2018 Annual American Control Conference (ACC)
June 27–29, 2018. Wisconsin Center, Milwaukee, USA

Coded Control over Lossy Networks

Ye Pu, Jingge Zhu, Karl H. Johansson, Kannan Ramchandran and Claire J. Tomlin

Abstract— **We consider a networked control system with an unreliable feedback link from the sensor to the controller. Specifically, a discrete-time linear system is to be controlled via a packet-drop channel where multiple packets are transmitted at each time instance. We propose a *coded control scheme* that jointly designs the coding strategy, which mitigates the channel unreliability, and the control strategy, which stabilizes the unstable system. This scheme is based on the idea of successive refinement, that more important system states (or linear combinations thereof) should be better protected against the unreliable channel. The proposed scheme is simple to implement as it uses a static encoder and decoder/controller, in the sense that all encoding and decoding procedures do not require information from previous time steps. Furthermore, we compare it with two other static schemes and show that our approach strikes a good balance between optimality and complexity.**

I. Introduction

Networked control systems (NCS) are distributed systems, where the communications between plants, sensors or/and controllers are subject to certain constraints. Most studied are the scenarios where the feedback link from the sensor to the controller is implemented via an unreliable communication channel. Various aspects of the problem are studied under different assumptions on the feedback link. For example, in the set-up where the feedback channel is a bit-pipe channel with a finite data rate, the classical *data-rate theorem* (see, e.g. [16]) provides a necessary condition for stabilizing the plant, characterized by the unstable eigenvalues of the system. In the set-up where the feedback channel is modelled as a discrete memoryless channel, the notion of *anytime capacity* [10] gives the necessary and sufficient condition under which the moments of the system states can be stabilized. In [12], a packet-drop channel is considered as the unreliable feedback link, where a real-valued vector is received with a certain probability by the controller in each time instance. A Kalman filter with intermittent observations is developed for this system in [12] to address the problem of estimating the states of a linear system over the erasure channel. The minimum erasure probability is found under which the estimation error is bounded. For a system without an explicit feedback constraint, a linear quadratic Gaussian (LQG) control problem is studied in [13] where prefix-free binary codes are used to feed back the system states. Both upper and lower bounds for the code rate are determined for a given control cost. NCS with a packet erasure channel is studied in [7], where repetition codes are used to improve the control performance. Authors in [4] studied the system with one controller and two sensors, where the outputs of the sensors are connected to the controller via an unreliable channel. For a more comprehensive overview of the results, the reader is referred to the survey papers [8], [5]. In addition to the theoretical results, practical coding techniques have also been developed for NCS. For example, LDPC (low-density parity check) codes are used in [14] to control a linear system with unreliable feedback links, and [11] exploited the real-time nature of rateless codes to stabilize a linear system with noisy feedback.

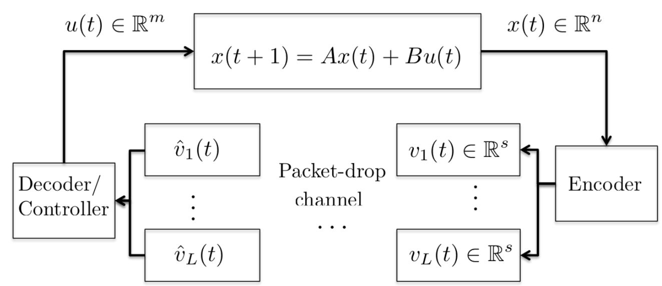

Fig. 1: A networked control system. The system state $x(t) \in \mathbb{R}^n$ is to be encoded into L vectors (packets) $v_\ell \in \mathbb{R}^s, \ell = 1, \dots, L$, and transmitted to the decoder. Every packet experiences independent erasures, and is received by the decoder/controller with probability $1-p$.

In this work, we study a system similar to that in [12], where the feedback link between the sensor and the controller is modeled by a discrete time packet-drop channel, with the difference that multiple packets are transmitted in each time instance (see Fig. 1). This model is relevant for the scenarios where a (possibly distributed) system is to be controlled over a wireless communication channel, and also provides new degrees of freedom that we could leverage in the system design. Particularly, we explore the idea of *successive refinement* from information theory, which assigns different priorities to the system states (or linear combinations thereof) according to their importance to the closed-loop system performance. Specifically, the paper makes the following contributions.

- A coded control strategy is proposed based on the principle of successive refinement, that more important system states (or linear combinations thereof) are better protected against the unreliable channel, and have a higher probability being utilized by the controller.

Ye Pu, Jingge Zhu, Kannan Ramchandran and Claire J. Tomlin are with the EECS Department, University of California, Berkeley, CA, USA, e-mail: {yepu, jingge.zhu, kannanr, tomlin}@eecs.berkeley.edu. Y. Pu and J. Zhu are supported by the Swiss NSF under Projects P2ELP2_165137 and P2ELP2_165155.

Karl H. Johansson is with the School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden, e-mail: kallej@kth.se.

- We provide an algorithm which *jointly* optimizes the coding scheme and the control policy based on the channel characteristic and the control specification. That is, the problems of coding against the unreliable channel and controlling the unstable system are addressed in a common framework.
- We compare our strategy with two other static schemes: the first scheme solves the control and coding problems separately hence is of low complexity; the second scheme, though with high complexity, jointly addresses the two problems in a globally optimal way. Numerical results show that the proposed scheme strikes a good balance between the two extremes. It achieves performance comparable to that of the optimal scheme, at the same time maintaining a lower complexity.

II. Problem statement

Consider a discrete-time linear system of the form

$$x(t+1) = Ax(t) + Bu(t) \tag{1}$$

where $x(t)$ and $u(t)$ denote the system state and input, respectively, and $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$. The system is assumed to be unstable, i.e, the absolute value of the largest eigenvalue of A is larger than 1.

The networked control system in consideration is depicted in Figure 1. We assume that the system state $x(t)$ is perfectly observed by an encoder. The unreliable channel is modelled as a so-called *packet-drop channel*. More precisely, at each time instance, L packets are sent from the encoder, and each packet arrives at the decoder/controller independently with probability $1-p$, where $p \in [0, 1]$ denotes the *erasure probability* of the channel. Notice that in contrast to some earlier work where only one packet is sent out at each time instance ([12], [5] for example), we assume that multiple packets are transmitted by the encoder. This assumption is justified under a variety of scenarios. If the actual feedback link is a wireless communication channel, due to the short coherence time of typical wireless channels [15], the encoder needs to send out multiple short packets instead of a long packet, so that the channel impulse response can be essentially invariant for one packet. Moreover, the packet-drop channel with multiple packets is also a suitable model for scenarios in which there are several non-collocated sensors, each of which has an observation of the system state. In this case, the total number of transmitted packets (from multiple sensors) is always larger than one in each time instance, and the packet(s) sent out by different sensors will be received by the controller via independent wireless channels.

We further assume that each packet contains a real-valued vector, as in previous works [5]. This assumption is appropriate when each packet holds sufficiently many bits so that the quantization effect can be ignored, but packet losses cannot. More precisely, a length-s vector is contained in each packet at time t, denoted as $v_\ell(t) \in \mathbb{R}^s, \ell = 1, \ldots, L$, and is assumed to be generated directly using the system state as

$$[v_1, \cdots v_L(t)] = \mathcal{E}(x(t)), \quad \ell = 1, \ldots, L. \tag{2}$$

With a slightly abuse of notation, we use $\hat{v}_\ell(t)$ to denote the received information. Under the erasure channel model, we have either $\hat{v}_\ell(t) = v_\ell(t)$ if the packet is received, or $\hat{v}_\ell(t) = e$, denoting that an erasure has occurred. At each time instance, a controller $\mathcal{C}$ generates the control signal $u(t)$ using the received packets from the current transmission

$$u(t) = \mathcal{C}(\hat{v}_1(t), \ldots, \hat{v}_L(t)). \tag{3}$$

We emphasize the fact that although $u(t)$ can be generated using all the information available to the controller (including all received information from previous time steps), we focus on a static decoder/controller in the form of (3) in the current paper.

Since packets are sent through a stochastic channel, the resulting control signal $u(t)$ and the system state $x(t)$ are also stochastic. In this regard, we will adopt the following definition of stability.

Definition II.1 (Second moment stable). *The system (1) is said to be second moment stable, if for any $x(0) \in \mathbb{R}^n$, it holds that $\lim_{t\to\infty} \mathbb{E}||x(t)||^2 = 0$.*

Sometimes we impose a more stringent requirement on the convergence of the system, given by the decay rate.

Definition II.2 (Decay rate). *Let $\beta \geq 1$, the system (1) is said to have a decay rate β, if for any $x(0) \in \mathbb{R}^n$, it holds that $\lim_{t\to\infty} \beta^t \mathbb{E}\,||x(t)||^2 = 0$.*

Now we can formally state our control problem as follows.
Problem: Given the discrete-time system in (1) and the packet-drop channel with parameters L (number of packets), s (size of each packet), and p (erasure probability), design the encoder and controller in the form of (2) and (3), respectively, such that the closed-loop system has a certain decay rate.

III. A successive refinement approach

In this section, we propose a simple control strategy inspired by the principle of successive refinement. A similar idea was used in the multiple description coding scheme in [9], and in the coded computation strategy in [18]. We also point out that the multiple description coding scheme is also used in [6] for state estimation over packet-drop channels.

A. The coding scheme

We first describe the controller. With simplicity in mind, we use a linear controller based on the decoded information. Namely,

$$u(t) = K(t)\tilde{x}(t) \tag{4}$$

where $\tilde{x}(t)$ is generated using a decoder $\mathcal{D}$ with the received packets:

$$\tilde{x}(t) = \mathcal{D}(\hat{v}_1(t), \ldots, \hat{v}_L(t)).$$

The encoder $\mathcal{E}$ and the decoder $\mathcal{D}$ are to be designed such that the following property is satisfied.

Definition III.1 (Successive refinement). *Let* $D_1, \ldots, D_L$ *be L matrices with the dimension* $D_\ell \in \mathbb{R}^{r_\ell \times n}, \ell = 1, \ldots, L$. *If any* ℓ *out of the total* L *packets are received by the controller, the controller can recover* $D_1 x, \ldots, D_\ell x$. *That is, we have*

$$\tilde{x}(t) = C_\ell x(t) = \mathcal{D}(\hat{v}_1(t), \ldots, \hat{v}_L(t)) \tag{5}$$

if any ℓ *packets are received, where we define*

$$C_\ell := \begin{pmatrix} D_1 \\ \vdots \\ D_\ell \end{pmatrix} \in \mathbb{R}^{(\sum_{i=1}^{\ell} r_\ell) \times n}. \tag{6}$$

The successive refinement coding scheme has the property that certain quantities are given higher priority than others during the transmission. More precisely, the vector $D_1 x(t) \in \mathbb{R}^{r_1}$ has the highest chance being correctly decoded by the controller. Indeed, it can be recovered with probability $1 - p^L$, when any one out of L packets is received. The term $D_2 x(t) \in \mathbb{R}^{r_2}$ has the second highest priority with a recovery probability of $\sum_{\ell=2}^{L} \binom{L}{\ell} (1-p)^\ell p^{L-\ell}$, when any two out of L packets are received, and so on. As there are L packets containing sL numbers in total, there is a tension between the length of the vector $D_\ell x(t)$ and its priority. For example, if we choose r_1 equal to s (the length of the packet), it is easy to see that the only coding option is to set $v_\ell(t) = D_1 x(t)$ to all $\ell = 1, \ldots, L$, which results $r_1 = s$ and $r_\ell = 0$ for all $\ell \neq 1$.

If the successive refinement requirement is satisfied, there are in total $L+1$ different resultant $\tilde{x}(t)$, determined by the matrix $C_\ell, \ell = 1, \ldots, L$. We set $C_0 = 0$ to include the case when no packet is received, where $\tilde{x}(t) = 0$. For each C_ℓ we can design a corresponding linear controller to generate the control signal as

$$u(t) = K_\ell C_\ell x(t). \tag{7}$$

where $K_\ell \in \mathbb{R}^{m \times \sum_{i=1}^{L} r_\ell}$ for $\ell = 1, \ldots, L$ and C_ℓ given in (6). We point out that as we allow $r_\ell = 0$ for some ℓ (as shown in the example from Fig. 2), it can happen that $C_\ell = C_{\ell'}$ for some $\ell \neq \ell'$.

Remark III.2 (Static scheme). *The current strategy sets* $C_0 = 0$, *and consequently* $u(t) = 0$ *in this case. This means that if no packet is received, the controller does not apply any control signal and the system evolves uncontrolled as* $x(t+1) = Ax(t)$. *Obviously in this case, a control signal could still be generated using the last known system states, e.g. letting* $u(t) = u(t-1)$. *More generally, we could obtain a better estimation of the current system state by exploiting previously received information. However, we do not consider this option, as our current design methodology insists on static controller which does not estimate current system state using past information. In general, static output feedback stabilization is an important open question in control theory, and more information could be found in, e. g. [1].*

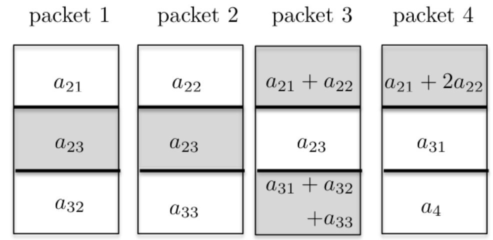

Fig. 2: A successive refinement coding scheme. The plot shows the generated packet $v_1(t), \ldots, v_4(t)$ from the information $D_\ell x(t), \ell = 1, \ldots, 4$. Since $r_1 = 0$, the controller is not required to decode with the information from one packet. It can be checked that the controller can recover $D_2 x(t) := (a_{21}, a_{22}, a_{23})$ with the information from any two packets, $D_3 x(t) := (a_{31}, a_{32}, a_{33})$ with the information from any three packets, and $D_4 x(t) := a_4$ with the information from all four packets. The shaded values can be seen as the parity check information.

Given the number L and the length s of the packets, the following theorem shows possible choices of $r_1, \ldots, r_L$.

Theorem III.3 (Feasible configurations). *Given* L *and* s, *there exists a coding scheme that satisfies the successive refinement property in Def. III.1, if the length* r_ℓ *of* $D_\ell x(t), \ell = 1, \ldots, L$ *satisfies the following condition*

$$\sum_{i=1}^{L} g_i \leq sL \tag{8}$$

where g_i *is defined as*

$$g_\ell := \begin{cases} \frac{r_\ell}{\ell} L & \ell \textit{ divides } k_\ell \\ \lfloor \frac{k_\ell}{\ell} \rfloor \cdot L + L - \ell + mod(k_\ell, \ell) & \textit{otherwise} \end{cases} \tag{9}$$

Proof: A proof of the result can be found in [18]. ■

A choice $(r_1, \ldots, r_L)$ is called a feasible configuration if it satisfies the condition in (8). The above theorem shows that we can find coding schemes for feasible configurations which guarantee the successive refinement requirement. We give an example of the coding scheme in Fig. 2 to illustrate the result. In this example we assume $L = 4$ and $s = 3$. A feasible configuration with $(r_1 = 0, r_2 = 3, r_3 = 3, r_4 = 1)$ is depicted in Fig. 2, where we denote $D_1 x(t) = \emptyset, D_2 x(t) = (a_{21}, a_{22}, a_{23}), D_3 x(t) = (a_{31}, a_{32}, a_{33})$, and $D_4 x(t) = a_4$.

IV. Closed-loop system

In this section, we study the performance of the closed-loop system under the proposed coded control scheme.

Theorem IV.1 (Decay rate of the closed-loop system). *Given the discrete-time system in (1) and the packet-drop channel with parameters* L *(number of packets),* s *(size of each packet), and* p *(erasure probability), there exists an encoder and a decoder in the form of (2) and (3) such that the closed-loop system is second moment stable and has a decay rate* $1/\alpha$, *if the following conditions are satisfied:*

$$S := \begin{pmatrix} \alpha P & A^T P & (A+BK_1C_1)^T P & \dots & (A-BK_LC_L)^T P \\ PA & p_0^{-1}P & 0 & \dots & 0 \\ P(A+BK_1C_1) & 0 & p_1^{-1}P & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P(A+BK_LC_L) & 0 & 0 & \dots & p_L^{-1}P \end{pmatrix} \tag{12}$$

1) *There exist matrices D_ℓ, K_ℓ for $\ell = 1, \dots, L$, a positive-definite matrix $P \succ 0$ and a constant $0 < \alpha < 1$, such that the matrix S defined in (12) satisfies*

$$S \succ 0. \tag{10}$$

Note that C_ℓ in (12) is defined as

$$C_\ell := \begin{pmatrix} D_1 \\ \vdots \\ D_\ell \end{pmatrix} \text{ where } D_\ell \in \mathbb{R}^{r_\ell \times n},$$

and p_ℓ is given by

$$p_\ell := \binom{L}{\ell} p^{L-\ell}(1-p)^\ell \tag{11}$$

2) *The dimension $r_\ell \in \mathbb{N}_+$ of the matrices D_ℓ for $\ell = 1, \dots, L$ satisfies (8).*

As shown in the sequel, the conditions can be verified by solving an optimization problem. We first prove the above result.

A. Proof of Theorem IV.1

Recall that under our successive refinement encoding scheme, the control signal is given in (7), where the matrix C_ℓ varies for different time instances. Hence effectively we are dealing with an i.i.d. jump linear system [3] of the form

$$x(t+1) = H(t)x(t) \tag{13}$$

where $H(t)$ is a random matrix taking values in the set $\{H_1, \dots, H_L\}$ with H_ℓ defined as

$$H_\ell := A + BK_\ell C_\ell \text{ for } \ell = 0, \dots, L \tag{14}$$

with C_ℓ given by (6) and $C_0 = 0$. Furthermore, as we assume i.i.d. channels for each time instance, the random matrix $H(t)$ is an i.i.d. matrix for each time t. To study the equivalent system in (13), we use the result in [2] on stability of jump linear systems. It is shown that [2, Thm. 2.1] for a jump linear system of the form $x(t+1) = H_{\tau(t)}x(t)$, where $\tau(t)$ is an i.i.d. sequence taking values in a set $\mathcal{T} = \{0, 1, \dots, L\} \subseteq \mathbb{Z}_+$, the necessary and sufficient condition of its second moment stability (cf. Def. II.1) is that for some positive-definite matrix $P \succ 0$, it holds that

$$\sum_{\tau \in \mathcal{T}} p_\tau H_\tau^T P H_\tau - P \prec 0 \tag{15}$$

where p_τ denotes the probability distribution of τ. Furthermore, the result can be extended to show that the jump linear system has a decay rate $1/\alpha$ (cf. Def. II.2), if the following condition is satisfied

$$\sum_{\tau \in \mathcal{T}} p_\tau H_\tau^T P H_\tau - \alpha P \prec 0. \tag{16}$$

The above result can be turned into an algorithm to design the system. To this end, we follow the approach in [17] to rewrite the condition (16) as

$$S \succ 0$$

using Schur complements and equations (14) and (11), where S is given in (12). Furthermore, due to the successive refinement requirement, r_ℓ (the dimension of D_ℓ) must satisfy the condition (8). Lastly, the probability that the random matrix $H(t)$ takes value H_ℓ, for $\ell = 0, \dots, L$ is equal to the probability that $\tilde{x}(t)$ equals $C_\ell x(t)$, i.e., any ℓ out of L packets are received by the controller. Hence we have

$$\mathbb{P}\{H(t) = H_\ell\} = \binom{L}{\ell} p^{L-\ell}(1-p)^\ell = p_\ell$$

for all t. This proves the claim.

B. Controller design

As shown in Theorem IV.1, there are many degrees of freedom in the design for the jump linear system in (13), such that the conditions in (10) are satisfied. More precisely, we can design the matrices D_ℓ, for $\ell = 1, \dots, L$ (or equivalently C_ℓ) and their sizes r_ℓ, and the corresponding control matrix K_ℓ in accordance with certain control performance criteria. Notice the matrices C_ℓ are nested in the sense that C_{ℓ_1} is always a sub-matrix of C_{ℓ_2} for any $\ell_1 \le \ell_2$. As a result, the quantity $D_\ell x(t)$ is available to the controller with probability $p_\ell + \dots + p_L$, confirming the successive refinement property in which more important system states or the linear combinations thereof (smaller ℓ) is received correctly with higher probability.

In fact, instead of checking the feasibility of the condition (10) for a given α, we could design the encoder and controller to maximize the decay rate of the equivalent system (13). Specifically, a lower bound on the decay rate of the system (13) can be found by solving the optimization problem

$$\begin{aligned} \text{minimize} \quad & \alpha \\ \text{s. t.} \quad & S \succ 0 \text{ and } P \succ 0 \\ & C_\ell := \begin{pmatrix} D_1 \\ \vdots \\ D_\ell \end{pmatrix} \text{ where } D_\ell \in \mathbb{R}^{r_\ell} \end{aligned} \tag{17}$$

$$r_\ell \in \mathbb{N}_+ \text{ satisifying } (8)$$

where the matrix S is defined in (12) and the variables are $P, r_\ell, D_\ell, K_\ell, \ell = 1, \ldots, L$ and α. If the above problem is feasible with some $0 < \alpha < 1$, then the system (13) is stable with a decay rate $1/\alpha$.

Due to the integer constraint on r_ℓ, we first fix the choice of r_ℓ, and solve the optimization problem (17) for the given r_ℓ. It should be noted that in this case, the optimization problem is not convex in its variables D_ℓ, K_ℓ, P and α. Nevertheless, by using an alternating method, a (local) minimizer can be found via executing a sequence of convex optimization problems involving linear matrix inequalities (LMI), as discussed in [17]. More precisely, we solve the problem in (17) using Algorithm 1. Notice that each step of (a), (b) and (c) in Algorithm 1 amounts to solving an convex optimization problem involving LMI, and can be readily carried out using off-the-shelf convex optimization solvers.

We point out that the coding scheme and the control policy enter the optimization procedure via matrices $\mathcal{C}_\ell$ and K_ℓ through the condition $S \succ 0$ in (17). Hence the problem (17) optimizes the coding scheme and the control policy in a joint manner, and its solution automatically gives the priority of the system states, with the objective that the decay rate of the equivalent system (13) is maximized. More precisely, $D_1 x(t), D_2 x(t), \ldots$ give (linear combinations of) the system states in the decreasing order of importance, and $K_1, K_2, \ldots$ give the corresponding controllers, designed jointly by the optimization problem (17). However, we should also point out that the optimization problem (17) is non-convex, hence Algorithm 1 may provide a locally optimal solution.

Remark IV.2. *To check the feasibility of the condition in (10) for a given α, we consider an optimization problem identical to (17) except that the objective function is simply* 0*. Algorithm 1 can be used to solve this problem with the given α without any further modification. A feasible solution to the problem implies the feasibility of the condition (10).*

Remark IV.3 (Encoding and decoding complexity). *It can be seen that $v_\ell(t)$ is a linear combination of the system state $x(t)$. Hence the encoding has the same computation complexity as a matrix-vector multiplication. The decoding procedure in (5) amounts to solving a set of linear equations. Also notice that with the linear controller (7), the two steps in (5) and (7) can be combined into one matrix-vector multiplication. Furthermore, since the coding scheme is based on MDS codes, the decoding procedure could be further simplified using MDS codes with more structures (Reed-Solomon codes, for example).*

Remark IV.4 (Non-adaptive controller). *By setting the control matrices to be identical $K_1 = \ldots = K_L = K$ in the problem (17), we obtain a simpler control rule, where the controller does not adapt according to different received information. This can simplify the controller design, as well as the decoding procedure.*

Algorithm 1 Algorithm for solving problem (17)

Require: Choose a feasible configuration $r_1, \ldots, r_L$ satisfying (8). Initialize α, D_ℓ, K_ℓ and P.
repeat{
(a) Fix P, K_ℓ, solve problem (17) with variables α and $D_\ell, \ell = 1, \ldots, L$
(b) Fix P, D_ℓ, solve problem (17) with variables α and $K_\ell, \ell = 1, \ldots, L$
(c) Fix K_ℓ, D_ℓ, α, solve problem (17) with the variable P. }
until α converges.

V. Comparisons and numerical results

In this section, we give a few numerical results to illustrate the proposed coded control scheme. In particular, we will compare our results with two other static schemes. The first scheme, described in Section V-A, always constructs the full system state $x(t)$, and generates control signal based on $x(t)$. The second scheme, described in V-B, is the optimal static scheme under our assumptions. These two schemes represent two extremes in the entire design spectrum, whereas the first scheme is simple, but fails to address the coding and the control problem jointly; and the second scheme has the highest complexity, applying different control laws according to all possible channel realizations. Numerical results show that our proposed scheme strikes a good balance between these two extremes.

A. Reconstructing the full system state

Under the premise that the control signal should be generated using only the currently received information, a less sophisticated but simpler coding scheme is to let the controller reconstruct the full system state $x(t)$ at each time instance, if possible. Then a control signal is generated based on the system state $x(t)$. This can be implemented with the maximum distance separable (MDS) codes, where L' received packets allow the controller to recover $x(t)$ if it holds that $sL' \geq n$. Under this coding scheme, each packet contains s linear combinations of the n system states. If the number of the linear combinations (sL' in this case) is larger than the number of unknowns (n in this case), the unknown system state can be solved using the received packets by solving a set of linear equations. If not, the unknown state cannot be recovered and no control input will be applied for this step.

With the full system state $x(t)$, the controller could use any stabilizing control policy (for example, an LQR controller). However, we can do better by treating it as a linear jump system. By the same token, the equivalent system under this scheme is also a jump linear system of the form (13), but with only two possible states $H(t) \in \{A+BK, A\}$. We have $H(t) = A + BK$ if enough packets are received to recover $x(t)$, otherwise $H(t) = A$. It is straightforward to find that $\mathbb{P}\{H(t) = A + BK\} = \sum_{\ell=L'}^{L} \binom{L}{\ell} p^{L-\ell}(1-p)^\ell$ with the smallest L' satisfying $sL' \geq n$. Following Remark III.2, the controller does not attempt to estimate $x(t)$ if less than L' packets are received.

B. Optimal strategy

With the encoder and controller in the form of (2) and (3), respectively, we could in principle design the optimal (linear) encoder and decoder, at the cost that the complexity of this scheme grows exponentially with the number of packets L.

In the optimal scheme, the packets are generated as

$$v_\ell(t) = \tilde{D}_\ell x(t), \ell = 1, \dots, L$$

where $\tilde{D}_\ell \in \mathbb{R}^{s\times n}$ for all ℓ. Since there are 2^L possible cases with different received packets, the control/decoder will generate a corresponding control signal $u(t)$ based on the received packets. Specifically, the control signal can be rewritten as

$$u(t) = K_{\ell'} E_{\ell'} \begin{pmatrix} \tilde{D}_1 \\ \vdots \\ \tilde{D}_L \end{pmatrix} x(t)$$

for $\ell' = 1, \dots, 2^L$ where $E_{\ell'}$ denotes a selection matrix which keeps the entries of $\tilde{D}_\ell x(t)$ if the corresponding packets are received.

Using the same argument as in Section IV, this strategy results in an equivalent jump linear system of the form $x(t+1) = \tilde{H}(t)x(t)$ where $\tilde{H}(t)$ takes value $\tilde{H}_{\ell'}$ for some $\ell' = 1, \dots, 2^L$ defined as

$$\tilde{H}_{\ell'} = A + BK_{\ell'} E_{\ell'} \begin{pmatrix} \tilde{D}_1 \\ \vdots \\ \tilde{D}_L \end{pmatrix} \tag{18}$$

where the probability is given by

$$\mathbb{P}\left\{\tilde{H}(t) = \tilde{H}_{\ell'}\right\} = \binom{L}{b} p^{L-b}(1-p)^b$$

where b denotes the number of received packets in this time instance. Using the same machinery in Section IV-B, the matrices D'_ℓ, K'_ℓ can be optimized using Algorithm 1 to maximize the decay rate of the system. The main drawback of the optimal scheme is that since there are 2^L different states (compared to L different states in the proposed scheme), the complexity of the scheme grows exponentially with the number of packets L, rendering it inapplicable for a real-time control system.

C. Comparison

Although the scheme in Section V-A is very simple, its performance is often inferior to the proposed scheme, because the designs of the communication and the control scheme are separated. Indeed, aiming to recover the full system state imposes a very stringent requirement on the channel quality, and is often impossible when the erasure probability of the channel is high. Moreover, it is easy to see that the scheme in Section V-A is a special case of our scheme.

It can be argued straightforwardly that our proposed scheme in Section III can be viewed as an instance of the optimal strategy proposed in Section V-B with certain choices of $K_{\ell'}$ and $\tilde{D}_{\ell'}$. However, using the optimal strategy has the following two difficulties. Firstly, the number of the possible states of the induced jump system grows exponentially with the number of packets L. Therefore, after the reception of packets, the controller needs to search for the corresponding control matrix $K_{\ell'}$ from a look-up table whose size grows exponentially with L, which is a bottleneck for real-time implementations. Secondly, even equipped with Algorithm 1, the problem of finding a good controller for the optimal strategy quickly becomes intractable for even a moderate-sized L. This is because the size of the LMI constraint, which guarantees the stability condition in (15), also grows exponentially with the number of packets L. In contrast, the number of the possible states of the jump system induced by our proposed scheme, as well as the size of the LMI constraint in (12), grows only linearly in L, for a fixed choice of r_ℓ. We point out that finding a good choice of r_ℓ is an interesting question of its own, and will be studied in the future work.

D. Example

To illustrate the results, we consider a linear system of the form (1) with the dynamical matrix $A \in \mathbb{R}^6$. The eigenvalues of A is given as

$$\text{eig}(A) := [7, 2, 1.8, 1.5, 1.2, 1.1].$$

The matrix B is a 6×6 matrix where the pair (A, B) is controllable. We assume at each time instance, $L = 3$ packets are transmitted where each packet contains a vector $v_\ell(t) \in \mathbb{R}^s$ with $s = 3$. The erasure probability is set to be $p = 0.25$. Furthermore, an i.i.d. Gaussian noise sequence with variance 0.5 is added to the system state.

We use Algorithm 1 to find encoding matrices D_ℓ and controllers K_ℓ to stabilize the system, based on three difference choices of configurations:

$$\begin{aligned}\mathbf{r}^{(1)} &= [r_1 = 1, r_2 = 2, r_3 = 3] \\ \mathbf{r}^{(2)} &= [r_1 = 2, r_2 = 0, r_3 = 3] \\ \mathbf{r}^{(3)} &= [r_1 = 0, r_2 = 6, r_3 = 0]\end{aligned}$$

It can be checked using (8) that all three configurations are feasible. In particular, the choice $\mathbf{r}^{(3)}$ corresponds to the scheme described in Section V-A, where the full system state $x(t) \in \mathbb{R}^6$ is recovered when any two packets out of three are received. As a comparison for this example, we also compute the optimal strategy as discussed in Section V-B, whose decay rate is denoted as $1/\alpha^*$.

Algorithm 1 produces following α for the three configurations and the optimal strategy

$$\begin{aligned}\alpha^{(1)} &= 0.7656 \\ \alpha^{(2)} &= 0.8325 \\ \alpha^{(3)} &= 7.6563 \\ \alpha^* &= 0.7656\end{aligned}$$

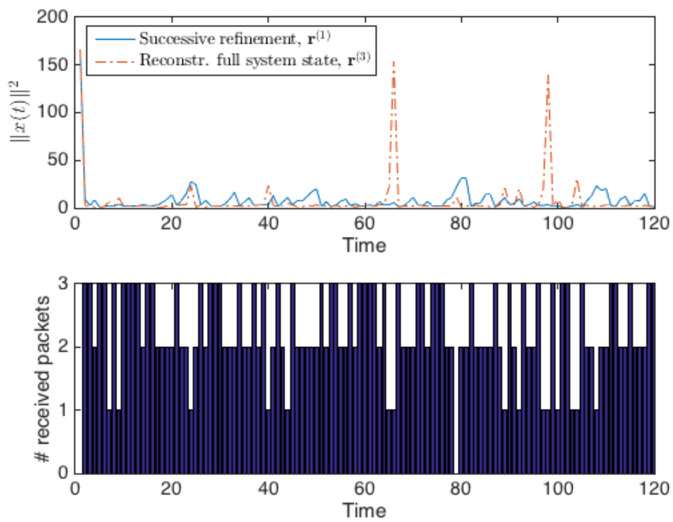

Fig. 3: A sample path of the networked control system. The plot on the top shows a sample path of the system $||x(t)||^2$ with two different control policies with $r^{(1)}$ and $r^{(3)}$, respectively.

This shows that the first two configurations provide a stabilizing control strategy with $\alpha^{(1)}, \alpha^{(2)} < 1$, and the third configuration fails to find coding matrices D_ℓ and controllers K_ℓ which stabilizes the system[1].

Finally, for the optimal strategy in Section V-B, Algorithm 1 produces $\alpha^* = 0.7656$, which is the same as $\alpha^{(1)}$. This shows that our proposed scheme with the choice of $\mathbf{r}^{(1)}$ gives the same decay rate (up to four digits after the decimal point) as the optimal strategy, but with a simpler implementation. Indeed, every time it applies one of 3 possible controllers, instead of 8 possible controllers as in the optimal strategy. This advantage is more pronounced when the number of packets L is large.

A closed-loop simulation for the above example is given in Figure 3. The bottom plot shows the number of received packets in each time instance, ranging from 0 to 3. The plot on the top shows a sample path of the system under two different control policies with $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(3)}$, respectively. The solid line shows the norm of the system state $||x(t)||^2$ using the control policy with $\mathbf{r}^{(1)}$, and the dashed line represents the control policy $\mathbf{r}^{(3)}$, with which the full system state is to be reconstructed with two received packets.

VI. Future Work

We conclue the paper with two interesting questions for the future work.

- Choosing the optimal r_ℓ is a combinatorial optimization problem. Find good choices of r_ℓ with a low computational method (instead of an exhaustive search) would be interesting to our approach.
- The current work uses a static encoder and decoder/controller, which do not attempt to estimate the system state using past information. It has benefits in terms of simplicity, but is in general inferior to a controller which utilizes all available information. One direction of the future work is to study a dynamic estimator/controller, and compare the results with the static ones.

[1] We point out that due to the non-convexity of the problem (17), failing to find a $\alpha^{(3)}$ smaller than 1 using Algorithm 1 does not eliminate its existence. However for this particular example, many rounds of simulations with different initializations always give the same result.

References

[1] G. I. Bara and M. Boutayeb. Static output feedback stabilization with H infin; performance for linear discrete-time systems. *IEEE Transactions on Automatic Control*, 50(2):250–254, February 2005.

[2] Yuguang Fang and K. A. Loparo. Stochastic stability of jump linear systems. *IEEE Trans. on Automatic Control*, 47(7), July 2002.

[3] Yuguang Fang, KENNETH A. Loparo, and Xiangbo Feng. Almost sure and Î´moment stability of jump linear systems. *International Journal of Control*, 59(5):1281–1307, 1994.

[4] V. Gupta, N. C. Martins, and J. S. Baras. Optimal Output Feedback Control Using Two Remote Sensors Over Erasure Channels. *IEEE Transactions on Automatic Control*, 54(7):1463–1476, July 2009.

[5] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu. A Survey of Recent Results in Networked Control Systems. *Proceedings of the IEEE*, 95(1):138–162, January 2007.

[6] Zhipu Jin, V. Gupta, B. Hassibi, and R. M. Murray. State Estimation Utilizing Multiple Description Coding over Lossy Networks. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 872–878, December 2005.

[7] A. R. Mesquita, J. P. Hespanha, and G. Nair. Redundant data transmission in control/estimation over wireless networks. In *2009 American Control Conference*, pages 3378–3383, June 2009.

[8] G. N. Nair, F. Fagnani, S. Zampieri, and R. J. Evans. Feedback Control Under Data Rate Constraints: An Overview. *Proceedings of the IEEE*, 95(1):108–137, January 2007.

[9] R. Puri and K. Ramchandran. Multiple description source coding using forward error correction codes. In *Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers (Cat. No.CH37020)*, volume 1, pages 342–346 vol.1, October 1999.

[10] Anant Sahai. *Anytime information theory*. Thesis, Massachusetts Institute of Technology, 2001.

[11] A. Shirazinia, L. Bao, and M. Skoglund. Sufficient conditions for stabilization in feedback control over noisy channels using anytime rateless codes. In *2012 American Control Conference (ACC)*, pages 1254–1259, June 2012.

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

[13] T. Tanaka, K. H. Johansson, T. Oechtering, H. Sandberg, and M. Skoglund. Rate of prefix-free codes in LQG control systems. In *2016 IEEE International Symposium on Information Theory (ISIT)*, July 2016.

[14] A. Tarable, A. Nordio, F. Dabbene, and R. Tempo. Anytime reliable LDPC convolutional codes for networked control over wireless channel. In *2013 IEEE International Symposium on Information Theory*, pages 2064–2068, July 2013.

[15] David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, May 2005.

[16] Wing Shing Wong and R. W. Brockett. Systems with finite communication bandwidth constraints. I. State estimation problems. *IEEE Transactions on Automatic Control*, 42(9):1294–1299, September 1997.

[17] Lin Xiao, Arash Hassibi, and Jonathan P. How. Control with random communication delays via a discrete-time jump system approach. In *American Control Conference, 2000. Proceedings of the 2000*, volume 3, pages 2199–2204, 2000.

[18] J. Zhu, Y. Pu, V. Gupta, C. Tomlin, and K. Ramchandran. A sequential approximation framework for coded distributed optimization. In *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1240–1247, October 2017.