Receding horizon control for constrained networked systems subject to data-losses

G. Franzè, F. Tedesco and D. Famularo
DIMES - Università degli Studi della Calabria, Via Pietro Bucci, Cubo 42C, Rende (CS), 87036, ITALY
{franze, ftedesco, famularo}@dimes.unical.it

Abstract

The paper addresses the stabilization problem for a constrained control system in which both plant measurements and command signals in the loop are sent through communication channels subject to time-varying delays and data-losses (Networked Control System). A novel receding horizon strategy is proposed by resorting to an uncertain polytopic linear plant framework. Sequences of pre-computed inner approximations of the one-step controllable sets are on-line exploited as target sets for the actual state prediction to compute the commands to be applied to the plant in a receding horizon fashion. The communication channel effects are taken into account by resorting to both Independent-of-Delay and Delay Dependent stability concepts that are used to initialize the one-step controllable sequences. The resulting framework guarantees Uniformly Ultimate Boundedness and constraints fulfillment regardless of plant uncertainties and data-loss occurrences.

1 Introduction

A significant part of current engineering applications can be viewed as the interconnection of a set of devices endowed with sensing and communication capabilities. Technological advances are in fact delivering “gadgets” which can be ubiquitously embedded in the physical world. From a control engineer perspective such plant categories are called as networked control systems (NCS) consisting of numerous physical and computing elements called agents, which have interactions and dependencies, supported by overlapping network resources. From an abstract point of view an NCS can be regarded as a system comprised of the plant to be regulated and of actuators, sensors, and controllers, coordinated via a communication channel. Due to these features the study of stability analysis and control design of NCSs is attracting considerable attention in the technical literature. Examples can be found in e.g. DC motors, robots, automobiles, etc. as described in [13, 26] and references therein.

Time-delay systems [23, 29] are one of the starting points for analyzing the delay effect in the NCS framework. However an NCS is different from a traditional time-delay system where the delay
is simply assumed to be constant or bounded and data transmission between system components is performed with arbitrary high accuracy. A network-induced latency is instead variable or even unbounded making the analysis and control design a challenging task. Also, quantization errors in the transmitted signals, packet dropouts, non-constant transmission/sampling instants and non-dedicated communication channels are additional ”features” making significantly different an NCS from a standard time-delay system. Recent contributions and tutorials on the analysis and modeling of NCSs have been conducted using:

- discrete-time modeling approaches [6] making sense when linear plant/controller components are exactly discretized between two transmission instants. This gives rise to multi-model approximations where newly uncertain affine parameters are introduced to take into account time-delay and transmission errors occurrences;

- sampled data approaches [10] where NCS dynamics are described by means of continuous time models and stability are analyzed by means of functional differential equations via Lyapunov-Krasovskii functionals;

- continuous time modeling approaches [12] including all the previously cited NCS defects via an hybrid system paradigm.

In literature, relevant results on feedback control strategies for NCS exploit several approaches: traditional Proportional-Integral control structures designed for time-delay systems [2], Lyapunov-based control strategies for nonlinear multivariable systems [28] and state/observer based feedback control methodologies have also been studied for a linear continuous time NCS [18]. In [16] the issues of stabilization and state estimation for limited capacity channels network is approached via a remote control setup, where the sensor signals are transmitted to the controller/observer via a noisy discrete memoryless channel (DMC). A linear time invariant model affected by additive stochastic disturbances, which are independent of the channel noise is considered and it is proved that the trajectory-wise stability or observability objective may be obtained for rather special channels. Finally, recent contributions point out their attention to a stochastic description of time-varying transmission delays, see e.g. [7, 30] and references therein.

Of interest here are constrained Receding Horizon Control strategies which are an extremely appealing methodology for NCS stabilization due to their intrinsic capability to generate, at each time instant, a sequence of virtual inputs which can be transmitted within a single data packet [22, 25]. Noticeable contributions are from [19, 11, 21]. In [19], the authors consider a receding horizon strategy for nonlinear networked systems under wireless and asynchronous measurement sampling. In order to regulate the state of the system towards an equilibrium point while minimizing a given performance index, a Lyapunov-based model predictive controller is designed by explicitly taking into account data losses, both in the optimization problem formulation and in the controller implementation. The
proposed scheme allows an explicit characterization of the stability region and guarantees that such a set is invariant for the closed-loop system under data-losses if the maximum time, in which the loop is open, is shorter than a given constant value depending on the system parameters and on the Lyapunov-based controller. Gupta and Quevedo [11] extend a control scheme for nonlinear plants, popular in real-time systems to tolerate the presence of time-varying processing resources (such as variable delays, packet losses/drops etc.), known as anytime algorithm providing a solution even with limited processing resources, and refining the solution as more resources become available. Two anytime control algorithms based on a stochastic LQR performance index are proposed, providing better performance as more processing time is available. The basic idea is to exploit the availability of extra processing time to determine tentative future control inputs. In the authors’ words, even if this approach resembles a receding horizon control scheme the proposed algorithms do not solve a sequence of optimization problems of increasing size. The control inputs are instead derived sequentially by eventually recycling the already computed commands for the next cycle. In [21], following the same ideas as [19], a nonlinear RHC scheme exploiting a Network Delay Compensation strategy is proposed to efficiently manage the simultaneous presence of constraints, model uncertainties, time-varying transmission delays, and data-packet losses. The main aim consists in overcoming the inherent difficulties related to a MPC design based on linear process models and the joint presence of constraints, nonlinearities and NCS features. constraint-tightening nonlinear MPC issues.

We will focus here on a novel discrete time receding horizon strategy for NCSs, described by means of uncertain multi-model linear systems, under the occurrence of time-varying delays, data loss on the sensor-to-controller link and feedback command loss on the controller-to-actuator link. Data loss events on the sensor side will be considered by comparing the delay with respect to the maximum allowable transmission interval (MATI) of the channel whereas feedback loss events on the actuator side will involve comparison of the related delay with respect to the chosen sampling time. These two events are separately accounted to make available an “usable” control move for the actuator logic within each sampling interval. The NCS stabilization problem will involve a dual-mode predictive strategy:

- first, a pair of stabilizing state feedback laws “capable” to efficiently manage normal and data-loss phases on actuator-sensor sides, are off-line derived by resorting to Independent-of-Delay (IOD) and Delay Dependent (DD) stability concepts. The objective consists in enlarging, as best as possible, the sets of initial states that, according to the network delay configuration, can be steered to the target in a finite number of steps;

- second, at each sample time, an on-line receding horizon strategy is obtained by deriving the smallest ellipsoidal set (DD or IOD type) compliant with the delay scenario. The input move is computed by minimizing a given running cost under the requirement that the one-step ahead state prediction belongs to its successor (largest ellipsoidal region contained inside the current
In order to deal with this idea, the key ingredients are:

1. the one-step controllable ellipsoidal regions must be capable to properly address time-delays occurrences. Up to our best acknowledge, this aspect in previous technical contributions is lacking while here is formally defined and characterized. This result represents the core of the overall strategy because is crucial for ensuring feasibility retention and uniformly ultimate boundedness with respect to unreliable communication channels and constraints fulfilment;

2. a time-stamp protocol necessary to give a measure of the current time delay is considered.

A second merit of the proposed scheme relies on the computational resources (CPU power, memory resources and bandwidth requirements) that are significantly modest when contrasted with competitor schemes. In fact, on the controller side the command input computation prescribes at most the solution of a Quadratic Programming (QP) problem under linear constraint, while the length of the buffer units (see Fig. 1) is limited to store a single input and a single state measurement.

The benefits of the proposed strategy are illustrated by means of a path-tracking problem for a two driving wheels mobile autonomous robot. By considering several network delay configurations, extensive comparisons with the recent method of Pin and Parisini [21] are provided both in terms of achieved performance and memory requirements.

![Networked control system structure](image)

**Figure 1:** Networked control system structure

**Notations**

Given a set $S \subseteq \mathbb{R}^n$, $In[S] \subseteq S$ denotes its inner ellipsoidal approximation.

Given a set $S \subseteq X \times Y \subseteq \mathbb{R}^n \times \mathbb{R}^m$, the projection of the set $S$ onto $X$ is defined as $\text{Proj}_X(S) := \{x \in X \mid \exists y \in Y \text{ s.t. } (x, y) \in S\}$. 
2 Problem formulation

In what follows we will refer to the networked control scheme depicted in Fig. 1 where delay effects are taken into consideration from the sensor and actuator sides. It consists of the plant, the actuator/sensor devices, the controller and an actuator logic both equipped with buffering units.

Specifically:

- **Process** - It is described by a multi-model discrete-time linear system

  \[
  x_p(t + 1) = \Phi(\alpha(t))x_p(t) + G(\alpha(t))u(t)
  \]  

  where \( t \in \mathbb{Z}_+ := \{0, 1, \ldots\} \), \( x_p(t) \in \mathbb{R}^n \) denotes the state plant and \( u(t) \in \mathbb{R}^m \) the control input. The parameter vector \( \alpha(t) \in \mathbb{R}^l \) is assumed to lie in the unit simplex

  \[
  \mathcal{P}_l := \left\{ \alpha \in \mathbb{R}^l : \sum_{i=1}^l \alpha_i = 1, \alpha_i \geq 0 \right\}
  \]  

  and the system matrices \( \Phi(\alpha) \) and \( G(\alpha) \) belong to

  \[
  \Sigma(\mathcal{P}_l) := \left\{ (\Phi(\alpha), G(\alpha)) = \sum_{i=1}^l \alpha_i (\Phi_i, G_i), \alpha \in \mathcal{P}_l \right\}
  \]  

  where the pairs \( (\Phi_i, G_i) \) denote the polytope vertices \( \Sigma(\mathcal{P}_l) \), viz. \( (\Phi_i, G_i) \in \text{vert}\{\Sigma(\mathcal{P}_l)\} \), \( \forall i \in \mathbb{I} := \{1, 2, \ldots, l\} \). Moreover, the control input is subject to the following saturation constraints

  \[
  u(t) \in \mathcal{U}, \forall t \geq 0, \quad \mathcal{U} := \{ u \in \mathbb{R}^m \mid u^T u \leq \bar{u} \},
  \]  

  with \( \bar{u} > 0 \) and \( \mathcal{U} \) a compact subset of \( \mathbb{R}^m \) containing the origin as an interior point.

- **Actuator and Controller buffers** - The actuator buffer is in charge to store the last received command, hereafter denote as \( u^R_{t-1} \), whereas the buffering unit on the controller side stores the last state measurement received from the sensor-to-controller channel, named \( x^R_{t-1} \), and the last computed command \( u^C_{t-1} \);

- **Actuator logic** - The actuator tracks data-losses on the feedback channel. In particular, as specified in Section IV, at each time instant \( t \) such a logic is instructed to apply the command \( u(t) \) if available or conversely \( u^R_{t-1} \).

Throughout the paper, the following definitions will be used.

**Definition 1** A set \( \mathcal{T} \subseteq \mathbb{R}^n \) is robustly positively invariant for (1) if once the state \( x_p(t) \) enters that set at any given time \( t_0 \), it remains in it for all future instants, i.e. \( x_p(t_0) \in \mathcal{T} \rightarrow x_p(t) \in \mathcal{T}, \forall t \geq t_0 \).

Given the plant (1) it is possible in principle to compute the sets of states \( i \)-step controllable to \( \mathcal{T} \) via the following recursion (see [4]):

\[
\mathcal{T}_i := \{ x_p : \exists u \in \mathcal{U} : \forall \alpha \in \mathcal{P}_l, \Phi(\alpha)x_p + G(\alpha)u \in \mathcal{T}_{i-1} \}
\]  

where \( \mathcal{T}_i \) is the set of states that can be steered into \( \mathcal{T}_{i-1} \) using a single move with a causal control. By induction we have that \( \mathcal{T}_i \) is the set of states that can be steered into \( \mathcal{T} \) in at most \( i \) control moves.
Definition 2 ([4]) Let $S$ be a neighbourhood of the origin. Given the plant (1), the trajectory is said to be Uniformly Ultimate Bounded in $S$ if there exists $u(t) \in \mathcal{U}$ such that for all $\mu > 0$ there exists $T(\mu) > 0$ such that, for every $\|x_p(0)\| \leq \mu$, $x_p(t) \in S$ for all $t \geq T(\mu)$ and for all $\alpha(t) \in \mathcal{P}$.

To properly treat data-loss both on the plant-controller and controller-plant links, the sensor-to-controller and the controller-to-actuator cases need to be separately analyzed. We will suppose first that the delay on the command channel side $\tau_{ca}(t) : \mathbb{Z}_{\geq 0} \rightarrow \mathbb{Z}_{\geq 0}$ is such that $\tau_{ca}(t) \leq \bar{\tau}_c$, $\forall t$, while the delay on the measurement side $\tau_{sc}(t) : \mathbb{Z}_{\geq 0} \rightarrow \mathbb{Z}_{\geq 0}$ could be unbounded. On the contrary, the controller-to-actuator link (the feedback) is only subject to the actual induced delay $\tau_{ca}(t)$. Moreover, at each time instant $t$ we shall define respectively with $\tau_m(t) \leq \tau_{ca}(t)$ and $\tau_c(t) \leq \tau_{sc}(t)$ the age of the state measurement used by the controller to compute the input and the age of the command used by the actuator.

Then, on the plant-controller link at each time instant $t$ when computing the input $u(t)$, the following age cumulative network latency should in principle be used: $\tau_{NL}(t) = \tau_m(t) + \tau_c(t - \tau_m(t))$.

The delay management is therefore achieved by using a time-stamp mechanism that makes available at each time instant $t$ the sensor latency $\tau_m(t)$ and the actuator latency $\tau_c(t)$ on the controller and actuator sides, respectively. By time-stamping each data packet sent from the sensor and controller nodes, buffer units have to be considered both for actuator and controller where one of their functions is to compare the time stamps on the received control and measurement signals in order to evaluate $\tau_m(t)$ and $\tau_c(t)$, respectively. More specifically, the data-packet sent by the sensor to the controller contains the time stamp $t_x$ while the the data-packet sent by the controller includes the time stamp $t_u$ and the cumulative delay $\tau(t)$ of the state measurement on which the input has been computed. Based on these information the stored commands $u_{C_1}^C$ and $u_{R_1}^R$ will be updated if more recent than the existing ones via a comparison with the stored time stamps $t_{u_{C_1}}^C$ and $t_{u_{R_1}}^R$. The latter prescribes that a suitable clock synchronization must be assumed amongst actuator, sensor and controller units, see e.g. [24]. Then, the control strategy must be capable to ensure that data-packets sent by the controller are such that both the worst-case delay and sequences of data-losses are accommodated. Although this will give rise to an unavoidable degradation in the overall control performance, up to our best knowledge the resource requirements (memory, cpu, etc.) are significantly mitigated with respect to the existing literature.

The above developments allow to define the time-delay scenarios occurring on the communication channels:

- Sensor-to-Controller link
  - **Normal phase** - each time-delay occurrence is bounded, $\tau(t) < \bar{\tau}$ and no data-loss events occur. The upper bound $\bar{\tau}$ represents the MATI [28];
  - **Data loss** - there exists a time instant $\bar{t}$ such that $\tau(\bar{t}) \geq \bar{\tau}$; the state measurement will no longer available for the control action computation.
• Controller-to-Actuator link
  
  - **Normal phase** - at each time instant $t$ the actuator receives a control action $u(t)$;
  
  - **Feedback loss** - there exists a time instant $\bar{t}$ such that $\tau_c(\bar{t}) \geq 1$, no control action is available for feedback.

Then, the problem we want to solve can be stated as follows:

**Network Constrained Stabilization (NCS) problem** - Given the networked system depicted in Fig. 1 and the model plant (1)-(3), determine a state-feedback regulation strategy

$$u(\cdot) = g(x_p(\cdot))$$

(6)

complying with (4) which steers the state of the closed-loop system into a suitable terminal set regardless of time-delay events on the Sensor-to-Controller and Controller-to-Actuator communication channels.

To formally deal with the NCS problem prescriptions, a crucial question regards the round-trip delay $\tau_c(t - \tau_m(t))$ that as it is well-known cannot be available at the controller side. A way to overcome such a drawback is consider the upper bound $\bar{\tau}_c$ on the controller-actuator link during both the controller design and the command computation $u(\cdot)$ phases, i.e.

$$\tau(t) = \tau_m(t) + \bar{\tau}_c, \ \forall t.$$  

(7)

In the sequel, the problem will be addressed by adopting a dual-mode predictive approach:

• **Off-line** - A stabilizing state-feedback control law (6) for (1)-(3) is first derived by resorting to DD and IOD stability concepts. The operating region of the algorithm is then enlarged by computing sets of states that can be steered into the target set in a finite number of steps;

• **On-line** - At each time $t$ and given the delayed state measurement from (1), a receding horizon control strategy is obtained by looking at the “smallest” ellipsoidal set (DD or IOD regions) with respect to the delay time instance. The current input is finally determined by minimizing a running cost under the condition that the one-step state prediction belongs to the successor set (DD or IOD) [4], i.e.

$$J(x_p(t - \tau(t)), u) := \max_{\alpha \in \mathcal{P}_1} \|\Phi(\alpha)x_p(t - \tau(t)) + G(\alpha)u\|_P^2$$

with $P_i = P_i^T > 0$ the shaping matrix of the one-step controllable set $\mathcal{T}_i$. Notice that even if such ellipsoidal regions can be smaller than the time-delay free case [1], the capability to accommodate large time-delay occurrences at the expense of low computational demands is one of the main features to achieve when NCSs are considered.
3 Off-line phase

In this section the core of the proposed approach is outlined and discussed. Specifically, the conditions under which adequate control actions can be computed on the remote side and then correctly applied to the plant regardless of data-loss and/or feedback loss events are formally stated. Moreover, it is important to underline that at this stage the available information are the vertices of the polytopic structure (1)-(3) and the upper bounds $\bar{\tau}, \bar{\tau}_c$.

3.1 Derivation of DD and IOD robustly invariant terminal regions

Here, we derive the terminal regions according to the dual-mode predictive strategy complying with the DD and IOD time-delay scenarios and the prescribed constraints (4). To this end we consider a state-feedback control law

$$u(t) = K_{DD} x_p(t - \tau(t))$$

which satisfies the NCS problem requirements for the regulated plant

$$x_p(t + 1) = \Phi(\alpha(t)) x_p(t) + G(\alpha(t)) K_{DD} x_p(t - \tau(t))$$

We will exploit a standard technicality in delayed systems (see [10] and references therein). By introducing the auxiliary state $y_p(t) = x_p(t + 1) - x_p(t)$ a descriptor form of (1) can be obtained

$$\begin{cases}
x_p(t + 1) & = y_p(t) + x_p(t) \\
0 & = -y_p(t) + \Phi(\alpha(t)) x_p(t) - x_p(t) + G(\alpha(t)) K_{DD} x_p(t - \tau(t))
\end{cases}$$

By defining the augmented state $\bar{x}_p(t) = \begin{bmatrix} x_p^T(t) & y_p^T(t) \end{bmatrix}^T$, we have

$$E \bar{x}_p(t + 1) = A_{DD} \bar{x}_p(t) + B_{DD} \sum_{j=t-\tau(t)}^{t-1} y_p(j)$$

with $E = \text{diag}\{I, 0\}$,

$$A_{DD} = \begin{bmatrix} I & I \\
\Phi(\alpha(t)) - I + G(\alpha(t)) K_{DD} & -I
\end{bmatrix}, B_{DD} = \begin{bmatrix} 0 \\
-G(\alpha(t)) K_{DD}
\end{bmatrix}$$

and via to the following DD Lyapunov-Krasovskii functional

$$V(t) = \bar{x}_p^T(t) E P_{DD} E \bar{x}_p(t) + \sum_{m=-\tau_{\max}}^{t} \sum_{j=t+m}^{t-1} y_p^T(j) [R + Q] y_p(j)$$

$$P_{DD} = P_{DD}^T \geq 0, R = R^T \geq 0, Q = Q^T \geq 0$$

it can be proved that the DD feedback control law (8) robustly stabilizes the plant and satisfies the prescribed constraints if the following matrix inequalities in the objective variables $K_{DD}, P_{DD}, Q, R$ and $\tau_{\max} \leq \bar{\tau}$, evaluated over the plant vertices (3), are satisfied

$$\begin{bmatrix} E^T P_{DD} E - S_{DD} & 0 & A_{DD}^T P_{DD} \\
0 & \tau_{\max}(R + Q) & B_{DD}^T P_{DD} \\
P_{DD} A_{DD} & P_{DD} B_{DD} & P_{DD}
\end{bmatrix} \geq 0, j = 1, \ldots, l$$
$\begin{bmatrix}
\bar{u}^2 E^T P E & K_{DD}^T \\
K_{DD} & 0
\end{bmatrix} \geq 0 \quad (14)$

where $S_{DD} \triangleq \text{diag}\{0, \tau_{max} (R + Q)\}$ and

$A_{jDD} = \begin{bmatrix} I \\
\Phi_j - I + G_j K_{DD}
\end{bmatrix}$, $B_{jDD} = \begin{bmatrix} 0 \\
-G_j K_{DD}
\end{bmatrix}$, $j = 1, \ldots, l$.

As regards the IOD case, we search for a pair $(P_{IOD}, K_{IOD})$ complying with $\tau(t) \leq \bar{\tau}$. To this end, by using the IOD Lyapunov-Krasovskii functional

$V(t) = \bar{x}_p(t) E P_{IOD} E \bar{x}_p(t) + \sum_{j=t-\bar{\tau}}^{t-1} x_p^T(j) S x_p(j)$, \quad (15)

with $S = S^T \geq 0$, the constrained IOD feedback control law

$u(t) = K_{IOD} x_p(t - \tau(t)) \quad (16)$

stabilizes the plant

$x_p(t + 1) = \Phi(\alpha(t)) x_p(t) + G(\alpha(t)) K_{IOD} x_p(t - \tau(t)) \quad (17)$

if

$\begin{bmatrix}
E^T P_{IOD} E - S_{IOD} & A_{jIOD}^T P_{IOD} \\
S_{IOD} & A_{jIOD}
\end{bmatrix} = 0, \quad j = 1, \ldots, l \quad (18)$

$\begin{bmatrix}
\bar{u}^2 E^T P_{IOD} E & K_{IOD}^T \\
K_{IOD} & 0
\end{bmatrix} \geq 0 \quad (19)$

where $S_{IOD} = \text{diag}\{S, 0\}$, $A_{jIOD} = \begin{bmatrix} I \\
(\Phi_j - I + G_j K_{IOD}) - I
\end{bmatrix}$, $j = 1, \ldots, l$.

Hence, the ellipsoidal sets

$E_{DD} := \text{Proj}_{x} \{ \bar{x}_p \in \mathbb{R}^{2n} | \bar{x}_p^T P_{DD} E^T \bar{x}_p \leq 1 \} = \{ x_p \in \mathbb{R}^n | x_p^T Q_{DD} x_p \leq 1 \} \subset \mathbb{R}^n$, $E_{IOD} := \text{Proj}_{x} \{ \bar{x}_p \in \mathbb{R}^{2n} | \bar{x}_p^T P_{IOD} E^T \bar{x}_p \leq 1 \} = \{ x_p \in \mathbb{R}^n | x_p^T Q_{IOD} x_p \leq 1 \} \subset \mathbb{R}^n$

arising from the inequalities (13), (14) and (18), (19) are robust positively invariant regions for the closed-loop state evolutions (9), (17) complying with the input constraints (4), viz. $K_{DD} E_{DD} \subset U$, $K_{IOD} E_{IOD} \subset U$. Note that, when compared with the IOD stabilizing condition, a DD strategy is less conservative and exhibits a potentially better control performance, see [23].
From now on we will assume that there exists a DD pair \((K_{DD}, \mathcal{E}_{DD})\), with \(\mathcal{E}_{DD} \neq \emptyset\), and an IOD pair \((K_{IOD}, \mathcal{E}_{IOD})\), with \(\mathcal{E}_{IOD} \neq \emptyset\), with \(\tau_{\text{max}}\) a scalar obtained by solving (13), (14). Note that, due to the occurrence of “large” time-delays, the ellipsoidal sets \(\mathcal{E}_{IOD}, \mathcal{E}_{DD}\) sizes can be arbitrary small (even if no zero measure sets may occur) and poor related control performance of the associated gains \(K_{IOD}\) and \(K_{DD}\) arise. It is important to underline that if the pair \((K_{IOD}, \mathcal{E}_{IOD})\) cannot be computed for \(\tau(t) \leq \bar{\tau}\), the strategy validity is not altered because in such a case the MATI will be considered as the delay satisfying the requirements (18)-(19).

3.2 One-step ahead Ellipsoidal controllable sets

The aim here is to characterize all the states one-step controllable to a given target set \(\mathcal{T}\) which comply with the proposed NCS problem. To extend such a concept to the proposed framework, it is important to notice that the one-step state prediction needs to be evaluated along the model

\[
    x(t+1) = \Phi(\alpha(t))x(t) + G(\alpha(t))u(t)
\]

with \(u(t)\) chosen according to a delayed state plant information. The rationale behind the introduction of (20) relies on the necessity to properly take into account an unavoidable time misalignment existing between the measured plant state which is sent to the controller and the delayed state exploited instead by the controller unit to compute the command input to be sent to the plant (see Fig. 1). To better clarify, let us take a look to Fig. 2.

![Figure 2: Process/model discrepancy](image)

There, at the generic time instant \(t\), the model (20) has knowledge on the state measurement generated \(\tau(t)\) instants before, i.e. \(x(t) = x_p(t - \tau(t))\), while the process state is \(x_p(t)\). Since in general \(x(t) \neq x_p(t)\), there exists a discrepancy between the process (1) and the model (20) that, if not properly treated, can lead to erroneous input evaluations. There is in fact no guarantee that if \(x(t) \in \mathcal{T}_i\) the same holds true for \(x_p(t)\). A possible way to comply with the above reasoning is to determine the sequence of sets \(\mathcal{T}_i\) by resorting to the following one-step transition map, valid for \(0 \leq \tau(t) \leq \bar{\tau}\)

\[
    x(t+1) = \Phi(\alpha(t)x(t - \tau(t)) + G(\alpha(t))u(t)
\]

Note that the delayed state \(x(t - \tau(t))\) in (21) is instrumental to take care, at each instant \(t\), the difference between the state measurement \(x(t)\) and the real plant state \(x_p(t)\). By virtue of (21), if
\(x(t) \in T_t\) with \(x(t) \neq x_p(t)\), the same holds for \(x_p(t)\), and there exists a command \(u(t)\) that drives \(x(t + 1)\) into \(T_{t-1}\) for all \(\tau(t) \in [0, \tau]\). To recast such an idea into a computable scheme, explicit time-delay dependencies in the auxiliary model (21) need to be derived. Without loss of generality (21) can be re-written as

\[
x(t + 1) = \Phi(\alpha(t))x(t) + \Phi(\alpha(t))x(t - \tau(t)) + G(\alpha(t))u(t), \forall \tau(t) \in [1, \tau]
\]  

(22)

and by considering the auxiliary state \(y(t)\) the following descriptor form results

\[
\begin{bmatrix}
  x(t + 1) \\
  y(t)
\end{bmatrix} =
\begin{bmatrix}
  y(t) + x(t) \\
  -y(t) + \Phi(\alpha(t))x(t) + \Phi(\alpha(t))x(t - \tau(t)) + G(\alpha(t))u(t) - x(t)
\end{bmatrix}
\]

(23)

Hence, by noticing that \(x(t - \tau(t)) = x(t) - \sum_{j=t-\tau(t)}^{t-1} y(j)\) and by imposing the worst-case occurrence on the time-varying delay \(\tau(t)\), i.e. \(\tau(t) = \bar{\tau}, \forall t\), we have that

\[
\bar{E}\bar{x}(t + 1) = \Phi(\alpha(t))\bar{x}(t) + \bar{G}(\alpha)u(t) - \bar{G}_y(\alpha(t)) \sum_{j=t-\bar{\tau}}^{t-1} y(j)
\]

(24)

where \(\bar{x}(t) = [x(t)^T \ y(t)^T]^T\), \(\bar{E} = \text{diag}\{I, 0\}\), \(\Phi(\alpha(t)) = \begin{bmatrix} I & I \\ 2\Phi(\alpha(t)) - I & -I \end{bmatrix}\), \(\bar{G}(\alpha(t)) = \begin{bmatrix} 0 \\ G(\alpha(t)) \end{bmatrix}\)

and \(\bar{G}_y(\alpha(t)) = \begin{bmatrix} 0 \\ \Phi(\alpha(t)) \end{bmatrix}\).

In principle the computation of the set recursions (5) along the system dynamics (24) prescribes that all the auxiliary variables \(y(j) = x(j + 1) - x(j), j = t - \bar{\tau}, \ldots, t - 1\) have to be included in the augmented system. Unfortunately the obtained state space description has huge dimension and could be computationally intractable. In order to mitigate such a difficulty it is possible to achieve a less demanding solution at the price of an increasing level of conservativeness. The following proposition provides inner approximations of the exact one-step controllable sets related to the description (21).

**Proposition 1** Let \(T_0 \neq \emptyset\) be a given robustly invariant ellipsoidal region complying with the input constraints and \(x_{\text{aug}} = [x^T \ y^T \ z_1^T \ z_2^T]^T\) the augmented state describing the dynamics (24) with \(z_1, z_2 \in \mathbb{R}^n\) accounting for all the cumulative sum vectors \(y(j), j = t - \bar{\tau}, \ldots, t - 1\). Then, the ellipsoidal set sequence

\[
\begin{align*}
E_0 &= T_0 \\
E_i &= \text{Proj}_{x}\{In[x_{\text{aug}}] \in \mathbb{R}^{4n} \text{ with } z_1, z_2 \in E_{i-1} : \exists u \in U, \forall \alpha \in P_t, \text{Proj}_{x}\{\Phi(\alpha)_{\text{aug}}\bar{x}_{\text{aug}} + \bar{G}(\alpha)_{\text{aug}}u\} \in E_{i-1}\}
\end{align*}
\]

(25)

\[
\Phi(\alpha)_{\text{aug}} = \begin{bmatrix} I & I & 0 & 0 \\ 2\Phi(\alpha) - I & -I & -\bar{\tau}\Phi(\alpha) & -\bar{\tau}\Phi(\alpha) \end{bmatrix},
\]

\[
\bar{G}(\alpha)_{\text{aug}} = \begin{bmatrix} 0 & G^T(\alpha) & 0 & 0 \end{bmatrix}^T
\]
if non-empty, satisfies $E_i \subset T_i$.

**Proof** - Assume that the sequence of one-step controllable sets $T_i$ for the descriptor form (24) has been computed by resorting to the whole augmented state description $[x^T(t) y^T(t) y^T(t-1) \ldots y^T(t-\bar{\tau})]^T$.

Note that, because the auxiliary variables $y(\cdot)$ are linear combinations of $x(\cdot)$, at each recursion all the initial vectors $x(j)$, $j = t - \bar{\tau}, \ldots, t - 1$ must lie in $T_i$ in order to ensure that the one-step evolution of (21) belongs to $T_{i-1}$. In view of this reasoning, a way to deal with the one-step controllable sets computation is to impose that at each recursion $x(j) \in T_{i-1}$, $j = t - \bar{\tau}, \ldots, t - 1$. The latter is admissible thanking to the nesting property of the one-step controllable set sequence, see [4].

Now, each sample of the initial segments $x(\cdot)$ belongs to the same set $T_{i-1}$, and a simple way to proceed is to consider two vectors, namely $z_1$, $z_2 \in \mathbb{R}^n$, characterizing the terms in the one-step differences $y(\cdot)$. Specifically $z_2$ denotes a slack variable representing the first element in the one-step difference whereas $z_1$ is a term characterizing all the time delayed states with respect to $z_2$ along all the possible initial segments. Therefore the computed one-step controllable set, say $E_i$, is an inner approximation of the exact set $T_i$ and the descriptor form (24) consequently becomes:

$$
E_{aug} \bar{x}_{aug}(t+1) = \Phi_{aug}(\alpha(t)) \bar{x}_{aug}(t) + \bar{G}_{aug}(\alpha(t))u(t)
$$

with $\bar{x}_{aug}(t) = [x^T(t) y^T(t) z_1^T(t) z_2^T(t)]^T$, $E_{aug} = \text{diag}\{I, 0, I\}$. We have

$$
\Phi_{aug}(\alpha(t)) = \begin{bmatrix} I & I & 0 & 0 \\
2\Phi(\alpha)(t) - I & -I & -\bar{\tau}\Phi(\alpha)(t) & \bar{\tau}\Phi(\alpha)(t) \\
0 & 0 & 0 & I \\
0 & 0 & 0 & \Phi(\alpha)(t) \end{bmatrix}
$$

$$
\bar{G}_{aug}(\alpha(t)) = \begin{bmatrix} 0 & G^T(\alpha)(t) & 0 & 0 \end{bmatrix}^T
$$

and the recursions (25) result. □

**Proposition** 1 provides the methodological arguments to explicitly deal with time-varying occurrences as it will be clarified in the next section. The significance of the model (26) relies on the conservativeness of the one-step controllable sets computation but it has the important advantage to jointly take care of polytopic uncertainty and plant/model discrepancy due to the network latency. A procedure for the computation of the sets $E_i$ can be straightforwardly deduced by referring to [15],[1].

### 3.3 Off-line time-delays and data-loss management

Here, we will discuss in detail all the time-delay configurations under which the networked system can operate.

#### 3.3.1 Sensor-to-controller and Controller-to-Actuator normal phases

In such a case the time-delay can be managed by computing two nested one-step controllable ellipsoidal sequences with $N + 1$ elements $(N > 0)$ $\{T_{i/DD}^N\}_{i=0}^N$ and $\{T_{i/OD}^N\}_{i=0}^N$, where both the sequences are
obtained by using the inner approximation (25), such that
\[
\mathcal{T}_{0}^{DD} \subseteq \mathcal{T}_{N}^{IOD}
\] (27)

The key idea can be stated as follows: the ellipsoidal sequences are achieved on the hypothesis that the time-delay occurrence is \(\tau(t) \leq \tau_{\text{max}}\) for \(\{T_{i}^{DD}\}_{i=0}^{N}\) and \(\tau_{\text{max}} < \tau(t) \leq \bar{\tau}\) for \(\{T_{i}^{IOD}\}_{i=0}^{N}\). Now, at each time instant and on the basis of the information \(\tau(t)\), if the current measurement \(x_{p}(t - \tau(t))\) belongs to \(T_{DD}^{i}\) (resp. \(T_{IOD}^{i}\)), there exists a command \(u_{i}\), compatible with (4) and capable to drive the state to \(T_{DD}^{i-1}\) (resp. \(T_{IOD}^{i-1}\)). Therefore, there exists an admissible control strategy which steers in a finite number of steps any initial state \(x(0) \in \mathcal{T}_{DD}^{N}\) (resp. \(\mathcal{T}_{IOD}^{N}\)) to the terminal (target) set \(\mathcal{T}_{0}^{DD}\) (resp. \(\mathcal{T}_{0}^{IOD}\)).

The rationale behind the additional condition (27) relies on the fact that the DD controller \(K_{DD}\) could be able to satisfy the NCS problem prescriptions for time-delay occurrences lower than \(\tau_{\text{max}}\).

As a consequence if a piece of information is sent with a latency \(\tau_{\text{max}} < \tau(t) \leq \bar{\tau}\), this unfavourable situation cannot be managed by \(K_{DD}\).

3.3.2 Data-loss

Let us define a data-loss event:

**Definition 3** Let \(\Delta^{DL} := [\bar{t}_{in}, \bar{t}_{fin}]\) be a time interval. A data-loss occurs if \(\tau(t) > \bar{\tau}\), \(\forall t \in \Delta^{DL}\).

Under data-losses, the combined use of the sequences \(\{T_{i}^{DD}\}_{i=0}^{N}\) and \(\{T_{i}^{IOD}\}_{i=0}^{N}\) is not able to deal with all the time-varying delay occurrences. In fact, due to the unpredictable nature of a data-loss event, it may happen that when the state lies in any set of the ellipsoid sequences \(\{T_{i}^{DD}\}_{i=0}^{N}\) and \(\{T_{i}^{IOD}\}_{i=0}^{N}\), there is no guarantee on the length of \(\Delta^{DL}\). To clarify such a key point, let us assume that at a generic time instant \(x(t) \in T_{i}^{DD}\), \(i < N\). Then, \(\Delta^{DL}\) lasts at most \(i\)-time steps without state measurements. The validity of the argument derives by construction of the family \(\{T_{i}^{DD}\}_{i=0}^{N}\), starting from \(T_{0}^{DD}\). There always exists in fact an input virtual sequence, namely \(\{u_{i}, u_{i-1}, \ldots, u_{1}\}\), such that the state trajectory is driven to the terminal region \(T_{0}^{DD}\) at the \((i + 1) - th\) time step, where the control action is generated by means of the state feedback law \(K^{DD}\) which requires an available state measurement. The same reasoning applies for the sequence \(\{T_{i}^{IOD}\}_{i=0}^{N}\). Then, two questions arise:

1) Which is the maximum admissible length of \(\Delta^{DL}\)?

2) How is it possible to proceed when \(x(\cdot) \in T_{0}^{DD}\) or \(T_{0}^{IOD}\)?

The question 1) is related to the fact that the command \(u(t)\) is strictly connected by construction of the sequences \(\{T_{i}^{IOD}\}_{i=0}^{N}\) and \(\{T_{i}^{DD}\}_{i=0}^{N}\) to the state measurement availability. By recalling from Proposition 1 that such ellipsoidal sequences are generated by taking into account the process/model discrepancy (see Fig. 2) due to time-delay occurrences within \([0, \bar{\tau}]\), then the command computed at
time \( t \) can be used in an open-loop fashion if \( \Delta^{DL} \) lasts less than \( \bar{\tau} \). In view of this reasoning, the length of \( \Delta^{DL} \) is at most equal to the given MATI value \( \bar{\tau} \).

As regards as 2), such a technical difficulty is avoided by imposing that the DD and IOD sequences are computed by satisfying the condition indicated in the following statement.

**Statement 1** Let \( x^+ := \Phi(\alpha) x, \forall \alpha \in \mathcal{P}, \forall x \in T^0_{DD} \) and \( \forall x \in T^0_{IOD} \) be the one-step state evolution under zero-input \( u \equiv 0_m \), then

\[
x^+ \subseteq T^N_{DD} \cup T^N_{IOD}
\] (28)

If (28) holds, one has that at the \((i + 1)\)th time step the zero input \( u \equiv 0_m \) can be applied in place of the feedback gain \( K^{DD} \) when no state measurements are available. In fact, since any admissible one-step state evolution lies in one element of the \( N + 1 \) regions \( \{T^i_{DD}\}_{i=0}^{N} \), say \( T^j_{DD} \), at the next time step a newly computed command sequence \( \{u_j, u_{j-1}, \ldots, u_1\} \) can be determined such that the state trajectory can be steered to the target set \( T^0_{DD} \). In what follows, we shall denote as \( T_{sup} \) the ellipsoidal region complying with (28).

### 3.3.3 Feedback-loss

The following definition for a feedback-loss event is considered:

**Definition 4** Let \( \Delta^{FL} := [\bar{t}_{in}, \bar{t}_{fin}] \) be a time interval such that \( \bar{t}_{fin} - \bar{t}_{in} \leq \bar{\tau}_c \). A feedback-loss occurs if \( \tau_c(t) \geq 1, \forall t \in \Delta^{FL} \).

In this case even if the actuator does not receive new packets, an admissible command input must be applied to the plant \( \forall t \in \Delta^{FL} \). To comply with this situation it is mandatory to guarantee that a new computed command \( u(t) \) is available after \( \bar{t}_{fin} - \bar{t}_{in} \) time instants before than \( u^{R}_1 \) becomes no longer admissible. Then the consequence is that the following conditions on the controller-to-actuator channel must be imposed

\[
\bar{\tau}_c \leq \tau_{max} \quad (29)
\]
\[
2 \tau_{max} \leq \bar{\tau} \quad (30)
\]

The necessity of (29) comes from the argument that since within the time interval \( \Delta^{FL} \) the plant operates in open-loop and the actuator applies the last received input \( u^{R}_1 \), the length of consecutive feedback-loss events is at most \( \tau_{max} \). The requirement (30) while ensures that the command \( u(t) \) computed with respect to the upper bound \( \bar{\tau} \) is usable for \( \tau_{max} \) time instants.

**Remark 1** - In view of (29), it results that the packet sent by the controller will be received at most after \( \tau_{max} \) time instants. If an out-of-order event occurs at the actuator side, the feasibility retention of the strategy is preserved because each input is computed such that it is a feasible, though not optimal, solution for \( \bar{\tau} \) time instants (see Section III.B) and under the requirement (30) it can be safety applied for additional \( \tau_{max} \) time instants. \[\Box\]
4 On-line phase

This section is devoted to consider time-delay occurrences by taking advantage of the MPC philosophy. This is achieved by translating into an on-line procedure the arguments developed in the previous section.

In what follows, we will resort to a time stamp mechanism in order to characterize the data-packet dispatch along the sensor-to-controller and controller-to-actuator channels. Specifically the pair \((x_p, t_x)\) denotes the state measurement \(x_p\) sent at the time instant \(t_x\), whereas \((u, t_u)\) has the following meaning: \(u\) is the command sent by the controller and marked by the stamp \(t_u\).

On the controller side, the computation of the control action \(u(t)\) is performed by using the following arguments. At each instant \(t\) the logic on the controller side first considers the most recent data-packet \((x_p, t_x)\) and selects the set containing \(x_p\). Then two scenarios could arise:

a) \(\tau(t) \leq \bar{\tau}:\) if \(x_p \in T_i^{IOD}\) the set \(T_i^{IOD}\) is selected, otherwise determine the smallest index \(i\) such that \(x_p \in T_i^{DD}\);

b) \(\tau(t) > \bar{\tau}:\) the state \(x_p\) is not available for checking its membership to IOD or to DD set sequences.

Then, an admissible input \(u(t)\) is computed by minimizing the performance index

\[
J_i(t)(x_p, u) = \max_j \|\Phi_j x_p + G_j u\|_P^2 (t_i-1)
\]

as detailed in what follows:

- If a) holds true then

  \[- x_p \in T_i^{IOD} : \]
  \[
u(t) = \arg \min J_i(t)(x_p, u) \]
  \[\text{subject to} \]
  \[
  \Phi_j x_p + G_j u \in T_i^{IOD} (t_i-1), u \in \mathcal{U}, j = 1, \ldots, l
  \]

  \[- x_p \in T_i^{DD} : \]
  \[
u(t) = \arg \min J_i(t)(x_p, u) \]
  \[\text{subject to} \]
  \[
  \Phi_j x_p + G_j u \in T_i^{DD} (t_i-1), u \in \mathcal{U}, j = 1, \ldots, l
  \]

- The case b) envisages the following events:

  1. if \(x_{-1} \in T_{i+1}^{DD}\) or \(x_{-1} \in T_{i+1}^{IOD}\) then \(u(t) = u_{C_1}\);
  2. if \(x_{-1} \in T_0^{DD}\) or \(x_{-1} \in T_0^{IOD}\) then solve (32)-(33) with \(x_{-1}\) in place of \(x_p\) and \(T_{sup}\) in place of \(T_i^{IOD}\).
The scenario a) is addressed as follows. If the delayed state measurement belongs to some $T_{DD}^i$, then $u(t)$ can be computed using the optimization (32)-(33) with $T_{DD}^i$ in place of $T_{IOD}^i$ and the feasibility property is retained thanks to the constraint (27). Note that the constrained optimization problem (32)-(33) can be simply converted into SDPs, see e.g. [1].

Let us now consider the data-loss scenario b). Such an event is treated by exploiting worst-case arguments. If $x_{-1} \in T_{DD}^{i+1}$ (or $T_{IOD}^{i+1}$) this implies that $x^p \in T_{DD}^i$, $i > 0$, the state $x^p \in T_{DD}^i$ (or $T_{IOD}^i$) is unknown and the command input cannot be correctly computed. Therefore the last computed command $u_{C-1}$ obtained under the uncertainty due to the upper bound $\bar{\tau}$ (see Proposition 1) is an admissible, even if not optimal, choice.

On the other hand, a direct approach can be used when $x_{-1} \in T_{DD}^0$ or $x_{-1} \in T_{IOD}^0$ because $T_{DD}^0$ and $T_{IOD}^0$ are robust positively invariant sets and $x_{-1}$ represents the worst case for the one-step state ahead prediction.

Finally, by referring to the scheme of Fig. 1 the actuator logic implements the following strategy:

$$u(t - \tau_c(t)) = \begin{cases} 
    u, & \text{if } (u, t_u) \text{ is available} \\
    u^R_{-1}, & \text{if a feedback-loss occurs}
\end{cases}$$

(36)

5 RHC algorithm

Thanks to the above developments the following Receding-Horizon control strategy is derived.

---

**NCS-MPC-Algorithm**

**Initialization**

0.1 Given the scalars $\bar{\tau}_c$, $\tau_{max}$ and $\bar{\tau}$ satisfying the requirements (29)-(30), compute the nonempty robust invariant ellipsoidal regions $T_{DD}^0 \subset \mathbb{R}^n$, $T_{IOD}^0 \subset \mathbb{R}^n$ and the stabilizing state feedback gains $K_{DD}$, $K_{IOD}$ complying with the prescriptions of Section III.A;

0.2 Generate the sequences of $N$ one-step controllable sets $T_{DD}^i$ and $T_{IOD}^i$ complying with (27) and (28). Compute $T_{sup}$;

0.3 Store the ellipsoids $\{T_{DD}^i\}_{i=0}^N$, $\{T_{IOD}^i\}_{i=0}^N$ and $T_{sup}$.

**On-line phase**

**Sensor side**

for all $t \in \mathbb{Z}_+$

---

16
1.1 send the packet \((x_p, t_{x_p})\) with \(t_{x_p} = t\) the time stamp;

Controller side

1 for all \(t \in \mathbb{Z}_+\)

1.1 If a packet \((x_p, t_{x_p})\) is arrived then

1.1.1 compute \(\tau(t) = \tau_m(t) + \bar{\tau}_c\) with \(\tau_m(t) = t - t_{x_p}\);

1.1.2 If \(\tau(t) \geq \bar{\tau}\) goto 1.2.1,

1.1.3 otherwise

1.1.3.1 If there exists \(i(t) := \min\{i : x_p \in T_{IOD}^i\}\) then

a- If \(i(t) = 0\) then \(u(t) = K_{IOD} x_p\)

b- else solve (32)-(33);

1.1.3.2 else find \(i(t) := \min\{i : x_p \in T_{DD}^i\}\)

a- If \(i(t) = 0\) then \(u(t) = K_{DD} x_p\)

b- else solve (34)-(35);

1.1.3.3 If \(t_{x_p} > t_{u_{-1}}\) then \(x_{-1} := x_p\) and \(u_{-1} := u(t)\);

1.2 If a data-loss occurs (no packed arrived) then

1.2.1 if \(x_{-1} \in T_{DD}^0\) or \(x_{-1} \in T_{IOD}^0\)

1.2.1.1 solve (32)-(33) with \(x_{-1}\) in place of \(x_p(t - \tau(t))\) and \(T_{sup}\) in place of \(T_{IOD}^i\);

1.2.1.2 update \(u_{-1} := u(t)\);

1.2.2 else \(u(t) = u_{-1}\).

1.3 Send the packet \((u, t_u)\) with \(t_u = t\) the time stamp;

Actuator side

1 for all \(t \in \mathbb{Z}_+\)

1.1 If a packet \((u, t_u)\) is arrived

1.1.1 apply \(u\).

1.1.2 If \(t_u > t_{u_{-1}}\) then \(u_{-1} := u\) and \(t_{u_{-1}} := t_u\);

1.2 else apply \(u_{-1}^{R}\);

One of the crucial points in any MPC design is to prove the feasibility retention and closed-loop stability of the scheme. The next proposition shows that the proposed **MPC-NCS-Algorithm** enjoys these properties.
Proposition 2 Let the sequences of sets $T_{i}^{DD}$ and $T_{i}^{IOD}$ be non-empty and $x_{p}(0) \in T_{N}^{DD} \cup T_{N}^{IOD}$. Then, the MPC-NCS-Algorithm always satisfies the constraints and ensures Uniformly Ultimate Boundedness for all $\alpha \in P_{l}$ and for all time-delay occurrences $\tau_{c}(t) \leq \bar{\tau}_{c}$ and $\tau_{m}(t)$.

Proof - On the actuator side, existence of solutions at time $t$ implies existence of solutions at time $t+1$, because the optimization problems in steps 1.2 and 1.3 are always feasible and under the boundedness of $\tau_{c}(t)$ the timely arrival feature of the computed command $u(t)$ is guaranteed. Let us consider without loss of generality that at the generic instant $t$ the state $x_{p}(t - \tau(t)) \in T_{i}^{DD}(t)$. First note that by construction there exists an input vector $u$ satisfying the input constraints (4) such that the one-step state evolution $\Phi(\alpha)x_{p}(t - \tau(t)) + G(\alpha)u$ belongs to $T_{i}^{DD}(t) - 1$, $\forall \alpha \in P_{l}$ and $\forall \tau(t)$ with $\tau_{c}(t) \leq \tau_{max}$ and $\tau_{m}(t)$ unbounded. Then, thanks to the recursion (5) and under the constraints (27) and (28) at the next time instant $t + 1$, the existence of a solution $u(t + 1)$ for the steps 1.2 and 1.3 is ensured. Under a data-loss occurrence, thanking to (29) the stored command $u_{C-1}$, though not optimal, is admissible at time $t + 1$ and, because the number of consecutive data-losses is less than $\tau_{max}$, feasibility is retained.

On the actuator side (step 1.5) the feasibility is guaranteed thanking to the properties of the ellipsoidal sequences $T_{i}^{DD}$ and $T_{i}^{IOD}$ and to the communication upper bound requirements (30) and (29). In fact, at each time instant the actuator logic is able to apply an admissible control input because the last stored command $u_{R-1}$ is updated at most after $\tau_{max}$ time instants.

Finally, Uniformly Ultimate Boundedness of the strategy follows by exploiting the same arguments:

- Under a Normal phases regime each initial state $x_{p}(0) \in T_{N}^{DD} \cup T_{N}^{IOD}$ is by construction driven to the terminal sets;
- Under data-loss or feedback-loss events the trajectory is in the worst case confined to $T_{N}^{DD} \cup T_{N}^{IOD}$ thanking to (27) and (28).

\[\square\]

6 Illustrative example

The aim of this example is to present results on the effectiveness of the proposed MPC strategy and make comparisons with existing competitors. In particular, the MPC scheme proposed by Pin and Parisini [21], hereafter denoted as MPC-NDC-Algorithm will be considered and contrasted with the MPC-NCS-Algorithm both in terms of control performance and memory resources. All computations have been carried out on a PC Intel Quad Core with the Matlab LMI and Optimization Toolboxes. The physical parameters are summarized in Table 1.

A path-tracking problem for a mobile autonomous robot is here illustrated. The robot moves thanking to a differential drive with two driving wheels and a single caster, as shown in Fig. 3. A state-space model of the mobile robot is as follows (see [27] for details):
Table 1: Model parameters

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value (MKS)</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M$</td>
<td>5</td>
<td>chassis mass</td>
</tr>
<tr>
<td>$M_L$</td>
<td>0.1</td>
<td>Left wheel mass</td>
</tr>
<tr>
<td>$M_R$</td>
<td>0.1</td>
<td>Right wheel mass</td>
</tr>
<tr>
<td>$J_L$</td>
<td>0.000125</td>
<td>moment of inertia of the Left wheel</td>
</tr>
<tr>
<td>$J_R$</td>
<td>0.000125</td>
<td>moment of inertia of the Right wheel</td>
</tr>
<tr>
<td>$J_\theta$</td>
<td>0.2250</td>
<td>moment of inertia of the wheel w.r.t. the z-axis</td>
</tr>
<tr>
<td>$R$</td>
<td>0.05</td>
<td>wheels radius</td>
</tr>
<tr>
<td>$D$</td>
<td>0.3</td>
<td>distance between the two wheels along the axle center</td>
</tr>
<tr>
<td>$b$</td>
<td>0.1</td>
<td>friction coefficient</td>
</tr>
<tr>
<td>$\phi$</td>
<td>$\pi/48$</td>
<td>tilt angle of the incline plane</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\dot{x}(t) &= v(t) \cos(\theta(t)) \\
\dot{y}(t) &= v(t) \sin(\theta(t)) \\
\dot{v}(t) &= \frac{T_L(t) + T_R(t)}{M + \frac{J_W}{R^2}} - (M + 2M_W) g \sin(\phi(t)) - b v(t) \quad (37) \\
\dot{\theta}(t) &= \omega(t) \\
\dot{\omega}(t) &= \frac{D}{J_\theta + \frac{J_W}{R^2} D^2} \left( \frac{T_L(t)}{J_W + \frac{J_W}{R^2} D^2} - \frac{T_R(t)}{J_W + \frac{J_W}{R^2} D^2} \right) + \frac{M W g D \sin(\phi(t)) \cos(\theta(t))}{J_\theta + \frac{J_W}{R^2} D^2} \frac{\sin(\theta(t))}{|\sin(\theta(t))|}
\end{align*}
\]

where $x_p(t) = [x(t), y(t), v(t), \theta(t), \omega(t)]$ is the plant state and $u(t) = [T_L(t), T_R(t)]^T$ the command input.

A Polytopic Linear Differential Inclusion (PLDI) of (37) can be derived by using the arguments outlined in [5]. First, the plant is linearized around a given nominal solution $(\tilde{x}_p(t), \tilde{u}(t))$

\[
\dot{\tilde{x}}_p(t) = \Phi(\tilde{\theta}_w, \tilde{v}_w) \tilde{x}_p(t) + G \tilde{u}(t) \quad (38)
\]
Table 2:

<table>
<thead>
<tr>
<th></th>
<th>MPC-NDC</th>
<th>NCS-MPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Time [s]</td>
<td>0.0061</td>
<td>0.002</td>
</tr>
<tr>
<td>Transmitted info [bit/sample]</td>
<td>1024</td>
<td>96</td>
</tr>
</tbody>
</table>

where \( \tilde{x}_p(t) := x_p(t) - \hat{x}_p(t) \), \( \tilde{u}(t) := u(t) - \hat{u}(t) \) and

\[
\Phi(\tilde{\theta}, \tilde{v}) = \begin{pmatrix}
0 & 0 & \cos(\hat{\theta}) & -\hat{\theta} & \sin(\hat{\theta}) & 0 \\
0 & 0 & \sin(\hat{\theta}) & \hat{\theta} & \cos(\hat{\theta}) & 0 \\
0 & 0 & -b & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & \frac{D}{2} M_{W} g D \sin(\phi(\hat{\theta} + \sin^2(\hat{\theta}) + \sin(\hat{\theta}) - 1) | \sin(\theta) | \sin(\theta))}{(J_{\theta} + \frac{D^2}{2} (\frac{D}{2} + M_{W}))} & 0 \\
\end{pmatrix}
\]  

(39)

\[
G = \begin{pmatrix}
0 & 0 & 0 \\
0 & 0 & \frac{1}{M + \frac{J_{\theta}}{2\pi}} \\
0 & \frac{1}{M + \frac{J_{\theta}}{2\pi}} & 0 \\
0 & 0 & 0 \\
\frac{D}{2} M_{W} g D \sin(\phi(\hat{\theta} + \sin^2(\hat{\theta}) + \sin(\hat{\theta}) - 1) | \sin(\theta) | \sin(\theta))}{(J_{\theta} + \frac{D^2}{2} (\frac{D}{2} + M_{W}))} & \frac{D}{2} M_{W} g D \sin(\phi(\hat{\theta} + \sin^2(\hat{\theta}) + \sin(\hat{\theta}) - 1) | \sin(\theta) | \sin(\theta))} \\
\end{pmatrix}
\]  

(40)

Then, by considering a maximum displacement on the nominal instantaneous azimuth angle \( \hat{\theta}(t) \)

\[
\tilde{\theta} \in [\hat{\theta}(t) - \overline{\theta}, \hat{\theta}(t) + \overline{\theta}], \overline{\theta} = \frac{1}{18}\pi
\]  

(41)

and under the following constraint on the nominal instantaneous linear velocity \( \hat{v}(t) \)

\[
\tilde{v} \in [\hat{v}(t) - \overline{v}, \hat{v}(t) + \overline{v}], \overline{v} = 3.5 \, \text{m/sec.}
\]  

(42)

we have that all the solutions of (37) are also solutions of the following parameter varying PLDI

\[
\dot{\tilde{x}}_p(t) \in \left( \sum_{i=1}^{4} \alpha_i(t) \Phi_i \right) \tilde{x}_p(t) + G \tilde{u}(t)
\]  

(43)

where \( \Phi_i, i = 1, \ldots, 4 \) have been computed by evaluating the Jacobian matrix (39) along the vertices of the constraints (41)-(42).

In order to derive a robust approximation of (43), we have selected a finite set of equally spaced in time couples \( (\hat{\theta}_{wi}(t), \hat{v}_{wi}(t)) \), \( i = 1, \ldots, L \), evaluated along the nominal path (see Fig. 4) and then we have computed the corresponding PLDIs. Hence, by computing the convex hull of all the obtained matrix vertices

\[
\Omega = \text{Conv} \left\{ [\Phi_i, G]_{i=1}^{4}, j = 1, \ldots, L \right\}
\]  

(44)
the following robust PLDI has been derived

\[
\dot{\tilde{x}}_p(t) \in \left( \sum_{i=1}^{4} \beta_i \text{Vert} \{ \Omega \}_i \right) \tilde{x}_p(t) + G \tilde{u}(t)
\]  \hspace{1cm} (45)

The model (45) is discretized by using the forward Euler difference increment \( \frac{\tilde{x}_p(t+1) - \tilde{x}_p(t)}{T_c} \), with \( T_c = 0.01 \text{ sec.} \) the sampling time. Moreover the input torque is subject to the component-wise constraints:

\[-0.5 \text{ Nm} \leq \tilde{u}_i < 0.5 \text{ Nm}, i = 1, 2 \]  \hspace{1cm} (46)

The data are supposed to be exchanged using a non-acknowledged communication protocol (UDP). The maximum round trip delay is approximately 200 ms [3] and in what follows it represents \( \bar{\tau} \) (MATI value). The aim of the simulation is to check the robot capabilities to track the path reference depicted in Fig. 4.

![Robot path](image)

**Figure 4: Robot path**

Here we have considered the time-delay scenario shown in Figs. 5-6 where Fig. 5 depicts time-delays occurring on the sensor-to-controller link whereas Fig. 6 accounts for the delay on the controller-to-actuator channel. Note that data-losses and feedback losses occur within the time interval [40, 90] sec., under the assumption that short dropout time slots randomly follow normal phases and *vice versa*.

For the proposed method, by iteratively solving the matrix inequalities (13)-(14), the time-delay \( \tau_{max} \) has been approximated to 100 ms and the corresponding DD stabilizing control law computed, while the IOD feedback has been achieved by solving (18)-(19). Then, two families \( \{ T_i^{DD} \}_{i=0}^{N} \) and \( \{ T_i^{IOD} \}_{i=0}^{N} \) of \( N = 50 \) ellipsoids have been computed under the satisfaction of conditions (27) and (28) with \( T_{sup} \subseteq T_{11}^{DD} \).

In order to implement the **MPC-NDC-Algorithm** the following computations and choices are off-line made:

- Uncertainty compact ball \( B(\bar{d}) \) with \( \bar{d} = 0.0021 \);
- Lipschitz constant of the transition map: \( L_{f_{sp}} = 1.0144 \);
• Auxiliary linear controller:

$$\kappa_f(\tilde{x}_p) = \begin{bmatrix} 6.9630 & -6.7976 & 7.5858 & -30.1671 & -28.4610 \\ 6.5411 & 7.2072 & 7.5513 & 29.5401 & 26.6957 \end{bmatrix} \tilde{x}_p;$$

• Controller sequence length: $N_c = 33$ due to the constraint $N_c \geq \bar{\tau} + \tau_{\text{max}} + 1$;

• Stage cost: $h(\tilde{x}_p) = \tilde{x}_p^T Q \tilde{x}_p + \tilde{u}^T R \tilde{u}$, with $Q = I_n$ and $R = I_m$.

The initial state has been set to $\tilde{x}_p(0) = [0.5, 0.3, 0, \frac{\pi}{2}, 0]^T$ and all the relevant results are summarized in Figs. 7-9. In Fig. 7 the regulated state evolutions with respect to the path to be tracked (Fig. 4) under time-delay occurrences are depicted. It can be noted that the NCS-MPC strategy is capable to achieve good tracking performance also in presence of data-loss events. Starting from its initial position on the plane $x - y ([0.5, 0.3])$, which is different from the nominal path starting point $([0, 0])$ the control action is capable to enforce the robot attitude in order to track the desired trajectory. During the first 40 sec. of simulation, the communication time-delays are such that $\tau(t) \leq \bar{\tau}$ and
Furthermore, the controller-to-sensor delay $\tau_c(t) < 1$ (see Figs. 5-6), therefore the proposed strategy and the MPC-NDC algorithm apply less conservative control actions so achieving shorter settling times, see Figs. 7 and 8.

When data-loss and feedback loss scenarios are concerned ([40, 90] sec.), note that the controller-to-sensor delay $\tau_c(t)$ goes randomly beyond the sensor-to-controller latency the MATI $\bar{\tau}$, see Fig. 6.

For the NCS-MPC scheme, this implies that at the controller side $u(t)$ is computed under the condition that $x^+ \subseteq T_{sup}$ (Step 1.3.1.1) or imposed equal to the stored command $u_{C-1}$ (Step 1.3.1.2)). On the other hand, the actuator logic alternatively applies the received input $u(t)$ and the stored feasible command $u_{R-1}$ and the consequence is an oscillating effect on the torque inputs as it results by looking to the input evolutions depicted in Fig. 8. As an example, at $t = 55$ sec. the application of $u(t)$ (computed by means of the Step 1.3.1.1) leads to a “jump” into $T_{8DD} \subseteq T_{sup}$ of the state $x^+$, see Fig. 9, and a new input sequence is computed by following the prescriptions of the NCS-MPC algorithm as shown in Fig. 8. During such a phase, the regulated robot attitude initially departs from the equilibrium condition because the packet dropouts impose the worst control action on the mobile robot, see the grey zone in Fig. 7. By looking to the control performance achieved by the MPC-NDC scheme during this phase, notice that, even if the regulated state evolutions do not suffer of abrupt divergences from the nominal path, they show a slow rate of convergence to the desired trajectory. As a conclusion, it appears that the two strategies perform quite similarly in all the time-delay scenarios.

The analysis on the memory requirements puts in light two aspects. First, the on-line complexity, evaluated by computing the average CPU time (seconds per step), has shown that the NCS-MPC algorithm (0.006 sec.) is less demanding than the MPC-NDC one (0.0098 sec). Therefore the on-line optimization of the MPC-NDC algorithm can at most update three control moves at each time instant which gives a reason of the control behaviour observed in the simulations. Then, the actuator buffer length of the MPC-NDC strategy requires higher memory resources (an input sequence of length $N_c = 33$ must be stored) than to that pertaining to the the proposed scheme (a single command $u_{R-1}$). As a consequence, because at each time instant an entire sequence of $N_c$ control moves must be sent from the controller to the actuator, the use of the MPC-NDC algorithm needs a large bandwidth of the controller-to-sensor channel.

7 Conclusions

In this paper, a novel receding horizon strategy for uncertain networked systems subject to input saturations and data-losses has been proposed. Viability theory and LMI techniques have been used to move off-line most of the computations.

The key idea was to develop a worst-case framework based on set-invariance concepts and ellipsoidal calculus to properly manage the absence on state measurements due to large network delays. First, the off-line phase has been built up in order to avoid any critical situation by imposing adequate constraints on the construction of the one-step controllable sets. Then, the on-line phase enjoys the
capability to deal with unreliable communication channels without affecting closed-loop stability and constraints satisfaction. Moreover it is worth noticing that the requested computational effort and resources requirements are modest when compared with those of other competitor schemes. Finally, an illustrative example testifies that the NCS-MPC strategy allows to hold the control performance at a satisfactory level in view of unrecoverable packet dropout events.

References


Figure 8: Command inputs

Figure 9: Switching signal


