Proceedings of the 2nd IFAC Workshop on Distributed
Estimation and Control in Networked Systems, Centre de
Congrès de L'Impérial Palace, Annecy, France,
September 13-14, 2010

PREPRINTS

A Model Predictive Control Approach for Stochastic Networked Control Systems ⋆

D. Bernardini * M.C.F. Donkers ** A. Bemporad ***
W.P.M.H. Heemels **

* *Department of Information Engineering, University of Siena, Italy, (email: bernardini@dii.unisi.it).*
** *Department of Mechanical Engineering, Eindhoven University of Technology, PO Box 513, 5600 MB Eindhoven, the Netherlands, (email: {m.c.f.donkers,m.heemels}@tue.nl).*
*** *Department of Mechanical Engineering, University of Trento, Italy, (email: bemporad@ing.unitn.it).*

Abstract: In this paper we present a stochastic model predictive control (SMPC) approach for networked control systems (NCSs) that are subject to time-varying sampling intervals and time-varying transmission delays. These network-induced uncertain parameters are assumed to be described by random processes, having a bounded support and an arbitrary *continuous* probability density function. Assuming that the controlled plant can be modeled as a linear system, we present a SMPC formulation based on scenario enumeration and quadratic programming that optimizes a stochastic performance index and provides closed-loop stability in the mean-square sense. Simulation results are shown to demonstrate the performance of the proposed approach.

1. INTRODUCTION

In recent years, a vast literature has been produced on modeling, analysis and control design of Networked Control Systems (NCSs) (see, e.g., Antsaklis and Baillieul (2007); Bemporad et al. (2010); Hespanha et al. (2007); Zhang et al. (2001) and references therein). Besides the many advantages offered by NCSs, such as increased system flexibility and low installation and maintenance costs, the presence of a network also introduces sources of uncertainty that need to be properly managed. These uncertainties are caused by time-varying delays, time-varying sampling intervals, and packet dropouts. A traditional approach to deal with such phenomena is to attribute deterministic bounds to them, neglecting any available statistical information. However, network-induced disturbances can be often, and more accurately, modeled as random processes described by a probability distribution. A common way to tackle such stochastic disturbances which have a probabilistic description is to assume that they can take only a finite or countable number of values, assigning a realization probability to every possible value (see, e.g, Montestruque and Antsaklis (2004); Seiler and Sengupta (2005)). Nonetheless, with this approach nothing can be concluded about the stability of the closed-loop system if the uncertain parameters have a continuous, uncountable domain.

In this paper, we consider a linear plant and propose a control scheme to stabilize the NCS system in the presence of time-varying sampling intervals and time-varying delays, which are modeled as random processes described by *continuous* probability density functions (PDFs). For stability analysis of such systems given a controller and dealing with continuous PDFs two lines of research can be distinguished. First there is the approach of Antunes et al. (2009), based on the modeling of continuous-time impulsive systems. Alternatively, Donkers et al. (2010) use a NCS model in the discrete-time domain, in such a way that the statistical properties of the network model are preserved in a suitable sense. As in this paper the objective is controller synthesis based on Model Predictive Control (MPC) for NCS, we will exploit the approach of Donkers et al. (2010) since MPC is typically suitable for discrete-time models.

The basic idea of MPC is to obtain the control input by solving at each sampling time an open-loop finite-horizon optimal control problem based on a given prediction model of the process, by taking the measured (or estimated) state as the initial state. Recently *stochastic* MPC (SMPC) control schemes were formulated, where the available statistical information on the disturbance is exploited in order to minimize a stochastic performance index (see, e.g., Couchman et al. (2006); Primbs (2007), and references therein). In this work we adopt a formulation derived from Bernardini and Bemporad (2009) based on scenario enumeration, which exploits ideas from multi-stage stochastic optimization to possibly improve closed-loop performances with respect to standard deterministic MPC algorithms. Integrating the NCS models of Donkers et al. (2010) with the SMPC of Bernardini and Bemporad (2009) offers a general framework for MPC control of stochastic NCSs.

⋆ This work was partially supported by the European Commission under project "WIDE - Decentralized and Wireless Control of Large-Scale Systems", contract number FP7-IST-224168.

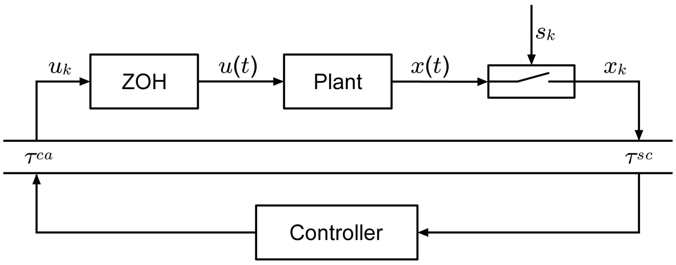

Fig. 1. NCS overview scheme

2. NCS MODEL AND PROBLEM STATEMENT

In the following we describe a NCS that includes unknown time-varying sampling intervals and unknown time-varying delays. A schematic of the considered NCS is shown in Fig. 1. It consists of a linear continuous-time plant

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

with $A \in \mathbb{R}^{n_x \times n_x}$ and $B \in \mathbb{R}^{n_x \times n_u}$, and a discrete-time controller, connected over a communication network that induces network delays, namely the sensor-to-controller delay τ^{sc} and the controller-to-actuator delay τ^{ca}. A complete measurement of the state vector $x(t)$ is assumed to be available at the sampling time instants

$$s_k = \sum_{i=0}^{k-1} h_i \quad \forall k \geq 1, \quad s_0 = 0, \tag{2}$$

which may not be equidistantly spaced in time due to the time-varying sampling intervals $h_k > 0$. The sequence $s_0, s_1, s_2, \ldots$ is assumed to be strictly increasing, i.e., $s_{k+1} > s_k$, for all $k \in \mathbb{N}$. We denote by $x_k = x(s_k)$ the kth sampled value of the state x and by u_k the corresponding control value. The zero-order-hold (ZOH) function in Fig. 1 transforms the discrete-time control input u_k to the continuous-time control input $u(t)$ applied to the plant.

In the presented model both the variable computation time τ_k^c, needed to evaluate the control law, and the time-varying network-induced delays, i.e., the sensor-to-controller delay τ_k^{sc} and the controller-to-actuator delay τ_k^{ca}, are taken into account. We assume that the sensor acts in a time-driven fashion (i.e., sampling occurs at the times s_k defined in (2)), and that both the controller and the actuator act in an event-driven fashion (i.e., they respond instantaneously to newly arrived data). Under these assumptions, all three delays can be captured by a single delay $\tau_k = \tau_k^{sc} + \tau_k^c + \tau_k^{ca}$ (see, e.g., Zhang et al. (2001)). Considering this total delay τ_k, the continuous-time input signal $u(t)$ can be defined as

$$u(t) = u_k \ \text{ if } \ t \in [s_k + \tau_k, \ s_{k+1} + \tau_{k+1}), \ \forall k \in \mathbb{N}. \tag{3}$$

Furthermore, we assume that both the sampling intervals and the delays are bounded, with the delays equal or smaller than the sampling intervals, i.e., $\tau_k \leq h_k$, for all $k \in \mathbb{N}$. We also assume that the realizations of h_k and τ_k are driven by an Independent and Identically Distributed (IID) random process, characterized by a given PDF, in accordance with the following assumption.

Assumption 1. There exists a h_{max} such that, for each $k \in \mathbb{N}$, the sampling interval h_k and the network delay τ_k are described by an IID random process, characterized by a PDF $p : \mathbb{R}^2 \to \mathbb{R}^+$, with $p(h,\tau) = 0$ for all $(h,\tau) \notin \Theta$, where

$$\Theta = \left\{(h,\tau) \in \mathbb{R}^2 \ | \ h \in (0, h_{max}] \ \wedge \ \tau \in [0, h]\right\}. \tag{4}$$

By discretizing the linear plant (1) at the sampling times s_k, $k \in \mathbb{N}$, we obtain

$$x_{k+1} = e^{Ah_k}x_k + \int_0^{h_k-\tau_k} e^{As}dsBu_k + \int_{h_k-\tau_k}^{h_k} e^{As}dsBu_{k-1}.$$

Using now the lifted state vector $\xi_k = \left[x_k^\mathsf{T} \ u_{k-1}^\mathsf{T}\right]^\mathsf{T}$, that includes the current system state and past system input and whose dimension is $n_\xi = n_x + n_u$, the NCS is formulated as the stochastically parameter-varying discrete-time system

$$\xi_{k+1} = \underbrace{\begin{bmatrix} e^{Ah_k} & \int_{h_k-\tau_k}^{h_k} e^{As}dsB \\ 0 & 0 \end{bmatrix}}_{=:\, \tilde{A}_{h_k,\tau_k}} \xi_k + \underbrace{\begin{bmatrix} \int_0^{h_k-\tau_k} e^{As}dsB \\ I \end{bmatrix}}_{=:\, \tilde{B}_{h_k,\tau_k}} u_k. \tag{5}$$

The problem studied in this paper is to design a control scheme for the NCS model given by system (5), in where the sampling intervals and transmission delays satisfy Assumption 1. The purpose of the control action is optimize a given performance index while guaranteeing mean-square closed-loop stability, according to the following definition.

Definition 1. System (5) is said to be *Uniformly Globally Mean-Square Exponentially Stable* (UGMSES) if there exist $c \geq 0$ and $0 \leq \lambda < 1$ such that for any initial condition $\xi_0 \in \mathbb{R}^{n_\xi}$ it holds that

$$\mathbb{E}[\|\xi_k\|^2] \ \leq \ c\|\xi_0\|^2\lambda^k, \ \ \forall k \in \mathbb{N}. \tag{6}$$

3. OVERAPPROXIMATION OF NCS MODEL

Direct controller synthesis based on (5) is difficult, due to the infinite number of possible values of the sampling intervals and delays $(h_k, \tau_k) \in \Theta$, and to the nonlinear appearance of these uncertain parameters in the matrices $\tilde{A}_{h_k,\tau_k}$, $\tilde{B}_{h_k,\tau_k}$of the discrete time NCS model. A way to make the system (5) amenable for controller synthesis is to overapproximate it by a system in which the uncertainties appear in a polytopic and/or additive manner. This can be achieved by using one of the available overapproximation methods (see Heemels et al. (2010) for an overview and thorough comparison of all the existing overapproximation techniques). Here, we take a method derived in Cloosterman et al. (2009), that is based on the real Jordan form of the continuous-time system matrix A, although other techniques can be used as well. In the following this method is briefly summarized.

Let the state matrix $A = TJT^{-1}$, with J the real Jordan form of A, and T an invertible matrix. The integrals in (5) are computed by substituting $e^{As} = Te^{Js}T^{-1}$, in order to obtain a model in which the uncertain parameters h_k and τ_k appear explicitly. This leads to a model of the form

$$\xi_{k+1} = \tilde{A}_{h_k,\tau_k}\xi_k + \tilde{B}_{h_k,\tau_k}u_k, \tag{7}$$

with $(h_k, \tau_k) \in \Theta$, for all $k \in \mathbb{N}$, where we can rewrite $\tilde{A}_{h_k,\tau_k}$ and $\tilde{B}_{h_k,\tau_k}$ in (5) as

$$\begin{aligned} \tilde{A}_{h_k,\tau_k} &= F_0 + \sum_{i=1}^{2\nu} \alpha_i(h_k,\tau_k)F_i, \\ \tilde{B}_{h_k,\tau_k} &= G_0 + \sum_{i=1}^{2\nu} \alpha_i(h_k,\tau_k)G_i. \end{aligned} \tag{8}$$

In (8), 2ν is the number of the functions $\alpha_i(\cdot,\cdot)$ due to the two time-varying parameters h_k and τ_k, with

$\nu \leq n_x$. We have $\nu = n_x$ when each distinct eigenvalue of A corresponds to one Jordan block only, and $\nu < n_x$ otherwise. The functions $\alpha_i(h_k, \tau_k)$ are typically of the form $h_k^{j-1} e^{\lambda h_k}$ or $(h_k - \tau_k)^{j-1} e^{\lambda(h_k-\tau_k)}$, $j = 1, 2, \ldots, r$, if λ is a real, nonzero eigenvalue of A, and h_k^{j-1} or $(h_k - \tau_k)^{j-1}$, $j = 2, 3, \ldots, r+1$, if $\lambda = 0$. When λ corresponds to a pair of complex conjugate eigenvalues ($\lambda = a \pm b\sqrt{-1}$) of A, the functions $\alpha_i(h_k, \tau_k)$ take the form $h_k^{j-1} e^{a h_k} \cos(b h_k)$, $h_k^{j-1} e^{a h_k} \sin(b h_k)$, $(h_k - \tau_k)^{j-1} e^{a(h_k-\tau_k)} \cos(b(h_k - \tau_k))$ or $(h_k - \tau_k)^{j-1} e^{a(h_k-\tau_k)} \sin(b(h_k - \tau_k))$, $j = 1, 2, \ldots, r$, where r is the size of the largest Jordan block corresponding to λ.

Now using the assumption that the sampling intervals and delays are bounded and contained in the set Θ, as in (4), we obtain the following set of pairs of matrices

$$\mathcal{F} \; = \; \left\{ \left(\tilde{A}_{h_k,\tau_k},\, \tilde{B}_{h_k,\tau_k} \right) \mid (h_k, \tau_k) \in \Theta \right\}$$

that contains all possible matrix combinations in (7). The set $\mathcal{F}$ is still not a finite set, due to the infinite number of values that (h_k, τ_k) can take. Hence, we compute a convex overapproximation of the set $\mathcal{F}$ in the form of a convex matrix polytope, i.e., of the convex hull of a finite number of vertex matrices. Contrarily to Cloosterman et al. (2009), we will not compute a single overapproximation intended to be valid for all $(h_k, \tau_k) \in \Theta$, as this would remove all information about the probability distribution of h_k and τ_k. Instead, following the approach presented in Donkers et al. (2010), we partition the set Θ in polygons $\theta_m \subseteq \Theta$, $m \in \{1, 2, \ldots, S\}$, assign a probability $\tilde{p}_m = \iint_{\theta_m} p(h, \tau) dh d\tau$ to each polygon, and make for every θ_m a different overapproximation of the pair $\left(\tilde{A}_{h_k,\tau_k},\, \tilde{B}_{h_k,\tau_k}\right)$.

Let $\theta_1, \ldots, \theta_S$ be a collection of polygons satisfying

$$\cup_{m=1}^{S} \theta_m = \Theta,\ \mathrm{int}\theta_i \neq \emptyset,\ \mathrm{int}\theta_i \cap \mathrm{int}\theta_j = \emptyset, \tag{9}$$

for all $i, j \in \{1, 2, \ldots, S\}$ and $j \neq i$. Then, we have $\mathcal{F} = \cup_{m=1}^{S} \mathcal{F}_m$, where

$$\mathcal{F}_m \; = \; \left\{ \left(\tilde{A}_{h_k,\tau_k},\, \tilde{B}_{h_k,\tau_k} \right) \mid (h_k, \tau_k) \in \theta_m \right\}. \tag{10}$$

The minimal and maximal values of all functions α_i over every polygon θ_m can be computed as

$$\underline{\alpha}_{i,m} = \inf_{(h,\tau)\in\theta_m} \alpha_i(h, \tau), \qquad \overline{\alpha}_{i,m} = \sup_{(h,\tau)\in\theta_m} \alpha_i(h, \tau),$$

for all $i \in \{1, 2, \ldots, 2\nu\}$ and $m \in \{1, 2, \ldots, S\}$. Since each $\alpha_i(h, \tau) \in [\underline{\alpha}_{i,m}, \overline{\alpha}_{i,m}]$ for all $(h, \tau) \in \theta_m$, the sets of matrices $\mathcal{F}_m$ can be individually overapproximated by $\mathrm{co}\{\mathcal{H}_m\}$, i.e.,

$$\mathcal{F}_m \subseteq \mathrm{co}\{\mathcal{H}_m\},\ m = 1, 2, \ldots, S, \tag{11}$$

where

$$\mathcal{H}_m = \left\{ \left(F_0 + \sum_{i=1}^{2\nu} \alpha_i F_i,\ G_0 + \sum_{i=1}^{2\nu} \alpha_i G_i \right) \right. \\ \left. \mid\ \alpha_i \in \{\underline{\alpha}_{i,m}, \overline{\alpha}_{i,m}\},\ i = 1, 2, \ldots, 2\nu \right\},$$

and thus it also holds that $\mathcal{F} \subseteq \cup_{m=1}^{S} \mathrm{co}\{\mathcal{H}_m\}$. For enumeration purposes we also write

$$\mathcal{H}_m = \left\{ (H_{F,m,j},\, H_{G,m,j}) \ \mid\ j = 1, 2, \ldots, 2^{2\nu} \right\}. \tag{12}$$

Moreover, we define the set of all possible combinations of S elements, obtained taking one element from each of the sets $\mathcal{H}_m$, $m \in \{1, 2, \ldots, S\}$, as $\mathcal{V} = \mathcal{H}_1 \times \mathcal{H}_2 \times \ldots \times \mathcal{H}_S$. We will also write $\mathcal{V}$ as

$$\mathcal{V} = \{((V_{F,1,j},\, V_{G,1,j}),\ (V_{F,2,j},\, V_{G,2,j}),\ \ldots, \\ (V_{F,S,j},\, V_{G,S,j}))\ \mid\ j = 1, 2, \ldots, 2^{2\nu S}\}. \tag{13}$$

Remark 1. In the special case that there exists h_{nom} such that $p(h, \tau) = 0$ for all $h \neq h_{\mathrm{nom}}$, i.e., the sampling interval is constant, the proposed overapproximation procedure has to be slightly modified. This is because we proposed to form polygons $\theta_m \subseteq \Theta \subset \mathbb{R}^2$, $m \in \{1, \ldots, S\}$, having the property that $\mathrm{int}\mathcal{S}_m \neq \emptyset$, which is not useful anymore. In this case, we propose to form line segments θ_m defined as $\theta_m = \mathrm{co}\{(h_{\mathrm{nom}}, \tilde{\tau}_{m,1}),\, (h_{\mathrm{nom}}, \tilde{\tau}_{m,2})\}$, for each $m \in \{1, \ldots, S\}$, where $(h_{\mathrm{nom}}, \tilde{\tau}_{m,l})$, $l \in \{1, 2\}$, denote the vertices of the line segment θ_m. All other properties of θ_m, $m \in \{1, \ldots, S\}$ still hold and the remainder of the procedure can be applied *mutatis mutandis.* Note that in this case the number of vertices in (12) is 2^{ν}. A similar adjustment is needed where there exists τ_{nom} such that $p(h, \tau) = 0$ for all $\tau \neq \tau_{\mathrm{nom}}$, i.e., the delay is constant.

4. STOCHASTIC MPC DESIGN

The overapproximation described in Section 3 is used here to design a SMPC controller that exploits the measurements received at every time step to improve closed-loop performance, while guaranteeing stability. This control policy is derived from the approach presented by Bernardini and Bemporad (2009), and relies on a decoupling between stability enforcement and performance optimization. Offline, a Lyapunov function and a feedback control law which provide mean-square stability are obtained by exploiting the NCS convex overapproximation. Online, a stochastic MPC controller based on scenario enumeration is applied to optimize the performance by relying on the current state measurements and on the available stochastic information on the network uncertainty, while retaining stability.

4.1 Lyapunov function synthesis

Our first goal is to compute a Lyapunov function and a control law which render the closed-loop NCS system UGMSES. Here we consider quadratic Lyapunov functions of the form $V(\xi_k) = \xi_k^{\mathsf{T}} P \xi_k$, and assume that the control law is given by a constant matrix gain K, i.e., $u_k = K\xi_k$, for all k. The Lyapunov matrix P will then serve to enforce a stability constraint in the online control problem, while the existence of the gain K will be used to prove the recursive feasibility of the receding horizon policy.

Theorem 1. Suppose there exist polygons $\theta_1, \theta_2, \ldots, \theta_S$ satisfying (9), and an overapproximation of the NCS model (5) defined by the set of vertices $\mathcal{V}$ as in (13) such that (11) holds. Assume that the matrices $Q \in \mathbb{R}^{n_\xi \times n_\xi}$, $W \in \mathbb{R}^{n_\xi \times n_\xi}$, $Y \in \mathbb{R}^{n_u \times n_\xi}$, such that $Q = Q^{\mathsf{T}} \succ 0$, $W = W^{\mathsf{T}} \succ 0$, are given by the solution of the semidefinite programming problem

$$\min_{Q,W,Y}\ \mathrm{trace}(W) \tag{14a}$$

$$\text{s.t. } \mathrm{trace}(Q) = 1 \tag{14b}$$

$$W \succeq W_0 \tag{14c}$$

$$\begin{bmatrix} Q & Q & M_j^{\mathsf{T}} \\ Q & W & 0 \\ M_j & 0 & \tilde{Q} \end{bmatrix} \succeq 0,\ \forall j \in \{1, 2, \ldots, 2^{2\nu S}\}, \tag{14d}$$

where

$$M_j = \begin{bmatrix} \sqrt{\tilde{p}_1}(V_{F,1,j}Q + V_{G,1,j}Y) \\ \sqrt{\tilde{p}_2}(V_{F,2,j}Q + V_{G,2,j}Y) \\ \vdots \\ \sqrt{\tilde{p}_S}(V_{F,S,j}Q + V_{G,S,j}Y) \end{bmatrix},$$

$\tilde{p}_m = \iint_{\theta_m} p(h,\tau)dhd\tau$, $\tilde{Q} = \operatorname{diag}\{\underbrace{Q,\ldots,Q}_{S \text{ times}}\}$, and $W_0 \succ 0$.

Then, the closed-loop NCS (5) with $u_k = K\xi_k$ and $K = YQ^{-1}$ is UGMSES.

Proof. We will show that $V(\xi_k) = \xi_k^\mathsf{T} P \xi_k$, $P = Q^{-1}$, is a Lyapunov function for system (5) with $u_k = K\xi_k$. Using (7) and letting $C_{h,\tau} = \tilde{A}_{h,\tau} + \tilde{B}_{h,\tau}K$, we have that

$$\begin{aligned} \mathbb{E}\left[V(\xi_{k+1})\right] &= \mathbb{E}\left[\xi_k^\mathsf{T} C_{h_k,\tau_k}^\mathsf{T} P C_{h_k,\tau_k} \xi_k\right] \\ &= \iint_\Theta \xi_k^\mathsf{T} C_{h_k,\tau_k}^\mathsf{T} P C_{h_k,\tau_k}^\mathsf{T} \xi_k p(h_k,\tau_k) dh_k d\tau_k \\ &\leq \sum_{m=1}^{S} \tilde{p}_m \max_{(h_k,\tau_k)\in\theta_m} \xi_k^\mathsf{T} C_{h_k,\tau_k}^\mathsf{T} P C_{h_k,\tau_k} \xi_k. \quad (15) \end{aligned}$$

According to Lemma 1 in Morozan (1983), UGMSES is implied by requiring that, for some $L = L^\mathsf{T} \succ 0$,

$$\mathbb{E}[V(\xi_{k+1})] - V(\xi_k) \leq -\xi_k^\mathsf{T} L \xi_k,$$

for all $k \in \mathbb{N}$, which, given (15) is satisfied, holds when

$$P - L - \sum_{m=1}^{S} \tilde{p}_m C_{h_m,\tau_m}^\mathsf{T} P C_{h_m,\tau_m} \succeq 0 \qquad (16)$$

for all $(h_m, \tau_m) \in \theta_m$, $m = 1,2,\ldots,S$. Since (16) still yields an infinite number of LMIs (due to the fact that (h_m,τ_m) can take an infinite number of values), we use the convex overapproximation of (5) and the collections of pairs of matrices $\mathcal{H}_m$, $m = 1,2,\ldots,S$, that satisfy (11). Hence, C_{h_m,τ_m} in (16) for $(h_m,\tau_m) \in \theta_m$ can be written as $C_{h_m,\tau_m} = \sum_{j=1}^{2^{2\nu}} \lambda_{m,j}(H_{F,m,j} + H_{G,m,j}K)$, for some $\lambda_{m,j} \geq 0$, $j = 1,2,\ldots,2^{2\nu}$, with $\sum_{j=1}^{2^{2\nu}} \lambda_{m,j} = 1$. Therefore, by convexity we have that (16) is satisfied if

$$P - L - \sum_{m=1}^{S} \tilde{p}_m \tilde{C}_{m,j}^\mathsf{T} P \tilde{C}_{m,j} \succeq 0, \qquad (17)$$

for all $j \in \{1,2,\ldots,2^{2\nu S}\}$, where $\tilde{C}_{m,j} = V_{F,m,j} + V_{G,m,j}K$. By substituting $P = Q^{-1}$, $L = W^{-1} \succ 0$, $K = YQ^{-1}$, pre- and post-multiplying by Q, and taking a Schur complement, we have that (17) is equivalent to (14d). Hence, the solution of (14) satisfies (16), and the closed-loop system is UGMSES. □

4.2 NCS prediction model

Although for (mean-square) stabilization purposes one could just apply the constant state-feedback control law $u_k = K\xi_k$, $\forall k \in \mathbb{N}$, we want to design a SMPC controller based on an approximated model of the NCS dynamics (5) to also optimize a certain performance criterion. We introduce a new set of s polygons $\phi_1, \phi_2, \ldots, \phi_s$, which partition the set Θ such that properties analogous to (9) hold. Then, as prediction model we use the collection of the averaged dynamics of the NCS model for every polygon ϕ_n, i.e., the switching linear system defined as

$$\xi_{k+1} = \begin{cases} \bar{A}_1\xi_k + \bar{B}_1 u_k & \text{if } (h_k,\tau_k) \in \phi_1, \\ \bar{A}_2\xi_k + \bar{B}_2 u_k & \text{if } (h_k,\tau_k) \in \phi_2, \\ \quad\vdots & \quad\vdots \\ \bar{A}_s\xi_k + \bar{B}_s u_k & \text{if } (h_k,\tau_k) \in \phi_s, \end{cases} \qquad (18)$$

where

$$\bar{A}_n = \iint_{\phi_n} \tilde{A}_{h,\tau} p(h,\tau) dhd\tau, \quad \bar{B}_n = \iint_{\phi_n} \tilde{B}_{h,\tau} p(h,\tau) dhd\tau,$$

for all $n \in \{1,2,\ldots,s\}$, with $\tilde{A}_{h,\tau}$ and $\tilde{B}_{h,\tau}$ as in (5). Since we assumed that (h,τ) is given by an IID random process, the realization probabilities of every dynamical mode of (18) are taken to be $\bar{p}_n = \iint_{\phi_n} p(h,\tau)dhd\tau$, for all $n \in \{1,2,\ldots,s\}$ and $k \in \mathbb{N}$. As model (18) will only be used to improve closed-loop performance w.r.t. the constant state-feedback $u_k = K\xi_k$, the accuracy of the (MPC) prediction model will not affect stability. The use of a different partition $\phi_1, \phi_2, \ldots, \phi_s$ of the set Θ for prediction purposes has the main goal to increase the decoupling of performance optimization from stability properties, which are solely based on the overapproximation computed over the polygons $\theta_1, \theta_2, \ldots, \theta_S$. Further details on the partitions tuning are given in Section 5.

4.3 Optimization tree design

The formulation of the online SMPC control problem is based on a maximum likelihood approach, where at every time step k an optimization tree is built using the updated information on the augmented system state ξ_k. Each node of the tree represents a future state which is taken into account in the optimization problem. Starting from the root node, which is defined by the current available measurement $\xi_k = \left[x_k^\mathsf{T}\ u_{k-1}^\mathsf{T}\right]^\mathsf{T}$, a list of candidate nodes is generated by considering all the s possible dynamics in (18) and their probabilities $\bar{p}_n$, $n = 1,2,\ldots,s$. Then, the node with maximum probability is added to the tree. This procedure is repeated until a desired number of nodes n_{max} is reached: at every iteration new candidates are generated as children nodes of the last node added to the tree, and the one with the biggest realization probability is selected (these realization probabilities are formally defined in the following). Hence, every node is identified by a distinct trajectory of the network uncertain parameters (h,τ), and by a distinct input sequence, which is a variable of the optimization problem. This procedure leads to a "multiple-horizon" control problem, where different tree paths have in general different prediction horizons. Causality of the resulting control law is enforced by allowing one, and only one, control move for every node, except leaf nodes (i.e., nodes with no successor). Moreover, since the tree structure depends only on the distribution $\bar{p}_n$, $n = 1,2,\ldots,s$, it can be computed off-line, thus keeping the computational burden low. More details on the tree design procedure can be found in (Bernardini and Bemporad, 2009) and are omitted here for space reasons.

4.4 Control problem formulation

The objective function to be minimized in the proposed SMPC problem is an approximation of the expected value of the finite-horizon closed-loop performance

$$\mathbb{E}\Big[\sum_{j=0}^{N-1}\left(\xi_{k+j}^{\mathsf{T}}Q_\xi\xi_{k+j}+u_{k+j}^{\mathsf{T}}Q_u u_{k+j}\right)+\xi_{k+N}^{\mathsf{T}}Q_\xi\xi_{k+N}\Big] \tag{19}$$

for a given horizon $N > 0$ and weight matrices Q_ξ, Q_u. In order to define the stochastic optimal control problem associated with the SMPC policy, let us introduce the following quantities:

- $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n\}$: the set of the tree nodes. Nodes are indexed progressively as they are added to the tree (i.e., $\mathcal{T}_1$ is the root node and $\mathcal{T}_n$ is the last node added).
- $\xi_{\mathcal{N}}$, $u_{\mathcal{N}}$: the state and the input, respectively, associated with node $\mathcal{N}$.
- $pre(\mathcal{N})$: the predecessor of node $\mathcal{N}$.
- $succ(\mathcal{N}, j)$: the successor of node $\mathcal{N}$ generated with dynamics of mode j in (18), $j \in \{1, 2, \ldots, s\}$.
- $\delta(\mathcal{N}) \in \{1, 2, \ldots, s\}$: the mode leading to node $\mathcal{N}$.
- $\pi_{\mathcal{N}}$: the realization probability of node $\mathcal{N}$, i.e., the probability of reaching node $\mathcal{N}$ from $\mathcal{T}_1$, recursively computed as $\pi_{succ(\mathcal{N},j)} = \bar{p}_j\pi_{\mathcal{N}}$, with $\pi_{\mathcal{T}_1} = 1$.
- $\mathcal{S} \subset \mathcal{T}$: the set of the leaf nodes, defined as $\mathcal{S} \triangleq \{\mathcal{T}_i \in \mathcal{T} \mid succ(\mathcal{T}_i, j) \notin \mathcal{T},\ j = 1, 2, \ldots, s\}$.

With a slight abuse of notation, in the following the abbreviate forms ξ_i, u_i, π_i, $\delta(i)$, $pre(i)$, will be used to denote $\xi_{\mathcal{T}_i}$, $u_{\mathcal{T}_i}$, $\pi_{\mathcal{T}_i}$, $\delta(\mathcal{T}_i)$, $pre(\mathcal{T}_i)$, respectively. The SMPC problem at the time step k is formulated as

$$\min_{\{u_i\}} \sum_{i\in\mathcal{T}\setminus\mathcal{T}_1}\pi_i\xi_i^{\mathsf{T}}Q_\xi\xi_i + \sum_{j\in\mathcal{T}\setminus\mathcal{S}}\pi_j u_j^{\mathsf{T}}Q_u u_j \tag{20a}$$

$$\text{s.t. } \xi_1 = \xi_k \tag{20b}$$

$$\xi_i = \bar{A}_{\delta(i)}\xi_{pre(i)} + \bar{B}_{\delta(i)}u_{pre(i)},\ \forall i \in \mathcal{T}\setminus\{\mathcal{T}_1\} \tag{20c}$$

$$\sum_{m=1}^{S}\tilde{p}_m G_{m,j}^{\mathsf{T}}PG_{m,j} \le \xi_1^{\mathsf{T}}(P-L)\xi_1, \quad \forall j \in \{1, 2, \ldots, 2^{2\nu S}\}, \tag{20d}$$

where $G_{m,j} = V_{F,m,j}\xi_1 + V_{G,m,j}u_1$. Note that (20a) tends to (19) with $N \to \infty$ if $n_{max} \to \infty$. Hence, the expected value of the closed-loop performance (19) can be approximated with arbitrary accuracy at the expense of a higher computational load. Problem (20) is a quadratically constrained quadratic problem (QCQP). Provided that (14) has solution, this problem is always feasible, as shown below.

Theorem 2. Suppose there exist polygons $\theta_1, \theta_2, \ldots, \theta_S$ and $\phi_1, \phi_2, \ldots, \phi_s$ satisfying (9), and an overapproximation of the NCS model (5) defined by the set of vertices $\mathcal{V}$ as in (13) such that (11) holds. Assume that the matrices Q, W are given from the solution of (14). Then, the closed-loop NCS (5) where $u_k = u_{\mathcal{T}_1}$ and $u_{\mathcal{T}_1}$ is given by the receding horizon solution of (20), with $P = Q^{-1}$ and $L = W^{-1}$, is UGMSES.

Proof. By similar reasonings as in Theorem 1, we have that mean-square stability is provided by the receding-horizon satisfaction of condition (20d), which now depends explicitly on the measured state ξ_k and on the decision variable u_k. We only need to show that the control problem (20) is recursively feasible at every time step. This follows by noting that (14d) implies (20d) if a state-feedback structure is imposed on the input u_k. Hence, $u_i = K\xi_i$, $\forall i \in \mathcal{T}\setminus\mathcal{S}$, is always a feasible solution for (20), where $K = YQ^{-1}$ is obtained by solving (14). □

Table 1. Simulation results

Controller	$\mu(J_i)$	$\sigma(J_i)$
Robust state-feedback	884.34	382.19
Stochastic MPC	678.01	134.74

5. ILLUSTRATIVE EXAMPLE

In this section we test the performance of the proposed approach using a numerical example, where the controlled plant is modeled by the second-order continuous-time linear system (1), with $A = \left[\begin{smallmatrix}1 & 15\\ -15 & 1\end{smallmatrix}\right]$ and $B = \left[\begin{smallmatrix}0.2\\ 0.8\end{smallmatrix}\right]$. This system is open-loop unstable and has two complex eigenvalues. In order to define the network model, we assume that the sampling interval h_k is constant and equal to $h_{\text{nom}} = 0.1$, i.e., $p(h, \tau) = 0$ for all $h \neq h_{\text{nom}}$. We define the set Θ in (4) as

$$\Theta = \left\{(h, \tau) \in \mathbb{R}^2 \mid h = h_{\text{nom}} \wedge \tau \in [0.02,\ 0.1]\right\}. \tag{21}$$

Moreover, we assume that the PDF modeling the realizations of the delay τ_k is given by a truncated (and normalized) normal distribution with mean $\mu = 0.04$ and standard deviation $\sigma = 0.012$.

In order to satisfy the conditions of Theorem 1, we first compute an overapproximation of the NCS as described in Section 3. Then, according to Remark 1, we construct $S = 4$ line segments to partition the set of possible values of τ_k, defined as $\theta_1 = \{h_{\text{nom}}\} \times [0.02,\ 0.033]$, $\theta_2 = \{h_{\text{nom}}\} \times [0.033,\ 0.046]$, $\theta_3 = \{h_{\text{nom}}\} \times [0.046,\ 0.06]$, and $\theta_4 = \{h_{\text{nom}}\} \times [0.06,\ 0.1]$. This allows us to find a feasible solution of problem (14), and to obtain a stabilizing controller of the form $u_k = K\xi_k$ for the closed-loop system (5). With the aim at improving the performance of the SMPC controller, we perform a finer partition for prediction purposes, using s=8 line segments defined as $\phi_n = \{h_{\text{nom}}\} \times [0.02 + 0.008(n-1),\ 0.02 + 0.008n]$, $n = 1, 2, \ldots, 7$, and $\phi_8 = \{h_{\text{nom}}\} \times [0.076,\ 0.1]$. The weight matrices in problem (20) are set as $Q_\xi = \text{diag}\{1,\ 10,\ 10^{-3}\}$, $Q_u = 10^{-3}$, and a number of nodes $n_{max} = 15$ is used to design the optimization tree.

A set of $N_s = 100$ simulations was run of $T_s = 15$ time steps each, with random initial state, comparing the proposed SMPC control scheme with a constant state-feedback controller which provides robust convergence to the origin. Such a deterministic controller can be obtained as a special case of the stochastic one, by solving problem (14) with $S = 1$ and $\theta_1 = \Theta$. Since a feasible solution could not be found with $(h_k, \tau_k) \in \Theta$ as in (21), we restricted to consider $\tau_k \in [0.02,\ 0.09]$ when solving the robust control synthesis problem.

To evaluate the performance achieved by the considered controllers, we define the experimental cost function

$$J_i = \sum_{k=1}^{T_s}\left(\xi_{k,i}^{\mathsf{T}}Q_\xi\xi_{k,i} + u_{k,i}^{\mathsf{T}}Q_u u_{k,i}\right),$$

where $i \in \{1, 2, \ldots, N_s\}$ indexes the values related to the ith simulation. Table 1 shows numerical results in terms of mean $\mu(J_i)$ and standard deviation $\sigma(J_i)$ of the experimental cost function J_i over all the simulations. A comparison between the different closed-loop trajectories

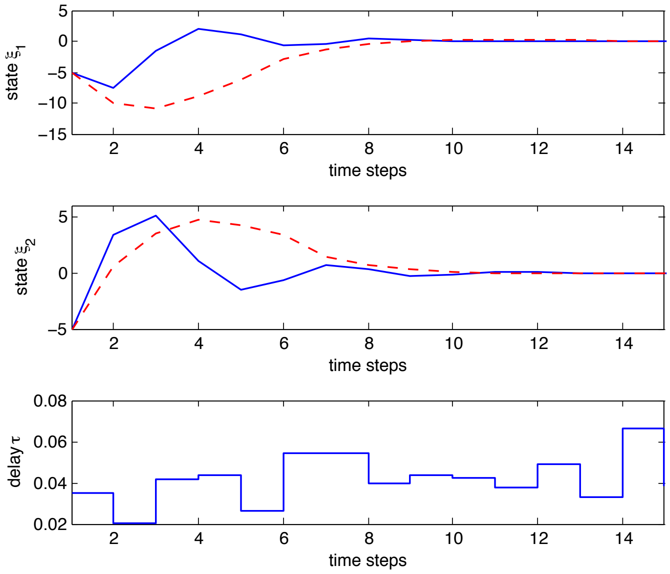

Fig. 2. Example of trajectories obtained with SMPC (solid line) and robust state-feedback (dashed line)

is shown in Fig. 2. As we can see from the results, the proposed SMPC policy provides an improved control action with respect to the robust controller. From a computational point of view, the CPU time needed to solve an instance of the SMPC online control problem on a 2.4GHz MacBook running Matlab 7.8 and Cplex 11 was 29 ms on average, with a maximum value of 41 ms.

In real applications several parameters can be tuned to make the trade-off between the desired closed-loop performance and the complexity of the resulting on-line control problem. As long as problem (14) remains feasible, the partition $\{\theta_m\}_{m=1}^{S}$ can be made coarser to decrease the number of quadratic constraints to be imposed online. Independently, the partition $\{\phi_n\}_{n=1}^{s}$ can be refined to improve the approximation of the continuous distribution $p(h, \tau)$ by the discretization $\Pr[(h, \tau) \in \phi_n] = \bar{p}_n$, $n = 1, 2, \ldots, s$, and the number of tree nodes n_{max} can be increased to have a more accurate prediction model, and thus better performances.

6. CONCLUSIONS

In this paper we presented a stochastic model predictive control approach for networked control systems that are subject to time-varying sampling intervals and time-varying delays. These uncertain parameters are assumed to be bounded, but modeled by a continuous PDF. The proposed control policy relies on a stochastic control Lyapunov function approach and consists of two steps. Offline, a Lyapunov function which provides mean-square stability is obtained by computing a discrete approximation of the continuous PDF, constructing a convex overapproximation of the NCS model, and solving an SDP problem. Online, a SMPC formulation based on scenario enumeration optimizes a quadratic performance by exploiting the current measurements and the stochastic information on the uncertain parameters, while retaining stability. The complexity of the proposed receding horizon control problem may grow with the number of partitions in which the set Θ is divided. However, an opportune design of the partitions can minimize the number of constraints to be imposed online. Moreover, the computational load could be substantially reduced by solving the SMPC control problem explicitly using multiparametric programming techniques, which is a current topic of research investigations. Future work will also include the integration of state and input constraints in the control problem formulation, which was not pursued in this paper for space limitations.

REFERENCES

Antsaklis, P. and Baillieul, J. (2007). Special issue on technology of networked control systems. *IEEE Proc.*, 95(1), 5–8.

Antunes, D., Hespanha, J., and Silvestre, C. (2009). Stability of impulsive systems driven by renewal processes. In *Proc. American Control Conf.*, 4032–4037. St. Louis, MO.

Bemporad, A., Heemels, W., and Johansson, M. (eds.) (2010). *Networked Control Systems*. Springer-Verlag. `http://ist-wide.dii.unisi.it/school09`.

Bernardini, D. and Bemporad, A. (2009). Scenario-based model predictive control of stochastic constrained linear systems. In *Proc. 48th IEEE Conf. on Decision and Control*, 6333–6338. Shanghai, China.

Cloosterman, M., van de Wouw, N., Heemels, W., and Nijmeijer, H. (2009). Stabilization of networked control systems with uncertain time-varying delays. *IEEE Trans. Autom. Control*, 1575–1580.

Couchman, P., Cannon, M., and Kouvaritakis, B. (2006). Stochastic MPC with inequality stability constraints. *Automatica*, 42, 2169–2174.

Donkers, M., Heemels, W., Bernardini, D., Bemporad, A., and Shneer, V. (2010). Stability analysis of stochastic networked control systems. In *Proc. American Control Conf.* Baltimore, MD.

Heemels, W., van de Wouw, N., Gielen, R., Donkers, M., Hetel, L., Olaru, S., Lazar, M., Daafouz, J., and Niculescu, S.I. (2010). Comparison of overapproximation methods for stability analysis of networked control systems. In *Hybrid Systems: Computation and Control*, 181–190. Stockholm, Sweden.

Hespanha, J., Naghshtabrizi, P., and Xu, Y. (2007). A survey of recent results in networked control systems. *IEEE Proc.*, 138–162.

Montestruque, L. and Antsaklis, P. (2004). Stability of model-based networked control systems with time-varying transmission times. *IEEE Trans. Autom. Control*, 1562–1572.

Morozan, T. (1983). Stabilization of some stochastic discrete-time control-systems. *Stochastic Analysis and Applications*, 1(1), 89–116.

Primbs, J. (2007). Stochastic receding horizon control of constrained linear systems with state and control multiplicative noise. In *Proc. American Control Conf.*, 4470–4475. New York, NY.

Seiler, P. and Sengupta, R. (2005). An $\mathcal{H}_\infty$ approach to networked control. *IEEE Trans. Autom. Control*, 356–364.

Zhang, W., Branicky, M.S., and Phillips, S.M. (2001). Stability of networked control systems. *IEEE Control Systems Magazine*, 21(1), 84–99.