Forty-Fourth Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
Sept 27-29, 2006

Receding Horizon Networked Control

Vijay Gupta, Bruno Sinopoli, Sachin Adlakha, Andrea Goldsmith, and Richard Murray

***Abstract*— This paper deals with the design of control systems over lossy networks. A network is assumed to exist between the sensor and the controller and between the latter and the actuator. Packets are dropped according to a Bernoulli independent process, with γ and μ being the probabilities of losing an observation packet and a control packet respectively, at time any instant t. A receding horizon control scheme is proposed for the Linear Quadratic Control (LQG) problem. At each instant N future control inputs are sent in addition to the current one. Under this scheme the separation of estimation and control is shown and stability conditions, dependent on loss probabilities, are provided. Simulations show how the overall performance, in terms of lower cost, increases with the length of the horizon.**

I. INTRODUCTION

The last two decades have witnessed the widespread adoption of new communicating technologies. Over the years there has been a surge of new applications. Recently wireless sensor actuator networks have enabled a series of new applications ranging from monitoring systems, such as environments, structures, traffic, to security systems. The application of networking technology to time and safety critical applications, such as estimation and control of dynamical systems, has yet to show its full potential. This is due mainly to the degree of reliability these systems require. Engineers are not so keen to design and implement control systems unless they can guarantee to meet the required reliability specifications. Guaranteeing deterministic behavior was traditionally implemented by imposing a deterministic system at the physical layer. In order to clarify this concept, let us consider the example of embedded systems in the automotive. Until recently, every subsystem in a car was a stand-alone, with its own sensors, actuators, computational units and communication infrastructure. While guaranteeing an acceptable level of reliability, this resulted into cars having several miles of wiring and tens of embedded processors, increasing the vehicle's manufacturing cost and final weight. The introduction of the Controller Area Network (CAN) bus, a standard communication shared by several subsystems, represents a step in right direction. With the massive introduction of electronics not only in cars, but more generally in the physical world, this trend is only likely to continue. Sharing computational and communication resources is the only viable solution.

Vijay Gupta and Richard Murray are with the Department of Electrical Engineering at the California Institute of Technology, Pasadena, CA {vijay, richard}@cds.caltech.edu

Bruno Sinopoli, Sachin Adlakha and Andrea Goldsmith are with the Department of Electrical Engineering at Stanford University, Stanford, CA {sinopoli, adlakha, andrea}@ee.stanford.edu

Certainly the use of networking technology for time-critical applications provides several advantages:

1) The availability of more information allows the designer to employ a more elaborate model of system. At the spatial level, interacting systems can be jointly controlled, when previously interaction was modeled as disturbance on each subsystem.
2) A networking technology allows the connection of spatially distributed systems.
3) Many systems can share the same infrastructure, thus decreasing overall implementation cost.

As usual, opportunities come with challenges. The presence of a common infrastructure introduces a new source of uncertainty since both communication and consequently data delivery are not deterministic. Phenomena such as congestion and interference introduce random delays and loss of information. While system and control theory provide a wealth of analytical results, the assumptions that the theory is traditionally based upon, i.e. timely availability of data, do not hold true in this setting, and neglecting these phenomena may yield to catastrophic system performance. Specific questions that arise are the following. What is the amount of data loss that the control loop can tolerate to reliably perform its task? Can communication protocols be designed to satisfy this constraint? Answering these questions requires a generalization of classical control techniques that explicitly take into account the stochastic nature of the communication channel. More generally, the use of networks in control systems imposes a paradigm shift. Deterministic methods need to be replaced by stochastic ones, as such is the nature of the network phenomena. This argument is particularly true in wireless networks, where the use of a shared channel with random disturbances and noise cannot be modeled deterministically.

In this paper we concentrate on the problem of design analysis and control of a dynamical system where information travels across lossy packet networks. Packet network communication channels typically use one of two fundamentally different protocols: TCP-like or UDP-like. In the first case there is acknowledgement of received packets, while in the second case no-feedback is provided on the communication link. The well known Transmission Control (TCP) and User Datagram (UDP) Protocols used in the Internet are specific examples of our more general notion of TCP-like and UDP-like communication protocol classes.

Study of stability of dynamical systems where components are connected asynchronously via communication channels has received considerable attention in the past few years and our contribution can be put in the context of the previous literature. In [6] and [18], the authors proposed to place an estimator, i.e. a Kalman filter, at the sensor side of the link without assuming any statistical model for the data loss process. In [16], Smith *et al.* considered a suboptimal but computationally efficient estimator that can be applied when the arrival process is modeled as a Markov chain, which is more general than a Bernoulli process. Other works include Nilsson *et al.* [10][11] who present the LQG optimal regulator with bounded delays between sensors and controller, and between the controller and the actuator. In this work, bounds for the critical probability values are not provided and there is no analytic solution for the optimal controller. The case where dropped measurements are replaced by zeros is considered by Hadjicostis and Touri [7], but only in the scalar case. Other approaches include using the last received sample for control [11], or designing a dropout compensator [8], which combines estimation and control in a single process. However, the former approach does not consider optimal control and the latter is limited to scalar systems. Yu *et al.* [19] studied the design of an optimal controller with a single control channel and deterministic dropout rates. Seiler *et al.* [14] considered Bernoulli packet losses only between the plant and the controller and posed the controller design as an H_∞ optimization problem. Other authors [12] [2] [1] [17] model networked control systems with missing packets as Markovian jump linear systems (MJLSs), however this approach gives suboptimal controllers since the estimators are stationary. Finally, Elia [4][3] proposed to model the plant and the controller as deterministic time invariant discrete-time systems connected to zero-mean stochastic structured uncertainty. The variance of the stochastic perturbation is a function of the Bernoulli parameters, and the controller design is posed an an optimization problem to maximize the mean-square stability of the closed loop system. This approach allows analysis of Multiple Input Multiple Output (MIMO) systems with many different controller and receiver compensation schemes [4], however, it does not include process and observation noise and the controller is restricted to be time-invariant, hence sub-optimal.

In this paper we build upon the results contained in [13] and [5] in particular. [13] studies the LQG problem under both TCP-like and UDP-like protocols, modeling the successful delivery of the current observation and control packet as two Bernoulli independent processes γ_t and ν_t, respectively. For the TCP-like case, the authors prove the separation of estimation and control holds for the TCP-like case, showing that the optimal estimator is the time-varying Kalman filter and that the optimal LQG controller is linear. In the case of an open loop unstable system it is shown that, in the infinite horizon case, there exist critical arrival probabilities γ_c and ν_c above which the system is able to regulate the system. Furthermore, if the arrival rates for either the observation or the control packet are below the critical ones, the system goes unbounded. [5] considers the problem of characterizing the performance of the optimal estimator if only a subset among many sensors can transmit their measurements at every time step. For the case when this subset is chosen stochastically (either independently at every time step or according to a more complicated model like a Markov chain), the behavior of the expected error covariance of the estimator is studied. In particular, lower and upper bounds are provided and conditions for the expected error covariance to be bounded as time evolves are identified. In this work we analyze the LQG problem in the receding horizon case, under TCP-like communication. In [13] each packet contains only the next control input. If lost, a zero input is applied. We explore the case where at each instant we send the N future control inputs. If, at any time, a packet is lost, either the open loop input from the previous arrival is applied or a zero input is the loss sequence is greater than N. Intuitively this should yield better performance. Also, since packet length is usually fixed, and we can afford to send the extra input virtually at no additional communication cost.

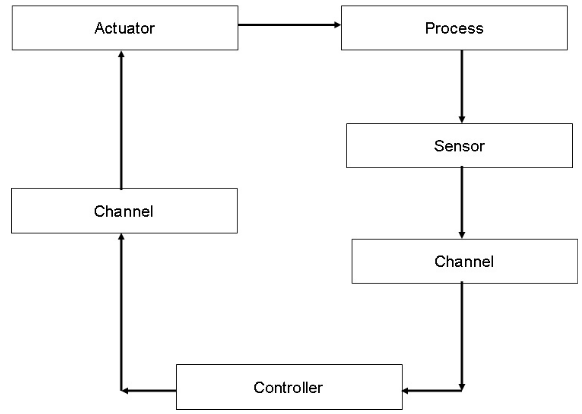

Fig. 1. System setup considered in the problem.

Section 2 contains a mathematical formulation of the problem. Section 3 provides analysis for the LQG regulator problem in the receding horizon case for the general output feedback case, providing stability conditions. In Section 4 we provide performance analysis in simulation for the state feedback case, showing how the cost decreases with the horizon's length. In the same section, we provide some preliminary results for the case of correlated losses, showing how the use of receding horizon techniques significantly improves overall system performance in this case. Finally Section 5 provides conclusions and directions for future work.

II. Problem Formulation

Consider the system setup shown in Figure II. 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,

[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.

Gaussian, zero mean and covariance matrix R_w. The state of the process is observed by a sensor with observations of the form

$$y_k = Cx_k + v_k,$$

where v_k is the measurement noise again assumed white, Gaussian, zero mean and covariance R_v. Further, the noises w_k and v_k are assumed independent of each other. We assume that the pair (A, B) is controllable and (A, C) is observable.

The measurements are transmitted over a sensor-controller channel to the controller. The channel is modeled using a packet erasure model. Thus the 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 not an issue. At every time step, the channel erases the data packet with a probability μ independently of all previous time steps. Thus at any time step, the controller receives a packet containing the measurement with a probability $1-\mu$.

At time step k, the controller calculates $N+1$ control inputs u_k^k, u_{k+1}^k, $\cdots$, u_{k+N}^k to minimize the finite-horizon cost function

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

The control inputs are transmitted over a controller-actuator channel to the actuator. This channel is again modeled using a packet erasure model. With a probability γ, independent of drops at all previous time steps and of other random variables in the system, the packet is lost at any time step. The actuator maintains a buffer of length $N+1$ control inputs. If it receives a packet, it erases all the information stored in the buffer previously and stores all the control inputs contained in the 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 picks the value 0.

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 control input it has picked to the process according to the equation (1). The following comments are in order now.

1) The packing of multiple control inputs assumes that the packet length is increased. In most communication protocols, the packet length is indeed more than required for transmitting a single control input. Nevertheless, more control inputs communicated translates to more power expended and we will view N as a design parameter.
2) The approach reminds one of the receding horizon based methods in which the controller calculates a trajectory of control inputs and updates it as new data comes in. The value N in some sense corresponds to the horizon over which the trajectory is calculated.
3) It has recently been demonstrated [6] that intelligent encoding and decoding strategies over the sensor-controller channel lead to great performance enhancement. The approach in this paper can be viewed as corresponding encoding and decoding over the controller-actuator channel.

A standing assumption in the paper will be that an acknowledgement 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. There are two questions we will tackle.

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

III. Analysis

A. Markov Chain Model

It is fairly easy to see that the situation at the actuator can be modeled as a Markov chain. When the controller transmits $N+1$ 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 chain. 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 $t \geq N+1$.

The Markov chain corresponds to the system evolving *backwards* in time. Thus the initial probability of being in the i-th state is the probability that at time step K, the last packet was received i time steps ago. Since the packet drops are independently and identically distributed, for the first $N+1$ states, the probability is $\gamma(1-\gamma)^{i-1}$ while for the $N+2$-th state it is $1 - \sum_{i=0}^{N} \gamma(1-\gamma)^{i-1}$. We will refer to this probability distribution set as $\{\pi_K^j\}$, where the superscript j stands for the state of the Markov chain while the subscript K refers to the fact that these probabilities correspond to the time step K of the system evolution. To calculate the probabilities at time step $K-1$, we need the transition probability matrix. While the probabilities are straight-forward to compute, they are hard to describe for the general case. We will, instead, illustrate the probabilities for the cases $N=0$ and 1.

For $N = 0$, there are two states in the chain: state 1 corresponds to packet not being lost at the current time step, while state 2 corresponds to packet being lost. If we denote q_{ij} to be the probability of being in state j at time step $k-1$ given that the system was in state i at time step k, then the transition probabilities are given by

$$\begin{aligned} q_{11} &= 1-\gamma & q_{12} &= \gamma \\ q_{21} &= 1-\gamma & q_{22} &= \gamma. \end{aligned}$$

For $N = 1$, the situation is similar. There are three states: state 1 corresponds to the last packet that arrived being transmitted at the same time step, state 2 to the last packet being transmitted one time step ago and state 3 to it being transmitted more than 1 time step ago. With a similar notation for q_{ij}'s, we can evaluate

$$\begin{array}{lll} q_{11} = 1-\gamma & q_{12} = \gamma(1-\gamma) & q_{13} = \gamma^2 \\ q_{21} = 1 & q_{22} = 0 & q_{23} = 0 \\ q_{31} = 0 & q_{32} = 1-\gamma & q_{33} = \gamma. \end{array}$$

Having defined the Markov chain model, we now proceed to the analysis of the cost function.

B. A Closer Look at the Cost Function

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

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

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

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

where we have defined

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

and represented the quantity P_{K+1} entering the i-th term in the summation as P_{K+1}^i. The state i fixes 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

$$T_k^i = E\Big[x_K^T Q x_K + u_K^T R u_K$$
$$+(Ax_K + Bu_K^{K-i+1} + w_K)^T P_{K+1}^i (Ax_K + Bu_K^{K-i+1} + w_K)\Big]$$

or as

$$T_k^i = E\Big[(u_K^{K-i+1} + \kappa^i)^T S^i (u_K^{K-i+1} + \kappa^i) + w_K^T P_{K+1}^i w_K$$
$$+ x_K^T (Q + A^T P_{K+1}^i A - P_{K+1}^i B (S^i)^{-1} B^T P_{K+1}^i) x_K\Big],$$

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

$$\begin{aligned} S^i &= R + B^T P_{K+1}^i B \\ \kappa^i &= (S^i)^{-1} B^T P_{K+1}^i A x_K \end{aligned}$$

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

$$E\left[(u_K^{K-i+1} + \kappa^i)^T S^i (u_K^{K-i+1} + \kappa^i)\right].$$

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 till 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 $K-i+1$ till time $K-1$. Thus the controller knows all the previous control inputs while estimating x_K. Thus 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

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

For ease of notation let us define an operation $f(.)$ as

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

Thus

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

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

$$T_K^{N+2} = E\left[x_K^T Q x_K + (Ax_K + w_K)^T P_{K+1}^i (Ax_K + w_K)\right]$$

or in turn

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

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

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

where we have defined

$$\begin{aligned} \Delta_K^{N+2} &= 0 \\ f^{N+2}(X) &= Q + A^T X A. \end{aligned}$$

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

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

The second summation involves only the noise terms and thus cannot be affected by the choice of the control inputs. The third 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 fourth summation and denote it by Γ_K. We have

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

where

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

Let us define

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

Thus

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

Finally the cost function can be written as

$$J_K = \sum_{k=0}^{K-1} E\left[x_k^T Q x_k + u_k^T R u_k\right] + \sum_{j=1}^{N+2} \pi_{K-1}^j E\left[x_K^T P_K^j x_K | m_{K-1}=j\right],$$

where we have ignored the second and the third summations that play no role in further minimization. But now we can again extract the term T_{K-1} in a form similar to the one in (2) and our argument from then on did 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:

$$\pi_{k-1}^i P_k^j = \sum_{i=1}^{N+2} \pi_k^i q_{ij} f\left(P_{k+1}^i\right), \tag{3}$$

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 $f(.)$ have been defined earlier. The initial values are

$$P_{K+1}^i = P_{K+1} \qquad \forall i.$$

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

Proposition 1 (Separation Principle): Consider the problem setting described in Section II. 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 till time step $k-1$ and the terms P_k^i evolve according to the coupled recursions (3).

C. 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, the system is stable. Looking at the analysis in Section III-B, 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 grows 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. This is because 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. In other words, the error terms Δ_k^i's are affine functions of the term Δ_k^1 and cannot grow unbounded as long as Δ_k^1 is finite.

Since by definition 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 Kalman filtering under intermittent observations that has been studied in the literature [15]. As an instance we can write down sufficient and necessary conditions for stability for the sensor-controller channel immediately.

Proposition 2: A necessary condition for the stability of the terms Δ_k^i is that the probability of drop μ and the spectral radius of matrix A should satisfy the relation

$$\mu \mid \rho(A) \mid^2 < 1.$$

A sufficient condition is that there exists a positive definite matrix X and a matrix K such that

$$X > (1-\mu)(AXA^* + R_w) + \mu\Big((A+KC)X(A+KC)^* + R_w + KR_vK^*\Big).$$

If, instead of measurements, the optimal coding is being done over the sensor-controller channel the necessary condition for stability in the proposition above turns sufficient as well.

For the terms P_k^j, we need to look at the recursions (3). The behavior of these equations has also been analyzed in previous work [5]. The following result holds.

Proposition 3: Consider the recursion defined in (3). Assume that the Markov chain transition probability matrix Q 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 (3) converges 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

$$\pi^j X^j = \sum_{i=1}^{N+2} f^j\left(X^i\right) q_{ij}\pi^i. \tag{4}$$

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

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

Proof: Proof is along the lines of the proof given for i.i.d. sensor choices in [5]. Redefine the quantities

$$\begin{aligned}\mathcal{L}(Y_j) &= \sum_{i=1}^{N} q_{ij}\pi_i(A+K_{ij}C_j)Y_j(A+K_{ij}C_j)^* \\ \phi_j(K_{ij}, Y_i) &= \Big((A+K_{ij}C_j)X_i(A+K_{ij}C_j)^* \\ &\quad + R_w + K_{ij}R_vK_{ij}^*\Big)q_{ij}\pi_i,\end{aligned}$$

and follow the arguments in that proof for the necessary condition. The sufficient condition follows from Theorem 5 of [5]. ■

Taken together, the results in Propositions 2 and 3 provide stability conditions for the entire system.

D. Performance Analysis

It is clear that an analysis similar to that for stability can be done for performance using the results of [15] and [5]. The terms Δ_k^i and P_k^i can be upper and lower bounded and consequently the cost J_K can be characterized. For the sake of illustration, we state the bounds for the infinite-horizon case in which we are once again interested in the cost J_∞.

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. The terms Δ^i can be bounded as follows. First note from Theorem 4 of [15] that Δ^1 satisfies the bounds

$$\bar{S}^1 \le \Delta^1 \le \bar{V}^1,$$

where

$$\begin{aligned}\bar{S}^1 &= (1-\mu)A\bar{S}^1A^* + R_w \\ \bar{V}^1 &= A\bar{A}^* + R_w - \\ &\quad \mu A\bar{V}^1C^*\left(C\bar{V}^1C^* + R_v\right)^{-1}C\bar{V}^1A^*.\end{aligned}$$

The terms Δ^i for $i = 2,\cdots,N+1$ can also be bounded using the above bounds and the relation

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

Let us call the corresponding bounds as $\bar{S}^i$ and $\bar{V}^i$ respectively. Finally the term Δ^{N+2} is zero by definition. Thus we have been able to lower and upper bound all the terms Δ^i.

For the terms P^i we first need to find the steady state probabilities of the Markov chain. It can be readily verified that the Markov chain is positive recurrent and irreducible and hence a unique stationary probability distribution does exist. Call the stationary probability of being in state i as π^i. Then the terms P^i are found from the coupled equations

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

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

The cost function J_∞ can thus be bounded as

$$\begin{aligned}\sum_{i=1}^{N+2}\pi^i \text{Trace}\left(P^iR_w\right) + \sum_{i=1}^{N+2}\pi^i\bar{S}^i &\le J_\infty \\ &\le \sum_{i=1}^{N+2}\pi^i\text{Trace}\left(P^iR_w\right) + \sum_{i=1}^{N+2}\pi^i\bar{V}^i.\end{aligned}$$

IV. Simulation Results

In this section we present numerical examples which demonstrate the benefit of sending multiple control inputs per packet. We consider a double integrator with state space system defined in equations 1 and II with the following parameters:

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad C = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

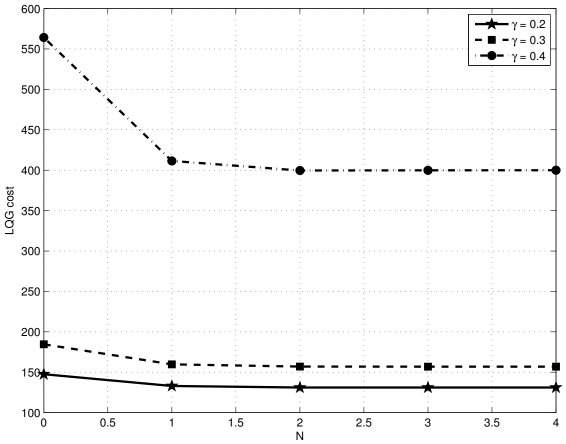

Fig. 2. LQG cost vs. N for different loss probability

The noise processes v_t and w_t are zero mean, with variance $R_v = 0.5$ and $R_w = I_{2x2}$ respectively. The controller is a LQG controller which minimized 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 $Q = C^T C$ and $R = 0.3$. The final state cost matrix $P_{K+1} = Q$.

Figure 2 plots the LQG cost vs. different number of control inputs per transmission for different probability of packets drop. 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 add more control inputs does not improve the cost. However as the probability of drop is increased to $\gamma = 0.4$ we observe that the flattening of the curve happens at $N = 2$, which is because of the fact that there is more chance of two consecutive losses. The above observation leads us to believe that this approach would be very useful in scenarios where the channel has correlated losses. This is frequently the case in wireless channel where the channel when enters a fade, stays in bad condition for sufficient duration of time during which lot of packet losses occurs. In the following subsection we simulate the algorithm for case of correlated losses, where the packet drops are not independent of each other. Note that the analysis done before is only valid for independent losses only.

A. Correlated Losses

In order to model the control packet losses, we consider a two state Markov model called the Gilbert-Elliott channel. The Gilbert-Elliott channel is a simple two state channel where the loss process is determined by the current state of the discrete time stationary binary Markov Process. Note that this Markov chain is different from the one desribed previously for independent losses. This Markov chain represents the channel state transition. The two states are defined as *good state* and *bad state* denoted by G and B respectively. It is assumed that no packets are lost in the 'Good state' while all packets are lost in the 'Bad state'. The error process depends on the underlying state of the Markov process. Following the notation of [9] 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

$$\mathbb{P}[s_l|\mathbf{s_{l-1}}] = \mathbb{P}[s_l|s_{l-1}] \tag{5}$$

where $\mathbf{s_{l-1}} = (s_{l-1}, s_{l-2}, \ldots, s_1)$. The transition probabilities for the states are given as

$$g = \mathbb{P}[s_l = G|s_{l-1} = B], \quad b = \mathbb{P}[s_l = B|s_{l-1} = G] \tag{6}$$

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

$$\mathbb{P}[s_0 = G] = \frac{g}{g+b}, \quad \mathbb{P}[s_0 = B] = \frac{b}{g+b} \tag{7}$$

As in [9] we define *channel memory* m as

$$m \triangleq 1 - g - b \tag{8}$$

When $m = 0$, the channel is memoryless and the losses are independent. When $m > 0$, the channel has persistent memory, that is the probability of remaining in any given state is higher that its steady state probability. When the channel is persistent then any value of m from zero to one is valid for any value of ρ.

Figure 3 shows the plot of LQG cost vs. different number of control inputs per transmission in correlated losses case. We fix the steady state probability of Markov chain in good state as $\frac{g}{g+b} = 0.8$ and vary the channel memory m. As can be seen from the graph, for higher value of channel memory, we get more benefit from increasing the number of control packets per transmission. This makes intuitive sense since with increasing channel memory, there is a burst of packet losses which are countered by increasing N.

V. Conclusions and Future Work

This paper presents a receding horizon-based scheme for the control of linear dynamical systems over lossy networks. A lossy network is assumed to exist between the sensor and the controller and between the latter and the actuator. Compared to previous work [13], here at each instant N future control inputs are sent in addition to the current one. If, at any time, a packet is lost, either the open loop input from the previous arrival is applied or a zero input if the loss sequence is greater than N.

For TCP-like networks, where packet acknowledgement is available at the sender, it is shown how the separation principle holds, allowing the designer to address the estimation and control independently. The paper also provides conditions for boundedness of the cost function, which depend on the loss probabilities of both measurement and control channels. Finally the simulation section shows how

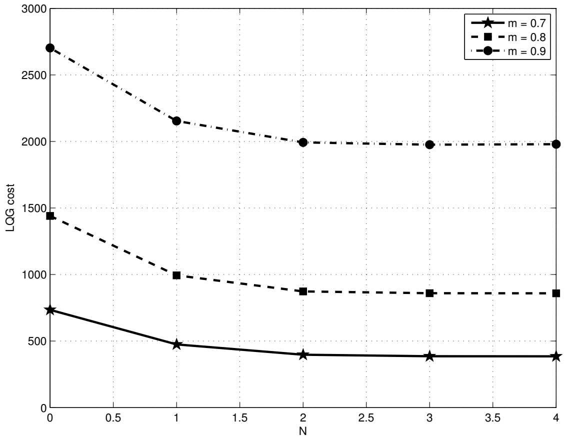

Fig. 3. LQG cost vs. N for correlated losses, m is the channel memory.

the final cost decreases with a larger horizon. This situation is more pronounced in the case of a Gilbert-Elliott Channel with memory. Simulation results show how the improvement takes place with small values of N. Using large values of N does not seem to provide further improvement. The systems designer can use this result to decide how long the horizon should be, trading off improvement in the overall performance and control packet length.

There are several directions for future work. The conditions for stability are provided in terms of implicit equations. While the critical measurement packet loss probability is the same as in [15], its control channel counterpart needs to be derived. Although we expect the length of the horizon N not to affect the critical loss probability at the control side, this should be proved rigorously. Also the relationship between the cost function and the length of the horizon needs to be derived.

References

[1] O.L.V. Costa. Stationary filter for linear minimum mean square error estimator of discrete-time markovian jump systems. *Transactions on Automatic Control*, 48(7):1351–1356, 2002.

[2] J.B.R. do Val, J.C. Geromel, and O.L.V. Costa. Solutions for the linear quadratic control problem of markov jump linear systems. *J. Optimization Theory and Applications*, 103(2):283–311, 1999.

[3] N. Elia. Remote stabilization over fading channels. *Systems and Control Letters*, 54:237–249, 2005.

[4] N. Elia and J.N. Eisembeis. Limitation of linear control over packet drop networks. In *Proceedings of IEEE Conference on Decision and Control*, volume 5, pages 5152–5157, Bahamas, December 2004.

[5] 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*, 42(2):251–260, February 2006.

[6] V. Gupta, D. Spanos, B. Hassibi, and R. M. Murray. On LQG control across packet-dropping links. *System and Control Letters*, Submitted July 2005.

[7] C. N. Hadjicostis and R. Touri. Feedback control utilizing packet dropping network links. In *Proceedings of the IEEE Conference on Decision and Control*, volume 2, pages 1205–1210, Las Vegas, NV, Dec. 2002.

[8] Q. Ling and M.D. Lemmon. Optimal dropout compensation in networked control systems. In *IEEE conference on decision and control*, Maui, HI, December 2003.

[9] M. Mushkin and I. Bar-David. Capacity and coding for gilbert elliott channels. *IEEE Transactions on Information Theory*, 35(6):1277–1290, November 1989.

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

[11] Johan Nilsson. *Real-Time Control Systems with Delays*. PhD thesis, Department of Automatic Control, Lund Institute of Technology, 1998.

[12] A. V. Savkin and I. R. Petersen. Robust filtering with missing data and a deterministic description of noise and uncertainty. *Int. J. Syst. Sci.*, 28(4):373–390, 1997.

[13] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry. Foundations of control and estimation over lossy networks. *Proceedings of the IEEE*, to be published in late 2006.

[14] P. Seiler and R. Sengupta. An H_∞ approach to networked control. *IEEE Transaction on Automatic Control*, 50(3):356–364, March 2005.

[15] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M.I. Jordan, and S. Sastry. Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49(9):1453–1464, September 2004.

[16] S.C. Smith and P. Seiler. Estimation with lossy measuraments: jump estimators for jump systems. *IEEE Transaction on Automatic Control*, 48(12):1453–1464, 2003.

[17] Z. Wang, D.W.C. Ho, and X. Liu. Variance-constrained control for uncertain stochastic systems with missing measurements. *IEEE Transactions on systems, man, and cybernetics Part A: systems and humans*, 35(5):746–753, September 2005.

[18] Y. Xu and J.P. Hespanha. Estimation under controlled and uncontrolled communications in networked control systems. In *To be presented at the IEEE Conference on Decision and Control*, Sevilla, Spain, December 2005.

[19] N. Yu, L. Wang, G. Xie, and T. Chu. Stabilization of networked control systems with data packet dropout via swithed system approach. In *IEEE Int. Symp. on Computer Aided control systems design*, pages 362–367, Taipei, Taiwan, September 2004.