Collaborative Project
Small-medium-scale focused research project (STREP)

Grant Agreement n. 224117 FP7-ICT-2007-2

WIDE

Decentralized Wireless Control of Large-Scale Systems

Starting date: 01 September 2008 Duration: 3 years

Deliverable number	**D4.2**
Title	**Synthesis methods for distributed MPC control and estimation algorithms using WSN**
Work package	WP4 - Network-aware control and estimation (RTD)
Due date	M24
Actual submission date	01/09/2010
Lead contractor for this deliverable	TUE
Author(s)	N.W. Bauer <n.w.bauer@tue.nl>, W.M.P.H. Heemels <w.m.p.h.heemels@tue.nl>
With the help of	A. Bemporad <bemporad@ing.unitn.it>, D. Barcelli <barcelli@dii.unisi.it> D. Bernadrini <bernardini@dii.unisi.it> M.C.F. Donkers <m.c.f.donkers@tue.nl> N. van de Wouw <n.v.d.wouw@tue.nl> M.B.G. Cloosterman <marieke.posthumus@yacht.nl> L. Hetel <larentiu.hetel@ec-lille.fr> J. Daafouz <jamal.daafouz@ensem.inplnancy.fr> J. Sijs <joris.sijs@tno.nl> M. Lazar <m.lazar@tue.nl> L. Baramov <Lubomir.Baramov@honeywell.com> D. Pachner <Daniel.Pachner@honeywell.com> V. Havlena <Vladimir.Havlena@Honeywell.com>
Revision	v1.0 (September 3, 2010)

Dissemination Level

PU	Public
PP	Restricted to other programme participants (including the Commission Services)
RE	Restricted to a group specified by the consortium (including the Commission Services)
→ **CO**	Confidential, only for members of the consortium (including the Commission Services)

Executive summary

Report describing the developed design methods for net-aware centralized (MPC) controllers and estimators, with the integrated distributed MPC and estimation algorithms of WP3 and the result of the benchmark example. Robustness with respect to network unreliability in the form of uncertain communication channels and aspects of reduction of the usage of network resources (e.g. battery energy) will be taken into account.

Contents

1 Introduction

This report describes the achievements of WP4 in the area of network-aware controller design. Following this short introduction, the report has 4 main topics:

- Network-Aware State-Feedback Control
- Network-Aware Estimation
- Network-Aware Observer-Based Control
- Benchmark Results.

We will shortly introduce the main contributions in each of these individual parts below.

Network-Aware State-Feedback Control

Section 2.1 presents a discrete-time model for networked control systems (NCSs) that incorporates the network-induced phenomena: time-varying sampling intervals, packet dropouts and time-varying delays that may be both smaller and larger than the sampling interval. Based on this model, constructive LMI conditions for controller synthesis are derived, such that stabilizing state-feedback controllers can be designed. Besides the proposed controller synthesis conditions a comparison is made between the use of Parameter-Dependent Lyapunov functions and Lyapunov-Krasovskii functions for stability analysis. It is concluded that the newly developed approach based on Parameter-Dependent Lyapunov functions is always less conservative than the existing approaches based on classical Lyapunov-Krasovskii functions.

Section 2.2 presents an approach for synthesizing a set of decentralized regulators for discrete-time linear systems subject to input and state constraints. Measurements and command signals are exchanged among a distributed set of sensors and actuators on a network where some links are subject to packet dropout. The resulting closed-loop system is guaranteed to asymptotically reach the origin, even if every local actuator can exploit only a (possibly time-varying) subset of state measurements. A model of packet dropout based on a finite-state Markov chain is optionally introduced to exploit available knowledge about the stochastic nature of the network. In this framework, an approach to the synthesis of decentralized switching linear controllers is presented to guarantee mean-square stability of the overall controlled process under packet dropout and soft input and state constraints. The proposed control techniques are compared with standard centralized linear controllers and simulation results are reported.

Section 2.3 addresses the fact that new control design methodologies are needed to address the energy-constrained nature of Wireless Sensor Networks (WSNs); in particular the discharge of batteries of sensor nodes, which is mainly due to radio communications, must be taken into account. In this work a novel transmission strategy for communication between controller and sensors is presented which is intended to minimize the data exchange over the wireless channel. Moreover, an energy-aware control technique is presented for constrained linear systems based on explicit Model Predictive Control (MPC), providing closed-loop stability in the presence of disturbances. The presented control schemes are tested and compared to traditional MPC techniques. The results show the effectiveness of the proposed energy-aware approach, which achieves a profitable trade-off between energy savings and closed-loop performance.

Section 2.4 presents a decentralized model predictive control (DMPC) design approach for set-point tracking under input constraints and possible loss of information packets for large-scale processes whose dynamics can be represented as the interaction of several dynamically-coupled linear subsystems. Following earlier results in [100, 101], the global model of the process is approximated as the decomposition of several (possibly overlapping) smaller models used for local predictions. Sufficient criteria are presented for asymptotic tracking of output set-points and rejection of constant measured disturbances when the overall process is in closed loop with the set of decentralized MPC controllers, under possible intermittent lack of communication of measurement data between controllers.

Section 2.5 presents 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,a SMPC formulation is presented which is 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.

Network-Aware Estimation

Section 3.1 presents novel approaches to state estimation under communication delay. State estimates can be used for control, optimization, process monitoring, fault detection, or whenever states of the process are not accessible for direct measurement. The estimator uses the process output observations as well as process model to infer the internal process states. Measured process data are subject to noise corruption and process model is subject to uncertainty affecting the quality of the state estimate. The state estimation techniques will support other tasks, such as distributed MPC control, control over networks and real-time optimization as they often require information on the internal process state. A particular attention will be paid to reducing computational complexity by using gains pre-computed off-line and exploiting the particular problem structure. A reduced complexity representation for the optimal Kalman filter has been developed. Furthermore, a suboptimal approach was obtained which results in a particularly simple implementation that can be used in fast control loops in the basic control layer for estimating process controlled values purged from measurement noise and communication delays. Clearly, the obtained results contributes to integrating WSN into Distributed Control Systems (DCSs) without compromising stability.

Section 3.2 considers a similar setting as in the previous section – state estimation for a linear, time-invariant model disturbed by Gaussian noise, where measurements arrive with variable delay and possibly out-of-order. It is further assumed that all measurements arrive within a maximum delay. It was proven in our previous work in [53], under these conditions, that the Kalman filter converges to a parameter-varying system, where the Kalman gain depends only on the set of missing samples. However, it is shown that the number of gains that the control law requires grows exponentially with the maximum delay which may result in storage problems when considering large delays. It is possible to use the technique proposed in the previous section to re-calculate the gains at each step, however this results in an increased computational load. The proposed way to keep the filter very simple is to resort to a sub-optimal filter. It is based on the technique of replacing a sample that has not arrived in the due time by a ‘fake’ measurement and upon receiving the correct sample later, canceling the effect of the wrong data and fusing the new information correctly. This results in restoring optimality. A similar approach was proposed in [56] for time-varying systems with full non-steady Kalman filter. Here, a solution is presented for time-invariant systems using a switched linear filter with a moderate number of pre-computed gains.

Network-Aware Observer-Based Control

Section 4.1 provides an approach to analyze and design decentralized observer-based controllers for large-scale linear plants subject to network communication constraints and varying sampling intervals. Due to communication constraints, it is impossible to transmit all input and output data simultaneously over the communication network that connects sensors, actuators and controllers. A protocol orchestrates what data is sent over the network at each transmission instant. To handle these communication constraints, it is fruitful to adopt a switched observer structure that switches based on the transmitted information. By taking a discrete-time switched linear system perspective, a general model is derived that captures all these aspects and provides insight into how they influence each other. Focusing on the class of so-called ‘periodic protocols’ (of which the well-known Round Robin protocol is a special case), a method to assess robust stability using a polytopic over-approximation and LMI-based stability conditions is presented. Although the design problem is in general non-convex, a procedure to find stabilizing control laws by simplifying the control problem is presented. The design of the controller exploits the periodicity of protocols and ignores the global coupling between subsystems of the plant and variation of the sampling intervals. To assess the robust stability of the resulting closed-loop system including the ignored effects, an *a posteriori* analysis is conducted based on the derived LMIs.

Section 4.2 addresses recent progress in the field of event-based estimation. The main purpose of event-based control, if compared to periodic control, is to minimize data transfer or processing power in networked control systems. Currently existing methods have an (implicit) dependency between triggering the events and the control algorithm. To decouple these two, an event-based state estimator is introduced in between the sensor and the controller. The event-based estimator is used to obtain a state estimate with a bounded covariance matrix in the estimation error at every synchronous time instant, under the assumption that the set in the measurement-space that is used for event generation is bounded. The estimation error is then translated into explicit polytopic bounds that are fed into a robust MPC algorithm. The resulting MPC closed-loop system is proven to be input-to-state stable (ISS) to the estimation error. Moreover, whenever the network requirements are satisfied, the controller could explicitly request for an additional measurement in case there is a desire for a better disturbance rejection.

Benchmark Results

In this section the proposed decentralized model predictive control (DMPC) approach, as presented in Section 2.4, is tested on the passenger railcar benchmark example. The test investigates the convergence properties when dropouts are present and compares the DMPC solution to the centralized solution. To simulate packet loss in the network, it is assumed that the probability of losing a packet depends on the state of the Markov chain. Different simulation outcomes are investigated depending on four ingredients: i) type of controller (centralized/decentralized), ii) packet-loss probability, iii) change in reference values, iv) changes of external temperature (acting as a measured disturbance).

2 Network-Aware State-Feedback Control

2.1 Linear State Feedback

M.G.B. Cloosterman, L. Hetel, N. van de Wouw, W.M.P.H. Heemels, J. Daafouz, H. Nijmeijer

2.1.1 Introduction

The literature on modeling, analysis and controller design of networked control systems (NCSs) expanded rapidly over the last decade [6, 65, 159]. The use of networks offers many advantages such as low installation and maintenance costs, reduced system wiring (in the case of wireless networks) and increased flexibility of the system. However, from a control theory point of view, the presence of the network also introduces several disadvantages such as time-varying networked-induced delays, aperiodic sampling or packet dropouts. To understand the impact of these network effects on control performance several models have been developed. Roughly speaking, these NCS models can be categorized in continuous-time and discrete-time models. A further discrimination can be given on the basis of which network phenomena they include.

In the continuous-time domain, Fridman et al. [20] applied a descriptor system approach to model the sampled-data dynamics of systems with varying sampling intervals in terms of (infinite-dimensional) delay-differential equations (DDEs) and study their stability based on the Lyapunov-Krasovskii functional method. In [23,82,83], this approach is used for the stability analysis of NCSs with time-varying delays and constant sampling intervals, using (linear) matrix inequality-based techniques. The recent results in [23] also involve H_∞ controller designs based on linear matrix inequalities (LMIs). However, Mirkin [43] showed that the use of such an approach for digital control systems neglects the piecewise constant nature of the control signal due to the zero-order-hold mechanism and that it introduces conservatism when exploiting such modeling for stability analysis. More specifically, the conservatism is introduced by the fact that the zero-order hold and delay jointly introduce a particular linearly increasing time-varying delay within each control update interval (sometimes indicated by the sawtooth behavior of the delay), whereas in the modelling approach mentioned above it is replaced by an arbitrary bounded time-varying delay. Moreover, in [43] one indicated that less conservative stability conditions are obtained using a robust parametric modeling of the delay operator as proposed in [33]. An alternative approach, proposed in [47, 48], is based on impulsive delay differential equations and does take into account the piecewise constant nature of the control signal due to the zero-order-hold mechanism and has also been shown by [43] to be less conservative than the descriptor approach. As also noted in [48], the usage of infinite-dimensional DDE models and Lyapunov functionals to analyse the stability of essentially *finite-dimensional* sampled-data NCS does not seem to offer any advantage. The approach in [47] is able to deal simultaneously with time-varying delays and time-varying sampling intervals but does not explicitly include packet dropouts in the model (although they might be considered as variations in the sampling intervals or delays). Moreover, the stability analysis leads to bilinear matrix inequalities (BMIs), which are generally difficult to solve. As a consequence, for the moment no effective control synthesis results exist within this framework.

The majority of NCS models are discrete-time formulations based on the exact discretization of the continuous-time linear plant over a sample interval (see [12, 19, 22, 24, 40, 52, 57, 67, 76, 159] and the references therein). Such models avoid the problem of an infinite dimensional state that is encountered in the continuous-time (DDE) models due to delays. Moreover, in these discrete-time models the piecewise constant nature of the control signal due to the zero-order-hold is taking into account exactly. Additionally, it has been shown in [67], that for systems with aperiodic sampling and time-varying delay less than the sampling interval the use of discrete-time models for stability analysis gives less conservative characterization of stability than the use of (impulsive) delay differential equations. On the other hand [67, 68] shows that the modeling in terms of impulsive difference equations is favorable for ISS gain analysis for perturbed NCS. Under simplified assumptions, such that the delay is a multiple of the sampling interval or it takes values in a finite set, the obtained models lead to switched linear systems and corresponding stability conditions can be applied [40, 51, 78, 84]. However, these models are not so realistic as in practice one typically encounters an infinite number of possible values for the delay. Moreover, more realistic models should take into account that the sampling periods might be aperiodic.

For systems with time-varying sampling intervals, [19, 22, 57, 67] address the stability analysis and control design using a discrete-time model. In [19], discrete difference inclusions are obtained for the different values of the sampling interval and sufficient algebraic conditions for existence of quadratic Lyapunov function are derived based on the construction of a solvable Lie algebra. A different approach is given in [22, 57], where the authors used the gridding of the set of possible sampling intervals to derive LMI-based stability conditions.

Several discrete-time approaches have been proposed for dealing with network-induced delays. In this context, using the exact discretization over a sampling period, the obtained model is generally a difference equation with time-varying delays in the input and unknown time-varying system matrices. When the variation of the delay is smaller than the sampling period, the analysis/control design problems can be addressed by using a lifted state vector and robust control methods for parametric uncertainties [12, 24, 28] or by applying the Lyapunov-Krasovskii function (LKF) approach [54, 77, 80, 81] (for the LKF approach in discrete-time, see [21]). In this context, the main problems are the conservatism inherent to the use of upper boundings in the increment of the LKF and the reduced applicability of the results since they are able to deal only with delay variations smaller than the sampling interval. Generalizing such models to the case of large delay variations, packet dropouts and time-varying sampling intervals is not a trivial task.

In the literature, two ways of modelling network-induced uncertainties (such as time-varying delays and sampling intervals and packet dropouts) can be distinguished. Firstly, in [22, 24, 40, 47, 49, 159] and many other works, bounds are imposed on the delays, sampling intervals and the maximum number of subsequent dropouts. Secondly, in e.g. [26, 46, 58, 61], a stochastic modelling approach is adopted. In this paper, we will adopt the first approach. Given bounds on delays, sampling intervals and subsequent dropouts, we will formulate stability conditions and constructive controller synthesis results independent of the probability distribution of the uncertain variables. So, such *robust* results also apply in the stochastic setting and can be seen as ‘probability distribution-free’ results for the stochastic case if the domain of the probability distribution function is bounded.

In the current paper, we propose a discrete-time NCS model that can deal simultaneously with packet dropouts and time-varying delays smaller and larger than a possibly time-varying sampling interval. This model is obtained using the exact discretization over a sampling interval and it takes into account also the complicated case in which the delay variations may be larger than the sampling interval. Moreover, the possibility of packet dropouts is modeled explicitly. Based on this model, controller synthesis conditions in terms of LMIs will be derived, using both a common quadratic and a parameter-dependent Lyapunov approach. Note that recently, in [29], a simplified event-based discrete-time model has been proposed for taking into account the different implementation problems in digital control systems. This model is obtained using the systems representation at both sampling and actuation times. The advantage of the model presented in this paper in comparison to this event-based model is that it generally leads to a discrete-time representation of a smaller dimension. Moreover, it generalizes several of the models that exist in the literature to the case in which all the network effects appear simultaneously. This enables the theoretical comparison with the existing approaches. A discussion on the stability characterization based on LKFs and on parameter-dependent Lyapunov functions (PDLF) will be given. This discussion is inspired by the results in [30] in which a comparison between LKFs and Lyapunov functions for switched system is presented in the case of difference equations with time-varying delays in the state. The approach in [30] can deal only with delays that are a multiple of the sampling time, and therefore it does not apply to continuous-time systems as the NCS studied here. We show that the stability analysis based on the most general LFK of a quadratic type is always more conservative than the novel stability characterization presented here. This result applies to the context of NCS in which we are faced with an interaction between continuous-time systems and discrete-time controllers under different perturbing networked effects. In particular, we will show that the existence of general LKFs as used in the literature, implies also the existence of a Lyapunov function in our framework. It is important to note that all existing LKFs are a particular case of the one proposed in this paper and our approach allows much more freedom in the Lyapunov function than the typical LKFs adopted in the literature [54, 77, 80, 81], which have repetitions of terms. Stated differently, the Lyapunov function that we consider corresponds to a general LKF for which LMI-based stability conditions never appeared in the literature before. In addition, it can formally be proven that our approach is never more conservative than the LKF approach. Next to the stability characterization, we will also present LMI-based synthesis techniques for feedback based on the state vector and on an augmented state vector that includes old inputs next to the system state.

In summary, the main contributions of the paper are as follows. Firstly, a model for NCSs including three network-induced uncertainties (large delays, time-varying sampling intervals and packet dropouts) is developed. Moreover, we present a procedure for the over-approximation of this model to arrive at a polytopic model suitable for stability analysis and controller synthesis. Secondly, we present a stability characterisation for NCSs using (parameter-dependent) quadratic Lyapunov functions, which generalises stability characterisations using Lyapunov-Krasovskii functionals existing in the literature. Thirdly, next to LMI-based stability conditions, we provide a solution to the (structured) state feedback synthesis problem in terms of linear matrix inequalities for NCS models including all the above network-induced uncertainties.

This paper is structured as follows: In Section II we present our NCS model. Section III is dedicated to the theoretical comparison of stability characterizations. Section IV presents LMI control design methods that are illustrated by numerical examples in Section IV. Section V closes with concluding remarks.

2.1.2 NCS modeling

In this section, the discrete-time description of a NCS including delays larger than the uncertain, and time-varying sampling interval and packet dropouts is presented. The NCS is depicted schematically in Figure 1. It consists of a linear continuous-time plant

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

with $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$, and a discrete-time static time-invariant controller which are connected over a communication network that induces network delays (τ^{sc} and τ^{ca}). The state measurements ($y(t) = x(t)$) are sampled resulting in the sampling time instants s_k:

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

which are non-equidistantly spaced in time due to the time-varying sampling intervals $h_k > 0$. The sequence of sampling instants $s_0, s_1, s_2, \ldots$ is strictly increasing in the sense that $s_{k+1} > s_k$, for all $k \in \mathbb{N}$. We denote by $y_k := y(s_k)$ the k^{th} sampled value of y and by u_k the control value corresponding to $y_k = x_k$. Packet drops may occur (see Figure 1) and is modeled by the parameter m_k. This parameter denotes whether or not a packet is dropped:

$$m_k = \begin{cases} 0, & \text{if } y_k \text{ and } u_k \text{ are received} \\ 1, & \text{if } y_k \text{ and/or } u_k \text{ is lost.} \end{cases} \tag{2}$$

In (2), we make no distinction between packet dropouts that occur in the sensor-to-controller connection and the controller-to-actuator connection in the network. This can be justified by realizing that, for static controllers, the effect of the packet dropouts on the control updates implemented on the plant is the same in both cases. Indeed, for packet dropouts between the sensor and the controller no new control update is computed and thus no new control input is sent to the actuator. In the case of packet dropouts between the controller and the actuator no new control update is received by the

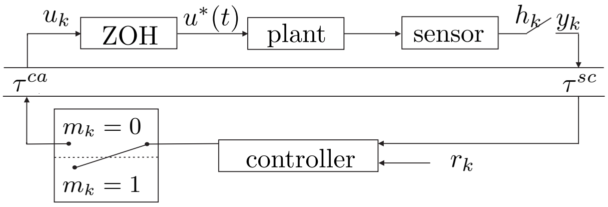


Figure 1: Schematic overview of the NCS with variable sampling intervals, network delays and packet dropouts

actuator either. Finally, the zero-order-hold (ZOH) function (in Figure 1) is applied to transform the discrete-time control input u_k to a continuous-time control input $u^*(t)$ being the actual actuation signal of the plant.

In the model, both the varying computation time (τ_k^c), needed to evaluate the controller, and the 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 (1)) and that both the controller and the actuator act in an event-driven fashion (i.e. responding instantaneously to newly arrived data). Furthermore, we consider that not all the data is used due to packet dropouts and message rejection, i.e. the effect that more recent control data is available before the older data is implemented and therefore the older data is neglected. 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 also [52], [159]. To include these effects in the continuous-time model, let us define the parameter $k^*(t)$ that denotes the index of the most recent control input that is available at time t as $k^*(t) := \max\{k \in \mathbb{N} | s_k + \tau_k \leq t \wedge m_k = 0\}$. The continuous-time model of the plant of the NCS is then given by:

$$\begin{array}{rcl} \dot{x}(t) & = & Ax(t) + Bu^*(t) \\ u^*(t) & = & u_{k^*(t)}, \end{array} \tag{3}$$

with $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$. Here, we assume that the most recent control input remains active in the plant if a packet is dropped.

We assume that the variation in the delays is bounded by $\tau_{\min}$ and $\tau_{\max}$, the variation in the sampling interval is bounded by $h_{\min}$ and $h_{\max}$ and that the number of subsequent packet dropouts is upper bounded by $\overline{\delta}$. The latter means that

$$\sum_{v=k-\overline{\delta}}^{k} m_v \leq \overline{\delta}. \tag{4}$$

as this guarantees that from the control inputs $u_{k-\overline{\delta}}, u_{k-\overline{\delta}+1}, \ldots, u_k$ at least one is implemented. In summary, the class $\mathcal{S}$ of admissible sequences $\{(s_k, \tau_k, m_k)\}_{k \in \mathbb{N}}$ can be described as follows:

$$\begin{aligned} \mathcal{S} := \Bigg\{ & \{(s_k, \tau_k, m_k)\}_{k \in \mathbb{N}} | \; h_{\min} \leq s_{k+1} - s_k \leq h_{\max}, \\ & s_0 = 0, \; \tau_{\min} \leq \tau_k \leq \tau_{\max}, \; \sum_{v=k-\overline{\delta}}^{k} m_v \leq \overline{\delta}, \forall k \in \mathbb{N} \Bigg\}, \end{aligned} \tag{5}$$

which includes variable sampling intervals, large delays, and packet dropouts.

Remark 1. *In the modelling of the network-induced uncertainties, we impose bounds on the delays, sampling intervals and the maximum number of subsequent dropouts as was also done in [22, 24, 40, 47, 49, 159] and many other works. Given such bounds, we will formulate stability conditions and constructive controller synthesis results independent of the probability distribution of the uncertain variables. So, such* robust *results also apply in the stochastic setting and can be seen as probability distribution-free results of the stochastic case if the domain of the probability distribution function is bounded.*

Next, the general description of the continuous-time control input $u^*(t)$ in (3) is reformulated to indicate explicitly which control inputs u_k are active in the sampling interval $[s_k, s_{k+1})$. Such a reformulation is needed to derive the discrete-time NCS model, which will ultimately be employed in the controller synthesis methods.

Lemma 1. *Consider the continuous-time NCS as defined in* (3) *and the admissible sequences of sampling instants, delays, and packet dropouts as defined in* (5)*. Define* $\underline{d} := \lfloor \frac{\tau_{\min}}{h_{\max}} \rfloor$*, the largest integer smaller than or equal to* $\frac{\tau_{\min}}{h_{\max}}$ *and* $\overline{d} := \lceil \frac{\tau_{\max}}{h_{\min}} \rceil$*, the smallest integer larger than or equal to* $\frac{\tau_{\max}}{h_{\min}}$*. Then, the control action* $u^*(t)$ *in the sampling interval* $[s_k, s_{k+1})$ *is described by*

$$u^*(t) = u_{k+j-\overline{d}-\overline{\delta}} \textit{ for } t \in [s_k + t_j^k, s_k + t_{j+1}^k), \tag{6}$$

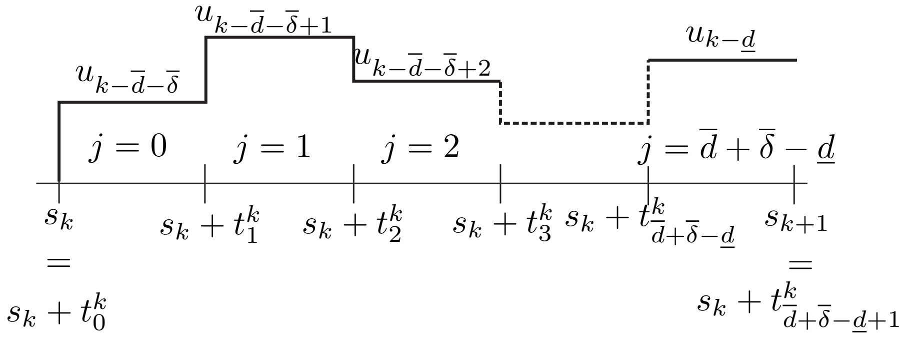

$u_{k-\overline{d}-\overline{\delta}}$
$u_{k-\overline{d}-\overline{\delta}+1}$
$u_{k-\overline{d}-\overline{\delta}+2}$
$u_{k-\underline{d}}$
$j=0$
$j=1$
$j=2$
$j=\overline{d}+\overline{\delta}-\underline{d}$
$s_k = s_k + t_0^k$
$s_k + t_1^k$
$s_k + t_2^k$
$s_k + t_3^k$
$s_k + t^k_{\overline{d}+\overline{\delta}-\underline{d}}$
$s_{k+1} = s_k + t^k_{\overline{d}+\overline{\delta}-\underline{d}+1}$

Figure 2: Graphical interpretation of t_j^k.

where the actuation instants $t_j^k \in [0, h_k]$ are defined as:

$$\begin{aligned}
&t_j^k = \\
&\min\Big\{ \max\{0, \tau_{k+j-\overline{d}-\overline{\delta}} - \sum_{l=k+j-\overline{\delta}-\overline{d}}^{k-1} h_l\} + m_{k+j-\overline{d}-\overline{\delta}} h_{\max}, \\
&\max\{0, \tau_{k+j-\overline{d}-\overline{\delta}+1} - \sum_{l=k+j+1-\overline{\delta}-\overline{d}}^{k-1} h_l\} + m_{k+j-\overline{d}-\overline{\delta}+1} h_{\max}, \\
&\ldots, \max\{0, \tau_{k-\underline{d}} - \textstyle\sum_{l=k-\underline{d}}^{k-1} h_l\} + m_{k-\underline{d}} h_{\max}, h_k \Big\},
\end{aligned} \tag{7}$$

with $t_j^k \leq t_{j+1}^k$ and $j \in \{0, 1, \ldots, \overline{d} + \overline{\delta} - \underline{d}\}$ (see Figure 2). Moreover, $0 = t_0^k \leq t_1^k \leq \ldots \leq t^k_{\overline{d}+\overline{\delta}-\underline{d}} \leq t^k_{\overline{d}+\overline{\delta}-\underline{d}+1} := h_k$.

Proof. The proof is given in Appendix 2.1.7. □

Based on Lemma 1 and $\sigma = \{(s_k, \tau_k, m_k)\}_{k\in\mathbb{N}} \in \mathcal{S}$, the discrete-time NCS model can be defined as:

$$x_{k+1} = e^{Ah_k} x_k + \sum_{j=0}^{\overline{d}+\overline{\delta}-\underline{d}} \int_{h_k - t_{j+1}^k}^{h_k - t_j^k} e^{As} ds B u_{k+j-\overline{d}-\overline{\delta}}, \tag{8}$$

with t_j^k as defined in Lemma 1. The minimum and maximum values of the t_j^k parameters are described in Lemma 2.

Lemma 2. *Consider the time instants t_j^k as defined in* (7)*, where $s_{k+j-\overline{d}-\overline{\delta}}$ (with $h_{k+j-\overline{d}-\overline{\delta}} = s_{k+j-\overline{d}-\overline{\delta}+1} - s_{k+j-\overline{d}-\overline{\delta}}$), $\tau_{k+j-\overline{d}-\overline{\delta}}$, and $m_{k+j-\overline{d}-\overline{\delta}}$ are taken from the class $\mathcal{S}$ defined in* (5)*. The minimum value of t_j^k, $j \in \{0, 1, \ldots, \overline{d} + \overline{\delta} - \underline{d}\}$, is given by*

$$t_{j,\min} = \begin{cases} \min\{\tau_{\min} - \underline{d} h_{\max}, h_{\min}\} & \textit{if } j = \overline{d} + \overline{\delta} - \underline{d} \\ 0 & \textit{if } 1 \leq j < \overline{d} + \overline{\delta} - \underline{d}, \end{cases} \tag{9}$$

and the maximum value of t_j^k, $j \in \{1, 2, \ldots, \overline{d} + \overline{\delta} - \underline{d}\}$, is given by

$$t_{j,\max} = \begin{cases} \min\{\tau_{\max} - (\overline{d} - j) h_{\min}, h_{\max}\} \\ \qquad \textit{if } 1 \leq j \leq \overline{d} - \underline{d} \\ h_{\max} \quad \textit{if } \overline{d} - \underline{d} + 1 \leq j \leq \overline{d} + \overline{\delta} - \underline{d}. \end{cases} \tag{10}$$

Additionally, $t_0^k := 0$ and $t^k_{\overline{d}+\overline{\delta}-\underline{d}+1} := h_k$, which gives for the minimum and maximum bound $t_{\overline{d}+\overline{\delta}-\underline{d}+1} \in [h_{\min}, h_{\max}]$.

Proof. The proof can be derived based on Lemma 1 if the bounds on the delay, sampling interval and number of subsequent packet dropouts are taken into account. The interested reader is referred to [13] for the detailed proof. □

Let θ_k denote the vector of uncertain parameters consisting of the sampling interval and the actuation instants:

$$\theta_k := (h_k, t_1^k, \dots, t_{\overline{d}+\overline{\delta}-\underline{d}}^k). \tag{11}$$

The description of θ_k does not include $t_{\overline{d}+\overline{\delta}-\underline{d}+1}^k$, as $h_k = t_{\overline{d}+\overline{\delta}-\underline{d}+1}^k$, which is already included in θ_k. Moreover, t_0^k is not considered either, since it represents a constant term, $t_0^k = 0$. Using the fact that $h_k \in [h_{\min}, h_{\max}]$ and the bounds $t_{j,\min}, t_{j,\max}$ on the actuation instants given in (9) and (10), we can define the set

$$\begin{aligned} \Theta \;=\; & \{\theta_k \in \mathbb{R}^{\overline{d}+\overline{\delta}-\underline{d}+1} \,|h_k \in [h_{\min}, h_{\max}], \\ & t_j^k \in [t_{j,\min}, t_{j,\max}],\; 1 \leq j \leq \overline{d} + \overline{\delta} - \underline{d}, \\ & 0 \leq t_1^k \leq \dots \leq t_{\overline{d}+\overline{\delta}-\underline{d}}^k \leq h_k\}. \end{aligned} \tag{12}$$

Note that this set does not depend on k. System (8) represents a discrete-time system with multiple delays in the input. Moreover, the system matrices are time-varying according to the uncertain parameters $\theta_k \in \Theta$. In the following section, we will show how to characterize the stability of this system based on LMIs and compare this to the Lyapunov-Krasovskii Function (LKF) approach.

2.1.3 Stability characterizations and relations with the LKF theory

In this section we discuss the stability characterization for the NCS (3) with a state feedback of the form

$$u_k = -\overline{K} x_k. \tag{13}$$

We can without loss of generality assume that $\overline{K}$ has a full row rank. When $\overline{K}$ does not have a full row rank, it is always possible to write the controller in the form

$$u_k = \begin{pmatrix} u_k^a \\ u_k^b \end{pmatrix} = \begin{pmatrix} I \\ G \end{pmatrix} K_a x_k = \begin{pmatrix} I \\ G \end{pmatrix} u_k^a,$$

where K_a has full row rank (possibly after a permutation of the inputs) and we obtain a model similar to (3) with K_a instead of $\overline{K}$ and $B \begin{pmatrix} I \\ G \end{pmatrix}$ instead of B that does satisfy the full row rank condition on the feedback gain.

To render the model (8) with the feedback (13) suitable for analysis, we consider an equivalent delay-free model, based on a lifted state vector

$$\xi_k = \begin{pmatrix} x_k^T & u_{k-1}^T & \dots & u_{k-\overline{d}-\overline{\delta}}^T \end{pmatrix}^T$$

that includes past system inputs.

This leads to the lifted model

$$\xi_{k+1} = \tilde{A}_1(\theta_k)\xi_k, \tag{14}$$

where

$$\tilde{A}_1(\theta_k) = \begin{pmatrix} \Lambda(\theta_k) & \widetilde{M}_{\overline{d}+\overline{\delta}-1}(\theta_k) & \widetilde{M}_{\overline{d}+\overline{\delta}-2}(\theta_k) & \dots & \widetilde{M}_1(\theta_k) & \widetilde{M}_0(\theta_k) \\ -\overline{K} & 0 & 0 & \dots & 0 & 0 \\ 0 & I & 0 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & & \ddots & \ddots & 0 & 0 \\ 0 & \dots & \dots & 0 & I & 0 \end{pmatrix}$$

with $\Lambda(\theta_k) = e^{Ah_k} - \tilde{M}_{\overline{d}+\overline{\delta}}(\theta_k)\overline{K}$ and

$$\widetilde{M}_j(\theta_k) = \begin{cases} \displaystyle\int_{h_k - t_{j+1}^k}^{h_k - t_j^k} e^{As} ds B & \text{if } 0 \leq j \leq \overline{d} + \overline{\delta} - \underline{d}, \\ 0 & \text{if } \overline{d} + \overline{\delta} - \underline{d} < j \leq \overline{d} + \overline{\delta}. \end{cases} \tag{15}$$

The goal of this section is to prove that characterizing the stability of the closed-loop NCS (8) using the lifted model (14) and (parameter-dependent) quadratic Lyapunov functions is less (or, in the worst case, equally) conservative than the methods available in the literature based on discrete-time LKF.

In order to show this, we will use an alternative lifted state space model as an intermediate step in the proof. This model uses the state vector $\chi_k = \left(x_k^T \; x_{k-1}^T \; \dots \; x_{k-\bar{d}-\bar{\delta}}^T\right)^T$, i.e.

$$\chi_{k+1} = \tilde{A}_2(\theta_k)\chi_k, \tag{16}$$

where

$$\tilde{A}_2(\theta_k) = \begin{pmatrix} \Lambda(\theta_k) & -\widetilde{M}_{\bar{d}+\bar{\delta}-1}(\theta_k)\overline{K} & -\widetilde{M}_{\bar{d}+\bar{\delta}-2}(\theta_k)\overline{K} & \dots & -\widetilde{M}_0(\theta_k)\overline{K} \\ I & 0 & 0 & \dots & 0 \\ 0 & I & 0 & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \dots & 0 & I & 0 \end{pmatrix}.$$

This second lifted model is important since it is easy to show that if there exists a LKF (even the most general LKF that can be obtained using quadratic terms), then there exists a parameter-dependent quadratic Lyapunov function for (16) as well. This relation will be described in detail at the end of the section. First we will show that the existence of a parameter-dependent Lyapunov function for (16) is equivalent to the existence of a parameter-dependent Lyapunov function for (14). This issue is relevant since it would formally prove that we can base the stability analysis for the NCS (8) with state feedback controller (13) on (14) without losing stability properties that could be obtained via (16). Note that the state dimension of ξ_k in (14) is smaller than the dimension of χ_k in (16), which clearly has modeling and numerical advantages.

Equivalence of stability characterizations for the two lifted models

Let us discuss the equivalence of the lifted models (16) and (14) with respect to stability and Lyapunov functions in more detail. Clearly, for a given *constant* parameter θ, the stability of (16) is equivalent to the stability of (14) and vice versa. Moreover, since for linear time-invariant systems the existence of a quadratic Lyapunov is a necessary and sufficient stability condition, there exists a quadratic Lyapunov function for (16) if and only if there exists one for (14) when θ is constant. However, assuming that there exists a quadratic Lyapunov function for one of the systems, (16) or (14), there is no constructive method available in the literature for deducing a Lyapunov function for the other one. We will provide such a constructive method, and moreover, we will even consider a more complicated problem as (16) and (14) are uncertain systems that vary over time as θ_k is changing. In this case, quadratic Lyapunov functions are known to be sufficient only for characterizing stability, not necessary. The question is now whether, in the time-varying uncertain case, the existence of a quadratic Lyapunov function for system (16) is equivalent to the existence of a quadratic Lyapunov function for (14). In Theorem 1, we will answer this question and we will show that there exists a quadratic-like Lyapunov function for system (14) if and only if there exists one for the alternative representation (16). To prove this result for any parameter-dependent quadratic Lyapunov function, the following lemma will be needed.

Lemma 3. *Consider the matrix $R \in \mathbb{R}^{q \times p}$ and the matrices $A(\theta) \in \mathbb{R}^{p \times p}$ that depend continuously on $\theta \in \Theta$, where $\Theta \subset \mathbb{R}^l$ is a compact set. Define the matrices*

$$\bar{A}(\theta) = \begin{pmatrix} A(\theta) & 0 \\ R & 0 \end{pmatrix} \in \mathbb{R}^{(p+q)\times(p+q)}, \tag{17}$$

for $\theta \in \Theta$. The following statements are equivalent:

- *There exist symmetric positive definite matrices $P(\theta) \in \mathbb{R}^{p+q \times p+q}$, $\theta \in \Theta$ such that*

$$\bar{A}(\theta_1)^T P(\theta_2)\bar{A}(\theta_1) - P(\theta_1) < 0, \; \forall\, \theta_1, \theta_2 \in \Theta. \tag{18}$$

- *There exist symmetric positive definite matrices $Q(\theta) \in \mathbb{R}^{p \times p}$, $\theta \in \Theta$ such that*

$$A(\theta_1)^T Q(\theta_2) A(\theta_1) - Q(\theta_1) < 0, \; \forall\, \theta_1, \theta_2 \in \Theta. \tag{19}$$

Moreover, there exists a common solution $P(\theta) = P > 0$, for all $\theta \in \Theta$ to (18) *if and only if there exists a common solution $Q(\theta) = Q > 0$ to* (19).

Proof. See the Appendix. □

Theorem 1. *Consider the NCS* (8) *with state feedback controller* (13) *and the two representations* (14) *and* (16)*. The following statements are equivalent :*

- *There exist symmetric positive definite matrices $P(\theta)$, $\theta \in \Theta$ such that*

$$\tilde{A}_1^T(\theta_k)P(\theta_{k+1})\tilde{A}_1(\theta_k) - P(\theta_k) < 0, \tag{20}$$

  for all $\theta_k, \theta_{k+1} \in \Theta$, thus

$$V(\xi_k) = \xi_k^T P(\theta_k)\xi_k \tag{21}$$

  is a parameter-dependent Lyapunov function for system (14)*.*

- *There exist symmetric positive definite matrices $Q(\theta)$, $\theta \in \Theta$ such that*

$$\tilde{A}_2^T(\theta_k)Q(\theta_{k+1})\tilde{A}_2(\theta_k) - Q(\theta_k) < 0, \tag{22}$$

  for all $\theta_k, \theta_{k+1} \in \Theta$, thus

$$V(\chi_k) = \chi_k^T Q(\theta_k)\chi_k \tag{23}$$

  is a parameter-dependent Lyapunov function for system (16)*. Moreover, system* (14) *has a common quadratic Lyapunov function $V(\xi_k) = \xi_k^T P\xi_k$ if and only if system* (16) *has a common quadratic Lyapunov function $V(\chi_k) = \chi_k^T Q\chi_k$.*

Proof. Since the state feedback matrix $\overline{K}$ has full row rank there exists a matrix $S \in \mathbb{R}^{(n-m)\times n}$ such that the matrix $\begin{pmatrix} \overline{K} \\ S \end{pmatrix}$ is invertible. Define the matrices

$$\tilde{A}_3(\theta_k) = \left(\begin{array}{c|cccc|c|c|c}
\Lambda(\theta_k) & \widetilde{M}_{\overline{d}+\overline{\delta}-1}(\theta_k) & \dots & \dots & \widetilde{M}_1(\theta_k) & \widetilde{M}_0(\theta_k) & 0 & 0 \\ \hline
-\overline{K} & 0 & \dots & \dots & 0 & 0 & 0 & 0 \\ \hline
0 & I & 0 & \dots & 0 & 0 & 0 & 0 \\
\vdots & 0 & \ddots & & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & & \ddots & 0 & \vdots & \vdots & \vdots \\
0 & 0 & \dots & 0 & I & 0 & 0 & 0 \\ \hline
S & 0 & \dots & \dots & 0 & 0 & 0 & 0 \\ \hline
0 & 0 & \dots & \dots & 0 & 0 & I & 0
\end{array}\right)$$

and

$$W = \left(\begin{array}{c|cccc|c}
I & 0 & \dots & \dots & 0 & 0 \\ \hline
0 & -\overline{K} & 0 & \dots & 0 & 0 \\
\vdots & 0 & \ddots & & \vdots & \vdots \\
\vdots & \vdots & & \ddots & 0 & \vdots \\
0 & 0 & \dots & 0 & -\overline{K} & 0 \\ \hline
0 & 0 & \dots & \dots & 0 & -\overline{K} \\ \hline
0 & S & 0 & \dots & 0 & 0 \\
\vdots & 0 & \ddots & & \vdots & \vdots \\
\vdots & \vdots & & \ddots & 0 & \vdots \\
0 & 0 & \dots & 0 & S & 0 \\ \hline
0 & 0 & \dots & \dots & 0 & S
\end{array}\right) \in \mathbb{R}^{(n+1)\cdot(\overline{d}+\overline{\delta})\times(n+1)\cdot(\overline{d}+\overline{\delta})}.$$

Notice that

$$\tilde{A}_3(\theta_k)W = W\tilde{A}_2(\theta_k) = \left(\begin{array}{c|cccc|c} \Lambda(\theta_k) & -\widetilde{M}_{\overline{d}+\overline{\delta}-1}(\theta_k)\overline{K} & \dots & \dots & -\widetilde{M}_1(\theta_k)\overline{K} & -\widetilde{M}_0(\theta_k)\overline{K} \\ \hline -\overline{K} & 0 & \dots & \dots & 0 & 0 \\ \hline 0 & -\overline{K} & 0 & \dots & 0 & 0 \\ \vdots & 0 & \ddots & & \vdots & \vdots \\ \vdots & \vdots & & \ddots & 0 & \vdots \\ 0 & 0 & \dots & 0 & -\overline{K} & 0 \\ \hline S & 0 & \dots & \dots & 0 & 0 \\ \hline 0 & S & 0 & \dots & 0 & 0 \\ \vdots & 0 & \ddots & & \vdots & \vdots \\ \vdots & \vdots & & \ddots & 0 & \vdots \\ 0 & 0 & \dots & 0 & S & 0 \end{array}\right).$$

This implies that $\tilde{A}_3(\theta_k)$ is similar to $\tilde{A}_2(\theta_k)$. It is easy to show that (22) holds if and only if there exists symmetric positive definite matrices $\tilde{P}(\theta_k) = W^{-T}Q(\theta_k)W^{-1}$ such that

$$\tilde{A}_3^T(\theta_k)\tilde{P}(\theta_{k+1})\tilde{A}_3(\theta_k) - \tilde{P}(\theta_k) < 0.$$

Furthermore, notice that $\tilde{A}_3(\theta_k)$ can be expressed as

$$\tilde{A}_3(\theta_k) = \left(\begin{array}{c|c|c} \tilde{A}_1(\theta_k) & 0 & 0 \\ \hline S \mid 0 & 0 & 0 \\ \hline 0 & I & 0 \end{array}\right).$$

Then, apply Lemma 3 with

$$A(\theta) := \left(\begin{array}{c|c} \tilde{A}_1(\theta_k) & 0 \\ \hline S \mid 0 & 0 \end{array}\right) \text{ and } R := \left(\begin{array}{c|c} 0 & I \end{array}\right).$$

Next apply Lemma 3 again for

$$A(\theta) := \tilde{A}_1(\theta_k) \text{ and } R := \left(\begin{array}{c|c} S & 0 \end{array}\right)$$

in order to complete the proof. □

Relations with the Lyapunov-Krasovskii stability characterization

For discrete-time uncertain systems with delay in the input such as (8), several stability results exist based on Lyapunov-Krasovskii functions (LKFs). Using an adequate partition of the Lyapunov matrix

$$Q(\theta_k) = \begin{pmatrix} Q^{0,0}(\theta_k) & Q^{0,1}(\theta_k) & \dots & Q^{0,\overline{d}+\overline{\delta}}(\theta_k) \\ Q^{0,1}(\theta_k) & Q^{1,1}(\theta_k) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ Q^{0,\overline{d}+\overline{\delta}}(\theta_k) & \dots & \dots & Q^{\overline{d}+\overline{\delta},\overline{d}+\overline{\delta}}(\theta_k) \end{pmatrix}, \tag{24}$$

it can be shown that the Lyapunov function (23) is equivalent to the LKF

$$V(x_k, \dots, x_{k-\overline{d}-\overline{\delta}}) = \sum_{i=0}^{\overline{d}+\overline{\delta}} \sum_{j=0}^{\overline{d}+\overline{\delta}} x_{k-i}^T Q^{i,j}(\theta_k) x_{k-j}, \tag{25}$$

which is the most general LKF that can be obtained using quadratic forms. Any of the quadratic LKFs found in the literature (see [54, 77, 80, 81]) are a particular case of (25). As a consequence of Theorem 1, we know that there exists a Lyapunov function (23) for (16) if and only if there exists one of the form (21) for (14), i.e. if and only if the equations (20) are satisfied. Consequently, condition (20) represents a necessary and sufficient condition for the existence of the most general form of LKFs that can be obtained using quadratic terms as in (25). Hence, using a stability characterization based on the model (14) is less (or, in the worst case, equally) conservative than the stability analysis results based on quadratic LKF that are available in the literature [54, 77, 80, 81].

In the next section, we will present a constructive LMI method for controller design using stability characterizations based on parameter-dependent Lyapunov functions such as in (21).

2.1.4 Controller synthesis

To render the model (8) suitable for controller synthesis, we rewrite it as:

$$\xi_{k+1} = \tilde{A}(\theta_k)\xi_k + \tilde{B}(\theta_k)u_k, \tag{26}$$

where

$$\tilde{A}(\theta_k) = \begin{pmatrix} e^{Ah_k} & \widetilde{M}_{\overline{d}+\overline{\delta}-1}(\theta_k) & \widetilde{M}_{\overline{d}+\overline{\delta}-2}(\theta_k) & \dots & \widetilde{M}_0(\theta_k) \\ 0 & 0 & 0 & \dots & 0 \\ 0 & I & 0 & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \dots & 0 & I & 0 \end{pmatrix}, \tag{27}$$

and

$$\tilde{B}(\theta_k) = \begin{pmatrix} \widetilde{M}^T_{\overline{d}+\overline{\delta}}(\theta_k) & I & 0 & \dots & 0 \end{pmatrix}^T. \tag{28}$$

This model is equivalent to (14) when the input is a state feedback of the form (13).

Let us now design a static state feedback controller of the form (13). The main difficulty to synthesize a state feedback (13) is that it results in a structured control synthesis problem, i.e. we need to design a control law of the form

$$u_k = -K\xi_k \tag{29}$$

with a specific structure in the feedback gain matrix: $K = \begin{pmatrix} \overline{K} & 0_{m\times(\overline{d}+\overline{\delta})m} \end{pmatrix}$. A solution to this structured controller synthesis problem is to apply the approach presented in [15]. Moreover, such an approach allows for the use of a parameter-dependent Lyapunov function [14] that might result in less conservative controller synthesis results than the use of a common quadratic Lyapunov function.

Remark 2. *From the control synthesis point of view, when dealing with a system such as* (26)*, a natural alternative would be to design a state feedback controller of the form* (29) *using the full state* ξ_k *of the underlying model* (26)*. However, from the point of view of the NCS* (3)*, this is equivalent to using a dynamical controller of the form*

$$u_k = -K_0x_k - K_1u_{k-1} \dots - K_{\overline{d}+\overline{\delta}}u_{k-\overline{d}-\overline{\delta}}. \tag{30}$$

The use of such a dynamic control law requires a reconsideration of the assumption made earlier to lump all the delays τ_k^{sc},τ_k^c *and* τ_k^{ca} *in one parameter* τ*. Using a dynamic control law as in* (30) *actually leads to more restrictive assumptions on the network modeling setup as* $y_k = x_k$ *should always arrive at the controller after the moment that* u_{k-1} *is sent to the actuator, i.e.* $s_k + \tau_k^{sc} > s_{k-1} + \tau_{k-1}^{sc} + \tau_{k-1}^{c}$ *as otherwise special precautions are needed to handle out-of-order arrival of measured outputs resulting in longer delays. In addition, the adopted modeling setup and controller in* (30) *require that no packet dropouts occur between the sensors and the controller. Namely, in the case of a packet dropout between the sensor and controller, it is possible that* $y_k = x_k$ *does not arrive at the controller and thus* u_k *cannot be computed; furthermore the controller* (30) *cannot be updated beyond the* k*-th update. Therefore, a deadlock in the controller can occur and the worst case scenario would be not sending control updates at all to the actuator. Although modeling dropouts alternatively as prolongations of the sampling interval (see, e.g., the comparison in [69]) might alleviate these issues to some extent, dropouts in the channel between the controller and the actuators introduce similar complications in this case. We care to stress that a static state feedback as in* (13) *does not suffer from such problems and additional assumptions, as explicated in above, are not needed, which greatly enhances its applicability.*

To derive the control synthesis conditions, the model (26) is rewritten using the real Jordan form of the continuous-time system matrix A. Basically, we express the state matrix $A = TJT^{-1}$ with J the real Jordan form, and T an invertible matrix. Next, we compute all the integrals in (15) using $e^{As} = Te^{Js}T^{-1}$ to obtain a model in which the uncertain parameters θ_k appear explicitly. This leads to a generic model of the form

$$\begin{aligned} \xi_{k+1} = {} & \left(F_0 + \sum_{i=1}^{\zeta} \alpha_i(\theta_k)F_i\right)\xi_k \\ & + \left(G_0 + \sum_{i=1}^{\zeta} \alpha_i(\theta_k)G_i\right)u_k, \end{aligned} \tag{31}$$

with θ_k defined in (11) and ζ the number of time-varying functions $\alpha_i(\cdot)$ given by $(\overline{d}+\overline{\delta}-\underline{d}+1)\nu$, with $\nu \leq n$, where n is the dimension of the state vector x. We have $\nu = n$ when the geometric multiplicity of each distinct eigenvalue of A is equal to one and $\nu < n$ when the geometric multiplicity of an eigenvalue is larger than one. A typical function $\alpha_i(\theta_k)$ is of the form $e^{\lambda(h_k - t_j^k)}$, with λ a real eigenvalue of A, and of the form $e^{a(h_k-t_j^k)}\cos(b(h_k - t_j^k))$ or $e^{a(h_k-t_j^k)}\sin(b(h_k - t_j^k))$ when λ is a complex eigenvalue ($\lambda = a + bj$) of A. For more details on the use of the Jordan form, including the case that $\nu < n$ the reader is referred to Appendix B in [13].

Using bounds on the uncertain parameters $\theta_k = (h_k, t_1^k, \ldots, t^k_{\overline{d}+\overline{\delta}-\underline{d}})$ described by the set Θ in equation (12) this gives rise to the set of matrices

$$\mathcal{FG} = \left\{ \left(F_0 + \sum_{i=1}^{\zeta} \alpha_i(\theta_k) F_i, G_0 + \sum_{i=1}^{\zeta} \alpha_i(\theta_k) G_i \right) | \theta_k \in \Theta \right\} \tag{32}$$

that contains all possible matrix combinations in (26) and (31). Based on this infinite set $\mathcal{FG}$ of matrices we will derive a stabilizing controller of the form (13) for the NCS (3). To overcome the infinite dimension of the set $\mathcal{FG}$ a convex overapproximation of the set is used. Denote the maximum and minimum value of $\alpha_i(\theta_k)$, respectively, by

$$\overline{\alpha}_i = \max_{\theta_k \in \Theta} \alpha_i(\theta_k), \; \underline{\alpha}_i = \min_{\theta_k \in \Theta} \alpha_i(\theta_k), \tag{33}$$

with Θ defined in (12). Then the set of matrices $\mathcal{FG}$, given in (32), is a subset of $co\,(\mathcal{H}_{FG})$ with

$$\begin{aligned} \mathcal{H}_{FG} = \Bigg\{ \Big((F_0 + \sum_{i=1}^{\zeta} \alpha_i F_i), (G_0 + \sum_{i=1}^{\zeta} \alpha_i G_i) \Big) : \\ \alpha_i \in \{\underline{\alpha}_i, \overline{\alpha}_i\}, i = 1, 2, \ldots, \zeta \Bigg\}, \end{aligned} \tag{34}$$

where 'co' denotes the convex hull.

We will also write the set of vertices $\mathcal{H}_{FG}$ as $\mathcal{H}_{FG} = \{(H_{F,j}, H_{G,j}) \mid j = 1, 2, \ldots, 2^{\zeta}\}$. Using this finite set of 2^{ζ} vertices, a finite number of LMI controller synthesis conditions are given for the state-feedback controller (13) in the following theorem.

Theorem 2. *Consider the NCS model* (3), (6), (7), (13)*, and its discrete-time representation* (26), (13) *for sequences of sampling instants, delays, and packet dropouts $\sigma \in \mathcal{S}$ with $\mathcal{S}$ as in* (5)*. Consider the equivalent representation* (31) *based on the Jordan form of A and the set of vertices $\mathcal{H}_{FG}$ defined in* (34)*.*

If there exist symmetric positive definite matrices $Y_j \in \mathbb{R}^{(n+(\overline{d}+\overline{\delta})m)\times(n+(\overline{d}+\overline{\delta})m)}$, a matrix $\overline{Z} \in \mathbb{R}^{m\times n}$, matrices $X_j = \begin{pmatrix} \overline{X}_1 & 0 \\ \overline{X}_{2,j} & \overline{X}_{3,j} \end{pmatrix}$, with $\overline{X}_1 \in \mathbb{R}^{n\times n}$, $\overline{X}_{2,j} \in \mathbb{R}^{(\overline{d}+\overline{\delta})m\times n}$, $\overline{X}_{3,j} \in \mathbb{R}^{(\overline{d}+\overline{\delta})m\times(\overline{d}+\overline{\delta})m}$, $j = 1, 2, \ldots, 2^{\zeta}$, and a scalar $0 \leq \gamma < 1$ that satisfy

$$\begin{pmatrix} X_j + X_j^T - Y_j & X_j^T H_{F,j}^T - \begin{pmatrix} \overline{Z} & 0 \end{pmatrix}^T H_{G,j}^T \\ H_{F,j} X_j - H_{G,j} \begin{pmatrix} \overline{Z} & 0 \end{pmatrix} & (1-\gamma) Y_l \end{pmatrix} > 0, \tag{35}$$

for all $j, l \in \{1, 2, \ldots, 2^{\zeta}\}$, then the closed-loop NCS (3), (6), (7), (13) *with $\overline{K} = \overline{Z}\,\overline{X}_1^{-1}$ is globally asymptotically stable.*

Proof. To prove this theorem, we first note that, due to the convex overapproximation based on the uncertain parameters $\alpha_i(\cdot)$, it holds for all $\theta_k \in \Theta$ that $\left(\tilde{A}(\theta_k), \tilde{B}(\theta_k)\right) \in \mathcal{FG} \subset$ co$(\mathcal{H}_{FG})$. Hence, for the stability of (26) and (31) with the state feedback controller (13), it is sufficient to prove stability of the system

$$\xi_{k+1} = \sum_{j=1}^{2^{\zeta}} \mu_j^k \left(H_{F,j} - H_{G,j} K \right) \xi_k, \tag{36}$$

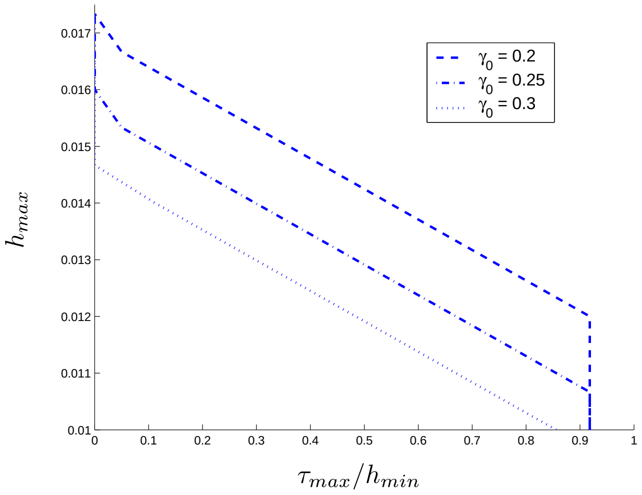


Figure 3: Feasibility regions for different transient response specifications.

where $K = \begin{pmatrix}\overline{K} & 0_{m\times(\bar{d}+\bar{\delta})m}\end{pmatrix}$ and $\mu_1^k, \mu_2^k, \dots, \mu_{2\varsigma}^k \geq 0$, satisfy $\sum_{j=1}^{2\varsigma} \mu_j^k = 1$, for all $k \in \mathbb{N}$. Assume that the inequalities (35) hold. Using the fact that $\overline{K}\,\overline{X}_1 = \overline{Z}$ we have that

$$\begin{pmatrix}\overline{Z} & 0\end{pmatrix} = K\begin{pmatrix}\overline{X}_1 & 0 \\ \overline{X}_{2,j} & \overline{X}_{3,j}\end{pmatrix},$$

and thus we obtain :

$$\begin{pmatrix} X_j + X_j^T - Y_j & X_j^T(H_{F,j} - H_{G,j}K)^T \\ (H_{F,j} - H_{G,j}K)X_j & (1-\gamma)Y_l \end{pmatrix} > 0. \tag{37}$$

Applying Theorem 3 in [14] this inequality implies that the function

$$V(\xi_k) = \xi_k^T P(\mu_1^k, \mu_2^k, \dots, \mu_{2\varsigma}^k)\xi_k, \tag{38}$$

with $P(\mu_1^k, \mu_2^k, \dots, \mu_{2\varsigma}^k) = \sum_{j=1}^{2\varsigma} \mu_j^k P_j$ and $P_j = Y_j^{-1}$, is strictly decreasing along the trajectories of system (36). Consequently system (26), (13) is globally asymptotically stable. Using similar arguments as in [28] it can be shown that the intersample behaviour is stable as well and, consequently, that the NCS (3), (6), (7), (13) for all $\sigma \in \mathcal{S}$ is globally asymptotically stable. □

Remark 3. *This theorem shows that* (38) *is a parameter-dependent Lyapunov function for the system* (26) *with the controller* (13)*. Using the results from the previous section, this shows that if the LMIs* (35) *are satisfied they imply the existence of a LKF of the form* (25)*. Notice that using this approach we avoid the conservative upper boundings in the difference of the LKF, which are usually encountered in the literature to arrive at LKF-based stability conditions in LMI form.*

The case of a common quadratic Lyapunov function (CQLF) $V(\xi) = \xi_k^T P \xi_k$ is a particular case of this theorem by taking $Y_j = Y,\ \forall\ j = 1, \dots, 2^\varsigma$, with $P = Y^{-1}$.

If one is still interested in using an extended state feedback (29) *despite the mentioned disadvantages, then Theorem 2 can be modified by replacing the matrices X_j, $\forall i \neq j$ with a constant matrix X without a specific structure and using Z instead of $\begin{pmatrix}\overline{Z} & 0\end{pmatrix}$. The extended controller is obtained then by $K = ZX^{-1}$.*

In this paper we adopted an overapproximation of the NCS model using the Jordan form and leading to (34)*. All the theory also applies if the overapproximation is obtained by other techniques (e.g. [22, 28] or any other).*

2.1.5 Illustrative examples

Consider a NCS represented by (3), with $A = \begin{pmatrix}0 & 1 \\ 0 & 0\end{pmatrix}$ and $B = \begin{pmatrix}0 \\ 1\end{pmatrix}$ (double integrator). First, let us show the applicability of the presented theory for time-varying sampling intervals and delays. We

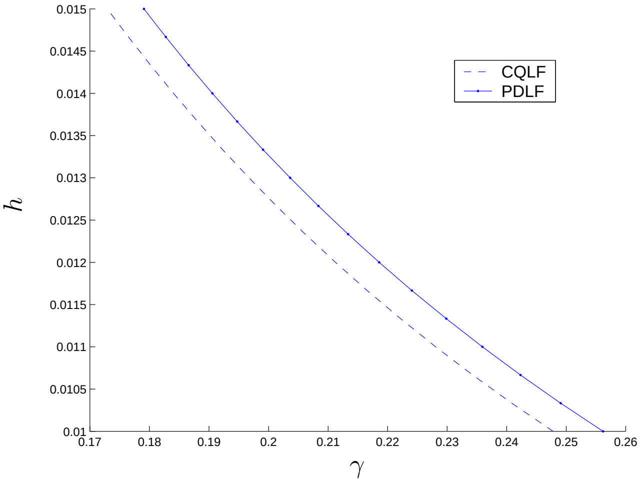


Figure 4: Comparison between the CQLF and PDLF approaches.

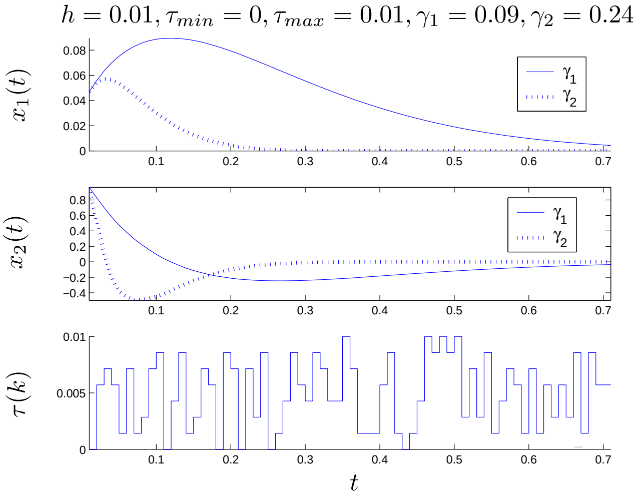


Figure 5: Evolution from an arbitrary initial condition.

consider $\overline{\delta} = 0$ and we analyze the feasibility of the LMI conditions in Theorem 2 for various values of τ_{max}, h_{max} and γ (keeping h_{min} fixed, i.e. $h_{min} = 0.01s$). In order to use the same continuous-time transient response specifications, the parameter γ used in the LMIs is scaled according to the different values of h_{max}, i.e. we use $\gamma = 1 - (1 - \gamma_0)^{h_{max}/h_{min}}$ where γ_0 represents the value taken for $h_{max} = h_{min} = 0.01s$. The delay is considered to be time-varying, and the LMIs are solved for $\tau_{min} = 0.1h_{min}$ and τ_{max} in the interval $[0.1h_{min}, 0.98h_{min}]$. Note that in this case message rejection can not occur since $\tau_{max} < h_{min}$. The tradeoff curves between transient performance (decay rate) and robustness versus uncertainties (h_{max},τ_{max}), i.e. the regions for which Theorem 2 provides a stabilizing state feedback, are depicted in Figure 3. We can remark that the feasibility of LMIs is reduced as γ_0 increases. This is due to the fact that if the parameter γ_0 is increased, a faster transient response is required. As an example, for the bounds $h_{\min} = 0.01$s and $h_{\max} = 0.014$s on the sampling interval and time-varying delays characterized by $\tau_{\min} = 0.1h_{\min}$ and $\tau_{\max} = 0.6h_{\min}$, the stabilizing state-feedback controller with gain $K = \begin{pmatrix} 0.622 & 1.089 \end{pmatrix}$ is obtained using Theorem 2 with $\gamma_0 = 0$. For $\gamma_0 = 0.17$ a faster transient response is obtained with the controller gain $K = \begin{pmatrix} 164.837 & 22.64 \end{pmatrix}$.

Next, a comparison between the use of a common quadratic (CQLF) and of a parameter-dependent Lyapunov function (PDLF) is given in Figure 4 for a constant sampling interval and time-varying delays characterized by $\tau_{\min} = 0$ and $\tau_{\max} = h$. The example illustrates the improvement of the transient response specification (γ) with the PDLF approach. A simulation is given in Figure 5, for two different values of γ with $h_{min} = h_{max} = 0.01s$ and $\tau_{max} = h_{min}$. The controller gain used in this simulation has been obtained using parameter-dependent Lyapunov functions.

Finally, we illustrate the situation with time-delays larger than the constant sampling interval h. In this case $\tau_{\max} = 2.8h$ and $\tau_{\min} = 0$. Note that the same results hold also for the situation with packet

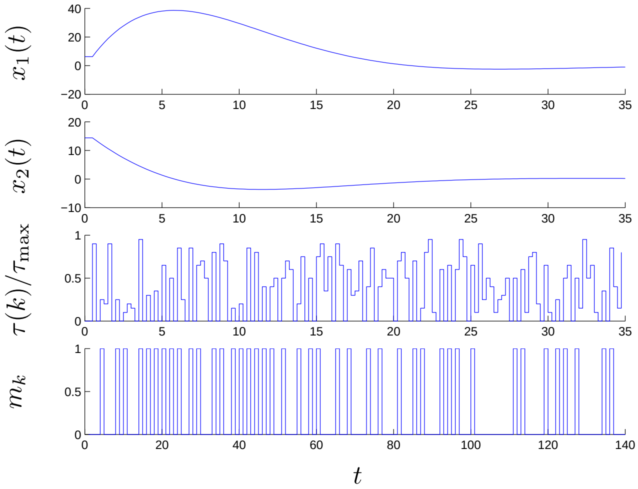


Figure 6: Time response with delay and packet-dropouts for $h = 0.25$, $\tau_{\max} = h$ and $\overline{\delta} = 1$.

dropouts $\overline{\delta} = 1$ and $\tau_{\max} = 1.8h$ or $\overline{\delta} = 2$ and $\tau_{\max} = 0.8h$. This is generally true for our result as long as $\overline{d} + \overline{\delta} = constant$. In this case a stabilizing controller can be found for sampling intervals up to $h = 0.55$s (e.g. for $h = 0.55$s a stabilizing controller is given by $K = \begin{pmatrix} 0.0363 & 0.2525 \end{pmatrix}$). A simulation with both delay and packet-dropouts using the latter controller gain is presented in Figure 6 for $h = 0.25$, $\tau_{\max} = h$ and $\overline{\delta} = 1$.

2.1.6 Conclusions

A discrete-time NCS model, based on an exact discretization of the continuous-time system at the sampling instants, is presented. This model includes all relevant network phenomena: the presence of time-varying delays that may be larger than the sampling interval, message rejection, packet dropouts, and variations in the sampling interval. Next, a stability characterization based on parameter-dependent Lyapunov functions is proposed. It is shown theoretically that the stability characterization presented here is generally less conservative than the methods available in the literature based on LKF. Based on the developed model and on the proposed stability characterization, constructive state feedback synthesis conditions are derived in terms of linear matrix inequalities (LMIs). Simulations are presented that show the applicability and effectiveness of the obtained controller synthesis results.

2.1.7 Appendix A: Proof of Lemma 1

To prove that $u_{k-\overline{d}-\overline{\delta}}$ is the oldest control input that might be active during the sampling interval $[s_k, s_{k+1})$, we consider, firstly, the case without packet dropouts, and secondly, the case with packet dropouts. From the definition of $\overline{d}$ in Lemma 1, it follows that the control input $u_{k-\overline{d}}$ is always available at the plant before or exactly at s_k, if $u_{k-\overline{d}}$ is not dropped. To prove this, we use the relation $s_k = s_{k-\overline{d}} + \sum_{l=k-\overline{d}}^{k-1} h_l$, which provides the upper and lower bounds on s_k, given by $s_{k-\overline{d}} + \overline{d}h_{\min} \leq s_k \leq s_{k-\overline{d}} + \overline{d}h_{\max}$. Combining the lower bound on s_k and $s_{k-\overline{d}} + \tau_{k-\overline{d}} \leq s_{k-\overline{d}} + \tau_{\max}$ gives: $s_{k-\overline{d}} + \tau_{k-\overline{d}} \leq s_{k-\overline{d}} + \tau_{\max} \leq s_k - \overline{d}h_{\min} + \tau_{\max} \leq s_k$, due to the definition of $\overline{d} = \lceil \frac{\tau_{\max}}{h_{\min}} \rceil$. Hence, in case that the control input $u_{k-\overline{d}}$ is not dropped (i.e. $m_{k-\overline{d}} = 0$), it is available before or on s_k and no older control inputs $u_{k+j-\overline{d}}$, with $j < 0$ will be active in the sampling interval $[s_k, s_{k+1})$. To show that $u_{k-\overline{d}}$ can be active in the sampling interval $[s_k, s_{k+1})$, we need to show that $u_{k-\overline{d}+1}$ can become active after s_k, if no packets are dropped. To do so, note that $\overline{d} - 1 < \tau_{\max}/h_{\min} \leq \overline{d}$ and thus we have that $s_{k-\overline{d}+1} + \tau_{\max} > s_{k-\overline{d}+1} + (\overline{d} - 1)h_{\min}$. As the smallest value of $s_k = s_{k-\overline{d}+1}, \sum_{l=k-\overline{d}+1}^{k-1} h_l$ is equal to $s_{k-\overline{d}+1} + (\overline{d} - 1)h_{\min}$, and the largest implementation time of $u_{k-\overline{d}+1}$ is $s_{k-\overline{d}+1} + \tau_{\max}$, the previous inequality shows that $u_{k-\overline{d}+1}$ might be available for implementation (strictly) after s_k. As a consequence, in the case without dropouts, $u_{k-\overline{d}+1}$ can indeed be active in the sampling interval $[s_k, s_{k+1})$.

To prove that in the case of packet dropouts $u_{k-\overline{d}-\overline{\delta}}$ is the oldest control input that can possibly be active in $[s_k, s_{k+1})$, note that, from (4), it follows that at least one of the control inputs $u_{k-\overline{d}-\overline{\delta}}, u_{k-\overline{d}-\overline{\delta}+1}, \dots, u_{k-\overline{d}}$ is not lost. If $u_{k-\overline{d}+1}$ is indeed implemented after s_k (which is possible as just shown), then at least one of the inputs $u_{k-\overline{d}-\overline{\delta}}, u_{k-\overline{d}-\overline{\delta}+1}, \dots, u_{k-\overline{d}}$ will be active in the sampling interval $[s_k, s_{k+1})$. The fact that the maximum number of subsequent packet dropouts equals $\overline{\delta}$ implies that $u_{k-\overline{d}-\overline{\delta}}$ is the oldest control input that might be implemented in the sampling interval $[s_k, s_{k+1})$.

From the definition of $\underline{d}$ in Lemma 1, it follows that the input $u_{k-\underline{d}}$ represents the most recent control input that might be implemented during the sampling interval $[s_k, s_{k+1})$. To prove this, consider the smallest time at which $u_{k-\underline{d}}$ might be implemented that is given by $s_{k-\underline{d}}+\tau_{\min}$. Based on the definition of $\underline{d}$, which gives that $\tau_{\min} < (\underline{d}+1)h_{\max}$, we can conclude that $s_{k-\underline{d}} + \tau_{\min} < s_{k-\underline{d}} + (\underline{d}+1)h_{\max}$. Combining this with the tight bounds on s_{k+1}, given by:

$$s_{k-\underline{d}} + (\underline{d}+1)h_{\min} \leq s_{k+1} \leq s_{k-\underline{d}} + (\underline{d}+1)h_{\max}$$

yields that it might hold that $s_{k-\underline{d}} + \tau_{\min} \leq s_{k+1}$ as s_{k+1} may attain the value $s_{k-\underline{d}} + (\underline{d}+1)h_{\max}$. Consequently, $u_{k-\underline{d}}$ might be implemented before s_{k+1}.

To show that $u_{k-\underline{d}}$ is the most recent data that can be active in $[s_k, s_{k+1})$, we prove that more recent control inputs always arrive after s_{k+1}. Consider $j > \overline{d}+\overline{\delta}-\underline{d}$. Then, we have that $s_{k+j-\overline{d}-\overline{\delta}} + \tau_{\min}$ is the earliest time at which $u_{k+j-\overline{d}-\overline{\delta}}$ might be implemented. To determine if this moment may occur before s_{k+1}, consider the upper bound on s_{k+1}, in terms of $s_{k+j-\overline{d}-\overline{\delta}}$, given by $s_{k+1} \leq s_{k+j-\overline{d}-\overline{\delta}} + (-j+\overline{d}+\overline{\delta}+1)h_{\max}$ for $j > \overline{d}+\overline{\delta}-\underline{d}$. However, for $j > \overline{d}+\overline{\delta}-\underline{d}$

$$s_{k+j-\overline{d}-\overline{\delta}} + \tau_{\min} \geq s_{k+j-\overline{d}-\overline{\delta}} + (-j+\overline{d}+\overline{\delta}+1)h_{\max} \geq s_{k+1},$$

due to the definition of $\underline{d} = \lfloor \frac{\tau_{\min}}{h_{\max}} \rfloor$. This proves that $u_{k-\underline{d}}$ is indeed the most recent control input that can be active in the sampling interval $[s_k, s_{k+1})$.

So far we proved that $u_{k+j-\overline{d}-\overline{\delta}},\ j \in \{0, 1, \dots, \overline{d}+\overline{\delta}-\underline{d}\}$ are the only control values that can be implemented in $[s_k, s_{k+1})$. Now, the times t_j^k with $j \in \{0, 1, \dots, \overline{d}+\overline{\delta}-\underline{d}\}$ will be constructed in such a manner that $[s_k + t_j^k, s_k + t_{j+1}^k)$ is the time interval in which the control input $u_{k+j-\overline{d}-\overline{\delta}}$ is active in the sampling interval $[s_k, s_{k+1})$. The time $t^k_{\overline{d}+\overline{\delta}-\underline{d}}$ (being the starting time of $u_{k-\underline{d}}$ in the interval $[s_k, s_{k+1}]$) is given by

$$t^k_{\overline{d}+\overline{\delta}-\underline{d}} = \min\left[h_k, \tau_{k-\underline{d}} - \sum_{l=k-\underline{d}}^{k-1} h_l + m_{k-\underline{d}} h_{\max}\right]. \tag{39}$$

Indeed, if $m_{k-\underline{d}} = 0$, then $s_k + \tau_{k-\underline{d}} - \sum_{l=k-\underline{d}}^{k-1} h_l$ is the time at which $u_{k-\underline{d}}$ is available at the plant. If $\tau_{k-\underline{d}} - \sum_{l=k-\underline{d}}^{k-1} h_l > h_k$, then $u_{k-\underline{d}}$ might be active after s_{k+1}, but not in $[s_k, s_{k+1})$. Since we are only interested in the interval $[s_k, s_{k+1})$, we take the minimum of this value and h_k in (39). Note that, by the definition of $\underline{d}$, $\tau_{k-\underline{d}} - \sum_{l=k-\underline{d}}^{k-1} h_l \geq 0$. Finally, if $u_{k-\underline{d}}$ is dropped, i.e. $m_{k-\underline{d}} = 1$, then the expression in (39) gives h_k, which means that the input $u_{k-\underline{d}}$ is not used in $[s_k, s_{k+1})$.

Next, as $u_{k-\underline{d}-1}$ can only be active before $u_{k-\underline{d}}$ is available, $t^k_{\overline{d}+\overline{\delta}-\underline{d}-1}$ is given by

$$\begin{aligned} t^k_{\overline{d}+\overline{\delta}-\underline{d}-1} = \min\Bigg[t^k_{\overline{d}+\overline{\delta}-\underline{d}}, \max\left\{0, \tau_{k-\underline{d}-1} - \sum_{l=k-\underline{d}-1}^{k-1} h_l\right\} \\ + m_{k-\underline{d}-1} h_{\max}\Bigg]. \end{aligned} \tag{40}$$

Similarly to $t^k_{\overline{d}+\overline{\delta}-\underline{d}}$, if $\max\{0, \tau_{k-\underline{d}-1} - \sum_{l=k-\underline{d}-1}^{k-1} h_l\} + m_{k-\underline{d}-1} h_{\max} \in [0, t^k_{\overline{d}+\overline{\delta}-\underline{d}})$, then $s_k + \tau_{k-\underline{d}-1} - \sum_{l=k-\underline{d}-1}^{k-1} h_l$ is the time at which $u_{k-\underline{d}-1}$ is available at the plant. In case $\tau_{k-\underline{d}-1} - \sum_{l=k-\underline{d}}^{k-1} h_l < 0$, then $u_{k-\underline{d}-1}$ might be active before s_k. Since, we are only interested, here, in the interval $[s_k, s_{k+1})$, we take the maximum of this value and zero in (40). For the other values of t_j^k, the recursion can be derived similarly, which leads to

$$t_j^k = \min\left[t_{j+1}^k, \max\left\{0, \tau_{k+j-\overline{d}-\overline{\delta}} - \sum_{l=k+j-\overline{d}-\overline{\delta}}^{k-1} h_l\right\}\right.$$
$$\left. + m_{k+j-\overline{d}-\overline{\delta}} h_{\max}\right],$$

for $0 \leq j \leq \overline{d}+\overline{\delta}-\underline{d}$, $m_{k+j-\overline{d}-\overline{\delta}}$ satisfying (4), and with $t_{\overline{d}+\overline{\delta}-\underline{d}+1}^k := h_k$. The evaluation of this recursive relation yields the explicit characterization of (7).

2.1.8 Appendix B: Proof of Lemma 3

Suppose that (18) holds for some matrices $P^T(\theta) = P(\theta) > 0, \forall \theta \in \Theta$. Decompose the matrices as follows:

$$P(\theta) = \begin{pmatrix} P_1(\theta) & P_2(\theta) \\ P_2^T(\theta) & P_3(\theta) \end{pmatrix}$$

in accordance with the matrix $\bar{A}(\theta)$. By expanding (18) we obtain for all $\theta_1, \theta_2 \in \Theta$ that

$$\begin{pmatrix} \begin{bmatrix} A^T(\theta_1)P_1(\theta_2)A(\theta_1) - P_1(\theta_1) + R^T P_2(\theta_2)A(\theta_1) \\ + A^T(\theta_1)P_2(\theta_2)R + R^T P_3(\theta_2)R \end{bmatrix} & -P_2(\theta_1) \\ -P_2^T(\theta_1) & -P_3(\theta_1) \end{pmatrix} < 0.$$

This is equivalent (using the Schur complement lemma) to

$$\begin{aligned} &A^T(\theta_1)P_1(\theta_2)A(\theta_1) - P_1(\theta_1) + R^T P_2(\theta_2)A(\theta_1) + \\ &A^T(\theta_1)P_2(\theta_2)R + R^T P_3(\theta_2)R + \\ &P_2(\theta_1)P_3^{-1}(\theta_1)P_2^T(\theta_1) < 0. \end{aligned}$$

Adding and subtracting

$$A^T(\theta_1)P_2(\theta_2)P_3^{-1}(\theta_2)P_2^T(\theta_2)A(\theta_1)$$

to the previous inequality implies for all $\theta_1, \theta_2 \in \Theta$ that

$$A^T(\theta_1)Q(\theta_2)A(\theta_1) - Q(\theta_1) + W(\theta_1, \theta_2) < 0,$$

where

$$Q(\theta) = P_1(\theta) - P_2(\theta)P_3^{-1}(\theta)P_2^T(\theta), \ \forall\, \theta \in \Theta \tag{41}$$

and

$$\begin{aligned} W(\theta_1, \theta_2) \ &= \ (P_2(\theta_2)A(\theta_1) + P_3(\theta_2)R)^T \times P_3^{-1}(\theta_2) \\ &\quad \times (P_2(\theta_2)A(\theta_1) + P_3(\theta_2)R). \end{aligned}$$

As $W(\theta_1, \theta_2) \geq 0$ and $Q(\theta) > 0$ (since $P_3(\theta) > 0$ and $Q(\theta)$ is the Schur complement of $P(\theta)$), clearly $Q(\theta)$, $\theta \in \Theta$, satisfy condition (19). Notice that when the matrices $P(\theta)$ are constant, i.e.

$$P(\theta) = P = \begin{pmatrix} P_1 & P_2 \\ P_2^T & P_3 \end{pmatrix}, \ \forall \theta \in \Theta,$$

the corresponding matrices $Q(\theta)$ as in (41) that satisfy (19) are constant as well:

$$Q(\theta) = Q = P_1 - P_2 P_3^{-1} P_2^T, \ \theta \in \Theta.$$

To prove the converse, assume that (19) holds. Then, due to the continuity of A with respect to θ and to the compactness of Θ, there exists $\epsilon > 0$ such that for all $\theta_1, \theta_2 \in \Theta$

$$\begin{pmatrix} A^T(\theta_1)Q(\theta_2)A(\theta_1) - Q(\theta_1) + \epsilon R^T R & 0 \\ 0 & -\epsilon I \end{pmatrix} < 0.$$

This inequality shows that the matrices $P(\theta)$ defined as

$$P(\theta) = \begin{pmatrix} Q(\theta) & 0 \\ 0 & \epsilon I \end{pmatrix} > 0,\ \theta \in \Theta$$

satisfy (18).

Clearly, when $Q(\theta) = Q,\ \theta \in \Theta$, the common matrix

$$P(\theta) = P = \begin{pmatrix} Q & 0 \\ 0 & \epsilon I \end{pmatrix} > 0,\ \theta \in \Theta$$

satisfies the inequality (18), which completes the proof.

2.2 LMI-Based Synthesis of Decentralized Controllers

D. Barcelli, D. Bernardini, A. Bemporad

2.2.1 Introduction

Networked control systems (NCSs) are characterized by a topological distribution over the physical space that sometimes prevents the use of centralized control solutions. In fact, the set of measurements might not be available at each control instant, due for instance to temporarily or permanently faulty sensors connections. A natural workaround is to define a set of controllers, each one in charge of commanding only a subset of actuators. The underlying idea is that the information provided by a subset of sensor measurements might be enough to control a subset of actuators satisfactorily. In this case, a decentralized control scheme clearly reduces the communication traffic over the network, allowing for a simpler network structure.

These considerations led, since the 70's, to look with interest to decentralized control, mainly investigating stability properties [87, 111]. In the 90's, the rise of convex optimization techniques allowed for convex formulations of decentralized control problems [88, 89], achieving results for some particular classes of systems, such as spatially invariant systems [90]. Decentralized estimation and control schemes based on distributed convex optimization ideas have been proposed recently by means of Lagrangian relaxations [91, 108], where global solutions are achieved after a (possibly large) number of inter-agent communications. Hence, looking at a real implementation, the sample time must be set conservatively high in order to let all the agreements to conclude without having consequences on the control action. Moreover, the need of mutual exchange of information between network agents produces an overhead in the communication channel which must be taken into account when dealing with network-related issues such as delay and packet loss.

In this work we present an approach for the off-line synthesis of a set of decentralized linear regulators for discrete-time linear systems subject to input and state constraints. Measurements are provided by a distributed set of sensors to a distributed set of actuators through a network connection, in which some of the links are subject to random packet dropout. We aim at enforcing stability of the closed-loop system for every possible combination of packet losses that can occur in the network at every time step. Conservativeness of the resulting control law is reduced by using a different set of local control laws for every possible network configuration, without the need of communication among different controllers. Moreover, we take into account a model of packet dropouts based on a finite-state Markov chain, in order to exploit available knowledge about the stochastic nature of the network, and improve the closed-loop performance.

In the last years, mean-square stability of networked control systems (NCSs) has been often analyzed in literature. For example, in [92] a stabilizing controller for linear systems subject to random but bounded delays in the feedback loop is designed by augmenting the state vector and modeling the overall process as a Markov jump linear system. A NCS subject to communication constraints is studied in [155], where a Markov model is used to represent the dynamics of the transmission update times, and stability is guaranteed by means of a stochastic quadratic Lyapunov function. More recently, a framework to analyze stability of stochastic linear NCSs subject to time-varying transmission intervals, delays, packet dropouts and communication constraints by means of overapproximation methods has been proposed in [93]. Most of these works (if not all, and this work makes no exception) rely on convex optimization, and more specifically on the formulation of optimization problems constrained by a set of linear matrix inequalities (LMIs) [94, 95].

2.2.2 Control over ideal networks

Consider the discrete-time time-invariant linear system

$$x(t+1) = Ax(t) + Bu(t), \tag{42}$$

where $x = [x_1, \dots, x_n]' \in \mathbb{R}^n$ is the state, $u = [u_1, \dots, u_m]' \in \mathbb{R}^m$ is the input, $t \in \mathbb{N}_0$ is the time index, and the matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$. We assume that states and inputs are subject to the constraints[1]

$$\|x(t)\|_2 \leq x_{\max},\ \|u(t)\|_2 \leq u_{\max},\ \forall t \in \mathbb{N}_0. \tag{43}$$

The process we consider is a networked control system, where spatially distributed sensor nodes provide measurements of the system state, and spatially distributed actuator nodes implement the control action. More in detail, at every time step t every sensor $s_1, \dots, s_n$ measures a component $x_i(t)$ of the state vector, $i = 1, \dots, n$. Then, measurements are sent to actuators $a_1, \dots, a_m$ through a user-defined networked connection. Given a process of the form (42) we define its *network topology* by means of an adjacency matrix $\Lambda \in \{0, 1\}^{m\times n}$ with elements

$$\lambda_{ij} = \begin{cases} 1 & \text{if sensor } s_j \text{ is linked to actuator } a_i, \\ 0 & \text{otherwise.} \end{cases} \tag{44}$$

for $i = 1, \dots, m$ and $j = 1, \dots, n$. In other words, $\lambda_{ij} = 1$ if and only if the measurement $x_j(t)$ can be exploited to compute the input signal $u_i(t)$, $\forall t \in \mathbb{N}_0$. We assume here that all the network links are *ideal* (no packet dropout, delays, etc.); the decentralized control problem is extended to consider packet dropouts in Section 2.2.3.

Linear controller synthesis

Our goal is to find a gain matrix $K \in \mathbb{R}^{m\times n}$ such that the system (42) in closed-loop with

$$u(t) = Kx(t) \tag{45}$$

is asymptotically stable. The desired control law must be decentralized, i.e., each actuator $a_1, \dots, a_m$ can only exploit the measurements that are available in accordance with the network topology (44). In other words, each row i of K can only have non-zero elements in correspondence with the state measurements available to actuator a_i, $i = 1, \dots, m$. This imposes the following structure on K:

$$\lambda_{ij} = 0 \ \Rightarrow\ k_{ij} = 0, \qquad i = 1, \dots, m,\ j = 1, \dots, n, \tag{46}$$

where k_{ij} is the (i, j)-th element of K.

Closed-loop stability is enforced through the condition

$$V(x(t+1)) - V(x(t)) \leq -x(t)'Q_x x(t) - u(t)'Q_u u(t), \tag{47}$$

where $V : \mathbb{R}^n \to \mathbb{R}$ is a Lyapunov function of the state x for the closed-loop system given by (42) and (45), and $Q_x \in \mathbb{R}^{n\times n}$, $Q_u \in \mathbb{R}^{m\times m}$ are weight matrices, with $Q_x = Q_x{}' \succ 0$, $Q_u = Q_u{}' \succeq 0$. In the following we consider quadratic Lyapunov functions, and define

$$V(x) \triangleq x'Px, \tag{48}$$

with $P \in \mathbb{R}^{n\times n}$, $P = P' \succ 0$. It is well known that satisfaction of (47) for all time steps $t \in \mathbb{N}_0$ implies asymptotical stability of the closed-loop system (see, e.g., [96]). If (47) is satisfied, then we can show that

$$V(x(t)) \geq J_\infty(t) \triangleq \sum_{i=0}^{\infty} \left(x(t+i)'Q_x x(t+i) + u(t+i)'Q_u u(t+i) \right),$$

[1] Other kinds of constraints, such as element-wise bounds, can be considered in a similar fashion (see, e.g., [96]).

i.e., $V(x(t))$ is an upper bound of the infinite-horizon quadratic cost-to-go $J_\infty(t)$ defined by Q_x, Q_u [96]. Our goal is to find the smallest scalar $\gamma > 0$ such that

$$x(t)'Px(t) \leq \gamma,\ \forall t \in \mathbb{N}_0, \tag{49}$$

or, equivalently, $x(t)'Q^{-1}x(t) \leq 1$, $\forall t \in \mathbb{N}_0$, by substituting $Q = \gamma P^{-1}$. Clearly, the satisfiability of (49) depends on the initial state $x(0)$. Rather than finding the proper value of γ for a given initial state $x(0) \in \mathbb{R}^n$, we look for a γ which is valid for all $x(0) \in \mathcal{X}_0 \subset \mathbb{R}^n$, where $\mathcal{X}_0 \triangleq \mathcal{H}(v_1, \ldots, v_{n_v})$ is a polytope with vertices $v_1, \ldots, v_{n_v}$, and $\mathcal{H}(\cdot)$ denotes the convex hull operator, so that the controller K that we are going to synthesize is valid for any initial condition $x(0) \in \mathcal{X}_0$. As noted in [95], by making the standard substitution $K = YQ^{-1}$, $Y \in \mathbb{R}^{m \times n}$, we can obtain any desired structure for K by imposing the same structure on Y and fixing the block-diagonal structure of Q

$$(\lambda_{ij} = 0) \Rightarrow y_{ij} = 0,\ i = 1, \ldots, m,\ j = 1 \ldots, n, \tag{50a}$$

$$(\lambda_{ij} = 0) \wedge (\lambda_{ih} = 1)\ \Rightarrow\ q_{hj} = 0,\ q_{jh} = 0, \quad i = 1, \ldots, m,\ j = 1, \ldots, n,\ h = 1, \ldots, n, \tag{50b}$$

where $\wedge$ denotes logical "and".

Theorem 3. *Consider an ideal network with topology $\Lambda \in \{0, 1\}^{m \times n}$ and let $P = \gamma Q^{-1}$, $K = YQ^{-1}$ be obtained by solving the semidefinite programming (SDP) problem*

$$\min_{\gamma, Q, Y}\ \gamma \tag{51a}$$

$$\textit{s.t.}\ \begin{bmatrix} Q & \star & \star & \star \\ AQ{+}BY & Q & \star & \star \\ Q_x^{1/2}Q & \mathbf{0} & \gamma I_n & \star \\ Q_u^{1/2}Y & \mathbf{0} & \mathbf{0} & \gamma I_m \end{bmatrix} \succeq 0, \tag{51b}$$

$$\begin{bmatrix} Q & \star \\ AQ{+}BY & x_{\max}^2 I_n \end{bmatrix} \succeq 0, \tag{51c}$$

$$\begin{bmatrix} u_{\max}^2 I_m & \star \\ Y & Q \end{bmatrix} \succeq 0, \tag{51d}$$

$$\begin{bmatrix} 1 & \star \\ v_i & Q \end{bmatrix} \succeq 0,\ i = 1, \ldots, n_v, \tag{51e}$$

$$Y \in \mathcal{Y},\ Q \in \mathcal{Q}, \tag{51f}$$

where I_n is the identity matrix in $\mathbb{R}^{n \times n}$, $\mathbf{0}$ is a matrix of appropriate dimension with all zero entries,

$$\mathcal{Q} \triangleq \{Q \in \mathbb{R}^{n \times n} : q_{hj} = q_{jh} = 0 \textit{ if } (\lambda_{ij} = 0) \wedge (\lambda_{ih} = 1),\ i = 1, \ldots, m,\ j = 1, \ldots, n,\ h = 1, \ldots, n\},$$

$$\mathcal{Y} \triangleq \{Y \in \mathbb{R}^{m \times n} : y_{ij} = 0 \textit{ if } \lambda_{ij} = 0,\ i = 1, \ldots, m,\ j = 1 \ldots, n\},$$

and q_{ij}, y_{ij} are the (i, j)-th elements of Q and Y, respectively. If problem (51) *is feasible, then system* (42) *with initial state $x(0) \in \mathcal{X}_0$ in closed-loop with the decentralized constant feedback control law* (45) *is asymptotically stable and satisfies the constraints* (43)*.*

Proof. In the particular case where $\mathcal{X}_0 = \{x_0\}$ is a singleton (i.e., the initial state $x(0) = x_0$ is fixed), and $\Lambda = \mathbf{1}_{m \times n}$ is a matrix with all one entries (i.e., the control law is centralized and we have no constraints on the structure of K), asymptotical stability is a well known result that follows by showing that $V(x(t)) = x(t)'Px(t)$ is a Lyapunov function for the closed-loop system (see, e.g., [94, 96]). Substituting $P = \gamma Q^{-1}$, condition (47) is converted by means of Schur complements to the LMI (51b). Using similar arguments, state and input constraints (43) are enforced by (51c) and (51d). It remains to prove (i) that stability is retained for every initial state $x(0) \in \mathcal{X}_0$ when $\mathcal{X}_0$ has dimension greater than 0, and (ii) that the control law $u(t) = Kx(t)$, with $K = YQ^{-1}$, can be implemented in a decentralized way, according to the network topology Λ. We see that (i) follows by convexity of the ellipsoid $\mathcal{E}_Q \triangleq \{x \in \mathbb{R}^n :\ x'Q^{-1}x \leq 1\}$. In fact, since $\mathcal{E}_Q$ contains the vertices v_i of $\mathcal{X}_0$ due to (51e), then $\mathcal{X}_0 \subset \mathcal{E}_Q$. Regarding (ii), as the structure of diagonal blocks is preserved by matrix inversion, Q block diagonal implies that Q^{-1} is also block-diagonal, and hence (50b) implies that $\tilde{q}_{hj} = 0$ and $\tilde{q}_{jh} = 0$, for all i, j, h such that $\lambda_{ij} = 0$ and $\lambda_{ih} = 1$, where $\tilde{q}_{hj}$ is the (h, j)-th element of Q^{-1}. Since $k_{ij} = \sum_{h=1}^{n} y_{ih}\tilde{q}_{hj}$, by (51f) it follows that the decentralized structure (46) is satisfied. □

2.2.3 Control over lossy networks

In this section we consider packet dropouts occurring in some of the links of the communication network, referred to as *lossy* links. To account for the presence of lossy links in the network, we extend the definition of the topology $\Lambda \in \{-1, 0, 1\}^{m\times n}$ as follows

$$\lambda_{ij} = \begin{cases} 1 & \text{if ideal link between } s_j \text{ and } a_i, \\ -1 & \text{if lossy link between } s_j \text{ and } a_i, \\ 0 & \text{if no link between } s_j \text{ and } a_i, \end{cases} \tag{52}$$

for $i = 1, \dots, m$, $j = 1, \dots, n$. No probabilistic model of packet loss is considered here; this will be introduced in Section 2.2.4.

We denote with l_i the number of lossy links connected with actuator a_i, $i = 1, \dots, m$ (i.e., the number of "-1" in the ith row of Λ), and with $L = \sum_{i=1}^{m} l_i$ the total number of lossy links in the network. Then, we can enumerate all the possible combinations of packet dropouts at a given time step t by replacing every "-1" in Λ with either a "1" or a "0". In this way we obtain a set of $\ell = 2^L$ matrices $\tilde{\Lambda}_h \in \{0, 1\}^{m\times n}$, $h = 1, \dots, \ell$, which describes all the possible network configurations. We denote by $\tilde{\Lambda}(t) \in \left\{\tilde{\Lambda}_1, \dots, \tilde{\Lambda}_\ell\right\}$ the network configuration at time $t \in \mathbb{N}_0$.

Switching controller synthesis

We want to design a set of gains $K_h \in \mathbb{R}^{m\times n}$, $h = 1, \dots, \ell$, to be used in the decentralized switching feedback control law

$$u(t) = \begin{cases} K_1 x(t) & \text{if } \tilde{\Lambda}(t) = \tilde{\Lambda}_1, \\ K_2 x(t) & \text{if } \tilde{\Lambda}(t) = \tilde{\Lambda}_2, \\ \vdots & \quad \vdots \\ K_\ell x(t) & \text{if } \tilde{\Lambda}(t) = \tilde{\Lambda}_\ell. \end{cases} \tag{53}$$

Note that in general the implementation of (53) requires the controllers to be aware of the whole network status $\tilde{\Lambda}(t)$. This hypothesis is obviously not realistic in a decentralized framework. Hence, we impose an appropriate structure of the gains $K_1, \dots, K_\ell$, so that every local actuator a_i, $i = 1, \dots, m$, needs only to know which *local* measurements have been lost, regardless of the links status in the rest of the network. To accomplish this, we need to have a control law which univocally defines $u_i(t)$, $\forall i$, for all the network configurations $\tilde{\Lambda}_h$ that have identical values in their ith row. Namely, $[M]_i$ being the ith row of a generic matrix M, we want to impose

$$[\tilde{\Lambda}_h]_i = [\tilde{\Lambda}_j]_i \quad \Rightarrow \quad [K_h]_i = [K_j]_i, \tag{54}$$

for all $h, j = 1, \dots, \ell$, $i = 1, \dots, m$. This relation greatly reduces the number of variables to be considered in our optimization problem. In fact, we have only 2^{l_i} possible values of $[\tilde{\Lambda}(t)]_i$, $i = 1, \dots, m$. We refer to these row vectors as $\Gamma_1^i, \dots, \Gamma_{2^{l_i}}^i$, where $\Gamma_j^i \in \{0, 1\}^{1\times n}$, $\forall i, j$. Hence, we look for $\sum_{i=1}^{m} 2^{l_i}$ local gains $F_1^i, \dots, F_{2^{l_i}}^i$ which define the set of element-wise feedback control laws

$$u_i(t) = \begin{cases} F_1^i x(t) & \text{if } [\tilde{\Lambda}(t)]_i = \Gamma_1^i, \\ F_2^i x(t) & \text{if } [\tilde{\Lambda}(t)]_i = \Gamma_2^i, \\ \vdots & \quad \vdots \\ F_{2^{l_i}}^i x(t) & \text{if } [\tilde{\Lambda}(t)]_i = \Gamma_{2^{l_i}}^i, \end{cases} \tag{55}$$

for all $i = 1, \dots, m$. These local gains $\{F_j^i\}$ are then combined to obtain the ℓ global gains $\{K_h\}$ used in (53). Our purpose is to guarantee the satisfaction of the stability constraint (47) in the presence of random packet dropouts. We are looking for a robust kind of stability, where no information on the dynamics regulating the evolution in time of $\tilde{\Lambda}(t)$ are exploited. Hence, here we take $V(x)$ in (48) as a common Lyapunov function for the switching closed-loop dynamics $x(t+1) = (A + BK_h)x(t)$, $h = 1, \dots, \ell$. In order to compute $K_1, \dots, K_\ell$ we substitute

$$K_h = Y_h Q^{-1}, \ \forall h, \tag{56}$$

allowing a different matrix $Y_h \in \mathbb{R}^{m\times n}$ for every possible network configuration $\tilde{\Lambda}_h$. However, since we have a unique Q, it must hold

$$[Y_h Q^{-1}]_i \;=\; [Y_j Q^{-1}]_i,\ \forall i, j, h, \tag{57}$$

in order to satisfy (54). In other words, the structure of Q needs to preserve the structure of every K_h, $h = 1, \ldots, \ell$.

Theorem 4. *Consider a network with topology $\Lambda \in \{-1, 0, 1\}^{m\times n}$, and let $K_h = Y_h Q^{-1}$, $h = 1, \ldots, \ell$, be obtained by solving the SDP problem*

$$\min_{\gamma, Q, \{Y\}} \ \gamma \tag{58a}$$

$$\text{s.t.} \begin{bmatrix} Q & \star & \star & \star \\ AQ+BY_h & Q & \star & \star \\ Q_x^{1/2}Q & \mathbf{0} & \gamma I_n & \star \\ Q_u^{1/2}Y_h & \mathbf{0} & \mathbf{0} & \gamma I_m \end{bmatrix} \succeq 0,\ h = 1, \ldots, \ell, \tag{58b}$$

$$\begin{bmatrix} Q & \star \\ AQ+BY_h & x_{\max}^2 I_n \end{bmatrix} \succeq 0,\ h = 1, \ldots, \ell, \tag{58c}$$

$$\begin{bmatrix} u_{\max}^2 I_m & \star \\ Y_h & Q \end{bmatrix} \succeq 0,\ h = 1, \ldots, \ell, \tag{58d}$$

$$\begin{bmatrix} 1 & \star \\ v_i & Q \end{bmatrix} \succeq 0,\ i = 1, \ldots, n_v, \tag{58e}$$

$$[\tilde{\Lambda}_h]_i = [\tilde{\Lambda}_j]_i \ \Rightarrow\ [Y_h]_i = [Y_j]_i,\ \ h, j = 1, \ldots, \ell,\ i = 1, \ldots, m, \tag{58f}$$

$$Y_h \in \tilde{\mathcal{Y}}_h,\ h = 1, \ldots, \ell, \tag{58g}$$

$$Q \in \tilde{\mathcal{Q}}, \tag{58h}$$

where

$$\tilde{\mathcal{Q}} \triangleq \{Q \in \mathbb{R}^{n\times n} : q_{wj} = q_{jw} = 0 \textit{ if } (\tilde{\lambda}_{ij}^h = 0) \wedge (\tilde{\lambda}_{iw}^h = 1),\ \ i = 1, \ldots, m,\ j, w = 1, \ldots, n,\ h = 1, \ldots, \ell\},$$
$$\tilde{\mathcal{Y}}_h \triangleq \{Y_h \in \mathbb{R}^{m\times n} : y_{ij}^h = 0 \textit{ if } \tilde{\lambda}_{ij}^h = 0,\ \ i = 1, \ldots, m,\ j = 1 \ldots, n\},$$

y_{ij}^h is the (i, j)-th element of Y_h, and $\tilde{\lambda}_{ij}^h$ is the (i, j)-th element of $\tilde{\Lambda}_h$, for all $h = 1, \ldots, \ell$. If problem (58) *is feasible, then system* (42) *with initial state $x(0) \in \mathcal{X}_0$ in closed loop with the decentralized switching feedback control law* (53) *is asymptotically stable and satisfies the constraints* (43) *for any possible realization of packet dropout.*

Proof. Constraints (58b), obtained by using (53), substituting $K_h = Y_h Q^{-1}$, $h = 1, \ldots, \ell$, and taking a Schur complement, are a sufficient condition to the satisfaction of (47) for every $\tilde{\Lambda}(t) \in \{\tilde{\Lambda}_1, \ldots, \tilde{\Lambda}_\ell\}$, $\forall t$. Hence robust asymptotical stability is provided for every possible realization of packet dropout. Fulfillment of state and input constraints (43) and robustness with respect to the initial state $x(0) \in \mathcal{X}_0$, which are respectively enforced by (58c)–(58d) and (58e), follow by similar arguments of Theorem 3. It remains to prove that the resulting control law (53) can be implemented in a decentralized way as a combination of (55), $\forall i$. We fix the structure of Y_h to be equal to the structure of K_h, $\forall h$, by means of (58g). Since $K_h = Y_h Q^{-1}$, we have $k_{ij}^h = \sum_{w=1}^n y_{iw}^h \tilde{q}_{wj}$ and we must enforce the counterpart of (46) for the case of a switching control law, i.e., $\tilde{\lambda}_{ih}^h = 0 \ \Rightarrow\ k_{ij}^h = 0$, or equivalently

$$\tilde{\lambda}_{ij}^h = 0 \ \Rightarrow\ \sum_{w=1}^n y_{iw}^h \tilde{q}_{wj} = 0,\ \forall i, j, h. \tag{59}$$

Being the structure of Y_h assigned for a fixed h, a sufficient condition for the satisfaction of (59) is given by

$$(\tilde{\lambda}_{ij}^h = 0) \wedge (\tilde{\lambda}_{iw}^h = 1) \ \Rightarrow\ \tilde{q}_{wj} = 0,\ \forall i, j, w, h,$$

which is enforced by (58h) noting that Q is symmetric and block-diagonal. Finally, constraint (58f) together with (56) implies satisfaction of (57), which is a sufficient condition for (54) to hold. This ensures the uniqueness of the local control law (55) to be implemented given $[\tilde{\Lambda}(t)]_i$, $\forall i$, regardless of the global value of $\tilde{\Lambda}(t)$, and proves (53) to be a decentralized control law with the requested structure. □

2.2.4 Stochastic control under packet dropout

The robust approach undertaken in the previous section can be conservative in some cases, as it requires the existence of a common Lyapunov function which must be decreasing at every time step for every possible network configuration. In this section we pursue a relaxed stability condition by introducing a probabilistic model of the network and exploiting the possibly available stochastic information on packet dropout. We consider stability in the mean-square sense, which in this framework is equivalent to

$$\lim_{t\to\infty} \mathbb{E}\left[\|x(t)\|^2\right] = 0. \tag{60}$$

In other words, we allow the closed-loop Lyapunov function to occasionally increase from one step to another, as long as a decreasing condition of the form (47) is guaranteed to hold on average. The expectation is taken with respect to the realizations of $\tilde{\Lambda}(t)$, which is now modeled as a stochastic process.

Stochastic network model

Following the model proposed in [97], we assume that the probability distribution of the network configurations $\{\tilde{\Lambda}_h\}$ is modeled by a finite-state Markov chain with 2 states[2], called Z_1 and Z_2. The dynamics of the Markov chain are defined by a transition matrix

$$T = \begin{bmatrix} q_1 & 1-q_1 \\ 1-q_2 & q_2 \end{bmatrix} \tag{61}$$

such that $t_{ij} = \Pr[z(t+1) = Z_j \,|\, z(t) = Z_i]$, and by an emission matrix $E \in \mathbb{R}^{2\times 2^L}$ such that $e_{ij} = \Pr[\tilde{\Lambda}(t) = \tilde{\Lambda}_j \,|\, z(t) = Z_i]$, being t_{ij} and e_{ij} the (i,j)-th element of T and E, respectively. In order to define the values in E we need to compute the probabilities of occurrence of $\tilde{\Lambda}_h$, $\forall h$. We assume that the occurrence of a packet dropout at a time step t in a given network link is an i.i.d. random variable, for every state of the Markov chain. In particular, we denote with d_1 and d_2, $0 < d_1 < d_2 < 1$, the probabilities of losing a packet at time t if $z(t) = Z_1$ and $z(t) = Z_2$, respectively (for example, in Z_1 we have "few" dropouts, and in Z_2 we have "many", according to Gilbert's model). Moreover, let $s_{1,h}$ and $s_{0,h}$ be the total number of lossy links in Λ which are mapped as ideal links and as no links in $\tilde{\Lambda}_h$, respectively, i.e.,

$$s_{1,h} = \sum_{(i,j)\in\mathcal{I}} \tilde{\lambda}_{ij}^h, \quad h = 1,\ldots,\ell, \tag{62a}$$

$$s_{0,h} = \sum_{(i,j)\in\mathcal{I}} (1-\tilde{\lambda}_{ij}^h), \quad h = 1,\ldots,\ell, \tag{62b}$$

where $\mathcal{I} \triangleq \{(i,j) \in \{1,\ldots,m\} \times \{1,\ldots,n\} : \lambda_{ij} = -1\}$. Then, we can define the elements $\{e_{ij}\}$ of E as

$$e_{ij} = {d_i}^{s_{0,j}}(1-d_i)^{s_{1,j}}, \; i = 1,2, \; j = 1,\ldots,\ell. \tag{63}$$

2.2.5 Stochastic switching controller synthesis

Our goal is to design two sets of control gains $K_{1,1},\ldots,K_{1,\ell}$, $K_{2,1},\ldots,K_{2,\ell}$, one for every state of the Markov chain, which define the switching control law

$$u(t) = \begin{cases} K_{1,1}x(t) & \text{if } z(t) = Z_1, \; \tilde{\Lambda}(t) = \tilde{\Lambda}_1, \\ \quad\vdots & \qquad\qquad\vdots \\ K_{1,\ell}x(t) & \text{if } z(t) = Z_1, \; \tilde{\Lambda}(t) = \tilde{\Lambda}_\ell, \\ K_{2,1}x(t) & \text{if } z(t) = Z_2, \; \tilde{\Lambda}(t) = \tilde{\Lambda}_1, \\ \quad\vdots & \qquad\qquad\vdots \\ K_{2,\ell}x(t) & \text{if } z(t) = Z_2, \; \tilde{\Lambda}(t) = \tilde{\Lambda}_\ell, \end{cases} \tag{64}$$

[2] More complex Markov chain models of packet loss could be considered here, such as the one used in [98].

so that the closed-loop system is asymptotically stable in mean-square. Consider the stochastic counterpart of the decreasing condition (47)

$$\mathbb{E}\left[V(x(t+1))\right] - V(x(t)) \leq -x(t)'Q_x x(t) - \mathbb{E}[u(t)'Q_u u(t)]. \tag{65}$$

As shown in [156], fulfillment of (65) for all $t \in \mathbb{N}_0$ implies (60). Here $V(x)$ is intended to be a switching stochastic Lyapunov function for the closed-loop system, defined as

$$V(x(t)) \triangleq \begin{cases} x(t)'P_1 x(t) & \text{if } z(t) = Z_1, \\ x(t)'P_2 x(t) & \text{if } z(t) = Z_2. \end{cases} \tag{66}$$

We assume that the state $z(t) = Z_j$ of the communication network is known[3] at time t, with $j \in \{1,2\}$. Hence, the expectations in (65) are

$$\mathbb{E}\left[V(x(t+1))\right] = \sum_{h=1}^{\ell}\sum_{l=1}^{2} e_{jh}t_{jl}x(t)'(A+BK_{j,h})'P_l\left(A+BK_{j,h}\right)x(t), \tag{67a}$$

$$\mathbb{E}\left[u(t)'Q_u u(t)\right] = \sum_{h=1}^{\ell} e_{jh}x(t)'K_{j,h}'Q_u K_{j,h}x(t). \tag{67b}$$

In light of the above considerations, by using (64), (66) and (67), and substituting $P_j = \gamma Q_j{}^{-1}$, $K_{j,h} = Y_{j,h}Q_j{}^{-1}$, $\forall j,h$, we can translate (65) to an appropriate LMI condition with standard methods, as detailed in the following theorem.

Theorem 5. *Consider a network with topology* $\Lambda \in \{-1,0,1\}^{m\times n}$*, where at each time step the packet dropout realizations are driven by the Markov chain defined by* (61)–(63)*, and let* $K_{j,h} = Y_{j,h}Q^{-1}$*,* $j = 1,2$*,* $h = 1,\ldots,\ell$*, be obtained by solving the SDP problem*

$$\min_{\gamma,\{Q\},\{Y\}} \quad \gamma \tag{68a}$$

$$\text{s.t.} \begin{bmatrix} Q_j & \star & \star & \star \\ Q_x^{1/2}Q_j & \gamma I_n & \star & \star \\ C_{j,1} & \mathbf{0} & D_{j,1} & \star \\ C_{j,2} & \mathbf{0} & \mathbf{0} & D_{j,2} \end{bmatrix} \succeq 0, \tag{68b}$$

$$\begin{bmatrix} Q_j & \star \\ AQ_j + BY_{j,h} & x_{\max}^2 I_n \end{bmatrix} \succeq 0,\ h = 1,\ldots,\ell, \tag{68c}$$

$$\begin{bmatrix} u_{\max}^2 I_m & \star \\ Y_{j,h} & Q_j \end{bmatrix} \succeq 0,\ h = 1,\ldots,\ell, \tag{68d}$$

$$\begin{bmatrix} 1 & \star \\ v_i & Q_j \end{bmatrix} \succeq 0,\ i = 1,\ldots,n_v, \tag{68e}$$

$$[\tilde{\Lambda}_h]_i = [\tilde{\Lambda}_w]_i \ \Rightarrow\ [Y_{j,h}]_i = [Y_{j,w}]_i, \quad h,w = 1,\ldots,\ell,\ i = 1,\ldots,m, \tag{68f}$$

$$Y_{j,h} \in \tilde{\mathfrak{Y}}_{j,h},\ h = 1,\ldots,\ell, \tag{68g}$$

$$Q_j \in \tilde{\mathcal{Q}}_j \tag{68h}$$

where $j = 1,2$*,*

$$\tilde{\mathcal{Q}}_j \triangleq \{Q_j \in \mathbb{R}^{n\times n} : q_{lw}^j = 0, q_{wl}^j = 0 \text{ if } (\tilde{\lambda}_{iw}^h = 0) \wedge (\tilde{\lambda}_{il}^h = 1),\ w,l = 1,\ldots,n,$$
$$i = 1,\ldots,m,\ h = 1,\ldots,\ell,\},\ j = 1,2,$$

$$\tilde{\mathfrak{Y}}_{j,h} \triangleq \{Y_{j,h} \in \mathbb{R}^{m\times n} : y_{iw}^{j,h} = 0 \text{ if } \tilde{\lambda}_{iw}^h = 0,\ i = 1,\ldots,m,\ w = 1\ldots,n\},\ \text{for all } j = 1,2,\ h = 1,\ldots,\ell,$$

q_{lw}^j *is the* (l,w)*-th element of* Q_j*,* $y_{iw}^{j,h}$ *is the* (i,w)*-th element of* $Y_{j,h}$*, and*

$$C_{j,1} = \begin{bmatrix} \sqrt{e_{j1}t_{j1}}(AQ_j+BY_{j,1}) \\ \vdots \\ \sqrt{e_{j\ell}t_{j1}}(AQ_j+BY_{j,\ell}) \\ \sqrt{e_{j1}t_{j2}}(AQ_j+BY_{j,1}) \\ \vdots \\ \sqrt{e_{j\ell}t_{j2}}(AQ_j+BY_{j,\ell}) \end{bmatrix},\ C_{j,2} = \begin{bmatrix} \sqrt{e_{j1}}(Q_u^{1/2}Y_{j,1}) \\ \vdots \\ \sqrt{e_{j\ell}}(Q_u^{1/2}Y_{j,\ell}) \end{bmatrix},$$

[3] In practice, one should estimate the state $z(t)$ of the communication network, see, e.g., [99].

$$D_{j,1} = \textit{Blkdiag}\{\underbrace{Q_1,\ \ldots,\ Q_1}_{\ell\ \textit{times}},\ \underbrace{Q_2,\ \ldots, Q_2}_{\ell\ \textit{times}}\},$$

$$D_{j,2} = \textit{Blkdiag}\{\underbrace{\gamma I_m,\ \gamma I_m,\ \ldots,\ \gamma I_m}_{\ell\ \textit{times}}\}.$$

If problem (68) *is feasible, then system* (42) *with initial state* $x(0) \in \mathcal{X}_0$ *in closed-loop with the decentralized switching feedback control law* (64) *is asymptotically stable in mean-square.*

Proof. Constraints (68b), obtained by using (64), substituting $K_{j,h} = Y_{j,h}Q^{-1}$, $j = 1, 2$, $h = 1, \ldots, \ell$, and taking a Schur complement, are a sufficient condition to the satisfaction of (65) for every $\tilde{\Lambda}(t) \in \{\tilde{\Lambda}_1, \ldots, \tilde{\Lambda}_\ell\}$, $\forall t$, distributed as modeled by (61)–(63). Hence asymptotical closed-loop stability in mean-square is provided. Robustness with respect to the initial state $x(0) \in \mathcal{X}_0$ and desired decentralized structure of the switching feedback control law (64), which are respectively enforced by (68e) and (68g)–(68h), follow by similar reasonings as in Theorems 3–4. The uniqueness of the switching feedback control law (64) is imposed by constraints (68f) for every state $z(t) \in \{Z_1,\ Z_2\}$ of the Markov chain, similarly to what proved in Theorem 4 for the case of a single Q and a single set of gains $\{Y_h\}$. Since by assumption the current Markov chain state $z(t)$ at time t is known to every actuator $a_1, \ldots, a_m$, the feedback control law is univocally determined by choosing $u(t) = K_{j,h}x(t)$ if $z(t) = Z_j$, $\tilde{\Lambda}(t) = \tilde{\Lambda}_h$, and this completes the proof. □

As convergence to the origin provided by Theorem 5 is intended in mean-square sense, we can no more refer to $\mathcal{E}_{Q_1} \triangleq \{x \in \mathbb{R}^n : x'Q_1^{-1}x \le 1\}$ and $\mathcal{E}_{Q_2} \triangleq \{x \in \mathbb{R}^n : x'Q_2^{-1}x \le 1\}$ as invariant ellipsoids for the closed-loop system, as we did in sections 2.2.2–2.2.3. In fact, the decreasing condition (65) only holds in expected value. Hence, even though mean-square stability is retained, we have that $x(t) \in \mathcal{E}_{Q_i} \not\Rightarrow x(t+1) \in \mathcal{E}_{Q_j}$, $\forall t \in \mathbb{N}_0$, $i, j = 1, 2$. In other words, constraints (68c)–(68d) do not imply fulfillment of (43) at every time step, but only in an averaged sense.

2.2.6 Simulation results

In this section the proposed decentralized control schemes are tested on an open-loop unstable system (42) with $n = 8$ states and $m = 4$ inputs. The matrices A, B in (42) are selected randomly[4] and hence with a high chance that state dynamics are strongly mutually coupled. Measurements are provided from sensors to actuators according to a topology Λ as in (52) with 8 ideal links and 4 lossy links, defined as

$$\Lambda = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & -1 & 0 & 0 \\ -1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & -1 \\ 0 & 0 & 0 & -1 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

The network topology is schematized in Fig. 7. Because of the network structure, the decentralized control law can only exploit a partial knowledge of the state value at each sample time. Since there are $L = 4$ unreliable links, the number of possible network configuration is $\ell = 2^L = 16$. Packet dropouts are modeled by a 2-states Markov chain defined by (61) with $q_1 = 0.8$, $q_2 = 0.5$, $d_1 = 0.1$ and $d_2 = 0.5$.

We run $N_{sim} = 50$ simulations of $T_{sim} = 50$ time steps each with constraints (43) defined by $x_{max} = 25$, $u_{max} = 3$, weight matrices $Q_x = I_n$, $Q_u = 10^{-2}I_m$, and a random initial state $x(0) \in \mathcal{X}_0$, with $\mathcal{X}_0 = \{x_c\} + \{x \in \mathbb{R}^8 :\ \|x\|_\infty \le 2\}$ and $x_c = 7 \cdot \mathbf{1}_{8\times 1}$. Fig. 8 shows the behavior of the entire state and input vectors under decentralized robust and stochastic control. Table 1 shows the results obtained by the proposed decentralized techniques in comparison with a centralized controller which implements the control law (45) without any restriction on the structure of K, or, in other words, which considers a topology $\Lambda = \mathbf{1}_{m\times n}$ where every actuator can exploit all the measurements. Performances are evaluated using the cumulated stage cost

$$J_i = \sum_{t=1}^{T_{sim}} (\|Q_x x(t)\|_2 + \|Q_u u(t)\|_2)$$

[4] The MATLAB routine `drss` has been used to obtain the matrices A, B, modified to enforce one eigenvalue of A to be equal to 1.05.

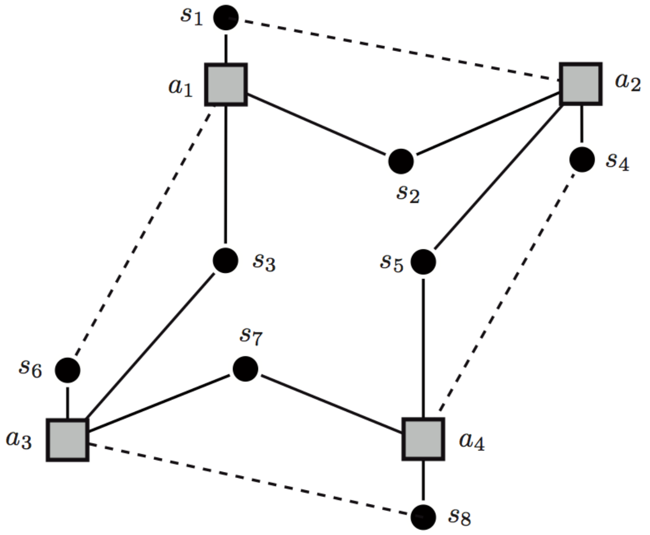


Figure 7: Network topology used in simulations

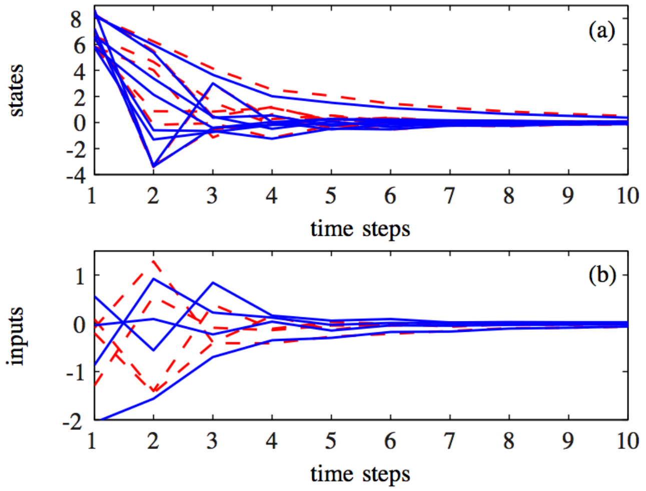


Figure 8: Total (a) state and (b) input trajectories for robust (dashed line) and stochastic (solid line) decentralized controllers

over the simulation horizon, where J_i refers to the ith run and $\mu(J_i)$, $\sigma(J_i)$ are the mean and the standard deviation of J_i over all the simulations. We can see that the stochastic decentralized controller achieves a good closed-loop behavior, being less conservative than the robust controller and still providing convergence to the origin. In Table 1 is also shown the computational time needed to solve the SDP problems off-line on a 2.8GHz Intel processor with the MATLAB modeling language YALMIP. Indeed, the complexity of the stochastic SDP problem (68), due mainly to the size of (68b), requires a CPU time of an order of magnitude larger. However, this increased computational load provides in turn a larger solution set, since the mean-square stability constraint (65) is less stringent than the robust counterpart (47).

2.2.7 Conclusions

We proposed a method based on semidefinite programming for synthesizing decentralized linear control laws for networked linear systems. Both the case of an ideal network, where sensors transmit measurements to actuators without data loss, and the case of packet dropout in a subset of the network communication links were considered. In the latter case, we also took into consideration a stochastic description of the network, where packet loss is modeled as a random process driven by a two-state Markov chain. The SDP problem formulation guarantees that the resulting switching controller enforces mean-square stability of the closed-loop system. Simulation results on a numerical example have shown that the performance deteriorates with respect to an ideal centralized controller,

Table 1: Simulation results

	$\mu(J_i)$	$\sigma(J_i)$	CPU
Ideal network			(off-line time)
Centralized control	41.0	0	2.8 s
Decentralized control	45.1	0	1.2 s
Lossy network			(off-line time)
Dec. robust control	50.0	1.57	8.1 s
Dec. stochastic control	47.1	2.38	59.2 s

on average, by 15% in the case of stochastic decentralized control, and by 22% in the case of robust decentralized control.

2.3 Energy-Aware MPC

D. Bernardini, A. Bemporad

2.3.1 Introduction

Wireless sensor networks (WSNs) is an emerging technology for collecting large amounts of measurement data that were previously cost prohibitive. By deploying a large number of cheap, small, and low-consumption sensors equipped with radios, wireless sensing in automation aims at reducing the costs of cabling and their maintenance due to wear and tear. Another advantage of WSNs is the possibility of rapidly reconfiguring the communications infrastructure in case of failures or additions of new components [131].

On the other hand, compared to standard wired sensors, WSNs pose new challenges for control design, such as energy consumption and channel reliability issues. While some interesting work has been done for the latter, such as modeling packet dropouts and addressing time delays (see e.g. [127–129]), energy-aware control is still a rather open problem. Energy budget is highly constrained to maximize the expected lifetime of battery-operated sensor nodes, therefore preventing frequent cumbersome and expensive maintenance for replacing batteries. Therefore the urge for developing new control techniques that, aware of communication and power consumption aspects of the wireless nodes, ensure an optimized controller-sensor operation.

In a wireless device the radio chip is the primary source of energy consumption; hence, radio usage must be minimized for achieving a satisfactory network lifetime [125, 126]. In recent years some work was done for addressing the energy problem from the point of view of both the communication network and the system architecture, i.e., by proposing consumption-efficient routing protocols [123, 124], or dynamic power management techniques [122, 130].

In this work we address the issue from the complementary angle of control design. We focus on control systems where feedback is provided by a WSN. Previous works related to this topic include [121], where a properly tuned control law is designed to achieve real-time regulation of the network transmission rate, and [120], where a predictive controller is proposed to optimize the trade-off between power transmission and system performance, as a function of the estimated wireless channel reliability. Moreover, communication between controller, sensors and actuators under nominal conditions is addressed in [119], where the authors present a new network transmission strategy, model the networked plant as a mixed logical dynamical (MLD) system [118], and formulate a nominal control problem based on mixed-integer programming (MIP). With respect to this work, we wish to avoid the need of on-line MIP to avoid excessive computation complexity, and to improve system performance by exploiting two-way channel communication.

In our framework we take into account a few key aspects to obtain effective energy savings. First, we consider the energy costs of both receiving and transmitting packets, which are usually very similar [116, 117]. Second, we avoid idle listening as much as possible, i.e., the radio chip of the sensor is completely turned off when no incoming packet is expected [126]. Finally, as common wireless protocols have a relatively large fixed cost due to communication overheads [117], although

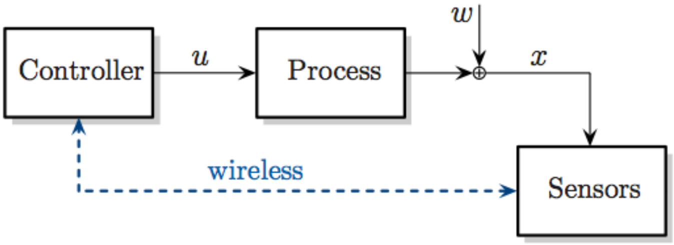


Figure 9: Control loop scheme with feedback from a single wireless node

a single measurement acquisition can be stored in few bytes, the transmission of small packets has a disproportionately high energy price compared to transmitting entire sets of measurements; for small n, transmitting n measurements costs almost as transmitting a single measurement.

By taking into account the above aspects, we address the energy problem of WSNs together with the optimal control problem of the global networked process. The main idea is to develop a network transmission strategy where measurements are transmitted to the controller *only when necessary*, and to design a control scheme aware of that transmission strategy, in order to allow a substantial reduction of radio usage without excessively sacrificing closed-loop performance. In our framework we assume the use of wireless sensors within "Class 2 - Closed-loop supervisory control" of the taxonomy of the ISA-SP100 standard[5], where information availability is often not safety-critical. Accordingly, we idealize the wireless communication channel, neglecting packet dropouts and delays, under the assumption that latency and jitter issues (due for instance to packet retransmission after a dropout) are not critical. Taking into account such practical aspects of real wireless networks is beyond the scope of this work; a discussion on the extension of the presented results to the case of real networks with delays and packet loss is given in Section 2.3.5.

We start in Section 2.3.3 by considering the case of noiseless state feedback, where a single wireless node, possibly embedding several on-board sensors, is in charge of closing the control loop. Then, in Section 2.3.4 we extend the approach to the case of uncertain state measurements, where feedback is provided by a wireless network formed by a remote controller and a short range WSN, where several sensors measure the same physical quantities to improve disturbance rejection. Finally, results of closed-loop simulations are reported in Section 2.3.5, and conclusions are drawn in Section 2.3.6.

A preliminary version of this work has appeared in [114, 115].

2.3.2 Notation

Hereafter, I_n is the $n \times n$ identity matrix and $\mathbf{0}_n$ is the null vector of $\mathbb{R}^n$. Given a vector $x \in \mathbb{R}^n$ and a matrix $Q \in \mathbb{R}^{n\times n}$, x^T denotes the transpose of x, $\lambda_i(Q)$, $i = 1, \dots, n$, are the eigenvalues of Q, and $\|Q^{\frac{1}{2}}x\|_2^2 \triangleq x^\mathsf{T}Qx$, $\|Qx\|_1 \triangleq \sum_{i=1}^n |(Qx)_i|$, $\|Qx\|_\infty \triangleq \max_{i\in\{1,\dots,n\}} |(Qx)_i|$, where $(Qx)_i$ denotes the i-th element of Qx. Given two sets $\mathcal{A}$ and $\mathcal{B}$, hull$\{\mathcal{A}, \mathcal{B}\}$ is the convex hull of $\mathcal{A} \cup \mathcal{B}$, $\mathcal{A} \oplus \mathcal{B} \triangleq \{a + b : a \in \mathcal{A},\ b \in \mathcal{B}\}$ is the Minkowski sum of $\mathcal{A}$ and $\mathcal{B}$, and $d(x, \mathcal{A}) \triangleq \inf_{y\in\mathcal{A}} ||x - y||$ is the distance of x form $\mathcal{A}$. The set $\mathbb{N}_0$ is the set of nonnegative integer numbers.

2.3.3 Feedback from a single wireless node

Consider a control loop where a single node, which possibly embeds several sensors, collects and transmit measurements to a controller through a wireless channel. The controller computes the input signals and delivers them to the actuation device through a wired channel (see Figure 9). In the following we focus on designing a control scheme that minimizes transmissions over the wireless feedback link, while providing acceptable closed-loop performance.

[5] ISA-SP100.14 Wireless Networks Optimized for Industrial Monitoring, 2006, `http://www.isa.org/filestore/ISASP100_14_CFP_14Jul06_Final(2).pdf`

Wireless transmission strategy

Consider a discrete-time linear system of the form

$$x(k+1) \quad = \quad Ax(k) + Bu(k) + w(k) \tag{69}$$

where $x(k) \in \mathbb{R}^{n_x}$ is the state, $u(k) \in \mathbb{R}^{n_u}$ is the input, $w(k) \in \mathcal{W}$ is an additive disturbance, and $k \in \mathbb{N}_0$ is the time index. $\mathcal{W} \subset \mathbb{R}^{n_x}$ is a given polytope containing the origin. State and input vectors are subject to the constraints

$$x \in \mathcal{X}, \quad u \in \mathcal{U} \tag{70}$$

where $\mathcal{X}$, $\mathcal{U}$ are polyhedra containing the origin in their interior. We assume that full state measurements are collected at every time step by n_x sensors embedded in a single wireless node. We define the following transmission strategy: At time step k, the wireless sensor node transmits the measurement $x(k) = [x_1(k),\ x_2(k),\ \ldots,\ x_{n_x}(k)]^\mathsf{T}$ to the controller if and only if

$$\exists i \in \{1, 2, \ldots, n_x\} \ : \ |x_i(k) - \hat{x}_i(k)| > \varepsilon_i \tag{71}$$

where $\varepsilon = [\varepsilon_1,\ \varepsilon_2,\ \ldots,\ \varepsilon_{n_x}]^\mathsf{T}$ is a vector of threshold values $\varepsilon_i \geq 0$ for every component of the state x. More compactly, condition (71) can be expressed by using a binary variable $\delta(k)$

$$[\delta(k) = 1] \leftrightarrow [x(k) - \hat{x}(k) \notin \mathcal{E}]$$

where $\mathcal{E} \triangleq \{x \in \mathbb{R}^{n_x} \ : \ |x_i| \leq \varepsilon_i,\ i = 1, 2, \ldots, n_x\}$ is the box defined by the threshold vector ε, and the vector $\hat{x}(k)$ is a prediction of the measured value $x(k)$ precalculated by the controller and transmitted beforehand to the wireless node. Predictions are updated in a two-way communication as follows: when $\delta(k) = 1$ the sensor transmits the measurement $x(k)$ to the controller, which computes a set of M updated predictions $\{\hat{x}(k+j)\}_{j=1}^{M} = \{\hat{x}(k+1),\ \hat{x}(k+2),\ \ldots,\ \hat{x}(k+M)\}$, and transmits them to the sensor. Moreover, if the controller does not receive any measurement for M time steps, i.e., $\delta(k) = \delta(k-1) = \ldots = \delta(k-M+1) = 0$, a one-way communication from controller to sensor node takes place to send M updated predictions $\{\hat{x}(k+j)\}_{j=1}^{M}$, computed using $\hat{x}(k)$ as an estimation of the current state $x(k)$. We refer to M as the *estimation horizon* and to $\{\hat{x}(k+j)\}_{j=1}^{M}$ as the *prediction buffer*.

Although the above network transmission strategy is introduced here for linear systems, it is very general and can be implemented in a wide set of scenarios, since the values of the predictions $\hat{x}$ can be calculated with any estimation technique depending on the application at hand. The threshold ε is an important tuning knob of the approach, as it trades off closed-loop performance versus traffic over the wireless channel. Note that the threshold logic (71) allows one to gather information on measured variables even when no measurement is received: if $\delta(k) = 0$, then $x(k) \in \mathcal{E} \oplus \{\hat{x}(k)\}$. In other words, from the controller's point of view a non-measurement is a set-valued measurement and, with an opportune choice of ε, this can be usefully exploited in set-membership estimation algorithms, as detailed in Section 2.3.4.

The value of the estimation horizon M must be chosen according to a trade-off between energy consumption and reliability of predictions. In fact, due to the presence of disturbances, the accuracy of the predicted state values decreases with the horizon length, i.e., if $\delta(k) = \delta(k+1) = \ldots = \delta(k+j) = 0$, the difference $|x_i(k+j) - \hat{x}_i(k+j|k)|$, in general, is likely to grow with $j \in \{0, 1, \ldots, M-1\}$. Hence, an overly large M will lead to unnecessary transmission of useless far-in-the-future predictions, that will be discarded by the sensor node because of (71).

A notable consequence of the proposed transmission strategy is that idle listening is almost completely avoided. In fact, assuming clock synchronization between the sensor node and the remote controller, the sensor node can wake up its radio chip only when a communications of next M predictions is expected to come.

Energy-aware controller design

Motivated by the search for a good trade-off between closed-loop performance and network transmission rate, we want to design a robust controller for (69) which guarantees asymptotical convergence of the state x to the origin despite disturbance and network-induced feedback error. Due to the presence of the threshold ε and of the persistent unknown disturbance w, clearly the state cannot be directly regulated to the origin. Therefore, following the idea of dual mode MPC [134], we set up an *outer control mode*, which drives the state x to a given set $\mathcal{X}_0$, and an *inner control mode* which robustly keeps the state in $\mathcal{X}_0$. The set $\mathcal{X}_0$ is assumed to be robust positively invariant with respect to additive disturbances, as from the following definition [132].

Definition 1. *The set $\mathcal{X}_0 \subseteq \mathbb{R}^n$ is* robust positively invariant (RPI) *for a system of the form $x(k+1) = f(x(k), w(k))$, with $x \in \mathbb{R}^n$ and $w \in \mathcal{W}$, if and only if $\forall x(0) \in \mathcal{X}_0$ and $\forall w(k) \in \mathcal{W}$ the solution $x(k) \in \mathcal{X}_0$, $\forall k \in \mathbb{N}_0$.*

Inner control mode

In the inner control mode we use a switching feedback control law, defined by the following lemma.

Lemma 4. *Let $K \in \mathbb{R}^{n_u \times n_x}$ and $A_c \triangleq A + BK$ be such that $|\lambda_i(A_c)| < 1$, $\forall i = 1, \ldots, n_x$, and let $\mathcal{X}_0 \subseteq \mathcal{X}$ be an RPI set for the system*

$$x(k+1) = A_c x(k) + f(k) \tag{72}$$

such that $K\mathcal{X}_0 \subseteq \mathcal{U}$, where $f(k)$ is an unknown but bounded disturbance such that

$$f(k) \ \in \ \mathcal{F} \ \triangleq \ \mathcal{W} \oplus \{-BK\mathcal{E}\} \tag{73}$$

Let x be the state of (69)-(70) *receiving feedback according to* (71) *and in closed loop with*

$$u(k) = \begin{cases} Kx(k) & \text{if } \delta(k) = 1 \\ K\hat{x}(k) & \text{otherwise} \end{cases} \tag{74}$$

where

$$\hat{x}(k+1) = \begin{cases} A_c x(k) & \text{if } \delta(k) = 1 \\ A_c \hat{x}(k) & \text{otherwise} \end{cases} \tag{75}$$

If $x(k) \in \mathcal{X}_0$, then $x(k+t) \in \mathcal{X}_0$, $\forall t \in \mathbb{N}_0$.

Proof. By combining (69), (74) and (75) we obtain the piecewise linear (PWL) system[6]

$$\begin{bmatrix} x(k+1) \\ \hat{x}(k+1) \end{bmatrix} = \begin{cases} \begin{bmatrix} A_c x(k) + w(k) \\ A_c x(k) \end{bmatrix} & \text{if } \delta(k) = 1 \\ \begin{bmatrix} Ax(k) + BK\hat{x}(k) + w(k) \\ A_c \hat{x}(k) \end{bmatrix} & \text{otherwise} \end{cases} \tag{76}$$

From (76) we have that

$$x(k+1) \ \in \ \{A_c x(k)\} \oplus \mathcal{W} \oplus \{-BK(x(k) - \hat{x}(k))\}$$

for all $\delta(k) \in \{0, 1\}$, which is implied by

$$x(k+1) \ \in \ \{A_c x(k)\} \oplus \mathcal{F}$$

because of (71), noting that if at the current time step we have $\delta(k) = 0$ then $x(k) - \hat{x}(k) \in \mathcal{E}$. Hence, system (76) is over-approximated by

$$\begin{bmatrix} x(k+1) \\ \hat{x}(k+1) \end{bmatrix} = \begin{cases} \begin{bmatrix} A_c x(k) + w(k) \\ A_c x(k) \end{bmatrix} & \text{if } \delta(k) = 1 \\ \begin{bmatrix} A_c x(k) + f(k) \\ A_c \hat{x}(k) \end{bmatrix} & \text{otherwise} \end{cases} \tag{77}$$

[6] The set of states $[x(k),\ \hat{x}(k)]^\mathsf{T}$ such that $\delta(k) = 1$ is not convex. This description of (76) is kept for ease of notation without loss of generality, as it is straightforward to build an equivalent PWL system with 2 modes and a partition of $2n_x + 1$ polyhedral sets.

with $f(k)$ as in (73). The dynamics of $\hat{x}$ can now be neglected, as they are no more needed in modeling the evolution of x in (77). Focusing on x only, since $\mathcal{W} \subseteq \mathcal{F}$ we have that (77) is conservatively approximated by (72), which is nominally asymptotically stable by hypothesis. Hence, as $\mathcal{X}_0$ is RPI for (72), it is RPI also for (76). Fulfillment of constraints (70) follows by $\mathcal{X}_0 \subseteq \mathcal{X}$ and $K\mathcal{X}_0 \subseteq \mathcal{U}$. □

We can use known results on RPI sets for linear systems to design $\mathcal{X}_0$, see e.g. [132–134]. A less conservative computation of $\mathcal{X}_0$ considers the RPI property with respect to (77) instead of (72), as shown in [114].

Outer control mode

To design the outer control mode we propose an algorithm based on explicit Model Predictive Control. MPC is widely spread in industry for control design of highly complex multivariable processes under constraints on input and state variables [135–138]. The idea behind MPC is to solve at each sampling time an open-loop finite-horizon optimal control problem based on a given prediction model of the process, by taking the current state of the process as the initial state. Only the first sample of the sequence of future optimal control moves is applied to the process. At the next time step, the remaining moves are discarded and a new optimal control problem based on new measurements is solved over a shifted prediction horizon. An alternative approach to evaluate the MPC law was proposed in [103]: rather then solving the optimization problem on line for the current state vector, by employing techniques of multiparametric programming the problem is solved off line for all state vectors within a given range, providing the *explicit* dependence of the control input on the state and reference, which is piecewise affine (PWA) and continuous. For a survey on explicit MPC the reader is referred to [139].

In our framework, an explicit formulation of MPC is a natural choice for many reasons: primarily, it can handle constraints and can be formulated to achieve robust control in presence of disturbances. Moreover, it allows the cheap computation of the prediction buffer $\{\hat{x}(k+j)\}_{j=1}^{M}$, by evaluating the future evolution of a simple closed-loop PWA system.

We present a scheme derived from min-max MPC [134, 140, 141], where the goal is to steer the state to a target set while minimizing a performance index over the worst-case disturbance realization. The basic idea of our approach is to include the knowledge of the transmission strategy (71) in the MPC optimization problem, so that the dynamics of the predicted state $\hat{x}$ can be coherently modeled into the optimizer.

Let $\{w^{\ell}_{k+j|k}\}_{j=0}^{N-1}$ denote the ℓ-th sequence of disturbance realizations over a prediction horizon of N steps given the state measurement at time k, $\ell \in \mathcal{L}$ (for simplicity of notation, since now on we will drop the subscript $_{j=0}$ and superscript $^{N-1}$). Further, let $\{u^{\ell}_{k+j|k}\}$ denote the sequence of control moves associated with the ℓ-th realization, and $\{x^{\ell}_{k+j|k}\}$ the corresponding state sequence. By ignoring the transmission strategy and assuming state feedback at every sampling instant, the min-max MPC problem at time k is expressed as in [134]

$$\min_{\{u^{\ell}_{k+j|k}\}} \max_{\ell \in \mathcal{L}} \sum_{j=0}^{N-1} L(x^{\ell}_{k+j|k}, u^{\ell}_{k+j|k}) \tag{78a}$$

$$\text{s.t. (69),} \tag{78b}$$

$$x^{\ell}_{k+j|k} \in \mathcal{X}, \tag{78c}$$

$$u^{\ell}_{k+j|k} \in \mathcal{U}, \tag{78d}$$

$$x^{\ell}_{k+N|k} \in \mathcal{X}_T, \tag{78e}$$

$$x^{\ell_1}_{k+j|k} = x^{\ell_2}_{k+j|k} \Rightarrow u^{\ell_1}_{k+j|k} = u^{\ell_2}_{k+j|k}, \tag{78f}$$

$$j = 0, \dots, N-1, \ \forall \ell, \ell_1, \ell_2 \in \mathcal{L},$$

where N is the control horizon, (78c)-(78d) are the state and input constraints, (78e) is the target set constraint, and (78f) is the *causality constraint*, which enforces a single control input for each state, reducing the freedom on the control sequence and making the control law independent of the path taken to reach that state.

Assumption 1. *The stage cost $L(x,u)$ is convex over $\mathcal{X} \times \mathcal{U}$ and such that*

$$L(x,Kx) \le L(y,u),\ \forall x \in \mathcal{X}_T,\ \forall y \notin \mathcal{X}_T,\ \forall u \in \mathcal{U} \tag{79a}$$

$$L(x,u) \ge \alpha(d(x,\mathcal{X}_T)),\ \forall x \notin \mathcal{X}_T,\ \forall u \in \mathcal{U} \tag{79b}$$

where α is a $\mathcal{K}$-function.

The following Lemma shows constructively how Assumption 1 can be satisfied.

Lemma 5. *Let $\mathcal{X}_0 = \{x \in \mathbb{R}^{n_x} : A_0 x \le b_0\}$, $A_0 \in \mathbb{R}^{n_r \times n_x}$, $b_0 \in \mathbb{R}^{n_r}$. For $i = 1,2,\ldots,n_r$, let*

$$c_{0,i} \ = \ \min_{x,s} s \tag{80a}$$

$$\text{s.t. } s \ge \|Q_x x\|_\infty, \tag{80b}$$

$$A_0^i x = b_0^i, \tag{80c}$$

and define

$$c_0 = \min_{i \in \{1,2,\ldots,n_r\}} c_{0,i} \tag{81}$$

with A_0^i the i-th row of A_0 and b_0^i the i-th element of b_0. Then the set

$$\mathcal{X}_T \triangleq \left\{ x \in \mathbb{R}^{n_x} : \begin{bmatrix} Q_x \\ -Q_x \end{bmatrix} x \le c_0 \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \right\} \tag{82}$$

is such that $\mathcal{X}_T \subseteq \mathcal{X}_0$, and the stage cost

$$L(x,u) = \|Q_x x\|_\infty \tag{83}$$

satisfies Assumption 1.

Proof. By (80), (81) and (82) it follows that $\mathcal{X}_T$ is the largest level set of $\|Q_x x\|_\infty$ such that $\mathcal{X}_T \subseteq \mathcal{X}_0$. By construction, $\forall y \notin \mathcal{X}_T$, $\|Q_x y\|_\infty > c_0$, since

$$\exists i \in \{1,2,\ldots,n_x\} : \ (Q_x^i y > c_0) \vee (Q_x^i y < -c_0)$$

where Q_x^i is the i-th row of Q_x, which proves (79a). Define $d(y,\mathcal{X}_T) = \inf_{x \in \mathcal{X}_T} \|Q_x(y-x)\|_\infty$. As $0 \in \mathcal{X}_T$, by definition of $\inf$, $d(y,\mathcal{X}_T) \le \|Q_x(y-0)\|_\infty$. By letting $\alpha(\phi) = \phi$, (79b) follows. Hence, Assumption 1 is satisfied. □

Note that the input u is not taken into account in (83). This is not likely to have a major impact on the closed-loop behavior of the process, because usually in the outer mode the state x is relatively far from the origin, and u saturates irrespective to the choice of the stage cost L. Note also that the stage cost (83) is only used in the outer mode: as soon as $x(k)$ enters $\mathcal{X}_0$, a constant feedback control loop of the form $u(k+j) = Kx(k+j)$ is applied for all $j \in \mathbb{N}_0$ (see next Theorem 6), where the gain K can be designed with arbitrary (yet stabilizing) performance criteria.

Under the hypothesis of exact state feedback, the optimal input sequence resulting from the solution of problem (78) ensures the asymptotical convergence of the state x of (69) to the target set $\mathcal{X}_T$ (see proof in [134]). By solving N mp-LPs as in [141], and by using $L(x,u) = \|Q_x x\|_\infty$, this solution is obtained in state-feedback piecewise affine form

$$u^*(x) = F_i x + g_i \qquad \text{if } x \in \mathcal{R}_i \tag{84}$$

where $F_i \in \mathbb{R}^{n_u \times n_x}$, $g_i \in \mathbb{R}^{n_u}$ and $\mathcal{R}_i = \{x \in \mathbb{R}^{n_x} : C_i x \le d_i\}$, $i \in \mathcal{I} \triangleq \{1,2,\ldots,r\}$.

Now we need to reintroduce the proposed transmission strategy into our control scheme. Rewriting (69) to include the transmission strategy (71), we obtain the PWL system

$$\begin{bmatrix} x(k+1) \\ \hat{x}(k+1) \end{bmatrix} = \begin{cases} \begin{bmatrix} Ax(k) + Bu(k) + w(k) \\ Ax(k) + Bu(k) \end{bmatrix} & \text{if } \delta(k) = 1 \\ \begin{bmatrix} Ax(k) + Bu(k) + w(k) \\ A\hat{x}(k) + Bu(k) \end{bmatrix} & \text{otherwise} \end{cases} \tag{85}$$

Suppose to apply the feedback control law derived from (84) to (85)

$$u(k) = \begin{cases} F_i x(k) + g_i & \text{if } \delta(k) = 1 \\ F_j \hat{x}(k) + g_j & \text{otherwise} \end{cases} \tag{86}$$

where $x(k) \in \mathcal{R}_i$ and $\hat{x}(k) \in \mathcal{R}_j$. We can see that if $\delta(k) = 1$, then $u(k) = u^*(x(k))$. Otherwise, if $\delta(k) = 0$, i.e. the exact value of $x(k)$ is not known, the difference between the optimal input obtained by (84) with full state knowledge and the input (86) actually applied to the system is

$$u(k) - u^*(x(k)) \;=\; F_j \hat{x}(k) + g_j - F_i x(k) - g_i$$

Hence the dynamics of (85) in closed loop with (86) can be recast as

$$\begin{bmatrix} x(k+1) \\ \hat{x}(k+1) \end{bmatrix} =$$

$$\begin{cases} \begin{bmatrix} (A + BF_i)x(k) + Bg_i + w(k) \\ (A + BF_i)x(k) + Bg_i \end{bmatrix} & \text{if } \delta(k) = 1 \\ \begin{bmatrix} (A + BF_i)x(k) + Bg_i + w(k) + q(k) \\ (A + BF_j)\hat{x}(k) + Bg_j \end{bmatrix} & \text{otherwise} \end{cases} \tag{87}$$

where

$$q(k) = B(F_j \hat{x}(k) + g_j - F_i x(k) - g_i) \tag{88}$$

is the error made with respect to the optimal trajectory achieved with (84) due to the lack of information on the exact value of $x(k)$.

The basic idea of our approach is to consider $q(k)$ as an additional unknown but bounded disturbance within a set $\mathcal{Q}$, with

$$\begin{aligned} \mathcal{Q} \;\triangleq\; & \{q \in \mathbb{R}^{n_x} : q = B(F_j \hat{x} + g_j - F_i x - g_i), \\ & x, \hat{x} \in \mathbb{R}^{n_x}, x - \hat{x} \in \mathcal{E}\} \end{aligned} \tag{89}$$

and to find a control law which is robust with respect to $q(k)$. We cannot directly set up a multiparametric optimization problem including q, since the polytope $\mathcal{Q}$ is dependent on $\{F_i\}_{i\in\mathcal{I}}$ and $\{g_i\}_{i\in\mathcal{I}}$ which are nonlinear functions of x. To overcome this issue, we propose to an iterative algorithm. At every step h of the algorithm, the updated set $\mathcal{Q}^h$, the gains $\{F_i\}_{i\in\mathcal{I}}^h$, $\{g_i\}_{i\in\mathcal{I}}^h$ and the partition $\{\mathcal{R}_i\}_{i\in\mathcal{I}}^h$ are computed as a function of the previous set $\mathcal{Q}^{h-1}$. The algorithm (to be executed off line) is defined in Algorithm 1, and is based on the linear system

$$x(k+1) = Ax(k) + Bu(k) + w(k) + q(k) \tag{90}$$

and the associated min-max MPC problem

$$\min_{\{u^\ell_{k+j|k}\}} \max_{\ell\in\mathcal{L}} \sum_{j=0}^{N-1} L(x^\ell_{k+j|k}, u^\ell_{k+j|k}) \tag{91a}$$

$$\text{s.t. (78c), (78d), (78e), (78f), (90),} \tag{91b}$$

together with its explicit solution in state feedback form.

Algorithm 1 Iterative explicit min-max MPC, single node case

```
set $h = 1$, $\mathcal{Q}^0 = \emptyset$, $\mathcal{Q}^1 = \{\mathbf{0}_{n_x}\}$;
while $\mathcal{Q}^h \not\subseteq \mathcal{Q}^{h-1}$ do
    solve (91) with $q \in \mathcal{Q}^h$;
    get the explicit solution data $\{F_i,\ g_i,\ \mathcal{R}_i\}^h$, $\forall i$;
    set $\mathcal{Q}^{h+1} = \text{hull}\{\mathcal{Q}_{ij}^{h+1},\ \forall i, j \in \mathcal{I}\}$, where
        $\mathcal{Q}_{ij}^{h+1} \triangleq \{q \in \mathbb{R}^{n_x} :\ x - \hat{x} \in \mathcal{E},\ x \in \mathcal{R}_i^h,\ \hat{x} \in \mathcal{R}_j^h,$
        $q = B(F_j^h \hat{x} + g_j^h - F_i^h x - g_i^h)\}$;
end while
set $\{F_i,\ g_i,\ \mathcal{R}_i\} = \{F_i,\ g_i,\ \mathcal{R}_i\}^{h-1}$, $\forall i \in \mathcal{I}$.
```

The transmission strategy (71) must be slightly modified in order to deal with the proposed dual mode MPC coherently. In addition to (71), the wireless sensor node is required to transmit the measurement also when the state $x(k)$ and its prediction $\hat{x}(k)$ lie in different control mode sets, i.e.,

$$[\delta(k)=0] \;\leftrightarrow\; [x(k)-\hat{x}(k)\in\mathcal{E}] \,\wedge \\ \big[[x(k)\in\mathcal{X}_0,\ \hat{x}(k)\in\mathcal{X}_0] \,\vee\, [x(k)\notin\mathcal{X}_0,\ \hat{x}(k)\notin\mathcal{X}_0]\big] \tag{92}$$

to avoid erroneous switches from outer to inner control mode.

We can finally define the Robust Energy-Aware MPC (REA-MPC) wireless feedback strategy in Algorithm 2 and state its properties in the following theorem.

Algorithm 2 Robust Energy-Aware MPC (REA-MPC)

```
Offline:
   run Algorithm 1 and get $\{F_i,\ g_i,\ \mathcal{R}_i\}$, $\forall i \in \mathcal{I}$;
   compute $K$, $\mathcal{X}_0$ and $\mathcal{X}_T$.
At $k = 0$:
   receive $x(0)$ from the wireless node;
   set $\hat{x}(0) = x(0)$;
   set $\hat{x}(j+1) = (A + BF_i)\hat{x}(j) + Bg_i$, $\hat{x}(j) \in \mathcal{R}_i$,
     $j = 0, \dots, M-1$;
   transmit $\{\hat{x}(j)\}_{j=1}^{M}$, $\mathcal{X}_0$ and $\varepsilon$ to the wireless node.
for all $k > 0$ do
   if $x(k)$ is received (because (92) is satisfied), then
      set $\delta(k) = 1$, otherwise $\delta(k) = 0$;
   end if
   if $\delta(k) = 1$ then
      if $x(k) \in \mathcal{X}_0$ then
         set $u(k) = Kx(k)$;
      else
         set $u(k) = B(F_i x(k) + g_i),\ x(k) \in \mathcal{R}_i$;
      end if
   else
      if $\hat{x}(k) \in \mathcal{X}_0$ then
         set $u(k) = K\hat{x}(k)$;
      else
         set $u(k) = B(F_i \hat{x}(k) + g_i),\ \hat{x}(k) \in \mathcal{R}_i$;
      end if
   end if
   if $\delta(k) = 1$ or $\hat{x}(k+1)$ has not yet been computed then
      if $x(k) \in \mathcal{X}_0$ then
         set $\hat{x}(k+j+1) = (A + BK)\hat{x}(k+j)$,
           $j = 0, \dots, M-1$;
      else
         set $\hat{x}(k+j+1) = (A + BF_i)\hat{x}(k+j) + Bg_i$,
           $\hat{x}(k+j) \in \mathcal{R}_i,\ j = 0, \dots, M-1$;
      end if
      transmit $\{\hat{x}(k+j)\}_{j=1}^{M}$ to the wireless node.
   end if
end for
```

Theorem 6. *Let Algorithm 1 admit a solution. Then the state x of* (69) *receiving feedback according to* (92) *and controlled by REA-MPC converges asymptotically to the target set $\mathcal{X}_T$ while satisfying constraints* (70)*. If $x(k) \in \mathcal{X}_0$ at any time step k, then $x(k+j) \in \mathcal{X}_0$, $\forall j \in \mathbb{N}_0$.*

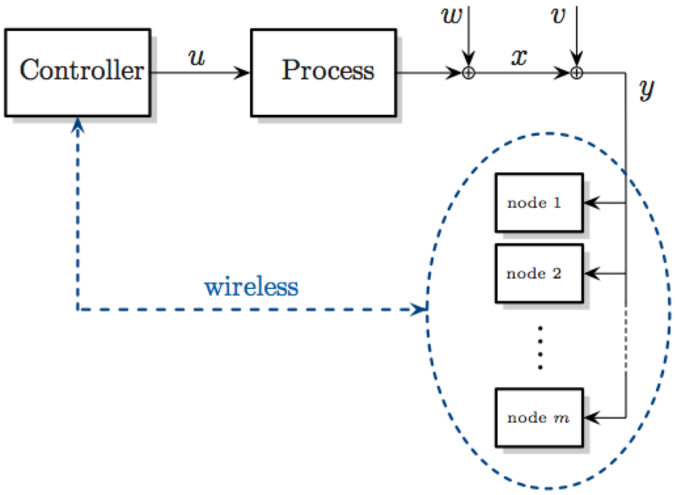


Figure 10: Control loop scheme with feedback from multiple wireless nodes

Proof. The outer mode explicit control law defined by Algorithms 1-2 is designed to be robust with respect to the additive disturbance w and the feedback error q, induced by the transmission strategy (92). Moreover, $x \in \mathcal{X}_T$ if and only if $\|Q_x x\|_\infty \leq c_0$. Then, $\forall x(k) \notin \mathcal{X}_T$, $\forall \ell \in \mathcal{L}$, $\|Q_x x^\ell_{k|k}\|_\infty - \|Q_x x^\ell_{k+N|k}\|_\infty \geq 0$. Hence, the outer mode stage cost can be shown to be non-increasing in time, and the proof of asymptotical convergence to $\mathcal{X}_T$ follows in a similar fashion of the proof of Theorem 1 in [134]. Once the controller switches to the inner control mode, robust invariance of closed-loop trajectories with respect to $\mathcal{X}_0$ follows by Lemma 4. □

2.3.4 Feedback from multiple wireless nodes

In this section we extend the results of Section 2.3.3 to the case of noisy state measurements. We consider a control loop where feedback is provided by a wireless network with several sensor nodes, where every node collects a noisy measurement of the state vector (see Figure 10). Redundancy is exploited here in order to mitigate output measurement errors. Sensor nodes communicate among them using low power (short-range) transmission, and determine whether to send (wide-area) the measurement to the remote controller (and which value to send) by means of an estimation algorithm.

Wireless transmission strategy

In the following we consider a system of the form (69) subject to state and input constraints (70), where we remove the hypothesis of exact state measurements. Instead, we assume that measurements are provided by a local WSN of m nodes, indexed by $i = 1, \ldots, m$, each one measuring the state vector $x(k)$. The current measurement given by the i-th node is defined as

$$y^i(k) = x(k) + v^i(k) \tag{93}$$

where $v^i \in \mathcal{V}^i$ is an unknown but bounded disturbance, and

$$\mathcal{V}^i = \{v \in \mathbb{R}^{n_x} : |v_j| \leq v^i_{j,max},\ j = 1, \ldots, n_x\} \tag{94}$$

The vectors $v^i_{max} = [v^i_{1,max},\ \ldots,\ v^i_{n_x,max}]$, $i = 1, \ldots, m$, are assumed to be known to every node. We define the transmission strategy as follows: At time step k let $h = 1 + \left(\lfloor \frac{k}{n} \rfloor \bmod m\right)$. We refer to the h-th node as the *master* node. The positive integer parameter n represents the number of consecutive time steps for which the master node does not change. Then, a short range communication transmitting the value $y^i(k)$ takes place from all the nodes $i \neq h$, called *slave* nodes, to the master node. Once all the measurements are delivered, the master node performs a simple set-membership estimation by computing the box set

$$\mathcal{Y}(k) = \bigcap_{i=1}^{m} \mathcal{Y}^i(k)$$
$$= \{x : b_{j,min} \le x_j \le b_{j,max},\ j = 1, \dots, n_x\} \tag{95}$$

where

$$b_{j,min} = \max_{i \in \{1,\dots,m\}} y_j^i(k) - v_{j,max}^i \tag{96a}$$
$$b_{j,max} = \min_{i \in \{1,\dots,m\}} y_j^i(k) + v_{j,max}^i \tag{96b}$$

and

$$\mathcal{Y}^i(k) = \{x : |y^i(k) - x| \le v_{max}^i\},\ i = 1, \dots, m \tag{97}$$

are all the feasible sets of states according to each node's measurements. Finally, the master node transmits $\mathcal{Y}(k)$ to the controller through a long range wireless communication if and only if

$$\exists j \in \{1, \dots, n_x\} :$$
$$[b_{j,min} - \hat{x}_j(k) < -\varepsilon_j] \ \vee\ [b_{j,max} - \hat{x}_j(k) > \varepsilon_j] \tag{98}$$

where $\hat{x}(k) \in \mathbb{R}^{n_x}$ is a prediction of the current state value $x(k)$, as in Section 2.3.3. Condition (98) is equivalent to

$$\mathcal{Y}(k) \not\subseteq \mathcal{E} \oplus \{\hat{x}(k)\} \tag{99}$$

where $\mathcal{E} = \{x : |x_j| \le \varepsilon_j,\ j = 1, \dots, n_x\}$. We represent the transmission condition (99) with $[\delta(k) = 1]$, where δ is a binary variable.

Note that the predictions $\hat{x}$ are exploited at every time step only by the current master node. Let $h_k = 1 + \left(\lfloor \frac{k}{n} \rfloor \bmod m\right)$ and $h_{k-1} = 1 + \left(\lfloor \frac{k-1}{n} \rfloor \bmod m\right)$. At time k, if $h_k \neq h_{k-1}$, then the h_{k-1}-th node (the previous master) is required to transmit, along with the measurement, also the prediction buffer to the h_k-th node (the current master). We adopt a time-varying master node in order to distribute long-range transmissions among all the sensor nodes, so to have uniform battery discharge. Other scheduling policies may be possible, for instance in case only one node is equipped with a communication device for the wide-area network.

Here we assume that the m nodes are spatially close to each other, so that the energy cost for a short range transmission is very small with respect to a wide-area transmission, and the communication activity between sensor nodes can be neglected. However, it is easy to extend the approach by using a threshold logic also for local transmissions, and having slave sensor nodes send the measurements to the master only when they are sufficiently far from a predicted value.

State estimation algorithm

Due to the output disturbance v, an exact state measurement is unavailable to the controller, either if the measurements have been transmitted or not. Hence we need to estimate the actual state value in order to feed the optimizer with an initial condition. It is easy to see that the transmission strategy (71) is such that the state $x(k)$ is always subject to a known set-membership relation, regardless of the master node's decision to forward the data. In particular, we have

$$x(k) \in \begin{cases} \mathcal{Y}(k) & \text{if } \delta(k) = 1 \\ \mathcal{E} \oplus \{\hat{x}(k)\} & \text{otherwise} \end{cases} \tag{100}$$

Hence, in the absence of packet dropouts, information on the measured variables is gathered even when no feedback is provided (i.e., when $\delta(k) = 0$). We propose to use an algorithm based on set-membership estimation [144], which takes into consideration this feature to reduce the uncertainty on the state.

Let $\mathcal{Z}(k|k)$ and $\mathcal{Z}(k+1|k)$ be the sets of all the possible values of the state x at time k and $k+1$, respectively, given the feedback information at time k. The set $\mathcal{Z}(k+1|k)$ is defined by the *prediction step*

$$\mathcal{Z}(k+1|k) \;=\; A\mathcal{Z}(k|k) \oplus B\{u(k)\} \oplus \mathcal{W} \tag{101}$$

and the set $\mathcal{Z}(k|k)$ is obtained by means of the *correction step*

$$\mathcal{Z}(k|k) \;=\; \mathcal{Z}(k|k-1) \cap \begin{cases} \mathcal{Y}(k) & \text{if } \delta(k)=1 \\ \mathcal{E} \oplus \{\hat{x}(k)\} & \text{otherwise} \end{cases} \tag{102}$$

The estimation $\bar{x}(k|k)$ of the actual state $x(k)$, given the feedback information at time k, is defined as the centroid of $\mathcal{Z}(k|k)$, computed as

$$\bar{x}(k|k) \;=\; c\big(\mathcal{Z}(k|k)\big) \;\triangleq\; \frac{1}{n_z}\sum_{i=1}^{n_z} z_i \tag{103}$$

where n_z is the number of the vertices $z_1,\ z_2,\ \ldots,\ z_{n_z}$ of $\mathcal{Z}(k|k)$.

This kind of set-membership algorithm can lead to very complex representations of $\mathcal{Z}$ (i.e., a high number of vertices). In order to lower the computational burden and preserve the implementability of the control scheme, in the simulations presented in Section 2.3.5 we use a sub-optimal estimation algorithm derived from [145], where outbounding parallelotopes are used to approximate the actual state set $\mathcal{Z}(k|k)$.

Energy-aware controller design

In the case of noisy state measurements we have to deal with three sources of uncertainty: the additive disturbance w, the feedback error due to the transmission strategy (98), and the measurement noise v^i, $i=1,\ldots,m$. In the rest of this section, we present an extension of the dual mode robust MPC proposed in Section 2.3.3 to the case of uncertain feedback from multiple sensor nodes.

Inner mode

Lemma 6. *Let $K \in \mathbb{R}^{n_u \times n_x}$ and $A_c \triangleq A + BK$ be such that $|\lambda_i(A_c)| < 1$, $\forall i = 1,\ldots,n_x$. Let $f(k)$ be an unknown but bounded disturbance such that*

$$f(k) \;\in\; \mathcal{F}_h \;=\; \mathcal{W} \oplus \{-BK\text{hull}\{\mathcal{E},\mathcal{V}\}\} \tag{104}$$

with $\mathcal{V} \triangleq \bigcap_{i=1}^m \mathcal{V}^i$. Let $\mathcal{X}_0 \subseteq \mathcal{X}$ be an RPI set for the system

$$x(k+1) = A_c x(k) + f(k) \tag{105}$$

such that $K\mathcal{X}_0 \subseteq \mathcal{U}$. Let x be the state of (69)–(70) *and* (93) *receiving feedback according to* (98) *and in closed-loop with*

$$u(k) \;=\; K\bar{x}(k|k) \tag{106}$$

If $x(k) \in \mathcal{X}_0$, then $x(k+t) \in \mathcal{X}_0$, $\forall t \in \mathbb{N}_0$.

Proof. Follows by similar arguments of Lemma 4. By combining (69), (93) and (106) we obtain

$$x(k+1) \;=\; Ax(k) + BK\bar{x}(k|k) + w(k) \tag{107}$$

Moreover, note that (93), (94), (98), (101), (102) and (103) imply

$$x(k) - \bar{x}(k|k) \;\in\; \mathcal{F}(k) \;\triangleq\; \begin{cases} \mathcal{V} & \text{if } \delta(k)=1 \\ \mathcal{E} & \text{otherwise} \end{cases} \tag{108}$$

and, conservatively, $x(k) - \bar{x}(k|k) \in \text{hull}\{\mathcal{V},\mathcal{E}\}$, $\forall\delta(k)$. Hence, system (107) is over-approximated by (105), which is nominally stable by hypothesis. Being $\mathcal{X}_0$ RPI for (105), it is RPI also for (107). Fulfillment of constraints (70) follows by $\mathcal{X}_0 \subseteq \mathcal{X}$ and $K\mathcal{X}_0 \subseteq \mathcal{U}$. □

Outer mode

Following the same lines of reasoning as in Section 2.3.3, consider again the explicit solution (84) of the exact state-feedback min-max problem (78). In order to take into account the transmission strategy (98) and the output noise (93), we consider the feedback control law

$$u(k) \;=\; F_j\bar{x}(k|k) + g_j \tag{109}$$

where $\bar{x}(k|k) \in \mathcal{R}_j$. Now, let $x(k) \in \mathcal{R}_i$. The difference between the optimal input (84) and the actual input (109) is

$$u(k) - u^*(x(k)) \;=\; F_j\bar{x}(k|k) + g_j - F_i x(k) - g_i$$

Hence, the evolution of the system (69)-(93) in closed-loop with (109) can be recast as

$$x(k+1) = (A + BF_i)x(k) + Bg_i + w(k) + q(k) \tag{110}$$

where

$$q(k) = B\big(F_j\bar{x}(k|k) + g_j - F_i x(k) - g_i\big) \tag{111}$$

models the input error due to the nonlinear transmission policy and the uncertain measurements. We consider $q(k) \in \mathcal{Q}$ as an additional unknown but bounded disturbance, where

$$\mathcal{Q} \;\triangleq\; \{q \in \mathbb{R}^{n_x} : q = B\big(F_j\bar{x} + g_j - F_i x - g_i,\; x, \bar{x} \in \mathbb{R}^{n_x},\; x - \bar{x} \in \mathcal{F}_h\} \tag{112}$$

and our goal is to obtain a robust control law with respect to both w and q. Similarly to previous Algorithm 1, we set up Algorithm 3 to compute this control law in explicit form, based on the linear system

$$x(k+1) \;=\; Ax(k) + Bu(k) + w(k) + q(k) \tag{113}$$

and the associated min-max MPC problem

$$\min_{\{u^\ell_{k+j|k}\}} \max_{\ell\in\mathcal{L}} \sum_{j=0}^{N-1} L(x^\ell_{k+j|k}, u^\ell_{k+j|k}) \tag{114a}$$

$$\text{s.t. (78c), (78d), (78e), (78f), (113).} \tag{114b}$$

Algorithm 3 Iterative explicit min-max MPC, multiple node case

```
set $h = 1$, $\mathcal{Q}^0 = \emptyset$, $\mathcal{Q}^1 = \{\mathbf{0}_{n_x}\}$;
while $\mathcal{Q}^h \not\subseteq \mathcal{Q}^{h-1}$ do
    solve (114) with $q \in \mathcal{Q}^h$;
    get the explicit solution data $\{F_i,\ g_i,\ \mathcal{R}_i\}^h$, $\forall i$;
    set $\mathcal{Q}^{h+1} = \text{hull}\{\mathcal{Q}^{h+1}_{ij},\ \forall i, j \in \mathcal{I}\}$, where
        $\mathcal{Q}^{h+1}_{ij} \triangleq \{q \in \mathbb{R}^{n_x} :\ x - \bar{x} \in \mathcal{F}_h,\ x \in \mathcal{R}^h_i,\ \bar{x} \in \mathcal{R}^h_j,$
        $q = B(F^h_j\bar{x} + g^h_j - F^h_i x - g^h_i)\}$;
end while
set $\{F_i,\ g_i,\ \mathcal{R}_i\} = \{F_i,\ g_i,\ \mathcal{R}_i\}^{h-1}$, $\forall i \in \mathcal{I}$.
```

The proposed algorithm for Robust Energy-Aware MPC with Noisy measurements (REAN-MPC) is described in Algorithm 4. Note that since the actual value of the state $x(k)$ is not known, the controller is supposed to switch from outer to inner mode at time k if and only if

$$\mathcal{Z}(k|k) \;\subseteq\; \mathcal{X}_0 \tag{115}$$

which ensures that $x(k) \in \mathcal{X}_0$. Moreover, we know that the terminal set $\mathcal{X}_0$ is RPI with respect to the closed-loop system; hence, we can reduce conservativeness in the estimation algorithm by using the prediction step

$$\mathcal{Z}(k+1|k) \;=\; (A\mathcal{Z}(k|k) \oplus B\,\{u(k)\} \oplus \mathcal{W}) \cap \mathcal{X}_0 \tag{116}$$

instead of (101), if at time k the inner control mode is active.

Algorithm 4 Robust Energy-Aware MPC with Noisy measurements (REAN-MPC)

```
Offline:
  run Algorithm 3 and get $\{F_i,\ g_i,\ \mathcal{R}_i\}$, $\forall i \in \mathcal{I}$;
  compute $K$, $\mathcal{X}_0$ and $\mathcal{X}_T$ as in Section 2.3.4-2.3.4.
At $k = 0$:
  receive $\mathcal{Y}(0)$ from the master node;
  set $\mathcal{Z}(0|0) = \mathcal{Y}(0)$;
  set $\bar{x}(0|0) = c(\mathcal{Z}(0|0))$;
  set $u(0) = B(F_i\bar{x}(0|0) + g_i),\ \bar{x}(0|0) \in \mathcal{R}_i$;
  set $\mathcal{Z}(1|0) = A\mathcal{Z}(0|0) \oplus B\,\{u(0)\} \oplus \mathcal{W}$;
  set $\hat{x}(0) = \bar{x}(0|0)$;
  set $\hat{x}(j+1) = (A + BF_i)\hat{x}(j) + Bg_i,\ \hat{x}(j) \in \mathcal{R}_i,$
    $j = 0, \ldots, M-1$;
  transmit $\{\hat{x}(j)\}_{j=1}^{M}$ and $\varepsilon$ to the master node.
for all $k > 0$ do
   if $\mathcal{Y}(k)$ is received (because (98) is satisfied) then
      set $\delta(k) = 1$, otherwise $\delta(k) = 0$;
   end if
   set $\mathcal{Z}(k|k) = \mathcal{Z}(k|k-1) \cap \begin{cases} \mathcal{Y}(k) & \text{if } \delta(k) = 1, \\ \mathcal{E} \oplus \{\hat{x}(k)\} & \text{otherwise;} \end{cases}$
   set $\bar{x}(k|k) = c\big(\mathcal{Z}(k|k)\big)$;
   if $\mathcal{Z}(k|k) \subseteq \mathcal{X}_0$ then
      set $u(k) = K\bar{x}(k|k)$;
      set $\mathcal{Z}(k+1|k) = (A\mathcal{Z}(k|k) \oplus B\,\{u(k)\} \oplus \mathcal{W}) \cap \mathcal{X}_0$;
   else
      set $u(k) = B(F_i\bar{x}(k|k) + g_i),\ \bar{x}(k|k) \in \mathcal{R}_i$;
      set $\mathcal{Z}(k+1|k) = A\mathcal{Z}(k|k) \oplus B\,\{u(k)\} \oplus \mathcal{W}$;
   end if
   if $\delta(k) = 1$ or $\hat{x}(k+1)$ has not yet been computed, then
      set $\hat{x}(k) = \bar{x}(k|k)$;
      if $\mathcal{Z}(k|k) \subseteq \mathcal{X}_0$, then
         set $\hat{x}(k+j+1) = (A + BK)\hat{x}(k+j),$
           $j = 0, \ldots, M-1$;
      else
         set $\hat{x}(k+j+1) = (A + BF_i)\hat{x}(k+j) + Bg_i,$
           $\hat{x}(k+j) \in \mathcal{R}_i,\ j = 0, \ldots, M-1$;
      end if
      transmit $\{\hat{x}(k+j)\}_{j=1}^{M}$ to the master node.
   end if
end for
```

Theorem 7. *Let Algorithm 3 admit a solution. Then the state x of* (69)-(93) *receiving feedback according to* (98) *and controlled by REAN-MPC converges asymptotically to the target set $\mathcal{X}_T$ while satisfying constraints* (70)*. If $\mathcal{Z}(k|k) \subseteq \mathcal{X}_0$ at any time step k, then $x(k+j) \in \mathcal{X}_0$, $\forall j \in \mathbb{N}_0$.*

Proof. Follows by similar arguments as in Theorem 6, noting that the outer mode explicit control law defined by Algorithms 3-4 is designed to be robust with respect to the additive disturbance w and the feedback error q, induced by the noisy measurements (93) and the transmission strategy (98). Once the controller switches to the inner control mode, because (115) is satisfied, robust invariance of closed-loop trajectories with respect to $\mathcal{X}_0$ follows by Lemma 6 □

2.3.5 Simulation results

To evaluate the performance of the proposed techniques we consider the open-loop unstable system (69)-(70), with

$$A = \begin{bmatrix} 0.21 & -0.39 \\ -0.39 & 0.82 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$\mathcal{X} = \{x \in \mathbb{R}^2 : |x_i| \leq 2,\ i = 1, 2\}$$
$$\mathcal{U} = \{u \in \mathbb{R} : |u| \leq 1\}$$
$$\mathcal{W} = \{w \in \mathbb{R}^2 : |w_i| \leq 0.06,\ i = 1, 2\}$$

and threshold vector $\varepsilon = [0.1,\ 0.1]^\mathsf{T}$. For the case of uncertain state measurements, we assume that the output noise is defined as (93)-(94) with $v^i_{max} = [0.06,\ 0.06]^\mathsf{T}$, $\forall i$, and that $m = 3$ sensor nodes provide feedback to the controller.

The presented energy-aware control schemes are compared to “traditional” robust MPCs, which do not take into account the minimization of transmissions over the wireless channel. In this case, no predictions are sent the sensor nodes, and the collected measurements are simply transmitted to the controller at every time step, regardless of any threshold logic. We refer to R-MPC as the robust MPC controller in the exact state feedback framework investigated in Section 2.3.3, and to RN-MPC as the robust MPC controller in the case of noisy measurements as described in Section 2.3.4. Note that R-MPC and RN-MPC algorithms are special cases of REA-MPC and REAN-MPC, respectively, where $\mathcal{E} = \emptyset$ and $M = 0$.

We run $N_s = 50$ simulations of $T = 10$ time steps each, randomly choosing the initial states $x(0) \in \mathcal{X}$. The weight matrices used to compute the outer and the inner controllers are $Q_x = I_2$, $Q_u = 0.1$. The control horizon is $N = 5$ and the estimation horizon is $M = 10$. The constant gain used in the inner control mode is $K = [0.3794,\ -0.7931]$. The terminal set $\mathcal{X}_0$, target set $\mathcal{X}_T$, and threshold set $\mathcal{E}$ used in simulations for REAN-MPC are shown in Figure 11 (corresponding sets for other controllers are analogous). By running Algorithms 1 and 3, we obtain an explicit control law for outer controllers in REA-MPC and REAN-MPC schemes, respectively, defined over 7 polyhedral partitions of the state space (see Figure 12).

As a benchmark index to compare the performance of the energy-aware schemes we consider two quantities: the rate of data transmission over the wireless channel, computed assuming equal power consumption in transmitting and receiving packets, as usual for short range wireless nodes [116], and the cumulated cost function

$$J^i_{exp} \triangleq \sum_{k=1}^{T} \left(\|Q_x x^i(k)\|_\infty + \|Q_u u^i(k)\|_\infty \right) \tag{117}$$

where i indexes the i-th simulation. The average results are reported in Table 2, where

$$J_{avg} \triangleq \frac{1}{N_s} \sum_{i=1}^{N_s} J^i_{exp}$$

A comparison of the data exchange over the wireless channel in the different control schemes is

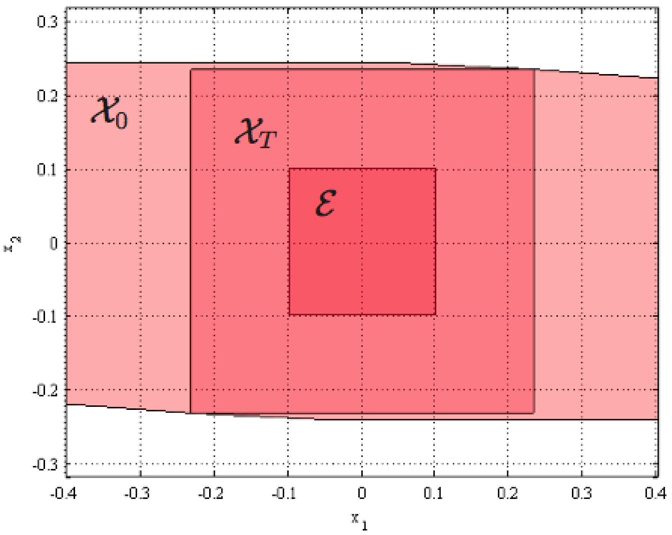


Figure 11: Terminal set $\mathcal{X}_0$, target set $\mathcal{X}_T$, and threshold set $\mathcal{E}$ used in simulations for REAN-MPC controller

Table 2: Energy-Aware MPC: simulation results

	J_{avg}	Tx. rate
Exact measurements		
R-MPC	2.606	100.0%
REA-MPC	2.687	47.2%
Noisy measurements		
RN-MPC	2.632	100.0%
REAN-MPC	2.724	54.0%

shown in Figure 13. Figures 14 and 15 show histograms of J_{exp} and radio utilization, respectively, for REA-MPC and REAN-MPC controllers. Data are given as percentage of analogue values obtained from R-MPC and RN-MPC.

REA and REAN model predictive controllers achieve a good trade-off between performance and transmission rate: REA-MPC grants a reduction in radio utilization by 52.8%, with a 3.1% loss in the experimental cost function, with respect to R-MPC; REAN-MPC obtains similar results, with a 46.0% saving in transmissions, and a loss of 3.5% in performance, compared to RN-MPC.

We can see from Figure 13 that with the proposed energy-aware approach a relevant part ($\simeq 50\%$) of the total wireless communications is spent to deliver updated predictions to the sensor nodes. In some frameworks, e.g. when the calculation of the control law does not require external information and nodes have sufficient computation capabilities, it could be possible to let the predictions be computed locally by the sensor nodes. In this way one would avoid the need of having transmissions from the controller, dramatically cutting the overall radio power consumption further.

Robust control schemes based on min-max optimization problems like those presented here can lead to conservative control action sometimes, since stability with respect to every possible disturbance realization sequence is required. In such cases nominal controllers can be adopted, which in general provide less conservative system performance at the expenses of lack of stability and constraint fulfillment properties. In particular, in this energy-aware framework it is straightforward to derive nominal counterparts of REA-MPC and REAN-MPC as explicit MPCs obtained by solving (91) and (114), respectively, with $\mathcal{W} = \mathcal{Q} = \{\mathbf{0}_{n_x}\}$ and $\mathcal{X}_T = \mathbb{R}^{n_x}$. Comparisons between robust and nominal energy-aware control schemes are reported in [114, 115].

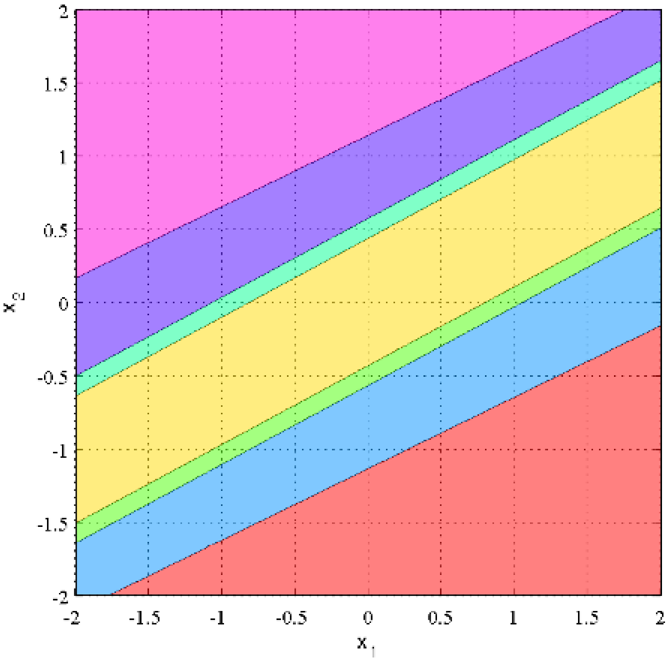


Figure 12: State space partition of explicit control law in REAN-MPC outer controller

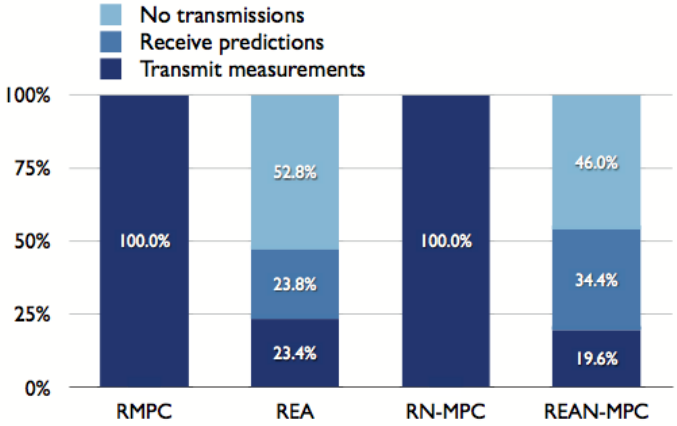


Figure 13: Comparison of traffic over the wireless channel

Discussion on network model

In the following we revise the assumptions made on the wireless network model (i.e., no delay nor packet loss can occur) and present considerations on how to extended the previous results if these assumptions are removed.

As mentioned in Section 2.3.1, the presence of delays in the network is not a crucial aspect to be explicitly considered in our framework, as they are often negligible if the sampling time is sufficiently large. Moreover, common delay compensation techniques can be implemented independently of the higher-level energy-aware control scheme, e.g., by augmenting the process model with the maximum estimated delay [142, 143].

On the contrary, the presence of packet loss in the controller-sensors wireless link needs to be properly addressed. Dropouts can occur in the forward channel (transmission of predictions from controller to sensors), or in the backward channel (transmission of measurements from sensors to controller). Data loss in the forward channel can be handled safely with minor modifications to the proposed control scheme. Namely, if at time k the sensor nodes do not receive an updated prediction buffer as expected, they simply transmit the new measurements to the controller at the next time step $k + 1$, regardless of the threshold logic. Hence, forward dropouts only affect the transmission rate over the wireless link, preserving convergence of the closed-loop system. Instead, packet dropouts in the backward channel may introduce stability issues if not properly managed. Let us consider the

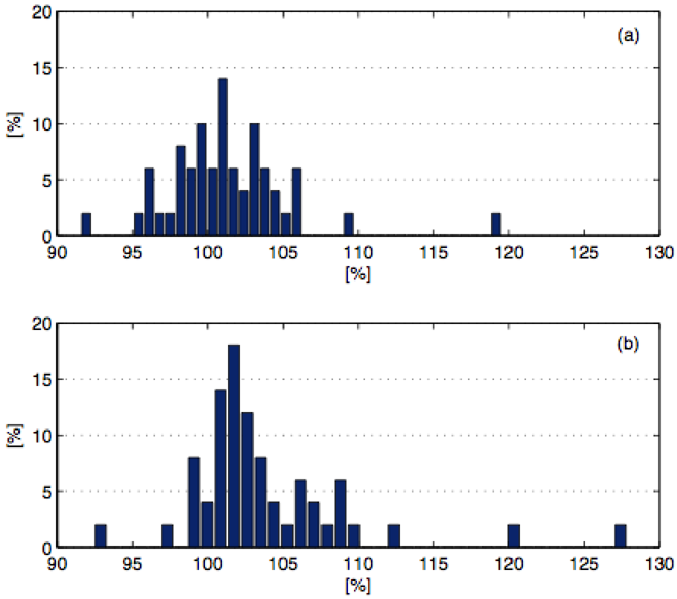


Figure 14: Histograms of J^{i}_{exp} for (a) REA-MPC and (b) REAN-MPC, normalized with respect to $J^{(R-MPC)i}_{exp}$ and $J^{(RN-MPC)i}_{exp}$, respectively.

case of feedback from multiple noisy sensors as described in Section 2.3.4 (exact state feedback can be handled with similar arguments). The absence of packet loss is assumed in the inner mode design (definition of $\mathcal{F}$ in (108)), in the outer mode design (definition of $\mathcal{Q}$ in (112)), and in the state estimation algorithm (correction step (102)). In particular, if transmitted measurements can be lost, then in general $x(k) - \bar{x}(k|k) \in \mathcal{E}$ does not hold, and it is necessary to obtain an analogous relation in order to preserve the control scheme functioning. Assuming a maximum number of consecutive packet dropouts p, this can be done by imposing a communication horizon c, defined as the maximum allowed time interval between a correctly delivered measurement transmission and the next sensors transmission: If the controller receives the last measurements packet at time k, then the sensors are required to transmit updated measurements at time $k + c$ regardless of the threshold logic. Then, $c + p$ is the maximum time interval between two consecutive measurements acknowledged by the controller. Based on this time bound, it is possible to compute a set $\tilde{\mathcal{E}}$ such that $x(k) - \bar{x}(k|k) \in \tilde{\mathcal{E}}$, by means of an iterative algorithm where the closed-loop system is considered to receive feedback every $c + p$ time steps.

2.3.6 Conclusions

We presented an energy-aware design approach for control systems based on feedback from battery-operated wireless sensors. We investigated both a single-node scenario where exact state measurements are available, and a multiple-node scenario where measurements are affected by noise and multiple redundant wireless nodes are used to mitigate the effects of disturbance. We proposed a novel WSN transmission strategy intended to save sensors battery by minimizing the communications over the wireless channel. This strategy is based on a threshold logic where the value of the threshold can be opportunely designed to tune the trade-off between closed-loop performance and transmission rate. Moreover, we presented two robust control schemes based on explicit model predictive control with guaranteed convergence and constraint fulfillment properties. Simulation results have shown that a substantial reduction in radio utilization ($\simeq 50\%$, which roughly corresponds to doubling the life of the wireless sensor nodes) can be achieved with a narrow loss in system performance ($\simeq 3\%$).

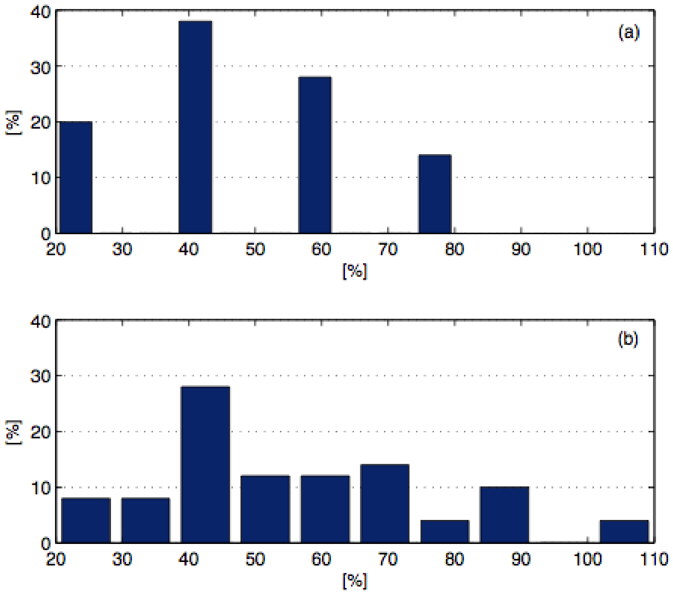


Figure 15: Histograms of radio utilization for (a) REA-MPC and (b) REAN-MPC.

2.4 DMPC of Dynamically-Coupled Linear Systems

D. Barcelli, A. Bemporad

2.4.1 Introduction

Ideas for decentralizing and hierarchically organizing the control actions in industrial automation systems date back to the 70's [111], but were mainly limited to the analysis of stability of decentralized linear control of interconnected subsystems, so the interest faded. Decentralized control and estimation schemes based on distributed convex optimization ideas based on Lagrangean relaxations have been proposed recently by various authors, see e.g. [108]. Here global solutions can be achieved after iterating a series of local computations and inter-agent communications.

Large-scale multi-variable control problems, such as those arising in the process industries, are often dealt with model predictive control (MPC) techniques. In MPC the control problem is formulated as an optimization one, where many different (and possibly conflicting) goals are easily formalized and state and control constraints can be included. Many results are nowadays available concerning stability and robustness of MPC, see e.g. [103] and references therein. However, centralized MPC is often unsuitable for control of large-scale networked systems, mainly due to lack of scalability and to maintenance issues of global models. The idea of decentralized MPC (DMPC) is to replace the original large size optimization problem by a number of smaller and easily tractable ones that work iteratively and cooperatively towards achieving a common, system-wide control objective. The goal of the decomposition is twofold: first, each subproblem is much smaller than the overall problem, and second, each subproblem is coupled to only a few other subproblems. Along with the benefits of a decentralized design, inherent issues in ensuring stability and feasibility of the system must be faced due to the mismatch of predictions that neighboring subsystems make about each other.

A few contributions have appeared in recent years in the context of DMPC. In [104] the system under control is composed by a number of unconstrained linear discrete-time subsystems with decoupled input signals and closed-loop stability is enforced through contractive constraints. In [112] the authors propose a cooperation-based distributed MPC algorithm based on a process of negotiations among DMPC agents, where the model considered for the subsystems only admits coupling through the control inputs. In [106] the authors consider the control of dynamically decoupled subsystems, whose state vectors are only coupled by a global performance objective. Closed-loop stability is ensured by constraining the state trajectory predicted by each agent close enough to the trajectory predicted at the previous time step that has been broadcasted. Dynamically decoupled submodels are also considered in [109]. Closed-loop stability is achieved by including sufficient stability conditions based on prediction errors in each DMPC subproblem.

A DMPC design approach was also proposed in [100] for *process* models that are not necessarily dynamically decoupled. The decoupling assumption only appears in the *prediction* models used by different MPC controllers. The degree of chosen decoupling represents a tuning knob of the approach. Sufficient criteria for the asymptotic stability of the process model in closed loop with the set of decentralized MPC controllers were given. A comparison of the above strategy with other decentralized strategy is reported in [105] on a problem of distributed power network control. Moreover, to cope with the case of a communication channel among neighboring MPC controllers which is faulty, a sufficient condition for ensuring closed-loop stability of the overall closed-loop system when a certain fixed number of packets containing state measurements may be lost was suggested in [101].

This work extends and generalizes the latter results and handles the case of tracking of constant output references, proposing a strategy to achieve offset-free tracking despite the decentralization. More general stability conditions are given for decentralized MPC in the presence of packet drops. The above are tested on a realistic simulation example of decentralized temperature control in a railcar.

2.4.2 Problem setup

We recall the MPC problem setup of [100].

Centralized model predictive control

Consider the problem of regulating the discrete-time linear time-invariant system

$$\left\{ \begin{array}{rcl} x(t+1) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) \end{array} \right. \tag{118}$$

to the origin while fulfilling the constraints

$$u_{\min} \leq u(t) \leq u_{\max} \tag{119}$$

at all time instants $t \in \mathbb{Z}_{0+}$, where $\mathbb{Z}_{0+}$ is the set of nonnegative integers, $x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^p$ are the state, input, and output vectors, and the pair (A, B) is stabilizable.

Assumption 2. *Matrix A in (118) has all its eigenvalues strictly inside the unit disc.*

Assumption 2 restricts the strategy and stability results to processes that are open-loop asymptotically stable, leaving to the controller the mere role of optimizing the performance of the closed-loop system.

Consider the following finite-horizon optimal control problem

$$\begin{aligned}
V(x(t)) = \min_{U} \quad & x'_{t+N} P x_{t+N} + \sum_{k=0}^{N-1} x'_k Q x_k + u'_k R u_k && \text{(120a)}\\
\text{s.t.} \quad & x_{k+1} = Ax_k + Bu_k,\ k = 0, \ldots, N-1 && \text{(120b)}\\
& y_k = Cx_k,\ k = 0, \ldots, N && \text{(120c)}\\
& x_0 = x(t) && \text{(120d)}\\
& u_{\min} \leq u_k \leq u_{\max},\ k = 0, \ldots, N_u - 1 && \text{(120e)}\\
& u_k = 0,\ k = N_u, \ldots, N-1 && \text{(120f)}
\end{aligned}$$

where, at each time t, $U \triangleq \{u_0, \ldots, u_{N_u-1}\}$ is the sequence of future input moves, x_k denotes the predicted state vector at time $t + k$, obtained by applying the input sequence $u_0, \ldots, u_{k-1}$ to model (118), starting from $x(t)$. In (120) $N > 0$ is the prediction horizon, $N_u \leq N$ is the input horizon, and "$\leq$" denotes component-wise inequalities. In (120) we assume that $Q = Q' \geq 0$, $R = R' > 0$, $P = P' \geq 0$ are square weight matrices defining the performance index, and P solves the Lyapunov equation $P = A'PA + Q$.

Problem (120) can be recast as a quadratic programming (QP) problem (see e.g. [103, 110]), whose solution $U^*(x(t)) \triangleq \{u^*_0 \ \ldots \ u^*_{N_u-1}\} \in \mathbb{R}^{m \times N_u}$ is a sequence of optimal control inputs. Only the first input $u(t) = u_0^*$ is actually applied to system (118), as the optimization problem (120) is repeated at time $t+1$, based on the new state $x(t+1)$.

Decentralized prediction models

In general the matrices A, B in (118) have a certain number of zero or negligible components corresponding to a partial dynamical decoupling of the process, or are even block diagonal in case of total dynamical decoupling.

Let M be the number of decentralized control actions that we want to design, for example $M = m$ in case each individual actuator is governed by its own controller. For all $i = 1, \ldots, M$, we define $x^i \in \mathbb{R}^{n_i}$ as the vector collecting a subset $\mathcal{I}_{xi} \subseteq \{1, \ldots, n\}$ of the state components,

$$x^i = W_i' x = [\, x_1^i \; \cdots \; x_{n_i}^i \,]' \in \mathbb{R}^{n_i}$$

where $W_i \in \mathbb{R}^{n \times n_i}$ collects the n_i columns of the identity matrix of order n corresponding to the indices in $\mathcal{I}_{xi}$, and, similarly, $u^i = Z_i' u = [\, u_1^i \; \cdots \; u_{m_i}^i \,]' \in \mathbb{R}^{m_i}$as the vector of input signals tackled by the i-th controller, where $Z_i \in \mathbb{R}^{m \times m_i}$ collects m_i columns of the identity matrix of order m corresponding to the set of indices $\mathcal{I}_{ui} \subseteq \{1, \ldots, m\}$. Note that

$$W_i' W_i = I_{n_i},\; Z_i' Z_i = I_{m_i},\; \forall i = 1, \ldots, M \tag{121}$$

where $I_{(\cdot)}$ denotes the identity matrix of order $(\cdot)$. By definition of x^i we obtain

$$x^i(t+1) = W_i' x(t+1) = W_i' A x(t) + W_i' B u(t) \tag{122}$$

An *approximation* of (118) is obtained by changing $W_i' A$ in (122) into $W_i' A W_i W_i'$ and $W_i' B$ into $W_i' B Z_i Z_i'$, therefore getting the new prediction model of reduced order

$$x^i(t+1) = A_i x^i(t) + B_i u^i(t) \tag{123}$$

where matrices $A_i = W_i' A W_i \in \mathbb{R}^{n_i \times n_i}$ and $B_i = W_i' B Z_i \in \mathbb{R}^{m_i \times m_i}$ are submatrices of the original A and B matrices, respectively, describing in a possibly approximate way the evolution of the states of subsystem #i.

The choice of the pair (W_i, Z_i) of *decoupling matrices* (and, consequently, of the dimensions n_i, m_i of each submodel) is a tuning knob of the DMPC procedure.

We want to design a controller for each set of moves u^i according to the prediction model (123) and based on feedback from x^i, for all $i = 1, \ldots, M$. Note that in general different states x^i, x^j and different u^i, u^j may share common components. To avoid ambiguities on the control action provided to the process, we impose that only a subset $\mathcal{I}_{ui}^{\#} \subseteq \mathcal{I}_{ui}$ of input signals computed by controller #i is actually applied to the process without ambiguity, and for the sake of simplicity of notation since now on we assume that $M = m$, i.e., that controller #i only controls the ith input signal.

Decentralized optimal control problems

Let the following assumption be satisfied:

Assumption 3. *Matrix A_i has all its eigenvalues strictly inside the unit disc, $\forall i = 1, \ldots, M$.*

Assumption 3 restricts the degrees of freedom in choosing the decentralized models (if $A_i = A$ for all $i = 1, \ldots, M$ is the only choice satisfying Assumption 3, then no decentralized MPC can be formulated within this framework). For all $i = 1, \ldots, M$ consider the following infinite-time constrained optimal control problems

$$\begin{aligned} V_i(x(t)) = \quad & \min_{\{u_k^i\}_{k=0}^{\infty}} \quad \sum_{k=0}^{\infty} (x_k^i)' W_i' Q W_i x_k^i + (u_k^i)' Z_i' R Z_i u_k^i = \\ & \min_{u_0^i} \quad (x_1^i)' P_i x_1^i + x^i(t)' W_i' Q W_i x^i(t) + (u_0^i)' Z_i' R Z_i u_0^i && (124) \\ & \text{s.t.} \quad x_1^i = A_i x^i(t) + B_i u_0^i \qquad x_0^i = W_i' x(t) = x^i(t) && (125) \\ & \qquad\;\; u_{\min}^i \le u_0^i \le u_{\max}^i \qquad u_k^i = 0,\; \forall k \ge 1 && (126) \end{aligned}$$

where $P_i = P_i' \geq 0$ is the solution of the Lyapunov equation

$$A_i'P_iA_i - P_i = -W_i'QW_i \tag{127}$$

that exists by virtue of Assumption 3. Problem (126) corresponds to a finite-horizon constrained problem with control horizon $N_u = 1$ and linear stable prediction models. Note that only the local state vector $x^i(t)$ is needed to solve Problem (126).

At time t, each controller MPC #i measures (or estimates) the state $x^i(t)$ (usually corresponding to local and neighboring states), solves problem (126) and obtains the optimizer $u_0^{*i} = [u_0^{*i1}, \ldots, u_0^{*ii}, \ldots, u_0^{*im_i}]' \in \mathbb{R}^{m_i}$. In the simplified case $M = m$ and $I_{ui}^{\#} = i$, only the i-th sample of u_0^{*i}, $u_i(t) = u_0^{*ii}$ will determine the i-th component $u_i(t)$ of the input vector actually implemented to the process at time t. The inputs u_0^{*ij}, $j \neq i$, $j \in \mathcal{I}_{ui}$ to the neighbors are discarded, their only role is to provide a better prediction of the state trajectories x_k^i, and therefore a possibly better performance of the overall closed-loop system.

The collection of the optimal inputs of all the M MPC controllers,

$$u(t) = [u_0^{*11} \ \ldots \ u_0^{*ii} \ \ldots \ u_0^{*mm}]' \tag{128}$$

is the actual input commanded to process (118). The optimizations (126) are repeated at time $t+1$, based on the new states $x^i(t+1) = W_i'x(t+1)$, according to the usual receding horizon control paradigm.

The degree of coupling between the DMPC controllers is dictated by the choice of the decoupling matrices (W_i, Z_i). Clearly, the larger the number of interactions captured by the submodels, the more complex the formulation (and, in general, the solution) of the optimization problems (126) and hence the computations performed by each control agent. Note also that a higher level of information exchange between control agents requires a higher communication overhead.

Convergence properties of decentralized MPC

The stability theorem proved in [100] provides closed-loop convergence results of the proposed DMPC scheme using the cost function $V(x(t)) \triangleq \sum_{i=1}^{M} V_i(W_i'x(t))$ as a Lyapunov function for the overall system. It is useful to recall here some quantities introduced in [100]

$$\begin{array}{ll} \Delta u^i(t) \triangleq u(t) - Z_iu_0^{*i}(t), & \Delta x^i(t) \triangleq (I - W_iW_i')x(t) \\ \Delta A^i \triangleq (I - W_iW_i')A, & \Delta B^i \triangleq B - W_iW_i'BZ_iZ_i' \end{array} \tag{129}$$

and

$$\Delta Y^i(x(t)) \triangleq \quad W_iW_i'(A\Delta x^i(t) + BZ_iZ_i'\Delta u^i(t)) + \Delta A^ix(t) + \Delta B^iu(t) \tag{130}$$

$$\Delta S^i(x(t)) \triangleq \quad \big(2(A_iW_i'x(t) + B_iu_0^{*i}(t))' + \Delta Y^i(x(t))'W_i\big)P_iW_i'\Delta Y^i(x(t)) \tag{131}$$

2.4.3 Decentralized MPC under arbitrary packet loss

In the previous section we assumed that the communication model among neighboring MPC controllers was faultless, so that each MPC agent could successfully receive the information about the states of its corresponding submodel. However, one of the main issues in networked control systems is the unreliability of communication channels, which may result in data packet dropout. In this section we derive a sufficient condition for ensuring convergence of the DMPC closed-loop in the case packets containing measurements are lost for an arbitrary but upper-bounded number N of consecutive time steps. The results shown here are based on formulation (126) and rely on the open-loop asymptotic stability Assumptions 2 and 3. The issue is still non-trivial, as if a set of measures for subsystem i is lost, this would affect not only the trajectory of subsystem i because of the improper control action u^i, but, due to the dynamical coupling, also the trajectories of subsystems $j \in J$, where $J = \{j \mid i \in \mathcal{I}_{xj} \cup \mathcal{I}_{uj}\}$, and thus the closed-loop stability of the overall system may be endangered.

By relying on open-loop stability, setting $u(t) = 0$ is a natural choice for backup input moves when no state measurements are available because of a communication blackout. The next theorem proves asymptotic closed-loop stability of decentralized MPC under packet loss. The proof of the theorem generalizes and unifies the results of [100, 101].

Theorem 8. *Let N be a positive integer such that no more than N consecutive steps of channel transmission blackout can occur. Assume $u(t) = 0$ is applied when no packet is received. Let Assumptions 2, 3 hold and $\forall i = 1, \dots, M$ define P_i as in* (127)*, $\Delta u^i(t)$, $\Delta x^i(t)$, ΔA^i, ΔB^i as in* (129)*, $\Delta Y^i(x(t))$ as in* (130)*,*

$$\Delta S^i_j(x) \quad \triangleq \quad [2(A_i W_i' x + B_i u_0^{*i}(x))' W_i' + \Delta Y^i(x)'](A^{j-1})' W_i P_i W_i' A^{j-1} \Delta Y^i(x) \tag{132}$$

and let $\xi_i(x) \triangleq A_i W_i' x + B_i u_0^{*i}(x)$. If the condition

$$\begin{aligned} &(i)\ \sum_{i=1}^{M} \big(x' W_i W_i' Q W_i W_i' x + \xi_i(x)'(P_i - W_i'(A^{j-1})' W_i P_i W_i' A^{j-1} W_i)\xi_i(x) - \Delta S^i_j(x)\big) \geq 0, \\ &\forall x \in \mathbb{R}^n, \qquad \forall j = 1, ..., N \end{aligned} \tag{133a}$$

is satisfied, or the condition

$$\begin{aligned} &(ii)\ \sum_{i=1}^{M} \big(x' W_i W_i' Q W_i W_i' x + \xi_i(x)'(P_i - W_i'(A^{j-1})' W_i P_i W_i' A^{j-1} W_i)\xi_i(x) - \Delta S^i_j(x) + u_0^{*i}(x)' Z_i' R Z_i u_0^{*i}(x)\big) \\ &- \alpha x' x \geq 0, \qquad \forall x \in \mathbb{R}^n,\ \forall j = 1, ..., N \end{aligned} \tag{133b}$$

is satisfied for some scalar $\alpha > 0$, then the decentralized MPC scheme defined in (126)–(128) in closed loop with (118) is globally asymptotically stable under packet loss.

Proof. Let $\{t_k\}_{k=0}^{\infty}$ be the sequence of sampling steps at which packet information is received, and let $j_k = t_{k+1} - t_k$ the corresponding number of consecutive packet drops, $1 \leq j_k \leq N$. We want to examine the difference $V_i(x(t_{k+1})) - V_i(x(t_k))$, where $V_i(x(t))$ is the optimal cost of subproblem (126) at time t. As the backup input $u(t_k + h) = 0$ is applied from time t_k to $t_{k+1} - 1$ ($h = 0, \dots, j_k - 1$), we have $x(t_{k+1}) = A^{j_k - 1}(Ax(t_k) + Bu(t_k)\ ) = A^{j_k - 1}(\Delta Y^i(x(t_k)) + W_i \xi(x(t_k)))$. Since $x^i(t_{k+1}) = W_i' x(t_{k+1})$, at time t_{k+1} the optimal cost $V_i(x(t_{k+1}))$ of subproblem (126) can be rewritten as $V_i(x(t_{k+1})) = (W_i' x(t_{k+1}))' W_i' Q W_i W_i' x(t_{k+1}) + (A_i W_i' x(t_{k+1}) + B_i u_0^{*i}(t_{k+1}))' P_i (A_i W_i' x(t_{k+1}) + B_i u_0^{*i}(t_{k+1})) + u_0^{*i}(t_{k+1})' Z_i' R Z_i u_0^{*i}(t_{k+1})$, where P_i is defined as in (127) and is such that $x_0^i P_i x_0^i = \sum_{k=0}^{\infty} (x_k^i)' W_i' Q W_i x_k^i$ with $x_{k+1}^i = A_i x_k$. Hence, considering that $u_0^i(t_{k+1}) = 0$ is a feasible suboptimal choice for problem (126), we obtain the following inequality

$$\begin{aligned} V_i(x(t_{k+1})) \quad \leq \quad & x'(t_{k+1}) W_i P_i W_i' x(t_{k+1}) \leq (A^{j_k - 1}(\Delta Y^i(x(t_k)) + \\ & W_i \xi(x(t_k))))' W_i P_i W_i' A^{j_k - 1}(\Delta Y^i(x(t_k)) + W_i \xi(x(t_k)))) = \\ & \Delta Y(x(t_k))'(A^{j_k - 1})' W_i P_i W_i' A^{j_k - 1} \Delta Y(x(t_k)) + 2\xi(x(t_k))' \cdot \\ & W_i'(A^{j_k - 1})' W_i P_i W_i' A^{j_k - 1} \Delta Y(x(t_k)) + \xi(x(t_k))' W_i'(A^{j_k - 1})' \\ & W_i P_i W_i' A^{j_k - 1} W_i \xi(x(t_k)) = \\ & \Delta S^i_{j_k}(x(t_k)) + \xi(x(t_k))' W_i'(A^{j_k - 1})' W_i P_i W_i' A^{j_k - 1} W_i \xi(x(t_k)) \end{aligned} \tag{134}$$

Since $V_i(x_i(t_k)) = (W_i' x(t_k))'(W_i' Q W_i) W_i' x(t_k) + \xi(x(t_k))' P_i \cdot\ \xi(x(t_k)) + u_0^{*i}(t_k)' Z_i' R Z_i u_0^{*i}(t_k)$ we get

$$\begin{aligned} V_i(x(t_{k+1})) - V_i(x_i(t_k)) \quad \leq \quad & \Delta S^i_{j_k}(x(t_k)) + \xi(x(t_k))' W_i'(A^{j_k - 1})' W_i P_i W_i' A^{j_k - 1} W_i \xi(x(t_k)) + \\ & -((W_i' x(t_k))'(W_i' Q W_i) W_i' x(t_k) + \xi(x(t_k))' P_i \xi(x(t_k)) u_0^{*i}(t_k)' Z_i' R Z_i u_0^{*i}(t_k)) \\ \leq\ & \Delta S^i_{j_k}(x(t_k)) - u_0^{*i}(t_k)' Z_i' R Z_i u_0^{*i}(t_k) - x(t_k)' W_i (W_i' Q W_i) W_i' x(t_k) + \\ & -\xi(x(t_k))'(P_i - W_i'(A^{j_k - 1})' W_i P_i W_i' A^{j_k - 1} W_i)\xi(x(t_k)) \end{aligned} \tag{135}$$

Let $V(x(t)) \triangleq \sum_{i=1}^{M} V_i(W_i' x(t))$. If (133a) holds, then it follows that $V(x(t_k))$ is a decreasing function of k lower-bounded by zero, and therefore converges as $k \to \infty$, which proves $\lim_{k \to \infty} V(x(t_{k+1})) - V(x(t_k)) = 0$. This in turn implies that $\lim_{k \to \infty} u_0^{i*}(t_k)' Z_i' R Z_i u_0^{i*}(t_k) = 0$. As $R > 0$ and Z_i are full-column-rank matrices, it follows that $Z_i' R Z_i > 0$ and hence that $\lim_{k\to\infty} u(t_k) = 0$. If (133b) holds, then in a similar way it is immediate to see that $\lim_{k\to\infty} x(t_k) = 0$ which again implies $\lim_{k\to\infty} u(t_k) = 0$, as around the origin $u(t_k)$ is a linear function of $x(t_k)$ (corresponding to the unconstrained solution of problem (126)). Since in the presence of packet drop $u(t) = 0$, the input sequence $\ldots, 0, 0, u(t_k), 0, \ldots, 0, u(t_{k+1}), 0, \ldots, 0, u(t_{k+2}), \ldots$ is actually applied to the process, clearly $\lim_{t\to\infty} u(t) = 0$. As asymptotically stable linear systems are also input-to-state stable [107], it immediately follows that $\lim_{t\to\infty} x(t) = 0$. □

Note again that around the origin the conditions in (133) become a quadratic form to be checked positive semidefinite, so local stability of (126)–(128) in closed loop with (118) under packet loss can be tested for any arbitrary fixed N.

Note also that conditions (133) are a generalization of the result of [100], as for $j = 1$ (no packet drop) matrix $P_i - W_i'(A^{j-1})' W_i P_i W_i' A^{j-1} W_i = P_i - P_i = 0$.

2.4.4 Extension to set-point tracking

Consider the following discrete-time linear global process model

$$\left\{ \begin{array}{rcl} z(t+1) & = & Az(t) + Bv(t) + F_d(t) \\ h(t) & = & Cz(t) \end{array} \right. \tag{136}$$

where $z \in \mathbb{R}^n$ is the state vector, $v \in \mathbb{R}^m$ is the input vector, $y \in \mathbb{R}^p$ is the output vector, $F_d \in \mathbb{R}^d$ is a vector of measured disturbances. Let A satisfy Assumption 2 and assume F_d is constant. We aim at solving a set-point tracking problem so that h tracks a given reference value $r \in \mathbb{R}^p$, despite the presence of F_d. In order to recast the problem as a regulation problem, assume steady-state vectors $z_r \in \mathbb{R}^n$ and $v_r \in \mathbb{R}^m$ exist solving the static problem

$$\left\{ \begin{array}{c} (I - A) z_r = B v_r + F_d \\ r = C z_r \end{array} \right. \tag{137}$$

and let $x = z - z_r$ and $u = v - v_r$. Input constraints $v_{\min} \le v \le v_{\max}$ are mapped into constraints $v_{\min} - v_r \le u \le v_{\max} - v_r$[7].

Proposition 1. *Under the global coordinate transformation* (137)*, the process* (136) *under the decentralized MPC law* (126)–(128) *is such that* $h(t)$ *converges asymptotically to the set-point* r*, under the assumption of Theorem in [100] or, in the presence of packet drops, of Theorem 8.*

Note that problem (137) is solved in a *centralized* way. Defining local coordinate transformations v_{ir}, z_{ir} based on submodels (123) would not lead, in general, to offset-free tracking, due to the mismatch between global and local models. This is a general observation one needs to take into account in decentralized tracking. Note also that both v_r and z_r depend on F_d as well as r, so problem (137) should be solved each time F_d or r change.

2.5 MPC of Stochastic Networked Control Systems

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

2.5.1 Introduction

In recent years, a vast literature has been produced on modeling, analysis and control design of Networked Control Systems (NCSs) (see, e.g., [146, 148, 154, 159] and references therein). Besides the many advantages offered by NCSs, such as increased system flexibility and low installation and

[7] In case $v_r \notin [v_{\min}, v_{\max}]$, perfect tracking under constraints is not possible, and an alternative is to set

$$\begin{array}{rcll} \left[\begin{smallmatrix} z_r \\ v_r \end{smallmatrix}\right] & = & \arg\min & \left\| \left[\begin{smallmatrix} I-A & -B \\ C & 0 \end{smallmatrix}\right] \left[\begin{smallmatrix} z_r \\ v_r \end{smallmatrix}\right] - \left[\begin{smallmatrix} F_d \\ r \end{smallmatrix}\right] \right\| \\ & & \text{s.t.} & v_{\min} \le v_r \le v_{\max} \end{array}$$

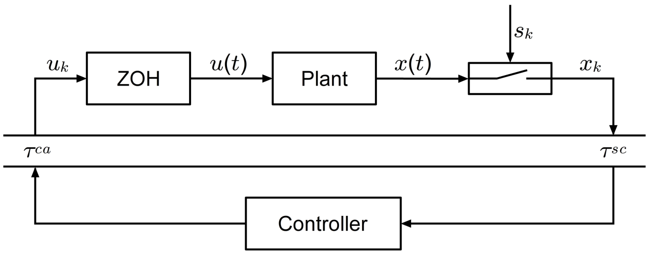


Figure 16: NCS overview scheme

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, [155, 158]). 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 work, 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 [147], based on the modeling of continuous-time impulsive systems. Alternatively, [152] 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 here the objective is controller synthesis based on Model Predictive Control (MPC) for NCS, we will exploit the approach of [152] 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., [151, 157], and references therein). In this work we adopt a formulation derived from [149] 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 [152] with the SMPC of [149] offers a general framework for MPC control of stochastic NCSs.

This work is organized as follows. The considered NCS model subject to stochastic uncertainty is introduced in Section 2.5.2. In Section 2.5.3, an overapproximation method intended to make the NCS model amenable for control synthesis, without removing the modeled stochastic information is proposed. Relying on the approximated model, a stabilizing stochastic MPC control scheme based on scenario enumeration is described in Section 2.5.4. Simulation results for a numerical example are shown in Section 2.5.5, and conclusions are drawn in Section 2.5.6.

2.5.2 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. 16. It consists of a linear continuous-time plant

$$\dot{x}(t) \;=\; Ax(t) + Bu(t) \tag{138}$$

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

$$s_k = \sum\nolimits_{i=0}^{k-1} h_i \quad \forall k \geq 1, \quad s_0 = 0, \tag{139}$$

which may not be equidistantly spaced in time due to the time-varying sampling intervals $h_k > 0$. The sequence $s_0, s_1, s_2, \dots$ 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. 16 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 (139)), 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., [159]). Considering this total delay τ_k, the continuous-time input signal $u(t)$ can be defined as

$$u(t) = u_k \ \text{ if } \ t \in [s_k + \tau_k, \ s_{k+1} + \tau_{k+1}), \ \forall k \in \mathbb{N}. \tag{140}$$

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 4. *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*

$$\Theta = \left\{ (h, \tau) \in \mathbb{R}^2 \ \mid \ h \in (0, h_{max}] \ \wedge \ \tau \in [0, h] \right\}. \tag{141}$$

By discretizing the linear plant (138) at the sampling times s_k, $k \in \mathbb{N}$, we obtain

$$x_{k+1} = e^{Ah_k} x_k + \int_0^{h_k - \tau_k} e^{As} ds B u_k + \int_{h_k - \tau_k}^{h_k} e^{As} ds B u_{k-1}.$$

Using now the lifted state vector $\xi_k = \begin{bmatrix} x_k^\mathsf{T} & u_{k-1}^\mathsf{T} \end{bmatrix}^\mathsf{T}$, 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

$$\xi_{k+1} = \underbrace{\begin{bmatrix} e^{Ah_k} & \int_{h_k-\tau_k}^{h_k} e^{As} ds B \\ 0 & 0 \end{bmatrix}}_{=: \tilde{A}_{h_k, \tau_k}} \xi_k + \underbrace{\begin{bmatrix} \int_0^{h_k-\tau_k} e^{As} ds B \\ I \end{bmatrix}}_{=: \tilde{B}_{h_k, \tau_k}} u_k. \tag{142}$$

The problem studied in this work is to design a control scheme for the NCS model given by system (142), in where the sampling intervals and transmission delays satisfy Assumption 4. 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 2. *System* (142) *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*

$$\mathbb{E}[\|\xi_k\|^2] \ \leq \ c \|\xi_0\|^2 \lambda^k, \ \ \forall k \in \mathbb{N}. \tag{143}$$

2.5.3 Overapproximation of NCS Model

Direct controller synthesis based on (142) 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 (142) 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 [153] for an overview and thorough comparison of all the existing overapproximation techniques). Here, we take a method derived in [150], 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 (142) 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

$$\xi_{k+1} \;=\; \tilde{A}_{h_k,\tau_k}\xi_k + \tilde{B}_{h_k,\tau_k}u_k, \tag{144}$$

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 (142) as

$$\begin{aligned} \tilde{A}_{h_k,\tau_k} &= F_0 + \sum\nolimits_{i=1}^{2\nu} \alpha_i(h_k,\tau_k)F_i, \\ \tilde{B}_{h_k,\tau_k} &= G_0 + \sum\nolimits_{i=1}^{2\nu} \alpha_i(h_k,\tau_k)G_i. \end{aligned} \tag{145}$$

In (145), 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 $h_k^{j-1}e^{ah_k}\cos(bh_k)$, $h_k^{j-1}e^{ah_k}\sin(bh_k)$, $(h_k-\tau_k)^{j-1}e^{a(h_k-\tau_k)}\cos(b(h_k-\tau_k))$ 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 (141), we obtain the following set of pairs of matrices

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

that contains all possible matrix combinations in (144). 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 [150], 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 [152], we partition the set Θ in polygons $\theta_m \subseteq \Theta$, $m \in \{1, 2, \ldots, S\}$, assign a probability $\tilde{p}_m = \iint_{\theta_m} p(h,\tau)dhd\tau$ to each polygon, and make for every θ_m a different overapproximation of the pair $\big(\tilde{A}_{h_k,\tau_k},\, \tilde{B}_{h_k,\tau_k}\big)$.

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

$$\cup_{m=1}^{S}\,\theta_m = \Theta,\ \text{int}\theta_i \neq \emptyset,\ \text{int}\theta_i \cap \text{int}\theta_j = \emptyset, \tag{146}$$

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

$$\mathcal{F}_m \;=\; \Big\{\big(\tilde{A}_{h_k,\tau_k},\, \tilde{B}_{h_k,\tau_k}\big) \mid (h_k,\tau_k) \in \theta_m\Big\}. \tag{147}$$

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

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

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 $\text{co}\{\mathcal{H}_m\}$, i.e.,

$$\mathcal{F}_m \subseteq \text{co}\{\mathcal{H}_m\},\ m = 1, 2, \ldots, S, \tag{148}$$

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

$$\mathcal{H}_m = \left\{ (H_{F,m,j},\, H_{G,m,j}) \ \mid\ j = 1, 2, \ldots, 2^{2\nu} \right\}. \tag{149}$$

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

$$\mathcal{V} = \{((V_{F,1,j},\, V_{G,1,j}),\ (V_{F,2,j},\, V_{G,2,j}),\ \ldots, \\ (V_{F,S,j},\, V_{G,S,j})) \ \mid\ j = 1, 2, \ldots, 2^{2\nu S} \}. \tag{150}$$

Hence, the procedure described above approximates the continuous PDF by a "discretized" version that takes the probability $\Pr[(h, \tau) \in \theta_m] = \iint_{\theta_m} p(h, t) dh d\tau$ for (h, τ) being contained in θ_m, and at the same time we overapproximate the system (142) on each of these polygons θ_m, $m = 1, 2, \ldots, S$, in the sense that $\mathcal{F}_m \subseteq \mathrm{co}\{\mathcal{H}_m\}$ for $m = 1, 2, \ldots, S$. The more refined the partitioning of Θ into $\theta_1, \theta_2, \ldots, \theta_m$ (at costs of more regions and then higher S, resulting in a higher computational complexity), the closer both the approximation of the continuous PDF $p(h, \tau)$ become and the tighter the overapproximation.

Remark 4. *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* (149) *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.*

2.5.4 Stochastic MPC design

The overapproximation described in Section 2.5.3 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 [149], 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. Hereafter, the different steps of the proposed control strategy design are given.

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 9. *Suppose there exist polygons* $\theta_1, \theta_2, \ldots, \theta_S$ *satisfying* (146)*, and an overapproximation of the NCS model* (142) *defined by the set of vertices* $\mathcal{V}$ *as in* (150) *such that* (148) *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*

$$\min_{Q,W,Y} \; \textit{trace}(W) \tag{151a}$$

$$\textit{s.t. } \textit{trace}(Q) = 1 \tag{151b}$$

$$W \succeq W_0 \tag{151c}$$

$$\begin{bmatrix} Q & Q & M_j^\mathsf{T} \\ Q & W & 0 \\ M_j & 0 & \tilde{Q} \end{bmatrix} \succeq 0, \; \forall j \in \{1, 2, \ldots, 2^{2\nu S}\}, \tag{151d}$$

where

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

$\tilde{p}_m = \iint_{\theta_m} p(h,\tau) dh d\tau$*,* $\tilde{Q} = \mathrm{Diag}\{\underbrace{Q, \ldots, Q}_{S \text{ times}}\}$*, and* $W_0 \succ 0$*. Then, the closed-loop NCS* (142) *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 (142) with $u_k = K\xi_k$. Using (144) and letting $C_{h,\tau} = \tilde{A}_{h,\tau} + \tilde{B}_{h,\tau}K$, we have that

$$\begin{aligned} \mathbb{E}\left[V(\xi_{k+1})\right] &= \mathbb{E}\left[\xi_k^\mathsf{T} C_{h_k,\tau_k}^\mathsf{T} P C_{h_k,\tau_k} \xi_k\right] \\ &= \iint_\Theta \xi_k^\mathsf{T} C_{h_k,\tau_k}^\mathsf{T} P C_{h_k,\tau_k}^\mathsf{T} \xi_k p(h_k,\tau_k) dh_k d\tau_k \\ &\leq \sum_{m=1}^{S} \tilde{p}_m \max_{(h_k,\tau_k)\in\theta_m} \xi_k^\mathsf{T} C_{h_k,\tau_k}^\mathsf{T} P C_{h_k,\tau_k} \xi_k. \end{aligned} \tag{152}$$

According to Lemma 1 in [156], 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 (152) is satisfied, holds when

$$P - L - \sum_{m=1}^{S} \tilde{p}_m C_{h_m,\tau_m}^\mathsf{T} P C_{h_m,\tau_m} \succeq 0 \tag{153}$$

for all $(h_m, \tau_m) \in \theta_m$, $m = 1, 2, \ldots, S$. Since (153) 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 (142) and the collections of pairs of matrices $\mathcal{H}_m$, $m = 1, 2, \ldots, S$, that satisfy (148). Hence, C_{h_m,τ_m} in (153) for $(h_m, \tau_m) \in \theta_m$ can be written as $C_{h_m,\tau_m} = \sum_{j=1}^{2^{2\nu}} \lambda_{m,j} \left(H_{F,m,j} + H_{G,m,j}K\right)$, 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 (153) is satisfied if

$$P - L - \sum_{m=1}^{S} \tilde{p}_m \tilde{C}_{m,j}^\mathsf{T} P \tilde{C}_{m,j} \; \succeq \; 0, \tag{154}$$

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 (154) is equivalent to (151d). Hence, the solution of (151) satisfies (153), and the closed-loop system is UGMSES. □

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 (142) to also optimize a certain performance criterion. We introduce a new set of s polygons $\phi_1, \phi_2, \dots, \phi_s$, which partition the set Θ such that properties analogous to (146) 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

$$\xi_{k+1} = \begin{cases} \bar{A}_1\xi_k + \bar{B}_1u_k & \text{if } (h_k, \tau_k) \in \phi_1, \\ \bar{A}_2\xi_k + \bar{B}_2u_k & \text{if } (h_k, \tau_k) \in \phi_2, \\ \quad\vdots & \qquad\vdots \\ \bar{A}_s\xi_k + \bar{B}_su_k & \text{if } (h_k, \tau_k) \in \phi_s, \end{cases} \tag{155}$$

where

$$\bar{A}_n = \iint_{\phi_n} \tilde{A}_{h,\tau} p(h,\tau) dh d\tau, \;\; \bar{B}_n = \iint_{\phi_n} \tilde{B}_{h,\tau} p(h,\tau) dh d\tau,$$

for all $n \in \{1, 2, \dots, s\}$, with $\tilde{A}_{h,\tau}$ and $\tilde{B}_{h,\tau}$ as in (142). Since we assumed that (h, τ) is given by an IID random process, the realization probabilities of every dynamical mode of (155) are taken to be $\bar{p}_n = \iint_{\phi_n} p(h,\tau) dh d\tau$, for all $n \in \{1, 2, \dots, s\}$ and $k \in \mathbb{N}$. As model (155) 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 $\phi_1, \phi_2, \dots, \phi_s$ 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 $\theta_1, \theta_2, \dots, \theta_S$. Further details on the partitions tuning are given in Section 2.5.5.

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 $\xi_k = \begin{bmatrix} x_k^\mathsf{T} & u_{k-1}^\mathsf{T} \end{bmatrix}^\mathsf{T}$, a list of candidate nodes is generated by considering all the s possible dynamics in (155) and their probabilities $\bar{p}_n$, $n = 1, 2, \dots, s$. 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 $\bar{p}_n$, $n = 1, 2, \dots, s$, it can be computed off-line, thus keeping the computational burden low. More details on the tree design procedure can be found in [149].

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

$$\mathbb{E}\left[\sum_{j=0}^{N-1} \left(\xi_{k+j}^\mathsf{T} Q_\xi \xi_{k+j} + u_{k+j}^\mathsf{T} Q_u u_{k+j} \right) + \xi_{k+N}^\mathsf{T} Q_\xi \xi_{k+N} \right] \tag{156}$$

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:

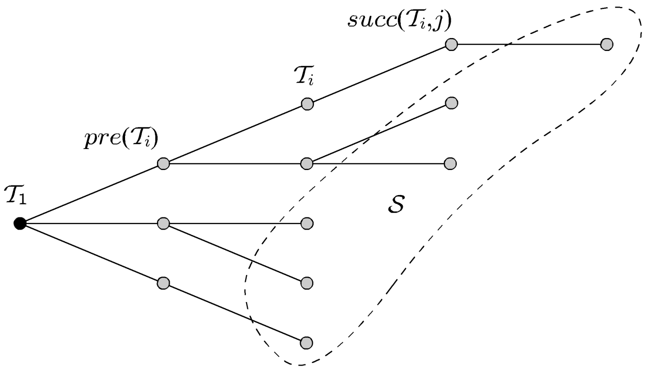


Figure 17: Optimization tree diagram

- $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_n\}$: 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 (155), $j \in \{1, 2, \dots, s\}$.
- $\delta(\mathcal{N}) \in \{1, 2, \dots, s\}$: 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 $\mathcal{S} \triangleq \{\mathcal{T}_i \in \mathcal{T} \mid succ(\mathcal{T}_i, j) \notin \mathcal{T},\ j = 1, 2, \dots, s\}$.

In Fig. 17 an illustrative optimization tree is shown to exemplify the notation. 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

$$\min_{\{u_i\}} \sum_{i \in \mathcal{T} \setminus \mathcal{T}_1} \pi_i \xi_i^{\mathsf{T}} Q_\xi \xi_i + \sum_{j \in \mathcal{T} \setminus \mathcal{S}} \pi_j u_j^{\mathsf{T}} Q_u u_j \tag{157a}$$

$$\text{s.t. } \xi_1 = \xi_k \tag{157b}$$

$$\xi_i = \bar{A}_{\delta(i)} \xi_{pre(i)} + \bar{B}_{\delta(i)} u_{pre(i)}, \ \forall i \in \mathcal{T} \setminus \{\mathcal{T}_1\} \tag{157c}$$

$$\sum_{m=1}^{S} \tilde{p}_m G_{m,j}^{\mathsf{T}} P G_{m,j} \leq \xi_1^{\mathsf{T}} (P - L) \xi_1, \quad \forall j \in \{1, 2, \dots, 2^{2\nu S}\}, \tag{157d}$$

where $G_{m,j} = V_{F,m,j}\xi_1 + V_{G,m,j}u_1$. Note that (157a) tends to (156) with $N \to \infty$ if $n_{max} \to \infty$. Hence, the expected value of the closed-loop performance (156) can be approximated with arbitrary accuracy at the expense of a higher computational load. Problem (157) is a quadratically constrained quadratic problem (QCQP). Provided that (151) has solution, this problem is always feasible, as shown below.

Remark 5. *In the borderline case where $\mathcal{T}$ is a complete tree, i.e., an s-ary tree in which all the leaf nodes are at some depth N and all nodes but the leaf nodes have exactly s successors, we have that minimizing the performance index* (157a) *at time step k is equivalent to minimizing* (156)*. Otherwise, if the tree is not complete,* (157a) *is an approximation of* (156)*, where the nodes related to the cut branches can be seen as terms in the cost function with null weight.*

Theorem 10. *Suppose there exist polygons $\theta_1, \theta_2, \dots, \theta_S$ and $\phi_1, \phi_2, \dots, \phi_s$ satisfying* (146)*, and an overapproximation of the NCS model* (142) *defined by the set of vertices $\mathcal{V}$ as in* (150) *such that* (148) *holds. Assume that the matrices Q, W are given from the solution of* (151)*. Then, the closed-loop NCS* (142) *where $u_k = u_{\mathcal{T}_1}$ and $u_{\mathcal{T}_1}$ is given by the receding horizon solution of* (157)*, with $P = Q^{-1}$ and $L = W^{-1}$, is UGMSES.*

Proof. By similar reasonings as in Theorem 9, we have that mean-square stability is provided by the receding-horizon satisfaction of condition (157d), 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 (157) is recursively feasible at every time step. This follows by noting that (151d) implies (157d) 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 (157), where $K = YQ^{-1}$ is obtained by solving (151). □

2.5.5 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 (138), 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 (141) as

$$\Theta = \left\{(h, \tau) \in \mathbb{R}^2 \;\mid\; h = h_{\text{nom}} \;\wedge\; \tau \in [0.02,\, 0.1]\right\}. \tag{158}$$

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 $\sigma = 0.012$.A plot of the considered PDF is shown in Fig. 18.

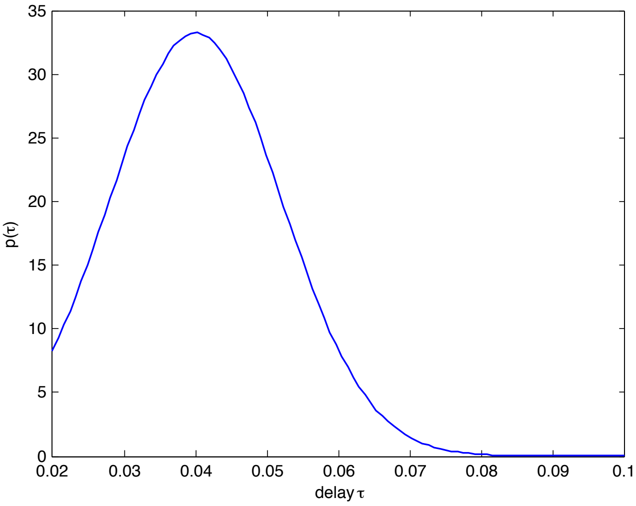


Figure 18: Probability density function $p(\tau)$ for $h = 0.1$

In order to satisfy the conditions of Theorem 9, we first compute an overapproximation of the NCS as described in Section 2.5.3. Then, according to Remark 4, 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 (151), and to obtain a stabilizing controller of the form $u_k = K\xi_k$ for the closed-loop system (142). 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 (157) 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 (151) with $S = 1$ and $\theta_1 = \Theta$. Since a feasible solution could not be found with $(h_k, \tau_k) \in \Theta$ as in (158), 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

Table 3: Simulation results

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

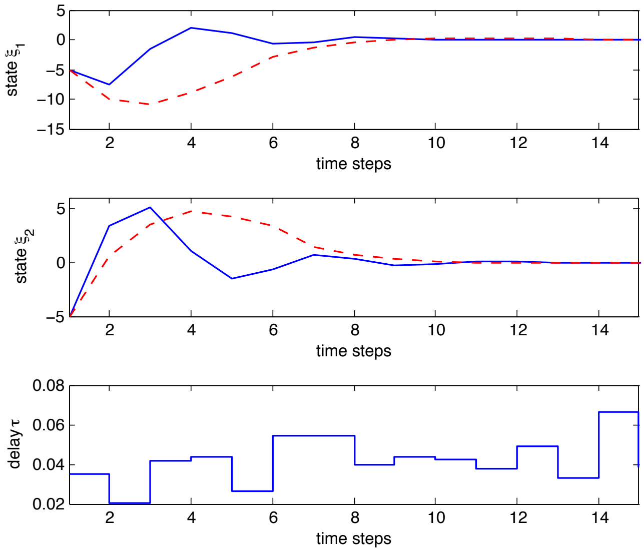


Figure 19: Example of trajectories obtained with SMPC (solid line) and robust state-feedback (dashed line)

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 3 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 is shown in Fig. 19. 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 (151) 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.

2.5.6 Conclusions

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.

3 Network-Aware Estimation

3.1 Optimal Kalman filter for systems with communication delays

L. Baramov, D. Pachner and V. Havlena

3.1.1 Motivation

The contributions of HPL to WP4 are novel approaches to state estimation under communication delay. State estimates can be used for control, optimization, process monitoring, fault detection, e.t.c., whenever states of the process are not accessible for direct measurement. The estimator uses the process output observations as well as process model to infer the internal process states. Measured process data are subject to noise corruption and process model is subject to uncertainty affecting the quality of the state estimate. Techniques developed under our effort in T4.2 will support other WIDE tasks, namely distributed MPC control, control over networks, real-time optimization. A particular attention was paid to reducing computational complexity by using gains pre-computed off-line and exploiting the particular problem structure. We have developed a reduced complexity representation for the optimal Kalman filter; further, a suboptimal approach was obtained which is of a particularly simple implementation that can be used in fast control loops in the basic control layer for estimating process controlled values purged from measurement noise and communication delays. Thus, it contributes to integrating WSN into Distributed Control Systems (DCSs) without compromising stability.

3.1.2 State of the art

Standard approaches to control and estimation consider an ideal point-to-point connection between the process and the controller and the estimator. A significant development in the estimation over networks has occurred since mid-1990's upon the emergence of networked control systems. A classical approach to state estimation is Kalman filtering that produces optimal estimates for linear process models and Gaussian noise; a classical reference is [5]. State estimation, and particularly Kalman filtering, which can handle correctly network delays, can separate control design, using the certainty equivalence principle, from handling communication delays from sensors. Kalman filter that takes into account variable communication delays can be used not only with control algorithms using process states explicitly, but also with output feedback algorithms as PID, for which Kalman filter supplies estimates of the undelayed process output.

A great deal of literature deals with lossy measurements – cases when data arrive either on time or never – see e.g., Wu and Chen (2007), Smith and Selier (2003), and Liu and Goldsmith (2004). There is also a number of papers dealing with variable delay considering delayed data arriving in arbitrary order (e.g., Schenato (2007), Pachner and Havlena (2008), and Challa et al (2002)). The optimal way to handle these problems in the linear—Gaussian framework is using Kalman filter for the augmented process model with time-varying parameters. The drawback of this optimal estimator is its need for an extensive computational effort. Hence, the recent research has concentrated on developing approximations of Kalman filter that are easy to implement and have low demands on on-line computations: for instance, observers with scheduled and/or switched injection gains were proposed, with proven stability (see, e.g., Schenato (2008), Smith and Selier (2003), and Liu and Goldsmith (2004)). Some of these approaches need a stochastic model for delay/loss distribution. An suboptimal approach to an approximate Kalman filter is in Larsen *et al* (1998). Pachner and Havlena (2008) proved that the optimal Kalman filter can be implemented as a switched observer with a finite number of injection gains (pre-computed off-line) on condition that all data are delivered within a maximum delay.

We have considered Kalman filtering over communication networks, where measured process data are subject to random communication delays and/or packet losses. We have assumed the discrete time settings; further, communication delays are assumed to be integer multiples of the sampling interval. We have further assumed that the communication protocol assigns a time stamp to each process value transmitted and hence, the time delay is known to the estimator. Naturally, the data may arrive out of order, and possibly in bursts when several measurements with different time stamps arrive at a time. The time stamp availability requires clock signals to be available on both transmitter and receiver sides. This assumption is fairly realistic for control systems with sampling periods from seconds to minutes, such as power, petrochemical or chemical processes. There, the clock inaccuracy is negligible compared to the sampling period.

3.1.3 Results overview

As was mentioned above, an optimal solution to this problem in the widely used linear-Gaussian setting is the time-varying Kalman filter where process model is augmented by chains of delay blocks modeling the network as shown in Figure 20. The length of the delay chain is limited to a pre-specified value, specifying the maximum communication delay considered. Measurements not arriving within this maximum delay are considered lost. Depending on the actual delay, particular states of the delay chain are designated as ‘measurements’. A classical time-varying Kalman filter implementation for this augmented plant may be computationally extensive; we have proposed a new implementation of the optimal Kalman filter of a lower complexity than the typical square-root algorithm (see [70]). Reducing complexity was achieved by pre-computing certain gains and a particular factorized representation of the state covariance matrix coming from a deep investigation of the problem structure. Moreover, it was also derived that states of the delay chain corresponding to data that has already been received are redundant and removed from the process, thus further reducing the computational load.

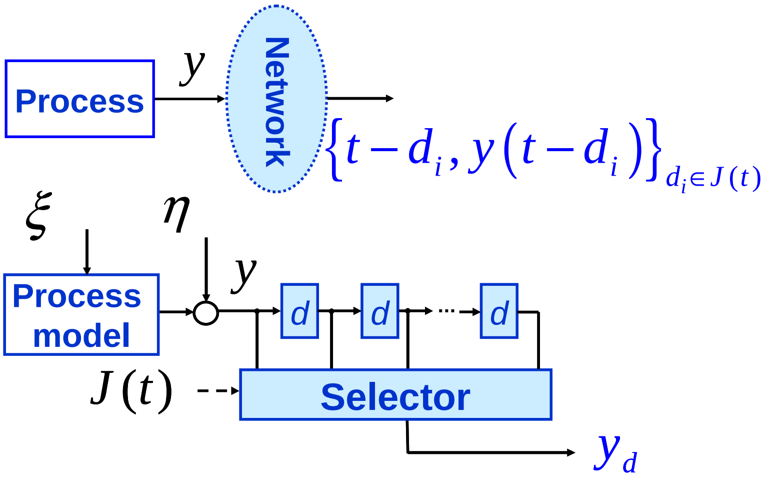


Figure 20: Augmenting plant model by a model of communication delay

The algorithm mentioned above, albeit more economical than standard Kalman filter implementations, may be still complex for some applications, in particular, when the maximum delay is high (or when there are several independently transmitted process outputs). Another line of research in HPL was in developing a suboptimal approximation of a Kalman filter whose implementation is sufficiently simple and fast for the implementation in distributed control systems (for process models of low complexity models occurring in simple control loops); nevertheless, the proposed technique is suitable for a wide range of advanced, medium to large complexity problems when implemented on an appropriate platform for advanced process control and using sampling interval typical for this control layer. This proposed method assumes again the time stamps for the data being available. Next, it is assumed that all samples arrive within a maximum time delay. This latter assumption is rather strict, but an occasional loss of a measurement sample would not have a serious impact on the estimator performance. The proposed solution is a simplification of our earlier result [53]; it further uses ideas similar to those of [56]. In [53] it was found that on condition of a lossless communication, an optimal Kalman filter for systems of time-invariant parameters converges to a filter with switched Kalman gains. Particular Kalman gain used at a specific time-step k depends only on time indices of temporarily missing measurement samples. The number of those gains is finite and can be pre-computed; however, it grows exponentially with the maximum delay. The innovation in this paper is in proposing a suboptimal filter that can replace some (or all) missing data by fictitious values; later, upon receiving a real measurement, the effect of the wrongly entered fictitious item is cancelled and the newly received value is used for a correct update.

3.1.4 Problem formulation

Here we shall outline describe a novel implementation of Kalman filter for systems shown in Figure 20. A detailed description is in our paper [13] State model of the augmented system is given by

$$\left[\begin{array}{c} x_0(k+1) \\ \hline x_d(k+1) \end{array}\right] = \left[\begin{array}{c|cc} A_0 & 0 & 0 \\ \hline C_0 & 0 & 0 \\ 0 & I & 0 \end{array}\right] \left[\begin{array}{c} x_0(k) \\ \hline x_d(k) \end{array}\right] + \left[\begin{array}{c} \xi_0(k) \\ \hline \eta(k) \\ 0 \end{array}\right], \tag{159}$$

where x_0 is a state of the process of dimension $n_0$and $x_d$is a state of the shift register for delayed outputs of dimension $n_{d\max}$. For simplicity, a single output case is described. Variables ξ_0 and η are mutually independent, zero-mean, Gaussian, white noises with covariance matrices given by Q and r_0, respectively. Equation 159 is written, in a compact form, as

$$x(k+1) = Ax(k) + \xi(k). \tag{160}$$

A candidate set of output of the augmented system is given by

$$\left[\begin{array}{c} y(k) \\ y(k-1) \\ y(k-2) \\ \vdots \\ y(k-n_{d\max}) \end{array}\right] = \left[\begin{array}{c|cccc} C_0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & \cdots & 1 \end{array}\right] \left[\begin{array}{c} x_0(k) \\ \hline x_d(k) \end{array}\right] + \left[\begin{array}{c} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{array}\right] \eta(k), \tag{161}$$

i.e., (undelayed) process output and the state of the shift register modeling network delay. The actual process output depends on the data received at the given time step, i.e.,

$$y_d(k) = \left[\begin{array}{ccc} y(k-j_1(k)) & \cdots & y(k-j_m(k)) \end{array}\right]^T, \quad J(k) = \{j_1(k), ..., j_m(k)\}. \tag{162}$$

It is assumed, that the set of measurements received at step k have not been received previously. Note, that the length of the observation vector is varying; if no data are received, this vector is empty. The observation equation thus contains selected rows of 161; it can be expressed as

$$y_d(k) = C_d(k)x(k) + D_r(k)\eta(k). \tag{163}$$

Note that the process noise in 160 and measurement noise in 163 are correlated, if a current, undelayed measurement is obtained.

3.1.5 Preliminaries

According to the standard Kalman filter formula, predicted state mean and covariance are given by

$$\begin{array}{rcl} \hat{x}_{k+1|k} & = & \left(A - L(k)C_d(k)\right)\hat{x}_{k|k-1} + L(k)y_d(k) \\ P_{k+1|k} & = & AP_{k|k-1}A^T - L(k)W(k)L(k)^T + Q \end{array} \tag{164}$$

where $Q$is the process noise covariance , and

$$\begin{array}{rcl} L(k) & = & \left(AP_{k|k-1}C_d^T(k) + S(k)\right)W(k)^{-1}, \\ W(k) & = & C_d(k)P_{k|k-1}C_d(k)^T + D_r(k)r_0D_r(k)^T, \\ S(k) & = & E\left(\xi(k)\eta(k)^T\right)D_r(k)^T. \end{array} \tag{165}$$

The second equation in 164 is the matrix difference Riccati equation, which is the major computational bottleneck in Kalman filter implementation. It should be noted, that a numerically robust implementation is the square root filter updating the Cholesky factor of the state covariance matrix.

Under mild technical assumptions, for time invariant systems, Riccati equation converges to the stabilizing solution of the discrete-time algebraic Riccati equation; in the present setting, this is possible only for transmission sequences with constant delay $J(k) = \{j\}$ for all k. The form of this equation is

$$\bar{P}_j = A\bar{P}_j A^T - \left(A\bar{P}_j C_{dj}^T + S_j\right)\left(C_{dj}\bar{P}_j C_{rj}^T + D_{rj} r_0 D_{rj}^T\right)^{-1}\left(A\bar{P}_j C_{dj}^T + S_j\right)^T + Q, \qquad (166)$$

where $C_{dj}$is the fixed output matrix corresponding to the constant sequence $J(k) = \{j\}$ and $S_j$corresponds to $S(k)$ in 165; it is zero for all $j > 0$.

3.1.6 Main results

The algebraic Riccati equation can be written in the factorized form; let $\bar{P}_j = \bar{H}_j^T \bar{H}_j$. The factor can be computed recursively, for $j = 0, 1, ...$, as

$$\bar{H}_0 = \left[\begin{array}{c} P_0 \\ 0 \end{array}\right], \quad \bar{H}_j = \left[\begin{array}{cc} \bar{H}_{j-1} & \bar{D}_j \end{array}\right]; \quad \bar{D}_1 = L_0\left(C_{d0}\bar{P}_0 C_{d0}^T + r_0\right)^{\frac{1}{2}}, \quad \bar{D}_j = A\bar{D}_{j-1} \qquad (167)$$

There, P_0 is the solution of the algebraic Riccati equation for the unaugmented process model. Vector $L_0$is the injection gain in 165 corresponding to $\bar{P}_0$. The recursive formula 167 corresponds to the Lyapunov difference equation run from $\bar{P}_0$. Vectors $\bar{D}_j$ for $j = 1, ..., n_{d\max}$ can be pre-computed off-line together with $\bar{H}_0$.

The newly proposed representation of the state covariance matrix is as follows:

$$P_{k+1|k} = \bar{P}_{i(k)} + D_{k+1|k} D_{k+1|k}^T. \qquad (168)$$

There, the time-varying index $i(k)$, also called ‘target delay’, is given by

$$i(k) = \begin{cases} \min J(k) & \text{if} \quad \min J(k) \leq i(k-1) \\ \min\{i(k-1)+1, n_{d\max}\} & \text{if} \quad \min J(k) > i(k-1) \text{ or } \quad J(k) = \{\}. \end{cases} \qquad (169)$$

Riccati equation is then replaced by a recursive formula for the factor $D_{k+1|k}$. This formula is a matrix difference equation with jumps. The jumps occur upon the change of the target delay $i(k)$. The factor $D_{k+1|k}$ then can be partitioned as

$$D_{k|k-1} = \left[\begin{array}{cc} D_{Ik|k-1}^T & D_{Fk|k-1}^T \end{array}\right]^T,$$

where $D_{Ik|k-1}$ and $D_{Fk|k-1}^T$ stand for the infinite and finite-time response parts, respectively. The partitioning is such that the number of rows of $D_{Ik|k-1}^T$ equals $n_0 + i(k)$. Further, under proper initialization, the infinite time response part can be represented as

$$D_{Ik|k-1} = E_{Ii(k)} Z_{k|k-1}, \quad E_{I0} = I, \quad E_{Ii} = \left[\begin{array}{c} E_{Ii-1}A_0 \\ C_0 \end{array}\right] \qquad (170)$$

The infinite response part thus can be reduced to a matrix whose number of rows equals to the number of states of the unaugmented process model. The dynamics of the matrix factor between the jumps, i.e., for $i(k) = i(k-1) < n_{d\max}$, is given by

$$\left[\begin{array}{c} Z_{k+1|k} \\ D_{Fk+1|k} \end{array}\right] = \left[\begin{array}{c|cc} A_0 - L_0 C_0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & J & 0 \end{array}\right] \left[\begin{array}{c} Z_{k|k-1} \\ D_{Fk|k-1} \end{array}\right] \Phi\left(Z_{k|k-1}, D_{Fk|k-1}\right). \qquad (171)$$

There, J is a diagonal matrix with diagonal elements containing ones or zeros, depending on the set $J(k)$. This corresponds to the fact that the process outputs, once received, are known and hence, the conditional covariance of the corresponding states of the delay chain is zero. Matrix Φ is computed using a QR decomposition of $C_d(k)D_{k|k-1}$; details are omitted and can be found in [13]. This matrix provides a loose coupling between the finite- and infinite-time response parts. Further, $\|\Phi(k)\| \leq 1$ and $\|\Phi(k)\| \to 1$ as $k \to \infty$. The dynamics of the infinite-time response is bounded from above by the matrix $A_0 - L_0 C_0$, which is the dynamics of the asymptotic Kalman filter of the unaugmented process model. Finally, it can be observed that

$$col \dim \left(D_{k+1|k}\right) = col \dim \left(D_{k|k-1}\right) - \dim\left(J(k)\right) + 1 \text{ for } \dim\left(J(k)\right) > 0. \tag{172}$$

This implies that if there is more than one missing measurement received at a time, the rank of the matrix factor $D_{k+1|k}$ decreases.

Now we only outline what happens when $i(k)$ changes: for $i(k) < i(k-1)$, the factors are augmented and subsequently rank-reduced as

$$\left[\begin{array}{c|c} Z^{+}_{k|k-1} & 0 \\ D^{+}_{Fk|k-1} & 0 \end{array}\right] = \left[\begin{array}{cccc} E_{I(m(k-1)-i(k)+1)} Z_{k|k-1} & \bar{D}_{1A} & \dots & \bar{D}_{(m(k-1)-i(k))A} \\ D_{Fk|k-1} & 0 & \dots & 0 \end{array}\right] T \tag{173}$$

where T is an orthogonal matrix and $\bar{D}_{jA}$ denote rows 1 to $n_0 + i(k-1) - i(k)$ of $\bar{D}_j$. Note that the vertical matrix partitioning is different on both sides of the equation: $row \dim \left(D^{+}_{Fk|k-1}\right)$ equals to $row \dim \left(D_{Fk|k-1}\right) + i(k-1) - i(k)$. Then, matrices $Z^{+}_{k|k-1}$ and $D^{+}_{Fk|k-1}$ are substituted for $Z_{k|k-1}$ and $D_{Fk|k-1}$, respectively, in 171. For cases when $i(k) = i(k-1) + 1$, we refer to our paper [13].

We note that the orthogonal rank reduction in 173 is the most demanding operation within the filter update cycle. It can be shown that in the worst-case it is of lower complexity than the QR decomposition involved in the square-root filter.

3.1.7 Summary

We proposed a new algorithm of Kalman filter for systems with communication delays, when packet loss, out-of-order measurements and bursts of received data are admissible. The algorithm exploits a particular problem structure and uses pre-computed gains. On the worst case of receiving a burst of data after a long period of receiving none, the filter update is slightly less complex than standard implementations. Under normal operation of slightly varying communication delay, the decrease of complexity is significant. In addition, an insight was gained into the structure of the state covariance matrix and its dynamics, which is of an independent interest.

3.2 Suboptimal Kalman filter

L. Baramov, D. Pachner and V. Havlena

3.2.1 Introduction

Here we shall consider a similar setting as in the previous section – state estimation for a linear, time-invariant model disturbed by Gaussian noise, where measurements arrive with variable delay, possibly out-of-order. We shall further assume that all measurements arrive within a maximum delay $n_{d\max}$. It was found in our previous work in [53], under these conditions, Kalman filter converges to a parameter-varying system, where Kalman gain depends only on the set of missing samples. Let this set be denoted as

$$.m(k) = \{(i,j)|\ y_i(k-j) \text{ is missing at } k\} \tag{174}$$

There, y_i denotes ith elements of the observation vector and j is the communication delay. Further, let $\bar{m}(k)$ denote the complement of $m(k)$ to the set $\{0, \dots, k\}$, i.e. the set of all received measurements. As was shown in [53], on the assumption of the maximum communication delay, Kalman filter converges to a filter whose gain is fully determined by this set, i.e., the data update of the state mean can be written as

$$\hat{x}_{k|\bar{m}(k)\cup(i,j)} = \hat{x}_{k|\bar{m}(k)} + K(m(k), i, j)\left(y_i(k-j) - C_i \hat{x}_{k-j|\bar{m}(k)}\right). \tag{175}$$

From this it follows: first, we have to keep track only on those past output estimates that correspond to missing samples, i.e., $\hat{y}_i\left(k-j|\bar{m}(k)\right) = C_i \hat{x}_{k-j|\bar{m}(k)}$, $j \in m(k)$. Second, as a result of the bounded delay assumption, there is a finite number of values of $m(k)$ and the same number of gains $K(m(k), i, j)$, which can be pre-computed off-line. The set $m(k)$ can be considered as filter state and thus Kalman filter is a time invariant system. On the other hand, the number of gains grows exponentially with $n_{d\max}$, which may result in storage problems with large delays. The circumventing the problem we would be re-calculating the gains each step, e.g., using the technique proposed in the previous sub-section, which inevitably results in an increased computational load. The proposed way to keep the filter very simple is to resort to a sub-optimal filter. It is based on the technique of replacing a sample that has not arrived in the due time by a 'fake' measurement and upon receiving the correct sample later, canceling the effect of the wrong data and fusing the new information correctly. This results in restoring optimality; a similar approach is done in [56] for time-varying systems with full non-steady Kalman filter. Here, we present a solution for time-invariant systems using a switched linear filter with a moderate number of pre-computed gains.

3.2.2 Main result

We will consider a situation when at time k a correct process value $y_i(k-j)$ is not available and some other value $y_i^{(1)}(k-j)$ is supplied to Kalman filter in place of it. Suppose that later on, at time step $k+h$, the correct value of $y_i(k-j)$ is received by the filter. The correction of the filter state and restoring the optimality is possible using a set of pre-computed gains. First, fusing the 'fake' data $y_i^{(1)}(k-j)$ results in the following data step at time k

$$\hat{x}_{k|\bar{m}(k)\cup(i,j)} = \hat{x}_{k|\bar{m}(k)} + K(m(k), i, j)\left(y_i^{(1)}(k-j) - C\hat{x}_{k-j|\bar{m}(k)}\right) \tag{176}$$

Now, assume that the correct sample arrives h time-steps later. Then, after updating the state as in 175 by all freshly arrived samples that have not been replaced by fakes in the past, the correction of the incorrectly fused sample $y_i^{(1)}(k-j)$ is done as

$$\hat{x}_{k+h|\bar{m}(k+h)} := \hat{x}_{k+h|\bar{m}(k+h)} + K\left(m(t+h)\ \cup\left\{(i, j+h)\right\}, i, j+h\right)\left(y_i(k-j) - y_i^{(1)}(k-j)\right) \tag{177}$$

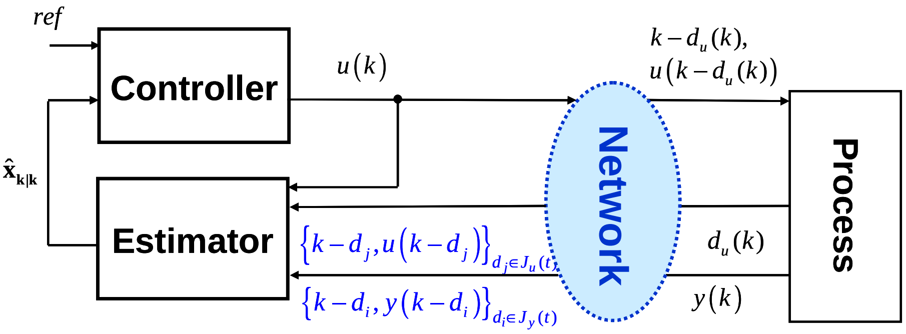
Note that the two Kalman gains used in 176 and 177 are generally different. Note that the correction requires the 'fake' measurement to be stored until the correct value is available. In the case when the filter was optimal up to time k and, if no further sample was missed, the optimality is restored at time $k+h$ after the correction 177 was performed. The non-optimality between times k and $k+h$ is due to the fact that the estimator was mislead by pretending that it was given correct data at time k. As a result, the Kalman filter wrongly considers the state covariance matrix smaller than the real one and therefore, its responsiveness to prediction errors is reduced. Nevertheless, due to the assumption that the correct sample arrives within a finite time horizon and the subsequent optimality restoration, the impact on the overall estimation quality is limited.

This approach greatly reduces the number of gains to be stored, depending on the strategy used. For instance, we can handle optimally samples arriving with delay up to n_{d1}, while others, whose delay is between n_{d1} and $n_{d\max}$, are handled by replacement/correction technique. The savings in the number of gains to be stored is due to the reduction of possible values of the set $m(k)$. As a special case we can consider the one (also used in [56] for time-varying filters) which replaces (and later corrects) all delayed samples. Then, the number of gains to be stored equals to the product of the number of sensors and the maximum delay, corresponding to $m(k) = \{(i, j)\}$.

As for the value to be used in place of the missing measurement, a natural choice is using the available estimate, i.e., $y_i^{(1)}(k-j) = C\hat{x}_{k-j|\bar{m}(k)}$.

3.2.3 Delayed actuator inputs

In this subsection we shall briefly outline the proposed way of handling delayed (deterministic) inputs to the process. In the standard settings with point-to-point wired configuration, actuator commands generated by a controller naturally affects the process states; but in standard cases with the wired connection between the actuator and the controller, these commands are deterministic and known so that are easily handled by Kalman filter in its time step. Also, transmitting actuator commands over networks occurs less often than transmitting process measurements – actuators, unlike sensors, are typically not of miniature dimensions and need external power supply; hence, they are likely to be accessible by the standard 'wired' plant automation infrastructure. On the other hand, there are indeed many applications where the manipulated variable is transmitted and this fact has to be taken into account by the estimator.



ref
Controller
$u(k)$
$k-d_u(k),$
$u(k-d_u(k))$
Network
Process
$\hat{\mathbf{x}}_{\mathbf{k|k}}$
Estimator
$\{k-d_j, u(k-d_j)\}_{d_j \in J_u(t)}$
$\{k-d_i, y(k-d_i)\}_{d_i \in J_y(t)}$
$d_u(k)$
$y(k)$

Figure 21: A possible configuration of the state estimator in the control loop

A possible configuration of closed loop controller with Kalman filter is in Figure 21. The controller here is considered in this section only as a generator of control signal $u(k)$. This signal is transmitted over the network and received at the process side by the actuator node. It is possible that these command signals arrive late or never; the actuator node needs to extrapolate the missing signal using the command history and an extrapolation model – details are omitted in this subsection. The actuator node has also to acknowledge whether and when the particular command was received; this acknowledgement is transmitted back over the network – again, subject to time delays and information loss; alternatively, as in Figure 21, the actuator can send a message containing the true value of the actuator command that was applied at the given time. To include uncertain input delay into the framework developed for the (sub)optimal Kalman filter with variable output delay, the actually applied actuator command is treated as a process output, possibly delayed. As process input we use a command estimate $\tilde{u}$ computed by from the command history and a network model (in the form of a Markov chain) predicting the most likely command value received by the actuator. The input model is introduced to account for discrepancies between the estimated commands $\tilde{u}$ and those received, that can be inferred from the information received by the estimator. The input model can be chosen

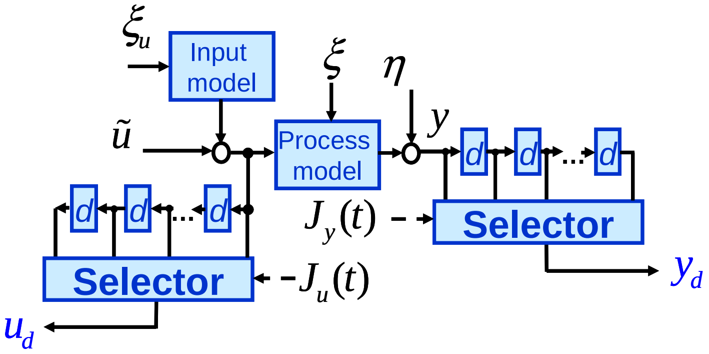


Figure 22: Observation model for a system with inputs and outputs subject to variable delays

as the above mentioned extrapolation model (in the simplest case it can be the random walk model consisting of one integrator).Thus, the same framework as described in the previous sub-section can be used in this extended case.

3.2.4 Summary

In this subsection we have proposed a method of sub-optimal Kalman filter. This filter handles (some of) temporarily missing measurements by replacing them by a fictitious value, e.g., a local prediction of the particular output. Upon arrival of the missing item, the overall state of Kaman filter is corrected and the optimality is restored. The algorithm uses pre-computed filter gains and hence is very fast.

4 Network-Aware Observer-Based Control

4.1 Decentralized Observed-Based Control of Large-Scale Networked Systems

N.W. Bauer, M.C.F. Donkers, W.M.P.H. Heemels, N. van de Wouw

4.1.1 Introduction

Recently, there has been an enormous interest in control of large-scale systems that are physically distributed over a wide area. Examples of such distributed systems are electrical power distribution networks, water transportation networks, industrial factories and energy collection networks (such as wind farms). This work considers stability analysis and controller design for this class of systems. This problem setting has a number of features that seriously challenge the controller design.

Firstly, the controller is decentralized in the sense that it consists of a number of local controllers that do not share information. Although a centralized controller could alternatively be considered, the achievable bandwidth associated with using a centralized control structure would be limited by long delays induced by the communication between the centralized controller and distant sensors and actuators over a (e.g. wireless) communication network [3].

Secondly, when considering control of a large-scale system, it would be unreasonable to assume that all states are measured. Therefore an output-based controller is needed. In particular, we consider an observer-based control setup. Note that an observer-based controller offers the advantage of reducing the number of sensors, which alleviates the demands on the network design. However, it has been proven that, in general, it is hard to obtain decentralized observers providing state estimate converging to the ‘true’ states [66].

Finally, the observer-based controller needs to have certain robustness properties when using a communication network. Indeed, the advantages of using a wired/wireless network are inexpensive and easily modifiable communication links. However, the drawback is that the control system is susceptible to undesirable (possibly destabilizing) side-effects see e.g. [27, 86]. There are roughly five recognized Networked Control System (NCS) side-effects: time-varying delays, packet dropouts, varying sampling intervals, quantization and communication constraints (the latter meaning that not all information can be sent over the network at once). For modeling simplicity, we only consider varying sampling times and communication constraints in this work.

Resuming, we note that although this decentralized observer-based control structure is reasonable to use, its design is extremely complex due to the fact that we simultaneously face the issues of (i) a decentralized control structure (ii) limited measurement information and (iii) communication side-effects. The contribution of this paper is threefold: a model describing the controller decentralization and the communication side-effects is derived for analysis, a way to assess robust stability of the closed loop in the face of communication imperfections is given and an approach towards the design of observer-based controllers is provided.

The outline of this paper is as follows: In Section 4.1.2 the general problem description and the closed-loop model will be constructed. Construction of the model covers the plant decomposition needed to establish a decentralized controller structure, the network constraints and the descriptions of the observer-based decentralized control design. We will then propose LMI-based stability conditions in Section 4.1.3. In Section 4.1.4 we present a constructive design procedure for these decentralized observer-based controllers for the case of periodic communication protocols. Finally an example will be presented in Section 4.1.5 and some suggestions for future work will be discussed in Section 4.1.6.

Nomenclature

The following notational conventions will be used. diag$(A_1, \dots, A_N)$ denotes a block-diagonal matrix with the matrices $A_1, \dots, A_N$ on the diagonal and $A^\top \in \mathbb{R}^{m \times n}$ denotes the transpose of the matrix $A \in \mathbb{R}^{n \times m}$. For a vector $x \in \mathbb{R}^n$, we denote $\|x\| := \sqrt{x^\top x}$ its Euclidean norm. We denote by $\|A\| := \sqrt{\lambda_{\max}(A^\top A)}$ the spectral norm of a matrix A, which is the square-root of the maximum eigenvalue of the matrix $A^\top A$. For brevity, we sometimes write symmetric matrices of the form $\left[\begin{smallmatrix} A & B \\ B^\top & C \end{smallmatrix}\right]$ as

$\begin{bmatrix} A & B \\ \star & C \end{bmatrix}$.

4.1.2 The Model & Problem Definition

We consider a continuous-time linear plant:

$$\begin{cases} \dot{x}(t) &= Ax(t) + B\hat{u}(t) \\ y(t) &= Cx(t) \end{cases} \tag{178}$$

with state $x \in \mathbb{R}^n$, control input $\hat{u} \in \mathbb{R}^m$ and measured output $y \in \mathbb{R}^p$. The goal of the paper is to present an approach for the analysis and design of a stabilizing controller that has the following features:

- discrete-time;
- decentralized;
- output-based;
- robust with respect to uncertain time-varying sampling intervals $h_k \in [\underline{h}, \bar{h}]$ for all $k \in \mathbb{N}$;
- communication constrained: not all outputs and inputs can be communicated simultaneously and a protocol schedules which information is sent at a transmission instant

The decentralized controllers $C^{(i)}, i = 1, ..., N$, communicate with the sensors and actuators of the plant via a shared network. The general setup is depicted in Fig. 23.

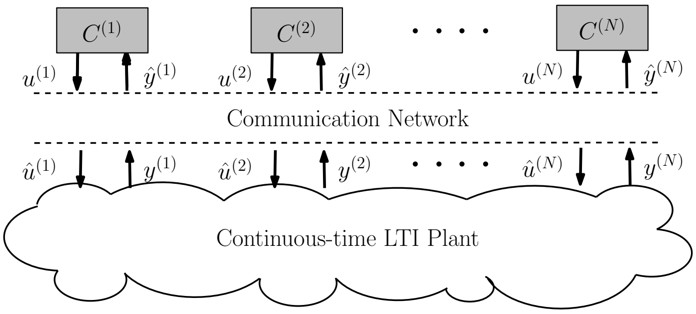


Figure 23: Decentralized Networked Control System.

In this paper, the plant will be divided into subsystems, each of which, are controlled by a discrete-time observer-based controller whose subsystem model is based on a nominal sampling interval. In Section 4.1.2, we determine a nominal sampling interval with a corresponding plant discretization and present a decomposition of the plant. In Section 4.1.2 a description of the network imperfections is provided. In Section 4.1.2 a switching observer-based control structure will be presented and, finally, in Section 4.1.2 a closed-loop model suitable for stability analysis is derived.

Plant Decomposition

Since we are aiming to design N model-based discrete-time linear observer-based controllers, the continuous-time plant needs to be divided into N discrete-time subsystems to use as sub-models. First, we will discretize the continuous-time plant, after which, the states of the continuous-time plant will be partitioned, leading to N disjoint discrete-time systems.

Plant Discretization - It is well known that a linear continuous-time system (178) with a zero-order-hold assumption on the inputs $\hat{u}(t)$ can be exactly discretized to

$$\mathcal{P} := \begin{cases} x_{k+1} &= A^\star x_k + B^\star \hat{u}_k \\ y_k &= Cx_k \end{cases}, \ k \in \mathbb{N}_{\geq 0}, \tag{179}$$

where $h_\star$ is a suitably chosen nominal constant sampling interval, $A^\star := e^{Ah_\star}$ and $B^\star := \int_0^{h_\star} e^{As} ds B$. In (179), $x_k = x(t_k), y_k = y(t_k)$, with t_k the sampling instants, and $\hat{u}_k$ is the discrete-time control action available at the plant at $t = t_k$, i.e. $\hat{u}(t) = \hat{u}_k$ for all $t \in [t_k, t_{k+1}), k \in \mathbb{N}$.

Plant Decomposition - The system $\mathcal{P}$ in (179) will be decomposed into N interconnected subsystems. Choosing the decomposition is a challenging task, as there are many aspects to be taken into account, i.e. sensor/actuator physical location, subsystem interaction, computational effort of the control law, input/output relations, etc. For *decentralized* design, reducing the interactions between subsystems is highly desired since smaller interaction will generally increase the likelihood of the decentralized controllers being successful. Considering that the goal of the decomposition is keeping the interaction between the subsystems small (while maintaining a minimal number of subsystems), we propose to use the ϵ-decomposition technique

The ϵ-decomposition is an algorithm for finding a permutation matrix P such that *each element* of all subsystem coupling matrices between subsystems, has magnitude no greater than $\epsilon \geq 0$. This algorithm offers the advantage of only searching for permutations of a system, which preserves any physical meaning of the original state vector. By using this algorithm we can find a P that expresses the entire plant as a collection of interconnected (coupled) subsystems

$$\mathcal{P}^{(i)} := \begin{cases} z_{k+1}^{(i)} &= \bar{A}_i z_k^{(i)} + \bar{B}_i \hat{u}_k^{(i)} \\ &\quad + \sum_{\substack{j=1 \\ j\neq i}}^{N} \left(\bar{A}_{i,j} z_k^{(j)} + \bar{B}_{i,j} \hat{u}_k^{(j)} \right) \\ y_k^{(i)} &= \bar{C}_i z_k^{(i)} + \sum_{\substack{j=1 \\ j\neq i}}^{N} \bar{C}_{i,j} z_k^{(j)} \end{cases}, \tag{180}$$

for $i = 1, ..., N$, where $z = P^{-1}x$ is the state vector of the permuted system $\bar{A} = P^{-1}A^\star P$, $\bar{B} = P^{-1}B^\star$, $\bar{C} = CP$, $u_k^{(i)} \in \mathbb{R}^{m_i}$, and $y_k^{(i)} \in \mathbb{R}^{p_i}$. Without loss of generality, we only consider disjoint decompositions, that is, $z = (z^{(1)\top}, z^{(2)\top}, ..., z^{(N)\top})^\top$, $u = (u^{(1)\top}, u^{(2)\top}, ..., u^{(N)\top})^\top$ and $y = (y^{(1)\top}, y^{(2)\top}, ..., y^{(N)\top})^\top$. As such, every state, output and input are attributed to only one subsystem and the subsystem interaction matrices are denoted $\bar{A}_{i,j}, \bar{B}_{i,j}, \bar{C}_{i,j}, j \neq i$. Throughout this paper we use the decomposition $\bar{A} = A_d + A_c$, where $A_d :=$ diag$(\bar{A}_1, \bar{A}_2, ..., \bar{A}_N)$. The $\bar{B}, \bar{C}$ matrices can be expressed similarly. With this notation, we can equivalently express (180) as

$$\mathcal{P} = \begin{cases} z_{k+1} &= A_d z_k + B_d \hat{u}_k + (A_c z_k + B_c \hat{u}_k) \\ y_k &= C_d z_k + C_c z_k \end{cases}.$$

The control structure for a chosen decomposition is depicted in Fig. 24, where the i^{th} controller is controlling only the i^{th} subsystem.

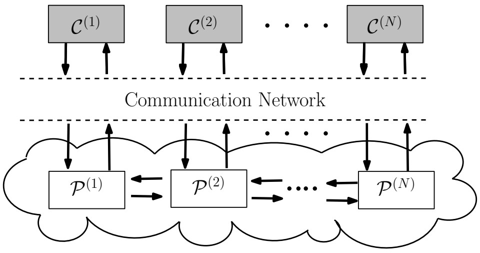


Figure 24: Decentralized NCS After Decomposition

Network Description

Communication between sensors, actuators and controllers will take place via a shared network, see Fig. 24. Here, we will consider two network effects: namely, varying sampling intervals and communication constraints, where the latter imposes the need for a scheduling protocol to determine what input and output data is transmitted at each sampling time.

In Section 4.1.2, we assumed a constant sampling interval $h_\star$ to arrive at subsystem models used by the controller. However, due to the nature of the network, the actual sampling times $t_k, k \in \mathbb{N}$, are not necessarily equidistant in time. Assuming that the sampling intervals $h_k = t_{k+1} - t_k$ are contained in $[\underline{h}, \bar{h}]$ for some $0 \leq \underline{h} \leq \bar{h}$, i.e. $h_k \in [\underline{h}, \bar{h}]$ for all $k \in \mathbb{N}$, the exact discrete-time equivalent of (178), after the permutation, is

$$\mathcal{P}_{h_k} := \begin{cases} z_{k+1} & = & \bar{A}_{h_k} z_k + \bar{B}_{h_k} \hat{u}_k \\ y_k & = & \bar{C} z_k, \end{cases} \tag{181}$$

where $\bar{A}_{h_k} := P^{-1} e^{Ah_k} P$, $\bar{B}_{h_k} := P^{-1} \int_0^{h_k} e^{As} ds B$ and $\bar{C} := CP$. It is important to note that the observer-based controllers will use subsystem models that are based on the constant sampling interval $h_\star$, so variation in the sampling interval prevent the state estimation error from converging to zero.

Since the plant and controller are communicating through a network with communication constraints, the actual input of the plant $\hat{u}_k \in \mathbb{R}^m$ is not equal to the controller output u_k and the actual input of the controller $\hat{y}_k \in \mathbb{R}^p$ is not equal to the plant output $y_k \in \mathbb{R}^p$. Instead, $\hat{u}_k$ and $\hat{y}_k$ are networked versions of u_k and y_k, respectively.

To explain the effect of communication constraints and thus the difference between $\hat{y}_k$ and y_k and $\hat{u}_k$ and u_k one has to realize that the plant has n_u sensors and n_y actuators. These sensors and actuators are grouped into $n_T \leq n_u + n_y$ nodes. At each sampling time t_k, one node obtains access to the network and transmits its corresponding u and/or y values. Only the transmitted values will be updated, while all other values remain unchanged. This means the constrained data exchange can be expressed as

$$\begin{cases} \hat{u}_k & = \Gamma^u_{\sigma_k} u_k + (I - \Gamma^u_{\sigma_k}) \hat{u}_{k-1} \\ \hat{y}_k & = \Gamma^y_{\sigma_k} y_k + (I - \Gamma^y_{\sigma_k}) \hat{y}_{k-1}, \end{cases} \tag{182}$$

where $\Gamma^u_l \in \mathbb{R}^{m \times m}$ and $\Gamma^y_l \in \mathbb{R}^{p \times p}$, for $l = 1, ..., n_T$, are diagonal matrices where the j^{th} diagonal value is 1 if the j^{th} input or output, respectively, belongs to node l and zero elsewhere. Without loss of generality, we will assume the matrices Γ^u_l and Γ^y_l can be divided in $\Gamma^u_{i,l}$ and $\Gamma^y_{i,l}$, for $i \in \{1, ..., N\}$ such that $\Gamma^u_l := \mathrm{Diag}(\Gamma^u_{1,l}, \Gamma^u_{2,l}, ..., \Gamma^u_{N,l})$ and $\Gamma^y_l := \mathrm{Diag}(\Gamma^y_{1,l}, \Gamma^y_{2,l}, ..., \Gamma_{N,l})$, where $\Gamma^u_{i,l} \in \mathbb{R}^{m_i \times m_i}$ and $\Gamma^y_{i,l} \in \mathbb{R}^{p_i \times p_i}$ are matrices corresponding to inputs and outputs, respectively, of the i^{th} subsystem.

The value of $\sigma_k \in \{1, 2, ..., n_T\}$ indicates which node is given access to the network. The switching functions determining σ_k are known as protocols. In this paper we focus on the general class of periodic protocols [18], which are characterized by $\sigma_{k+\tilde{N}} = \sigma_k$ for some period $\tilde{N} \geq n_T,\ \tilde{N} \in \mathbb{N}$. The well-known Round Robin protocol [73] belongs to this class of periodic protocols.

Finally, we introduce the network-induced errors

$$\begin{cases} e^u_k & := \hat{u}_{k-1} - u_k \\ e^y_k & := \hat{y}_{k-1} - y_k, \end{cases} \tag{183}$$

where e^u_k and e^y_k will be referred to as the (network-induced) input error and output error, respectively.

Decentralized Networked Observer-Based Controllers

In this paper we will use decentralized observer-based controllers in the sense that for each subsystem of the plant we have one observer-based controller and the controllers do not exchange information. Therefore, the individual observers have no information about externally coupled states. As a consequence, it is desired to ensure that coupling between subsystems is minimal since ignored coupled dynamics will act as an unknown disturbance input to the decoupled observers. Furthermore, the model-based controllers will adopt switching gains to deal with the communication constraints

$$\tilde{A}_{c,h_k,\sigma_k} = \begin{bmatrix} A_d - L_{\sigma_k} \Gamma^y_{\sigma_k} C_d + \Delta B_{c,h_k} K_{\sigma_k} & L_{\sigma_k} \Gamma^y_{\sigma_k} C_c - \Delta A_{c,h_k} + \Delta B_{c,h_k} K_{\sigma_k} & -\Delta B_{c,h_k} (I - \Gamma^u_{\sigma_k}) & 0 \\ -\bar{B}_{h_k} K_{\sigma_k} & \bar{A}_{h_k} - \bar{B}_{h_k} K_{\sigma_k} & \bar{B}_{h_k} (I - \Gamma^u_{\sigma_k}) & 0 \\ K_{\sigma_k} (A_d - L_{\sigma_k} \Gamma^y_{\sigma_k} C_d - B_d K_{\sigma_k} - I) & K_{\sigma_k} (A_d + L_{\sigma_k} \Gamma^y_{\sigma_k} C_c - B_d K_{\sigma_k} - I) & (K_{\sigma_k} B_d + I)(I - \Gamma^u_{\sigma_k}) & 0 \\ \bar{C} \bar{B}_{h_k} K_{\sigma_k} & \bar{C}(I - \bar{A}_{h_k} + \bar{B}_{h_k} K_{\sigma_k}) & -\bar{C} \bar{B}_{h_k} (I - \Gamma^u_{\sigma_k}) & I - \Gamma^y_{\sigma_k} \end{bmatrix} \tag{187}$$

effectively. The i^{th} networked observer-based controller is given by

$$\mathcal{C}^{(i)}_{\sigma_k} := \begin{cases} \tilde{z}^{(i)}_{k+1} &= \bar{A}_i \tilde{z}^{(i)}_k + \bar{B}_i \hat{u}^{(i)}_k \\ & \quad + L_{i,\sigma_k} \Gamma^y_{i,\sigma_k} (\hat{y}^{(i)}_k - \bar{C}_i \tilde{z}^{(i)}_k) \\ u^{(i)}_k &= -K_{i,\sigma_k} \tilde{z}^{(i)}_k, \end{cases} \tag{184}$$

where $\tilde{z}^{(i)}_{k+1}$ represents the state estimate of the i^{th} observer at time $k+1$ for the plant state $z^{(i)}_{k+1}$ and the output injection matrices L_{i,σ_k}, $i \in \{1,..,N\}$, $\sigma_k \in \{1,...,n_T\}$ will be designed to stabilize the dynamics of the state estimation error $\eta_k := \tilde{z}_k - z_k$. We adopt a switched-observer structure (notice the σ_k-dependence in (184)) to deal with the presence of communication constraints. Switched observers have received much attention in the past decade [4,74,79]. Γ^y_{i,σ_k} in (184) is used so that the standard output injection is only applied to the newly received measurements. If no measurements are received ($\Gamma^y_{i,\sigma_k} = 0$ for some σ_k) then (184) reduces to a standard model-based prediction step.

Similar to the plant, the dynamics of all the controllers (184) can be described by a single discrete-time system, which will consist of block diagonal matrices due to the decoupled nature of the controllers

$$\mathcal{C}_{\sigma_k} = \begin{cases} \tilde{z}_{k+1} &= A_d \tilde{z}_k + B_d \hat{u}_k \\ & \quad + L_{\sigma_k} \Gamma^y_{\sigma_k} (\hat{y}_k - C_d \tilde{z}_k) \\ u_k &= -K_{\sigma_k} \tilde{z}_k, \end{cases} \tag{185}$$

where $L_{\sigma_k} = \text{diag}(L_{1,\sigma_k},\ L_{2,\sigma_k}, ..., L_{N,\sigma_k})$ and $K_{\sigma_k} = \text{diag}(K_{1,\sigma_k},\ , K_{2,\sigma_k}, ..., K_{N,\sigma_k})$.

Closed-Loop Model

To derive an expression for the closed loop, we will adopt the state vector $\bar{x}_k = [\eta_k^\top \quad z_k^\top \quad e^{u\,\top}_k \quad e^{y\,\top}_k]^\top$. Combining (181), (182), (183), and (185) the entire closed-loop system can be represented by the following switched uncertain discrete-time system

$$\bar{x}_{k+1} = \tilde{A}_{c,h_k,\sigma_k} \bar{x}_k, \tag{186}$$

where $\tilde{A}_{c,h_k,\sigma_k}$ is given by (187) with

$$\Delta A_{c,h_k} := \bar{A}_{h_k} - A_d = (\bar{A}_{h_k} - \bar{A}) + A_c$$
$$\Delta B_{c,h_k} := \bar{B}_{h_k} - B_d = (\bar{B}_{h_k} - \bar{B}) + B_c.$$

The $\Delta A_{c,h_k}$ term consists of two terms being $(\bar{A}_{h_k} - \bar{A})$, which is caused by a 'clock skew' $(h_k - h_\star)$ effect, added to A_c, which is caused by the subsystem coupling. The same applies to $\Delta B_{c,h_k}$. The 'clock skew' effect is an arbitrarily time-varying term (due to time-varying sampling intervals) while the 'neglected coupling' effect is a deterministic disturbance, which is the result of both the nominal sampling time and the chosen decomposition.

4.1.3 Stability Analysis

In this section, we analyze whether the system (186), (187), with given K_{σ_k} and L_{σ_k}, is stable for some given bounds on the sampling interval, i.e. $h_k \in [\underline{h}, \overline{h}]$ for all $k \in \mathbb{N}$. The stability analysis is based on the ideas in [17], in which stability of networked control systems is discussed. As in [17], the uncertain parameter h_k, $k \in \mathbb{N}$ appears nonlinearly in (187) through $\bar{A}_{h_k}$ and $\bar{B}_{h_k}$. To make the system amenable for analysis, a procedure is proposed to overapproximate system (186), (187) by a polytopic system with norm-bounded additive uncertainty, i.e.,

$$\bar{x}_{k+1} = \sum_{j=1}^{M} \alpha^l_k \left(F_{\sigma_k,j} + G_j \Delta_k H_{\sigma_k}\right) \bar{x}_k, \tag{188}$$

where $F_{l,j} \in \mathbb{R}^{n \times n}$, $G_j \in \mathbb{R}^{n \times q}$, $H_l \in \mathbb{R}^{q \times n}$, for $l \in \{1, \ldots, n_T\}$ and $j \in \{1, \ldots, M\}$, with M the number of vertices of the polytope. The vector $\alpha_k = [\alpha^1_k \ldots \alpha^M_k]^\top \in \mathcal{A}$, $k \in \mathbb{N}$, is time-varying with

$$\mathcal{A} = \Big\{ \alpha \in \mathbb{R}^M \,\Big|\, \textstyle\sum_{j=1}^{M} \alpha^j = 1 \text{ and } \alpha^j \geq 0 \quad \text{for } j \in \{1, \ldots, M\} \Big\}$$

and $\Delta_k \in \boldsymbol{\Delta}$, where $\boldsymbol{\Delta}$ is a norm-bounded set of matrices in $\mathbb{R}^{q\times q}$ that describes the additive uncertainty. Equation (188) is an overapproximation of (186) in the sense that for all $l \in \{1,\dots,n_T\}$, it holds that

$$\begin{aligned} &\Big\{\tilde{A}_{c,h,l} \mid h \in [\underline{h},\overline{h}]\Big\} \\ &\quad\subseteq \Big\{ \textstyle\sum_{j=1}^{M} \alpha^j \left(F_{l,j} + G_j \Delta H_l\right) \lvert\, \alpha \in \mathcal{A}, \Delta \in \boldsymbol{\Delta}\Big\}. \end{aligned} \tag{189}$$

We now provide a gridding-based procedure to overapproximate system (186), such that (189) holds, after which we can provide conditions for stability.

Procedure 1.

- *Select M distinct sampling intervals $\tilde{h}_1,\dots,\tilde{h}_M$ as grid points, such that $\underline{h} =: \tilde{h}_1 \le \tilde{h}_2 < \ldots < \tilde{h}_{M-1} \le \tilde{h}_M := \overline{h}$.*
- *Define*

  $$F_{l,j} := \tilde{A}_{c,\tilde{h}_j,l}.$$

  Decompose the matrix A, as given in (178)*, into its real Jordan form [31], i.e. $A := T\Lambda T^{-1}$, where T is an invertible matrix and*

  $$\Lambda = \operatorname{diag}(\Lambda_1,\dots,\Lambda_L)$$

  with $\Lambda_i \in \mathbb{R}^{n_i\times n_i}$, $i \in \{1,\dots,L\}$, the i-th real Jordan block of A.
- *Compute for each line segment $S_m = [\tilde{h}_m, \tilde{h}_{m+1}]$, $m \in \{1,\dots,M-1\}$, and for each real Jordan block Λ_i, $i \in \{1,\dots,L\}$, the worst case approximation error, i.e.*

  $$\begin{aligned} &\tilde{\delta}^A_{i,m} = \\ &\sup_{\substack{\tilde{\alpha}^1+\tilde{\alpha}^2=1,\\ \tilde{\alpha}^1,\tilde{\alpha}^2\ge 0}} \left\| e^{\Lambda_i(\tilde{\alpha}^1\tilde{h}_m + \tilde{\alpha}^2\tilde{h}_{m+1})} - \sum_{j=1}^{2} \tilde{\alpha}^j e^{\Lambda_i \tilde{h}_{m+j-1}} \right\|, \end{aligned} \tag{190a}$$

  $$\tilde{\delta}^E_{i,m} = \sup_{\substack{\tilde{\alpha}^1+\tilde{\alpha}^2=1,\\ \tilde{\alpha}^1,\tilde{\alpha}^2\ge 0}} \left\| \sum_{j=1}^{2} \tilde{\alpha}^j \int_{\tilde{h}_{m+j-1}}^{\tilde{\alpha}^1\tilde{h}_m + \tilde{\alpha}^2\tilde{h}_{m+1}} e^{\Lambda_i s} ds \right\|. \tag{190b}$$

  For a detailed explanation of the origin of the approximation error bounds, see [17].
- *Map the obtained bounds* (190) *at each line segment S_m, $m \in \{1,\dots,M-1\}$, for each Jordan block Λ_i, $i \in \{1,\dots,L\}$, to their corresponding vertices $j \in \{1,\dots,M\}$, according to*

  $$\delta^A_{j,i} = \begin{cases} \tilde{\delta}^A_{j,i} & i \in \{1,N\}, \\ \max\{\tilde{\delta}^A_{j,i-1}, \tilde{\delta}^A_{j,i}\} & i \in \{2,\dots,N-1\}, \end{cases}$$

  $$\delta^E_{j,i} = \begin{cases} \tilde{\delta}^E_{j,i} & i \in \{1,N\}, \\ \max\{\tilde{\delta}^E_{j,i-1}, \tilde{\delta}^E_{j,i}\} & i \in \{2,\dots,N-1\}. \end{cases}$$

- *Finally, with B and C given in* (178)*, define*

  $$H_\sigma := \begin{bmatrix} T^{-1} & 0 & 0 & 0 \\ T^{-1}BK_l & T^{-1}BK_l & 0 & T^{-1}B(I-\Gamma^u_l) \end{bmatrix}$$

  and

  $$G_j := \begin{bmatrix} T & T \\ T & T \\ -CT & -CT \\ 0 & 0 \end{bmatrix} \cdot U_j$$

in which

$$U_j = \mathrm{Diag}(\delta^A_{1,j} I_1, \dots, \delta^A_{L,j} I_L, \delta^E_{1,j} I_1, \dots, \delta^E_{L,j} I_L)$$

with I_i is the identity matrix with size of the i-th real Jordan Block. The additive uncertainty set $\boldsymbol{\Delta} \subseteq \mathbb{R}^{2n_x \times 2n_x}$ is now given by

$$\begin{aligned}\boldsymbol{\Delta} = \big\{ \mathrm{Diag}(\Delta^1, \dots, \Delta^{2L}) \mid \Delta^{i+jL} \in \mathbb{R}^{n_i \times n_i}, \\ \|\Delta^{i+jL}\| \leq 1, i \in \{1, \dots, L\}, j \in \{1, 2\} \big\}.\end{aligned}$$

Theorem 11. *Consider system* (186), (187)*, where $h_k \in [\underline{h}, \overline{h}], k \in \mathbb{N}$. If system* (188) *is obtained by following Procedure 1,* (188) *is an overapproximation of* (186) *in the sense that* (189) *holds.*

Proof. The proof can be obtained along the lines of the proof of Theorem 1 of [17] and is omitted for the sake of brevity. □

Using this overapproximation, stability of system (186), (187) can be analyzed using the following theorem from [17], in which

$$\begin{aligned}\mathcal{R} := \{ \mathrm{Diag}(r_1 I_1, \dots, r_L I_L, r_{L+1} I_1, \dots, r_{2L} I_L) \\ \in \mathbb{R}^{2n_x \times 2n_x} \mid r_i > 0 \}\end{aligned}$$

with I_i the identity matrix of size n_i, complying with the i-th real Jordan Block.

Theorem 12. *Consider the closed-loop NCS* (186), (187)*, dictated by a periodic protocol with period $\tilde{N}$, and an overapproximation constructed using Procedure 1. Assume that there exist positive definite matrices P_ℓ, $\ell \in \{1, \dots, \tilde{N}\}$, and matrices $R_{\ell,\sigma_j} \in \mathcal{R}$, $\ell \in \{1, \dots, \tilde{N}\}$ and $j \in \{1, \dots, M\}$, satisfying the LMIs*

$$\begin{bmatrix} F^\top_{\sigma_\ell, j} P_{\ell+1} F_{\sigma_\ell, j} - P_\ell + H^\top_{\sigma_\ell} R_{\ell,\sigma_j} H_{\sigma_\ell} & F^\top_{\sigma_\ell, j} P_{\ell+1} G_j \\ G_j^\top P_{\ell+1} F_{\sigma_\ell, j} & G_j^\top P_{\ell+1} G_j - R_{\ell,\sigma_j} \end{bmatrix} \prec 0, \tag{192}$$

where $P_{\tilde{N}+1} := P_1$, for all $\ell \in \{1, \dots, \tilde{N}\}$ and $j \in \{1, \dots, M\}$. Then, the system (188) *is GAS and consequently, the system* (186), (187) *is globally asymptotically stable (GAS).*

Proof. The proof is given in [17]. □

Remark 6. *Using a reasoning similar as in [50], it can be shown that GAS of the discrete-time model also implies stability of the sampled-data NCS including intersample behavior.*

Remark 7. *NCS with other protocols (e.g. TOD) can be analyzed in a similar manner using the ideas in [17].*

$$\tilde{A}_{\sigma_k} = \left[\begin{array}{c|cc|c} A_d - L_{\sigma_k}\Gamma^y_{\sigma_k} C_d & 0 & 0 & 0 \\ \hline -B_d K_{\sigma_k} & A_d - B_d K_{\sigma_k} & B_d(I - \Gamma^u_{\sigma_k}) & 0 \\ K_{\sigma_k}(A_d - L_{\sigma_k}\Gamma^y_{\sigma_k} C_d - B_d K_{\sigma_k} - I) & K_{\sigma_k}(A_d - B_d K_{\sigma_k} - I) & (K_{\sigma_k} B_d + I)(I - \Gamma^u_{\sigma_k}) & 0 \\ \hline C_d B_d K_{\sigma_k} & C_d(I - A_d + B_d K_{\sigma_k}) & -C_d B_d (I - \Gamma^u_{\sigma_k}) & I - \Gamma^y_{\sigma_k} \end{array}\right] \tag{194a}$$

$$\tilde{A}_{\sigma_k} = \left[\begin{array}{c|c|c} A_d - L_{\sigma_k}\Gamma^y_{\sigma_k} C_d & 0 & 0 \\ \hline -B_d K_{\sigma_k} & A_d - B_d K_{\sigma_k} & 0 \\ \hline C_d B_d K_{\sigma_k} & C_d(I - A_d + B_d K_{\sigma_k}) & I - \Gamma^y_{\sigma_k} \end{array}\right] \tag{194b}$$

4.1.4 Design for Periodic Protocols

In the previous sections, we derived a model describing an LTI plant interconnected with a decentralized switched observer-based controller by a communication network and presented a procedure to assess stability of the model for given K_l and L_l, $l \in \{1, 2, ..., n_T\}$. In this section, we will present a procedure for obtaining the controller and observer gains K_l and L_l, respectively, in (185) for periodic protocols.

As the decentralized and networked constrained control problem is known to be non-convex and hard to solve in general, for the design of K_l and L_l we ignore two aspects of the problem; namely the clock skew effects and the coupling terms between the subsystems. Due to the available robust stability test (see Section 4.1.3), we can verify stability including these ignored effects *a posteriori*. In other words, first we design a switched observer-based controller for the system with constant sampling interval $h_\star$ and without subsystem coupling and, second, perform a robust stability analysis including varying sampling-time effects and subsystem coupling terms using Theorem 12. In case the *a posteriori* test fails, a modification is made to the design problem and solved again.

Alternatively, the robust stability analysis can be split into two steps. The first step is to include only the coupling terms and assess stability. Since the resulting is a periodic switched linear system, stability can be assessed using a standard eigenvalue test [39]. This will provide the designer with a check whether to re-design. Depending on the design freedom available, a re-design may consist of choosing a coarser subsystem decomposition or modifying the periodic protocol. If the coupled system is stable, then we are guaranteed to have some margin of robustness against time-varying sampling intervals, i.e. the NCS will be stable for $h_k \in [\underline{h}, \bar{h}], k \in (N)$ for some $\underline{h} < h_\star < \bar{h}$. Therefore the second step is to add the clock skew terms and use the stability analysis of Section 4.1.3 to determine the values of $\underline{h}$ and $\bar{h}$ that guarantee stability and verify if the range $[\underline{h}, \bar{h}]$ is sufficiently large.

Returning to the design, if we ignore clock skew effects and subsystem coupling terms, the system (186) changes into

$$\bar{x}_{k+1} = \tilde{A}_{\sigma_k} \bar{x}_k \tag{193}$$

with $\tilde{A}_{\sigma_k}$ as in (194a). This simplifies the design problem to stabilizing a cascade of three smaller systems (recognize the block triangular structure in $\tilde{A}_{\sigma_k}$ in (194a)). We can further reduce design complexity by assuming that the controller can access all actuators at every transmission time ($\Gamma^u_{\sigma_k} = I$). This assumption yields the system (193) with $\tilde{A}_{\sigma_k}$ as in (194b). In the following, we will first design for the case when $\Gamma^u_{\sigma_k} = I$ and then briefly provide insight into the design for general $\Gamma^u_{\sigma_k}$ matrices.

For the special case, $\Gamma^u_{\sigma_k} = I$ (actuators always accessible), one can modify the model (193) with (194a) by removing the third column and row (as $e^u_k = 0, k \in \mathbb{N}$) and design for the model (193) with $\tilde{A}_{\sigma_k}$ given by (194b). This case is certainly of practical interest since it is a common industrial configuration to hardwire actuators directly to a controller while measurement data is received through (wireless) sensor networks. The following theorem formalizes the LMI-based design of a switched observer-based controller under periodic protocols with $\Gamma^u_{\sigma_k} = I$ for all k.

Theorem 13. *Consider the system* (193) *with* $\tilde{A}_{\sigma_k}$ *as in* (194b)*. Moreover, consider the protocol to be periodic, such that* $\sigma_{k+\tilde{N}} = \sigma_k$ *holds for all* $k \in \mathbb{N}$ *with* $\tilde{N} \geq n_T$ *and* $\{\, \sigma_k \mid 1 \leq k \leq \tilde{N} \,\} = \{1, 2, ..., n_T\}$, *i.e. all nodes are addressed in one period of the protocol. Suppose that, for each* i^{th} *subsystem,* $i = 1, ..., N$*, the following conditions are satisfied:*

1. *There exist matrices* $P_{i,l} = P_{i,l}^\top \succ 0 \in \mathbb{R}^{n_i \times n_i}$ *and* $S_{i,l} \in \mathbb{R}^{n_i \times p_i}$ *for* $l = 1, 2, ..., n_T$*, such that for all* $\ell = 1, 2, ..., \tilde{N}$
$$\begin{bmatrix} P_{i,\sigma(\ell-1)} & \bar{A}_i^\top P_{i,\sigma_\ell} - \bar{C}_i^\top (\Gamma^y_{i,\sigma_\ell})^\top S_{i,\sigma_\ell}^\top \\ \star & P_{i,\sigma_\ell} \end{bmatrix} \succ 0; \tag{195}$$

2. *There exist matrices* $Q_i = Q_i^\top \succ 0 \in \mathbb{R}^{n_i \times n_i}$ *and* $Z_i \in \mathbb{R}^{m_i \times n_i}$ *such that*
$$\begin{bmatrix} Q_i & Q_i A_i^\top - Z_i^\top \bar{B}_i^\top \\ \star & Q_i \end{bmatrix} \succ 0. \tag{196}$$

Then the controller gains $K_l = K =$ diag$(Z_1 Q_1^{-1}, ..., Z_N Q_N^{-1})$ and observer gains $L_l =$ diag$(L_{1,l}, ..., L_{N,l})$ with $L_{i,l} = P_{i,l}^{-1} S_{i,l}$ will render the system $\bar{x}_{k+1} = \tilde{A}_{\sigma_k} \bar{x}_k$, with $\tilde{A}_{\sigma_k}$ as in (194b) *GAS.*

Note that Theorem 13 provided convex LMI conditions (195), (196) to design switched observer-based controllers for the special case that all actuators can be updated at each sampling interval. The conditions in Theorem 13 are independent LMIs to solve for K_l and L_l, $l \in \{1, 2, ..., n_T\}$ separately, such that the independent periodic systems corresponding to the diagonal blocks of (194b) are stable. In the general case, i.e. (193) with $\tilde{A}_{\sigma_k}$ given by (194a), the convexity of the design problem is lost. Indeed, K_{σ_k} appears in a quadratic form in the second diagonal block of (194a). However, considering a time-dependent quadratic Lyapunov function candidate and using currently available software, PENBMI [41], it is possible to solve for K_l directly (using the second diagonal block of (194a)) as a polynomial matrix inequality [25].

4.1.5 Example

In this section, we will illustrate the design procedure by using a numerical example. Let us consider the unstable continuous-time plant given by (178) with

$$\left[\begin{array}{c|c} A & B \\ \hline C & \end{array}\right] = \left[\begin{array}{cc:cc|cc:c} 0.6 & -4.2 & 0.1 & 2.1 & 0.7 & 1.9 & -0.02 \\ 0.1 & -2.1 & 0.01 & 0 & 0 & 1 & -0.01 \\ \hdashline 0 & 0 & -3.2 & 0.2 & 0 & 0 & 0.8 \\ 0 & -0.03 & 5.3 & -0.2 & 0 & 0 & -0.4 \\ \hline 1 & 4 & 0 & 0.05 & & & \\ 0.2 & 1 & 0 & 0 & & & \\ \hdashline 0 & 0 & 2 & 0 & & & \end{array}\right], \tag{197}$$

where the decomposition into two subsystems ($N = 2$) is shown using dashed lines and the nominal sampling interval is chosen as $h_\star = 1$ second. The periodic protocol, with $\tilde{N} = 3$, is given by $\sigma_1 = 1, \sigma_2 = 2, \sigma_3 = 3$ and

$$\begin{array}{cccc} \Gamma_1^y = & \Gamma_2^y = & \Gamma_3^y = & \Gamma_l^u = \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \tag{198}$$

for all $l \in \{1, 2, 3\}$. This specific protocol indicates that the controllers have access to the actuators at each transmission time, but the sensor data available to the controller is constrained. Solving the LMI's (195), the following innovation (output injection) matrices were found

$$\begin{array}{ccc} L_1 = & L_2 = & L_3 = \\ \begin{bmatrix} 6.24 & -24.89 & 0 \\ -0.73 & 3.46 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & \begin{bmatrix} 0.32 & 2.65 & 0 \\ 0.16 & 0.15 & 0 \\ 0 & 0 & 0.28 \\ 0 & 0 & 3.27 \end{bmatrix}, & \begin{bmatrix} 0.57 & 0.44 & 0 \\ 0.04 & 0.02 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \end{array} \tag{199}$$

Solving the LMIs (196), the following state feedback matrix were found

$$K_l = K = \begin{bmatrix} 1.94 & -1.40 & 0 & 0 \\ -0.56 & -0.86 & 0 & 0 \\ 0 & 0 & 1.36 & 0.81 \end{bmatrix} \tag{200}$$

for all $l \in \{1, 2, 3\}$. The matrices shown in (199) and (200) will stabilize the decoupled version of (197) (the off-diagonal blocks of $\bar{A}, \bar{B}, \bar{C}$ equal to zero) under the protocol given in (198) and for the nominal sampling interval $h_\star$. Including the off-diagonal blocks into the closed-loop model shows a degradation in performance but preservation of stability. Finally, using Procedure 1 and Theorem 12 with sampling intervals $\tilde{h}_l = \{0.9, 0.96, 1, 1.04, 1.1\}$ as grid points, it was determined that this control system can withstand all possible sampling interval variations in the interval $h_k \in [0.9, 1.1]$ for all k.

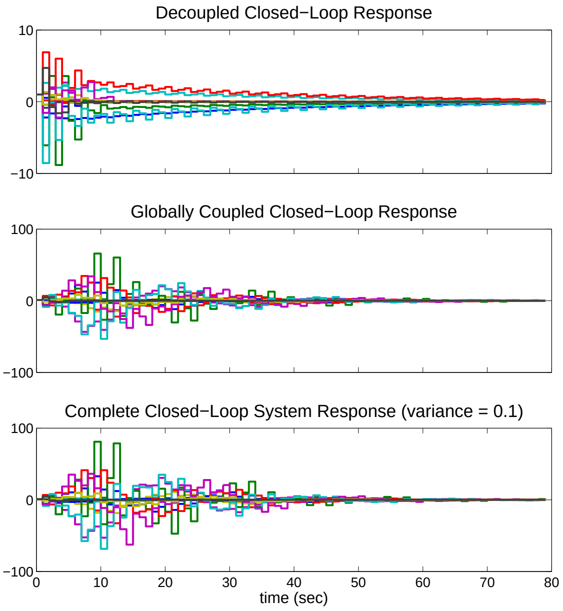


Figure 25: Design Example: Closed-loop state evolution to non-zero initial conditions for the system without coupling and nominal sampling interval $h_\star = 1$ after (top) implementing Theorem 13, (middle) including global subsystem coupling and (bottom) simulating uniformly distributed random sampling intervals $h_k \in [0.9, 1.1]$.

Fig. 25 illustrates this design procedure by plotting the closed-loop state evolution for $\bar{x}_0 = [1, ..., 1]^\top$ after each step of the procedure.
This example shows that the derived theory can be used to design stabilizing output-based decentralized controllers in the presence of communication constraints and network imperfections.

4.1.6 Conclusions and Future Work

In this paper, we developed a model for analyzing decentralized observer-based controllers in the presence of network-induced communication constraints and time-varying sampling intervals. We provided LMI-based stability conditions for verifying stability of the closed-loop NCS. A procedure was presented to design the decentralized observer-based controllers which guarantee stability in the face of communication constraints on the measurement data, but for constant sampling intervals and a decoupled plant. In the case all control inputs are transmitted at each sampling instant, LMI-based design conditions were obtained, otherwise PMI conditions can be solved. Robust stability of the designed controller was verified *a posteriori* by first assessing stability when including coupling in the plant and then testing for the range of time-varying sampling intervals that the closed loop can withstand.

The derived results show the overall structure and complexity of observer-based control design over shared networks. This confirms earlier observations that indicated the complexity of decentralized control design (even without the presence of communication constraints and variations of sampling intervals and availability of full state information). Interestingly, by ignoring varying sampling intervals and global coupling terms, the closed-loop system matrix reveals a lower block triangular structure that can be exploited to obtain simpler LMI conditions for controller/observer synthesis and smaller polynomial matrix inequalities to perform the overall design. In the particular (but industrially relevant) case that all control inputs are communicated at each sampling interval, the design reduces to an LMI.

The presented results can serve as a platform for future developments towards more efficient design conditions for the general case under periodic and other protocols. Another topic for future work is to extend the design to include a distributed control structure, where the controllers can communicate their state information over the network.

4.2 Event-based Estimation and Robust MPC

J. Sijs, M. Lazar, W.M.P.H. Heemels

4.2.1 Introduction

Event-based control has emerged recently as a viable alternative to classical, periodic control, with many relevant applications in networked control systems (NCS). A recent overview of the main pros and cons of event-based control can be found in [7]. The main motivations for event-based control are the limitations imposed by NCS, such as limited bandwidth and computational power, which led to the objective of reducing data transfer or energy consumption. Basically, it was proposed that measurements should be sent to the controller only when an event occurs, of which "Send-on-Delta" (or Lebesque sampling) [8, 44] and "Integral sampling" [45] are some examples. Subsequent studies on control that are based on the event sampling method "Send-on-Delta" were presented in [11,16,35,63,71,72]. The conclusion that can be drawn from these works is that when measurements are sent only at event instants, i.e. dictated by NCS requirements such as minimizing data transfer, it is difficult to guarantee (practical) stability of the closed-loop system.

The natural solution that emerged for solving this problem was to include the controller in the event-triggering decision process. Various alternatives are presented in [11, 16, 38, 42, 64, 75] and the references therein. The generic procedure within this framework is to define a specific criterion for triggering events as a function of the state vector. This function can either be related to guaranteeing closed-loop robust stability, see, e.g., [75], or to improving disturbance rejection, see, e.g., [38]. One of the concerns regarding this framework is that data transfer or energy consumption might be compromised. Another relevant aspect is the fact that controllers are designed for a specific type of event sampling method or, the sampling method is designed specifically for the controller. This implies that both functionalities, i.e. event sampling and control, of the process depend heavily on each other and changing one requires a re-design of the other to guarantee the same properties for the closed-loop system.

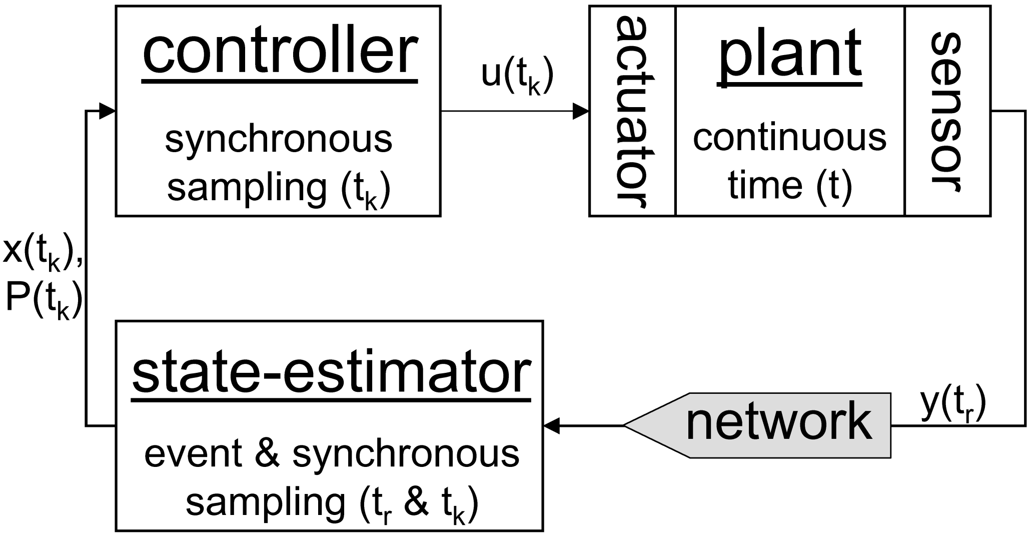


Figure 26: Schematic representation of the feedback loop.

In this paper we investigate the possibility of designing an event-based control system where closed-loop robust stability is decoupled from event generation. To that extent, an event-based state estimator (EBSE) is introduced in the feedback loop, as depicted in Figure 26. The purpose of the EBSE is to deliver a state estimation to the controller synchronously in time, while it receives measurements only at events. Such an EBSE with a synchronous update was recently developed in [60] for autonomous systems. The first contribution of this work is to extend the estimation algorithm of [60] to systems with control inputs. It is shown that under certain assumptions, the EBSE has a bounded covariance matrix. This is possible because the state is updated both when an event occurs, at which a measurement sample is received, as well as at sampling instants synchronous in time, without receiving a measurement sample. In the latter case the update is based on the knowledge that the monitored variable, i.e., the measurement, is within a bounded set that is used to define the event.

The controller that uses the state estimate is based on a new robust MPC algorithm, which forms the second contribution of this paper. This MPC scheme achieves input-to-state stability with respect to the estimation error. Moreover, the MPC algorithm offers the possibility to optimize on-line the closed-loop trajectory-dependent ISS gain, which enhances disturbance rejection. The controller is chosen to run synchronously in time. Therefore, this setup provides most benefits in situations where the sensors are connected to the controller via a (wireless) network link but the controller itself is wired to the actuator/plant. Such a setup is often seen in applications where there are more limitations regarding sensing as to actuation. To integrate the EBSE and MPC in a feedback loop, we develop an efficient method for translating, at each synchronous time instant, the bounds on the covariance matrix of the EBSE into a polytope where the estimation error lies. The latter bound is then fed to the MPC algorithm that uses it to optimize the closed-loop ISS gain. Obviously, if the EBSE receives more real measurements, the resulting bounds on the estimation error will be smaller, which will ultimately improve the trade-off between event generation and closed-loop performance.

The remainder of the paper is structured as follows. Preliminaries are presented in Section 4.2.2, while the EBSE is described in Section 4.2.3. Section 4.2.4 presents the MPC algorithm. Section 4.2.5 discusses several issues related to integration of the EBSE and the robust MPC controller in a feedback loop. An example illustrates the effectiveness of the proposed event-based control scheme in Section 4.2.6. Conclusions are summarized in Section 4.2.7.

4.2.2 Preliminaries

Let $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{Z}$ and $\mathbb{Z}_+$ denote the field of real numbers, the set of non-negative reals, the set of integers and the set of non-negative integers, respectively. For any $\mathcal{C} \subset \mathbb{R}$, let $\mathbb{Z}_{\mathcal{C}} := \{c \in \mathbb{Z} | c \in \mathcal{C}\}$. For a set $\mathcal{S} \subseteq \mathbb{R}^n$, we denote by $\partial\mathcal{S}$ the boundary, by $\mathrm{int}(\mathcal{S})$ the interior and by $\mathrm{cl}(\mathcal{S})$ the closure of $\mathcal{S}$. For two arbitrary sets $\mathcal{S} \subseteq \mathbb{R}^n$ and $\mathcal{P} \subseteq \mathbb{R}^n$, let $\mathcal{S} \oplus \mathcal{P} := \{x + y \mid x \in \mathcal{S}, y \in \mathcal{P}\}$ denote their Minkowski sum. A polyhedron (or a polyhedral set) in $\mathbb{R}^n$ is a set obtained as the intersection of a finite number of open and/or closed half-spaces. Given $(n+1)$ *affinely independent* points $(\theta_0, \ldots, \theta_n)$ of $\mathbb{R}^n$, i.e. $(1\ \theta_0^\top)^\top, \ldots, (1\ \theta_n^\top)^\top$ are linearly independent in $\mathbb{R}^{n+1}$, we define the corresponding simplex S as

$$S := \mathrm{Co}(\theta_0, \ldots, \theta_n) := \left\{ x \in \mathbb{R}^n \middle| x = \sum_{l=0}^{n} \mu_l \theta_l,\ \sum_{l=0}^{n} \mu_l = 1 \right.$$
$$\left. \mu_l \in \mathbb{R}_+ \ \text{ for } l \in \mathbb{Z}_{[0,n]} \right\},$$

where $\mathrm{Co}(\cdot)$ denotes the convex hull.

The notation $\underline{0}$ is used to denote either the null-vector or the null-matrix. Its size will be clear from the context. The transpose, inverse, determinant and trace of a matrix $A \in \mathbb{R}^{n \times n}$ are denoted as $A^\top$, A^{-1}, $|A|$ and tr(A), respectively. The i^{th}, minimum and maximum eigenvalue of a square matrix A are denoted as $\lambda_i(A)$, $\lambda_{\text{min}}(A)$ and $\lambda_{\text{max}}(A)$, respectively. The Hölder p-norm of a vector $x \in \mathbb{R}^n$ is defined as $\|x\|_p := (|[x]_1|^p + \ldots + |[x]_n|^p)^{\frac{1}{p}}$ for $p \in \mathbb{Z}_{[1,\infty)}$ and $\|x\|_\infty := \max_{i=1,\ldots,n} |[x]_i|$, where $[x]_i$, $i \in \mathbb{Z}_{[1,n]}$, is the *i*-th element of x. For brevity, let $\|\cdot\|$ denote an arbitrary p-norm. For a matrix $Z \in \mathbb{R}^{m \times n}$ let $\|Z\| := \sup_{x \neq 0} \frac{\|Zx\|}{\|x\|}$ denote its corresponding induced matrix norm. It is well known that $\|Z\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^{n} |Z^{\{ij\}}|$, where $Z^{\{ij\}}$ is the *ij*-th entry of Z. Let $\mathbf{z} := \{z(l)\}_{l \in \mathbb{Z}_+}$ with $z(l) \in \mathbb{R}^o$ for all $l \in \mathbb{Z}_+$ denote an arbitrary sequence. Define $\|\mathbf{z}\| := \sup\{\|z(l)\| \mid l \in \mathbb{Z}_+\}$ and $\mathbf{z}_{[k]} := \{z(l)\}_{l \in \mathbb{Z}_{[0,k]}}$.

The Gaussian function (shortly noted as Gaussian) is defined as $G : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{n \times n} \ \to \mathbb{R}_+$,

$$G(x, \mu, P) = \frac{1}{\sqrt{(2\pi)^n |P|}} e^{(x-\mu)^T P^{-1} (x-\mu)}. \tag{201}$$

By definition it follows that if $x \in \mathbb{R}^n$ is a random variable with a probability density function (PDF) $p(x) = G(x, \mu, P)$, then the expectation and covariance of x are given by $E[x] = \mu$ and $cov(x) = P$, respectively.

For a bounded Borel set [2] $Y \subset \mathbb{R}^n$, the set PDF is defined as $\Lambda_Y : \mathbb{R}^n \to \{0, \nu\}$, with $\nu \in \mathbb{R}$ the Lebesque measure [37] of Y, i.e.,

$$\Lambda_Y(x) = \begin{cases} 0 & \text{if} \quad x \notin Y, \\ \nu^{-1} & \text{if} \quad x \in Y. \end{cases} \tag{202}$$

A function $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ *belongs to class* $\mathcal{K}$ if it is continuous, strictly increasing and $\varphi(0) = 0$. A function $\beta : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ *belongs to class* $\mathcal{KL}$ if for each fixed $k \in \mathbb{R}_+$, $\beta(\cdot, k) \in \mathcal{K}$ and for each fixed $s \in \mathbb{R}_+$, $\beta(s, \cdot)$ is decreasing and $\lim_{k\to\infty} \beta(s, k) = 0$.

Considered the following discrete time system,

$$x(t_{k+1}) \in \Phi(x(t_k), w(t_k)), \quad t_k = k\tau_s, \tag{203}$$

where $x(t_k)$ is the state and $w(t_k) \in \mathbb{R}^n$ is the unknown disturbance at time instant $t_k = k\tau_s$, $k \in \mathbb{Z}_+$ and for some $\tau_s \in \mathbb{R}_+$. The mapping $\Phi : \mathbb{R}^n \times \mathbb{R}^n \hookrightarrow \mathbb{R}^n$ is an arbitrary compact and non-empty set-valued function. For zero input in (203) we assume that $\Phi(0, 0) = \{0\}$. Suppose $w(t_k)$ takes a value in a bounded set $\mathbb{W} \subset \mathbb{R}^n$ for all $t_k \in \mathbb{R}_+$.

Definition 3. *We call a set $\mathcal{P} \subseteq \mathbb{R}^n$* robustly positively invariant (RPI) *for system* (203) *with respect to $\mathbb{W}$ if for all $x \in \mathcal{P}$ it holds that $\Phi(x, w) \subseteq \mathcal{P}$ for all $w \in \mathbb{W}$.*

Definition 4. *Let $\mathbb{X}$ with $0 \in \operatorname{int}(\mathbb{X})$ and $\mathbb{W}$ be subsets of $\mathbb{R}^n$. We call system* (203) ISS in $\mathbb{X}$ for inputs in $\mathbb{W}$ *if there exist a $\mathcal{KL}$-function $\beta(\cdot,\cdot)$ and a $\mathcal{K}$-function $\gamma(\cdot)$ such that, for each $x(t_0) \in \mathbb{X}$ and all $\mathbf{w} = \{w(t_l)\}_{l\in\mathbb{Z}_+}$ with $w(t_l) \in \mathbb{W}$ for all $l \in \mathbb{Z}_+$, it holds that all corresponding state trajectories of* (203) *satisfy the following inequality: $\|x(t_k)\| \leq \beta(\|x(t_0)\|, k) + \gamma(\|\mathbf{w}_{[t_{k-1}]}\|)$, $\forall k \in \mathbb{Z}_{\geq 1}$.*

We call $\gamma(\cdot)$ an ISS gain of system (203).

Theorem 14. *Let $\mathbb{W}$ be a subset of $\mathbb{R}^n$ and let $\mathbb{X}$ be a RPI set for* (203) *with respect to $\mathbb{W}$, with $0 \in \operatorname{int}(\mathbb{X})$. Furthermore, let $\alpha_1(s) := as^\delta$, $\alpha_2(s) := bs^\delta$, $\alpha_3(s) := cs^\delta$ for some $a, b, c, \delta \in \mathbb{R}_{>0}$, $\sigma \in \mathcal{K}$ and let $V : \mathbb{R}^n \to \mathbb{R}_+$ be a function such that:*

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|), \tag{204a}$$

$$V(x^+) - V(x) \leq -\alpha_3(\|x\|) + \sigma(\|w\|) \tag{204b}$$

for all $x \in \mathbb{X}$, $w \in \mathbb{W}$ and all $x^+ \in \Phi(x, w)$. Then the system (203) *is ISS in $\mathbb{X}$ for inputs in $\mathbb{W}$ with*

$$\begin{aligned} &\beta(s, k) := \alpha_1^{-1}(2\rho^k \alpha_2(s)), \quad \gamma(s) := \alpha_1^{-1}\left(\frac{2\sigma(s)}{1-\rho}\right), \\ &\rho := 1 - \frac{c}{b} \in [0, 1). \end{aligned} \tag{205}$$

The proof of Theorem 14 can be found in [34]. We call a function $V(\cdot)$ that satisfies the hypothesis of Theorem 14 an *ISS Lyapunov function*.

4.2.3 event-based state-estimation

In this section we will present the extension of the EBSE, as recently developed in [60], to systems with control inputs. Therefore, let us assume that a dynamical system with state vector $x \in \mathbb{R}^n$, control input $u \in \mathbb{R}^m$, process noise $q \in \mathbb{R}^n$, measurement vector $y \in \mathbb{R}^l$ and measurement noise $v \in \mathbb{R}^l$ is given. This process is described by a generic discrete-time state-space model, i.e.,

$$x(t) = A_\tau x(t-\tau) + B_\tau u(t-\tau) + q(t, \tau), \tag{206a}$$

$$y(t) = Cx(t) + Du(t) + v(t). \tag{206b}$$

with $A_\tau \in \mathbb{R}^{n\times n}$ and $B_\tau \in \mathbb{R}^{n\times m}$, for all $\tau \in \mathbb{R}_+$, $C \in \mathbb{R}^{l\times n}$ and $D \in \mathbb{R}^{l\times m}$. It is assumed that $u(s)$ remains constant for all $t - \tau \leq s < t$. Basically, the above system description (206a) could be perceived as a discretized version of a continuous-time plant $\dot{x}(t) = Ax(t) + Bu(t)$. In this case the matrices A_τ and B_τ would then be defined with the time difference τ of two sequential sample instants, i.e.,

$$A_\tau := e^{A\tau} \quad \text{and} \quad B_\tau := \int_0^\tau e^{A\eta} d\eta B.$$

However, we allow for the more general description (206). We assume that the process- as well as the measurement-noise have Gaussian PDFs with zero mean, for some $Q_\tau \in \mathbb{R}^{n\times n}$, $\tau \in \mathbb{R}_+$ and $R_v \in \mathbb{R}^{l\times l}$, i.e.,

$$p(q(t,\tau)) := G(q(t,\tau), \underline{0}, Q_\tau) \quad \text{and} \quad p(v(t)) := G(v(t), \underline{0}, R_v).$$

The sensor uses an event sampling method which is based on y. Its sample instants are indexed by r, i.e. $y(t_r)$ denotes a measurement taken at the event instant t_r. As proposed in [60], $H_r \subset \mathbb{R}^{l+1}$ is a set, determined at the event instant t_{r-1}, in the *time-measurement*-space that induces the event instants. An example of this set, in case the measurement-space is 2D, is graphically depicted in Figure 27. To be precise, given that t_{r-1} was the latest event instant, the next event instant t_r is defined as:

$$t_r := \inf \left\{ t \in \mathbb{R}_+ \mid t > t_{r-1} \text{ and } \begin{pmatrix} y(t) \\ t \end{pmatrix} \notin H_r \right\}. \tag{207}$$

To prevent that more than one sample action occurs at t_{r-1}, it should hold that $(y^\top(t_{r-1}), t_{r-1})^\top \in$ int(H_r).

To illustrate the event triggering, let us present two examples of how to choose the set H_r. In the first example the events are triggered by applying the sampling method "Send-on-Delta" [8, 44]. A new measurement sample $y(t_r)$ is generated when $|y(t) - y(t_{r-1})| > \Delta$. Notice that this is equivalent with (207) in case

$$H_r := \{(y^\top, t)^\top \big| \, |y - y(t_{r-1})| \leq \Delta\}.$$

The second example of a method for triggering the events is taken from [9, 38]. Therein, a sampling method is described which is similar to "Send-on-Delta", although $y(t_{r-1})$ is replaced with the current predicted measurement $C\hat{x}(t)$. Notice that in this case $H_r := \{(y^\top, t)^\top \big| \, |y - C\hat{x}(t)| \leq \Delta\}$.

As the sensor uses an event-sampling method on $y(t)$, the EBSE receives $y(t_r)$ to perform a state-update. However, typically the event instants t_r do not occur at the same time as the synchronous instants at which the controller needs to calculate a new control-input. Hence, the EBSE has to keep track of the state at both the event instants as well as the synchronous instants. Let us define $\mathbb{T}_r(t)$ and $\mathbb{T}_c(t)$ as the set of time instants that correspond to all event instants and synchronous instants, respectively. Therefore, if $\tau_s \in \mathbb{R}_+$ denotes the controller's sampling time, we have that

$$\mathbb{T}_r := \{t_r \mid r \in \mathbb{Z}_+\} \quad \text{and} \quad \mathbb{T}_c := \{k\tau_s \mid k \in \mathbb{Z}_+\},$$

where the event instants t_r are generated by (207). Notice that it could happen that an event instant coincides with a synchronous instant. Therefore $\mathbb{T}_r \cap \mathbb{T}_c$ might be non-empty. The EBSE calculates an estimate of the state-vector and an error-covariance matrix at each sample instant $t \in \mathbb{T}$, with $\mathbb{T} := \mathbb{T}_r \cup \mathbb{T}_c$. At an event instant, i.e. $t \in \mathbb{T}_r$, the EBSE receives a new measurement $y(t_r)$ with which a state-update can be performed. At the synchronous instants $t \in \mathbb{T}_c \backslash \mathbb{T}_r$, the EBSE does not receive a measurement. Standard estimators would perform a state-*prediction* using the process-model. However, from (207) we observe that if no measurement y was received at $t > t_{r-1}$, still it is known that $(y(t)^\top, t)^\top \in H_r$. The estimator can exploit this information to perform a state-*update* not only at the event instants but also at the synchronous instants $t \in \mathbb{T}_c \backslash \mathbb{T}_r$. Next, we describe how this is implemented. Let us define $H_{r|t} \subset \mathbb{R}^l$ as a section of H_r at the time instant $t \in (t_{r-1}, t_r)$, which is graphically depicted in Figure 27, and formally defined as:

$$H_{r|t} := \left\{ y \in \mathbb{R}^l \middle| \begin{pmatrix} y \\ t \end{pmatrix} \in H_r \right\}.$$

Therefore, the following two conditions hold for any $t \in \mathbb{T}$:

$$y(t) \in \begin{cases} \{y(t)\} & \text{if} \quad t \in \mathbb{T}_r, \\ H_{r|t} & \text{if} \quad t \in \mathbb{T}_c \backslash \mathbb{T}_r. \end{cases} \tag{208}$$

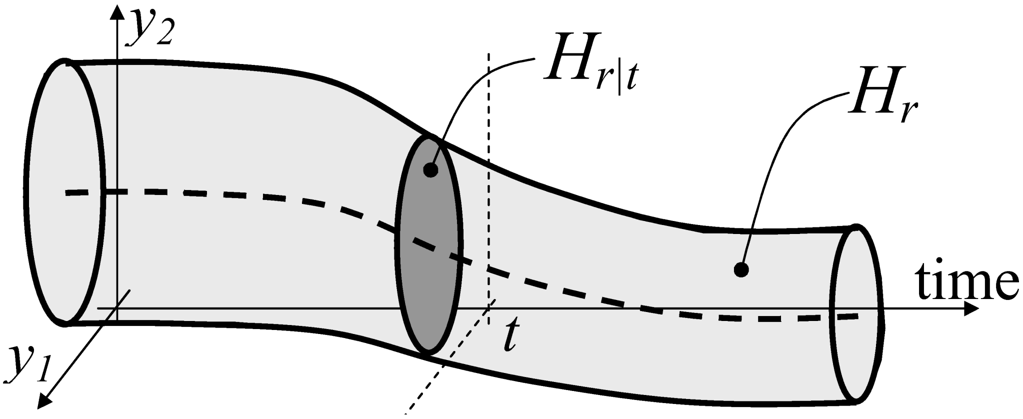


Figure 27: An example of H_r defining a set in the *time-measurement*-space and $H_{r|t}$ defining a section in the *measurement*-space at a certain time-instant $t \in (t_{r-1}, t_r)$.

The estimator must first determine a PDF of the measurement $y(t)$. Therefore if $\delta(\cdot)$ denotes the Dirac-pulse and $\Lambda_Y(\cdot)$ denotes the set PDF as defined in (202), then from equation (208) it follows that:

$$p(y(t)) = \begin{cases} \delta(y(t)) & \text{if} \quad t \in \mathbb{T}_r, \\ \Lambda_{H_{r|t}}(y(t)) & \text{if} \quad t \in \mathbb{T}_c \backslash \mathbb{T}_r. \end{cases} \tag{209}$$

In [62] is was shown that any PDF can be approximated as a sum of Gaussians. Therefore, let us assume that $p(y(t))$ of (209) is approximated by $\sum_{i=1}^{N} \frac{1}{N} G(y(t), \hat{y}_i(t), R_H(t))$, for some $N \in \mathbb{Z}_+$, $R_H(t) \in \mathbb{R}^{l \times l}$ and $\hat{y}_i(t) \in \mathbb{R}^l$ for all $i \in \mathbb{Z}_{[1,N]}$. Notice that in case $t \in \mathbb{T}_r$ it follows that $N = 1$, $\hat{y}_1(t) = y(t)$ and $R_H(t)$ can be taken arbitrary small to approximate the Dirac pulse. Let $\hat{x}(t)$ denote the estimated state-vector of the EBSE and let $P(t)$ denote the error-covariance matrix both at $t \in \mathbb{T}$. Furthermore, let $R(t) := R_v + R_H(t)$. Then, the set of equations of the EBSE, in standard Kalman filter form, yields:

Step 1: prediction

$$\begin{aligned} &\hat{x}^-(t) = A_\tau \hat{x}(t-\tau) + B_\tau u(t-\tau), \\ &P^-(t) = A_\tau P(t-\tau) A_\tau^\top + Q_\tau, \end{aligned} \tag{210a}$$

Step 2: measurement-update, $\forall i \in \mathbb{Z}_{[1,N]}$

$$\begin{aligned} &K(t) = P^-(t) C^\top (C P^-(t) C^\top + R(t))^{-1} C, \\ &P_i(t) = (I - K(t)C) P^-(t), \\ &\hat{x}_i(t) = \hat{x}^-(t) + K(t)(\hat{y}_i(t) - Du(t) - C\hat{x}^-(t)), \\ &\beta_i(t) = G(y_i(t), C\hat{x}^-(t) + Du(t), CP^-(t)C^\top + R(t)), \end{aligned} \tag{210b}$$

Step 3: state-approximation

$$\begin{aligned} &\hat{x}(t) = \sum_{i=1}^{N} \frac{\beta_i(t)}{\sum_{i=1}^{N} \beta_i(t)} \hat{x}_i(t), \\ &P(t) = \sum_{i=1}^{N} \frac{\beta_i(t)}{\sum_{i=1}^{N} \beta_i(t)} \left(P_i(t) + (\hat{x}(t) - \hat{x}_i(t)) (\hat{x}(t) - \hat{x}_i(t))^\top \right). \end{aligned} \tag{210c}$$

The main reason for the approximation of (210c) is to limit the amount of processing demand of the EBSE. Next, we present a brief account of the numerical complexity of the proposed EBSE in comparison with the original Kalman filter [32], i.e.,

- Prediction: $O((m+1)n) + O(2n^2) + O(2n^3)$;
- Update: $O(l^3) + O(l^2(1+2n)) + O(n^2(2l+1)) + O(2n^3) + O(nm + l(n+1)) + O(N(2l^2 + 3l + nl + n))$;
- Approximation: $O(n + (2+3n)N) + O((4N+1)n^2)$.

For clarity, let us neglect terms that are of second order or less. Then the computational complexity of the EBSE is

$$O(4n^3 + l^3 + 2l^2n + 2ln^2 + N(2l^2 + nl + 4n^2),$$

which is still proportional to the complexity of the original Kalman filter, i.e., $O(4n^3 + l^3 + 2l^2n + 2ln^2)$.

Remark 8. *The set of equations of the EBSE is based on a Sum-of-Gaussians approach. The main reason for choosing the Sum-of-Gaussians approach is that it enables an asymptotic bound on $P(t)$. This property is important as it enables a guarantee, in a probabilistic sense, that the estimation error is bounded.* □

Next, we recall the main result of [59, 60], where it was proven that all the eigenvalues of $P(t)$, i.e. $\lambda_i(P(t))$, are asymptotically bounded. To that extent, let us define $\bar{H} \subset \mathbb{R}^l$ as a bounded set such that $H_{r|t} \subseteq \bar{H}$ for all $t \in \mathbb{T}$. This further implies that each set $H_{r|t}$ is bounded for all $t \in \mathbb{T}$. With the set $\bar{H}$ we can now determine a covariance-matrix $R \in \mathbb{R}^{l \times l}$, such that $R \succeq R(t)$ for all $t \in \mathbb{T}$. Notice that $R = R_v + R_{\bar{H}}$, in which $R_{\bar{H}}$ can be derived from $\Lambda_{\bar{H}}(y)$ by approximating this PDF as a single Gaussian. Similarly, there exist a $\omega \in \mathbb{R}_+$, such that the Euclidean distance between any two elements of $H_{r|t}$ is less than $(\omega+1)\lambda_{\text{min}}^{-1}(R(t))$, for all $t \in \mathbb{T}_c \backslash \mathbb{T}_r$. The next preliminary result [55] states the standard conditions for the existence of a bounded asymptotic covariance matrix $\Sigma_\infty \in \mathbb{R}^{n \times n}$ for a scaled synchronous Kalman filter. The update of the covariance matrix for this type of Kalman filter, denoted with $\Sigma[k] \in \mathbb{R}^{n \times n}$ at the synchronous instant $t = k\tau_s$, yields:

$$\Sigma[k] = \omega \left((A_{\tau_s}\Sigma([k-1]A_{\tau_s} + Q_{\tau_s})^{-1} + C^\top R^{-1} C \right)^{-1}, \quad \forall k \in \mathbb{Z}_+.$$

Proposition 2. [55] *Let Σ_∞ be defined as the solution of $\Sigma_\infty^{-1} = \left(\omega A_{\tau_s}\Sigma_\infty A_{\tau_s}^\top + Q_{\tau_s}\right)^{-1} + C^\top R^{-1} C$. If Σ_∞ exists, (A_{τ_s}, C) is an observable pair and $\lambda_i(\bar{A}) \leq 1$, for all $i \in \mathbb{Z}_{[1,n]}$, where $\bar{A} := \sqrt{\omega}\Big(A_{\tau_s} - A_{\tau_s}\bar{\Sigma}C^\top \left(C\bar{\Sigma}C^\top + R\right)^{-1} C\Big)$ and $\bar{\Sigma} := A_{\tau_s}\Sigma_\infty A_{\tau_s}^\top + Q_{\tau_s}$, then it holds that $\lim_{k\to\infty} \Sigma[k] = \Sigma_\infty$.*

Now we can state the main result on the asymptotic bound of $P(t)$.

Theorem 15. *Let the scalars a_{τ_s} and b_{τ_s} be defined as follows: $a_{\tau_s} := \sup_{\tau\in[0,\tau_s]} \sigma_{max}(A_\tau)$ and $b_{\tau_s} := \sup_{\tau\in[0,\tau_s]} \sigma_{max}(B_\tau)$. If the hypothesis of Proposition 2 holds, then we have that*

$$\lim_{t\to\infty} \lambda_{\mathit{max}}(P(t)) \leq a_{\tau_s}^2 \lambda_{\mathit{max}}(P_\infty) + b_{\tau_s}^2 \lambda_{\mathit{max}}(Q_{\tau_s}).$$

This result guarantees a bound on the covariance of the estimation error at all time instants. The proof of the above theorem is obtained *mutatis mutandis* from the proof given in [59, 60] for the case of an autonomous process. Indeed, the difference in the EBSE algorithm (210) corresponding to the system with a control input versus the autonomous system is given by the term $B_\tau u(t-\tau)$ in (210a). However, as this additional term is present in both $\hat{x}_i(t)$ and $\hat{x}(t)$, it cancels out in $\hat{x}(t) - \hat{x}_i(t)$ and therefore, it is not present in the expression of $P(t)$. As such, the proof given in [59, 60] applies to a system with control input as well.

In Section 4.2.5 we will show how $P(t)$ is used to determine, with a certain probability, a bound on the estimation error at every synchronous instant. Knowledge of this bound is used to design the robust MPC algorithm, as it is explained in the next section.

4.2.4 Robust MPC algorithm

In this section we start from the fact that a state estimate $\hat{x}(t_k)$, provided by the EBSE at each time instant $t_k = k\tau_s$, $k \in \mathbb{Z}_+$, is available for the controller at all synchronous instants. Moreover, we assume that the corresponding estimation error $w(t_k) := \hat{x}(t_k) - x(t_k)$ satisfies $w(t_k) \in \mathbb{W}(t_k)$ at all t_k, where $\mathbb{W}(t_k)$ is a known polytope (closed and bounded polyhedron). An efficient procedure for determining $\mathbb{W}(t_k)$ from $P(t_k)$, $k \in \mathbb{Z}_+$, will be presented in Section 4.2.5.

As the controller samples synchronously in time, the process-model of (206) can be rewritten from an asynchronous model into a synchronous one. Hence, consider the following discrete-time model of the system used for controller synthesis:

$$\begin{aligned} x(t_{k+1}) :&= Ax(t_k) + Bu(\hat{x}(t_k)) \\ &= Ax(t_k) + Bu(x(t_k) + w(t_k)),\ k \in \mathbb{Z}_+, \end{aligned} \tag{211}$$

where $x(t_k) \in \mathbb{X} \subseteq \mathbb{R}^n$ is the real state, $\hat{x}(t_k) \in \mathbb{X} \subseteq \mathbb{R}^n$ is the estimated state, $u(t_k) \in \mathbb{U} \subseteq \mathbb{R}^m$ is the control action and $w(t_k) \in \mathbb{W}(t_k) \subset \mathbb{R}^n$ is an unknown estimation error at the discrete-time instant t_k. The matrices $A = A_{\tau_s}$ and $B = B_{\tau_s}$ correspond to the discretized model of system (206). From the above equations it can be seen that at any synchronous time instant t_k there exists a $w(t_k) \in \mathbb{W}(t_k)$ such that $x(t_k) = \hat{x}(t_k) - w(t_k) \in \{\hat{x}(t_k)\} \oplus \mathbb{W}(t_k)$ and, any $w(t_k)$ can be obtained as a convex combination of the vertices of $\mathbb{W}(t_k)$. We assume that $0 \in \mathrm{int}(\mathbb{X})$ and $0 \in \mathrm{int}(\mathbb{U})$ and indicate that $0 \in \mathrm{int}(\mathbb{W}(t_k))$ for all $k \in \mathbb{Z}_+$, as it will be shown in Section 4.2.5. For simplicity and clarity of exposition, the index t_k is omitted throughout a part of this section, i.e. $\hat{x}$, x, $\mathbb{W}$, etc. will denote $\hat{x}(t_k)$, $x(t_k)$, $\mathbb{W}(t_k)$ and so on.

Our goal is now to design a control algorithm that finds a control action $u(\hat{x}) \in \mathbb{U}$ such that for all $w \in \mathbb{W}$

$$V(Ax + Bu(\hat{x})) - V(x) + \alpha_3(\|x\|) - \sigma(\|w\|) \leq 0. \tag{212}$$

Satisfaction of the above inequality for some $\sigma \in \mathcal{K}$ would guarantee ISS of the corresponding closed-loop system, see Theorem 14. Moreover, to improve disturbance rejection, we will adopt the idea of optimizing the closed-loop ISS gain by minimizing the gain of the function σ, which was recently proposed in [36]. Therein, the case of additive disturbances was considered. In what follows we extend the results of [36] to estimation errors, which act as a disturbance on the measurement that is fed to the controller.

The first relevant observation is that an optimization problem based directly on the constraint (212) is not finite dimensional in w. However, we demonstrate that by considering *continuous and convex*[8] Lyapunov functions and bounded polyhedral sets $\mathbb{X}, \mathbb{U}, \mathbb{W}$ (with non-empty interiors containing the origin) a solution to inequality (212) can be obtained via a finite set of inequalities that only depend on the vertices of $\mathbb{W}$.

Let w^e, $e = 1, ..., E$, be the vertices of $\mathbb{W}$ (notice that $E > n$, as $\mathbb{W}$ is assumed to have a non-empty interior). Next, consider a finite set of simplices $S_1, \dots, S_M$ with each simplex S_i equal to the convex hull of a subset of the vertices of $\mathbb{W}$ and the origin, and such that $\cup_{i=1}^{M} S_i = \mathbb{W}$, $\mathrm{int}(S_i) \cap \mathrm{int}(S_j) = \emptyset$ for $i \neq j$, $\mathrm{int}(S_i) \neq \emptyset$ for all i. More precisely, $S_i = \mathrm{Co}\{0, w^{e_{i,1}}, \dots, w^{e_{i,l}}\}$ and

$$\{w^{e_{i,1}}, \dots, w^{e_{i,l}}\} \subseteq \{w^1, \dots, w^E\}$$

(i.e. $\{e_{i,1}, \dots, e_{i,l}\} \subseteq \{1, \dots, E\}$) with $w^{e_{i,1}}, \dots, w^{e_{i,l}}$ linearly independent. For an illustrative example see Figure 28: the polyhedron $\mathbb{W}$ consists of $S_1, S_2, \dots, S_5$, where, for instance, the simplex S_3 is generated by $0, w^{e_{3,1}}, w^{e_{3,2}}$, with $e_{3,1} = 2$ and $e_{3,2} = 3$. For each simplex S_i we define the matrix $W_i := [w^{e_{i,1}} \ \dots \ w^{e_{i,l}}] \in \mathbb{R}^{l \times l}$, which is invertible. Let $\gamma_e \in \mathbb{R}_+$ be variables associated with each vertex w^e, which are both depending on the time instant t_k.

Next, suppose that both x and $\hat{x}$ are known. Notice that the assumption that x is known is only used here to show how one can transform (212) into a finite dimensional problem. The dependence on x will be removed later, leading to a main stability result and an MPC algorithm that only use the estimated state $\hat{x}$, see Problem 1. Let $\alpha_3 \in \mathcal{K}_\infty$ and consider the following set of constraints:

$$V(A\hat{x} + Bu(\hat{x})) - V(x) + \alpha_3(\|x\|) \leq 0, \tag{213a}$$

$$V(A(\hat{x} - w^e) + Bu(\hat{x})) - V(x) + \alpha_3(\|x\|) - \gamma_e \leq 0 \tag{213b}$$

for all $e = 1, \dots, E$.

[8] This includes quadratic functions, $V(x) = x^\top Px$ with $P \succ 0$, and functions based on norms, $V(x) = \|Px\|$ with P a full-column rank matrix.

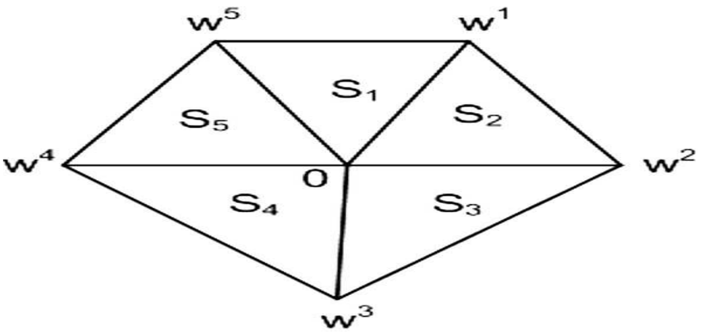


Figure 28: An example of the set $\mathbb{W}$.

Theorem 16. *Let V be a continuous and convex Lyapunov function. If for $\alpha_3 \in \mathcal{K}_\infty$, $\hat{x}$ and x there exist $u(\hat{x})$ and $\{\gamma_e\}_{e=1,\ldots,E}$, such that* (213a) *and* (213b) *hold, then* (212) *holds for the same $u(\hat{x})$, with $\sigma(s) := \eta s$ and*

$$\eta := \max_{i=1,\ldots,M} \|\bar{\gamma}_i W_i^{-1}\|, \tag{214}$$

where $\bar{\gamma}_i := [\gamma_{e_{i,1}} \ \ldots \ \gamma_{e_{i,l}}] \in \mathbb{R}^{1\times l}$ and $\|\cdot\|$ is the corresponding induced matrix norm.

Proof: For any $w \in \mathbb{W} = \bigcup_{i=1}^M S_i$ there exists an i such that $w \in S_i = \mathrm{Co}\{0, w^{e_{i,1}}, \ldots, w^{e_{i,l}}\}$, which means that there exist non-negative $\mu_0, \mu_1, \ldots, \mu_l$ with $\sum_{j=0,1,\ldots,l} \mu_j = 1$ such that

$$w = \sum_{j=1,\ldots,l} \mu_j w^{e_{i,j}} + \mu_0 0 = \sum_{j=1,\ldots,l} \mu_j w^{e_{i,j}}.$$

In matrix notation we have that $w = W_i[\mu_1 \ \ldots \ \mu_l]^\top$ and thus

$$[\mu_1 \ \ldots \ \mu_l]^\top = W_i^{-1} w.$$

By multiplying each inequality in (213b) corresponding to the index $e_{i,j}$ and the inequality (213a) with $\mu_j \geq 0$, $j = 0, 1, \ldots, l$, summing up and using $\sum_{j=0,1,\ldots,l} \mu_j = 1$ yields:

$$\mu_0 V(A\hat{x} + Bu(\hat{x})) + \sum_{j=1,\ldots,l} \mu_j V(A(\hat{x} - w^{e_{i,j}}) + Bu(\hat{x})) \\ - V(x) + \alpha_3(\|x\|) - \sum_{j=1,\ldots,l} \mu_j \gamma_{e_{i,j}} \leq 0.$$

Furthermore, using $\sum_{j=0,1,\ldots,l} \mu_j = 1$ and convexity of V yields

$$V(A(\hat{x} - \sum_{j=1,\ldots,l} \mu_j w^{e_{i,j}}) + Bu(\hat{x})) - V(x) \\ + \alpha_3(\|x\|) - \sum_{j=1,\ldots,l} \mu_j \gamma_{e_{i,j}} \leq 0,$$

or equivalently

$$V(A(\hat{x} - w) + Bu(\hat{x})) - V(x) + \alpha_3(\|x\|) - \bar{\gamma}_i[\mu_1 \ \ldots \ \mu_l]^\top \leq 0.$$

Using that $[\mu_1 \ \ldots \ \mu_l]^\top = W_i^{-1}w$ and $x = \hat{x} - w$ we obtain (212) for $\sigma(s) = \eta s$ and $\eta \geq 0$ as in (214).□

Based on the result of Theorem 16 we are now able to formulate a finite dimensional optimization problem that results in closed-loop ISS with respect to the estimation error $w(t_k)$ and moreover, in optimization of the closed-loop ISS gain. This will be achieved only based on the estimate $\hat{x}(t_k)$ and the set $\mathbb{W}(t_k)$.

Let $\bar{\gamma} := [\gamma_1, \ldots, \gamma_E]^\top$ and let $J : \mathbb{R}^E \to \mathbb{R}_+$ be a function that satisfies $\alpha_4(\|\bar{\gamma}\|) \leq J(\gamma_1, \ldots, \gamma_E) \leq \alpha_5(\|\bar{\gamma}\|)$ for some $\alpha_4, \alpha_5 \in \mathcal{K}_\infty$. Define next:

$$V_{\text{min}}(t_k) := \min_{x \in \{\hat{x}(t_k)\} \oplus \mathbb{W}(t_k)} V(x) \tag{215}$$

and

$$\alpha_{3,\mathsf{max}}(t_k) := \max_{x\in\{\hat{x}(t_k)\}\oplus\mathbb{W}(t_k)} \alpha_3(\|x\|). \tag{216}$$

Problem 1. *Let $\alpha_3 \in \mathcal{K}_\infty$, a cost J and a Lyapunov function V be given. At time $k \in \mathbb{Z}_+$ let an estimate of the state $\hat{x}(t_k)$ be known and minimize $J(\gamma_1(t_k),\ldots,\gamma_E(t_k))$ over $u(t_k), \gamma_1(t_k),\ldots,\gamma_E(t_k)$, subject to the constraints*

$$u(t_k) \in \mathbb{U},\ \gamma_e(t_k) \geq 0 \tag{217a}$$

$$Az + Bu(t_k) \in \mathbb{X}, \forall z \in \{\hat{x}(t_k)\} \oplus \mathbb{W}(t_k) \tag{217b}$$

$$V(A\hat{x}(t_k) + Bu(t_k)) - V_{\mathit{min}}(t_k) + \alpha_{3,\mathit{max}}(t_k) \leq 0, \tag{217c}$$

$$\begin{aligned} V(A(\hat{x}(t_k) - w_e(t_k)) + Bu(t_k)) - V_{\mathit{min}}(t_k) \\ + \alpha_{3,\mathit{max}}(t_k) - \gamma_e(t_k) \leq 0 \end{aligned} \tag{217d}$$

for all $e = 1,\ldots,E$. □

Let $\pi(\hat{x}(t_k)) := \{u(t_k) \in \mathbb{R}^m \mid$ (217) holds$\}$ and let

$$x(t_{k+1}) \in \phi_{\mathsf{cl}}(x(t_k), \pi(\hat{x}(t_k))) := \{Ax(t_k) + Bu \mid u \in \pi(\hat{x}(t_k))\}$$

denote the difference inclusion corresponding to system (211) in "closed-loop" with the set of feasible solutions obtained by solving Problem 1 at each $k \in \mathbb{Z}_+$.

Theorem 17. *Let $\alpha_1, \alpha_2, \alpha_3 \in \mathcal{K}_\infty$ of the form specified in Theorem 14, a continuous and convex Lyapunov function V and a cost J be given. Let $\overline{\mathbb{W}}$ be a bounded set such that $\mathbb{W}(t_k) \subseteq \overline{\mathbb{W}}$ for all t_k. Suppose that Problem 1 is feasible for all states $\hat{x}$ in $\mathbb{X} \oplus \overline{\mathbb{W}}$. Then the difference inclusion*

$$x(t_{k+1}) \in \phi_{\mathit{cl}}(x(t_k), \pi(\hat{x}(t_k))), \quad k \in \mathbb{Z}_+ \tag{218}$$

is ISS in $\mathbb{X}$ for inputs in $\overline{\mathbb{W}}$.

Proof: As $x(t_k) \in \{\hat{x}(t_k)\}\oplus\mathbb{W}(t_k)$ for all $k \in \mathbb{Z}_+$ we have from (217b) that $x(t_{k+1}) = Ax(t_k)+Bu(t_k) \in \mathbb{X}$, i.e. the real state satisfies state constraints and, $\hat{x}(t_{k+1}) \in \mathbb{X} \oplus \mathbb{W}(t_{k+1}) \subseteq \mathbb{X} \oplus \overline{\mathbb{W}}$, i.e. Problem 1 remains feasible at $k+1$ and $\mathbb{X}$ is an RPI set w.r.t. the estimation error. Then, from the definitions (215) and (216), and from (217c), (217d) we have that

$$\begin{aligned} V(A\hat{x}(t_k) + Bu(t_k)) - V(x(t_k)) + \alpha_3(\|x(t_k)\|) \leq 0, \\ V(A(\hat{x}(t_k) - w_e(t_k)) + Bu(t_k)) - V(x(t_k)) \\ + \alpha_3(\|x(t_k)\|) - \gamma_e(t_k) \leq 0, \end{aligned}$$

for all $x(t_k) \in \{\hat{x}(t_k)\} \oplus \mathbb{W}(t_k)$, as $V(x(t_k)) \geq V_{\mathsf{min}}(t_k)$ and also $\alpha_3(\|x(t_k)\|) \leq \alpha_{3,\mathsf{max}}(t_k)$. Then, from Theorem 16 we have that (204b) holds with $\sigma(s) := \eta(t_k)s$ and $\eta(t_k)$ as in (214), i.e.

$$V(Ax(t_k) + Bu(t_k)) - V(x(t_k)) + \alpha_3(\|x(t_k)\|) - \sigma(\|w(t_k)\|) \leq 0,$$

for all $x(t_k) \in \{\hat{x}(t_k)\} \oplus \mathbb{W}(t_k)$. Next let

$$\gamma^* := \max_{x\in\mathrm{cl}(\mathbb{X}), u\in\mathrm{cl}(\mathbb{U})} \{V(Ax + Bu) - V(x) + \alpha_3(\|x\|)\}.$$

Since $\mathbb{X}$, $\mathbb{U}$ and $\overline{\mathbb{W}}$ are assumed to be bounded sets, γ^* exists, and inequality (217d) is always satisfied for $\gamma_e(t_k) = \gamma^*$ for all $e = 1,\ldots,E$, $k \in \mathbb{Z}_+$, irrespective of x, u and the vertices of $\mathbb{W}(t_k) \subseteq \overline{\mathbb{W}}$. This in turn, via (214) ensures the existence of a positive η^* such that $\eta(t_k) \leq \eta^*$ for all t_k and for all $w(t_k) \in \overline{\mathbb{W}}$. Hence, we proved that inequality (212) holds, and thus, the continuous and convex Lyapunov function V is a ISS Lyapunov function. Then, due to RPI of $\mathbb{X}$, ISS in $\mathbb{X}$ for inputs in $\overline{\mathbb{W}}$ follows directly from Theorem 14. □

Note that in Theorem 17 we used a worst case evaluation of $\gamma_e(k)$ to prove ISS, which corresponds to a worst case evaluation of the set $\mathbb{W}(t_k)$. However, in reality the gain $\eta(t_k)$ of the function σ can be much smaller for $k \geq k_0$, for some $k_0 \in \mathbb{Z}_+$. This is achieved via the minimization of the cost $J(\cdot)$, which produces small values of $\gamma_e(t_k)$, $e = 1,\ldots,E$. This in turn, via (214), will result in a small $\eta(t_k)$. Furthermore, this will ultimately yield a smaller ISS gain for the closed-loop system, due to the relation (205). Hence, Problem 1, although it inherently guarantees a constant ISS gain, as shown in the proof of Theorem 17, it provides freedom to optimize the ISS gain of the closed-loop system, by minimizing the variables $\gamma_1(t_k),\ldots,\gamma_E(t_k)$ via the cost $J(\cdot)$.

Remark 9. *The relations* (205) *and* (214) *yield an explicit expression of the gain of* γ *at every time instant. This can be used to set-up an event triggering mechanism as follows: at every* $k \in \mathbb{Z}_+$*, if NCS requirements allow event generation and* $\eta(t_k) \geq \eta_{\mathsf{bound}}$ *trigger event. In this way, a healthy trade-off between minimization of data transmission and performance can be achieved.* □

A cost on the future states can be added to Problem 1. As ISS is guaranteed for any feasible solution, optimization of the cost can still be used to improve performance, without requiring that the global optimum is attained in real-time. An example of such a cost will be given next.

In what follows, we will indicate certain ingredients which allow the implementation of Problem 1 via linear programming. For this, we restrict our attention to Lyapunov functions defined using the infinity norm, i.e.,

$$V(x) = \|P_V x\|_\infty, \tag{219}$$

where $P_V \in \mathbb{R}^{p\times n}$ is a full-column rank matrix. Note that this type of function satisfies (204a), for $\alpha_1(s) := \frac{\nu(P_V)}{\sqrt{p}}s$ (where $\nu(P_V) > 0$ is the smallest singular value of P_V) and for $\alpha_2(s) := \|P_V\|_\infty s$. Letting $\alpha_3(s) := cs$ for some $c \in \mathbb{R}_+$ we directly obtain from (216) that $\alpha_{3,\mathsf{max}}(t_k)$ is equal to the maximum of α_3 over the vertices of the set $\{\hat{x}(t_k)\} \oplus \mathbb{W}(t_k)$. Similarly, it is sufficient to impose (217b) only for the vertices of $\{\hat{x}(t_k)\} \oplus \mathbb{W}(t_k)$, i.e., for $\{w^e(t_k) + \hat{x}(t_k)\}_{e\in\mathbb{Z}_{[1,E]}}$.

By definition of the infinity norm, for $\|x\|_\infty \leq c$ to be satisfied for some vector $x \in \mathbb{R}^n$ and constant $c \in \mathbb{R}_+$, it is necessary and sufficient to require that $\pm [x]_i \leq c$ for all $i \in \mathbb{Z}_{[1,n]}$. So, for (217c)-(217d) to be satisfied, it is necessary and sufficient to require that

$$\begin{aligned} &\pm [P_V(A\hat{x}(t_k) + Bu(t_k))]_i - V_{\mathsf{min}}(t_k) + \alpha_{3,\mathsf{max}}(t_k) \leq 0,\\ &\pm [P_V(A(\hat{x}(t_k) - w_e(t_k)) + Bu(t_k))]_i - V_{\mathsf{min}}(t_k)\\ &\qquad\qquad\qquad\qquad + \alpha_{3,\mathsf{max}}(t_k) - \gamma_e(t_k) \leq 0 \end{aligned} \tag{220}$$

for all $i \in \mathbb{Z}_{[1,p]}$ and $e \in \mathbb{Z}_{[1,E]}$. Moreover, by choosing an infinity-norm based cost function

$$\begin{aligned} J(x(t_k), u(t_k), \gamma_i(t_k)) &:= \|P_J(A(\hat{x}(t_k) - w(t_k)) + Bu(t_k))\|_\infty\\ &+ \|Q_J(\hat{x}(t_k) - w(t_k))\|_\infty + \|R_J u(t_k)\|_\infty + \sum_{i=1}^{E} \|\Gamma_i\gamma_i(t_k)\|_\infty, \end{aligned} \tag{221}$$

with full-column rank matrices $P_J \in \mathbb{R}^{p_J\times n}$, $Q_J \in \mathbb{R}^{q_J\times n}$, $R_J \in \mathbb{R}^{r_J\times m}$ and $\Gamma_i \in \mathbb{R}_+$, we can reformulate the optimization of the cost J subject to the constraints (217) as the linear program

$$\min_{u(t_k),\gamma_1(t_k),\ldots,\gamma_E(t_k),\varepsilon_1,\varepsilon_2} \varepsilon_1 + \varepsilon_2 + \sum_{i=1}^{E} \Gamma_i\gamma_i(t_k) \tag{222}$$

subject to (217a), (217b), (220) and

$$\begin{aligned} &A(w_e(t_k) + \hat{x}(t_k)) + Bu(t_k) \in \mathbb{X}, \quad \forall e \in \mathbb{Z}_{[1,E]},\\ &\pm [P_J(A(\hat{x}(t_k) - w_{e_1}(t_k)) + Bu(t_k))]_i + \|Q_J(\hat{x}(t_k) - w_{e_2}(t_k))\|_\infty \leq \varepsilon_1\\ &\forall(e_1, e_2) \in \mathbb{Z}_{[1,E]} \times \mathbb{Z}_{[1,E]}, \forall i \in \mathbb{Z}_{[1,p_J]},\\ &\pm [R_J u(t_k)]_i \leq \varepsilon_2, \quad \forall i \in \mathbb{Z}_{[1,r_J]}. \end{aligned}$$

The only thing left for implementing Problem 1 is to compute $V_{\mathsf{min}}(k)$. Using the same reasoning as above, it can be shown that the optimization problem (215) can be formulated as a linear program. As such, finding a solution to Problem 1 amounts to solving 2 linear programs and calculating the maximum over a finite set of real numbers, which can be performed efficiently.

This completes the design procedure of the robust MPC and we continue with the integration of the EBSE and MPC.

4.2.5 Integration of EBSE and MPC

In this section we provide a method for designing a set $\mathbb{W}(t_k)$ based on $P(t_k)$. To that extent, we will use the fact that $P(t_k)$ is a model for $cov(x(t_k) - \hat{x}(t_k))$, i.e., the covariance matrix of $w(t_k)$.

Hence, $G(w(t_k), \underline{0}, P(t_k))$ is a model for the true PDF $p(w(t_k))$. The first step in this design is defining ellipsoidal sets that are based on $P(t_k)$. Each ellipsoidal set represents a model of the probability that the true estimation-error, i.e., $w(t_k)$, is within that set. The second step is to define $\mathbb{W}(t_k)$ as an over-approximation of a certain ellipsoidal set that can be computed efficiently and such that $w(t_k) \in \mathbb{W}(t_k)$ has a high probability. As all variables of this section are at the time-instant t_k, we will omit t_k from every time-dependent variable, i.e. $w(t_k)$ and $P(t_k)$ become w and P, respectively. Let us start by describing the sub-level-sets of a Gaussian. For any Gaussian $G(w, \underline{0}, P)$ one can define the sub-level set $\varepsilon(P, c) \subset \mathbb{R}^n$, for some $c \in \mathbb{R}_+$, as follows:

$$\varepsilon(P, c) := \left\{ w \in \mathbb{R}^n \middle| w^\top P^{-1} w \leq c \right\}. \tag{223}$$

The shape of this sub-level-set is an ellipsoid, or an ellipse in the 2D case as it is graphically depicted in Figure 29 (for $c = 1$). This figure illustrates the relation of the ellipsoid with the eigenvalues of the corresponding covariance matrix P. In this case the relation between eigenvalues of P yields $\lambda_1(P) > \lambda_2(P)$.

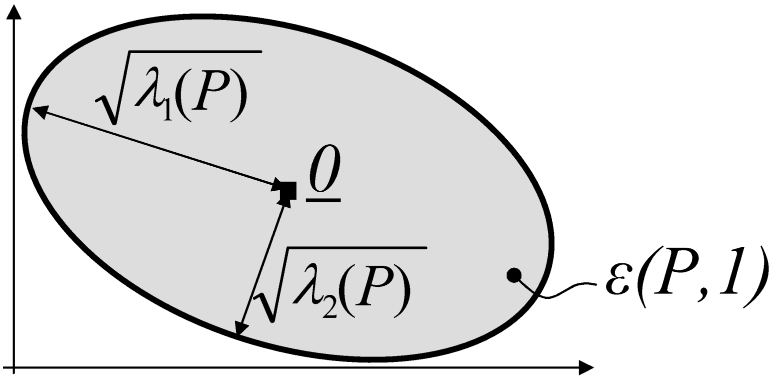


Figure 29: Graphical representation of the Gaussian $G(w, \underline{0}, P)$ as the sub-level-set $\varepsilon(P, 1)$. The direction of each arrow is defined as the eigenvector of the corresponding eigenvalue $\lambda_i(P)$.

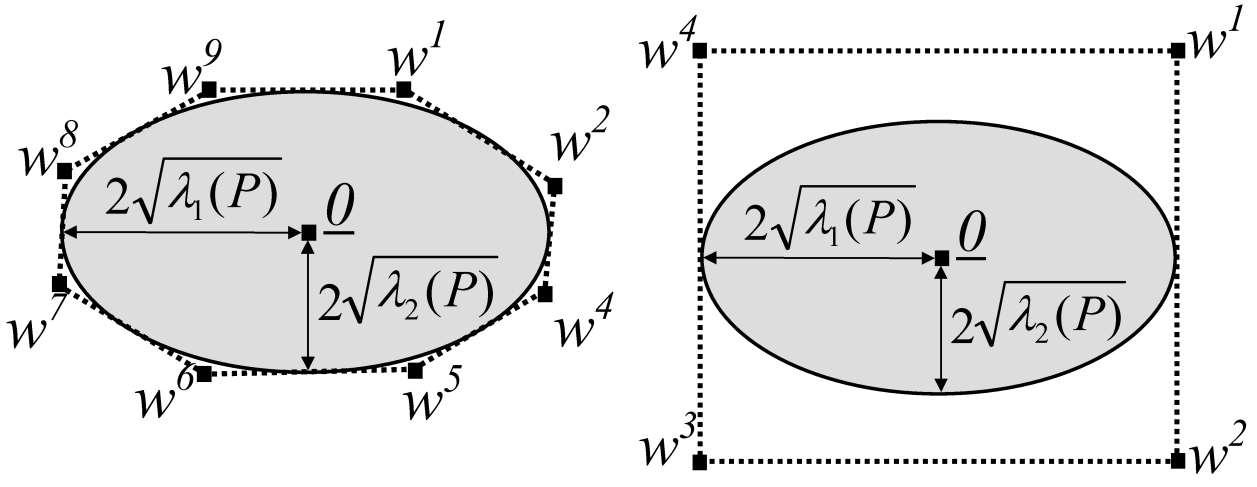
Applying some basics of probability theory one can calculate the probability, depending on c, that the random vector w is within the set $\varepsilon(w, P, c)$. Some examples of this probability, denoted with $Pr(\cdot)$, for different values of c are:

$$\begin{aligned} Pr\left(w \in \varepsilon(P, 1)\right) &\approx 0.68, \\ Pr\left(w \in \varepsilon(P, 4)\right) &\approx 0.95, \\ Pr\left(w \in \varepsilon(P, 9)\right) &\approx 0.997. \end{aligned}$$

Notice that $\varepsilon(P, c)$ defines an ellipsoid. As such, to use the MPC law as proposed in the previous section, we need to obtain a polytopic over-approximation of this set. This over-approximated set can then be used as the set $\mathbb{W}$ where the estimation error is bounded. Recall that the vertices of $\mathbb{W}$ are used in the controller to determine the control action. There is no optimal method to calculate this set, as it amounts to the ancient problem of “squaring the circle”. See for example the recent results in [1] and the references therein. What can be stated is that a trade-off should be made between the size of the set $\mathbb{W}$ on the one hand, indicating the worst case estimation error bound, and the computational complexity of obtaining $\mathbb{W}$ on the other hand, which should be kept reasonable for on-line calculation.

If for example one chooses that $Pr(w \in \mathbb{W}) = 0.95$, then $\mathbb{W}$ can be taken as a tight over-approximation of $\varepsilon(P, 4)$, as illustrated in Figure 30(a). However, to obtain this tight over-approximation, knowledge of all the eigenvalues of P and vertex computation is required, which is computationally expensive. If the real-time properties of the resulting algorithm allow this tight over-approximation, then Figure 30(a) can be considered. However, here we want to have the least processing-time, i.e. computational complexity. Therefore, we aim at describing the vertices of $\mathbb{W}$ by a single parameter that depends on the maximum eigenvalue of P. An example of such a set is shown in Figure 30(b). Therein, $\mathbb{W}$ is fully determined by the scalar $d = \sqrt{c\lambda_{\text{max}}(P)} = 2\sqrt{\lambda_{\text{max}}(P)}$, which represents an upper bound on the infinity norm of w.

Notice that the vertices of $\mathbb{W}$ as shown in Figure 30(b) are explicitly defined as all possible realizations of the vectors $(w^1, \ldots, w^E)^\top$ when $[w^i]_j \in \{-d, d\}$ for all $j \in \mathbb{Z}_{[1,n]}$ and $i \in \mathbb{Z}_{[1,E]}$. The next result



(a) Accurate over-approximate.

(b) Fast over-approximation.

Figure 30: Examples of over-approximating $\varepsilon(P, 4)$ by $\mathbb{W}$.

provides an expression for $d \in \mathbb{R}_+$, which is calculated at each synchronous sample instant t_k, such that $\varepsilon(P, c) \subseteq \mathbb{W}$, for a given $c \in \mathbb{R}_+$.

Lemma 7. *Suppose a random vector $w \in \mathbb{R}^n$ is given of which its PDF is a Gaussian with zero mean and a covariance-matrix $P \in \mathbb{R}^{n \times n}$. Let us define $\mathbb{W} := \{w \mid \|w\|_\infty \leq d\}$ with $d := \sqrt{c * \lambda_{\mathit{max}}(P)}$. Then for any $c \in \mathbb{R}_+$, it holds that $\varepsilon(P, c) \subset \mathbb{W}$.*

Proof. Let $w \in \varepsilon(P, c)$. Then it holds that

$$\lambda_{\mathsf{min}}(P^{-1})\|w\|_\infty^2 \leq \lambda_{\mathsf{min}}(P^{-1})\|w\|_2^2 \leq w^\top P^{-1} w \leq c,$$

from which $\|w\|_\infty \leq \sqrt{c(\lambda_{\mathsf{min}}(P^{-1}))^{-1}}$ follows. Applying the fact that $\lambda_{\mathsf{min}}(P^{-1}) = (\lambda_{\mathsf{max}}(P))^{-1}$ gives that

$$\|w\|_\infty \leq \sqrt{c * \lambda_{\mathsf{max}}(P)}.$$

Hence, $w \in \mathbb{W}$ and as $w \in \varepsilon(P, c)$ was arbitrary, the proof is complete. □

The above results shows that in order to compute the vertices of the set $\mathbb{W}$ at each t_k, one only needs to calculate the current $\lambda_{\mathsf{max}}(P)$. The computational efficiency of the procedure for obtaining the set $\mathbb{W}$ can be further improved by using the known fact that $\lambda_{\mathsf{max}}(P) \leq \mathrm{tr}(P)$, at the cost of a more conservative error bound. To increase the probability that $w(t_k) \in \mathbb{W}(t_k)$, one can use Lemma 7 to observe that:

$$Pr\left(w(t_k) \in \varepsilon(P(t_k), c)\right) \leq Pr\left(w(t_k) \in \mathbb{W}(t_k)\right).$$

Therefore, by choosing $c = 9$ yields:

$$\begin{aligned} 0.997 &\approx Pr\left(w(t_k) \in \varepsilon(P(t_k), 9)\right) \\ &\leq Pr\left(w(t_k) \in \mathbb{W}(t_k)\right). \end{aligned} \tag{224}$$

Remark 10. *As the covariance matrix $P(t)$ is bounded for all $t \in \mathbb{T}$, shown in [59, 60], it follows that $\lambda_{\mathit{max}}(P(t))$ is also bounded for all $t \in \mathbb{T}$. Then, by the definition of the set $\mathbb{W}(t_k)$, it holds that $\mathbb{W}(t_k) \subseteq \overline{\mathbb{W}}$ for all t_k and*

$$\overline{\mathbb{W}} := \left\{ w \in \mathbb{R}^n \mid \|w\|_\infty \leq \sup_t \sqrt{c\lambda_{\mathit{max}}(P(t))} \right\}.$$

□

This completes the overall design of the feedback loop that consists of *(i)* the EBSE algorithm, which provides an estimate of the state $\hat{x}(t_k)$ at all t_k, *(ii)* the algorithm for computation of the vertices of $\mathbb{W}(t_k)$ at all t_k such that $w(t_k) = \hat{x}(t_k) - x(t_k) \in \mathbb{W}(t_k)$ has a high probability and *(iii)* the robust MPC algorithm, which provides inherent ISS w.r.t. to estimation errors $w(t_k)$ and optimized ISS w.r.t. to $w(t_k) \in \mathbb{W}(t_k)$.

The only aspect left to be treated is concerned with the fact that the EBSE is a stochastic estimator, while the robust MPC algorithm is a deterministic controller. Due to the stochastic nature of the estimator, one does not have a guarantee that $w(t_k) \in \mathbb{W}(t_k)$ for all t_k and, as such, that the real state is bounded. Instead, one has the information that $w(t_k) \in \mathbb{W}(t_k)$ for all t_k only with a certain (high) probability, which implies that the ISS property of the MPC scheme applies to the overall integrated closed-loop system in a probabilistic sense only. If for some $t_k \in \mathbb{R}_+$, $w(t_k) \notin \mathbb{W}(t_k)$ but it is still bounded, it would be desirable to still have an ISS guarantee (i.e., independent of the bound on $w(t_k)$) for the closed-loop system. A possible solution could be to use a uniformly continuous control Lyapunov function to establish inherent ISS to the estimation error, which forms the object of future research.

4.2.6 Illustrative example

In this section we illustrate the effectiveness of the developed EBSE and robust MPC scheme. The case study is a 1D object-tracking system. The states $x(t)$ of the object are position and speed while the measurement-vector $y(t)$ is position. The control input $u(t)$ is defined as the object's acceleration. The states and control input are subject to the constraints $x(t) \in \mathbb{X} = [-5,5] \times [-5,5]$ and $u(t) \in \mathbb{U} = [-2,2]$. Both the process-noise as well as the measurement-noise are chosen to have a zero-mean Gaussian PDF with $Q_\tau = 3\tau \cdot 10^{-4} I$ and $R_v = 1 \cdot 10^{-4}$. As the process is a double integrator, the process model becomes:

$$\begin{aligned} x(t+\tau) &= \begin{pmatrix} 1 & \tau \\ 0 & 1 \end{pmatrix} x(t) + \begin{pmatrix} \frac{\tau^2}{2} \\ \tau \end{pmatrix} u(t) + q(t,\tau), \\ y(t) &= \begin{pmatrix} 1 & 0 \end{pmatrix} x(t) + v(t). \end{aligned} \tag{225}$$

The sampling time of the controller is $\tau_s = 0.7[s]$. For simplicity, we use "Send-on-Delta" as the sampling method with $\Delta = 0.1[m]$. This means that in case the object drove an additional 0.1 meter with respect to its last sampled position, a new measurement of the position is taken. Therefore, the set which defines event sampling becomes $H_{r|t} = [y(t_{r-1}) - \Delta, y(t_{r-1}) + \Delta]$. The PDF $\Lambda_{H_{r|t}}(y(t))$, for all $t \in \mathbb{T}$, of the EBSE is approximated as a sum of 5 Gaussians that are equidistantly distributed along $[y(t_{r-1}) - \Delta, y(t_{r-1}) + \Delta]$. Therefore, we set

$$\begin{aligned} &N = 5, \quad y_i(t) = y(t_{r-1}) - \left(\frac{N - 2(i-1) - 1}{N} \right) \Delta, \quad \forall i \in \mathbb{Z}_{[1,N]}, \\ &R_H(t) = \left(\frac{2\Delta}{N} \right)^2 \left(0.25 - 0.05 e^{-\frac{4(N-1)}{15}} - 0.08 e^{-\frac{4(N-1)}{180}} \right). \end{aligned}$$

Next, let us design the parameters of the robust MPC. The technique of [36] was used to compute the weight $P_V \in \mathbb{R}^{2\times 2}$ of the Lyapunov function $V(x) = \|P_V x\|_\infty$ for $\alpha_3(s) := 0.01s$, yielding

$$P_V = \begin{pmatrix} 2.7429 & 0.7121 \\ 0.1989 & 4.0173 \end{pmatrix}.$$

Following the procedure described in Section 4.2.5, the set $\mathbb{W}(t_k)$ will have 4 vertices at all $k \in \mathbb{Z}_+$. As such, to optimize robustness, 4 optimization variables $\gamma_1(t_k), \cdots, \gamma_4(t_k)$ were introduced, each one assigned to a vertex of the set $\mathbb{W}(t_k)$. The MPC cost was chosen as $J(x(t_k), u(t_k), \gamma_1(t_k), \ldots, \gamma_4(t_k))$ with $P_J = 0.4I$, $Q_J = 0.2I$, $R_J = 0.1$ and $\Gamma_i = 4$ for all $i \in \mathbb{Z}_{[1,4]}$. The resulting linear program has 11 optimization variables and 108 constraints. In this cost-function, the variable $V_{\text{min}}(t_k)$ of equation (215) is used. Its value is also calculated via solving a linear program with 3 optimization variables and 5 constraints. During the simulations, the worst case computational time required by the CPU over 100 runs was 20 $[ms]$ for the controller and 5 $[ms]$ for the EBSE, which shows the potential for controlling fast linear systems.

In the simulation scenario we tested the closed-loop system response for $x(t_0) = [3,1]^\top$ with the origin as reference. The initial state estimates of the EBSE were chosen as $\hat{x}(t_0) = [3.5, 1.2]^\top$ and

$P(t_0) = I$. The evolution of the *true* state is graphically depicted in Figure 31. Figure 32 presents the control input $u(t_k)$. The evolution of the different values for $\gamma_1(t_k), \cdots, \gamma_4(t_k)$ is shown in Figure 33. Figure 34 presents the absolute error between the estimated and true state, i.e. position $|[w(t_k)]_1|$ and speed $|[w(t_k)]_2|$, and the modeled error bound which is chosen to be $d = \sqrt{9\lambda_{\mathsf{max}}(P(t_k))}$ and defines $\mathbb{W}(t_k)$.

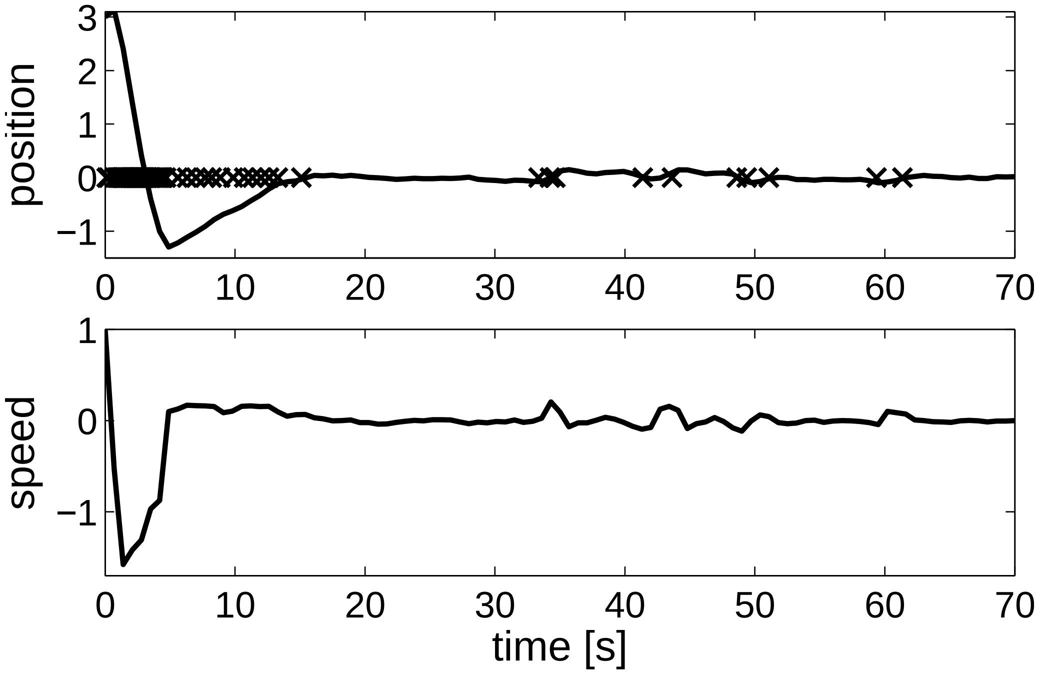


Figure 31: Evolution of the true state: position, i.e. $x_1(t_k)$ and speed, i.e. $x_2(t_k)$.

The symbols "$\times$" in the plot of the true object position of Figure 31 denote the instants when an event occurs, i.e., whenever the object drove an additional 0.1 $[m]$ with respect to its previous sampled position measurement. Notice that the number of events increases when position is changing fast. Therefore, a large amount of samples are generated in the first 5 seconds. After 20 seconds, both the position and speed of the true state are zero and no event occurs anymore.

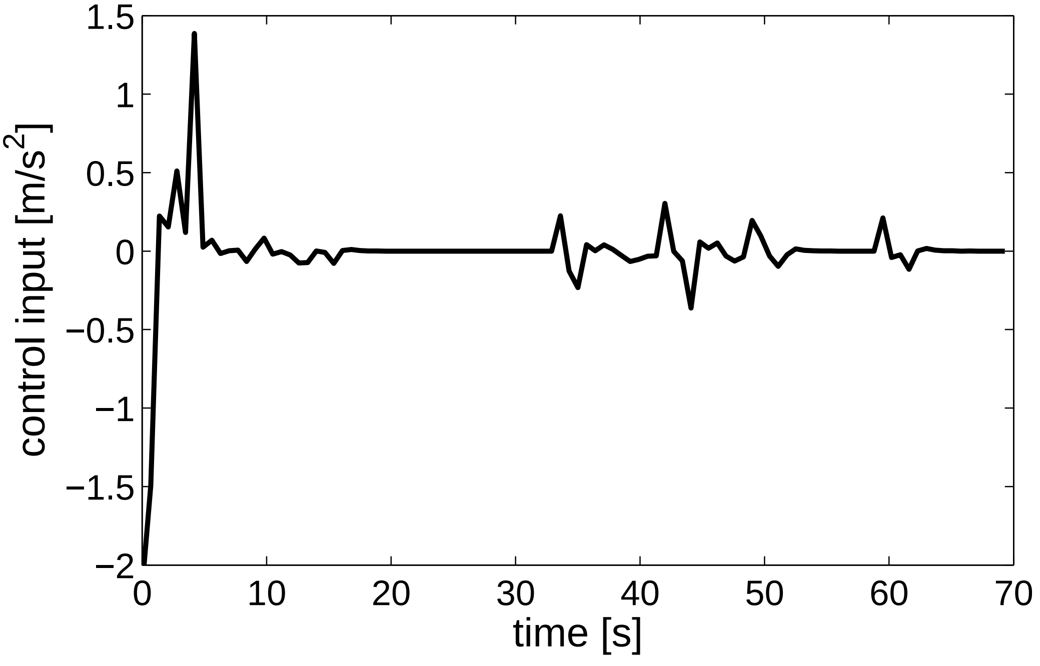


Figure 32: Evolution of the the control input $u(t_k)$.

Figure 32 shows that the input constraints are fulfilled at all times, and sometime they are active.

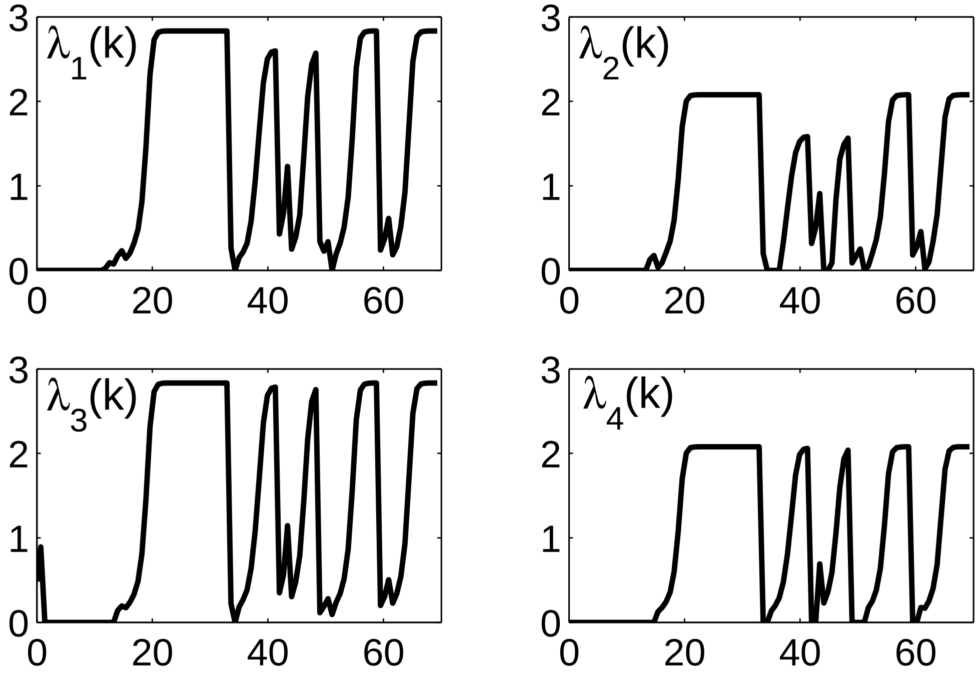


Figure 33: Evolution of the 4 optimization variables $\gamma_i(t_k)$.

Notice that when the state is close to zero, due to the optimization of the cost J, this pushes the control input to zero. As such, the optimization variables $\gamma_i(t_k)$ of Figure 33 must satisfy $V(w^e(t_k)) - \gamma_e(t_k) \leq 0$, $e = 1, \ldots, E$. This explains the non-zero value of $\gamma_i(t_k)$ when the state reaches the equilibrium, for example in between 20 and 30 seconds.

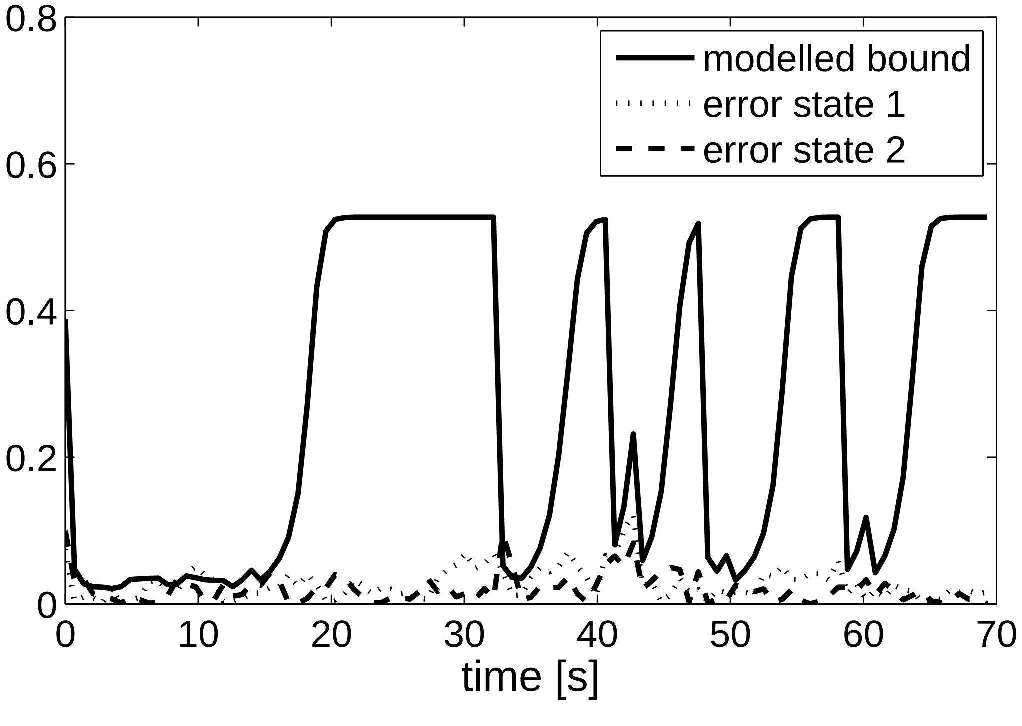


Figure 34: Evolution of the true estimation error compared to the bounds on the modeled one.

Figure 34 shows that the true estimation-error remained within the limits of the modeled one at almost all sample instances. Furthermore, after 20 seconds no new measurement is received anymore for some longer time period. Standard state-estimators would predict the state in that case, causing $\lambda_{\text{max}}(P(t_k))$ to diverge. Due to the EBSE, which makes use of the bounded set $H_{r|t}$, $\lambda_{\text{max}}(P(t_k))$ converges to a constant, although no events are generated anymore. This confirms that boundedness of the covariance matrix of the EBSE is independent of the number of events for a bounded set $H_{r|t}$, although the actual value of the bound is influenced by the choice of $H_{r|t}$. Notice the variety in the modeled estimation-error at 33, 41, 49 and 60 seconds. This is caused by the fact that an event occurs at these instances. Hence, the state-update is based on a received measurement rather then a bounded set, which reduces its uncertainty. A final remark is to be made on the fact that at around 10 seconds the true estimation-error exceeds the modeled one. This means that at this time instant $w(t_k) \notin \mathbb{W}(t_k)$. Nevertheless, the trajectory of the closed-loop system remains bounded with reasonable robust performance, which encourages us to further analyze the ISS of the integrated closed-loop system. In light of the solution proposed at the end of the previous section, it is worth to mention that the CLF used in the example is globally Lipschitz.

For future research it would also be interesting to compare the results obtained with the developed robust MPC scheme with a stochastic MPC set-up, such as the algorithm presented in [10].

4.2.7 Conclusions

In this paper an event-based control system was designed with the property that control actions can take place synchronously in time but data transfer between the plant and the controller is kept low. This was achieved by introducing an event-based state estimator in the feedback loop. The event-based estimator was used to obtain a state estimate with a bounded covariance matrix in the estimation error at every synchronous time instant, under the assumption that the set used for event generation is bounded in the measurement-space. This covariance matrix was then used to estimate explicit polytopic bounds on the estimation-error that were fed into a robust MPC algorithm. We proved that the resulting MPC controller achieves ISS to the estimation error and, moreover, it optimizes the closed-loop trajectory-dependent ISS gain. We provided justification of our main ideas on all the parts of the overall ”output-based controller” (e.g., a bounded covariance matrix of ESBE-plant interconnection and ISS of MPC-plant loop) that show that in principle such an event-based controller should work. Several aspects of the integration of the stochastic event-based estimator and the deterministic MPC algorithm were discussed. The formal proof of closed-loop properties of the EBSE-MPC-plant interconnection or its variations is a topic of future research, although simulations provide convincing and promising evidence of the potential of the proposed methods.

5 Benchmark Results

5.1 Decentralized Temperature Control of a Passenger Train

D. Barcelli, A. Bemporad

Below the model of the Passenger train is recalled in order to improve readability of the present deliverable.

Consider the following discrete-time linear global process model

$$\left\{\begin{array}{rcl} z(t+1) & = & Az(t)+Bv(t)+F_d(t) \\ h(t) & = & Cz(t) \end{array}\right. \tag{226}$$

where $z \in \mathbb{R}^n$ is the state vector, $v \in \mathbb{R}^m$ is the input vector, $y \in \mathbb{R}^p$ is the output vector, $F_d \in \mathbb{R}^d$ is a vector of measured disturbances. Let A be stable and assume F_d is constant. We aim at solving a set-point tracking problem so that h tracks a given reference value $r \in \mathbb{R}^p$, despite the presence of F_d. In order to recast the problem as a regulation problem, assume steady-state vectors $z_r \in \mathbb{R}^n$ and $v_r \in \mathbb{R}^m$ exist solving the static problem

$$\left\{\begin{array}{r} (I-A)z_r = Bv_r + F_d \\ r = Cz_r \end{array}\right. \tag{227}$$

and let $x = z - z_r$ and $u = v - v_r$. Input constraints $v_{\min} \leq v \leq v_{\max}$ are mapped into constraints $v_{\min} - v_r \leq u \leq v_{\max} - v_r$[9].

Proposition 3. *Under the global coordinate transformation* (227)*, the process* (226) *under the decentralized MPC law* (126)–(128) *is such that* $h(t)$ *converges asymptotically to the set-point* r*, under the assumption of Theorem in [100] or, in the presence of packet drops, of Theorem 8.*

Note that problem (227) is solved in a *centralized* way. Defining local coordinate transformations v_{ir}, z_{ir} based on submodels (123) would not lead, in general, to offset-free tracking, due to the mismatch between global and local models. This is a general observation one needs to take into account in decentralized tracking. Note also that both v_r and z_r depend on F_d as well as r, so problem (227) should be solved each time F_d or r change.

In this section we test the proposed DMPC approach for decentralized control of the temperature in different passenger areas in a railcar. The system is schematized in Figure 35. Each passenger area has its own heater and air conditioner but its thermal dynamics interacts with surrounding areas (neighboring passenger areas, external environment, antechambers) directly or through windows/walls/doors. The internal doors can be opened by passengers, external doors automatically open at train stops. Passenger areas are composed by a table and the associated four seats. Temperature sensors are located in each four-seat area, in each antechamber, and along the corridor. The goal of the controller is to adjust each passenger area to its own temperature set-point to maximize passenger comfort.

Let $2N$ be the number of four-seat areas ($N = 8$ in Figure 35), N the number of corridor partitions, and 2 the number of antechambers. Under the assumption of perfectly mixed fluids in each jth volume, $j = 1, \ldots, n$ where $n = 3N + 2$, the heat transmission equations by conduction lead to the linear model $\frac{dT_j(\tau)}{d\tau} = \sum_{i=0}^{n} Q_{ij}(\tau) + Q_{uj}$, $Q_{ij}(\tau) = \frac{S_{ij}K_{ij}(T_i(\tau)-T_j(\tau))}{C_j L_{ij}}$, $j = 1, \ldots, n$, where $T_j(\tau)$ is the temperature of volume $\#j$ at time $\tau \in \mathbb{R}$, $T_0(\tau)$ is the ambient temperature outside the railcar, $Q_{ij}(\tau)$ is heat flow due to the temperature difference $T_i(\tau) - T_j(\tau)$ with the neighboring volume $\#i$, S_{ij} is the contact surface area, Q_{uj} is the heat flow of heater $\#j$, K_{ij} is the thermal coefficient that depends on the materials, $C_j = K_c^j V_j$ is the the (material dependent) heat capacity coefficient K_c^j times the fluid volume V_j, and L_{ij} is the thickness of the separator between the two volumes $\#i$ and $\#j$. We

[9] In case $v_r \notin [v_{\min}, v_{\max}]$, perfect tracking under constraints is not possible, and an alternative is to set

$$\begin{array}{rcll} \left[\begin{smallmatrix} z_r \\ v_r \end{smallmatrix}\right] & = & \arg\min & \left\|\left[\begin{smallmatrix} I-A & -B \\ C & 0 \end{smallmatrix}\right]\left[\begin{smallmatrix} z_r \\ v_r \end{smallmatrix}\right] - \left[\begin{smallmatrix} F_d \\ r \end{smallmatrix}\right]\right\| \\ & & \text{s.t.} & v_{\min} \leq v_r \leq v_{\max} \end{array}$$

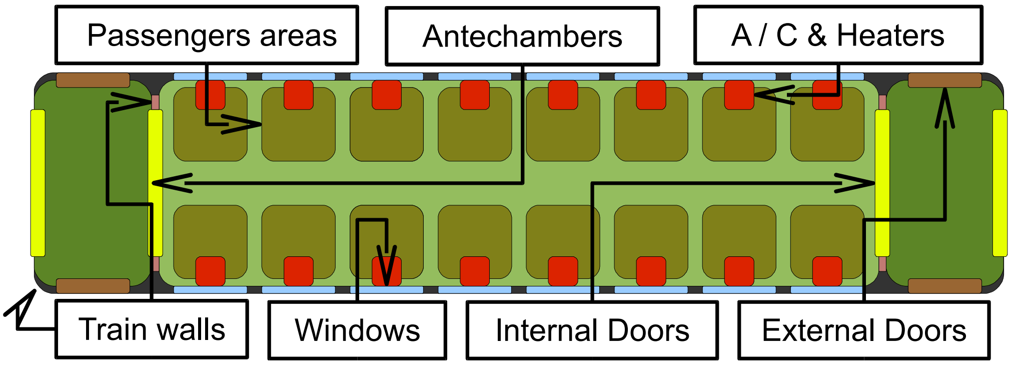


Figure 35: Physical structure of the railcar

assume that $Q_{ij}(\tau) = 0$ for all volumes i, j that are not adjacent, $\forall \tau \in \mathbb{R}$. Hence, the process can be modeled as a linear time-invariant continuous-time system with state vector $z \in \mathbb{R}^{3N+2}$ and input vector $v \in \mathbb{R}^{2N}$

$$\begin{cases} \dot{z}(\tau) &= A_c z(\tau) + B_c v(\tau) + F T_0(\tau) \\ h(\tau) &= C z(\tau) \end{cases} \tag{228}$$

where $F \in \mathbb{R}^n$ is a constant matrix, $T_0(\tau)$ is treated as a piecewise constant measured disturbance, and $C \in \mathbb{R}^{p \times n}$ is such that $h \in \mathbb{R}^p$ contains the components of z corresponding to the temperatures of the passenger seat areas, $p = 2N$. Since we assume that the thermal dynamics are relatively slow compared to the sampling time T_s of the decentralized controller we are going to synthesize, we use first-order Euler approximation to discretize dynamics (228) without introducing excessive errors:

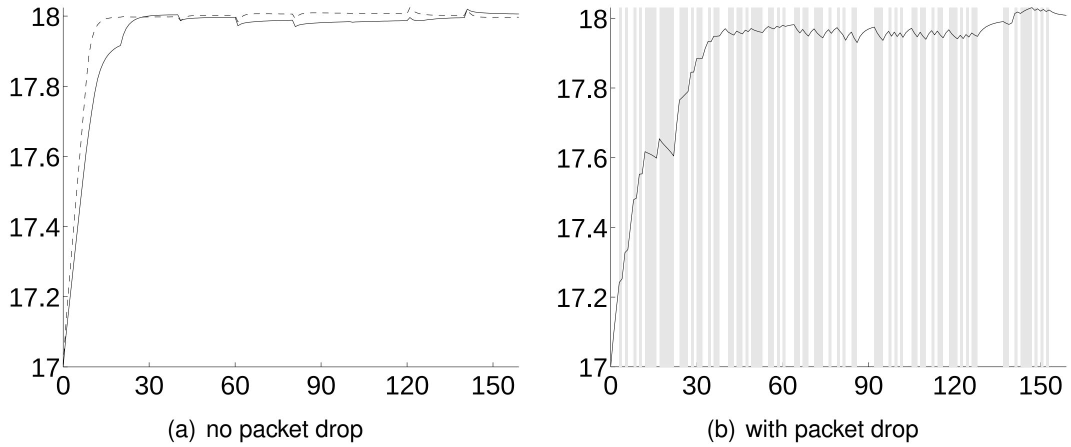
$$\begin{cases} z(t+1) &= A z(t) + B v(t) + F_d T_0(t) \\ h(t) &= C z(t) \end{cases} \tag{229}$$

where $A = I + A_c T_s$, $B = B_c T_s$, and $F_d = F T_s$. We assume that A is asymptotically stable, as an inheritance of the asymptotic stability of matrix A_c.

In order to track generic temperature references $r(t)$, we adopt the coordinate shift defined by (227). The next step is to decentralize the resulting global model. The particular topology of the railcar suggests a decomposition of model (118) as the cascade of four-seat areas. There are two kinds of four-seat areas, namely (i) the ones next to the antechambers, and (ii) the remaining ones. Besides interacting with the external environment, the areas of type (i) interact with another four-seat-area, an antechamber, and a section of the corridor, while the areas of type (ii) only with the four-seat areas at both sides and the corresponding section of the corridor. Note that the decentralized models overlap, as they share common states and inputs. The decoupling matrices Z_i are chosen so that in each subsystem only the first component of the computed optimal input vector is actually applied to the process.

As a result, each submodel has 5 states and 2 or 3 inputs, depending whether it describes a seat area of type (i) or (ii), which is considerably simpler than the centralized model (118) with 26 states and 16 inputs.

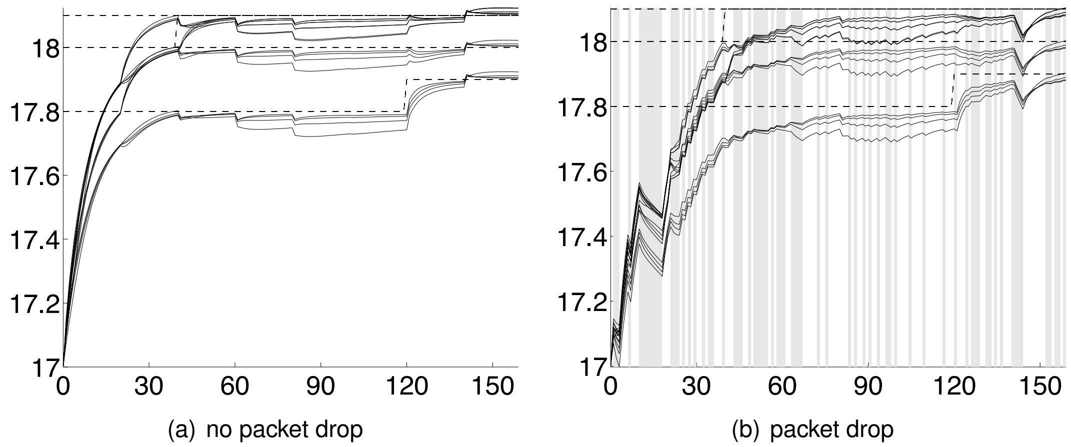
We apply the DMPC approach (126) with $Q = \left[\begin{smallmatrix} 200 I_{16} & 0 \\ 0 & 0.2 I_{10} \end{smallmatrix}\right]$, $R = 10^5 I_{16}$, $\left[\begin{smallmatrix} v_{\min} = -0.03 \\ v_{\max} = 0.03 \end{smallmatrix}\right]$ W, $T_s = 9$ min, where $v_{\min}$ is the lower bound on the heat flow subtracted by the air-conditioners, and $v_{\max}$ is the maximum heating power of the heaters (with a slight abuse of notation we denoted by $v_{\min}$, $v_{\max}$ the entries of the corresponding lower and upper bound vectors of $\mathbb{R}^{16}$). Note that the first sixteen diagonal elements of matrix Q correspond to the temperatures of the four-seat areas. It is easy to check that with the above parameters condition in [100] for local stability is satisfied. For comparison, a centralized MPC approach (120) with the same weights, horizon, and sampling time is also designed. The associated QP problem has 16 optimization variables and 32 constraints, while the complexity of each DMPC controller is either 2 (or 3) variables and 4 (or 6) constraints. The DMPC approach is in fact largely scalable: for longer railcars the complexity of the DMPC controllers remains the same, while the complexity of the centralized MPC problem grows with the increased model size. Note also that, if a multiple cores computation is taken, the DMPC approach can be immediately parallelized.



(a) no packet drop

(b) with packet drop

Figure 36: Comparison between centralized MPC (dashed lines) and decentralized MPC (continuous lines): output h_1 (upper plots) and input v_1 (lower plots). Gray areas denote packet drop intervals



(a) no packet drop

(b) packet drop

Figure 37: Decentralized MPC results. Upper plots: output variables h (continuous lines) and references r (dashed lines). Lower plots: command inputs v. Gray areas denote packet drop intervals

Simulation results

We investigate different simulation outcomes depending on four ingredients: i) type of controller (centralized/decentralized), ii) packet-loss probability, iii) change in reference values, iv) changes of external temperature (acting as a measured disturbance).

The initial condition is $17°$C for all seat-area temperatures, except for the antechamber, which is $15°$C. Note that the steady-state value of antechamber temperatures is not relevant for the posed control goals.

The closed-loop trajectories of centralized MPC feedback vs. decentralized MPC with no packet-loss are shown in Figure 36(a) (we only show the first state and input for clarity). In both cases the temperatures of the four-seat areas converge to the set-point asymptotically.

Figure 37(a) shows the temperature vector $h(t)$ tracking the time-varying reference $r(t)$ in the absence of packet-loss, where the coordinate transformation (227) is repeated after each set-point and external temperature change.

To simulate packet loss, we assume that the probability of losing a packet depends on the state of the Markov chain depicted in Figure 38.

The Markov chain is in the jth state if $j-1$ consecutive packets have been lost. The probability of losing a further packet is $1-p$, $0 \leq p \leq 1$, except for the $(N+1)$th state where no packet can be lost any more. Such a probability model is partially confirmed by the experimental results on relative frequencies of packet failure burst length observed in [113]. The simulation results obtained with

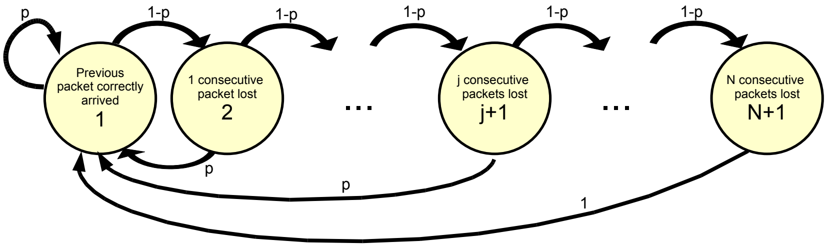

p
1-p
1-p
1-p
1-p
1-p
Previous packet correctly arrived
1
1 consecutive packet lost
2
...
j consecutive packets lost
j+1
...
N consecutive packets lost
N+1
p
p
1

Figure 38: Markov chain describing the loss probability

$p = 0.5$ are shown in Figure 36(b) and Figure 37(b). The stability condition (133a) of Theorem 8 was tested and proved satisfied for values of j up to 160.

The simulations were run on a MacBook Air 1.86 GHz running Matlab R2008a under OS X 10.5.6 and the Hybrid Toolbox for Matlab [102]. The average CPU time for solving the centralized QP problem associated with (120) is 6.0 ms (11.9 ms in the worst case). For the decentralized case, the average CPU time for solving the QP problem associated with (126) is 3.3 ms (7.4 ms in the worst case).

References

[1] A. Bemporad A. Alessio, M. Lazar and W. P. M. H. Heemels. Squaring the circle: An algorithm for obtaining polyhedral invariant sets from ellipsoidal ones. *Automatica*, 43, 2007.

[2] L. Aggoun and R. Elliot. *Measure Theory and Filtering*. Cambridge University Press, 2004.

[3] A.T. Al-Hammouri, M.S. Branicky, V. Liberatore, and S.M. Phillips. Decentralized and dynamic bandwidth allocation in networked control systems. In *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, pages 8 pp.–, April 2006.

[4] A. Alessandri and P. Coletta. Switching observers for continuous-time and discrete-time linear systems. In *Proceedings of the American Control Conference, 2001*, volume 3, pages 2516–2521 vol.3, 2001.

[5] B.D.O. Anderson and J.B. Moore. *Optimal Filtering*. Prentice Hall, 1979.

[6] P. Antsaklis and J. Baillieul. Special issue on technology of networked control systems. *Proceedings of the IEEE*, 95(1), 2007.

[7] K. J. Åström. Event based control. In *Analysis and Design of Nonlinear Control Systems: In Honor of Alberto Isidori*, eds. A. Astolfi and L. Marconi, Springer-Verlag, 2008.

[8] K. J. Åström and B. M. Bernhardsson. Comparison of Riemann and Lebesque sampling for first order stochastic systems. In *41st IEEE Conf. on Dec. and Contr.*, 2002.

[9] D. Bernardini and A. Bemporad. Energy-Aware Robust Model Predictive Control with Feedback from Multiple Noisy Wireless Sensors. In *Proceedings of the European Control Conference*, 2009.

[10] D. Bernardini and A. Bemporad. Scenario-based model predictive control of stochastic constrained linear systems. In *48th IEEE Conference on Decision and Control*, 2009.

[11] R. W. Brockett and D. Liberzon. Quantized feedback stabilization of linear systems. *IEEE Transactions on Automatic Control*, 97(7):1279–1289, 2000.

[12] M. B. G. Cloosterman, N. van de Wouw, W. P. M. H. Heemels, and H. Nijmeijer. Robust stability of networked control systems with time-varying network-induced delays. In *Proc. of the 45th*

IEEE Conference on Decision and Control, pages 4980–4985, San Diego, CA, USA, December 2006.

[13] M. B. G. Posthumus Cloosterman. *Control over Communication Networks: Modeling, Analysis, and Synthesis*. PhD thesis, Technische Universiteit Eindhoven, Eindhoven, the Netherlands, 2008.

[14] J. Daafouz and J. Bernussou. Parameter dependent Lyapunov functions for discrete time systems with time varying parametric uncertainties. *Systems and Control Letters*, 43(5):355–359, 2001.

[15] M. C. de Oliveira, J. Bernussou, and J. C. Geromel. A new discrete-time robust stability condition. *Systems & Control Letters*, 37:261–265, 1999.

[16] D. V. Dimarogonas and K. H. Johansson. Event-triggered control for multi-agent systems. In *IEEE Conference on Decision and Control*, 2009.

[17] M. C. F. Donkers, L. Hetel, W. P. M. H. Heemels, N. van de Wouw, and M. Steinbuch. Stability analysis of networked control systems using a switched linear systems approach. In *HSCC '09: Proceedings of the 12th International Conference on Hybrid Systems: Computation and Control*, pages 150–164, Berlin, Heidelberg, 2009. Springer-Verlag.

[18] M.C.F. Donkers, W.P.M.H. Heemels, N. van de Wouw, L. Hetel, and M. Steinbuch. Stability analysis of networked control systems using a switched linear systems approach. *Submitted for journal publication*.

[19] F. E. Felicioni and S. J. Junco. A Lie algebraic approach to design of stable feedback control systems with varying sampling rate. In *Proceedings of the 17th IFAC World Congress*, pages 4881–4886, Seoul, Korea, July 2008.

[20] E. Fridman, A. Seuret, and J.P. Richard. Robust sampled-data stabilization of linear systems: an input delay approach. *Automatica*, 40:1441–1446, 2004.

[21] E. Fridman and U. Shaked. Stability and guaranteed cost control of uncertain discrete delay systems. *International Journal of Control*, 78(4):235–246, 2005.

[22] H. Fujioka. Stability analysis for a class of networked / embedded control systems: Output feedback case. In *Proceedings of the 17th IFAC World Congress*, pages 4210–4215, Seoul, Korea, July 2008.

[23] H. Gao, T. Chen, and J. Lam. A new delay system approach to network-based control. *Automatica*, 44(1):39–52, 2008.

[24] M. García-Rivera and A. Barreiro. Analysis of networked control systems with drops and variable delays. *Automatica*, 43(12):2054–2059, 2007.

[25] D. Henrion, J. Lofberg, M. Kocvara, and M. Stingl. Solving polynomial static output feedback problems with PENBMI. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pages 7581–7586, Dec. 2005.

[26] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. *Proc. of the IEEE*, 95(1):138–162, 2007.

[27] J.P. Hespanha, P. Naghshtabrizi, and Yonggang Xu. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162, Jan. 2007.

[28] L. Hetel, J. Daafouz, and C. Iung. Stabilization of arbitrary switched linear systems with unknown time-varying delays. *IEEE Transactions on Automatic Control*, 51(10):1668 – 1674, 2006.

[29] L. Hetel, J. Daafouz, and C. Iung. Analysis and control of LTI and switched systems in digital loops via an event-based modeling. *International Journal of Control*, 81(7):1125 1138, 2008.

[30] L. Hetel, J. Daafouz, and C. Iung. Equivalence between the Lyapunov-Krasovskii functionals approach for discrete delay systems and that of the stability conditions for switched systems. *Nonlinear Analysis: Hybrid Systems*, 2(3):697–705, 2008.

[31] R. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[32] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASME Journal of Basic Engineering*, 82, 1960.

[33] C.-Y. Kao and B. Lincoln. Simple stability criteria for systems with time-varying delays. *Automatica*, 40(8):1429–1434, 2004.

[34] C. M. Kellett and A. R. Teel. On the robustness of $\mathcal{KL}$-stability for difference inclusions: Smooth discrete-time lyapunov functions. *SIAM Journal on Control and Optimization*, 44, 2005.

[35] E. Kofman and J. H. Braslavsky. Level crossing sampling in feedback stabilization under data-rate constraints. In *IEEE Conference on Decision and Control*, 2006.

[36] M. Lazar and W. P. M. H. Heemels. Optimized input-to-state stabilization of discrete-time nonlinear systems with bounded inputs. In *American Control Conference*, 2008.

[37] H. L. Lebesque. *Integrale, longueur, aire - Ph.D. dissertation*. University of Nancy, 1902.

[38] D. Lehmann and J. Lunze. Event-based control: A state-feedback approach. In *European Control Conference*, 2009.

[39] D. Liberzon. *Switching Systems and Control*. Birkhäuser, 2003.

[40] H Lin and P J Antsaklis. Stability and persistent disturbance attenuation properties for a class of networked control systems: switched system approach. *International Journal of Control*, 78(18):1447–1458, 2005.

[41] M. Stingl. M. Kočvara. PENBMI Version 2.1. 2004.

[42] P. Marti M. Velasco and C. Lozoya. On the Timing of Discrete Events in Event-Driven Control Systems. In *Hybrid Systems: Computation and Control*, volume 44981, 2008.

[43] L. Mirkin. Some remarks on the use of time-varying delay to model sample-and-hold circuits. *IEEE Transactions on Automatic Control*, 52(6):1109–1112, June 2007.

[44] M. Miskowicz. Send-on-delta concept: An event-based data-reporting strategy. In *Sensors*, 2006.

[45] M. Miskowicz. Asymptotic Effectiveness of the Event-Based Sampling according to the Integral Criterion. In *Sensors*, 2007.

[46] L. A. Montestruque and P. Antsaklis. Stability of model-based networked control systems with time-varying transmission times. *IEEE Transactions on Automatic Control*, 49(9):1562–1572, 2004.

[47] P. Naghshtabrizi and J. P. Hespanha. Stability of network control systems with variable sampling and delays. In Andrew Singer and Christoforos Hadjicostis, editors, *Proc. of the Forty-Fourth Annual Allerton Conference on Communication, Control, and Computing*, September 2006.

[48] P. Naghshtabrizi and A.R. Teel J.P. Hespanha. Exponential stability of impulsive systems with application to uncertain sampled-data systems. *Systems & Control Letters*, 57(5):378–385, 2008.

[49] D Nešić and A R Teel. Input-to-state stability of networked control systems. *Automatica*, 40(12):2121–2128, 2004.

[50] D. Nešić, A. R. Teel, and E. D. Sontag. Formulas relating KL stability estimates of discrete-time and sampled-data nonlinear systems. *Systems & Control Letters*, 38(1):49 – 60, 1999.

[51] G. Nikolakopoulos, A. Tzes, and I. Koutroulis. Development and experimental verifcation of a mobile client-centric networked controlled system. *European Journal of Control*, 11, 2005.

[52] J. Nilsson. *Real-Time Control Systems with Delays*. PhD thesis, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1998.

[53] D. Pachner and V. Havlena. An approach to out-of-sequence measurements in feedback control systems. In *Proceedings of 6th IEEE International Workshop on Factory Communication Systems*, 2008.

[54] Y-J. Pan, H. J. Marquez, and T. Chen. Stabilization of remote control systems with unknown time varying delays by LMI techniques. *International Journal of Control*, 79(7):752–763, 2006.

[55] G. C. Goodwin S. W. Chan and K. S. Sin. Convergence Properties of the Riccati Difference Equation in Optimal Filtering of Nonstabilizable Systems. *IEEE Transactions on Automatic Control*, 29, 1984.

[56] M. Sahebsara, T. Chen, and S. Shah. Optimal h2 filtering in networked control systems with multiple packet dropout. *IEEE Transactions on Automatic Control*, 52, 2007.

[57] A. Sala. Computer control under time-varying sampling period: An LMI gridding approach. *Automatica*, 41(12):2077–2082, 2005.

[58] P. Seiler and R. Sengupta. An $\mathcal{H}_\infty$ approach to networked control. *IEEE Transactions on Automatic Control*, 50(3):356–364, 2005.

[59] J. Sijs and M. Lazar. *Event based state estimation with time synchronous updates*. Eindhoven Universtiy of Technology.

[60] J. Sijs and M. Lazar. On event-based state estimation. In *Hybrid Systems: Computation and Control ser. Lecture Notes in Computer Science*, volume 5469, 2009.

[61] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry. Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49(9):1453–1464, 2004.

[62] H. W. Sorenson and D. L. Alspach. Recursive Bayesian estimation using Gaussian sums. *Automatica*, 7, 1971.

[63] E. Johannesson T. Henningsson and A. Cervin. Sporadic event-based control of first-order linear stochastic systems. *Automatica*, 44(11), 2008.

[64] P. Tabuada. Event-Triggered Real-Time Scheduling for Stabilizing Control Tasks. *IEEE Transactions on Automatic Control*, 52, 2007.

[65] Y. Tipsuwan and M.-Y. Chow. Control methodologies in networked control systems. *Control Engineering Practice*, 11(10):1099–1111, 2003.

[66] D.D. Šiljak. *Decentralized control of complex systems*. Academic Press, Boston, 1991.

[67] N. van de Wouw, P. Naghshtabrizi, M. Cloosterman, and J. P. Hespanha. Tracking control for networked control systems. In *Proc. of the 46th IEEE Conference on Decision and Control*, pages 4441–4446, New Orleans, LA, USA, December 2007.

[68] N. van de Wouw, P. Naghshtabrizi, M. Cloosterman, and J. P. Hespanha. Tracking control for sampled-data systems with uncertain time-varying sampling intervals and delays. *International Journal of Robust and Nonlinear Control*, 20(4), DOI: 10.1002/rnc.1433, 2010.

[69] J.J.C. van Schendel, M.C.F. Donkers, W.P.M.H. Heemels, and N. van de Wouw. On dropout modelling for stability analysis of networked control systems. In *Proc. American Control Conference*, Baltimore, Maryland, U.S.A., 2010.

[70] M. Verhaegen and P. Van Dooren. Numerical aspects of different kalman filter implementations. *IEEE Transactions of Automatic Control*, 31, 1986.

[71] A. van Zijl P. P. J. van den Bosch S. Weiland W. H. A. Hendrix W. P. M. H. Heemels, R. J. A. Gorter and M. R. Vonder. Asycnhronous measurement and control: A case study on motor synchronization. In *Control Engineering Practice*, volume 7, 1999.

[72] J. H. Sandee W. P. M. H. Heemels and P. P. J. van den Bosch. Analysis of event-driven controllers for linear systems. *International Journal of Control*, 81(4), 2008.

[73] G.C. Walsh and H. Ye. Scheduling of networked control systems. *Control Systems Magazine, IEEE*, 21(1):57–65, Feb 2001.

[74] B. Wang, G. Guo, and H. Jin. State estimation problem with communication constraints in NCSs. In *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on*, pages 2643–2648, Sept. 2008.

[75] X. Wang and M. Lemmon. Self-triggered Feedback Control Systems with Finite-Gain L_2 Stability. *IEEE Transactions on Automatic Control*, 45, 2009.

[76] Y. Wang and G. Yang. Multiple communication channels-based packet dropout compensation for networked control systems. *IET Control Theory and Applications*, 2:717–727, 2008.

[77] J. Wu and T. Chen. Design of networked control systems with packet dropouts. *Automatic Control, IEEE Transactions on*, 52(7):1314–1319, July 2007.

[78] L. Xiao, A. Hassibi, and J. How. Control with random communication delays via a discrete-time jump system approach. In *In Proceedings of 2000 American Control Conference*, pages 1148–1153, Chicago, IL,USA, 2000.

[79] D. Xie, N. Xu, and X. Chen. Stabilisability and observer-based switched control design for switched linear systems. *Control Theory & Applications, IET*, 2(3):192–199, March 2008.

[80] G. Xie and L. Wang. Stabilization of networked control systems with time-varying network-induced delay. In *Proc. of the 43rd IEEE Conference on Decision and Control*, pages 3551–3556, Atlantis, Paradise Island, Bahamas, December 2004.

[81] H.-J. Yoo and O.-K. Kwon. Networked control systems design with time varying delays. In *Proc. of 2005 IFAC World Congress*, Prague, July 2005.

[82] M. Yu, L. Wang, and T. Chu. Sampled-data stabilization of networked control systems with nonlinearity. *IEE Proc.-Control Theory Appl.*, 152(6):609–614, 2005.

[83] D. Yue, Q-L. Han, and C. Peng. State feedback controller design of networked control systems. *IEEE Transactions on circuits and systems*, 51(11):640–644, 2004.

[84] L. Zhang, Y. Shi, T. Chen, and B. Huang. A new method for stabilization of networked control systems with random delays. *IEEE Transactions on Automatic Control*, 50(8):1177–1181, 2005.

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

[86] W. Zhang, M.S. Branicky, and S.M. Phillips. Stability of networked control systems. *IEEE Control Systems Magazine*, 21:84–99, 2001.

[87] S. Wang and E. J. Davison, “On the stabilization of decentralized control systems,” *IEEE Trans. Automatic Control*, vol. 18, no. 5, pp. 473–478, 1973.

[88] R. D’Andrea, “A linear matrix inequality approach to decentralized control of distributed parameter systems,” in *Proc. American Contr. Conf.*, 1998, pp. 1350–1354.

[89] M. Rotkowitz and S. Lall, “A characterization of convex problems in decentralized control,” *IEEE Trans. Automatic Control*, vol. 51, no. 2, pp. 1984–1996, 2006.

[90] B. Bamieh, F. Paganini, and M. A. Dahleh, “Distributed control of spatially invariant systems,” *IEEE Trans. Automatic Control*, vol. 47, no. 7, pp. 1091–1107, 2002.

[91] S. Samar, S. Boyd, and D. Gorinevsky, “Distributed estimation via dual decomposition,” in *Proc. European Control Conf.*, Kos, Greece, 2007, pp. 1511–1516.

[92] L. Xiao, A. Hassibi, and J. How, “Control with random communication delays via a discrete-time jump system approach,” in *Proc. American Contr. Conf.*, Chicago, IL, 2000, pp. 2199–2204.

[93] M. Donkers, W. Heemels, D. Bernardini, and A. Bemporad, “Stability analysis of stochastic networked control systems,” in *Proc. American Contr. Conf.*, Baltimore, MD, 2010.

[94] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA: SIAM, 1994.

[95] C. Crusius and A. Trofino, “Sufficient LMI Conditions for Output Feedback Control Problems,” *IEEE Trans. Automatic Control*, vol. 44, no. 5, pp. 1053–1057, 1999.

[96] M. Kothare, V. Balakrishnan, and M. Morari, “Robust constrained model predictive control using linear matrix inequalities,” *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.

[97] X. Yu, J. Modestino, and X. Tian, “The accuracy of Gilbert models in predicting packet-loss statistics for a single-multiplexer network model,” in *INFOCOM. 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 4, 2005, pp. 2602–2612.

[98] D. Barcelli and A. Bemporad, “Decentralized model predictive control of dynamically-coupled linear systems: Tracking under packet loss,” in *1st IFAC Workshop on Estimation and Control of Networked Systems*, Venice, Italy, 2009, pp. 204–209.

[99] S. D. Cairano, K. Johansson, A. Bemporad, and R. Murray, “Dynamic network state estimation in networked control systems,” in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, M. Egerstedt and B. Mishra, Eds. Berlin Heidelberg: Springer-Verlag, 2008, no. 4981, pp. 144–157.

[100] A. Alessio and A. Bemporad. Decentralized model predictive control of constrained linear systems. In *Proc. European Control Conf.*, pages 2813–2818, Kos, Greece, 2007.

[101] A. Alessio and A. Bemporad. Stability conditions for decentralized model predictive control under packet dropout. In *Proc. American Contr. Conf.*, pages 3577–3582, Seattle, WA, 2008.

[102] A. Bemporad. *Hybrid Toolbox – User’s Guide*. December 2003. http://www.dii.unisi.it/hybrid/toolbox.

[103] A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[104] E. Camponogara, D. Jia, B.H. Krogh, and S. Talukdar. Distributed model predictive control. *IEEE Control Systems Magazine*, pages 44–52, February 2002.

[105] A. Damoiseaux, A. Jokic, M. Lazar, P.P.J. van den Bosch, I.A. Hiskens, A. Alessio, and A. Bemporad. Assessment of decentralized model predictive control techniques for power networks. In *16th Power Systems Computation Conference*, Glasgow, Scotland, 2008.

[106] W.B. Dunbar and R.M. Murray. Distributed receding horizon control with application to multi-vehicle formation stabilization. *Automatica*, 42(4):549–558, 2006.

[107] Z.-P. Jiang and Y. Wang. Input-to-state stability for discrete-time nonlinear systems. *Automatica*, 37(6):857–869, 2001.

[108] B. Johansson, T. Keviczky, M. Johansson, and K.H. Johansson. Subgradient methods and consensus algorithms for solving convex optimization problems. In *Proc. 47th IEEE Conf. on Decision and Control*, pages 4185–4190, Cancun, Mexico, 2008.

[109] T. Keviczky, F. Borrelli, and G.J. Balas. Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica*, 42(12):2105–2115, 2006.

[110] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, June 2000.

[111] N.R. Sandell, P. Varaiya, M. Athans, and M.G. Safonov. Survey of decentralized control methods for large scale systems. *IEEE Trans. Automatic Control*, 23(2):108–128, 1978.

[112] A.N. Venkat, J.B. Rawlings, and J.S. Wright. Stability and optimality of distributed model predictive control. In *Proc. 44th IEEE Conf. on Decision and Control and European Control Conf.*, Seville, Spain, 2005.

[113] A. Willig and R. Mitschke. Results of bit error measurements with sensor nodes and casuistic consequences for design of energy-efficient error control schemes. In *Proc. 3rd European Workshop on Wireless Sensor Networks (EWSN)*, volume 3868 of *Lecture Notes in Computer Science*, pages 310–325. Springer-Verlag, 2006.

[114] D. Bernardini and A. Bemporad. Energy-Aware Robust Model Predictive Control Based On Wireless Sensor Feedback. In *Proc. 47th IEEE Conf. on Decision and Control*, Cancun, Mexico, 2008.

[115] D. Bernardini and A. Bemporad. Energy-Aware Robust Model Predictive Control with Feedback from Multiple Noisy Wireless Sensors. In *Proc. European Control Conf.*, Budapest, Hungary, 2009.

[116] Moteiv Corporation. Tmote Sky: Datasheet, 2006.

[117] L.M. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *20th Annual Joint Conference of the IEEE Computer and Communications Societies*, 3:1548–1557, 2001.

[118] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999.

[119] G. Zhang, X. Chen, and T. Chen. A model predictive control approach to networked systems. In *Proc. of the 46th IEEE Conference on Decision and Control*, New Orleans, Louisiana, USA, 2007.

[120] D. Quevedo, A. Ahlén, and G.C. Goodwin. Predictive Power Control of Wireless Sensor Networks for Closed Loop Control. In *Proc. Int. Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control*, Pavia, Italy, 2008.

[121] M. Walsh and M. Hayes. A Robust Throughput Rate Control Mechanism for an 802.15.4 Wireless Sensor Network - An Anti-Windup Approach. In *Proc. American Control Conference*, pages 3071–3076, 2007.

[122] V. Raghunathan, C.L. Pereira, M.B. Srivastava, and R.K. Gupta. Energy Aware Wireless Systems with Adaptive Power-Fidelity Tradeoffs. *IEEE Transactions On Very Large Scale Integration (VLSI) Systems*, 13(2), 2005.

[123] V. Rajendran, K. Obraczka, and J.J. Garcia-Luna-Aceves. Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks. *Wireless Networks*, 12(1):6378, 2006.

[124] C.E. Jones, K.M. Sivalingam, P. Agrawal, and J.C. Chen. A Survey of Energy Efficient Network Protocols for Wireless Networks. *Wireless Networks*, 7(4):343–358, 2004.

[125] C. Schurgers. Energy-aware Wireless Communications. PhD thesis, University of California, Los Angeles, 2002.

[126] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava. Energy aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 19:40–50, 2002.

[127] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S. Sastry. Distributed control applications within sensor networks. In *Proc. of the IEEE Special Issue on Distributed Sensor Networks*, volume 91, pages 1235–1246, 2003.

[128] P. Antsaklis and J. Baillieul. Special Issue on Networked Control Systems. *IEEE Trans. Automatic Control*, 49(9), 2004.

[129] P.R. Kumar. New technological vistas for system and control: The example of wireless networks. *IEEE Control Systems Magazine*, 21:24–37, 2001.

[130] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. *IEEE Transactions on Mobile Computing*, 3(3):325–349, 2005.

[131] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

[132] F. Blanchini. Set invariance in control – a survey. *Automatica*, 35(11):1747– 1768, 1999.

[133] I.V. Kolmanovsky and E.G. Gilbert. Theory and computation of disturbance invariant sets for discrete-time linear systems. *Mathematical Problems in Engineering*, 4(4):317–367, 1998.

[134] P.O.M. Scokaert and D.Q. Mayne. Min-max feedback model predictive control for constrained linear systems. *IEEE Trans. Automatic Control*, 43:1136–1142, 1998.

[135] J.B. Rawlings. Tutorial Overview of Model Predictive Control. *IEEE Control Systems Magazine*, pages 38–52, 2000.

[136] J.M. Maciejowski. Predictive Control with Constraints. Prentice Hall, Harlow, UK, 2002.

[137] S.J. Qin and T.A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, 2003.

[138] E.F. Camacho and C. Bordons. Model Predictive Control. *Advanced Textbooks in Control and Signal Processing*. Springer-Verlag, London, 2nd edition, 2004.

[139] A. Alessio and A. Bemporad. A survey on explicit model predictive control. In *Proc. Int. Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control*, Pavia, Italy, 2008.

[140] E.C. Kerrigan and J.M. Maciejowski. Robustly stable feedback min-max model predictive control. In *Proc. of the American Control Conference*, volume 4, pages 3490–3495, Denver, Colorado, USA, 2003.

[141] A. Bemporad, F. Borrelli, and M. Morari. Min-max control of constrained uncertain discrete-time linear systems. *IEEE Trans. Automatic Control*, 48, 2003.

[142] S. Trimboli, S. Di Cairano, A. Bemporad, and I.V. Kolmanovsky. Model predictive control for automotive time-delay processes: An application to air-to-fuel ratio control. In *Proc. 8th IFAC Workshop on time delay systems*, Sinaia, Romania, 2009.

[143] A. Bemporad. Predictive control of teleoperated constrained systems with unbounded communication delays. In *Proc. 37th IEEE Conf. on Decision and Control*, pages 2133–2138, Tampa, FL, 1998.

[144] M. Milanese and A. Vicino. Optimal estimation theory for dynamic systems with set membership uncertainty: an overview. *Automatica*, 27(6):997–1009, 1991.

[145] L. Chisci, A. Garulli, and G. Zappa. Recursive state bounding by parallelotopes. *Automatica*, 32:1049–1055, 1996.

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

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

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

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

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

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

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

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

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

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

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

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

[158] P. Seiler, and R. Sengupta. An H_∞ approach to networked control. *IEEE Trans. Autom. Control*, 356– 364, 2005.

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