Multiple Descriptions for Packetized Predictive Control over Erasure Channels

Jan Østergaard, *Member, IEEE,* and Daniel E. Quevedo, *Member, IEEE*

***Abstract*— We consider a networked control system with random delays and erasures on a data-rate limited forward channel between the controller and the plant. The feedback channel from the plant to the controller is assumed noiseless. We combine two techniques to enhance the reliability of the system. First, we use packetized predictive control, where a quantized control vector with future predicted control signals is transmitted to the plant at each time instant. Second, we utilize multiple descriptions to further aid in the robustness towards packet erasures. In particular, we transmit M redundant packets, which are constructed such that when receiving any $1 \leq J \leq M$ packets, the current control signal as well as $J-1$ future control signals can be reliably reconstructed at the plant side. For the particular case of LTI systems and when the packets are not received out-of-order, we prove stability by showing that the system can be cast as a Markov jump linear system with $M+1$ states. We further show by simulations that, by use of multiple descriptions, the system state variance can be significantly reduced without increasing the total data rate.**

I. Introduction

Networked control systems (NCSs) generalize the notion of control systems by allowing the control loops to be closed through networks. While NCSs open the door for new control applications e.g., remote and distributed control, it also brings forth a wealth of new challenges that go beyond conventional control techniques. Specifically, networks are generally unreliable, e.g., they could be data-rate or SNR limited and be suscept to random bit-errors or data-dropouts due to fading on wireless channels or router congestions on packet-switched networks. Moreover, the network could introduce (random) delays or even break down, which would leave the plant in open loop. Several issues of NCSs are discussed in detail in the survey papers [1]–[3] see also [4], [5].

In this paper, we are interested in quantized packetized predictive control (PPC) over data-rate limited (digital) channels [6]–[9]. In PPC, a control vector with the current and $N-1$ future predicted plant inputs are constructed at the controller side to compensate for random delays and packet dropouts in the forward channel (i.e., the channel between the controller and the plant). We will assume that the feedback channel (i.e., the channel between the plant and the controller) is noiseless. The main contribution of the paper is to show that PPC can be combined with multiple descriptions (MDs) and thereby increase the reliability of the system in the case of channel failures (i.e., packet delays and dropouts). MDs is a joint source-channel coding technique, which has received increasingly attention in the information theoretic and signal processing communities [10]. The main idea of MDs is to encode the source signal (e.g., control vector) into multiple descriptions (packets), which individually provide a desired approximation of the control vector. Moreover, when the descriptions are combined, they refine each other and, thus, jointly provide a better approximation of the control vector. In this work, we construct M partially redundant descriptions (or packets) based on each control vector. The packets are constructed so that the current control signal and $J-1$ future control signals can be obtained by combining any subset of $J \in \{1, \dots, M\}$ packets. Thus, the more packets are received at the plant, the more future plant predictions becomes available. Notice that on reception of at least one packet out of the M packets, the current control signal can be completely recovered at the plant input side.

While our idea is applicable to general PPC systems, we will focus on the case of discrete-time noisy LTI systems and i.i.d. packet dropouts. We first describe the control law when packets are received out-of-order. In this case, however, we are not able to assess the stability of the system. Then we consider the case where packets are received in order, i.e., out-of-order packets are discarded. In this case, due to the MDs, the control values may, at times, be applied out-of-order. We show that the system can be cast as an $M+1$ state Markov jump linear system (MJLS). Thus, the stability can addressed via linear matrix inequalities. Through a simulation study, we then address the issue of bit-rates due to quantization, and show that for a fixed data rate, the use of MDs significantly increases the system reliability in the presence of i.i.d. packet dropouts.

II. Packetized Control over Erasure Channels

In this section, we provide a summary of existing results on quantized PPC and relate them to the present situation.

A. System Model

We consider the following discrete-time stochastic linear time invariant (LTI) possibly unstable dynamical plant model with state $x_t \in \mathbf{R}^z$, $z \geq 1$ and scalar input $u_t \in \mathbf{R}$:

$$x_{t+1} = Ax_t + B_1 u_t + B_2 w_t, \quad t \in \mathbb{N}. \tag{1}$$

Jan Østergaard is with the Department of Electronic Systems, Aalborg University, Denmark; e-mail: janoe@ieee.org.

Daniel Quevedo is with the School of Electrical Engineering & Computer Science, The University of Newcastle, NSW 2308, Australia; e-mail: dquevedo@ieee.org.

This research was partially supported by the Danish Research Council for Technology and Production Sciences, grant no. 274-07-0383 and partially supported under Australian Research Council's Discovery Projects funding scheme (project number DP0988601).

In (1), $w_t \in \mathbf{R}^{z'}$, $z' \geq 1$, is an unmeasured disturbance, modelled as an arbitrarily distributed (and with possibly unbounded support) zero-mean stochastic process with bounded variance. Throughout this work, we will assume that the pair (A, B_1) is stabilizable. The initial state x_0 is arbitrarily distributed with bounded variance.

B. Cost Function

At each time instant t and for a given plant state x_t, the following cost function is minimized:

$$J(\bar{u}\,', x_t) \triangleq \|x'_N\|_P^2 + \sum_{\ell=0}^{N-1} \left(\|x'_\ell\|_Q^2 + \lambda(u'_\ell)^2\right), \qquad (2)$$

where $N \geq 1$ is the horizon length. The design variables $P \succeq 0$, $Q \succeq 0$ and $\lambda > 0$ allow one to trade-off control performance versus control effort.

The cost function in (2) examines a prediction of the plant model over a finite horizon of length N. The predicted state trajectories at time t are independent of the buffer contents at the decoder (i.e., they are independent of what has been received at the plant input side), the dropout probabilities and the external disturbances w, and are generated by $x'_{\ell+1} = Ax'_\ell + B_1 u'_\ell$, $x'_0 = x_t$, whilst the entries in $\bar{u}\,' = \begin{bmatrix} u'_0 & \dots & u'_{N-1} \end{bmatrix}^T$ represent the associated predicted plant inputs. Thus, the current control vector $\bar{u}_t = [u_t(1), \dots, u_t(N)]^T$ contains the control signal $u_t(1)$ for the current time instant t as well as $N-1$ future predictive control signals for time up to $t+N-1$.

Following the ideas underlying PPCs, see, e.g., [6], at each time instant t, and for current state x_t, the controller sends the *entire* optimizing sequence, $\bar{u}_t$, to the actuator node. Depending upon future packet dropout scenarios, a subsequence of $\bar{u}_t$ will be applied at the plant input, or not. Following the receding horizon paradigm, at the next time instant, x_{t+1} is used to carry out another optimization, yielding $\bar{u}_{t+1}$, etc.

C. Network Effects

We consider an unreliable packet-switched network, which due to transmission errors and congestion, introduces packet dropouts and packet delays. We assume the availability of M separate and independent channels. We model transmission effects via the discrete processes $\{d^i_{t,t'}\}_{t'=0}^{\infty}$, where $0 \leq t \leq t'$ and $i = 1, \dots, M$, defined via:

$$d^i_{t,t'} \triangleq \begin{cases} 1, & \text{if packet } c^i_t \text{ is in the buffer at time } t' \geq t, \\ 0, & \text{else.} \end{cases}$$

These processes are generally not i.i.d., since if a packet has been received at time instant t' it is still received at time instances $t'+n, n \geq 1$. However, for $t' = t$, the outcomes $d^i_{t,t}, i = 1, \dots, M, t \geq 0$, are considered mutually independent. In other words, we assume that the packet dropout probabilities are independent across channels as well as across time.

D. Quantization Constraints

We consider a bit-rate limited digital network between controller output and plant input and all data to be transmitted needs therefore to be quantized. This introduces a quantization constraint into the problem of minimizing $J(\bar{u}\,', x_t)$. A closed form solution to this problem was derived in [11]. Furthermore, in [8] the problem was cast into the framework of entropy-constrained (subtractively) dithered (lattice) quantization (ECDQ) [12].

For future reference, we present below some important results of [8], [11] that we will be needing in the sequel:

Let $\bar{Q} \triangleq \mathrm{diag}(Q, \dots, Q, P)$ and let Φ, Υ be defined by

$$\Phi \triangleq \begin{bmatrix} B_1 & 0 & \dots & 0 \\ AB_1 & B_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B_1 & A^{N-2}B_1 & \dots & B_1 \end{bmatrix}, \Upsilon \triangleq \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}. \quad (3)$$

Theorem 2.1 (Quantized Predictive Control [11]): Consider any quantized set $\mathcal{U} \subset \mathbb{R}^{\mathcal{N}}$, the matrices $\overline{Q}$, Φ, and Υ given in (3), and define:

$$\xi_t \triangleq \Gamma x_t, \quad \Gamma \triangleq -\Psi^{-T}\Phi^T\overline{Q}\Upsilon, \qquad (4)$$

where $\Psi \in \mathbb{R}^{N \times N}$ is obtained from the factorization $\Psi^T\Psi = \Phi^T\overline{Q}\Phi + \lambda I$, where λ is as in (2).

Then the constrained optimizer $\tilde{u}_t = \arg\min_{\bar{u}\,' \in \mathcal{U}} J(\bar{u}\,', x_t)$, see (2), satisfies:

$$\tilde{u}_t = \Psi^{-1}\mathcal{Q}(\xi_t), \qquad (5)$$

where $\mathcal{Q}(\cdot)$ is a (nearest neighbour) vector quantizer with alphabet $\Psi\mathcal{U}$. □

When using ECDQs, a dither vector ζ_t is added to the input prior to quantization and then subtracted again at the decoder to obtain the reconstruction.[1] Specifically, let $\mathcal{Q}_\Lambda$ denote an ECDQ with underlying lattice Λ. Then the discrete output ξ'_t of the ECDQ is given by $\xi'_t = \mathcal{Q}_\Lambda(\xi_t + \zeta_t)$. Furthermore, the reconstruction $\hat{\xi}_t$ at the decoder is then obtained by subtracting the dither, i.e., $\hat{\xi}_t = \xi'_t - \zeta_t$. Interestingly, this quantization operation may be exactly modelled by an additive noise channel, i.e., $\hat{\xi}_t = \xi_t + n_t$, where the noise n_t is independent of ξ_t, see [12] for details. Using this, it was furthermore shown in [8] that the quantized (and reconstructed) control variable $\vec{u}_t$ can be written as

$$\vec{u}_t = \Psi^{-1}(n_t + \xi_t), \qquad (6)$$

where n_t and ξ_t are mutually independent and $\xi_t = \Gamma x_t$. Throughout the paper, we will use $\vec{u}_t$ to denote the quantized (and reconstructed) control vector, which has been found by using an ECDQ on ξ_t. Note that $\vec{u}_t$ is a continuous variable whereas $\tilde{u}_t = \Psi^{-1}\xi'_t$ is the corresponding discrete variable, which is entropy coded and thereby converted into a bitstream (to be transmitted over the network).

[1] It follows that we require the dither sequence to be known both at the encoder and at the decoder.

III. Multiple Descriptions

To aid in robustness towards packet loss we will be using multiple descriptions (MDs). In particular, we will construct M packets $\bar{c}_t^i, i = 1, \ldots, M$, at time t based on the single control vector $\tilde{u}_t$. These M packets, are created in such a way that they are individually useful for the plant. Moreover, the more of the M packets that are received, the better information about future predictive control signals will become available to the plant.

We will be using MD based on forward error correction (FEC) codes [13]. Specifically, we use (n,k)-erasure codes, i.e., error correction codes, which as input take k symbols $\bar{y}_t = (y_t(1), \ldots, y_t(k))$ and output n symbols $\bar{\phi}_t = (\phi_t(1), \ldots, \phi_t(n))$, where $n \geq k$. With an (n,k)-erasure code, the original k input symbols can be completely recovered using any subset of at least k output symbols. For example, a $(3,2)$-erasure code may be constructed by letting $\phi_t(1) = y_t(1), \phi_t(2) = y_t(2)$, and $\phi_t(3) = y_t(1)$ XOR $y_t(2)$. Thus, using any two $\phi_t(i), \phi_t(j), i \neq j$ both $y_t(1)$ and $y_t(2)$ may be perfectly recovered.

For the NCS studied, we apply a sequence of erasure codes on the control vector $\tilde{u}_t$ in order to obtain M packets. First, we recall that $\tilde{u}_t$ contains quantized elements, which belong to a discrete alphabet, i.e., each symbol $u_t(i)$ can be identified with a bit string $b_t(i)$ of L_i bits. Let us for the moment assume that $L_i = L = 24$ bits and that we wish to construct $M = 3$ packets $\bar{c}_t^i, i = 1, \ldots, 3$. Then, we create the first packet $\bar{c}_t^1$ in the following way: Let $c_t^1(1) = b_t(1)$. At this point we split $b_t(2)$ into two bit strings $b_t^{(1)}(2)$ and $b_t^{(2)}(2)$ of equal size (each containing $L/2 = 12$ bits) so that $b_t(2) = (b_t^{(1)}(2), b_t^{(2)}(2))$. Then we let $c_t^1(2) = b_t^{(1)}(2)$. Finally, we split $b_t(3)$ into three equal lenght bitstreams $b_t^{(1)}(3), b_t^{(2)}(3)$, and $b_t^{(3)}(3)$ and let $c_t^1(3) = b_t^{(1)}(3)$. Thus, the first packet is given by $\bar{c}_t^1 = (b_t(1), b_t^{(1)}(2), b_t^{(1)}(3))$. Similarly, we construct the second packet as $\bar{c}_t^2 = (b_t(1), b_t^{(2)}(2), b_t^{(2)}(3))$. Finally, we construct the third packet as $\bar{c}_t^3 = (b_t(1), b_t^{(1)}(2)$ XOR $b_t^{(2)}(2), b_t^{(3)}(3))$.

The $M = N = 3$ packets are transmitted over separate channels.[2] It may be noticed that upon reception of any single packet, the first control signal $u_t(1)$ for the current time t may be recoverd. Upon reception of any two packets, the first control signal $u_t(1)$ as well as the future predicted control signal $u_t(2)$ for time $t+1$ may be recovered. Finally, upon reception of all three packets, all N control signals $\tilde{u}_t = (u_t(1), u_t(2), u_t(3))$ are recovered.

The size of a packet (i.e., the packet bit-rate R_i) is on average $R_i = L_1 + L_2/2 + L_3/3 + \cdots + L_M/M$ bits and there are M packets. When not using MDs, the coding rate is $R = L_1 + \cdots + L_M$. Thus, the total coding overhead compared to when not using multiple descriptions is $M(L_1 + L_2/2 + L_3/3 + \cdots + L_M/M) - (L_1 + \cdots + L_M)$. We refer the reader to the simulation study for further details on the operational coding rates.

$\hat{u}_t$	$\bar{c}_t^1$	$\bar{c}_t^2$	$\bar{c}_t^3$	$\bar{c}_{t-1}^1$	$\bar{c}_{t-1}^2$	$\bar{c}_{t-1}^3$	$\bar{c}_{t-2}^1$	$\bar{c}_{t-2}^2$	$\bar{c}_{t-2}^3$
$u_t(1)$	1	x	x	x	x	x	x	x	x
$u_t(1)$	x	1	x	x	x	x	x	x	x
$u_t(1)$	x	x	1	x	x	x	x	x	x
$u_{t-1}(2)$	0	0	0	1	1	x	x	x	x
$u_{t-1}(2)$	0	0	0	1	x	1	x	x	x
$u_{t-1}(2)$	0	0	0	x	1	1	x	x	x
$u_{t-2}(3)$	0	0	0	x	0	0	1	1	1
$u_{t-2}(3)$	0	0	0	0	x	0	1	1	1
$u_{t-2}(3)$	0	0	0	0	0	x	1	1	1

TABLE I: Control value $\hat{u}_t$ at time t from available buffer contents. "1" indicates that the packet is in the buffer and "0" indicates that it is not. "X" indicates that the control value does not depend on the given packet. In all other cases, we set $\hat{u}_t = 0$.

IV. System Analysis with Time-Delays and Dropouts

We will in this section consider the general case where the network introduces time-delays and therefore packets may be received out-of-order.

A. Buffering and Reconstruction of Control Signals

At time t, the buffer at the plant input side contains all received packets, which are not older than $t - N + 1$. These will be used for obtaining the current control signal $\hat{u}_t$ giving preference to newer data. For example, assume the buffer is initially empty. Then, for the case of $M = N = 3$, if we at time t receive $\bar{c}_t^2$, then clearly we obtain $\hat{u}_t = u_t(1)$. If we then at time $t+1$ receive $\bar{c}_{t+1}^1$ and the delayed packet $\bar{c}_t^3$ then we should form $\hat{u}_{t+1} = u_{t+1}(1)$ from $\bar{c}_{t+1}^1$ and, thus, simply ignore $\bar{c}_t^3$. However, if we now at time $t+2$, only receive the very late $\bar{c}_t^1$, then we recover $\hat{u}_{t+2} = u_t(3)$. Thus, we use the older packets to obtain the control signal. This process is clarified in Table I for $M = N = 3$ and given for general $M = N \geq 1$ below:[3]

Let $\Theta_{t,t'}^{k,M} \in \{0,1\}$ be an indicator function, which is "1" if at least k out of M packets of time stamp t are in the buffer at time t' and "0" otherwise. In particular,

$$\begin{aligned}
\Theta_{t,t}^{1,M} &= \prod_{i=1}^{M} d_{t,t}^i + \sum_{j=1}^{M} (1 - d_{t,t}^j) \prod_{i=1, i\neq j}^{M} d_{t,t}^i \\
&\quad + \sum_{j=1}^{M-1} \sum_{l=j+1}^{M} (1 - d_{t,t}^j)(1 - d_{t,t}^l) \prod_{i=1, i\neq j,l}^{M} d_{t,t}^i \\
&\quad + \cdots + \sum_{j=1}^{M} d_{t,t}^j \prod_{i=1, i\neq j}^{M} (1 - d_{t,t}^i) \\
\Theta_{t-1,t}^{2,M} &= \prod_{i=1}^{M} d_{t-1,t}^i + \sum_{j=1}^{M} (1 - d_{t-1,t}^j) \prod_{i=1, i\neq j}^{M} d_{t-1,t}^i \\
&\quad + \sum_{j=1}^{M-1} \sum_{l=j+1}^{M} (1 - d_{t-1,t}^j)(1 - d_{t-1,t}^l) \prod_{i=1, i\neq j,l}^{M} d_{t-1,t}^i \\
&\quad + \cdots + \sum_{j=1}^{M-1} \sum_{l=j+1}^{M} d_{t-1,t}^j d_{t-1,t}^l \prod_{o=1, i\neq j,l}^{M} d_{t-1,t}^o \\
&\vdots \\
\Theta_{t-N+1,t}^{M,M} &= \prod_{i=1}^{M} d_{t-N+1,t}^i .
\end{aligned}$$

[2] The procedure may be extended in a trivial manner to any $M \leq N$.

[3] The more general case where $1 \leq M \leq N$ is straight-forward.

With this, the control signal $\hat{u}_t$ to be used at time t is given by:

$$\begin{aligned}\hat{u}_t = {} & u_t(1)\Theta^{1,M}_{t,t} + (1-\Theta^{1,M}_{t,t})[\Theta^{2,M}_{t-1,t}u_{t-1}(2) + (1-\Theta^{2,M}_{t-1,t}) \\ & \times [\Theta^{3,M}_{t-3,t}u_{t-2,t}(3) + \cdots \\ & + (1-\Theta^{M-1,M}_{t-N+2,t})\Theta^{M,M}_{t-N+1,t}u_{t-N+1}(N)]\cdots], \end{aligned} \tag{7}$$

where it follows that $\hat{u}_t = 0$ in the event that $\Theta^{1,M}_{t,t} = \Theta^{2,M}_{t-1,t} = \cdots = \Theta^{M,M}_{t-N+1,t} = 0$, which is the case if no packets have arrived in $N = M$ consecutive time instances.

B. State Evolution with Packets Out-of-Order

Let Ξ_t be the augmented state variable given by

$$\Xi_t \triangleq \begin{bmatrix} x_t \\ \hat{u}_t \end{bmatrix}, \tag{8}$$

then

$$\begin{aligned}\Xi_{t+1} = {} & \begin{bmatrix} A & B_1e_1^T \\ 0 & 0 \end{bmatrix}\Xi_t + \begin{bmatrix} 0 \\ \Theta^{1,M}_{t,t} \\ (1-\Theta^{1,M}_{t,t})\Theta^{2,M}_{t-1,t} \\ (1-\Theta^{1,M}_{t,t})(1-\Theta^{2,M}_{t-1,t})\Theta^{3,M}_{t-3,t} \\ \vdots \\ \Theta^{M,M}_{t-N+1,t}\prod_{i=1}^{M-1}(1-\Theta^{i,M}_{t-i,t}) \end{bmatrix}^T \\ & \times \begin{bmatrix} 0 \\ u_t(1) \\ u_{t-1}(2) \\ u_{t-2}(3) \\ \vdots \\ u_{t-N+1}(N) \end{bmatrix} + \begin{bmatrix} B_2 \\ 0 \end{bmatrix} w_t. \end{aligned} \tag{9}$$

However, since $\vec{u}_t = \Psi^{-1}\Gamma x_t + \Psi^{-1}n_t$ from (6), it follows that Ξ_{t+1} in (9) depends linearly upon N past state values i.e., $\Xi_t = \varphi_1\Xi_{t-1} + \varphi_2\Xi_{t-2} + \varphi_3\Xi_{t-3} + \cdots$. In other words, due to the fact that packets can be out-of-order, the system does not immediately admit a first-order MJLS-structure. Conventional MJLS stability results do therefore not apply. We suspect that the stability of this system with out-of-order packets could be assessed with the stability techniques introduced in [14] and [8]. However, such an analysis is left for future works.

V. System Analysis without Time-Delays

In this section, we consider the case where the network does not introduce delays. Thus, packets are not received out-of-order, i.e., at given time t, we receive only a subset of the *current* M packets.

Remark 1: We would like to emphasize that one is still allowed to apply out-of-order control signals. To see this, assume that we at time $t-2$ receive three packets and apply $u_{t-2}(1)$. Then at time $t-1$ we receive only one packet and apply $u_{t-1}(1)$. If then at time t, we do not receive any packets, then we apply the older $u_{t-2}(3)$ control signal.

A. State Evolution with Packets in Order

Let $\bar{f}_k$ denote the buffer vector at the plant input side holding past control values and let S be the one-step shift-forward matrix. Then

$$\bar{f}_k = \bar{\Theta}^M_k\vec{u}_k + (I - \bar{\Theta}^M_k)S\bar{f}_{k-1}, \tag{10}$$

where $\bar{\Theta}^M_k$ is a diagonal matrix given by

$$\bar{\Theta}^M_k = \begin{bmatrix} \Theta^{1,M}_{k,k} & 0 & \cdots & \\ 0 & \Theta^{2,M}_{k,k} & 0 & \cdots \\ \vdots & & \ddots & \\ 0 & \cdots & 0 & \Theta^{M,M}_{k,k} \end{bmatrix}. \tag{11}$$

Let Ξ_t be the augmented state variable given by

$$\Xi_t \triangleq \begin{bmatrix} x_t \\ \bar{f}_{t-1} \end{bmatrix}. \tag{12}$$

Then

$$\Xi_{t+1} = \begin{bmatrix} A & B_1e_1^T \\ \mathbf{0} & (I-\bar{\Theta}^M_k)S \end{bmatrix}\Xi_t + \begin{bmatrix} 0 \\ \bar{\Theta}^M_k \end{bmatrix}^T \vec{u}_k + \begin{bmatrix} B_2 \\ \mathbf{0} \end{bmatrix} w_t. \tag{13}$$

B. Markov Jump Linear System

We will now show that (13) can be rewritten in a form, which satisfies the conditions of an MJLS with $M+1$ states. Let

$$\bar{A}(i_t) \triangleq \begin{bmatrix} A + B_1e_1^TI_{i_t}\Psi^{-1}\Gamma & B_1e_1^T(I - I_{i_t})S \\ I_{i_t}\Psi^{-1}\Gamma & (I - I_{i_t})S \end{bmatrix}, \tag{14}$$

where $i_t \in \{0,\dots,M\}, \forall t \geq 0, e_1^T$ is the all-ones vector, and where I_{i_t} is a time-varying diagonal $M \times M$ matrix with ones on the first i_t entries on the diagonal and zero elsewhere, i.e., $I_{i_t} = diag([1,\dots,1,0,\dots,0]), e_1^TI_{i_t}e_1 = i_t$. Moreover, let

$$\bar{B}(i_t) \triangleq \begin{bmatrix} B_2 & B_1e_1^TI_{i_t}\Psi^{-1} \\ 0 & I_{i_t}\Psi^{-1} \end{bmatrix}. \tag{15}$$

Then it is easy to show that

$$\Xi_{t+1} = \bar{A}(i_t)\Xi_t + \bar{B}(i_t)\nu_t, \tag{16}$$

where $\nu_t = [w_t, n_t]^T$.

Notice that the sequence $\{i_t\}$ is i.i.d., since $i_t \in \{0,\dots,M\}$ corresponds to the number of received packets at time t and the dropouts are independent across channels and over time. Thus, the probability $p_{i_t|i'_t}$ of entering state i_t when the previous state was i'_t is independent of i'_t and given by $p_{i_t|i'_t} = p_{i_t}$, where $p_{i_t} = \text{Prob}\{\text{receiving } i \text{ packets at time } t\}$. Moreover, $p_{i_t} = p_i$ since p_{i_t} does not depend on t. If $i_t = k, \forall t$ for some fixed $k \geq 0$, then the system (16) is clearly linear. On the other hand, since the dropouts are i.i.d., it follows that the state sequence $\{i_t\}$ is ergodic and the process $\{\Xi_{t+1}, \bar{A}(i_t)\}$ is first-order Markovian. Thus, (16) is an MJLS with $M+1$ states [15]. From an analysis point of view, this is very convenient since it allows one to assess the stability of an MJLS using e.g., linear matrix inequalities. In particular, we include the following well-known results:

Corollary 5.1 (Sufficient Condition for MSS): The system (16) is MSS if there exists $\Gamma \succ 0$ such that $\Gamma - \sum_{i=0}^{M} p_i \bar{A}^T(i)\Gamma\bar{A}(i) \succ 0$.

Proof: Follows immediately from [15, Corollary 3.26] since the state sequence $\{i_t\}$ is i.i.d., and the number of jump-states is finite. ■

Corollary 5.2 (Necessary Condition for Stability): A necessary condition for MSS is that $\max\{|\text{eigs}(\bar{A}(i))|\} < \sqrt{1/p_i}$, for all $i = 0, \dots, M$.

Proof: Follows immediately from [16]. ■

VI. Simulation Study

In the following simulations we let $B_1 = B_2 = [1,1]^T$, and $Q = P = diag([1,1])$. Moreover, we let $\lambda = 1, N = 3, w \sim \mathcal{N}(0,1)$, and

$$A = \begin{bmatrix} 4.1208 & 1.1124 \\ -1.5219 & 0.2102 \end{bmatrix}, \qquad (17)$$

which has eigenvalues $\{3.6250, 0.7060\}$.

A. Quantization

We use a (non-dithered) scalar uniform quantizer with step-size $\Delta > 0$. We quantize the N-dimensional vector $\xi_t = \Gamma x_t$ independently along each dimension to obtain $\xi'_t = (\lfloor \xi_t(1) \rceil_\Delta, \lfloor \xi_t(2) \rceil_\Delta, \lfloor \xi_t(3) \rceil_\Delta)$, where $\lfloor \cdot \rceil_\Delta$ denotes rounding towards nearest $i\Delta, i \in \mathbb{Z}$. After quantizing ξ_t to obtain the vector ξ'_t we further apply Ψ^{-1} to obtain the quantized control vector $\tilde{u}_t$. Since Ψ^{-1} is generally a full real matrix, it follows that each elements of $\tilde{u}_t$ is a function of the complete quantized vector ξ'_t. Moreover, the alphabet of $\tilde{u}_t$, say $\mathcal{U}$, i.e., the discrete set $\mathcal{U} = \Psi^{-1}(\Delta\mathbb{Z} \times \cdots \times \Delta\mathbb{Z})$ obtained by using $\Psi^{-1} \in \mathbb{R}^{N \times N}$ on ξ'_t, may be large. We therefore discretize Ψ^{-1} to obtain $\hat{\Psi}^{-1} = \lfloor \Psi^{-1} \rceil_{\frac{1}{30}}$, which is a discrete approximation of Ψ^{-1} that greatly reduces the size of $\mathcal{U}$ without introducing significant distortion, i.e., the average state variance $\mathcal{J}_1$ given by

$$\mathcal{J}_1 = \frac{1}{T}\sum_{t=1}^{T} \|x_t\|^2 \qquad (18)$$

is only slightly increased.

B. Discrete Entropies

We now assess the average empirical discrete entropy $\bar{H}(\{u_t(i)\})$, where

$$\bar{H}(\{u_t(i)\}) = -\sum_{j \in \mathbb{Z}} \tilde{P}_j \log_2(\tilde{P}_j), \qquad i = 1,2,3, \qquad (19)$$

and where $\tilde{P}_j$ denotes the empirical probability that $u_t(i)$ belongs to the j^{th} quantization cell of the i^{th} codebook $\mathcal{U}_i$, i.e., $\text{Prob}\{u_t(i) = \mathcal{U}_i(j)\}$, where $\mathcal{U}_i$ denotes the discrete alphabet of $u_t(i)$. We estimate $\tilde{P}_j$ from the histogram of the complete sequence $\{u_t(i)\}_{t=1}^{T}$. It is useful to work with entropies, since the average operational bit-rate R_{op} obtained when using an entropy coder designed on the true statistics of $\{u_t(i)\}_{t=1}^{T}$, can be sandwiched in the following way:

$$\bar{H}(\{u_t(i)\}) \le R_{\text{op}} \le \bar{H}(\{u_t(i)\}) + c, \qquad (20)$$

where c is a function that is upper-bounded by 1 and tends to zero as the size of the optimum codebook (of the entropy coder) becomes large [17], [18].

As p increases, the variance of $\{\xi_t\}_{t=1}^{T}$ will also increase. However, if the variance of the input to the quantizer increases, the resulting bit-rate also increases (if the step-size is kept fixed). It is well-known that under high-resolution assumptions, and when there is no feedback, the bit-rate of a uniform scalar quantizer when quantizing a stationary sequence, say $\{\varphi_t\}$, decreases logarithmically with its step-size Δ, i.e., $R \approx \bar{h}(\{\varphi_t\}) - \log_2(\Delta)$, where $\bar{h}(\{\varphi_t\})$ denotes the differential entropy rate of $\{\varphi_t\}$ [17], [18]. Such a simple property has not been rigorously proven for quantized closed-loop control. However, in this particular case, a similar behavior can be observed. Specifically, to seek that the bit-rate is held approximately fixed as the packet-loss rate is increased, we adapt Δ accordingly by using

$$\Delta(p) = \Delta_0\left(1 + \frac{7.5p^2}{1-p^2}\right), \qquad (21)$$

where $\Delta_0 = 2$ is chosen as the initial step-size at zero packet-loss probability.

C. Construction of Packets

The packets are constructed by first entropy coding $u_t(1)$ to obtain the bit sequence $b_t(1) = \mathcal{E}(u_t(1))$ using a scalar entropy coder $\mathcal{E}(\cdot)$ designed on the statistics of $u_t(1)$. Then, the bit-stream $b_t(1)$ is repeated in both packets. Next, the remaining control vectors $u_t(2)$ and $u_t(3)$ are jointly entropy coded, conditional upon $u_t(1)$. It should be noted that $u_t(i), i > 1$ are generally correlated with $u_t(1)$ and to achieve efficient encoding, one should use conditional codebooks. With this, the bit-stream $b_t(2)$ is obtained as $b_t(2) = \mathcal{E}((u_t(2), u_t(3))|u_t(1))$ using a conditional vector entropy coder $\mathcal{E}(\cdot|\cdot)$. Finally, $b_t(2)$ is split into two parts $b_t(2) = (b'_t(2), b''_t(2))$, one for each packet. Thus, $c_t^1 = (b_t(1), b'_t(2))$ and $c_t^2 = (b_t(1), b''_t(2))$.

D. Data Rates

The operational average data rate $R_{\text{op}}(1)$ when encoding $\{u_t(1)\}$ using an optimal entropy coder $\mathcal{E}(\cdot)$ designed on the source statistics of $\{u_t(1)\}$ is shown in Table II. For comparison, the average entropy $\bar{H}(\{u_t(1)\})$ is also shown. In practice, the exact source statistics might not be available and an optimal entropy coder cannot be constructed. Towards that end, we have designed a suboptimal entropy coder $\tilde{\mathcal{E}}$ on i.i.d., Gaussian samples having a variance identical to $u_t(1)$. Thus, only knowledge of the variance is required. The operational data rate $R_{\text{sub}}(1)$ of the suboptimal entropy coder is also shown in Table II.

As mentioned above, the remaining two quantized control variables $(u_t(2), u_t(3))$ should theoretically be losslessly encoded using conditional vector entropy coding. However, generally it is complicated to form optimal conditional codebooks. For simplicity, we will encode $u_t(2)$ independently, and will therefore incur a rate penalty.

p [%]	$\bar{H}(\{u_t(1)\})$	$R_{\text{op}}(1)$	$R_{\text{sub}}(1)$
0	6.6171	6.6363	6.6473
1	6.6167	6.6359	6.6402
2	6.6157	6.6347	6.6352
3	6.6148	6.6339	6.6631
4	6.6203	6.6390	6.6383
5	6.6179	6.6368	6.6878
10	6.6204	6.6393	6.6611
15	6.6289	6.6505	6.7531
20	6.6467	6.6748	7.1904

TABLE II: Average discrete entropy $\bar{H}(\{u_t(1)\})$. Also shown are the average operational data rates $R_{\text{op}}(1)$ and $R_{\text{sub}}(1)$ obtained using optimal $\mathcal{E}$ and suboptimal $\tilde{\mathcal{E}}$ entropy coding, respectively.

It turns out that $u_t(3)$ is deterministically related to $u_t(1)$ and $u_t(2)$ in the simulations (this would generally not be the case for different prediction horizons or plant models). Thus, knowing the pair $(u_t(1), u_t(2))$, one can immediately obtain $u_t(3)$. The resulting total (average) operational data rate R_T for the $M = 2$ packets system is therefore given by

$$R_T = 2\big(R_{\text{op}}(1) + \frac{1}{2}R_{\text{cond}}\big). \qquad (22)$$

E. Comparison to $M = 1$ Packet System

In the next simulation, we assess the performance $\mathcal{J}_1$ (18) for a fixed bit-rate (entropy), as a function of the packet-loss probability p for a network which does not introduce delays. We construct an MD system with $M = 2$ packets and $N = 3$. The two packets are constructed so that receiving any single packet, allows one to reconstruct the first control signal $u_t(1)$. Moreover, if both packets are received, the entire 3-dimensional control sequence $\tilde{u}_t$ can be obtained. The step size for the MD system is given by (21). For comparison, we also construct a baseline system without MDs, i.e., with $M = 1$ packet, cf. [8]. The baseline system also uses $N = 3$, and to obtain a fair comparison, we scale the step-size of the baseline system appropriately, in order to increase the rate so that it matches that of the MD system.[4] In both simulations $T = 10^6$ and p is varied in the interval 0.01 to 0.2 in increments of 0.01. The packet dropouts are i.i.d.

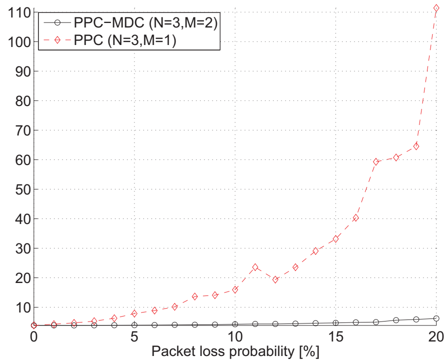

Fig. 1: Performance $\log(\mathcal{J}_1)$ (18) as a function of the packet-loss probability for the cases with and without MDs.

At $p = 0$, the resulting state variance is $\mathcal{J}_1 = 47.46$, which is slightly better than the $\mathcal{J}_1 = 47.93$ obtained with the $M = 2$ system. Since there are no erasures at $p = 0$, and due to the high bit-rates, the performance is dominated by the variance of the external disturbance. At lower rates, the differences in performances would be significantly greater (and favoring the $M = 1$ system). As p increases, it quickly becomes advantageous to use the MD system as is apparent from Fig. 1, where $\log(\mathcal{J}_1)$ of the baseline system is compared to that of the $M = 2$ system described in the previous section. A sufficient condition for stability of the baseline system is given in [8, Lemma 6], from which it can be shown that at least for $p < 0.054$, the baseline system is stable. It is clear from Fig. 1 that for this particular setup, a substantial improvement can be obtained by using MDs, i.e., PPC with $M = 2$ significantly outperforms PPC with $M = 1$.

[4] Note that the baseline system uses vector entropy coding, and does not have the problem of conditional entropy coding.

References

[1] J. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proc.IEEE*, vol. 1, pp. 138 – 162, January 2007.

[2] G. N. Nair, F. Fagnani, S. Zampieri, and R. J. Evans, "Feedback control under data rate constraints: An overview," *Proc. IEEE*, vol. 95, pp. 108 – 137, January 2007.

[3] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of control and estimation over lossy networks," *Proc. IEEE*, vol. 95, pp. 163 – 187, January 2007.

[4] G. Pin and T. Parisini, "Networked predictive control of uncertain constrained nonlinear systems: Recursive feasibility and input-to-state stability analysis," *IEEE Trans. Automat. Contr.*, vol. 56, pp. 72 – 87, January 2011.

[5] D. Muñoz de la Peña and P. D. Christofides, "Lyapunov-based model predictive control of nonlinear systems subject to data losses," *IEEE Trans. Automat. Contr.*, pp. 2076 – 2089, September 2008.

[6] D. E. Quevedo, E. Silva, and G. Goodwin, "Packetized predictive control over erasure channels," *Proc. Amer. Contr. Conf.*, 2007.

[7] G. Liu, Y. Xia, J. Chen, D. Rees, and W. Hu, "Networked predictive control of systems with random network delays in both forward and feedback channels," *IEEE Trans. Industrial Electronics*, pp. 1282 – 1297, June 2007.

[8] D. E. Quevedo, J. Østergaard, and D. Nesić, "Packetized predictive control of stochastic systems over digital channels with random packet loss," *IEEE Trans. Automatic Control*, 2011. Accepted for publication.

[9] D. E. Quevedo and D. Nešić, "Input-to-state stability of packetized predictive control over unreliable networks affected by packet-dropouts," *IEEE Trans. Automat. Contr.*, vol. 56, pp. 370 – 375, February 2011.

[10] V. Goyal, "Multiple description coding: compression meets the network," *IEEE Signal Processing Magazine*, vol. 18, pp. 74 – 93, September 2001.

[11] D. E. Quevedo, G. Goodwin, and J. De Doná, "Finite constraint set receeding horizon quadratic control," *Int. J. Robust Nonlin. Contr.*, vol. 14, pp. 355 – 377, March 2004.

[12] R. Zamir and M. Feder, "On universal quantization by randomized uniform/lattice quantizers," *IEEE Trans. Inform. Theory*, pp. 428 – 436, March 1992.

[13] R. Puri and K. Ramchandran, "Multiple description source coding using forward error correction codes," *Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers*, vol. 1, pp. 342 – 346, 1999.

[14] V. Gupta and D. E. Quevedo, "On anytime control of nonlinear processes through calculation of control sequences," in *Proc. IEEE Conf. Decis. Contr.*, (Atlanta, GA), December 2010.

[15] O. Costa, M. Fragoso, and R. Marques, *Discrete-time Markov jump linear systems*. Springer, 2005.

[16] Y. Fang and K. Loparo, "Stochastic stability of jump linear systems," *IEEE Trans. Automatic Control*, pp. 1204 – 1208, July 2002.

[17] T. Cover and J. Thomas, *Elements of information theory*. Wiley-Interscience, 2nd ed., 2006.

[18] T. Linder and R. Zamir, "Causal coding of stationary sources and individual sequences with high resolution," *IEEE Trans. Inform. Theory*, vol. 52, pp. 662 – 680, February 2006.

[19] E. I. Silva, *A Unified Framework for the Analysis and Design of Networked Control Systems*. PhD thesis, The University of Newcastle, NSW, Australia, July 2010.