LQG Control for Networked Control Systems with Random Packet Delays and Dropouts via Multiple Predictive-Input Control Packets

Maryam Moayedi*, Yung K. Foo**, Yeng C. Soh***

*Nanyang Technological University, Singapore (Tel: 65-67906349; e-mail: mary0008@e.ntu.edu.sg).
**EW Electrical and Mechanical Engineering Private Limited, Singapore (e-mail: jooyk@leunwah.com.sg)
***Nanyang Technological University, Singapore (Tel: 65-67906349; e-mail: eycoh@ntu.edu.sg).

Abstract: This article addresses the Linear Quadratic Gaussian (LQG) optimal control problem for networked control systems when data is transmitted through a TCP-like network and both measurement and control packets are subject to random transmission delays and packet dropouts. Instead of adopting a “zero-input” or “hold-input” strategy, we propose a “predictive optimal-input” strategy, where at each time instant, $K$ future control inputs are computed and sent in addition to the current one in the same control packet. The stabilizability conditions of the system are then derived. Simulation results are presented to demonstrate the effectiveness of the proposed approach.

1. INTRODUCTION

Networked control systems (NCSs) where data networks are used for connections between spatially distributed components of a control loop have recently attracted much attention. While using a communication network in NCS offers many advantages, it also leads to new challenges. Random time delay and data packet dropout are two main unavoidable problems in data transmission over unreliable communication networks,[10]. Usually the packets in NCSs suffer both random time-varying delays and dropout during the network transmission which would in general have a detrimental effect on the stability and performance of NCSs.

Existing works in control and stability in NCS context mostly considered the packet loss or packet delay uncertainty in data transmission through the network. See for example [1]. However, we do remark that the results of [1] have been also extended to allow for a constant unit time delay in the control input, assuming the states are available to the controller for state feedback control. LQG optimal control problem over lossy TCP-like channels with randomly dropped sensor and control packets according to a Bernoulli process has been investigated in [12] and [11]; [3] also considered the problem of optimal LQG control when sensors and the controller are communicating across a packet erasure channel.

There are quite a number of results considering both time delays and packet losses in NCSs. Most of them only considered the stabilization rather than performance e.g. [7], [17] or they assumed all of the states of the system are available for state feedback control, ([5]) or they considered delay and/or dropout in only one of the channels, [3]. In [4] and [7], discrete-time networked control system in the presence of random delays and dropouts is considered. However, both works assumed that all of the states are available to the controller for the state-feedback control. [17] addressed the stability analysis and synthesis problems for a class of NCSs under the effect of both delay and dropout where the full state vector is not available but the network is only present between the sensor to the controller. [6] dealt with the control problem of networked systems in which random delays, packet loss and limitations of the communication channels between sensors, actuators, and controllers were taken into account. However, no stability analysis, taking the arrival probability of packets into account was presented.

A Predictive controller where, at each sampling instant, a finite number ($K$) of future control inputs are computed and sent in addition to the current one in the same control packet (of size $K+1$) has been investigated in [8] via a “receding horizon” approach. It was then shown via simulations that under this scheme the overall performance of the closed-loop system, in terms of lower cost, has improved. However, the approach of [8] was developed to handle packet dropouts only (We acknowledge that this would not rule out applications of the approach to NCS involving both packet delays and losses because one could always discard a delayed packet and treat it as a lost packet).

Moreover, the solutions obtained in [8] is dependent on the availability of some transitional probabilities, $q_{ij}$’s — the probabilities of the system being in state $j$ at time step $k-1$ given that the system was in state $i$ at time step $k$ (i.e. a back-ward type of conditional probability, see equation 3 of [8]), and are hence quite tedious to compute. Apart from that, the complexity of the computations of $q_{ij}$’s with increase in $K$, increases exponentially. The problems with the availability of $q_{ij}$’s and the corresponding computational complexity involving $q_{ij}$, as well as the desirability of considering both packet delays and dropouts in measurement and control packets with the stability analysis, motivate us to
propose a “predictive-input” strategy for the LQG control of networked control systems, with both measurement and control input packets subject to random delay and loss and where the states are not available to the controller.

The paper is organized as follows. Section 2 will provide the mathematical formulation of the problem. In section 3, the optimal estimation problem with possible packet delay and dropout in both measurement and control input is studied. The controller design, i.e., predictive optimal inputs, is then derived in section 4. In section 5, the main results of the paper, i.e., the finite and infinite horizon LQG under TCP-like protocols via multiple predictive input control packets and the stabilizability analysis of the system with predictive-optimal input packets, are presented. In section 6, we give some examples to illustrate the applicability and effectiveness of proposed LQG filter-controller design schemes. Finally we give our conclusions in section 7.

2. PROBLEM FORMULATION

Consider the following discrete-time linear time-invariant system:

\[ x(k+1) = Ax(k) + Bu^r(k) + w(k) \]
\[ z(k) = Cx(k) + v(k) \]

\[
y(k) = \begin{cases} \sum_{i=0}^{d} \alpha_{k,i} z(k-i) + \sum_{i=0}^{d} \beta_{k,i} y(k) & \text{if } \gamma_i = 1 \\
\text{unavailable} & \text{if } \gamma_i = 0 \end{cases}
\]  

(2.1)

where \( u^r(k) \) is the actual input applied to the actuator. \( y(k) \) is the measurement received by the estimator at time step \( k \), \( x(0), z(k), v(k) \) are uncorrelated, white, with mean \( \{x_0, 0, 0\} \) and covariance \( \{P_0, Q, R\} \), respectively. \( \alpha_{k,i}, \beta_{k,i} \) are random variables that may take values of 0 or 1 only, which model the delay suffered by the measurement packets. \( d \) denotes the maximum delay a measurement packet is allowed (longer than which the packet would be discarded and treated as lost), and \( \gamma_i \) is a Bernoulli random process with \( \gamma_i \) taking value of either 0 or 1 with probabilities \( \Pr(\gamma_i = 1) = \gamma \). The stochastic variable \( \gamma_i \) models the loss of packets between the sensor and the estimator - if the estimator receives a measurement packet with a delay of no more than \( d \) steps at discrete time \( k \), then \( \gamma_i = 1 \), otherwise \( \gamma_i = 0 \).

It is further assumed that arrival of packets from the controller to the actuator containing information about control input to be applied may be characterized by another sets of random variables \( \beta_{k,i} \) that may take values of 0 or 1 to model the delay suffered by the control packets, and another \( \{0, 1\} \) Bernoulli random variable \( \nu \) with probability \( \Pr(\nu = 1) = \nu \) to model packet losses. If the actuator receives a control packet with delay of less than or equal to some \( D \) at time step \( k \), then \( \nu = 1 \) otherwise \( \nu = 0 \).

Suppose the remote controller has decided that \( u(k) \) is the desired input to be applied at the actuator at time \( k \). However, because of network condition, the packet containing this information may or may not reach the actuator on time. As we have already mentioned, we assume the network adopted a TCP-like protocol, [11], so that the acknowledgement is always available to the controller, on time and with no loss, if the control packet has been received by the actuator. It has been shown that under such condition, the separation property holds and the optimal control law is in the form of \( u(k) = L(k)x_i(k/k) \) where \( x_i(k/k) \) is the Kalman estimate of \( x(k) \) (after measurement update, [11]) at time \( k \). The objective of the estimation-control law is to minimize the following cost function:

\[ J_s = E[x(N)^T W_s x(N) + \sum_{k=0}^{N-1} (x(k)^T W_e x(k) + a^*(k)^T U_s u^*(k)) \]  

(2.2)

where \( I_{N-1}, x, P_0 \) is defined in section 3, \( U_s > 0 \) and \( W_e \geq 0 \). We define \( u^s(k+i) \) as the control input calculated by the controller at time step \( k \) corresponding to the optimal control input to be applied at time \( k+i \). Thus the controller computes and dispatches a packet containing \( K+1 \) projected control inputs \( \{u^s(k), u^s(k+1), \ldots, u^s(k+K)\} \) (instead of a packet containing only one control input \( u(k) \)) at each discrete time step \( k \) (Note \( u^s(k) \) is the same as \( u(k) \) ) from the estimation-control unit to the actuator. The actuator maintains a buffer to store one of the previous control packets received. If at time \( k \) a new control packet arrives, then it checks the time-stamp of this packet - if it is later than the one in the buffer, then it replaces the previous packet with this packet in the buffer. It then looks into the packet stored in the buffer and finds out how old it is. If this packet corresponds to a packet time-stamped (and hence dispatched by the controller) at time step \( k-m \) where \( m \leq K \), it then applies \( u^s(k) = u^{s-m}(k) \) (Note that this packet would have contained \( \{u^{s-m}(k-m), u^{s-m}(k-m+1), \ldots, u^{s-m}(k-m+K)\} \). Otherwise (i.e. \( m > K \)), it sets \( u^s(k) = 0 \).

We shall refer to this strategy as a “predictive-input” strategy. Obviously, to apply this strategy, we wish to compute \( \{u^s(k), u^s(k+1), \ldots, u^s(k+K)\} \) such that (1.2) is minimized and it is required that \( u^s(k+j) \) will be in the linear form of \( u^s(k+j) = F_{k+j} x^s(k+j) \) where \( F_{k+j} \) are pre-computable off-line and \( x^s(k+j) \) denotes the best (predictive) estimate of \( x(k+j) \) at time \( k \). Note that this strategy requires the packet to contain \( K+1 \) future control inputs, and this generally may be achieved without any additional communication cost, [8], if \( K \) is not excessively large.

3. ESTIMATOR DESIGN

3.1 Optimal State Estimation
The problem of the optimal state estimation under the conditions we have considered (i.e. with packet delays and dropouts) can easily be solved as in [9] or [10]. The reader is referred to the two references for more detail. We denote the optimal state estimate at time \( k \), as \( x_\epsilon^v(k) \).

But because of the possibility of packet delays and dropouts between the controller and actuator, the computed desired control input \( u(k) = F_k x_\epsilon^v(k / k) \) may not be available to the actuator at time \( k \). In fact, if \( m \ (0 < m \leq K) \) consecutive control packet dropouts have occurred immediately before time \( k \), then the best state estimate that may be “deemed available to the actuator” via \( u^{k-m}(k) = F_k x^{k-m}(k) \) at time \( k \) is \( x^{k-m}(k) \), where \( x^{k-m}(k) \) denotes the best estimate for \( x(k) \) based on the information set \( F_{k-m} \) defined as follows:

\[
F_{k-m} \Delta \{ y^{k-m}, u^{k-m} \} = \{ y(k-m), y(k-m-1), ..., y(0) \}, \text{ and } u^{k-m} = \{ u^{(k-1)}, u^{(k-2)}, ..., u^{(0)} \}.
\]

Define \( x^{k-m}_v(k) = E[x(k) | F_{k-m}] \) \( e^{k-m}(k) = x(k) - x^{k-m}_v(k) \) \( \Lambda^{k-m}(k) = E[ e^{k-m}(k) e^{k-m}(k)^T ] | F_{k-m} \) \( (3.1) \)

Correspondingly, \( y^{k-m} \) may be regarded as the set of measurements deemed available to the actuator at time step \( k \). Similarly, if \( z(j) \), \( j \leq K - m \) is contained in \( y^{k-m} \), then \( z(j) \) may be deemed available to the actuator at time step \( k \).

Let \( I^*_m \) denote the set of information on input and measurements which is “deemed available” to the actuator at time \( k \). Clearly, if \( F_{k-m} \) is the latest (and hence the largest set of) information (deemed available to the actuator at time \( k \), then we may denote the information deemed available to the actuator at time \( k \) by \( I^*_m = F_{k-m} \). In such case, \( x_v^{k-m}(k) = E[x(k) | I^*_m] = E[x(k) | F_{k-m}] \) would be the best estimate of \( x(k) \) (deemed available to the actuator).

The relevant equations for the optimal state estimates deemed available to the actuator at some future time \( k+m \) may then be (conceptually) computed as follows:

**Compute best current state estimate based on \( I^*_m \):**

For \( j = k-d \) to \( k \) do (3.4)-(3.9):

\[
x^{j+1}_v(k) = A x^{j}_v(k) + B u^{j}_v(k), \quad x^{0}_v(k) = x_0 \quad (3.4)
\]

\[
P^j_v = AP^j_v A^T + Q \quad (3.5)
\]

Update \( j \) to \( j + 1 \)

\[
x^{j}_v(k) = x_v^{j+1}(k) + \beta_j K_j [ y(j) - C x^{j-1}_v(j) ] \quad (3.6)
\]

\[
P^j_v = (I - \beta_j K_j C)^T P^{j+1}_v (I - \beta_j K_j C) + \beta_j^2 K_j R K_j^T \quad (3.7)
\]

\[
K_j = \begin{cases} 
P^{j-1}_v C^T (C P^{j-1}_v C^T + R)^{-1} & \text{if } \beta_j = 1 \\
0 & \text{if } \beta_j = 0 
\end{cases} \quad (3.8)
\]

where \( \beta_j = 1 \) if \( z(j) \) may be deemed available to the actuator at time step \( k \), and \( \beta_j = 0 \) otherwise.

Furthermore, if \( u^k(k), u^{k+1}(k), ..., u^{k+K}(k) \) are known a priori, then the predictive state estimates may be given by the following recursive equation:

**Compute best future state estimate based on \( I^*_m \):**

\[
x^{j+1}_v(k+j+1) = A x^{j}_v(k+j) + B u^{j}_v(k+j), \quad j = 1, 2, 3, ..., K \quad (3.9)
\]

In particular, if \( u^*(k+j) = F_{k+j} x^{*(k+j)} \) (assuming \( F_{k+j} \) known), then we may obtain

\[
x^{j}_v(k+j+1) = (A + BF_{k+j}) x^{j}_v(k+j), \quad j = 1, 2, 3, ..., K \quad (3.10)
\]

\[
P^j_v = AP^j_v A^T + Q \quad (3.11)
\]

**4. CONTROLLER DESIGN (PREDICTIVE OPTIMAL INPUTS)**

The probability of \( K+1 \) consecutive packet dropout is \((1-\nu)^{K+1}\). Hence, the probability of \( u^*(k) = 0 \) is \((1-\nu)^{K+1}\).

**Lemma 4.1:** Suppose \( x_v^{k-m}(k) = E[x(k) | I^*_m] \). Then the following facts are true:

- \( E[(x(k) - x_v^{k-m}(k)) y^{k-m}_v(k)^T | I^*_m] = E[e^{k-m}(k) x_v(k) y^{k-m}_v(k)^T | I^*_m] = 0 \)
- \( E[x(k) C_x(k)] | I^*_m] = x^{k-m}_v(k)^T S_x x^{k-m}_v(k) \)
- \( + E[e^{k-m}(k)^T S_x x^{k-m}_v(k)] = x^{k-m}_v(k)^T S_x x^{k-m}_v(k) + tr(S_x P_{k-m}^{k-m}) \)
- \( E[g(x(k+1)) | I^*_m] | I^*_m] = E[g(x(k+1)) | I^*_m] , \text{ all } g(.) \)

**Proof:** Similar to lemma 4.1 of [11], with \( F_i \) replaced by \( I^*_m \).

Note that because the separation property holds, in general the estimation \( x_v(k) \) does not affect the controller gain at time \( k \). Therefore, we may consider the modified LQ cost function

\[
J_w = E[x(N)^T \Omega_{xw} x(N) + \sum_{k=0}^{N-1} E[x(k)^T W_x x(k)] + \nu_x(k) u^{k-m}_v(k)^T U_u u^{k-m}_v(k) | I^*_m, x_0, P_x] \quad (4.1)
\]

where \( \nu_x(k) = 1 \) if \( u^*(k) = u^{k-m}_v(k) \) where \( m \) may take any value from 0 to \( K \) as explained before and \( \nu_x(k) = 0 \) if \( u^*(k) = 0 \). To facilitate the computations of the control law, we approximate \( \nu_x(k) \) by an i.i.d. Bernoulli variable with probabilities \( \text{Prob} \{ \nu_x = 1 \} = (1-(1-\nu)^{K+1}) \) and \( \text{Prob} \{ \nu_x = 0 \} = (1-\nu)^{K+1} \).

Suppose that the best available estimate for \( x(k) \) deemed available to the actuator at discrete instance \( k \) is \( x^{k-m}_v(k) \) with the estimation error covariance \( P_{k-m}^{k-m}(x) \). We then apply the dynamic programming approach based on the cost-to-go iterative procedure to derive the optimal control law. Define the optimal value function \( V^*_v(x(k)) \) as below:
\[ V_q(x(N)) \Delta E[x(N)'W_qx(N) \mid I'_q] \quad (4.2) \]
\[ V'_q(x(k)) \Delta \min E[x(k)'W_qx(k) \mid \{w_{(i)} \}] \]
\[ + V_{q'}(k) u(k)' U_q u(k) + V_{q'}(x(k+1)) \mid I'_q \]  
where \( k = N-1, \ldots, 0 \). Using the dynamic programming theory, we know that \( J^*_q = V_q(x_0) \).

**Lemma 4.2:** For the system defined by (2.1) under TCP-like protocols, the value function \( V_q(x(k)) \) can be written as
\[ V_q(x(k)) = E[x(k)' \Gamma_q x(k) \mid I'_q] + c_q(k) \]  
where the matrix \( \Gamma_q \) and the scalar \( c_q(k) \) may be computed recursively as follows:
\[ \Gamma_q = W_q + A' \Gamma_q A \]
\[ -(1-(1-\nu)^{k+1}) A' \Gamma_q B (U_q + B' \Gamma_q B)^{-1} B' \Gamma_q A \quad (4.5) \]
\[ c_q = tr(\Gamma_q + W_q - \Gamma_q) E[m(k)|I'_q] \]
\[ + tr(\Gamma_q Q) + E[c(k)|I'_q] \quad (4.6) \]
with initial values \( \Gamma_q = W_q \) and \( c_q = 0 \). Moreover, the optimal control decision is given by
\[ u^*(k) = -(U_q + B' \Gamma_q B)^{-1} B' \Gamma_q A x^{\pi(k)} - (U_q + B' \Gamma_q B)^{-1} B' \Gamma_q A \quad (4.7) \]
where \( F_q = \Delta \Gamma_q - (U_q + B' \Gamma_q B)^{-1} B' \Gamma_q A \quad (4.8) \)

**Proof:** Similar to proof of lemma 5.1 in [11], with \( F_q \) replaced with \( I'_q \), omitted due to space constraint.

**Remark 1:** Note that in Lemma 4.2, we have expressed \( m(k) \) as \( m(k) \). This is because at different time \( k \), the corresponding \( m \) may vary. We further note that the probabilistic distribution of \( m(k) \) in general depends on both \( \nu \) and \( v \).●

### 5. THE MAIN RESULTS

#### 5.1 Finite Horizon LQG under TCP-like Protocols via Multiple Predictive Input Control Packets

Consider the system (2.1) and the problem of minimizing the cost function (2.2). Then the suboptimal control is a linear function of the estimated state given by equation (4.7) and (4.8) where the matrix is computed recursively by (4.5). The separation principle holds under TCP-like protocol. The optimal state estimator is given by (3.4)-(3.12).

Therefore the combined estimation-control scheme may be summarized by the following conceptual algorithm:

**Conceptual Algorithm**

1. Controller pre-computes \( F_q, k = 0, 1, 2, \ldots, N \) via (4.8).
2. **Filter’s actions:** a. Filter estimates \( x_k(k/l) \) based on \( F_q \).
   b. Filter dispatches \( x_k(k/l) \) to Controller.
3. **Controller’s actions:**
   a. Controller assigns \( x_{k}^k = x_k(k/l) \) and computes
      \[ x_{k}^k(\text{j} + 1), \quad j = 0, 1, 2, \ldots, K-1 \]  
      via (3.10)
   b. Controller computes \( u_{k}^k(\text{j} + 1) = F_{k,j} x_{k}^k(\text{j} + 1) \), \( j = 0, 1, 2, \ldots, K-1 \) via (3.10)

**4. Actuator’s action**

a. Controller dispatches control packet containing \( \{u_{k}^k, u_{k}^k(\text{j} + 1), \ldots, u_{k}(k + K)\} \) to Actuator

**5.2 Stabilizability of the System with Predictive Optimal-inputs Packets**

For simplicity and ease of exposition we shall first consider the case of only possible packet dropout but not packet delay for the control packets, and later we extend it to possible delay and dropout in the control input.

We know that delay in measurement in general does not cause the Kalman filter to diverge, [9]. Since the separation property applies for the TCP-like networked systems [11], for the purpose of analyzing the stability of the control scheme we shall assume \( \gamma = 1 \) and \( \Delta_{i,j} = 0 \) for \( i \neq 0 \), for simplicity. As mentioned earlier, at each discrete time \( k \), we may set \( u_{k}^k(\text{j} + 1) = F_{k,j} x_{k}^k(\text{j} + 1) \), \( j = 0, 1, 2, \ldots, K \), and dispatch them in a control packet to the actuator. Thus, the optimal input to be applied at time \( k \) is \( F_{k,j} x_{k}^k(\text{j}) \), where \( x_{k}^k(\text{j}) \) is some minimum error variance estimate of \( x_{k}^k(\text{j}) \) available to the controller-actuator at time \( k \).

Restricting ourselves to the case where there is no delay (i.e. there is only packet dropout), we consider an equivalent system of which the controller is co-located with the actuator so that the controller-actuator knows what \( F_q \) is. Suppose the controller-actuator depends on “some” (fictitious) estimator (which is different from the real estimator of the closed-loop system) to provide it with the best \( x_{k}^k(\text{j}) \) and we assume any \( x_{k}^k(\text{j}) \) obtained and dispatched by the (fictitious) estimator can reach the controller-actuator instantaneously and without fail. Hence, the control input is \( u_{k}^k(\text{j} + 1) = F_{k,j} x_{k}^k(\text{j} + 1) \) and the control action behaviour of the original system (with multi-control packet of size \( K \) and \( \gamma = 1 \) in Fig. 1) will be equivalently modelled by the control action behaviour of the equivalent system (with single-step-input control packet) shown in Fig. 2 with the following probabilities:

- **Prob (control-packet loss between (fictitious) estimator and controller-actuator = 0**;
- **Prob (measurement loss between sensor and (fictitious) estimator = (1-\nu)**;

The (fictitious) estimator set \( x_{k}^k(\text{j}) = 0 \) after it fails to receive measurement in last consecutive \( K \) sampling time.

It follows that the closed-loop system will be stable if and only if the “fictitious” state estimations are stable.

Next, we consider the case in which delay is permitted. If delayed control packets are admitted, then when a delayed control packet arrives at the actuator at time step \( k \), two cases could happen. The first one is when the packet has arrived
before all of its successors. In such case, it will be stored in the buffer for control purposes (and hence considered “useful”) and we may deem it as a delayed measurement has arrived. However, if it arrives after one or more of its successors, it will be discarded and treated as a lost packet. Nevertheless, all the measurements that are deemed to have incorporated, would have been incorporated by its successors arrived earlier, so no information is actually lost despite its being discarded. Hence, a delayed control packet arrival may be deemed as a delayed measurement arrival at the actuator. Since delays in measurements in general do not lead the state estimations to instability as long as they arrive eventually [9], employing the same argument as above shows the stability of the overall closed-loop will not be affected by the delays of the control packets, if $\gamma = 1$.

![Diagram](image)

**Proposition 5.1** (necessary condition): Suppose $A$ of system (2.1) is unstable. Let $(A, C)$ be observable, $(A, Q^{1/2})$ and $(A, B)$ be controllable and $R > 0$. Assume the probability of control packet loss between the controller and actuator is $1 - \nu$. Then the system cannot be stabilized via a linear regulator using multiple predictive optimal-input control packets of size $K+1$ and zero-input strategy if

$$\nu > 1 - (\max_i \lambda_i^\ast(A) i^{-1})^{-1} \quad (5.2)$$

where $\lambda_i^\ast(A)$ are the unstable eigenvalues of $A$.

**Proof:** The proof is omitted due to space constraint.

**Remark 2:** Proposition 5.2 implies theorem 5.1 holds.

5.3 Infinite Horizon LQG under TCP-like Protocols via Multiple Predictive Input Control Packets

**Theorem 5.1** (Infinite Horizon LQG under TCP-like protocols via multiple predictive input control packets): If $W_V = W$ and $U_V = U$ are constants, then a positive definite solution for $\Gamma_\infty$ exists for the following discrete-time Riccati equation $\Gamma_\infty = W + A^T \Gamma_\infty A - (1 - (1 - \nu)\gamma)A^T \Gamma_\infty B (U_V + B^T \Gamma_\infty B)^{-1} B^T \Gamma_\infty A$. If $(1 - (1 - \nu)\gamma) > 1 - (\max_i \lambda_i^\ast(A) i^{-1})$. Furthermore, the multiple-input control packets with $u^i(k + j) = F_u x^i_j (k + j)$, $j = 0, 1, 2, ..., K$, stabilizes the system in closed-loop with $F_u$ $A^T + (U + B^T \Gamma_\infty B)^{-1} B^T \Gamma_\infty A$ if $\nu > 1 - (\max_i \lambda_i^\ast(A) i^{-1})^{-1}$

where $\lambda_i^\ast(A)$ are the unstable eigenvalues of $A$.

**Proof:** The proof is omitted due to space constraint.

6. EXAMPLES

**Example 1:** We consider the system used in [8]:

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = [1 \ 0]$$

The noise processes $\nu(k)$ and $w(k)$ are assumed to be zero mean, with variances $R = 0.5$ and $Q = I_{2 \times 2}$ respectively. $N = 30$ (horizon length), $W_V = C^T C$, $U_V = 0.3$, $W_u = C^T C$ (final cost-weight matrix) and $x(0) = 0$. We use the same estimation scheme and compare the performance of the time-varying predictive optimal input controller with control packet size $K+1$ where $K = 0, 1, 2, 3$ and 4.

We construct and simulate results using $\gamma = 1$ and $\nu = 0.6, 0.7$, and 0.8. The average (over 200 simulations) of empirical costs versus different values of control packet arrival probability is shown in Fig. 3 where we have incorporated the performance of the single-control packet based zero-input strategy and hold-input strategy for comparison. We note that $K = 0$ is equivalent to zero-input strategy of [11].
Fig 3. LQG cost for different controllers versus control packet arrival probability.

Fig. 4. LQG cost vs. number of control inputs per packet for different probabilities of control packet arrival.

Fig 4 plots the LQG costs versus different number of control inputs per packet for different probabilities of control packet arrival. From Fig 4, it can be seen that the LQG cost decreases as the number of control inputs per packet is increased but it is getting flatter as the number of control inputs is further increased, especially for higher probabilities of control input arrival. This is because for high probabilities of arrival, the chance of consecutive packet dropouts is small. One may take the “knee-point” of leaving-off as the “optimal” packet size.

Next, we focus on \( \nu = 0.6 \) to investigate the effect of admitting a one-step delay. Let \( \rho_1, \rho_2 \) and \( \rho_3 \) denote the probabilities that the packet which arrives at the actuator is a current control packet, a one-step delayed control packet, and one with more than one-step delay or no packet arrives at the actuator respectively. We assume that by admitting a one-step delay, the value of \( \nu \) may be increased to 0.72 (with \( \rho_1 = 0.6, \rho_2 = 0.12 \) and \( \rho_3 = 0.28 \). Note that \( \rho_1 \) and \( \rho_2 \) are not required for the purpose of controller design and if we decide not to admit the one-step delay control packet and discard it and treat it as a packet loss, then \( \nu \) decreases back to 0.6.) The average of the empirical costs over 200 simulations versus \( K = 0, 1, 2, 3, 4 \) have been computed and tabulated in Table 1 below. For a more realistic simulation with precedence constraint [2], we have used the following conditional probabilities for the simulation of one-step delays and packet dropouts (\( \rho_{i1} = \text{Pr}(\text{the system is at state } i \text{ at time step } k | \text{ the system has been at state } j \text{ at time step } k - 1) \)).

<table>
<thead>
<tr>
<th>( \nu )</th>
<th>( J_{K=0} )</th>
<th>( J_{K=1} )</th>
<th>( J_{K=2} )</th>
<th>( J_{K=3} )</th>
<th>( J_{K=4} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6</td>
<td>1.6708</td>
<td>1.3075</td>
<td>1.1661</td>
<td>1.1232</td>
<td>1.1084</td>
</tr>
<tr>
<td>0.72</td>
<td>1.4951</td>
<td>1.1843</td>
<td>1.1195</td>
<td>1.1046</td>
<td>1.1007</td>
</tr>
</tbody>
</table>

From the table, it is clear that when \( \nu = 0.72 \), the LQG cost for predictive-input controller would be smaller than the cost for the case \( \nu = 0.6 \) for different values of \( K \). In other words admitting a one-step delayed packet as a delayed packet (i.e. not a lost one) in the controller design, helps to improve the system performance.

6. CONCLUSIONS

In this paper, we considered the LQG control problem over a communication network where both the measurement and control may be delayed or lost. We adopted a multiple-control packet-based predictive-input strategy. Simulations results showed that the strategy proposed in this paper performs better than the single-control packet based zero-input strategy and the hold-input strategy.

REFERENCES