Submitted, Systems and Control Letters
http://www.cds.caltech.edu/~murray/papers/2004o_gshm04-scl.html

Optimal LQG Control Across Packet-Dropping Links

Vijay Gupta *, Demetri Spanos, Babak Hassibi, Richard M Murray

Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA-91125, USA.

Abstract

We examine two special cases of the problem of optimal Linear Quadratic Gaussian control of a system whose state is being measured by sensors that communicate with the controller over packet-dropping links. We extend the LQG separation principle using a standard LQR state-feedback design, along with an optimal algorithm for propagating and using the information across the unreliable link. Our design is optimal for any 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.

Key words: LQG control, Networked control systems, Packet-dropping links, Separation principle

1 Introduction

Recently, much attention has been directed toward systems which are controlled over a communication link (see, e.g., [1,2] and the references therein). Understanding and counter-acting effects such as quantization error, random delays and packet drops that are introduced by communication links will become increasingly important as emerging applications of decentralized control mature. In this note, we consider systems communicating over links that randomly drop packets. The nominal system is shown in Figure 1, where the links

* Corresponding author.
Email addresses: vijay@cds.caltech.edu (Vijay Gupta), demetri@caltech.edu (Demetri Spanos), hassibi@caltech.edu (Babak Hassibi), murray@caltech.edu (Richard M Murray).

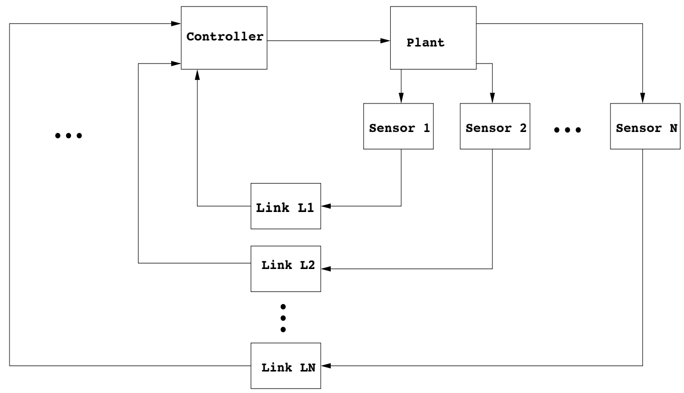

Fig. 1. The architecture of a packet-based control loop. The links are unreliable and unpredictably drop packets.

L_1, L_2, $\cdots$, L_N are communication channels or networks that randomly drop packets being communicated from the sensors to the controller. In particular, we discuss two special cases of the problem.

(1) Case C_1: There is only one sensor present.
(2) Case C_2: There are 2 sensors present. However, while L_1 drops packets randomly, L_2 transmits all packets.

While the case C_1 is important in its own right, it is also the basic system we need to understand for more general systems with multiple plants, sensors and controllers. Preliminary work for this case has studied stability and performance of systems utilizing lossy packet-based communication, e.g., in [18,20]. Approaches for compensation for data loss have been proposed, among others, by Nilsson [16] and by Ling and Lemmon [14], who posed the problem of optimal compensator design for the case when data loss is independent and identically distributed (i.i.d.) as a nonlinear optimization. A sub-optimal estimator and regulator to minimize a quadratic cost was proposed by Azimi-Sadjadi [3] and this approach was extended by Imer et al. in [13] and Sinopoli et al. in [8]. The related problem of optimal estimation across a packet-dropping link was considered by Sinopoli et. al in [19] and extended in [15]. However, most of the designs proposed in these references aim at designing a packet-loss compensator as shown in Figure 2. The compensator uses the successfully transmitted packets to come up with an estimate of the plant state. 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. Recognizing that sensors equipped with wireless or network communication capabilities will likely have some computational power available as well, we introduce an encoder at the sensor end. The compensator then effectively becomes a deocder for the information being transmitted over the link. We jointly design the controller, the encoder and the decoder to solve the optimal LQG problem.

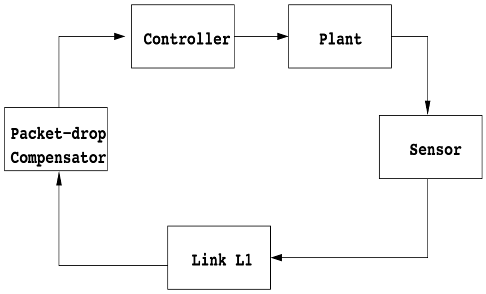

Fig. 2. The usual architecture for compensation for packet drops by the link.

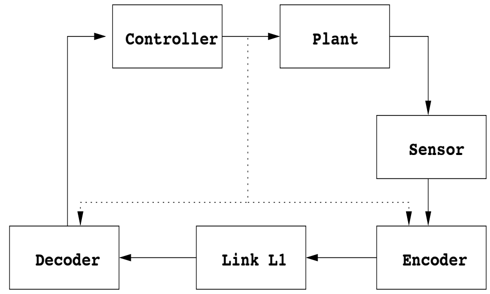

Fig. 3. The structure of our optimal LQG control solution.

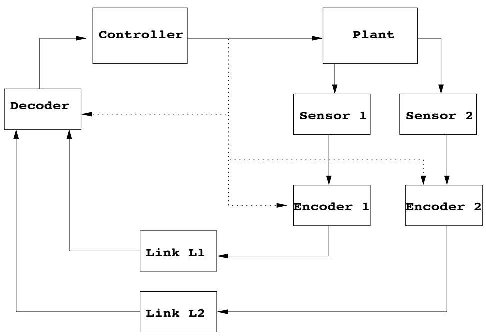

Fig. 4. The structure of our optimal LQG control solution for the two-sensor case.

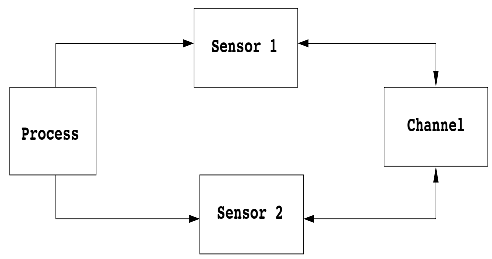

Fig. 5. Structure of the joint estimation problem.

There does not appear to be existing work dealing with the case C_2 specifically. We encounter this case in our work on the multi-vehicle wireless testbed [10]. In the testbed, each vehicle is equipped with an on-board gyro. In addition, each vehicle also obtains measurement from an overhead camera. While the gyro-controller link is hard-wired and hence does not drop packets, the camera communicates to the controller over a wireless link that randomly drops packets. Thus this situation is identical to the case C_2. Our solution to this problem again adopts the philosophy of using some computation at the sensor end to combat the effects of the channels. Our architecture is as shown in Figure 4. We again provide the optimal design of encoders, decoder and the controller.

An implicit assumption made in the entire paper is that the encoders and the decoder at time step k know the control signals applied to the plant till time step $k-1$. Depending on the particular application, this may or may not be a reasonable assumption. We present simulations later in the paper for the case when only a noise-corrupted version of the control signal is available to the encoders.

As an intermediate step, we will also solve the following problem. Suppose, as shown in Figure 5, two sensors are estimating a process jointly while communicating over a link that drops packets stochastically. What information should the sensors exchange? Related work to this problem has dealt with fusion of data from multiple sensors and track-to-track fusion. A usual starting point for such works is decentralization of the Kalman filter as described, e.g., in [11,17,21]. Alternative approaches include the Federated filter [6], Bayesian method [9] and many others. However all these approaches assume a fixed communication topology among the nodes with a link, if present, being perfect. Random loss of information due to the communication channels dropping packets reintroduces the problem of correlation between the estimation errors of various nodes as shown by Bar-Shalom [4] and renders the approaches proposed in the literature as sub-optimal. Schemes to counter this through state estimate fusion proposed, e.g., in [5] have been shown to not be optimal by Chang et al [7]. In this note, we solve for the optimal information to be transmitted by each sensor for the case of two sensors being present.

This paper is organized as follows. We begin in the next section by posing 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. Finally, we analyze the stability of our system and compare its performance with some other approaches in the literature.

2 Problem Formulation

Consider a discrete-time linear system evolving according to

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

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 process noise assumed to be 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 two sensors according to the equations

$$y_k^i = C^i x_k + v_k^i \qquad\qquad i = 1, 2. \tag{2}$$

The measurement noises v_k^i's are assumed white, zero-mean, Gaussian (with covariance matrix Q_v^i) and independent of the plant noise w_k and of each other. Note that substituting $C^2 = 0$ and $Q_v^2 = 0$ would reduce the case C_1 to be a special case of C_2. Hence, from now on, we will carry out the derivation for case C_2 only and adapt the results for the case of one sensor. Each sensor communicates its own measurements (or some function of the measurements) to the controller. We impose the constraint that the function communicated should be a finite vector, whose size does not increase with time. Sensor 1

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

communicates over link L_1 that randomly drops packets while sensor 2 utilizes link L_2 that is a perfect channel. For the moment we ignore delays and packet reordering in L_1; it will be shown that these effects can be accounted for with time-stamping and a slight modification to our design. The packet dropping in L_1 is a random process. We refer to individual (i.e. deterministic) realizations of this random process as *packet drop sequences*. A packet drop sequence P is a binary sequence $\{\eta_k\}_{k=0}^{\infty}$ in which η_k takes the value "*received*" if the link delivers the packet at time step k, and "*dropped*" otherwise.

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 which is undesirable for control applications. In order to make the class of controllers that are allowed more precise, we introduce the following terminology. Denote by s_k^i the finite vector transmitted from the sensor i to the controller at time step k. By causality, s_k^i can depend (possible in a time-varying manner) on y_0^i, y_1^i, $\cdots$, y_k^i, i.e., $s_k^i = f_k^i\left(y_0^i, y_1^i, \cdots, y_k^i\right)$. The *information set*, I_k available to the controller at time k is the union of two sets I_k^1 and I_k^2 defined by

$$I_k^1 = \{s_j^1 | \forall j \text{ s.t. } \eta_j = \text{ received}\} \qquad \text{and} \qquad I_k^2 = \{s_j^2 | \forall j = 0 \cdots k\}$$

Also denote by $t_l(k) \leq k$ the last time-step at which a packet was delivered over link L_1. That is $t_l(k) = \max\{j \leq k \mid \eta_j = \text{"}\textit{received}\text{"}\}$. The *maximal information set*, I_k^{max} at time-step k is then the union of I_k^2 and the set $I_k^{1,max}$ defined by $I_k^{1,max} = \{y_j^1 \mid 0 \leq j \leq t_l(k)\}$. 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 depends will be less than this set, since earlier packets, and hence measurements, may have been dropped. As stated earlier, the only restriction we impose is that the vectors s_k^i not increase in size as k increases. We will call the set of f_k^i's which fulfill this requirement as $\mathbf{F}$. Without loss of generality, we will only consider controllers of the form $u_k = u(I_k, k)$. We denote the set of control laws allowed by U. We shall assume perfect knowledge of the system parameters A, B, C, Q_w and Q_v^i's 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. Finally, as noted earlier, we assume that the encoders and the decoder at time step k know the control signals applied to the plant till time step $k-1$.

We can thus pose the packetized LQG problem as:

$$\min_{u\in U, f^i\in\mathbf{F}} J_K(u,f^i,P) = E\left[\sum_{k=0}^{K}\left(u_k^TQ^cu_k + x_k^TR^cx_k\right) + x_{K+1}^TP_{K+1}^cx_{K+1}\right]. \quad (3)$$

Here K is the horizon on which the plant is operated and the expectation is taken over the uncorrelated variables x_0, $\{w_k\}$ and $\{v_k^i\}$. Note that the cost functional J above depends on the random 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.* We now present our solution to the problem.

3 Separation of Control and Estimation

In this section, we re-visit the familiar separation principle in the packet-based setting of our problem. Consider the K-horizon cost functional given in (3). Following [12], we gather terms that depend on the choice of u_K and x_K and rewrite them as

$$T_K = E\left[u_K^TQ^cu_K + x_K^TR^cx_K\right] + E\left[x_{K+1}^TP_{K+1}^cx_{K+1}\right] = S_K + O_K$$

$$S_K = E\left[\begin{bmatrix}u_K^T & x_K^T\end{bmatrix}\Delta\begin{bmatrix}u_K\\ x_K\end{bmatrix}\right] \qquad O_K = E\left[w_K^TP_{K+1}^cw_K\right]$$

$$\Delta = \begin{bmatrix}Q^c + B^TP_{K+1}^cB & B^TP_{K+1}^cA\\ A^TP_{K+1}^cB & R^c + A^TP_{K+1}^cA\end{bmatrix}$$

Thus we can write

$$J_K(u,f^i,P) = E\left[\sum_{k=0}^{K-1}u_k^TQ^cu_k + \sum_{k=0}^{K-1}x_k^TR^cx_k\right] + S_K + O_K. \quad (4)$$

We aim to choose u_K to minimize $J_K(u,f^i,P)$ for given f^i's. From (4), it is clear that the only term where u_K enters is S_K. S_K can be written as

$$\begin{aligned}
S_K &= E\left[\left(u_K-\bar{u}_K\right)^TR_{e,K}^c\left(u_K-\bar{u}_K\right)\right] + E\left[x_K^TP_K^cx_K\right]\\
R_{e,K}^c &= Q^c + B^TP_{K+1}^cB\\
P_K^c &= R^c + A^TP_{K+1}^cA - A^TP_{K+1}^cB\left(Q^c + B^TP_{K+1}^cB\right)^{-1}B^TP_{K+1}^cA,
\end{aligned}$$

where $\bar{u}_K$ is the standard optimal LQ control, $\bar{u}_K = -\left(R_{e,K}^c\right)^{-1}B^TP_{K+1}^cAx_K$. In the absence of the packetized link, the controller could simply use the standard optimal control $\bar{u}_K$. However, this control law does not lie in the set of allowable solutions U because it is not realizable for any non-trivial packet-dropping sequence. Instead, we will calculate u_K based on the information set I_K (and the previous controls $u_0, u_1, \cdots, u_{K-1}$) and choose it to minimize S_K. The control problem thus reduces to an optimal estimation problem. We denote the least mean square (lms) estimate of a random variable Γ based on the information set at time k, I_k, and the previous controls by $\hat{\Gamma}_{|I_k}$. Then we can write the optimal control at time step K as

$$u_K = \hat{\bar{u}}_{K|I_K} = -\left(R_{e,K}^c\right)^{-1} B^T P_{K+1}^c A \hat{x}_{K|I_K}. \tag{5}$$

Thus, we only need to find the lms 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^i's that will, surprisingly, allow the errors to actually coincide.

Denote the estimation error incurred due to the minimizing choice of u_K by Υ_K. Note that Υ_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 (5). Thus we can write

$$J_K(u, f^i, P) = J_{K-1}(u, f^i, P) + \Upsilon_K + O_K.$$

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 Υ_K do not involve these control inputs). But 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 proved the following.

Proposition 1 (Separation) *Consider the packet-based optimal control problem defined in section 2. 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*

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

where $\bar{u}_k$ is the optimal LQ control law while $\hat{\alpha}_{k|I_k}$ denotes the lms estimate of α given the information set I_k and the previous control laws $u_0, \cdots, u_{k-1}$.

Remarks:

(1) 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 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.

(2) Note that we have not yet said anything about the design of the encoders or the decoder for coming up with the estimate $\hat{x}_{k|I_k}$ (e.g., whether they are linear or not). Proposition 1 simply says that whatever be the way information is encoded and then decoded, given an information set I_k on which the control has to depend, the best thing to do is to calculate $\hat{\hat{u}}_{k|I_k}$. In the next section, we will give a design for which $\hat{\hat{u}}_{k|I_k}$ coincides with $\hat{\hat{u}}_{k|I_k^{max}}$ at each time step.

(3) Since all past controls are supposed to be available at both the encoder and the decoder, control does not have a dual effect in this problem.

4 Optimal Encoder and Decoder Design

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^i's that will allow the controller to compute that. If L_1 does not drop packets, sending the current measurement y_k^i in the current packets is sufficient. When L_1 randomly drops packets, a naíve solution would be to send the entire history of the output variables at each time step. However, as mentioned earlier, this is not allowed since 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. We propose the following algorithm. Denote by $\hat{x}_{k|l}^i$ the estimate of x_k based on all the measurements of sensor i up to time l and all previous control inputs. Also denote the corresponding error covariance by $P_{k|l}^i$.

Optimal Transmission and Estimation Algorithm:

- *Encoder for sensor 1:* At each time step k,
  - Obtain measurement y_k^1 and run a local Kalman filter for $\hat{x}_{k|k}^1$ and $P_{k|k}^1$.
  - Calculate $\lambda_k^1 = \left(P_{k|k}^1\right)^{-1} \hat{x}_{k|k}^1 - \left(P_{k|k-1}^1\right)^{-1} \hat{x}_{k|k-1}^1$.
  - Calculate global error covariance matrices $P_{k|k}$ and $P_{k|k-1}$ using

$$\left(P_{k|k}\right)^{-1} = \left(P_{k|k-1}\right)^{-1} + \left(C^1\right)^T \left(Q_v^1\right)^{-1} \left(C^1\right) + \left(C^2\right)^T \left(Q_v^2\right)^{-1} \left(C^2\right)$$
$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q_w.$$

· Obtain $\gamma_k = \left(P_{k|k-1}\right)^{-1} A_{k-1} P_{k-1|k-1}$.

· Finally calculate $i_k^1 = \lambda_k^1 + \gamma_k i_{k-1}^1$ with $i_{-1}^1 = 0$ and transmit it.

- *Encoder for sensor 2:* At each time step k, transmit the measurement y_k^2.
- *Decoder:* At each time step k,

· Use y_k^2 to come up with i_k^2 using an algorithm similar to the one followed by the encoder for sensor 1.

· Maintain a local variable $\hat{x}_k^{dec}$ which is updated as follows.

(1) If $\eta_k = received$, both links L_1 and L_2 have successfully transmitted packets. In that case, calculate $\psi_k = \left(P_{k|k-1}\right)^{-1} Bu_{k-1} + \gamma_k \psi_{k-1}$ with $\psi_0 = 0$ and obtain the estimate through

$$\left(P_{k|k}\right)^{-1} \hat{x}_k^{dec} = i_k^1 + i_k^2 + \psi_k.$$

(2) If $\eta_k = dropped$, only L_2 has transmitted the packet. In this case, propagate the estimate $\hat{x}_{k-1}^{dec}$ using the measurement y_k^2 and the control u_{k-1} through a Kalman filter.

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

PROOF. Consider a centralized filter that has access to measurements from a sensor of the form

$$y_k = Cx_k + v_k$$

where

$$C = \begin{bmatrix} C^1 \\ C^2 \end{bmatrix} \qquad v_k = \begin{bmatrix} v_k^1 \\ v_k^2 \end{bmatrix}. \tag{6}$$

Let R be the covariance matrix of the noise v_k. Since R is block-diagonal, the time and measurement update equations of the Kalman filter are

$$\left(P_{k|k}\right)^{-1} = \left(P_{k|k-1}\right)^{-1} + C^T R^{-1} C = \left(P_{k|k-1}\right)^{-1} + \sum_i \left[\left(P_{k|k}^i\right)^{-1} - \left(P_{k|k-1}^i\right)^{-1}\right]$$

$$\begin{aligned} \left(P_{k|k}\right)^{-1} \hat{x}_{k|k} &= \left(P_{k|k-1}\right)^{-1} \hat{x}_{k|k-1} + C^T R^{-1} y_k \\ &= \left(P_{k|k-1}\right)^{-1} \hat{x}_{k|k-1} + \sum_i \left[\left(P_{k|k}^i\right)^{-1} \hat{x}_{k|k}^i - \left(P_{k|k-1}^i\right)^{-1} \hat{x}_{k|k-1}^i\right] \\ P_{k|k-1} &= AP_{k-1|k-1}A^T + Q_w \qquad \hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_{k-1}. \end{aligned}$$

Thus at time step k, the covariance matrices can be calculated offline while for the estimate the sensor i needs to send $\Lambda_k^i = \left(P_{k|k}^i\right)^{-1} \hat{x}_{k|k}^i - \left(P_{k|k-1}^i\right)^{-1} \hat{x}_{k|k-1}^i$. We can write

$$\left(P_{k|k}\right)^{-1}\hat{x}_{k|k} = \left(P_{k|k-1}\right)^{-1}\hat{x}_{k|k-1} + \sum_i \Lambda_k^i = \sum_i I_k^i + \Psi_k$$

$$I_k^i = \Lambda_k^i + \Gamma_k \Lambda_{k-1}^i + \Gamma_k \Gamma_{k-1} \Lambda_{k-2}^i + \cdots + \left(\Gamma_k \Gamma_{k-1} \cdots \Gamma_1\right) \Lambda_0^i$$

$$\Psi_k = \left(P_{k|k-1}\right)^{-1} B u_{k-1} + \Gamma_k \Psi_{k-1} \qquad \Gamma_k = \left(P_{k|k-1}\right)^{-1} A P_{k-1|k-1},$$

with $\Psi_0 = 0$. In the above derivation, we have used the fact that x_0 was zero mean and thus $\hat{x}_{0|-1} = 0$. Thus, the information needed from sensor i at time step k is precisely I_k^i. Now for the case when $\eta_k = received$, the decoder in the algorithm has access to i_k^1 and i_k^2 that are the same as I_k^1 and I_k^2. Thus it can calculate the centralized Kalman filter output $\hat{x}_{k|k}$ which is $\hat{x}_{k|I_{max}}$. For the case when $\eta_k = dropped$, the decoder propagates the best Kalman filter estimate $\hat{x}_{k-1|k-1}$ with sensor 2's measurement. Thus in this case too, $\hat{x}_k^{dec} = \hat{x}_{k|I^{max}}$ □

Proposition 2 presents the solution to the estimation problem posed in case C_3 since we can use an encoder and a decoder described in the algorithm at each sensor. Moreover, taken together, propositions 1 and 2 solve the packet-based LQG control problem posed in Section 2.

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

Remarks:

(1) Note that the computation and memory required for calculating I_k^i does not grow with time since we can use the recursion $I_k^i = \Lambda_k^i + \Gamma_k I_{k-1}^i$.

(2) The information vector I_k^i 'washes away' the effect of any previous packet losses, If $\eta_k = received$, $\hat{x}_{k|k}$ is calculated as if all the previous measurements from both sensors were available.

(3) We have made no assumption about the packet dropping behavior. The algorithm 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 or unknown to the plant and the controller.

(4) For the case C_1, the algorithm reduces to the following:
- The encoder (at the sensor end) receives as input the measurement y_k. It runs a Kalman filter that provides the llms estimate of x_k based on all the measurements until time step k, denoted by $\hat{x}_{k|k}$ and transmits this vector across the link.
- The decoder (at the controller end) maintains a local variable $\hat{x}_k^{dec}$. It is updated as follows:

· If $\eta_k = received$, the decoder receives $\hat{x}_{k|k}$, and sets $\hat{x}_k^{dec} = \hat{x}_{k|k}$.
· If $\eta_k = dropped$, then the decoder implements the linear predictor:

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

(5) The solution can readily 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, if we assume that there is a provision for time-stamping the packets sent by the encoder. For ease of notation, we present the solution for optimal asynchronous estimation for the case C_1. The case C_2 is similar. At each time step, the decoder will face one of four possibilities, and will update its estimate as described below:
- It receives $\hat{x}_{k|k}$. It uses this as its estimate.
- It does not receive anything. It uses the predictor equation (7) on $\hat{x}_{k-1}^{dec}$.
- 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 (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 obtains $\hat{x}_k^{dec}$ through (7).

(6) Note that we do not assume knowledge of the cost matrices Q and R at the sensor end. Thus the controller can be changed at will without affecting the sensor/encoder operation. This is important, e.g., in our MVWT work where the matrices Q and R are user-specified while the encoder code is much harder to change.

(7) As pointed out by Imer et al in [13] if we have a channel between the controller and the plant, the separation principle would still hold, provided there is a provision for acknowledgment over the channels.

5 Analysis of the Proposed Algorithm

In this section, we model the channel erasures as occurring according to a Markov chain and analyze the stability and performance of our design. Thus the channel exists in either of two states, state 1 corresponding to a packet drop and state 2 corresponding to no packet drop and it transitions probabilistically between these states according to the transition probability matrix Q. Note that i.i.d. drops can be handled by a special choice of Q. We assume strict causality in the Kalman filter used by the encoder. Thus to calculate the estimate of x_k, only the measurements till time step $k-1$ are used. The analysis for the causal case is similar. Finally we assume that (A, B) is stabilizable and the pair (A, C) is detectable, where C is defined in (6). We will denote the Kronecker product of matrices A and B by $A \otimes B$.

We begin with the stability analysis. Denote by y_k the vector formed by stacking y_k^1 and y_k^2. We have three dynamical systems. The plant state x_k evolves as in (1). The state $\hat{x}_k$ of a centralized Kalman filter with access to measurements from both sensors at every time step would evolve as

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

Finally the state $\hat{x}_k^{dec}$ of the estimator at the decoder evolves according to

$$\hat{x}_{k+1}^{dec} = \begin{cases} A\hat{x}_k^{dec} + Bu_k + K_k^d \left(y_k^2 - C^2\hat{x}_k^{dec}\right) & \text{channel in state 1} \\ \hat{x}_{k+1} & \text{otherwise.} \end{cases}$$

Denote $e_k = x_k - \hat{x}_k$ and $t_k = \hat{x}_k - \hat{x}_k^{dec}$. Since $u_k = F_k\hat{x}_k^{dec}$, (1) implies

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

Since (A, B) is stabilizable and F_k is the optimum control law, the system would be stable in the bounded covariance sense as long as the disturbances w_k, t_k and e_k have bounded covariances. We assume the noise w_k has bounded covariance matrix. Also e_k has bounded covariance matrices by our detectability assumption. Finally t_k evolves according to

$$t_{k+1} = \begin{cases} \left(A - K_k^d C^2\right) t_k + L^1(e_k) + L^2(v_k^1) + L^3(v_k^3) & \text{channel in state 1} \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

where $L^n(\beta)$ denotes a term linear in β. Again note that v_k^i's and e_k have bounded covariance. For t_k to be of bounded variance, the Markov jump system of (8) needs to be stable. Finally, 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. We can thus say the following.

Proposition 4 (Stability Condition) *Consider the control problem defined in Section 2 in which the packet erasure channel is modeled as a Markov chain with transition probability matrix $Q = [q_{ij}]$. Let the matrix pair (A, B) be stabilizable and the matrix pair (A, C) be detectable. The system is stabilizable, in the sense that the variance of the state is bounded, if and only if $q_{22}|\lambda_{\max}\left(\bar{A}\right)|^2 < 1$, where $\lambda_{\max}\left(\bar{A}\right)$ is the maximum magnitude eigenvalue of the unobservable part of matrix A when (A, C^2) is put in the observer canonical form. Further, if the system is stabilizable, one controller and encoder/decoder design that stabilizes the system is given in Proposition 3.*

Using the results of [16], we can also calculate the total quadratic cost incurred by the system for the infinite-horizon case (the case when $K \to \infty$ in (3)) if we make the additional assumption that the Markov chain is stationary and regular. We state the result for the case C_1. We consider the cost

$$J_\infty = \lim_{K\to\infty} E\left[x_K^T R^c x_K + u_K^T Q^c u_K\right] = \text{trace}\left(P_x^\infty R^c\right) + \text{trace}\left(P_u^\infty Q^c\right), \quad (9)$$

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

$$P_x^\infty = \begin{bmatrix} I & 0 & 0 \end{bmatrix} P^\infty \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix} \qquad P_u^\infty = F \begin{bmatrix} I & -I & -I \end{bmatrix} P^\infty \begin{bmatrix} I \\ -I \\ -I \end{bmatrix} F^T,$$

where $P^\infty = \tilde{P}_1 + \tilde{P}_2$ and $\tilde{P} = \begin{bmatrix} \mathrm{vec}(\tilde{P}_1)^T & \mathrm{vec}(\tilde{P}_2)^T \end{bmatrix}^T$. Then, it can be shown that $\tilde{P}$ is the unique solution to the linear equation

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

In the above equation, $A_i = \mathbf{A}_i \otimes \mathbf{A}_i$, and $G = \begin{bmatrix} \mathrm{vec}(G_1)^T & \mathrm{vec}(G_2)^T \end{bmatrix}^T$, where

$$\mathbf{A}_1 = \begin{bmatrix} A + BF & -BF & -BF \\ A - KC & 0 & 0 \\ 0 & -KC & A \end{bmatrix} \qquad \mathbf{A}_2 = \begin{bmatrix} A + BF & -BF & -BF \\ A - KC & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{B}_1 = \begin{bmatrix} I & 0 \\ I & -K \\ 0 & -K \end{bmatrix} \qquad \mathbf{B}_2 = \begin{bmatrix} I & 0 \\ I & -K \\ 0 & 0 \end{bmatrix} \qquad G_i = \mathbf{B}_i \begin{bmatrix} Q_w & 0 \\ 0 & Q_v \end{bmatrix} \mathbf{B}_i^T.$$

Example

We now consider some examples to illustrate the performance of our algorithm. First, we consider the example system considered by Ling and Lemmon in [14]. The system evolves as

$$x_{k+1} = \begin{bmatrix} 0 & -2 \\ 1 & -1 \end{bmatrix} x_k + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u_k + \begin{bmatrix} 2 \\ 1 \end{bmatrix} w_k.$$

There is only one sensor of the form

$$y_k = \begin{bmatrix} 0 & 1 \end{bmatrix} x_k.$$

The process noise w_k is zero mean with unit variance and the packet drop process is i.i.d. The cost considered is the steady state output error $\lim_{K\to\infty} y_K^2$.

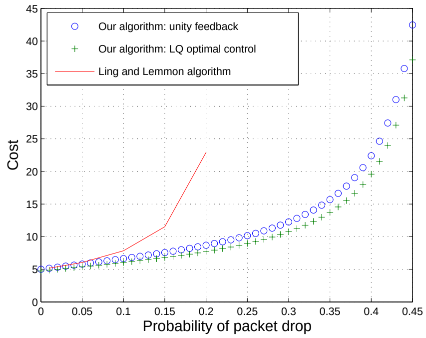

Fig. 6. Comparison of performance for our algorithm with the one obtained using optimal compensator.

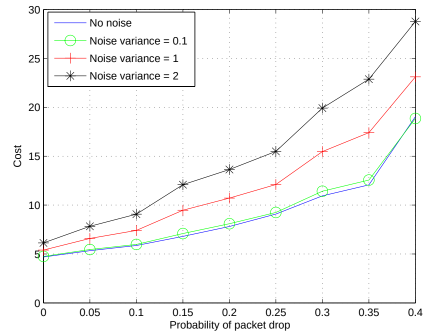

Fig. 7. Comparison of performance for when the encoder has access only to a noise corrupted value of the control.

[14] assumes unity feedback when packets are delivered 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 [14] is stable only if the probability is less than 0.25. Also if we analyze the performance we obtain the plot given in Figure 6. The performance is much better throughout the range of operation for our algorithm, even if we assume unity feedback in our algorithm. This shows that the difference in performance is mainly due to the novel encoding-decoding algorithm proposed. In the above plots we assumed that the encoder had perfect access to the control signal. Figure 7 shows the performance when the encoder uses a noise corrupted value of the control. Four different curves for noise variances 0, 0.1, 1 and 2 are plotted. The curves show the simulated expected performance for the system. We see that the increase in stability margin remains valid in all four cases. Furthermore, even though the performance degrades as the noise is increased, the performance still remains better than the no encoding strategy.

In the next example, we consider the same system being observed through two sensors of the form

$$y_k^1 = \begin{bmatrix} 1 & 0 \end{bmatrix} x_k + v_k^1 \qquad\qquad y_k^2 = \begin{bmatrix} 0 & 1 \end{bmatrix} x_k + v_k^2.$$

The sensor noises are zero mean with variance 10 and 1 respectively. We consider the cost function $\lim_{K\to\infty} \left(y_K^2\right)^2$. Figure 8 shows the simulated performance of our algorithm as a function of the packet loss probability. We also plot the performance for a hypothetical sensor that received information from both sensors without any packet drop and for a scheme in which sensors exchange only measurements. It can be seen that even in this very simple case, our algorithm can lead to a performance gain of up to 40% over simply sending measurements.

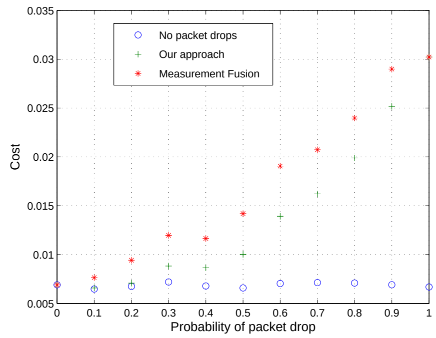

Fig. 8. Comparison of performance for the two sensor case.

6 Conclusions and Future Work

In this paper, we considered the problem of optimal LQG control when the sensor and controller are communicating across a channel or a network. We modeled the link as a switch that drops packets randomly and proved that a separation 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 also designed the decoder that uses the information it receives across the link to construct an estimate of the state of the plant. The proposed 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.

The work can potentially be extended in many ways. One obvious extension is to consider multiple sensors and communication links. Another intriguing possibility is considering the effect of allowing only finite number of bits in the packet.

References

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

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

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

[4] Y. Bar-Shalom, *On the track-to-track correlation problem"*, IEEE Trans. Automat. Contr., AC-26(2):571-572, 1981.

[5] Y. Bar-Shalom and L. Campo, *The effect of the common process noise on the two-sensor fused-track covariance*, IEEE Trans. Aerospace and Electronic Systems, AES-22(6):803-805, 1986.

[6] N. A. Carlson, *Federated square root filter for decentralized parallel processes*, IEEE Trans. Aerospace and Electronic Systems, AES-26(3):517-525, 1990.

[7] K. C. Chang, R. K. Saha and Y. Bar-Shalom, *On optimal track-to-track fusion*, IEEE Trans. Aerospace and Electronic Systems, AES-33(4):1271-1276, 1997.

[8] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla and S. S. Sastry, *LQG Control with Missing Observation and Control Packets*, IFAC World Congress on Automatic Control, 2005.

[9] C. Y. Chong, S. Mori and K. C. Chang, *Information Fusion in Distributed Sensor Networks*, Proc. of the American Control Conference, pp 830-835, 1985.

[10] L. Cremean, B. W. Dunbar, D. van Gogh, J. Hickey, E. Klavins, J. Meltzer and R. M. Murray, *The Caltech Multi-Vehicle Wireless Testbed*, Proc of the 2002 Conference on Decision and Control, 2002.

[11] H. R. Hashemipour, S. Roy and A. J. Laub, *Decentralized structures for parallel Kalman filtering*, IEEE Trans. Automat. Contr., AC-33(1): 88-94, 1988.

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

[13] O.C. Imer, S. Yuksel and T. Basar, *Optimal control of LTI systems over communication networks*, Automatica, (submitted 2005).

[14] Q. Ling and M.D. Lemmon, *Optimal Dropout Compensation in Networked Control Systems*, Proc. of the IEEE Conference on Decision and Control, 2003.

[15] X. Liu and A. J. Goldsmith, *Kalman Filtering with partial observation losses*, IEEE Trans. Automat. Contr., (submitted 2004).

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

[17] B. S. Rao and H. F. Durrant-Whyte, *Fully decentralized algorithm for multi-sensor Kalman filtering*, IEE proceedings, 138(5), 1991.

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

[19] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan and S. Sastry, *Kalman Filtering with Intermittent Observations*, IEEE Trans. Automat. Contr., Sep 2004.

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

[21] B. C. Levy, D. A. Castanon, G. C. Verghese and A. S. Willsky, *A scattering framework for decentralized estimation problem*, Automatica, 19(4): 373-384, 1983.