A Stochastic Model Predictive Controller for Systems with Unreliable Communications

Daniel E. Quevedo∗ Prabhat K. Mishra** Rolf Findeisen*** Debasish Chatterjee**

∗Department of Electrical Engineering (EIM-E), University of Paderborn, Germany. dquevedo@ieee.org
**Systems & Control Engineering, Indian Institute of Technology Bombay, India. prabhat@sc.iitb.ac.in, dchatter@iitb.ac.in
***Institute for Automation Engineering (IFAT), Otto-von-Guericke University Magdeburg, Germany. rolf.findeisen@ovgu.de

Abstract: There is a strong trend towards feedback control over communication channels, such as wireless networks. However often communication resources are limited and unreliable, leading to random transmission loss. To deal with this issue, model predictive control seems to be well suited, since it allows to generate future input sequences and the consideration of constraints. A variety of model predictive control formulations that take into account the loss of information have been proposed in the literature. Such methods typically use online optimization of deterministic cost functions and compensate communication effects through buffering at receiving nodes. The present work presents a model predictive control scheme, which adopts a stochastic cost function that explicitly accounts for random packet dropouts. Numerical studies illustrate potential performance benefits of the proposed controller.

Keywords: model predictive control; MPC; stochastic control; networked control systems; packet dropouts; fading communication channel

1. INTRODUCTION

Wireless sensor-actuator technology is of growing interest for process and automation industry, see, e.g., Chen et al. (2011); Kim and Kumar (2012); Johansson et al. (2014); Lunze (2014); You et al. (2014). A driving force behind using wireless technology in monitoring and control applications is its lower deployment and reconfiguration costs. In addition, wireless devices can be placed where wires cannot be placed, or where power sockets are not available. However, a major drawback of wireless communication technology lies in that wireless channels are subject to fading and interference, which frequently lead to packet errors and consequently packet-dropouts. Therefore, controller design for systems using wireless technology in general needs to carefully take the communication resource into account. For that propose, communication issues need to be suitably abstracted. Of particular interest to the current paper are stochastic communication models, as commonly adopted in the communications literature; see, e.g., Tse and Viswanath (2005); Paulraj et al. (2003).

A variety of models have been utilized to describe communication issues, and in particular packet-dropouts, in estimation and control problems. The most basic approach describes network effects via independent and identically distributed (i.i.d.) random variables; see, e.g., Imre et al. (2006); Schenato et al. (2007); Tabbura and Nešić (2008); 1

The situation is different to that of wired channels, where packet errors and delays are mostly caused by congestion.

Tsumura et al. (2009); Liang et al. (2010); Kögel and Findeisen (2011); Donkers et al. (2012); Antunes et al. (2013); Yu and Fu (2016). Whilst i.i.d. communication models will often give better insight into networked control systems (NCSs) than deterministic ones, fading communication channel gains and network congestion levels are, in general, correlated, see Baddour and Beaulieu (2005); Lindbom et al. (2002); Leizarowitz et al. (2006). The time variability of wireless channel may be caused by moving machines, vehicles, people, and so forth, or when the receiver or the transmitter are mounted on a moving object.² This motivates the adoption of models where dropouts are allowed to be correlated. For example, the work Wu and Chen (2007) models the times between successful transmissions as a finite, and thereby bounded, Markov chain Kemeny and Snell (1960). In contrast, Huang and Dey (2007); Smith and Seiler (2003); Xie and Xie (2009); You and Xie (2011) describe the dropout process directly as a Markov chain and, thus, allow for the time between transmission successes to be unbounded. More general semi-Markov models have been investigated in Censi (2011); Quevedo et al. (2013a) within the context of Kalman filtering with intermittent observations.

Through the use of online optimizations over finite horizons, Model Predictive Control (MPC) methods have a remarkable capability to handle a variety of constrained

² The time-variability of the fading channel can be partially compensated for through control of the power levels used by the radio amplifiers, e.g., using predictive control, see Quevedo et al. (2014b).
control problems, see, e.g., Mayne (2014) for a recent survey or the textbooks Gröne and Pannek (2011); Rawlings and Mayne (2009). It is therefore not surprising that significant research efforts have been devoted to develop interesting MPC formulations for NCSs. In particular, to compensate for the effect of unreliable communications, so-called packetized predictive control (PPC) methods have been proposed. These exploit the capability of modern communication protocols to send, possibly large, and time-stamped packets. This opens the possibility to transmit packets of data containing finite sequences, rather than individual values. Through buffering and appropriate selection logic at the receiver node, time delays and packet dropouts can, at least to some extent, be compensated for. When dealing with controller-to-actuator connections, PPC based on MPC is especially convenient, since tentative plant input sequences are directly provided by the MPC optimizer.

Deterministic stability results of PPC algorithms when communication effects can be deterministically bounded, have been obtained in various works, including Tang and de Silva (2007); Muñoz de la Peña and Christofides (2008); Chaillot and Bicchi (2008); Findeisen and Varutti (2009); Pin and Parisini (2009); Quevedo and Nešić (2011) and, more recently, in Nagahara et al. (2014) which adopts a sparsity promoting cost function. Despite the widespread use of stochastic network models in the communications community, such models have only recently been used to investigate stability and performance of PPC formulations for unreliable channels. In particular, the work presented in Quevedo et al. (2011) studies quantized control of perturbed LTI systems with i.i.d. packet dropouts. By adapting results from Markov Jump-linear systems and investigating random-time drift conditions, Quevedo et al. (2011) derived sufficient conditions for mean-square stability and presents closed form expressions for spectra of interest, see also Quevedo et al. (2013b). For nonlinear plant models and Markovian dropouts, Quevedo and Nešić (2012) established sufficient conditions for stochastic stability of nonlinear plant models and Reble et al. (2013) developed performance estimates.

Common to the works mentioned in the above paragraph is that, whilst dropout effects are compensated through the use of PPC and some of the analysis methods allow for stochastic channels, the controller itself is deterministic, i.e., the MPC cost function does not take into account the fact that the communication channel is unreliable. It is intuitively clear that such deterministic controllers will, in general, be outperformed by carefully designed stochastic PPC formulations.

Stochastic versions of MPC-related techniques initially evolved within the operations research community, with inventory and manufacturing systems as primary application areas, and have steadily filtered into the domain of control systems, with current applications in financial engineering, process control, industrial electronics, power systems, etc.; see Chatterjee et al. (2011); de la Peña et al. (2005); Kouvaritakis et al. (2010); Bertsimas and Brown (2007); Primbs and Sung (2009); Felt (2003); Goodwin et al. (2009); Bernardini and Bemporad (2009); Batina et al. (2001); Mesbah et al. (2014); Skaf and Boyd (2009); van Hessem and Bosgra (2003) and the references therein.

However, the analysis of stochastic PPC for systems with unreliable communications has been largely unexplored, save for the work of Fischer (2014), which is restricted to LTI plants and i.i.d. communication effects. Unfortunately, extending this approach to nonlinear plant models will in general lead to highly involved online optimizations, which are needed to derive closed-loop control policies.

Motivated by the above, the present work presents a stochastic MPC formulation, which is related to PPC and can be applied to plant models with linear dynamics and constrained inputs. By adapting ideas from Chatterjee et al. (2011), the focus is on policies that are affine in bounded functions of the past disturbances. Thereby, our current approach directly works with hard constraints on the controls, i.e., does not require artificially relaxing the hard constraints on the controls to soft probabilistic ones (to ensure large feasible sets), and still provides a globally feasible solution to the problem. The effects of the noise and channel dropouts appear in the resulting convex optimization problem as fixed cross-covariance matrices, which may be computed off-line via Monte Carlo methods and stored.

The remainder of this paper is organized as follows: Section 2 presents the control architecture of interest. The deterministic PPC method of Quevedo and Nešić (2012) is briefly revised in Section 3. Section 4 presents the stochastic MPC formulation. Section 5 documents numerical studies which compare the performance of the two controllers. Section 6 draws conclusions.

Notation: We write $\mathbb{R}$ for the real numbers, $\mathbb{N}$ for $\{1, 2, 3, \ldots\}$, and $\mathbb{N}_0$ for $\mathbb{N} \cup \{0\}$. The $p \times p$ identity matrix is denoted via $I_p$, whereas $0_p \equiv 0 \cdot I_p$. We adopt the definition $\sum_{i=1}^n a_i = 0$, if $\ell_1 > \ell_2$ and irrespective of $a_i$. The weighted Euclidean norm of a vector $x$ is denoted via $\|x\|_p = \sqrt{x^T P x}$; if $X$ is a set, then $|X|$ denotes its cardinality. For scalars, $\cdot$ refers to absolute value, and $\lfloor \cdot \rfloor$ denotes the floor function. To denote the unconditional probability of an event $\Omega$, we write $\Pr(\Omega)$. The conditional probability of $\Omega$ given an event $\Gamma$ is denoted via $\Pr(\Omega | \Gamma)$. The expected value of a random variable $\nu$ given $\Gamma$, is denoted by $\mathbb{E}[\nu | \Gamma]$, whereas for the unconditional expectation we will write $\mathbb{E}\{\nu\}$. We use the same notation for random variables and their realizations; what is meant will depend upon the context.
Throughout this work, we shall focus on a single-loop control architecture with an unreliable input communication channel, as depicted in Fig. 1. The plant model is described in discrete-time via:

\[ x_{k+1} = f(x_k, u_k, w_k), \quad k \in \mathbb{N}_0, \]  

where \( x \in \mathbb{R}^n \) is the plant state, and \( u \in \mathbb{U} \subseteq \mathbb{R}^p \) denotes the (constrained) plant input. The plant model is perturbed by a random process \( \{w_k\}_{k \in \mathbb{N}_0} \), which in general does not have bounded support. As is common in most MPC formulations, we shall assume that the controller has direct access to the plant state. \(^3\) We also assume that clocks are synchronized and that there are no transmission delays.

Transmissions of controller outputs, say \( c_k \) to the actuator node is through an unreliable communication channel which is affected by packet dropouts. We shall model transmission effects via a binary random process \( \{\gamma_k\}_{k \in \mathbb{N}_0} \):

\[ \gamma_k = \begin{cases} 0 & \text{if packet-dropout occurs at instant } k, \\ 1 & \text{if packet-dropout does not occur at instant } k. \end{cases} \]

This leads to the following temporal ordering:

\[ x_k \rightarrow c_k \rightarrow \gamma_k \rightarrow u_k, w_k \rightarrow x_{k+1} \rightarrow \ldots. \]  

In the simplest instance, the dropouts are modeled as i.i.d., leading to a Bernoulli channel model. More general descriptions allow \( \{\gamma_k\}_{k \in \mathbb{N}_0} \) to be correlated. For instance, Fig. 2 illustrates a simplified version of the model adopted in Quevedo et al. (2013a). Here, communication reliability is modeled via an underlying network state process \( \{\Xi_k\}_{k \in \mathbb{N}_0} \) taking on values in a finite set

\[ \mathbb{B} = \{1, 2, \ldots, |\mathbb{B}|\}. \]

Each \( i \in \mathbb{B} \) corresponds to a different environmental condition (such as network congestion or positions of mobile objects). For a given network state the dropouts are modeled as i.i.d. with probabilities

\[ p_i = \Pr(\gamma_k = 1 | \Xi_k = i), \quad i \in \mathbb{B}. \]  

Correlations can be captured in this model by allowing \( \{\Xi_k\}_{k \in \mathbb{N}_0} \) to be time-homogeneous Markovian with transition probabilities:

\[ p_{ij} = \Pr(\Xi_{k+1} = j | \Xi_k = i), \quad \forall i, j \in \mathbb{B}, k \in \mathbb{N}. \]  

It is easy to see that the above model encompasses Markovian dropouts and i.i.d. dropouts as special cases.

### 3. DETERMINISTIC PACKETIZED PREDICTIVE CONTROL

This section briefly revises the PPC formulation adopted in Quevedo and Nešić (2012), which is similar to other methods described in the literature. As foreshadowed in the introduction, at each time instant \( k \) and for plant state \( x_k \), with PPC the control packet \( c_k \), is computed and transmitted over the dropout channel to the actuator node. To achieve good performance despite unreliable communication, \( c_k \) contains tentative plant inputs for times \( \{k, k+1, \ldots, k+N-1\} \).

Key to the idea of PPC is the use of a “smart actuator”, which contains a buffer and selection logic. The buffering-mechanism amounts to a parallel-in serial-out shift register, which acts as a safeguard against dropouts. For that purpose, the buffer state, say \( b_k \in \mathbb{U}^N \), is overwritten whenever a valid (i.e., uncorrupted and undelayed) control packet arrives. Actuator values are passed on to the plant sequentially until the next valid control packet is received. If the buffer runs out of data, then the plant input is set to zero. More formally, we have:

\[ b_k = (1 - \gamma_k) S b_{k-1} + \gamma_k c_k, \]  

\[ u_k = e^T b_k \]

where

\[ S \triangleq \begin{bmatrix} 0_p & I_p & \cdots & 0_p \\ \vdots & \ddots & \ddots & \vdots \\ 0_p & \cdots & 0_p & I_p \\ 0_p & \cdots & 0_p & 0_p \end{bmatrix} \in \mathbb{R}^{pN \times pN}, \]

\[ e^T \triangleq [I_p, 0_p, \ldots, 0_p] \in \mathbb{R}^{p \times pN}. \]

The control packets \( c_k \) are formed by adapting the ideas underpinning MPC: At each time instant \( k \) and for a given plant state \( x_k \), the following cost function is minimized:

\[ J(u', x_k) \triangleq F(x'_N) + \sum_{\ell=0}^{N-1} L(x'_{\ell}, u'_\ell). \]  

The cost function in (8) examines predictions of the plant (1) over a finite horizon of length \( N \), which is taken equal to the buffer size. For simplicity, the predicted state trajectories do not take into account packet-losses or disturbances and are, thus, generated by the nominal model:

\[ x'_{\ell+1} = f(x'_{\ell}, u'_\ell, 0), \quad \ell \in \{0, 1, \ldots, N-1\} \]

starting from \( x'_0 = x_k \) and where the entries in

\[ u' = \left[ (u'_0)^T \ldots (u'_{N-1})^T \right]^T \in \mathbb{U}^N \]

are the associated plant inputs. Predicted plant states and inputs are penalized via the per-stage weighting function \( L(\cdot, \cdot) \) and the terminal weighting \( F(\cdot) \). These design variables allow one to trade-off control performance versus control effort. As in control loops without dropouts, see, e.g., Rawlings and Mayne (2009), the choices made for \( L(\cdot, \cdot) \) and \( N \) influence closed loop (stochastic) stability and performance, see Quevedo and Nešić (2012) for details.
The control packet $c_k$, see Fig. 1, is set equal to the constrained optimizer\(^4\),

$$c_k \triangleq \arg \min_{u \in U} J(u', x_k)$$

(10)
and is sent through the network to the buffer. Therefore, at each instant, $N$ tentative plant-input values are transmitted through the communication channel.

Following the receding horizon optimization idea, at the next sampling step and given $x_{k+1}$, the horizon is shifted by one and another optimization is carried out, providing $c_{k+1} = \arg \min_{u \in U} J(u', x_{k+1})$, sequence, which is transmitted over the unreliable channel to the smart actuator. This procedure is repeated \textit{ad infinitum}.

Note that $c_k$ in (10) contains tentative constrained plant input values for instants $\{k, \ldots, k+N-1\}$. If $c_k$ is received at time $k$, then these values are written into the buffer and implemented sequentially until some future (valid) control packet arrives. If there are no dropouts then the scheme reduces to regular deterministic MPC.

In the case of Markovian dropouts (i.e., where in (4), $p_1 = 1$ and $p_0 = 0$), Quevedo and Nešić (2012) showed how to design the tuning parameters $N$, $L(\cdot, \cdot)$ and $F(\cdot)$ to ensure boundedness of moments of the plant state.

It is important to emphasize that in this deterministic formulation, the plant input design is done dynamically such as to optimize performance, but without using knowledge on the dropout distribution parameters $p_0$ and $p_1$, or receipt acknowledgements, see Fig. 1. The optimization (8) does not explicitly take into account the communication issues, save for what can be captured by the tuning parameters. It is intuitively clear, that a stochastic controller, which includes a model of communications will in general give better performance. In the following section, we will present such an algorithm. It uses a quadratic cost function and focuses on plant models with linear dynamics, but goes beyond the work of Fischer (2014) by also taking into account constraints on system inputs.

4. A STOCHASTIC MPC FORMULATION

With the aim of developing a high-performance MPC formulation with moderate online computational complexity, in the present section we will embellish the approach of Chatterjee et al. (2011) to handle communication issues.

4.1 Setup

We shall restrict our attention to perturbed LTI plant models of the form

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

(11)
where the elements of the plant input are constrained in magnitude as per:

$$u_k \in U \triangleq \{v \in \mathbb{R}^p : \|v\|_\infty \leq U_{\text{max}}\}, \quad \forall k \in \mathbb{N}_0.$$  

(12)
Communication is over an i.i.d. dropout channel with reliable causal acknowledgements as in Fig. 1, i.e., the sequence $\{\gamma_0, 1, \ldots, \gamma_{k-1}\}$ is known to the controller at time $k \in \mathbb{N}$, see (3), \(^5\)

\(^4\) We assume that the optimizer is unique.

\(^5\) If acknowledgements are unreliable, then optimisations are significantly harder, see, e.g., Nourian et al. (2014).

There are two sources of randomness in this system, namely, the disturbance process $\{w_k\}_{k \in \mathbb{N}_0}$ and the dropout process $\{\gamma_k\}_{k \in \mathbb{N}_0}$, see (2). Both are here taken as i.i.d. processes.

4.2 Cost function and policy class

To incorporate the random phenomena in the controller, the objective function is chosen as the expectation of the predicted quadratic cost given the current plant state, cf. (8):

$$V(\pi, x_k) = \mathbb{E}\left\{\|x'_N\|_P^2 + \sum_{\ell=0}^{N-1} \left\{\|x'_\ell\|_Q^2 + \|u'_\ell\|_R^2\right\} | x_k\right\},$$

(13)
where $P$ and $Q$ are positive semi-definite, and $R$ is a positive definite matrix. As before, primed variables $x'_\ell$ and $u'_\ell$ are predicted values which respect system dynamics and constraints (12). However, in this case the system model with disturbances in (11) and not the nominal model (9), is used.

A perhaps subtle but important feature of this formulation is that the optimization is not over finite-length sequences, but over causal history-dependent feedback policies of finite length, i.e.,

$$\pi = \{\pi_0, \pi_1, \ldots, \pi_{N-1}\},$$

see, e.g., Bertsekas (2005), so that

$$u'_\ell = \pi_\ell(x_k, x_{k+1}, \ldots, x_{k+\ell}), \quad \forall \ell \in \{0, 1, \ldots, N-1\}. \quad (14)$$

Minimizing (13) over all classes of causal policies commonly leads to a computationally intractable problem. Hence, we will restrict our attention to a specific class of policies which are affine in past disturbances. Following as in Chatterjee et al. (2011), the N-history-dependent policies in (14) are parameterized as per

$$u'_\ell = \eta_\ell + \sum_{i=0}^{\ell-1} \theta_{\ell, i}\phi_{i+1}(w_{k+i}), \quad \ell \in \{0, 1, 2, \ldots, N-1\}, \quad (15)$$

where $\eta_\ell \in \mathbb{R}^p$ and $\theta_{\ell, i} \in \mathbb{R}^{p \times n}$ are the policy parameters to be determined using $x_k$, statistics of disturbances and dropouts and system parameters. In particular, $\eta_\ell$ can be interpreted as predicted plant input sequences, whereas $\theta_{\ell, i}$ gauges how future information on disturbances is used to readjust the control signal. In (15), $\phi_i : \mathbb{R}^n \rightarrow \mathbb{U}$ are given bounded maps, i.e., there exists $\varphi_{\text{max}} \in \mathbb{R}$, such that

$$\|\varphi_i(w)\|_\infty \leq \varphi_{\text{max}}, \quad \forall (i, w) \in \{1, 2, \ldots, N\} \times \mathbb{R}^n.$$  

(16)

The terms $\phi_i$ may represent functions that are easy or inexpensive to implement, or be the only ones available for specific applications. For instance, piecewise constant policy elements with finitely many elements in their range may be viewed as a hybrid controller with a finite control alphabet, cf., Aguilera and Quevedo (2015).

\textbf{Remark 1.} Due to acknowledgements, at any time $k$ the controller has access to $c_{k-1}$, $x_k$, $x_{k-1}$ and $\gamma_{k-1}$, see (3), and thereby also knows the plant input $u_{k-1}$. Thus, using

\textbf{Boundness is important to be able to ensure that the controls satisfy the constraint $u_k \in U$.}
the system model (11), disturbances can be recovered by the controller after a one-step delay, by simply evaluating \( w_{k+1} = x_k - Ax_{k-1} - Bu_{k-1} \).

Expression (15) can be rewritten in vector form, leading to

\[
\begin{bmatrix}
  u_0^* \\
  u_1^* \\
  \vdots \\
  u_{N-1}^*
\end{bmatrix} = \eta + \Theta \left[ \begin{array}{c}
  \varphi_1(w_k) \\
  \varphi_2(w_{k+1}) \\
  \vdots \\
  \varphi_{N-1}(w_{k+N-2})
\end{array} \right] := \eta + \Theta \varphi(W_k),
\]

(16)

where \( \eta \in \mathbb{R}^{pN} \) and \( \Theta \in \mathbb{R}^{pN \times n(N-1)} \) is strictly lower block-diagonal. Thus, the policy \( \pi \) is characterized by its parameters contained in the pair \( (\eta, \Theta) \). The hard constraint on control (12) for the above policy is satisfied by the following element-wise inequality:

\[
|\eta^{(i)}| + \varphi_{\text{max}}\|\Theta^{(i)}\|_1 \leq U_{\text{max}} \quad \forall i \in \{1, \ldots, pN\},
\]

where \( k \in \mathbb{K} \triangleq \{0, N_r, 2N_r, \ldots\} \).

### 4.3 Optimization and Transmission Protocol

Different to the method in Section 3, the minimization of (13) is not carried out at every instant \( k \in \mathbb{N} \), but only every \( N_r \leq N \) time steps. We shall refer to \( N_r \) as the recalculation interval, not to be confused with the constraint horizon used in some MPC schemes. Using notation as presented in the preceding section, this leads to the sequence of optimizing policy parameters:

\[
(\eta^*_k, \Theta^*_k) = \arg \min_{\eta, \Theta} V(\pi, x_k),
\]

(17)

where \( k \in \mathbb{K} \).

Recalling Remark 1, at each time

\[
t \in \{k, k+1, \ldots, k+N_r-1\}, \quad k \in \mathbb{K},
\]

the pair \( (\eta^*_k, \Theta^*_k) \) is used by the controller to calculate the optimal control input using the expression (see (15)):

\[
u_t^* = \eta^*_{k,t} + \sum_{i=0}^{\ell-1} \Theta^*_{k,t,i} \varphi_i(w_{k+i}),
\]

where

\[
\ell = t - k \in \{0, 1, 2, \ldots, N_r - 1\}
\]

and where \( \eta^*_k, \Theta^*_k \in \mathbb{R}^p \) denotes the \( \ell \)-th block of \( \eta^*_k \) and \( \Theta^*_k \) is the \( (\ell, i) \)-th block of \( \Theta^*_k \).

To improve robustness against packet dropouts, akin to what is done in PPC, buffering at the actuator node is used. To be more specific, at each time \( t \), the controller sends the current value \( u_t^* \). It also transmits the remaining \( N_r - \ell \) blocks of the current sequence \( \eta_k \), with

\[
k = N_r \lfloor t/N_r \rfloor,
\]

provided they have not already been received by the smart actuator.

Successfully received values \( u_t^* \) are directly applied at the plant input. At instances where dropouts occur, the actuator uses the corresponding block of \( \eta_k \) if available, or otherwise sets the plant input to zero. We thus have:

\[
u_t = \begin{cases} u_t^*, & \text{if } \gamma_t = 1 \\ u_t^*_{k,t}, & \text{if } \gamma_t = 0 \text{ and } \sum_{i=0}^{\ell-1} \gamma_{k+i} \geq 1 \\ 0, & \text{if } \sum_{i=0}^{\ell-1} \gamma_{k+i} = 0, \end{cases}
\]

(18)

for all \( t \in \mathbb{N}_0 \).

#### 4.4 Properties

As described above, if the proposed stochastic MPC algorithm is used, then what is transmitted depends on the dropout realizations. The number of values transmitted over a period of length \( N_r \) corresponding to a given optimizer in (17), say \( C_k \), becomes a finite random process.

To elucidate this situation, we shall set \( N = N_r \) and denote the transmission success probability (assumed i.i.d.) by \( p = \mathbb{E}\{\gamma_k\} \). Then, it is easy to see that

\[
\Pr\{C_k = j\} = \begin{cases} (1-p)^{p_i}, & \text{if } j = N - 1 + \sum_{i=0}^{\ell - 1} (N - \ell) \\ (1-p)^{N-1}, & \text{if } j = N - 1 + \sum_{i=0}^{\ell - 1} (N - \ell) \end{cases}
\]

with \( i \in \{0, 1, \ldots, N - 2\} \). This stands in contrast to the deterministic PPC approach described in Section 3, where at every instant a fixed number of \( N \) values is transmitted, so that \( C_k = N^2 \) for all \( k \in \mathbb{N}_0 \).

In a worst case (corresponding to \( p = 0 \)), with the stochastic controller, the number of values transmitted during a period of \( N \) steps is given by

\[
N - 1 + \frac{N(N + 1)}{2} = \frac{N^2 - (N - 2)(N - 1)}{2}
\]

Thus, for \( N_r = N > 2 \) or success probabilities \( p > 0 \), the proposed stochastic MPC method uses strictly less communication resources than deterministic PPC.

The optimization in (17) can be carried out by taking into account system dynamics, policy class and the effect of dropouts as in (18). Interestingly, despite the apparent complexity, it can be shown that the resulting programme is convex. Its parameters depend on the problem data (including disturbance statistics) and can be computed offline. In future work, we will investigate how to tune controller parameters \( P, Q, R, N_r \) and \( N_r \) such that, for Lyapunov stable plant models (e.g., featuring an orthogonal system matrix \( A \)), the system state \( x_k \) is mean square bounded under the hard control constraints (12).

#### 5. NUMERICAL STUDY

Consider the two dimensional system

\[
x_{k+1} = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.8 \end{bmatrix} x_k + \begin{bmatrix} 0.14 \\ 0.12 \end{bmatrix} u_k + w_k, \quad |u_k| \leq 10,
\]

where \( w_k \) is i.i.d. Gaussian of mean zero with variance \( 4I_2 \).

We focus on a special case of the packet dropout model in Fig. 2: When the channel is in “good” state, packets are received...
always successfully delivered, whereas in the “bad” state packets are always dropped, i.e., we set $p_1 = 1, p_2 = 0$. The transition probability to switch from a bad channel state to the good state is taken as $p_{21} = 0.9$; the failure rate is taken as $p_{12} = 0.1$.

The stochastic controller proposed in Section 4 is designed with state and control weights $Q = I_2, P = 1.5I_2, R = 0.95$, optimization horizon $N = 3$ and recalculation interval $N_r = 2$. The disturbance feedback terms in the policy (16) are chosen as a vector of element-wise scalar sigmoidal functions $\varphi_k(\xi) = (1 - e^{-\xi})/(1 + e^{-\xi})$, which are applied to each coordinate of $w_k$. The various covariance matrices that are required to solve the (off-line) optimization problem were computed empirically via classical Monte Carlo methods, see Robert and Casella (2013) using $10^6$ i.i.d. samples. Computations for determining our policy were carried out in the MATLAB-based software package YALMIP, see Löfberg (2004), and were solved using SDPT3-4.0, see Toh et al. (2006).

For the PPC approach described in Section 3, the deterministic counterpart of the quadratic cost in (13) with parameters $Q = I_2, P = 1.5I_2$ and $N = 3$ was used. We consider two cases for the control weighting $R = 0.95$ (as in the stochastic controller case above) and also $R = 1.65$.

Fig. 3 documents the simulation results averaged over 100 sample paths with initial state $x_0 = [10 \ 10]^T$. In addition to the averaged Euclidean norm of the system states, the figure also illustrates the empirical bound on the absolute values of the control inputs observed. The expected total cost for the deterministic controller comes out to be 68.6168 units and that for the stochastic controller is 70.3898 units, which are roughly comparable. However, the deterministic controller transmits 300 control values through the channel while the stochastic controller needs to transmit on average only 150.67 control values, a significant reduction in communication. Further, the average of total actuator energy is $104.8296$. The performance in terms of the expected norm of state and the expected total cost is observed to be approximately equal for both the controllers but the stochastic controller has the added advantage of significantly less communication and the saving of actuator energy.

6. CONCLUSIONS

Control with unreliable communications resources has received significant attention. Due to their versatility, MPC methods are good candidates to tackle a variety of design problems arising in this context. However, typical MPC formulations often adopt deterministic communication models. In contrast, communications theory and practice has shown that modeling communication links via stochastic processes is in general preferable. With the above as background, the present work has presented a stochastic MPC formulation for control when access to the actuator is affected by random dropouts. Simulation results suggest that, when comparing to earlier deterministic packetized predictive control, performance gains can be obtained with the proposed control algorithm.

The current formulation is restricted to control with direct state feedback and where dropouts in the actuator channel are i.i.d. In future work, we will focus on overcoming these limitations. More general extensions of the ideas presented here may include multi-channel systems (see, e.g., Kögel and Finkel (2013); Lješnjačin et al. (2014)), allowing for self-triggered or event-triggered operation (see, e.g., Donkers and Heemels (2012); Araújo et al. (2014); Nagahara et al. (2016); Quevedo et al. (2014a)) and distributed control; see, e.g., Maestre and Negenborn (2014). Furthermore, the challenging issue of satisfaction of state constraints might be an important issue to tackle.

REFERENCES

