Submitted, IEEE T. Automatic Control
http://www.cds.caltech.edu/~murray/papers/2004o_gshm04-tac.html

Optimal LQG Control Across a Packet-Dropping Link

Vijay Gupta, Demetri Spanos, Babak Hassibi, Richard M Murray
Division of Engineering and Applied Science
California Institute of Technology
{vijay,demetri}@cds.caltech.edu,{hassibi,murray}@caltech.edu

August 12, 2004

Abstract

We examine optimal Linear Quadratic Gaussian control for a system in which communication between the sensor (output of the plant) and the controller occurs across a packet-dropping link. We extend the familiar LQG separation principle to this problem that allows us to solve this problem using a standard LQR state-feedback design, along with an optimal algorithm for propagating and using the information across the unreliable link. We present one such optimal algorithm, which consists of a Kalman filter at the sensor side of the link, and a switched linear filter at the controller side. Our design does not assume any statistical model of the packet drop events, and is thus optimal for an arbitrary packet drop pattern. Further, the solution is appealing from a practical point of view because it can be implemented as a small modification of an existing LQG control design.

1 Introduction

Recently, much attention has been directed toward systems which are controlled over a communication link (see, for example, [1] and the references therein). In such systems, the control performance can be severely affected by the properties of the network or the channel. In extreme cases, poor network performance can even destabilize a nominally stable control loop. Communication links introduce many potentially detrimental phenomena, such as quantization error, random delays, data corruption and packet drops to name a few. Understanding and counter-acting these effects will become increasingly important as emerging applications of decentralized control mature. These applications will require the exchange of critical pieces of information over unreliable communication media.

The above issues have motivated much of the study of networked systems. Beginning with the seminal paper of Delchamps [5], quantization effects have been studied by Tatikonda [28], Elia and Mitter [6], Brockett and Liberzon [4], Hespanha et al. [12], Ishii and Francis [14], Nair and Evans [22, 23], and many others. The effects of delayed packet delivery have also been considered in many works, such as Nilsson [24], Blair and Sworder [3], Luck and Ray [21], Gupta et al. [8], Tsai and Ray [29], and Zhang et al. [32], using various models for the network delay.

In this work, we are specifically interested in systems communicating over links which randomly drop packets. The nominal system is shown in Figure 1 where the link is modeled as one that randomly drops packets being communicated from the plant to the controller. Preliminary work in this area studied stability of systems utilizing lossy packet-based communication, as in [10, 25, 32]. Performance of such systems as a function of packet loss rate was analyzed by Seiler in [25] and Ling and Lemmon in [17] assuming certain statistical dropout models. Nilsson [24] proposed two approaches for compensation for data loss in the link by the controller, namely keeping the old control or generating a new control by estimating the lost data, and presented an analysis of the stability and performance of these approaches. Analysis of the performance when lost data is replaced by zeros is given by Hadjicostis and Touri in [9]. Ling and Lemmon, in a series of papers [17, 18, 19] proposed some compensators for specific statistical data loss models. Optimal compensator design for the case when data loss is i.i.d. was considered in [18], in which the problem was posed as a nonlinear optimization. An alternative approach was taken by Azimi-Sadjadi in [2]. In this work, a quadratic cost was

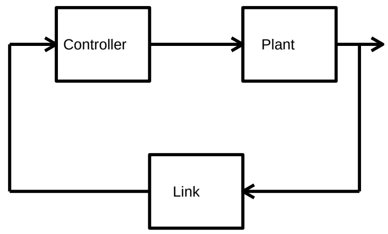

Figure 1: The architecture of a packet-based control loop. The feedback loop is closed across an unreliable link, which unpredictably drops packets.

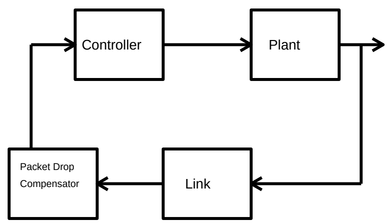

Figure 2: A common design for control over packet-based links. The compensation block attempts to mitigate the effects of packet losses.

sought to be minimized and a sub-optimal estimator and regulator were proposed. Sinopoli et al. [27] and Imer et al. [13] extended this approach further and obtained optimal controllers by assuming that the packet drops were i.i.d. The related problem of optimal estimation across a packet-dropping link was considered by Sinopoli et. al in [26] and extended by Liu and Goldsmith in [20] where analysis of the stability of estimation error covariance was also carried out.

Most of the designs proposed in the above references focus on design of a packet-loss compensator, as shown in Figure 2. The compensator accepts those packets that the channel successfully transmitted and comes up with an estimate for the time steps when data was lost by the link. This estimate is then used by the controller. Our work takes a more general approach by seeking the LQG optimal control for this packet-based problem. In particular, our architecture is as shown in Figure 3. We aim to jointly design the controller, the encoder and the decoder to solve the optimal LQG problem.

The remainder of this paper is organized as follows. In the next section, we present our mathematical model and pose the LQG problem in a packet-based setting. We then discuss a separation between control and estimation costs, and present an optimal solution to the estimation problem. We analyze the stability of our system and compare its performance with some other approaches in the literature. We finish by pointing out some directions for future research.

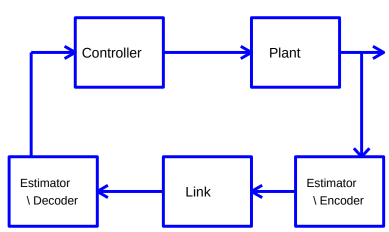

Figure 3: The structure of our optimal LQG control solution.

2 Problem Formulation

Consider a discrete-time linear system evolving according to the equation

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

where $x_k \in \mathbf{R}^n$ is the process state, $u_k \in \mathbf{R}^m$ is the control input and w_k is random noise entering the system. The noise process is assumed white, Gaussian, and zero mean with covariance matrix Q_w. [1] The initial condition x_0 is assumed to be independent of w_k and to have mean zero and covariance matrix Q_0. The state of the plant is measured by a sensor according to the equation

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

Here v_k is the measurement noise, again assumed white, zero-mean, Gaussian (with covariance matrix Q_v) and independent of the plant noise w_k. The sensor needs to communicate these measurements (or some function of the measurements) to the controller. The only constraint we impose on this communication is that the function communicated should be a finite vector, whose size does not increase as time goes on.

The communication is done over a channel that randomly drops packets. For the moment we will ignore delays and reordering of packet delivery; it will be shown that these effects can be accounted for with time-stamping and a slight modification to our design. Thus, at each time-step k:

- A packet is created at the sensor side of the link. We do not specify in advance what data these packets will contain.
- The packet is sent across the link.
- The packet is either received instantaneously, or dropped according to some probability distribution.

The packet dropping as discussed so far is a random process, but we will find it more useful to refer to individual (i.e. deterministic) realizations of this random process, which we call *packet drop sequences*. A packet drop sequence is a binary sequence,

$$\Lambda = \{\lambda_k\}_{k=0}^{\infty}$$

in which λ_k takes the value "*received*" if the link delivers the packet at time step k, and "*dropped*" if the packet is dropped.

We assume sufficient bits per packet and a high enough data rate so that quantization error is negligible. We also assume that enough error-correction coding is done within the packets so that the packets are either dropped or received without error. Finally, we assume no coding is done *across* packets; that is, no packet contains information about any other packet. We impose this constraint because coding across packets can induce a large encoding and decoding delay, and this is undesirable for control applications.

The packetized communication link described above warrants some discussion regarding the class of controllers we will allow. Clearly, the absolutely optimal LQG performance achievable is given by the classical LQR controller/Kalman estimator pair. However, this design does not respect the packetized nature of the communication. Specifically, the LQR controller requires continual access to the Kalman filter output, which in turn requires continual access to the sensor measurements. The sensor measurements may not be continuously available because of data loss in the communication link. In order to make the class of controllers that are allowed more precise, we introduce the following terminology.

Consider the control calculation at time-step k. Denote by s_k the finite vector that is transmitted from the sensor to the controller. By causality, s_k can depend (possible in a time-varying manner) on y_0, y_1, $\cdots$, y_k, i.e.,

$$s_k = f_k\left(y_0, y_1, \cdots, y_k\right).$$

Call the *information set* available to the controller as

$$I_k = \{s_k | \forall k \text{ s.t. } \lambda_k = \text{ received}\}.$$

[1] The results of the paper continue to hold for time-varying systems, but we concentrate on the time-invariant case to simplify the presentation.

Also denote by $t_l(k) \leq k$ the last time-step at which a packet was delivered. That is,

$$t_l(k) = \max\{i \leq k \mid \lambda_i = \text{"}received\text{"}\}.$$

The *maximal information set* at time-step k is defined as follows:

$$I_k^{max} = \{y_i \mid 0 \leq i \leq t_l(k)\}.$$

The maximal information set is the largest set of output measurements on which the control at time-step k can depend. In general, the set of output measurements on which the control can depend will be less than this set, since earlier packets, and hence measurements, may have been dropped. The information contained in I_k^{max} thus upper bounds the information contained in I_k. We do not yet specify how the encoder designs or the controller uses I_k. As stated earlier, the only restriction we impose is that the vector s_k not increase in size as k increases. We will call the set of f_k's which fulfil this requirement as **F**. [2] Without loss of generality, we will only consider information-set feedback controllers, i.e. controllers of the form

$$u_k = u(I_k, k). \tag{3}$$

Thus, we allow the control to depend on the information set, and on the current time-index. Clearly, this is the broadest class of controllers one can sensibly consider for this problem since we have not assumed anything about the functional form of the control. Moreover, it is impossible for a physical realization of the controller to have more feedback information than is contained in the information set. In addition, perfect knowledge of the system parameters A, B, C, Q_w and Q_v is assumed at the controller. Moreover we assume that the controller has access to the previous control signals $u_0, u_1, \cdots, u_{k-1}$ while calculating the control u_k at time k. For notational convenience, we denote the set of control laws of the form described by equation 3 by U.

We can now pose the packetized LQG problem as follows:

$$\min_{u \in U, f \in \mathbf{F}} J_K(u, f, P) = E\left[\sum_{k=0}^{K} \left[x_k^T R^c x_k + u_k^T Q^c u_k\right] + x_{K+1}^T P_{K+1}^c x_{K+1}\right].$$

Here K is the horizon on which the plant is operated and the expectation is taken over the uncorrelated variables x_0 and the noise variables w_i and v_i. The expectation is taken over the primitive system random variables x_0, w_0, w_1, $\cdots$, w_K, v_1, v_2, $\cdots$, v_K. Note that the cost functional J above depends on the packet-drop sequence P. However, we do *not* average across packet-drop processes; *the solution we will present is optimal for an arbitrary realization of the packet dropping process.* That is, for any given packet-drop sequence P, the controller, encoder and decoder we propose will minimize $J(u, f, P)$ over the set of allowable controllers U and allowable functions **F**. Because of this, we will occasionally suppress the packet-drop dependence in the cost functional, and merely write $J(u, f)$ or just J.

Note that the usual addition and scalar multiplication operations make U a vector space. For a given f and P, $J(u, f, P)$ is a convex cost functional, and so existence of optima follows immediately. We make no claim regarding uniqueness of optimal control in this problem.

Our goal, then, is to solve the standard LQG problem with the additional complication of the packet-dropping link. While this may appear a small modification, it is unclear *a priori*, what the structure of the optimal control algorithm should be, and in what way the packetized link should be used through the design of the encoder and the decoder. We will show that one optimal algorithm is to utilize an LQR state-feedback design at the controller side, and to use the link to send the state estimates from a Kalman filter at the sensor side.

3 Separation of Control and Estimation

In this section we will briefly revisit the LQG separation principle in the packet-based setting. This will motivate the structure of our optimal controller/encoder design.

[2]The information set is reminiscent of the 'information pattern' introduced by Witsenhausen [30]. We assume that the controller at time step k has access to all the previous controls u_0, u_1, $\cdots$, u_{k-1}. Thus the 'information pattern' of the controller consists of the set I_k and all the previous controls.

Consider again the K-horizon cost functional:

$$J_K(u,f,P) = E\left[\sum_{k=0}^{K} u_k^T Q^c u_k + \sum_{k=0}^{K} x_k^T R^c x_k\right] + E\left[x_{K+1}^T P_{K+1}^c x_{K+1}\right]. \tag{4}$$

We need to choose u_0, u_1, $\cdots$, u_K that minimize $J_K(u,f,P)$. Following [11], we gather terms that depend on the choice of u_K and x_K and write them as

$$\begin{aligned} T_K &= E\left[u_K^T Q^c u_K + x_K^T R^c x_K\right] + E\left[x_{K+1}^T P_{K+1}^c x_{K+1}\right] \\ &= E\left[\left[\begin{array}{cc} u_K^T & x_K^T \end{array}\right] \Delta \left[\begin{array}{c} u_K \\ x_K \end{array}\right]\right] + E\left[w_K^T P_{K+1}^c w_K\right] \\ &= S_K + O_K \end{aligned}$$

where

$$\begin{aligned} \Delta &= \left[\begin{array}{cc} Q^c + B^T P_{K+1}^c B & B^T P_{K+1}^c A \\ A^T P_{K+1}^c B & R^c + A^T P_{K+1}^c A \end{array}\right] \\ S_K &= E\left[\left[\begin{array}{cc} u_K^T & x_K^T \end{array}\right] \Delta \left[\begin{array}{c} u_K \\ x_K \end{array}\right]\right] \\ O_K &= E\left[w_K^T P_{K+1}^c w_K\right]. \end{aligned}$$

In the above equation, we have used the system dynamics,

$$x_{k+1} = Ax_k + Bu_k + w_k$$

and the fact that the plant noise is zero mean. Thus we can write

$$J_K(u,f,P) = E\left[\sum_{k=0}^{K-1} u_k^T Q^c u_k + \sum_{k=0}^{K-1} x_k^T R^c x_k\right] + S_K + O_K. \tag{5}$$

We aim to choose u_K to minimize $J_K(u,f,P)$ for a given f. From equation (5), it is clear that the only term where the choice of u_K can make a difference is S_K. Now consider

$$S_K = E\left[\left[\begin{array}{cc} u_K^T & x_K^T \end{array}\right] \Delta \left[\begin{array}{c} u_K \\ x_K \end{array}\right]\right].$$

Completing squares, we obtain

$$S_K = E\left[\left(u_K - \bar{u}_K\right)^T R_{e,K}^c \left(u_K - \bar{u}_K\right)\right] + E\left[x_K^T P_K^c x_K\right]$$

where

$$R_{e,K}^c = Q^c + B^T P_{K+1}^c B$$

and $\bar{u}_K$ is the standard optimal LQG control,

$$\bar{u}_K = -\left(R_{e,K}^c\right)^{-1} B^T P_{K+1}^c A x_K.$$

Furthermore, the matrix P_K^c is given by

$$P_K^c = R^c + A^T P_{K+1}^c A - A^T P_{K+1}^c B \left(Q^c + B^T P_{K+1}^c B\right)^{-1} B^T P_{K+1}^c A.$$

In the absence of the packetized link, the controller could simply use the standard optimal control $\bar{u}_K$. However, as discussed before, this control law does not lie in the set of allowable solutions U because it is not physically realizable for any non-trivial packet-dropping sequence. Instead, we will calculate u_K based only on the information set I_K (and the previous controls u_0, u_1, $\cdots$, u_{K-1} that are assumed known to the controller) and choose it so as to minimize S_K. The control problem thus reduces to an optimal estimation problem. Note that since all the random variables are Gaussian, and the cost function to be optimized is quadratic, the optimal estimator is linear.

Given the information set at time k, I_k, we denote the linear least mean square (llms) estimate of a random variable Γ based on this information as $\hat{\Gamma}_{|I_k}$ [3] Then we can write the optimal control at time step K as

$$\begin{aligned} u_K &= \hat{\bar{u}}_{K|I_K} \\ &= -\left(R^c_{e,K}\right)^{-1} B^T P^c_{K+1} A \hat{x}_{K|I_K}. \end{aligned} \tag{6}$$

Here we have used the fact that the llms estimate of a linear function of a random variable Γ, say $L\Gamma$, is given by $L\hat{\Gamma}$, where $\hat{\Gamma}$ is the llms estimate of Γ. Thus, we only need to find the llms estimate of x_K, given the information I_K available to the controller. Note that since the information content in I_k is upper bounded by the information contained in I_k^{max}, the error in $\hat{x}_{K|I_K}$ is lower bounded by the error in calculating $\hat{x}_{K|I_K^{max}}$. In the next section, we will provide a way to design the functions f_k's that will surprisingly allow the errors to actually coincide.

Denote the estimation error thus incurred by Λ_K. So we obtain

$$S_K = \Lambda_K + E\left[x_K^T P_K^c x_K\right].$$

Note that Λ_K is independent of the previous control inputs $u_0, \cdots, u_{K-1}$ since these are assumed known to the controller when it calculates u_K in equation (6). For the minimizing choice of u_K, we can write the cost function as

$$\begin{aligned} J_K(u,f,P) &= E\left[\sum_{k=0}^{K-1} u_k^T Q^c u_k + \sum_{k=0}^{K-1} x_k^T R^c x_k\right] + S_K + O_K \\ &= E\left[\sum_{k=0}^{K-1} u_k^T Q^c u_k + \sum_{k=0}^{K-1} x_k^T R^c x_k\right] + \Lambda_K + E\left[x_K^T P_K^c x_K\right] + O_K \\ &= J_{K-1}(u,f,P) + \Lambda_K + O_K. \end{aligned}$$

Thus we now need to choose control inputs for time steps 0 to $K-1$ to minimize J_{K-1}, independently of the associated estimation cost at time step K (the terms O_K and Λ_K do not involve these control inputs). However, our argument so far was independent of the time index K. Thus we can recursively apply this argument for time steps $K-1$, $K-2$ and so on. We have thus obtained the familiar separation result, in the packet-based setting:

Proposition 1 (Separation). *Consider the packet-based optimal control problem,*

$$\min_{u\in U, f\in \mathbf{F}} \quad J_K(u,f,P) = E\left[\sum_{k=0}^{K}\left[x_k^T R^c x_k + u_k^T Q^c u_k\right] + x_{K+1}^T P_{K+1}^c x_{K+1}\right].$$

Then, for an optimizing choice of the control, the control and estimation costs decouple. Specifically, the optimal control input at time k is calculated by using the relation

$$u_k = \hat{\bar{u}}_{k|I_k} = -\left(R^c_{e,k}\right)^{-1} B^T P^c_{k+1} A \hat{x}_{k|I_k},$$

where $\bar{u}_k$ is the optimal LQ control law while $\hat{\bar{u}}_{k|I_k}$ and $\hat{x}_{k|I_k}$ are the llms estimate of $\bar{u}_k$ and x_k respectively, given the information set I_k and the previous control laws $u_0, u_1, \cdots, u_{k-1}$.

[3] This notation is a bit misleading in that it suppresses the fact that the previous controls are known to the controller and are also used for the purpose of estimation. However we adopt it for simplicity and because it makes explicit the notion that the information quantity we aim to optimize over is contained in the set I_K.

This result must be viewed in light of the limited information available to the controller. At every time step, the controller tries to estimate the optimal control input based on the information set I_k, and uses this estimate in the optimal LQR control law. Thus, the state-feedback portion of an LQG controller need not be reworked for a packet-based implementation.

The packet-based LQG question thus reduces to choosing what information should be sent from the sensor so that the optimal estimate can be formed at the controller, given that some of the packets might be lost. We address this issue in the next section.

4 Optimal Encoder and Decoder Design

In this section we present an algorithm for encoding and transmitting sensor measurements so as to achieve optimal estimation performance. Recall that we wish to construct the optimal estimate based on the information set I_k^{max}, but we have not yet specified how to design f_k's that will allow the controller to compute that. For a link which does not drop packets, it is clear that sending the current measurement y_k in the current packet can achieve optimal performance. The controller need only run a Kalman filter on the output measurements it receives. However, it is not clear that just sending the measurements can achieve optimality when packets are dropped. In particular, the Kalman filter input will be interrupted by the packet dropping. A naíve solution would be to send the entire history of the output variables at each time step. However, as mentioned earlier, it can be easily seen that this is impractical as it requires increasing data transmission as time increases.

Surprisingly, we can achieve performance equivalent to the naíve solution using a constant amount of transmission, and a constant amount of memory at the receiver end. To do so, we will use the fact that the output of a Kalman filter is the optimal estimate of x_k in the llms sense, *given all of the previous measurements*, in order to encode or "compress" the entire history into a single estimate to be transmitted at each time step.

The solution we propose has the structure of an encoder/decoder pair, with the encoder producing messages to send across the link, and the decoder using these messages to construct an optimal estimate. This procedure is described below.

Optimal Transmission and Estimation Algorithm

- The encoder receives as input the measurement y_i.
- It runs a Kalman filter that provides the llms estimate of x_k based on all the measurements until time step k, denoted by $\hat{x}_{k|k}$. It transmits this vector across the link.
- The decoder maintains a local variable $\hat{x}_k^{dec}$. It is updated as follows:
  - If $\lambda_k =$ *received*, the decoder receives $\hat{x}_{k|k}$, and sets $\hat{x}_k^{dec} = \hat{x}_{k|k}$.
  - If $\lambda_k =$ *dropped*, then the decoder implements the following linear predictor:

$$\hat{x}_k^{dec} = A\hat{x}_{k-1}^{dec} + Bu_{k-1}. \quad (7)$$

Proposition 2 (Optimal Estimation). *In the procedure described above,* $\hat{x}_k^{dec} = \hat{x}_{|I_k^{max}}$.

Proof. There are two cases. First, suppose $\lambda_k =$ *received*. Then, $\hat{x}_k^{dec}$ is precisely the Kalman filter output, and thus is the optimal least-squares estimate of the state x_k.

Now, suppose $\lambda_k =$ *dropped*. Recall that $t_l(k)$ is the last time at which a packet was received. Now, note that the predictor equation (7) is the optimal least-squares predictor for x_k given the llms estimate of x_{k-1}, since the plant noise is Gaussian and has mean zero. The decoder estimate $\hat{x}_k^{dec}$ is thus precisely the output of the optimal predictor, initialized at $\hat{x}_{t_l(k)}^{dec}$, and run for $k - t_l(k)$ steps. However, from the previous argument, $\hat{x}_{t_l(k)}^{dec}$ is the optimal least-squares estimate of x at time $t_l(k)$. The output of the optimal predictor, initialized at the optimal estimate, is the optimal estimate of x_k, given all the measurements until $t_l(k)$. Thus, $\hat{x}_k^{dec} = \hat{x}_{|I_k^{max}}$. □

Note that throughout this discussion, we have made no assumption about the packet dropping behavior. The encoding and estimation algorithm described above provides the optimal estimate based on I_k^{max} for an arbitrary packet drop sequence, irrespective of whether the packet drop can be modeled as an i.i.d. process (or a more sophisticated model like a Markov chain) or whether its statistics are known to the plant and the controller. This, combined with our previous discussion regarding separation of control and estimation, allows us to state our main result.

Proposition 3 (Optimal Packet-Based LQG Control). *For the packet-based optimal control problem stated in section 2, the combination of an LQR state feedback design with the optimal transmission-estimation algorithm described above achieves the minimum of $J(u, f, P)$ for all P.*

Thus we have solved the packet-based LQG control problem using the separation theorem and the results presented in this section. Also note that the solution can be extended to the case when the channel applies a random delay to the packet so that packets might arrive at the decoder delayed or even out-of-order, as long as we assume that there is a provision for time-stamping the packets sent by the encoder. At each time step, the decoder will face one of four possibilities, and will update its estimate as described below:

Optimal Asynchronous Estimation Algorithm Followed at the Decoder

- It receives $\hat{x}_{k|k}$. It uses this as its estimate.
- It does not receive anything. It uses the predictor equation (7) on $\hat{x}_{k-1}^{dec}$ as before.
- It receives $\hat{x}_{m|m}$ while at a previous time step, it has already received $\hat{x}_{n|n}$, where $n > m$. It discards $\hat{x}_{m|m}$ and uses the predictor equation (7) on $\hat{x}_{k-1}^{dec}$.
- It receives $\hat{x}_{m|m}$ and at no previous time step has it received $\hat{x}_{n|n}$, where $n > m$. It uses $\hat{x}_{m|m}$ as $\hat{x}_m^{dec}$ and uses the predictor equation until time step k to obtain $\hat{x}_k^{dec}$.

5 Analysis of the Proposed Algorithm

Note that our analysis so far has made no assumption about the packet drop model. In this section, we make some assumptions about the packet dropping random process and provide some stronger results on the stability and performance of our algorithm. We will model the channel erasures as occurring according to a Markov chain, which includes the case of independent packet drops as a special case. The model of Markov jump system is a popular way to deal with communication channels and networks (see, e.g., [31]). Thus the channel will be assumed to exist in two states, state 1 corresponding to a packet drop and state 2 corresponding to no packet drop. At any time, the channel is in one of these states and it transitions probabilistically between these states according to the transition probability matrix Q which is of the form

$$Q = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}.$$

The (i, j)-th element of Q represents the probability of the state changing from i to j at the next time step. We also assume that in the Kalman filter used by the encoder, there is strict causality in the sense that to calculate the estimate of x_k, only the measurements till time step $k-1$ are used. The arguments given below can be easily extended to the case when strict causality is replaced by causality (the estimate of x_k depends on $y_0, y_1, \cdots, y_k$). Finally we assume that the system is stabilizable and the pair (A, C) is observable, so that the estimation error for the Kalman filter at the encoder is stable. We will use the following mathematical notation. The operation $A \otimes B$ will denote the Kronecker product of matrices A and B, while $\text{vec}(A)$ will represent the vectorizing operation that results in a vector formed by stacking the columns of A (see [16] for more details).

5.1 Stability Analysis

The plant evolves as

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

the Kalman filter at the encoder according to

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + K_k\left(y_k - Cx_k\right)$$

and the estimator at the decoder according to the relation

$$\hat{x}^{dec}_{k+1} = \begin{cases} A\hat{x}^{dec}_k + Bu_k & \text{channel in state 1} \\ \hat{x}_{k+1} & \text{channel in state 2.} \end{cases}$$

Denote $e_k = x_k - \hat{x}_k$ and $t_k = \hat{x}_k - \hat{x}^{dec}_k$. Also note that the control input is given by

$$u_k = F_k\hat{x}^{dec}_k.$$

Thus the plant dynamics equation can be rewritten as

$$x_{k+1} = \left(A + BF_k\right)x_k + w_k - BF_k\left(t_k + e_k\right).$$

If (A, B) is stabilizable, by construction F_k is the optimum control law. Thus in particular, it stabilizes the system as long as the disturbances w_k, t_k and e_k remain bounded. We only need to ensure that w_k, t_k and e_k remain bounded for x_k to be stable. We assume the noise w_k has bounded covariance matrix. Also e_k has bounded covariance matrices by assumption of observability of (A, C). Finally for t_k, we see that it evolves according to the equation

$$t_{k+1} = \begin{cases} At_k + Kv_k - KCe_k & \text{channel in state 1} \\ 0 & \text{channel in state 2.} \end{cases} \tag{8}$$

Again note that v_k and e_k have bounded covariance. For t_k to be of bounded variance, the Markov jump system of equation (8) needs to be stable. Further note that since our controller and encoder/decoder design is optimal, if the closed loop is unstable with our design, it is not stabilizable by any other design. Following [24], we can write the stability condition for t_k as follows.

Proposition 4 (Stability Condition). *Consider the system given in equation (1) being observed through a sensor of the form in equation (2) in which the sensor information is encoded and transmitted to the controller over a packet erasure channel which transitions between the 'drop packet' and 'transmit packet' states according to a Markov chain with transition probability matrix Q. The system is stabilizable, in the sense that the variance of the state is bounded, if and only if*

1. *the matrix pair (A, B) is stabilizable.*
2. *the matrix pair (A, C) is detectable.*
3. *the matrix*

$$\left(Q^T \otimes I\right)\begin{bmatrix} 0 & 0 \\ 0 & A \otimes A \end{bmatrix}$$

has eigenvalues strictly less than unity in magnitude, where I is identity matrix and 0 is the zero matrix of suitable dimensions.

Further, if the system is stabilizable, one controller and encoder/decoder design that stabilizes the system is given in Proposition 3.

As a simple example, suppose the channel has two states between which it jumps independently. With a probability p at each time step, the channel drops the packet. Also assume that the plant is scalar with the system matrix given by a. Then the above condition reduces to the condition $pa^2 < 1$.

5.2 Performance Analysis

In this subsection, we calculate the total quadratic cost incurred by the system for the infinite-horizon case (the case when $K \to \infty$ in equation (4)). We will make the additional assumption that the Markov chain is stationary and regular (see [7]) and that the probability of channel being in state i at time when the Markov chain reaches the stationary distribution is given by $\pi(i)$. For the infinite horizon case, the cost has to be slightly modified to prevent divergence. We consider the cost

$$J_\infty = \lim_{K\to\infty} E\left[\frac{1}{K}\sum_{k=0}^{K}\left[x_k^T R^c x_k + u_k^T Q^c u_k\right]\right].$$

Assuming ergodicity, this reduces to

$$\begin{aligned} J_\infty &= \lim_{K\to\infty} E\left[x_K^T R^c x_K + u_K^T Q^c u_K\right] \\ &= \left[\operatorname{trace}\left(P_x^\infty R^c\right) + \operatorname{trace}\left(P_u^\infty Q^c\right)\right], \end{aligned} \tag{9}$$

where $P_x^\infty = \lim_{K\to\infty} E\left[x_K x_K^T\right]$ and $P_u^\infty = \lim_{K\to\infty} E\left[u_K u_K^T\right]$.

With the assumptions of stability and observability stated above the control law matrix F_k and the Kalman gain matrix K_k can be considered as constant matrices F and K respectively. From the discussion given in section 5.1, we can write the evolution of the system in the following manner. Denote

$$z_k = \begin{bmatrix} x_k \\ e_k \\ t_k \end{bmatrix}.$$

Then,

$$z_{k+1} = \begin{cases} \begin{bmatrix} A+BF & -BF & -BF \\ A-KC & 0 & 0 \\ 0 & -KC & A \end{bmatrix} z_k + \begin{bmatrix} I & 0 \\ I & -K \\ 0 & -K \end{bmatrix} \begin{bmatrix} w_k \\ v_k \end{bmatrix} & \text{channel in state 1} \\[2ex] \begin{bmatrix} A+BF & -BF & -BF \\ A-KC & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} z_k + \begin{bmatrix} I & 0 \\ I & -K \\ 0 & 0 \end{bmatrix} \begin{bmatrix} w_k \\ v_k \end{bmatrix} & \text{channel in state 2,} \end{cases}$$

where 0 denotes the zero matrix and I the identity matrix of suitable dimensions. Following [24], we introduce the following notation. Define the stationary covariance P^∞ as

$$P^\infty = \lim_{k\to\infty} E\left[z_k z_k^T\right].$$

Also denote

$$
\begin{aligned}
A_1 &= \begin{bmatrix} A+BF & -BF & -BF \\ A-KC & 0 & 0 \\ 0 & -KC & A \end{bmatrix} \otimes \begin{bmatrix} A+BF & -BF & -BF \\ A-KC & 0 & 0 \\ 0 & -KC & A \end{bmatrix} \\
A_2 &= \begin{bmatrix} A+BF & -BF & -BF \\ A-KC & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} A+BF & -BF & -BF \\ A-KC & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
G_1 &= \begin{bmatrix} I & 0 \\ I & -K \\ 0 & -K \end{bmatrix} R \begin{bmatrix} I & 0 \\ I & -K \\ 0 & -K \end{bmatrix}^T \\
G_2 &= \begin{bmatrix} I & 0 \\ I & -K \\ 0 & 0 \end{bmatrix} R \begin{bmatrix} I & 0 \\ I & -K \\ 0 & 0 \end{bmatrix}^T \\
R &= E\left[\begin{bmatrix} w_k \\ v_k \end{bmatrix} \begin{bmatrix} w_k^T & v_k^T \end{bmatrix}\right] \\
\tilde{P}_i &= \pi_i \lim_{k\to\infty} E\left[z_k z_k^T | \text{ channel in state } i\right] \\
\tilde{P} &= \begin{bmatrix} \text{vec}(\tilde{P}_1) \\ \text{vec}(\tilde{P}_2) \end{bmatrix} \\
G &= \begin{bmatrix} \text{vec}(G_1) \\ \text{vec}(G_2) \end{bmatrix}.
\end{aligned} \tag{10}
$$

Then the following proposition can readily be obtained by using the results of [24].

Proposition 5 (Performance Analysis). *$\tilde{P}$ as defined in equation (11) is the unique solution to the following linear equation*

$$
\tilde{P} = \left(Q^T \otimes I\right) \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \tilde{P} + \left(Q^T \otimes I\right) \left(\begin{bmatrix} \pi_1 & 0 \\ 0 & \pi_2 \end{bmatrix} \otimes I \right) G,
$$

where I is the identity matrix and 0 is the zero matrix of suitable dimensions. The matrix Q is the transition probability matrix of the Markov chain and other quantities have been defined above.

Once we calculate $\tilde{P}$, we can readily evaluate the cost in equation (9) by using the following relations.

$$
\begin{aligned}
\tilde{P} &= \begin{bmatrix} \text{vec}(\tilde{P}_1) \\ \text{vec}(\tilde{P}_2) \end{bmatrix} \\
P^\infty &= \tilde{P}_1 + \tilde{P}_2 \\
P_x^\infty &= \begin{bmatrix} I & 0 & 0 \end{bmatrix} P^\infty \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix} \\
P_u^\infty &= F \begin{bmatrix} I & -I & -I \end{bmatrix} P^\infty \begin{bmatrix} I \\ -I \\ -I \end{bmatrix} F^T.
\end{aligned}
$$

Thus the cost can readily be calculated.

5.3 Example

In this section, we consider an example to illustrate the performance of the method outlined above. We consider the example system considered by Ling and Lemmon in [18]. The plant transfer function is

$$
H(z) = \frac{z^{-1} + 2z^{-2}}{1 + z^{-1} + 2z^{-2}},
$$

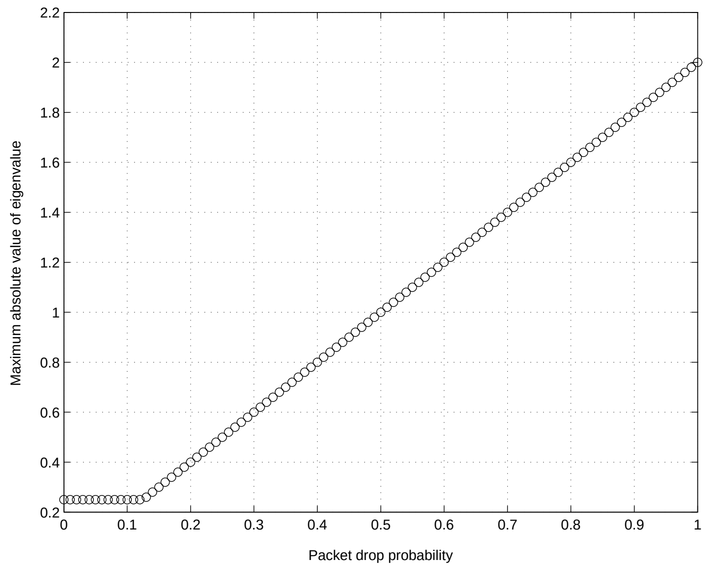

Figure 4: Stability margin of our algorithm as a function of packet drop probability.

so that the system evolves as

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Bw_k \\ y_k &= Cx_k, \end{aligned}$$

where the system matrices are given by

$$A = \begin{bmatrix} 0 & -2 \\ 1 & -1 \end{bmatrix} \qquad B = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

The process noise w_k is assumed zero mean with unit variance. The cost to be minimized is the steady state output error $\lim_{K\to\infty} y_K^2$. The work of Ling and Lemmon [18] assumes unity feedback when no packet is lost and gives an optimal compensator design when packets are being lost.

On analyzing the system with our algorithm, we observe that our algorithm allows the system to be stable up to a packet drop probability of 0.5 while the optimal compensator in [18] is stable only if the probability is less than 0.25. Figure 4 shows the stability margin of our algorithm as a function of the packet drop probability. It shows the maximum absolute value of the eigenvalue of the matrix given in Proposition 4. Also if we analyze the performance we obtain the plot given in Figure 5. The performance is much better throughout the range of operation for our algorithm. The performance of the two algorithms is not the same even at zero probability of packet drop since the optimal compensator presented in [18] assumes unity feedback. If we assume unity feedback in our algorithm and compare the performance, we obtain the plot in Figure 6. This shows that the difference in performance is mainly due to the novel encoding-decoding algorithm proposed.

6 Conclusions and Future Work

In this paper, we considered the problem of optimal LQG control in situations when the sensor and controller are communicating across a communication channel or a network. We modeled the communication medium as a switch that drops packets randomly. We showed that a separation principle exists between the optimal estimate and the optimal control law. For the optimal estimate, we identified the information that the sensor should provide to the controller. This can be viewed as constructing an encoder for the channel. We

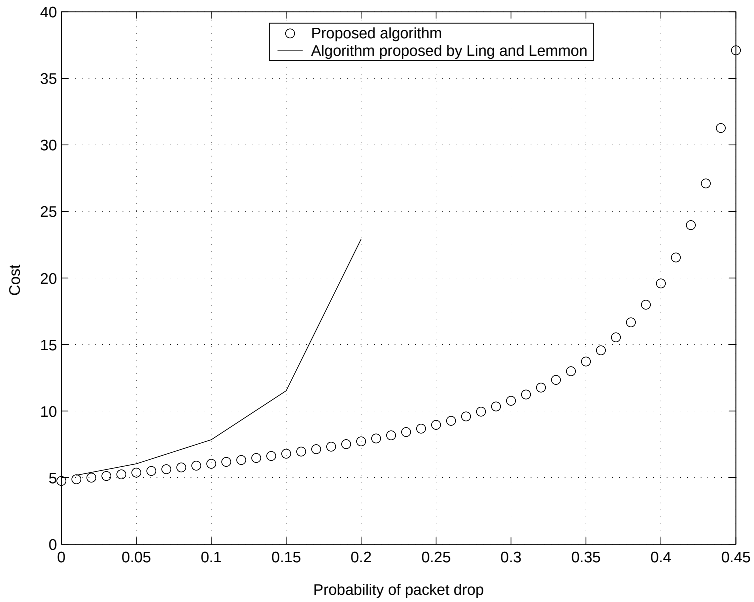

Figure 5: Comparison of performance for the two algorithms assuming optimal controller for our algorithm.

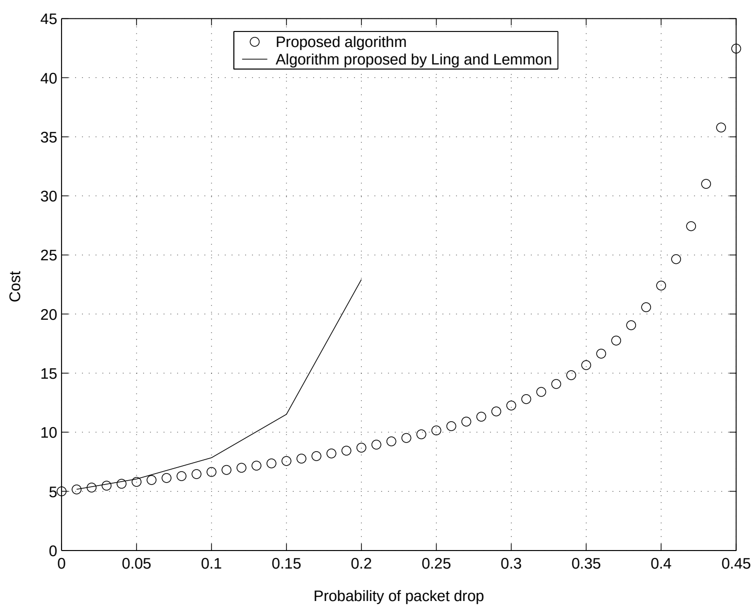

Figure 6: Comparison of performance for the two algorithms assuming unity feedback controller.

also provided the design for the decoder that uses the information it receives across the medium and the information it already has to construct an estimate of the state of the plant. We saw that our algorithm is optimal irrespective of the packet drop pattern. For the case of packet drops occurring according to a Markov chain, we carried out stability and performance analysis of our algorithm and compared it to other approaches presented in the literature.

The work can potentially be extended in many ways. One possible direction is to consider a channel between the controller and the actuator. In this scenario, it will be important to identify whether we want the decoder to have any knowledge of the control law or the cost function. Another intriguing possibility is considering the effect of allowing only finite number of bits in the packet. Ishwar et al. [15] have showed that if the optimal vector to send in the infinite rate case is the state estimate, even for the finite rate case, the quantized version of state estimate remains as the optimal thing to send. However, from the view of optimal control, this issue has to be examined in greater detail. Extensions to decentralized control are another exciting avenue of research.

References

[1] Special issue on networks and control. *IEEE Control Systems Magazine*, 21(1), Feb 2001.

[2] B. Azimi-Sadjadi. Stability of networked control systems in the presence of packet losses. In *Proceedings of 2003 IEEE Conference on Decision and Control*, Dec 2003.

[3] W.P. Blair and D.D. Sworder. Feedback control of a class of linear discrete systems with jump parameters and quadratic cost criteria. *Int. J. Contr.*, 21(5):833–841, 1975.

[4] R. W. Brockett and D. Liberzon. Quantized feedback stabilization of linear systems. *IEEE Transactions on Automatic Control*, 45(7):1279–89, 2000.

[5] D. F. Delchamps. Stabilizing a linear system with quantized state feedback. *IEEE Transactions on Automatic Control*, 35:916–924, 1990.

[6] N. Elia and S. K. Mitter. Stabilization of linear systems with limited information. *IEEE Transactions on Automatic Control*, 46(9):1384–1400, 2001.

[7] J. E. Elliot, L. Aggoun, and J. B. Moore. *Hidden Markov Models: Estimation and Control*. Springer-Verlag, 1995.

[8] V. Gupta, R.M. Murray, and B. Hassibi. On the control of jump linear markov systems with markov state estimation. In *Proc. of the American Control Conference*, 2003.

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

[10] A. Hassibi, S. P. Boyd, and J. P. How. Control of asynchronous dynamical systems with rate constraints on events. In *Proc. IEEE Conf. Decision and Control*, pages 1345–1351, Dec 1999.

[11] B. Hassibi, A. H. Sayed, and T. Kailath. *Indefinite-Quadratic Estimation and Control*. Studies in Applied and Numerical Mathematics, 1999.

[12] J. Hespanha, A. Ortega, and L. Vasudevan. Towards the control of linear systems with minimum bit-rate. In *Proc. of the 15th Int. Symp. Math. The. Netw. Sys.*, 2002.

[13] O. C. Imer, S. Yuksel, and T. Basar. Optimal control of dynamical systems over unreliable communication links. In *NOLCOS 2004*, Stuttgart,Germany, 2004.

[14] H. Ishii and B. A. Francis. Quadratic stabilization of sampled-data systems with quantization. *Automatica*, 39:1793–1800, 2003.

[15] P. Ishwar, R. Puri, K. Ramchandran, and S. S. Pradhan. On rate-constrained distributed estimation in unreliable sensor networks. *submitted to the IEEE Journal on Selected Areas in Communications: Special issue on self-organizing distributed collaborative sensor networks*, Dec 2003.

[16] P. Lancaster. *Theory of Matrices*. Academic Press, 1969.

[17] Q. Ling and M.D. Lemmon. Robust performance of soft real-time networked control systems with data dropouts. In *Proc. of the IEEE Conference on Decision and Control*, 2002.

[18] Q. Ling and M.D. Lemmon. Optimal dropout compensation in networked control systems. In *Proc. of the IEEE Conference on Decision and Control*, 2003.

[19] Q. Ling and M.D. Lemmon. Soft real-time scheduling of networked systems with dropouts governed by a Markov chain. In *Proc. of the American Conference on Control*, 2003.

[20] X. Liu and A. J. Goldsmith. Kalman filtering with partial observation losses. *IEEE Transactions on Automatic Control*. submitted.

[21] R. Luck and A. Ray. An observer-based compensator for distributed delays. *Automatica*, 26(5):903–908, 1990.

[22] G. N. Nair, S. Dey, and R. J. Evans. Infimum data rates for stabilising Markov jump linear systems. In *Proc. of the 42nd IEEE Conference on Decision and Control*, pages 1176–81, 2003.

[23] G. N. Nair and R. J. Evans. Stabilizability of stochastic linear systems with finite feedback data rates. *SIAM Journal on Control and Optimization*, 2004. accepted.

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

[25] P. Seiler. *Coordinated Control of unmanned aerial vehicles*. PhD thesis, University of California, Berkeley, 2001.

[26] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry. Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, Sep 2004.

[27] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, and S. S. Sastry. Time varying optimal control with packet losses. In *IEEE Conference on Decision and Control*, Bahamas, 2004. To be Presented.

[28] S. Tatikonda. *Control under Communication Constraints*. PhD thesis, MIT, Cambridge, MA, 2000.

[29] N. C. Tsai and A. Ray. Stochastic optimal control under randomly varying distributed delays. *International Journal of Control*, 68(5):1179–1202, Nov 1997.

[30] H. S. Witsenhausen. Separation of estimation and control for discrete time systems. *Proceedings of the IEEE*, 59(11):1557–1566, 1971.

[31] Q. Zhang and S. A. Kassam. Finite-state Markov model for Rayleigh fading channels. *IEEE T. Communications*, 47(11):1688–1692, 1999.

[32] W. Zhang, M. S. Branicky, and S. M. Philips. Stability of networked control systems. *IEEE Control System Magazine*, 21(1):84–89, Feb 2001.