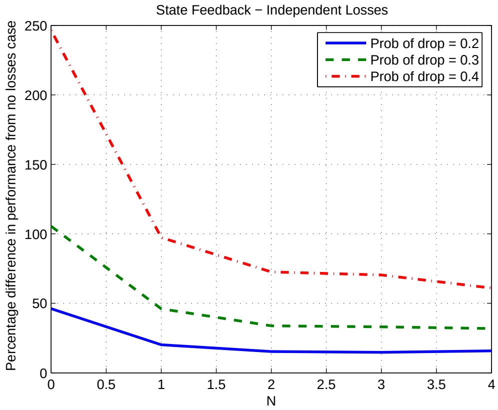

Fig. 4. Improvement in LQG Cost with N - State Feedback and Independent Losses

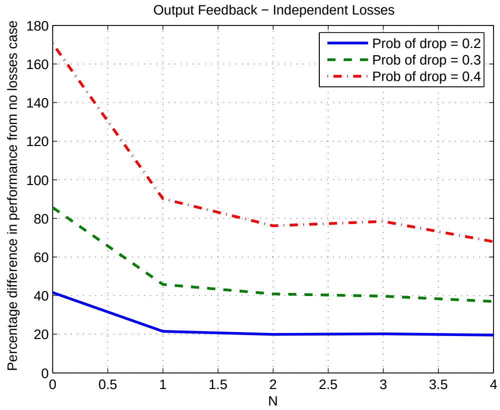

Fig. 5. Improvement in LQG Cost with N - Output Feedback and Independent Losses

Note that the Markov chain Φ_{back} is positive recurrent and irreducible and hence a unique stationary probability exists.

4 Examples

In this section we present numerical examples which demonstrate the benefit of sending multiple control inputs per packet. We consider the system defined in equations (1) and (2) with the following parameters:

$$A = \begin{bmatrix} 1.2 & 0.22 \\ 0 & 1.2 \end{bmatrix} \quad B = \begin{bmatrix} 0.02 \\ 0.2 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

The noise processes v_t and w_t are zero mean, with variance $R_v = 0.5$ and $R_w = I_{2\times 2}$ respectively. The controller is a LQG controller which minimizes the finite horizon cost function given by

$$J_K = \sum_{k=0}^{K} E\left[x_k^T Q x_k + u_k^T R u_k\right] + E\left[x_{K+1}^T P_{K+1} x_{K+1}\right]$$

The horizon length K is chosen to be 30 and the cost weight matrices $Q = 0.2 I_{2\times 2}$ and $R = 0.4$. The final state cost matrix $P_{K+1} = Q$.

Figure 3 plots the percentage improvement in the LQG cost over the lossless case vs. different numbers of control inputs per transmission for different probabilities of packet drops. The packet drops are independent of each other. As can be observed from the graph, the LQG costs decreases as the number of control inputs is increased and it soon flattens out. This is because for low probability of drop, there is small chance of losing many consecutive packets and hence adding more control inputs does not improve the cost. However, the performance improves significantly with quite moderate values of N. Moreover, as the probability of drop increases, the value of N at which the curve 'flattens out' also increases. Similar effects can be seen for the output feedback case (Figure 4). Once again, the performance gains are significant, although it is less than the gain obtained with full state feedback in this example.

As observed in previous figures, the effect of increasing N is more pronounced if there are several consecutive losses. Such situations frequently exist in wireless channels where if the channel enters a fade, it stays in a bad state for a long time. Due to such behavior, the packet loss occurs in a bursty fashion. Markov chain based models are used to model such channels. We analyze the improvement in LQG cost when the packet loss process is governed by the Gilbert-Elliott channel model. We choose the steady state probability of being in a good state (as given in Eq 5) to be 0.2 and the channel memory as defined in Eq. 6 is varied from 0.2 to 0.4. From Figure 5 and Figure 6, we note that as the channel memory is increased, the difference in LQG cost between lossy and lossless case is increased. Also since the correlated losses lead to frequent bursts of error, the cost flattens out at higher values of N as compared to independent losses.

The previous figures considered the cases where only the controller-actuator channel has losses. We now consider the scenario where both the sensor-controller and controller-actuator channels have independent losses. Note that the losses are independent over channels and also over time. The optimal encoding strategy for the sensor-controller channel is the one given in [10]. Figure 7 plots the percentage gain in LQR cost when both the sensor-controller and the controller-actuator have independent losses. As is evident from the graph, there is a significant improvement in the cost with increasing N even for the case where sensor measurements are dropped. The improvement in the LQR cost is more pronounced when there is a higher probability of loss in the sensor-controller channel.

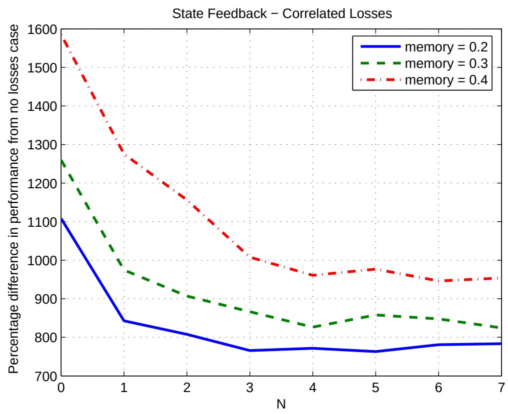

Fig. 6. Improvement in LQG Cost with N - State Feedback and Correlated Losses

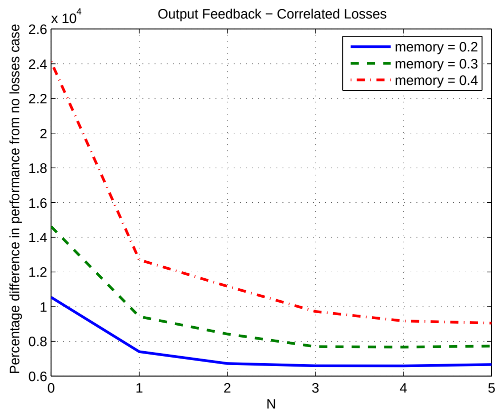

Fig. 7. Improvement in LQG Cost with N - Output Feedback and Correlated Losses

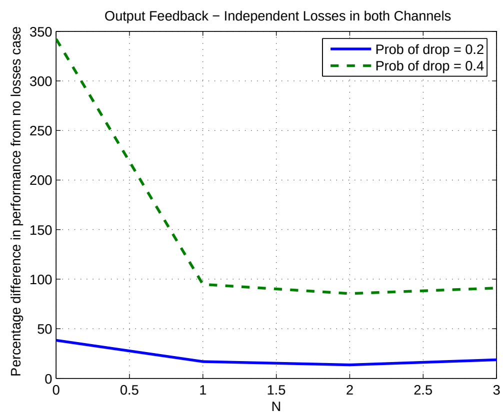

Fig. 8. Improvement in LQG Cost with N - Output Feedback and Independent Losses in both Channels

5 Conclusions and Future Work

Motivated by receding horizon control, we analyzed a networked control system which used actuation buffers and transmission of future control trajectories. Using a separation principle, we showed how to optimally compute the trajectory and also carried out a stability and performance analysis. Simulations showed that a fairly small number of future control inputs transmitted can significantly improve the performance of the system. The work reinforces the general idea that pre-processing and transmitting extra information to utilize the data packet to the maximum is an important design principle for networked control systems.

This work is but a first step toward receding horizon networked control. The full power of receding horizon control is realized when the system model is not fully known or there are dynamic constraints on the state and the inputs. It will be interesting to consider these more general cases. Also, obtaining the optimal quantities for the controller to transmit when the actuator is slightly more 'intelligent' is also an important avenue of research.

References

[1] B. Azimi-Sadjadi, *"Networked Control Systems with Packet Losses"*, Systems and Control Letters, (submitted 2005).

[2] P. Antsaklis and J. Baillieul (Editors), *Special Issue on Networked Control Systems*, IEEE Trans. Automat. Contr., 49(9), Sep 2004.

[3] L. Bushnell (Editor), *Special Issue on Networks and Control*, IEEE Control Systems Magazine, 21(1), Feb 2001.

[4] O. L. V. Costa and M. D. Fragoso, *"Stability results for discrete-time linear systems with Markovian jumping parameters"*, J. Mathematical Anal. and Applications, 179:154-178, 1993.

[5] L. B. Cremean, T. B. Foote, J. H. Gillula, G. H. Hines, D. Kogan, K. L. Kriechbaum, J. C. Lamb, J. Leibs, L. Lindzey, C. E. Rasmussen, A. D. Stewart, J. W. Burdick and R. M. Murray, *"Alice: An Information-Rich Autonomous Vehicle for High-Speed Desert Navigation"*, To appear, Journal of Field Robotics, 2006.

[6] S. Graham, G. Baliga and P. R. Kumar, *"Issues in the convergence of control with communication and computing: Proliferation, architecture, design, services, and middleware"*, Proceedings of the 43rd IEEE Conference on Decision and Control, Bahamas, 2004, pp. 1466-1471.

[7] V. Gupta, T. Chung, B. Hassibi and R. M. Murray, "On a Stochastic Sensor Selection Algorithm with Applications in Sensor Scheduling and Dynamic Sensor Coverage," *Automatica*, Vol. 42, No. 2, February 2006, Pages: 251-260.

[8] V. Gupta, A. F. Dana, J. P. Hespanha, R. M. Murray and B. Hassibi, *"Data Transmission over Networks for Estimation and Control"*, IEEE Transactions on Automatic Control, Submitted 2006.

[9] V. Gupta and N. Martins, *"Optimal Stability Regions for Estimation using Multiple Sensors"*, Automatica, Submitted 2006.

[10] V. Gupta, D. Spanos, R. M. Murray and B. Hassibi, *"Optimal LQG Control across Packet Dropping Links"*, Systems and Control Letters, (Accepted 2006).

[11] D. Georgiev and D. M. Tilbury, *"Packet-based Control: The H_2-optimal Solution"*, Automatica, 42(1), January 2006, pp. 137-144.

[12] C. N. Hadjicostis and R. Touri, *"Feedback control utilizing packet dropping network links"*, Proc. IEEE Conference on Decision and Control, 2002.

[13] O.C. Imer, S. Yuksel and T. Basar, *"Optimal Control of LTI Systems over Communication Networks"*, Automatica, 42(9):1429-1440, September 2006.

[14] P. A. Kawka and A. G. Alleyne, *"Stability and performance of packet-based feedback control over a Markov channel"*, Proceedings of the American Control Conference, 2006, pp. 2807-2812.

[15] Q. Ling and M. D. Lemmon, *"Power spectral analysis of Networked Control Systems with Data Dropouts"*, IEEE Trans. Automat. Contr., 49(6), June 2004, pp. 955-960.

[16] F.L. Lian, J.R. Moyne and D.M. Tilbury, *"Performance evaluation of control networks"*, IEEE Control Systems Magazine 21 (2001) (1), pp. 6683.

[17] L. A. Montestruque and P. Antsaklis, *"On the Model-based Control of Networked Systems"*, Automatica, 39(10), October 2003, pp. 1837-1843.

[18] L. A. Montestruque and P. Antsaklis, *"Stability of Model-based Networked Control Systems with Time-varying Transmission Times"*, IEEE Transactions on Automatic Control, 49(9), September 2004, pp. 1562-1572.

[19] M. Mushkin and I. Bar-David, *"Capacity and Coding for Gilbert Elliott Channels"*, IEEE Transactions on Information Theory, 35(6), November 1989, pp. 12771290.

[20] J. Nilsson, *"Real-Time Control Systems with Delays"*, PhD Thesis, Department of Automatic Control, Lund Institute of Technology, 1998.

[21] P. Naghshtabrizi and J. Hespanha, *"Anticipative and Non-anticipative Controller Design for Network Control Systems"*, In Panos J. Antsaklis, Paulo Tabuada, Networked Embedded Sensing and Control, volume 331 of Lect. Notes in Contr. and Inform. Sci., pages 203218, 2006.

[22] P. Seiler, *"Coordinated Control of unmanned aerial vehicles"*, PhD Thesis, University of California, Berkeley, 2001.

[23] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla and S. S. Sastry, *"Foundations of Control and Estimation over Lossy Networks"*, Proc. of the IEEE, To Appear 2006.

[24] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, S. Sastry and M. Jordan, "Kalman Filtering with intermittent observations", *IEEE Transactions on Automatic Control*, 49(9), September 2004.

[25] W. Zhang, M. S. Branicky and S. M. Philips, *"Stability of Networked Control Systems"*, IEEE Control System Magazine, 21(1), Feb 2001, pp. 84-89.