SEVENTH FRAMEWORK PROGRAMME
THEME – ICT
[Information and Communication Technologies]

Contract Number:	223854
Project Title:	Hierarchical and Distributed Model Predictive Control of Large-Scale Systems
Project Acronym:	HD-MPC

Deliverable Number:	D3.4.1
Deliverable Type:	Report
Contractual Date of Delivery:	March 1, 2010
Actual Date of Delivery:	**March 1, 2010**
Title of Deliverable:	**Report of literature survey and analysis regarding timing and delay issues**
Dissemination level:	Public
Workpackage contributing to the Deliverable:	WP3
WP Leader:	RWTH Aachen
Partners:	TUD, POLIMI, RWTH, USE, UNC, SUP-ELEC
Author(s):	Daniel Limon (USE) Jairo Espinosa, Felipe Valencia and José García (UNC) Inga J. Wolf, Wolfgang Marquardt (RWTH) Alfredo Núñez (TUD)

Table of contents

Project co-ordinator

Name: Bart De Schutter
Address: Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 Delft, The Netherlands
Phone Number: +31-15-2785113
Fax Number: +31-15-2786679
E-mail: b.deschutter@tudelft.nl

Project web site: `http://www.ict-hd-mpc.eu`

Executive Summary

This report describes the results of a literature survey regarding timing and delay issues and delay issues in the context of hierarchical and distributed MPC. More specifically, the following topics are considered:

- When a control system is implemented in a distributed fashion, with multiple processors communicating over a network, both the communication delays associated with the network and the computation delays associated with the processing time can degrade the systems performance. In this case, the performance of the system may depends not only on the performance of the individual components but also on their interaction and cooperation. Therefore, Chapter 1 discusses modeling and control of time-delay systems, including stability and robustness.
- Chapter 2 focuses on communication and computational delay in MPC in the context of networked control systems. We characterize the issues related to communication delays and dropped network packets. Next, we discuss model-based compensation of the dynamic effects of the network, and efficient schemes for on-line optimal control and MPC in networked control systems.
- The topic of Chapter 3 is robust MPC for delayed systems. There we consider in particular stability and prediction.

Chapter 1

Modeling and Control of Time-delay Systems

1.1 Introduction

When a control system is implemented in a distributed fashion, with multiple processors communicating over a network, both the communication delays associated with the network and the computation delays associated with the processing time can degrade the system's performance. In this case, the performance of the system may depends not only on the performance of the individual components but also on their interaction and cooperation. [60].

1.2 Delay Modeling

In ([2], [3]) the classical approach of delay approximation by Taylor series expansion [47] is replaced by an the injection of a delay term (from the system state equation) into the interconnection variables. This gives a reduced system compared with the classical approach. Eq. (1.1) shows an approximation of the scalar linear system with delay using the Taylor expansion.

$$\dot{x}(t) = ax(t) + bx(t-\tau) \qquad \cong \qquad \dot{x}^{ex} = \hat{A}x^{ex}(t) + \hat{B}S(\tau)x^{ex} \tag{1.1}$$

where $\hat{A}$ and $\hat{B}$ represent operators a and b for the scalar case, x^{ex} represents the Taylor coefficient vector for the scalar function $x(t)$ and $\dot{x}^{ex}$ represents the Taylor coefficient vector for the scalar function $\dot{x}(t)$, and $S(\tau)$ is the Taylor delay operational matrix. This method increases the system order as the number of terms were taken in the Taylor series expansion, while the method used in [2] and [3] preserve the system order making the computational efficiency more tractable when the system becomes a large-scale one. This approach basically take a model of the as it is shown in Eq. (1.2) which arise from a partition of a global model into M subsystems with interactions among them.

$$X_i(k+1) = A_iX_i(k) + B_iU_i(k) + C_iZ_i(k) \tag{1.2}$$

where X_i denote the state vector, U_i is the input vector, and Z_i (with $i = 1,\ldots,M$) are the interaction variables. In this case, Z_i is described by Eq. (1.3):

$$Z_i = \sum_{j=1}^{M} [L_{ij}X_j(k) + K_{xij}X_j(k-d_{xj}) + K_{uij}U_j(k-d_{uj})] \tag{1.3}$$

with L_{ij}, K_{xij} and K_{uij} are the interaction gain matrices among subsystems with respect to the states, delayed states and delayed inputs, and d_{xj} and d_{uj} denote the delay of the states and the inputs respectively.

In [17], [19] the authors consider the delay as a positive constant. In [17], this constant is determined by the so-called transmission time delay. As it will be seen in the next section, this variable is used to determine the maximum transmission time delay allowable to guarantee the stability of the system. In [19], the communication delay produced by the acquisition of measurements is considered different to the delay produced by the controller when the control input is sent. This values are used to find sufficient conditions for guaranteeing system stability in an input-feedforward-output-feedback-passive nonlinear systems.

In [6], the delay is assumed as a random variable inside a defined set $d \in D := \{0, 1 \ldots, \overline{d}\}$, being $\overline{d}$ the maximum delay. Thus the model of the system is given by

$$x(k+1) = Ax(k) + Bu(k-d) \tag{1.4}$$

Using a delay depended state-feedback control law $u = k_d^T x,\ k_d \in \mathbb{R}^n$, and assuming that the delay d can be modeled by an independent and identically distributed random process or a Markov chain, the system (1.4) can be formulated as a jumped linear system:

$$z(k+1) = A_d z(k) \tag{1.5}$$

where the augmented state vector $z(k) = [x(k)^T x(k-1)^T \ldots x(k-\overline{d})^T]^T$ and

$$A_d = \begin{bmatrix} A & \leftarrow & Bk_d^T & \rightarrow & 0 \\ I & 0 & \cdots & 0 & 0 \\ 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix}$$

Note that the position of the matrix Bk_d^T is determined by the delay d.

On the other hand, in [8] the authors consider the delay as a positive constant ε_{ij} such that

$$v_{ij}^c(s) = e^{\varepsilon_{ij} s} w_{ij}^c(s) \tag{1.6}$$

where v_{ij} represent the interconnection input variables, and w_{ij} the interconnection output variables. Then the closed loop system becomes

$$\begin{bmatrix} \dot{x}_i^c(t) \\ w_i^c(t) \\ z_i^c(t) \end{bmatrix} = \begin{bmatrix} A_{TTi}^c & A_{TSi}^c & B_{Ti}^c \\ A_{STi}^c & A_{SSi}^c & B_{Si}^c \\ C_{Ti}^c & C_{Si}^c & D_i^c \end{bmatrix} \begin{bmatrix} x_i^c(t) \\ v_i^c(t) \\ d_i^c(t) \end{bmatrix} \tag{1.7}$$

being $z_i(t)$ and $d_i(t)$ the performance output and the local exogenous disturbance in the standard H_∞ formulation.

The approach proposed in [16] is based on the representation of the delays of the system as linear operators with no delay, allowing to represent the system as a norm-bounded uncertainty ones. To model the system, consider the Ito type differential equation

$$\begin{aligned} dx &= [\widetilde{A}_0 x(t) + \widetilde{A}_1 x(t-\tau(t)) + B_1 w(t)]dt + \widetilde{B}_2 u(t)dt + Gx(t)d\beta(t) + Hx(t-\tau(t))d\nu(t) \\ dy &= [C_2 x(t) + \widetilde{C}_2 x(t-\tau(t)) + D_{21}\eta(t)]dt + Fx(t)d\zeta(t) \\ z(t) &= C_1 x(t) + D_{12} u(t) \\ x(\theta) &= 0,\ \forall\theta \leqslant 0 \end{aligned} \tag{1.8}$$

where $x(t)$ is the state vector, $w(t)$ is the exogenous disturbance, $y(t)$ is the measurements vector, $\eta(t)$ is an additive measurements noise, $z(t)$ is the objective vector, $u(t)$ is the control input signal, $B_1, \widetilde{B}_2,\ C_1, \widetilde{C}_2,\ D_{12},\ D_{21},\ F,\ G,\ H$, are time invariant matrices and $\widetilde{A}_0, \widetilde{A}_1, \widetilde{B}_2$ are matrices satisfying the following norm-bounded uncertainties:

$$\begin{aligned} \widetilde{A}_0 &= A_0 + E_0 F_0 \widetilde{H}_0 \\ \widetilde{A}_1 &= A_1 + E_1 F_1 \widetilde{H}_1 \\ \widetilde{B}_2 &= B_2 + E_0 F_0 \widetilde{H}_2 \end{aligned} \tag{1.9}$$

where $F_i^T F_i \leqslant I,\ \forall i = 0,1$, and being $A_0,\ A_1,\ B_2$ the matrices of the nominal system. $E_i,\ i = \{0,1\},\ \widetilde{H}_i,\ i = \{0,1,2\}$ are constant matrices. In equation (1.8), $\tau(t)$ is an unknown time-delay such that for positive scalars $h,\ d$ satisfies:

$$\begin{aligned} 0 \leqslant \tau(t) \leqslant h \\ \dot{\tau}(t) \leqslant d < 1 \end{aligned} \tag{1.10}$$

and $\beta(t),\ \nu(t),\ \zeta(t)$ are Weiner processes.

In [36] the delays are assumed to be bounded, a priori unknown, but they must be available in real time. In order to show this, consider the following discrete-time system:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ z(k) &= Cx(k) \end{aligned} \tag{1.11}$$

with $x(k) \in \mathbb{R}^n$ the system state, $u(k) \in \mathbb{R}^r$ the control input, and $z(k) \in \mathbb{R}^p$ the output. Let $\tau_{max} \in \mathbb{N}$ the maximal delay considered and $\tau_{min} \in \mathbb{N}$ the minimal delay. In this paper it is assumed that the system has at each time k, a delay $\tau_k \in \mathscr{T} = [\tau_{min}, \tau_{max}]$, that is multiple of the sampling time. Therefore, the system 1.11 can be rewritten to take into account the delays:

$$x(k+1) = Ax(k) + Bu(k-\tau_k) \tag{1.12}$$

Next, when the formulation is presented, the authors assume a switched system with parameter $\alpha(k) \in \Omega$, where:

$$\Omega = \left\{ \alpha = \begin{bmatrix} \alpha_{\tau_{min}} \\ \vdots \\ \alpha_{\tau_{max}} \end{bmatrix} \in \mathbb{R}^{\tau_{max}-\tau_{min}+1} \,\middle|\, \forall i \in \mathscr{T}, \alpha_i \in \{0;1\}; \sum_{\tau_{min}}^{\tau_{max}} \alpha_i = 1 \right\} \tag{1.13}$$

Hence the system has the following form:

$$\tilde{x}_{k+1} = \tilde{A}(\alpha(k))\tilde{x}_k + \tilde{B}(\alpha(k))u_k \tag{1.14}$$

with $\alpha(k)$ the characteristic function of the delay τ_k in the system (1.12). This characteristic function is defined as:

$$\alpha : \begin{cases} \mathbb{N} \longrightarrow \Omega \\ k \longmapsto \alpha(k) \ \text{ s.t } \ \forall \tau \in \mathscr{T} \ \begin{cases} \alpha_\tau(k) = 1, & if \ \ \tau = \tau_k \\ \alpha_\tau(k) = 0, & if \ \ \tau \neq \tau_k \end{cases} \end{cases} \tag{1.15}$$

This is defined in order to establish a bijection between $\alpha(k)$ and τ_k. In fact τ_k can be expressed using $\alpha(k)$ as follows:

$$\tau : \begin{cases} \mathbb{N} \longrightarrow \mathscr{T} \\ k \longmapsto \tau_k = \sum\limits_{i \in \mathscr{T}} i\alpha_i(k) \end{cases} \tag{1.16}$$

In [56] a nonlinear continuous-time system is considered as follows:

$$\begin{aligned} \dot{x}(t) = f(x(t), u(t)), \quad & x(0) = x_0 \\ x(t) \in X \subseteq \mathbb{R}^n, & u(t) \in U \subset \mathbb{R}^m \end{aligned} \tag{1.17}$$

where U is assumed to be compact, X to be connected, $(0,0) \in X \times U$, and $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is locally Lipschitz continuous and such that $f(0,0) = 0$. It is assumed that the system is connected to a Nonlinear Model Predictive Controller (NMPC) through a shared network. Some assumptions are made to solve the problem: all packages are time-stamped; a common time-frame is available for all components either by a global clock or either by a set of synchronized clocks; the network is subject to random but bounded delays:

$$\tau_{sc}(t) \in [0, \tau_{sc}^{max}], \ \ \tau_{ca}(t) \in [0, \tau_{ca}^{max}] \tag{1.18}$$

where τ_{sc} is the delay between the system and the controller (measurement), and τ_{ca} is the delay between the controller and the actuator (control input).

The network is also subject to random but limited information losses; computational delays are embedded onto controller-to-actuator delays; the whole state is completely available either by direct measurement or observation.

The time-varying delays are analyzed separately. First, for time-varying measurement delays, it is assumed that the state is available for the controller only at $t_i + \tau_{sc}$, i.e. the NMPC controller has only old information to solve the optimal control problem. However, since the previous assumptions are verified, and an exact plant model is available, the current delayed state can be obtained by forward prediction. On the other hand if there exist time-varying delays between the controller and the actuator the input consistency is lost. Then, there must be buffers at the controller and the actuator to re-gain the input consistency, using the worst case delay τ_{ca}^{max}. In this paper, it is referred other contribution where this artifice has been proved to be effective.

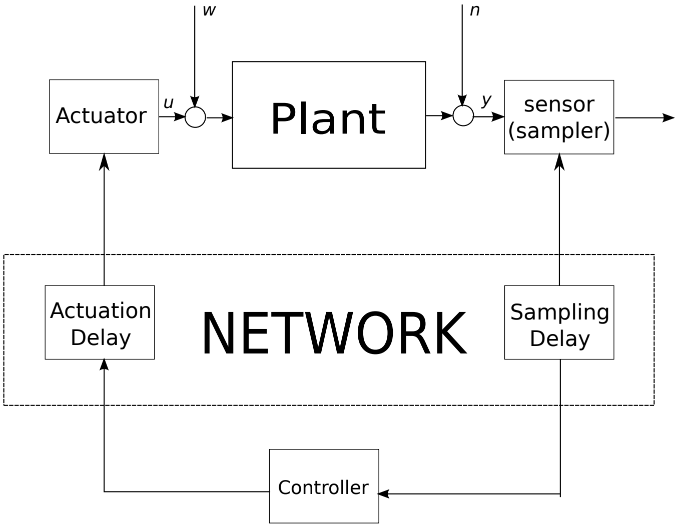

Figure 1.1: Assumed delays in the scheme

In [10] a linear, discrete-time system as (1.12) is considered. First it is assumed a plant delay as $\tau_c = dT$, where $d \in \mathbb{N}^+$, and T is the sampling period. In this paper it is stated that a simple state feedback taking into account the delay is sufficient to tackle the problem. However some problems arise: (1) the time-delay is not perfectly known, (2) the system output is not available at every kT sampling instant since, due to scarce data or event detection, only some data can arrive to the controller (and delayed). Then, in this work both sensor-to-controller ($\tau_{sc}(t)$), and controller-to-actuator ($\tau_{ca}(t)$) time-varying delays was considered as time-varying. In Figure (1.1) are depicted these network delays. Moreover, they are defined as follows:

$$\tau_{sc}(t) \leq \bar{\tau}_{sc}T, \quad \tau_{ca}(t) \leq \bar{\tau}_{ca}T, \quad \tau_c(t) \leq \bar{\tau}_c T, \quad \bar{\tau}_{sc}, \bar{\tau}_{ca}, \bar{\tau}_c \in \mathbf{N}^+ \tag{1.19}$$

with the upper bar indicating the upper bound. Then the total time delay satisfies the next condition:

$$d(t) = \tau_c(t) + \tau_{ca}(t) + \tau_{sc}(t) \tag{1.20}$$

furthermore it is assumed that

$$d(t) = dT + \varepsilon(t) = hT + \alpha T + \varepsilon(t), \tag{1.21}$$

where hT is the delay being considered for computing the prediction scheme ($h \in \mathbb{N}$), αT is the delay variation ($\alpha \in \mathbb{Z}$) and $\varepsilon(t) \in \mathbb{R}$ is a small uncertainty in the knowledge of the delay. Looking for robustness in the time delay modeling, the delay is not anymore considered as a function of the sampling period but the $k-th$ sampling instant, then:

$$t_{k+1} - t_k = T + \xi_k \tag{1.22}$$

with T the ideal sampling time and ξ_k the uncertainty between consequent sampling instants. The above mentioned uncertainties must be bounded as:

$$|\xi_k| < \bar{\xi} \ll T, \quad |\varepsilon(t)| < \bar{\varepsilon} \ll T \tag{1.23}$$

1.3 Robustness

In order to tackle robustness issues in several systems, particularly networked systems, it is necessary to define the Maximum Allowable Delay Bound (MADB [59]) or a similar criteria [12] in order to determine the maximum interval of stability for a given control system or control structure. In [12] a sufficient condition for robust stability is expressed by means of a Linear Matrix Inequality (LMI) as Eq. (1.24) shows.

$$\begin{bmatrix} (A^0_{sf})^T P A^0_{sf} - P + \varepsilon(\delta^s)^2 N_1^T N_1 & (A^0_{sf})^T P S_1 \\ * & -(\varepsilon I_n - S_1^T P S_1) \end{bmatrix} < 0 \tag{1.24}$$

where P and S_1 are positive definite matrices, ε is a positive constant and A^0_{sf} is a matrix of the closed-loop representation of the system. Such a matrix corresponds to a nominal part of a delay (τ) which in turn is partitioned in a nominal part (τ^0) and an uncertain part (τ_Δ), being N_1 and δ^s defined by:

$$\begin{aligned} N_1(B_c, K_{sf}) &= -B_c(K^a_{sf} + S_2) \\ \delta^s = \sup\{\sigma_{max}(\Delta\Gamma(\tau_k))\} &\leq \max_{\tau \in [\tau_{min}, \tau_{max}]} \| \int_{h-\tau_{min}}^{h-\tau_k} e^{A_c \lambda} d\lambda \|_2 \end{aligned} \tag{1.25}$$

with $S_2 = [\bar{0}\ 1]$, and K^a_{sf} the state-feedback matrix. The cited method present a way to compute the maximum allowed delay and it can be used to compute the maximum allowable computer delay. Hence, the usefulness of several control algorithms is determined in order to be used in a network system.

In [8] the closed-loop model of the system was used to find the robustness and stability conditions due to occurred delays such that $0 \leqslant \varepsilon_{ij} \leqslant \varepsilon$. This analysis is based on structured singular values. Some theorems and corollaries are presented in order to demonstrate the authors point of view. Finally some related works are listed, as the proposed in [4], [30] and [35]. In [16] the introduction of the operators given by equations (1.26, 1.27), allows to express the system (1.8) as a non-delayed one. Moreover the system is also expressed with norm-bounded operators.

$$(D_0 g)(t) \triangleq g(t - \tau(t)) \tag{1.26}$$

$$(D_1 g)(t) \triangleq \int_{t-\tau(t)}^{t} g(s) ds \tag{1.27}$$

This allows to design a robust controller such that a cost function $J(\cdot)$ is less than zero, and derive sufficient conditions for the stability of the system. Moreover, the formulation proposed in [16] deals with uncertain time-delay systems.

1.4 Stability

The stability of a control system is linked with the robustness. The MADB criteria is used in [59] to propose a theorem expressed as a LMI to guarantee asymptotically stability for any time-delay τ_j satisfying $0 \leq \tau_j \leq \overline{\tau_j}$.
Consider the following dynamical system:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + \sum_{i=1}^{N} A_i x(t-\delta_j) \\ x(t) &= \phi(t) \end{aligned} \tag{1.28}$$

where $\phi(.)$ describes the initial condition of $x(t)$, and $t \in [-\delta, 0]$, with δ the upper band of δ_i. Then the stability theorem for these systems is as follows:

Theorem 1 *[26] If there exist $P > 0,\ Q_j > 0,\ X_j,\ Y_j$ and $Z_j,\ j = 1, \ldots, N$ such that*

$$\begin{bmatrix} \beta_{11} & \zeta^T \eta \\ \eta \zeta & -\Gamma \end{bmatrix} < 0, \begin{bmatrix} X_j & Y_j \\ Y_j^T & Z_j \end{bmatrix} \geq 0, \tag{1.29}$$

where,

$$\begin{aligned} \beta_{11} &= \begin{bmatrix} \xi_{11} & P\zeta_1 - \gamma \\ \zeta_1^T P - \gamma & -\theta \end{bmatrix} \\ \zeta &= [A, A_1, \cdots, A_N] \\ \zeta_1 &= [A_1, \cdots, A_N] \\ \gamma &= [Y_1 \cdots, Y_N] \\ \eta &= \overline{\tau}[Z_1, \cdots, Z_N] \\ \theta &= diag\{Q_1, \cdots, Q_N\} \\ \Gamma &= \overline{\tau} diag\{Z_1, \cdots, Z_N\} \\ \xi_{11} &= A^T P + PA + \sum_{j=1}^{N} Y_j + Y_j^T + \overline{\tau} X_j + Q_j \end{aligned}$$

Then closed-loop system with delays is asymptotically stable for any time-delay τ_j satisfying $0 \leq \tau_j \leq \overline{\tau}$.

For positive systems the asymptotically stability is proved in a simpler way [24]. Consider the system with multiple delays described by Eq. (1.30). The system is (internally) positive if and only if $x(t) \in R_+^n$, $x(t-\tau) \in R_+^n$, $y(t) \in R_+^p$ for any $x_0(t) \in R_+^n$ and for all inputs $u(t) \in R_+^m$, $t \geq 0$. The stability for positive systems with multiple delays does not depend on the delays as Theorem 2 states.

$$\begin{aligned} \dot{x}(t) &= A_0 x(t) + \sum_{k=1}^{q} A_k x(t-d_k) + Bu(t), \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{1.30}$$

Theorem 2 *[24] The positive system with delays (1.30) is asymptotically stable if and only if the positive system without delays*

$$\dot{x} = Ax, \qquad A = A_0 + \sum_{k=1}^{q} A_k \tag{1.31}$$

is asymptotically stable.

The principal advantage of the Theorem 2 is the easier form to prove it compared with other stability theorems based on Lyapunov theories.

In [17], the authors present a general framework including communication constraints, varying delays and varying transmission intervals. Based on the presented framework, the authors provide a Lyapunov-based procedure to compute bounds on the maximally allowable transmission interval and the maximally allowable delay in order to guarantee stability of the system. In order to show this, consider the continuous-time system with a controller:

$$\begin{aligned} \dot{x}_p(t) &= f_p(x_p, \widehat{u}) \\ y_p &= g_p(x_p) \\ \dot{x}_c(t) &= f_c(x_c, \widehat{y}) \\ u_c &= g_c(x_c) \end{aligned} \tag{1.32}$$

where x_p, x_c are the system and controller states respectively, $\widehat{u}$, $\widehat{y}$ denotes the most recent values of the control input and the system output respectively, and u, y are the control input and the system output respectively.

Assume that at the transmission time $t_{si}, i = 1, 2, \ldots, n$, with n the number of subsystems, the controller input u and/or the subsystem output y are sampled and transmitted over the network. The transmission times, t_{si}, satisfy $0 \leqslant t_1 < t_2 < \ldots < t_n$ and $\delta < t_{s_{i+1}} - t_{s_i} < \tau_{mati}$, where τ_{mati} is the maximum allowable transmission interval, and $\delta \in [0, \tau_{mati}]$. Then, at transmission time t_{si}, the subsystem accessing to the network saves their values on $y(t_{si})$ or send to the communication network their values of $u(t_{si})$. These values arrives after a transmission delay τ_i at the controller or actuator, depending on if the output values are sampled or the control actions are sent.

Thus, based on the last assumptions, the authors reformulate the system model, considering the error dynamics, the memory of the updated error, the number of transmission, the delay associated with the transmission interval, the delay and the event allowing to each subsystem the use of the communication network, as states. With the resulting model, a Lyapunov-based procedure is proposed in order to find the maximum transmission interval and the maximum delay at which the system is stable.

Consider again the reference [36]. The stability of the closed loop system with time delays is guaranteed following the Lyapunov-Krasovskii criteria. This is done with the following theorem:

Theorem 3 *Given the extended system*[1] *defined by*

$$\begin{aligned} \tilde{x}(k+1) &= \tilde{A}_{\tau_k}\tilde{x}(k) + \tilde{B}_{\tau_k}u(k) \\ z(k) &= \tilde{C}\tilde{x}(k) \end{aligned} \tag{1.33}$$

[1] In this work is referred as the original system augmented by the delayed inputs as new states.

it is supposed the existence of matrices $\tilde{G}_i \in \mathbb{R}^{(n+\tilde{n})\times(n+\tilde{n})}$, $\tilde{Y}_i \in \mathbb{R}^{r\times(n+\tilde{n})}$, symmetric matrices $\tilde{S}_i \in \mathbb{R}^{(n+\tilde{n})\times(n+\tilde{n})}$, and a scalar β, for $i \in \mathscr{T}$, such as $\forall(i,j) \in \mathscr{T}^2$

$$\tilde{M}_{ij} = \begin{bmatrix} \tilde{G}_i + \tilde{G}_i^T - \tilde{S}_i & \star & \star & \star \\ \tilde{A}_i\tilde{G}_i + \tilde{B}_i\tilde{Y}_i & \tilde{S}_i & \star & \star \\ \tilde{C}\tilde{G}_i & 0 & I & \star \\ R^{1/2}\tilde{Y}_i & 0 & 0 & I \end{bmatrix} > 0 \tag{1.34}$$

and

$$\begin{bmatrix} \beta I_{n+\tilde{n}} & \star \\ \begin{bmatrix} I_n & 0_{n\times\tilde{n}} \\ 0_{n\times\tilde{n}} & 0_{\tilde{n}} \end{bmatrix} & \tilde{S}_i \end{bmatrix} > 0 \tag{1.35}$$

The control law, on the form of extended state feedback $\tilde{x}(k)$ is given by:

$$u_k^*(\alpha(k),\tilde{x}(k)) = \tilde{K}(\alpha(k))\tilde{x}(k) = \left(\sum_{i\in\mathscr{T}} \alpha(k)\tilde{K}_i\right)\tilde{x}(k) \tag{1.36}$$

where $\tilde{K}_i = \tilde{Y}_i\tilde{G}_i^{-1}$ stabilizes the system, for all possible delay system $\{\tau_k\}$ and leads to the inequality:

$$\hat{J}(\{u_k^*\},x_0) \leq \beta \parallel x_0 \parallel_2^2 \tag{1.37}$$

In this Theorem, both stability and performance arise. Note that the given upper bound is such that a performance of the closed loop system is guaranteed under stability conditions. This upper bound is the worst case of the cost function, that is, the maximal degradation for all possible delay sequences.

In [56] the stability of the scheme is treated as a convergence issue. In fact, the main contribution of this work is based in a convergence Theorem. The first part of this theorem deals with the definition of the dynamical systems (1.17) and the kind of time-varying delays assumed. Then, it is defined some conditions to be fulfilled in order to the following limit is given:

$$\lim_{x\to\infty} \parallel x(t) \parallel = 0 \tag{1.38}$$

with $x(t)$ the state of the system. In qualitative way, the above mentioned conditions are:

- There exist x_0 initial condition, a feasible control u^*, and a bounded total delay τ such that the state $x(t)$ is in a subspace of the state-space, the dynamical system, and an inequality concerning a terminal cost on the total delay time are fulfilled.
- The assumptions presented in the first Section (concerned with this paper) are satisfied.
- The system without delays is stabilizable.
- The optimal control problem is solvable for a $t_0 \in \pi$, with π as defined before.

Stability issues are considered at the predictor and observer steps in [10]. The first one is treated as follows: consider the existence of an error in the knowledge of the integer part of the real delay dT. Thus, from (1.21), the delay can be written as $d = h + \alpha$, α the delay error. Then a simple state feedback is assumed as a control law:

$$u_k = -K\bar{x}_{k+h} \tag{1.39}$$

with $K \in \mathbf{R}^{m\times n}$, and $\bar{x}_{k+h}$ is predicted using the assumed delay without uncertainty. Robust stability is proven for time delays under certain upper bound and even for some uncertainties in the system matrices. On the other hand observer stability was proven taking into account both the control and measurement delays in a non-uniform approach. Then the state is estimated using the following structure:

$$x_{ob}(k+1) = Ax_{ob}(k) + Bu(k-h) + L_{MR}(y_s(k-\tau_{sc}) - Cx_{ob}(k)) \tag{1.40}$$

where L_{MR} is null if there is no measurement of the outputs. Then, the gain is selected at the N sampling as:

$$A_L^N = \bar{A}_L^N - L_{MR}C \tag{1.41}$$

where $\bar{A}_L^N$ is built according to the time delays.

1.5 Performance

In the selection of a suitable control structure for Networked Control Systems, some criteria must help to select the more appropriate one. In a past Section the robustness criteria were shown. Henceforth, the performance of the entire structure is considered. In [60] a theoretical framework is presented which allows the effect of time delays on the mechanical performance of the system (such as speed of response, trajectory following error, etc.) to be precisely modeled, and these models are used to determine the optimal network architecture for the given control system.

The performance of a mechanical control system is defined by how close is the system tracking for a given reference trajectory. In other words, given a desired reference trajectory $r(t)$ for the system, the performance is the difference between the actual system output $y(t)$ and the reference as Eq. (1.42) shows. Then it is possible to define a function that quantify the performance degradation due to a delay as Eq. (1.43), where $y^*(t)$ is the system output in an ideal system without delay.

$$P = \|y(t) - r(t)\| \tag{1.42}$$

$$\Phi(t) = \|y(t) - y^*(t)\| \tag{1.43}$$

The performance degradation function (Φ) can be expanded using Taylor series and finally an expression composed of differential functions multiplied by the respective delay is obtained. Each of these functions are called performance differential functions and they represent the performance degradation due to a unit time delay between nodes i and j.

In [36], a study of performance degradation of the networked control systems is made in the presence of time-varying delays. The complete scheme is composed by two steps. First, a set of Lyapunov functions depending on the time-varying delays, greater than $\hat{J}(\{u(k), x_0\})$ are designed. Second, the control law, is chosen in order to obtain the minimal element among this class of admissible Lyapunov functions. The upper bound is given by means of the Theorem 3 of this deliverable, in which a level of performance is guaranteed for any sequence of time-varying delays.

1.6 Control Structure Design

The design of control structures and hence the implementations requires all afore mentioned items. A supervisory structure for Networked Control Systems based on a fuzzy controller is proposed in [37]. The fuzzy model describes the behavior of the plant (linear or non-linear) using fuzzy groups based on linear representations of the plant and their delays as Eq. (1.44).

$$\text{Type 1 rules}: if\ x_i(k)\ is\ \mu_{ij}\ then\ x_j^N(k+1) = A_j x(k) + B_0 u(k) \quad (1.44a)$$

$$\text{Type 2 rules}: if\ \delta(k)\ is\ \nu_h\ then\ x_h^D(k+1) = B_h u(k) \quad (1.44b)$$

The supervisory control law is computed with Eq. (1.45)

$$if\ \widehat{x}_i(k)\ is\ \mu_{ij}\ then\ u_j^s(k+1) = -K_j \widehat{x}(k), \quad (1.45)$$

where the total control action is the sum of a nominal contribution computed by pole placement ($K\widehat{x}(k)$) and the contribution of the fuzzy supervisor controller ($-\sum R_j K_j \widehat{x}(k)$). Figure (1.2) helps to understand the controller structure. The supervisory level minimize the effects of time delays due to communications among agents in the network by considering a behavior described by the fuzzy model.

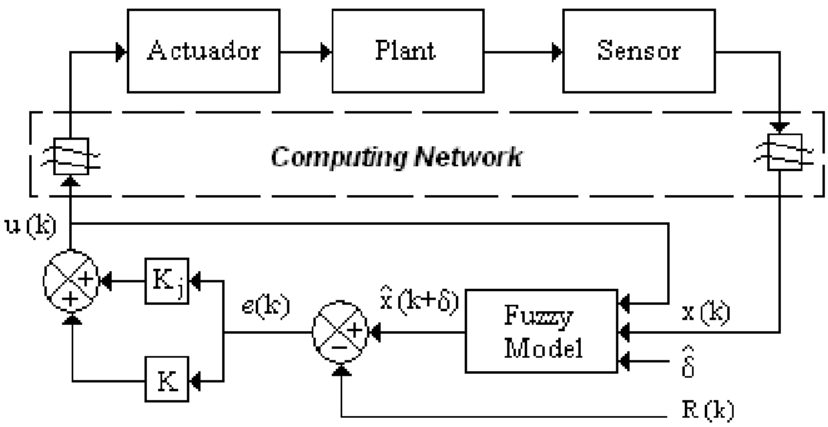

Figure 1.2: Diagram of Supervisory Fuzzy Control System (Taken from [37])

The design of a control structure can also be based on the MADB (Section 1.3) described in [59]. They propose a set of controllers that can be switched according to the MADB of the system governed by a local supervisor. The control actions are generated by a main controller which take into account the system delays. This main controller also uses the control inputs generated by local controllers with models without delays. Figure (1.3) completes the description of the control structure.

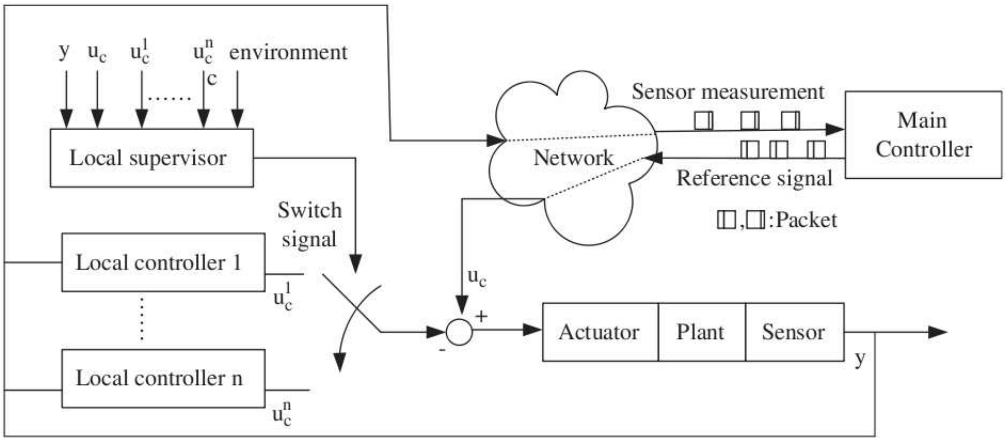

Figure 1.3: Hierarchical Controller Structure Based on MADB (Taken from [59])

In [6], a controller k_d with $u = k_d x$ is designed such that all subsystems have a common mode, based on the lemmas and following the methodology suggested on this paper. This allows to translate the delayed system to a non-delayed one. Also the control design proposed in this work allows to minimize the second moment decay rate, that is the classic criteria of the design of control systems for jump linear systems. However two questions arise from the work presented in this paper: how many common modes are necessary for guarantee the minimal decay rate of the second moment, and whether or not common eigenmodes are necessary and/or sufficient for a minimal second moment decay rate.
In [36] the control structure is defined by means of the Theorem 3. With this Theorem is guaranteed a system performance and stability under any sequence of time-varying delays.

In [56] a predictive approach is used to tackle the time-varying delays as stated earlier. Then a state prediction is calculated taking into account these delays:

$$\bar{x}(t_i + \tau_{sc} + \tau_{ca}^{max}) = \int_{t_i}^{t_i+\tau_{sc}+\tau_{ca}^{max}} f(x(\tau), u(\tau))d\tau + x(t_i) \tag{1.46}$$

to solve an optimization problem to the future time $ts_i = t_i + \tau_{sc} + \tau_{ca}^{max}$, with t_i the time-stamp associated with the measurement packet, and the other variables as stated before.

As it was mentioned, in [10] a predictor-observer structure was used to overcome the package loss and time delays in the networked control system. These components are considered as follows:

- The nonuniform observer receives non-uniform patterns assuming data loss. Then, its work is the reconstruction of the sent information to feed it to the controller in the desired time.
- Due to the network delay, data arrive to the observer delayed. The predictor must eliminate this influence.

Chapter 2

Communication and Computational Delay in MPC

2.1 Control over networks

The feedback control basis is the calculation of the manipulable variables of the system based on the knowledge of the measurement of some system variables. The designed control law is typically implemented in control systems which receive the system variables measurements provided by the sensors, calculate the control action to regulate the controlled variables at desired values and send it to the actuators, which manipulate the system inputs. Traditionally control systems use dedicated point-to-point wired communication links between a central control computer and the sensors and actuators. This has motivated that most of the control systems has been designed under the assumption of flawless communication links, namely, synchronized control and non-delayed measurements and actuation.

Distributed control systems (DCS) is widely used in process control systems since it was introduces in the 70s. The control modules locally regulate the process variables while they are interconnected by means of a lossy and shared serial network. The price drop of ASIC chips and hence electronic devices, has produced the development of control modules, sensors and actuators capable to cope with smart tasks as well as to be connected to a shared network. This has motivated that shared networks and networked smart sensors, actuators and controllers are more commonly adopted in modern control systems. Among the most used serial networks, we can find Controller Area Network (CAN), Profibus, Modbus, DeviceNet or Ethernet. These control systems are known as Networked Control Systems (NCS).

Some of the reasons of the success of Networked Control Systems are the following:

- Reduced cost of installation, reconfiguration and maintenance of the control system.
- Capability of distributed control systems.
- Flexibility of the control structure.
- Availability of sensor-actuator data
- Capability of plant management to prevent or deal with abnormal plant situations more quickly or effectively.
- Ability to improve the efficiency and reliability of the plant operation.

In networked control systems, sensors, actuators and controllers acts as network nodes which transmit the data through the shared data network. Therefore, the communication network becomes in a dynamic system inserted in the feedback control loop. This makes the analysis of networked control systems more complex since the effect of the network in the closed loop dynamics must be taken into account (See for instance the recent surveys [18, 58, 62] and the references there in).

This fact is particularly relevant in the case of decentralized control techniques. These are based on different control modules which share information to achieve the suit operation of the whole plant. The interconnected nature of this control structure is subject to the availability of shared network, and hence it can be intrinsically considered a networked control system.

2.2 Dynamic issues of Networked Control Systems

In order to analyze and design networked control systems, the effect of the communication channels in the control loop must be studied. These are now briefly presented:

2.2.1 Communication delay

When the information is transmitted through communication channels, the submitted information does not arrive the receiver immediately, but there are an elapsed time since this is sent and the information is available in the receiver node. The delay time may came from different sources such as: the class of network (the different layers that compounds the network, such as the physical layer, etc) used for the communication, the size of the network (this is typically classified into Local Area Network (LAN) or a Wide Area Network (WAN)) , etc.

In case of tele-operation or tele-control systems the distance between the controller and plant nodes together with the physical layer determine the delay, where the transport delay is dominant. In case of local area networks, the network protocols, such as the medium access control (MAC), affect to the communication delay. For instance, depending on the chosen MAC in such a way that this delay may be constant, time-varying or even random. MAC protocols is divided into two classes: random access and scheduling. The most often used protocol in random access networks, such as Ethernet for instance, is the Carrier sense multiple access (CSMA). In this case the delay of this class of networks is time-varying and random. Then, these networks are considered non-deterministic and the worst-case transmission time of packets is unbounded. If prioritized access is considered, the delay of higher priority packets is better conditioned.

Token passing (TP) and time division multiple access (TDMA) are usually adopted in scheduling networks. The TP protocol is used for instance in token bus and token ring architectures while TDMA is used in FireWire . The access to the medium in Scheduling networks is governed by an automaton which is responsible to determine which node is allowed to transmit. Then the communication delay is typically bounded and constant.

The communication delay can degrade the performance of control systems if the controller is designed ignoring it and can even destabilize the system.

2.2.2 Dropping Network Packets

Packet dropouts happen when the data transmission fails and hence the transmitted information is lost. Network packet drops may happen due to node failures or message collisions, namely transmission errors in the physical layer (which is far more common in wireless than in wired networks) or buffer overflows due to congestion. In tele-control systems data dropouts may happen for instance when packets with long transmission delays are discarded by the receiver if they are considered as outdated.

In order to achieve a successful transmission of the data, most network protocols adopt transmission-retry mechanisms. However, these protocols only retransmit for a limited time in order to avoid an enlargement of the network traffic, which produces a data loss. This is particularly interesting in networked control systems, where real-time data is sent, and if this is obsolete the data is not relevant and can be discarded.

For a networked control systems, this effect may make that the controller node does not receive the sampled data for a period of time or what is worse, the actuator node does not receive the new input to be applied to the plant. Notice that this issue can not be modeled as a communication delay previously reported, but typically is referred as unbounded or infinite delay. Therefore, data dropouts can potentially degrade the closed-loop dynamics more heavily than the communication delay.

2.2.3 Other issues

Although communication delays and packet dropouts are the most important effect of the network on the control loop, there are other effects that can be considered in the networked control systems:

- Band-Limited Channels: communication network can only carry a finite amount of packets per second. This imposes constraints that must be considered in the design of the networked control systems.
- Single or multiple packet transmission: depending of the class of network used in the NCS, the amount of data that can be sent in a packet is determined. If all the data to be transmitted can be compressed in one packet, then single-packet transmission is used. On the other hand, if the data takes more than one packet to be transmitted, then multiple-packet transmission is used. It is clear that single-packet transmission is more reliable and the probability of data losses and the expected communication delay is lower than in the multiple-packet case. However, multiple-packet transmission can be used also when the sources of the data to be transmitted are several nodes, as in the case of sensors or actuators distributed in a physical area.
- Sampling: the nodes in a networked control system are typically time-driven by built-in clocks. The signal to be transmitted is sampled, coded and packed. Different issues such as communication delays (when they are small), decoding or multiple packet transmission make that the sampling process be different to the sampling process of conventional control systems

2.3 Model-based compensation of the dynamic effects of the network

As it was mentioned before, the communication network may induce delays in the data transfer from the sensor of the plant to the controller (s-c delay) and from the controller to the actuator (c-a delay). Delays may produce a reduction of closed-loop performance and even un-stability. In order to deal with the delay, two approaches can be considered: taking into account the delay in the design of the controller or considering control system devices and communication protocols aimed to compensate the delay and make the network channel *transparent* for the controller. In this section, some techniques of the second method are reported.

The existing methods to compensate the delay are usually based on the availability of the following ingredients on the networked control system:

- Time and Event-driven buffered actuators: these are smart actuators capable to store a number of data packets in a buffer. Moreover they have an inner clock built-in. The actuator decide when and which control action is applied according to a prescribed logic (i.e. the output of an automaton).

- Time-driven sensors: this samples the plant variables and periodically transmit the measurement data.

- Time-stamped data: each signal value (namely, measurements or control actions) is transmitted together with the time when this value has been generated. This idea, firstly proposed in [40], plays a relevant role in the compensation techniques.

- Time-driven buffered controllers: these devices has the capability to store past measurements and control actions in buffers. Furthermore, it has an inner clock.

- Synchronization of control devices: in order to compensate the effect of communication delays, it is important that the times stamped into the data packets by the sender is coherent with the time of the inner clock of the receiver. This can be achieved by synchronizing the inner clocks of each network node, namely, sensor nodes, actuator nodes and controller nodes. Methods to cope with the synchronization can be software, hardware or a combination of both. One common method to synchronize two nodes consists in a node sends a message to the other node and this answer back with its inner time. This information can be used to calculate the time offset. [62]. A similar method based on master-slave nodes are used in [40]. In the case of Wide Area Networks, the synchronization of nodes is a very hard task, if possible.

The basic idea for the compensation of the communication delay is the well-known technique of using a predictor to forecast the current state of the plant. So, the estimated state is used as to calculate de current control action as if the delay was not present. To this aim, it is assumed that the real delay of the signal is known (assumed constant or calculated from the time-stamped data). Then, from the available (delayed) measurement and the control actions applied to the plant, an observer or Kalman filter, can be used to estimate the delayed state. Based on this, the current state can estimated by an open-loop predictor.

Different approaches to the solution of this problem have been proposed according to the conditions of the network (s-c delay, c-a delay or both, deterministic or random nature of the delay, data packet losses, noises, etc). It is remarkable the difference between communication delays and packet losses: in the case of communication delays it is assumed that every data packet sent to the actuator is applied if it is not received too late, while in the latter case it is not apparent which is the real control action applied to the plant. Since the forecasting of the state requires to know the sequence of inputs applied to the plant, this sequence of control inputs must be determined or estimated by the controller node.

A first approach to the compensation of network random delay has been proposed in [32]. This paper states the basics of the delay compensation of both s-c and c-a delays by means of an observer (or Kalman filter). They are assumed to vary randomly along the time but in such a way that the sum of both delays is a known constant. The authors highlight that the cumulated delay (this is called the total *Round Trip Time*) suffices for the calculation of the predicted control action sent to the actuator. To this aim, the history of the plant must be stored in the controller node. Nilsson et al. propose in [40] a similar approach where the delays are assumed to be random but with a known probability distribution. The necessity of knowing the cumulated delay is removed by introducing the time-stamp technique and assuming synchronized clocks in the nodes. The Round Trip Time (RTT) is assumed to be less than one sampling period. Stochastic control theory is used to derive the estimator and the optimal control law.

Remarkably, Bemporad in [5] has proposed a network compensator for large, but bounded, communication delays and packet dropouts. The NCS is modeled as two nodes, the plant node (with the sensor and actuator node) and the controller node. It is assumed that the plant node has a smart buffered actuator. Compensation is based on the open loop prediction and the calculation of the sequence of control inputs. This is packed and sent to the plant where it is stored in the buffer. In the case that the delay is larger than the sampling period the actuator apply the suit predicted action stored in the buffer. In the case of packet dropout event, the actuator apply the last control action and send an time-stamped error code to the controller node. The controller then updates the sequence of applied control inputs, recalculates the sequence of control actions and submit the new sequence in an error acknowledge packet, which makes that the actuator refresh the buffer with the submitted sequence.

In the excellent survey [62], the author summarize most of the topics on NCS and presents a compensation method for varying delays in the case of full state measurement and output measurement under the assumption that the total delay is lower than the sampling time.

In [54], communication delays are measured and compensated by means of an open-loop adaptive estimator based on a CARIMA model. Time-stamped data technique is adopted and this is used to synchronize the clocks built-in each node as proposed in [40]. Data packets with the time at each node are transmitted from controller to actuator and to the sensor and these echo back the to the controller. These are used to synchronize the clock based on a least-square estimation of the read delay. Also these are used by the controller to estimate the delay controller-actuator. The delay sensor-controller are directly estimated from time-stamped data. The actuator is assumed to be buffered in order to store predicted inputs to be applied in the case that the new control action is read later than sampling instant.

A model-based compensation for nonlinear plants has been proposed in [46] under the assumption of unknown time-varying delays in the case that the synchronization between nodes may be not possible. The problem of packet dropouts are also dealt with by the authors. In this case, the sensor and actuator node are assumed to be a single node called plant node. The plant node has a buffer to store the sequence of predicted inputs to be applied to the plant. The controller node has a buffer to store the sequence of inputs successfully sent to the plant together with the current state of the controller. This node is also equipped with a temporary buffer similar to the previous one. Based on the time-stamped data, the sensor-controller delay is determined in the controller node and the current state estimated using the stored sequence of applied inputs. The future sequence of control inputs and the future sequence of the controller state is calculated, stored in the temporary buffer and sent to the plant together with the Round Trip Time considered for the calculation of the predictions. Once the plant receives the data, the real RTT is calculated and it is checked if real RTT is lower to the estimated RTT. If so, the sequence of remaining inputs is updated. If not, the whole sequence is discarded.

In order to cope with packet dropouts, Polushin and coworkers propose in [46] to add a counter to data packet which is updated each time the control input sequence is updated. When the counter value received from the plant differs from the value stored in the controller network, then this means that the last data packet arrived successfully, and then the submitted sequence of inputs stored in the temporary buffer is used to update the buffer of the applied inputs. If the value of the counter remains the same, then the update stage is not carried out.

In [53, 20] a model-based compensation technique based on the measurement of the Round Trip Time has been proposed. The networked control structure of the model-based compensator is different to the previous ones, since the network compensator is located in the plant node (integrating the actuator node and the sensor node). In order to measure the Round Trip Time, the plant send the state of the plant in a time-stamped packet to the controller. The controller node calculates the sequence of predicted inputs to be applied and packet the data together with the time stamped in the received packet from the plant. When the plant receives the packet from the controller, the plant node measures the RTT by substracting the current time to the stamped time. Notice that since the Round Trip Time is measured in the plant node, no synchronization is required. In fact, controller node is not a time-driven node anymore, but an event-driven node: the controller node is only devoted to calculate the future sequence of control inputs whenever the data packet with the state of the plant is received. Clearly, this requires an estimation of the horizon for the calculation of the predicted sequence of inputs. The network compensator module in the plant node determines which is the input to be applied to the plant according to the measured RTT, that is, which is the input of the received sequence that corresponds to the current sampling time. If the data packet from the controller arrives later than the next sampling time, the compensator applies the corresponding input of the available sequence of inputs. of synchronization.

A different procedure to compensate the effect of the network is reducing the network traffic. If the rate of data packets to be transmitted is reduced (that is, the sampling time is enlarged), then the delay induced by communication network can be reduced making its effect negligible. In [39] the authors study the effect of reducing the data transmission between the sensor node and the controller node while it is assumed that no controller-actuator delay exists. Necessary and sufficient conditions to determine the maximum allowable transfer time which ensures closed loop stability are given. If the state is measured this is transmitted, but if the measurements are not the whole state, then the sensor node estimates the estate of the plant by means of an observer and this is sent. Based on the state received, the delay is compensated by means of an open-loop estimator. Since the controller is a continuous-time system, this considers the real state when available and the estimated state by an open-loop model of the plant between samples.

Another method to compensate the effect of the network is based on the so-called smart sensors. The main objective of these sensor is to transmit the measurement whenever this is necessary to maintain closed loop performance reducing the network traffic, and hence the delay. This topic is thoroughly studied in the excellent survey paper [18].

2.4 Model Predictive Control for NCS

The design of controllers for networked control systems has been widely studied, although still can be considered an open issue. Among the different control techniques that have been proposed (see the survey papers [62, 18] and the references there in), one of the most widely used is the predictive control. This choice seems natural taking into account that the most successful technique to compensate the effect of the delay in the closed loop is by means of a predictor and calculating a future sequence of inputs. Predictive controllers provide these future sequences as own terms of the controller.

Bemporad in [5] proposes the predictive control as a suit choice for control over unreliable networks with communication delays and data dropouts. The plant is assumed to be locally pre-stabilized and the predictive control law is calculated as if the network was transparent. A network compensation method is proposed which allows to use the predictive controller. Data dropout is tackled by maintaining the last control input which is safe since the plant is locally stabilized. An illustrative example where a predictive reference governor is used.

In [54] a Generalized Predictive Control together with a model-based compensation technique is used. From the synchronization methodology, the controller-actuator delay can be estimated, while the sensor-controller delay is read from the time-stamped data. From the delayed measurement, an open-loop estimator based on the CARIMA model is used to calculate the state of the plant. An adaptive multivariable Generalized Predictive Control (GPC) is used to calculate the predicted sequence of control inputs. The whole predicted sequence of control inputs are transmitted to the buffered actuator. This node reads the sequence and overwrite the old inputs. The authors propose a time-varying prediction horizon for the GPC according to the estimated communication delay, in order to reduce the size of the data packet and hence the network traffic.

The works [53] and [20] propose predictive networked control techniques using a different method for the network compensation which is based on the measurement of the Round Trip Time delay. The compensator is located in the plant node, where the RTT is measured from the time-stamped data. The state of the plant is assumed to be the past sequence of applied inputs and measured outputs, which acts as a (non-minimum) state of the plant. In the controller node is implemented a GPC which is based on the plant data received from the plant node and calculates the sequence of predicted inputs throughout a prediction horizon which is assumed to be larger than the possible communication and calculation delay. In [20] the proposed controller has been tested by a real application of a control-lab servo-motor system using Internet.

Millan and coworkers presents in [38] a networked predictive controller for communication channels with possible data dropouts or largely delayed. It is also assumed that the Round Trip Time is negligible w.r.t the sampling time. The nodes are assumed to be time-driven and the actuator and controller nodes are assumed to be buffered. In order to know which is the sequence of inputs successfully applied on the plant under the eventual loss of packets, Millan proposes a protocol between the actuator and the controller in such a way that the actuator sends an high-priority acknowledgment packet to the controller. This is assumed to arrive successfully within the sampling interval. Then the control action is calculated for the forecasted state to compensate the network effect. Moreover, in order to reduce the network traffic, the controller sends the packet only when the current control sequence differs from the buffered in a certain threshold. Under the stabilizing design of the predictive controller, networked closed-loop stability is derived from the nominal MPC inherent robustness.

In [22], a Networked Control System based on Dynamic Matrix Control (DMC) is studied. The authors assume a hierarchical structure where the regulatory level is located close to the plant while the advanced control level is remote and connected to the regulatory level by means of a communication network. The two nodes (plant and controller) are assumed to be synchronized and the communication delay to be bounded by known constant. Moreover, the plant node has a smart buffered actuator that stores the input packet and apply the input in such a way that the controller-actuator delay appears as constant. Then, robust analysis of the DMC is studied and sufficient stability conditions are provided.

In [45], a model predictive controller for uncertain nonlinear systems controlled using communication channels with time-varying delays and packet dropouts is presented. Delays are compensated using the Polushin's method and reconstruction of the sequence of applied inputs in the controller node is achieved by implementing a successful communication acknowledgment protocol. The robust MPC formulation is based on nominal predictions and on the constraint tightening method to ensure the robust constraint satisfaction [31]. The authors provide uncertainty bounds under which the controller ensures input-to-state stability of the networked closed-loop system.

In [14, 55] an event-driven networked Model Predictive Control (MPC) for constrained nonlinear continuous-time system is proposed. Buffered smart (time-driven and event-driven) actuator, time-driven sensor and event-driven control nodes are considered and all of them have a synchronized built-in clock. Communication delays are compensated by assuming known a wort-case bound of the Round Trip Time and adding a buffer in the actuator node that acts as a variable delay in order to make the total delay constant and equal to the worst-case delay. This allows to the controller node to determine the sequence of inputs that have really applied to the plant in order to compensate the communication delay by means of the open-loop predictor. The authors propose different methods to deal with the case of data losses. In [14], a called prediction-consistency method is proposed. This method assumes that the discrepancy between the sequence of inputs buffered in the controller and the sequence of input really applied on the plant may differ within a certain quantity. If this is small enough, closed-loop stability is ensured. In [55], the high-priority acknowledgment packet method is considered.

2.5 Computational Delay in MPC

In the process industry, the demand for large-scale NMPC applications based on detailed dynamic process models increases, since NMPC promises an improved production efficiency on increasingly competitive markets with less margins. Consequently, the computational time required to solve those large-scale dynamic optimization problems is often significant. If the solution time is not explicitly considered in the NMPC schemes, a delay between the application of the control trajectory and the current state information emerges. This causes a decrease in control performance. In this report, the best control performance is defined as the optimal closed-loop control with respect to the objective function defined in the optimization problem. For example, the optimal closed-loop control is the control that minimizes the deviation of the state trajectory from a given set-point. Note that the best control performance is not necessarily achieved by the fastest closed-loop control, since computational delay is considered in this work. At worst, the stability of the closed-loop system is endangered. This can already occur if the computational delay exceeds a fraction of the basic sampling period (cf. [13]). Hence, fast updating schemes such as explicit NMPC, Neighboring Extremal Updates, Newton-type methods and NLP sensitivity-based controllers (cf. [61]) are applied which are based on off-line state information, sensitivity information and disturbance rejection mechanism, respectively. In this report, explicit NMPC is not further considered because it is not suitable for large-scale problems. The computational delay for the other aforementioned schemes is small, and good control performance is achieved for several horizons succeeding a nominal solution. However, if the deviation from the optimal trajectory becomes too big, the optimal control problem is again solved neglecting computational delay. There are just few schemes considering the computational delay in linear MPC or NMPC (cf. [9] and [13]). In [13], the control trajectory of the preceding horizon is implemented during the maximal solution time, i.e. until an optimal solution for the current horizon is found. The maximal solution time required for obtaining the optimal solution is assumed to be known. This leads to a reduced amount of controls which can be adjusted during optimization and thus the degrees of freedom for the optimal control problem diminishes. The disadvantage of these schemes is, however, that the trade-off between the increasing solution accuracy and the reduction of degrees of freedom for every additional iteration made is not addressed. [1] presents a general framework that monitors the optimal control updating period, i.e. how many additional iterations lead to the best control performance.

In section 2.6, two additional fast-updating schemes are presented, the advanced-step NMPC controller by [61] and the real-time iteration scheme by [11]. Hereafter, an approach considering computational delay is described in detail (cf. [13]).

2.6 Efficient Schemes for On-line Optimal Control

Neighboring Extremal Updates ([57]): The Neighboring Extremal Updates (NEU) algorithm is an efficient and fast scheme for on-line optimal control. The new control trajectory is obtained almost instantaneously for the successive horizon with negligible computational effort. This is due to the fact that the relevant sensitivities and second-order derivatives, which are used for the NEU of the next horizon, are already computed during the current horizon. Thus, the computation is split up into a longer preparation phase on the current horizon and a fast updating phase in which an extended QP is solved after the new perturbed parametric uncertainties are available. Both the advanced-step NMPC controller and the real-time iteration scheme are similar to the NEU algorithm because they are also divided in a long preparation phase and a fast updating phase considering model mismatch as an uncertainty in the initial conditions.

The advanced-step NMPC controller ([61]): Starting from a nominal control trajectory, the advanced-step NMPC controller approximates the controls of the next horizon in a fast updating phase via a tangential predictor containing the new initial states as uncertainties. The advanced-step controller guarantees that the approximated control trajectory is computed based on an optimal control trajectory. This is achieved because the approximated control vector is implemented on the successive horizon while the controls are iterated to convergence in the preparation phase. An optimal control trajectory is known at the end of each horizon. Hence, the assumption has to hold that the preparation phase is smaller or equal to the sampling time. In the next updating phase, the controls are again determined by a tangential predictor, because the optimal solution of the last horizon is known. In this way, the scheme yields fast updates as well as good control performance since the optimal solution is computed for every horizon. Note that the approximated control trajectories computed by the NEU algorithm, however, are first iterated until the optimality criteria are fulfilled and then implemented on the successive horizon. A drawback of the advanced-step NMPC controller is that the scheme cannot be applied if the time needed to determine the optimal control trajectory is longer than the sampling time.

The real-time iteration scheme ([11]): In the real-time iteration scheme, an SQP-type iteration is performed from one horizon to the next. A tangential predictor based on the so called Initial Value Embedding is included in the SQP-type iteration. The parametric uncertainties are again the initial values like for the advanced-step controller. In general, the real-time iteration scheme also resembles the NEU algorithm and the advanced-step NMPC controller, though the approximated controls are neither improved by further QP iterations nor iterated to convergence. On the basis of the current approximated control vector, the next control trajectory is computed. Hence, deviations from the optimal trajectory can arise for strong perturbations or for error propagations over many horizons.

2.7 MPC Scheme for Considering Computational Delay

An NMPC scheme considering computational delay is presented in [13]. It extends the results of [9], which is one of few works on computational delay. The main idea of this approach is to apply the controls which are optimized on the previous horizon during the maximum solution time δ^c of the optimal control problem on the current horizon. Hence, these implemented controls are no longer available as degrees of freedom for optimization and fixed on the current horizon.

The open-loop optimal control problem formulated in [13] is equivalent to

$$\min_{u^j(t)} \quad \Phi(x^j(t), u^j(t), t_0^j, t_f^j) \tag{2.1a}$$

$$s.t. \quad 0 = f(\dot{x}^j(t), x^j(t), u^j(t), d^j(t), \hat{\theta}, t) \quad \forall t \in I^j \tag{2.1b}$$

$$y^j(t) = s(x^j(t), u^j(t), d^j(t), \hat{\theta}, t) \quad \forall t \in I^j \tag{2.1c}$$

$$x^j(t_0^j) = \hat{x}^j \quad \forall t \in I^j \tag{2.1d}$$

$$0 \geq h(x^j(t), u^j(t), y^j(t)) \quad \forall t \in I^j \tag{2.1e}$$

$$0 \geq e(x^j(t_f^j)) \quad \forall t \in I^j \tag{2.1f}$$

$$I^j = \left[t_0^j, t_f^j\right], \quad t^j = t^{j-1} + \Delta t, \quad j = 0, 1, \ldots J \tag{2.1g}$$

Here, sampled-data NMPC is considered, i.e. the open-loop optimal control problem is solved at discretized time instants. The so-called recalculation time is given by $\delta_j^r = t_{j+1} - t_j$. Hereby, the length of the recalculation time is not fixed, but it is dependent on the discretized points in time t_{j+1} and t_j. Two important assumptions are made. Firstly, the smallest recalculation time is larger than the maximum solution time, $\delta_{min}^r > \delta^c$. In this way, it is ensured that an optimal solution can be found before the next optimization starts. Note that the new optimization starts at the new recalculation instant t_{j+1} and not at the next sampling instant. Generally, the sampling time is significantly smaller than the recalculation time. According to [13], the optimal control trajectory which is applied to the system during the recalculation time can be approximated by a sample and hold staircase. Secondly, the horizon length T_p^j is assumed to be sufficiently long in order to guarantee that controls are available also during the maximum solution time on the next horizon, $T_p^j > \delta_{max}^r + \delta^c$.

In order to compute the optimized trajectory for fixed controls in the maximum solution time δ^c, the degrees of freedom of equation 2.1 are reduced. This is achieved via adding the additional constraint

$$u^j(\tau) = u_{opt}^{j-1}(t^{j-1}) \qquad \tau \in \left[t_0^j, t_0^j + \delta^c\right), \; t \in I^{j-1}, \tag{2.2}$$

where u_{opt}^j is the optimized control trajectory of horizon j. Hence, the optimal control problem is given by

$$\min_{u^j(t)} \quad \Phi(x^j(t), u^j(t), t_0^j, t_f^j) \tag{2.3a}$$

$$s.t. \quad 0 = f(\dot{x}^j(t), x^j(t), u^j(t), d^j(t), \hat{\theta}, t) \quad \forall t \in I^j \tag{2.3b}$$

$$y^j(t) = s(x^j(t), u^j(t), d^j(t), \hat{\theta}, t) \quad \forall t \in I^j \tag{2.3c}$$

$$x^j(t_0^j) = \hat{x}^j \quad \forall t \in I^j \tag{2.3d}$$

$$u^j(\tau) = u_{opt}^{j-1}(t^{j-1}) \qquad \tau \in \left[t_0^j, t_0^j + \delta^c\right), \; t \in I^{j-1} \tag{2.3e}$$

$$0 \geq h(x^j(t), u^j(t), y^j(t)) \quad \forall t \in I^j \tag{2.3f}$$

$$0 \geq e(x^j(t_f^j)) \quad \forall t \in I^j, \tag{2.3g}$$

where $I^j = \left[t_0^j, t_f^j\right]$, $T_p^j = t_f^j - t_0^j$ and $t^j = t^{j-1} + \Delta t$. This system of equations yields an effective algorithm considering computational delay in NMPC.

It shall be mentioned that [13] also established stability conditions in order to guarantee the stability of the closed-loop system.

Chapter 3

Robust MPC for delayed systems

3.1 Introduction

Among the advantages that predictive controllers exhibit, its capability to control systems with long dead-time is one of the most interesting [7]. The predictive nature of the controller makes that the effect of the delay be compensated by the prediction. In effect, once the controller receives the delayed measure of the system, the predicted trajectory can be calculated based on the prediction model. Hence the predicted sequence of inputs that minimizes the cost of the predicted trajectory can be calculated as it is done in the case that the plant does not exhibit delay.

Consider that the system exhibits a delay of d samples, then the predicted first d values of the predicted trajectory do not depend on the predicted control input, but on state of the system and the past inputs. This implies that the cost function to minimize must only consider the predicted tracking error from $d+1$ to N. This can be interpreted as an standard MPC without delay which control input is calculated for the predicted state for the current time. That is, the MPC for delayed systems can be posed as a MPC controller for the plant without delay plus a predictor in series. See [7, Section 4.9] for a more detailed explanation of this property.

The best-known predictive controllers such as DMC, GPC, etc. incorporates the delay compensation. However, the main concern of this natural delay compensation of MPC is the sensibility of the approach to uncertainties in the model system as well as in the modeled delay. In order to cope with this problem, several solutions have been proposed.

A well-known robust MPC based on the solution of an optimization problem with Linear Matrix Inequality constraints has been proposed in [27]. In this case, it is assumed that the uncertain system can be modeled by a Linear Difference Inclusion and it is subject to constraints. A robust receding horizon optimal controller for the constrained uncertain system is proposed. This is also extended to the case of uncertain systems with fixed and known delays in the input and state. To this aim, an extended state-space model is considered, which allows to use the procedure design proposed for systems without delays.

In [33], a predictive controller for linear systems with delays distributed in the input signals and in the output signals is presented. Moreover, each signal may exhibit a different delay. This work is motivated by the effect of communication channels on the signals. The authors propose a steady-state Kalman predictor to compensate the output delay while the input delay is considered in the controller design by means of an extended state-space formulation. Robustness of the proposed controller is studied and filtered predictions are proposed to improve the closed-loop robustness.

In [28] a predictive control formulation for delayed continuous-time linear systems is presented.

Closed-loop stability is derived by adding a new term to the standard cost function which penalize the integral error of the trajectory of a terminal period of time of length equal to the delay. The authors provide also LMI conditions for the design of the controller. This method improves a receding-horizon controller for delayed systems proposed by the authors in [29] where a cost function without weighting on the state is used and closed-loop stability is not ensured by construction, but it must be checked a-posteriori.

The Kothare's predictive control has been extended to the case of polytopic uncertain systems with bounded time-varying delays in [23]. The proposed controller consider that input is constrained and moreover closed-loop stability is ensured under feasibility of the optimization problem. In [21] presents a robust MPC for input constrained linear systems with time-varying bounded delay. The LMI-based robust MPC proposed in [27] is extended to cope with this problem thanks to a proposed method to enclose the uncertain system in a Linear Difference Inclusion. A delay dependent predictive control based on Kothare's for the case of uncertain linear systems with a constant delay has been proposed in [51]. The authors claim that the delay-dependence of the solution allows to achieve less conservative controllers.

Another interesting approach to the design of robust predictive controllers for delayed systems is based on the predictor-structure of the controller. As it is demonstrated in [41, Chapter 9], predictive controllers has the same inner structure that the dead-time compensator (DTC). Then stability and robustness of predictive controllers in presence of delays can be studied using the theory of DTC. It is remarkable that this approach is particularly interesting in the case of networked control systems where the network compensators also exhibit the structure of DTC.

As shown in [43] for generalized predictive controller (GPC) and in [41] for dynamic matrix controller (DMC), MPC strategies may be very sensitive to dead-time uncertainty. Moreover, robustness can be related to MPC algorithm predictor structures used to compute the output predictions up to the dead time. To overcome the robust tuning limitations, it was proposed a modified GPC algorithm based on the filtered Smith predictor called SPGPC [43, 44]. Initially, the multi-input multi-output (MIMO) SPGPC version, which was presented in [44], could be applied to control open loop stable process. Later on, it was shown in [41] that this approach can be extended to control unstable MIMO processes using a general formulation called DTC-GPC.

In the following sections, robust MPC for delayed systems based on filtered predictions are briefly presented. This is based on the results of the paper [50], where these ideas have been extended to the standard stabilizing MPC with terminal conditions [34]. A more detailed exposition on this topic can be found in [41].

3.2 Multivariable discrete model description

In this work, it is considered a MIMO process modeled by

$$\mathbf{y}(k) = \mathbf{P}(z^{-1})\mathbf{u}(k)$$

where $\mathbf{u}(k) \in \mathbb{R}^{n_u \times 1}$ is the input vector, $\mathbf{y}(k) \in \mathbb{R}^{n_y \times 1}$ is the output vector, $\mathbf{P}(z^{-1})$ is composed by $n_y \times n_u$ SISO transfer functions and z^{-1} is the backward shift operator. Each transfer function can be expressed as

$$p_{ij}(z^{-1}) = z^{-d_{ij}} \frac{z^{-1} b'_{ij}(z^{-1})}{a_{ij}(z^{-1})}$$
$$= z^{-d_i} \frac{z^{-1} b_{ij}(z^{-1})}{a_i(z^{-1})} = z^{-d_i} g_{ij}(z^{-1})$$

where d_{ij} is the dead-time of the transfer function between the j-input and the i-output; $d_i = \min_{j=1...n_u} d_{ij}$ and $a_i(z^{-1})$ is the least common multiple of $a_{ij}(z^{-1})$ for $j = 1...n_u$.

This representation is interesting because n_y independent auto-regressive models can be defined as

$$a_i(z^{-1}) y_i(k) = \sum_{j=1}^{n_u} b_{ij}(z^{-1}) u_j(k - d_i - 1) + \frac{e_i(k)}{\Delta} \tag{3.1}$$

where $e_i(k)$ is a noise signal and $\Delta = 1 - z^{-1}$ by definition.

However, in order to use the benefits of the standard stable MPC formulation proposed in [34], a non-minimal state-space representation can be used to derive the MPC control law.

In the proposed dead-time process formulation, it is necessary to predict the process output after the dead-time only. Thus, the predicted output vector is defined as

$$\mathbf{y}(k+d|k) = [y_1(k+d_1|k) \; y_2(k+d_2|k) \; \dots \; y_{ny}(k+d_{ny}|k)]^T$$

where d_1, d_2 ... d_{ny} are the effective delay of each output. If the noise is not considered at this point, the regressive model can be rewritten in terms of a difference equation as follows

$$\mathbf{y}(k+d|k) = -\sum_{l=1}^{na} A_l \mathbf{y}(k+d-l) + \sum_{m=1}^{nb} B_m \mathbf{u}(k-m)$$

where $A_l \in \mathbb{R}^{n_y \times ny}$ and $B_m \in \mathbb{R}^{n_y \times nu}$. It should be noticed that each regressive model has a order (na_i) in order that $na = \max(na_i)$, $i = 1, \dots, ny$. Hence, the predicted output realigned form the states will be

$$\mathbf{x}(k+d|k) = \left[\mathbf{x}_y(k+d|k)^T \;\; \mathbf{x}_{\Delta u}(k+d|k)^T\right]^T$$

where

$$\mathbf{x}_y(k+d|k) = \begin{bmatrix} \mathbf{y}(k+d|k) \\ \mathbf{y}(k-1+d|k) \\ \vdots \\ \mathbf{y}(k-na+d|k) \end{bmatrix} \in \mathbb{R}^{(na+1)ny};$$

$$\mathbf{x}_{\Delta u}(k+d|k) = \begin{bmatrix} \Delta\mathbf{u}(k-1) \\ \Delta\mathbf{u}(k-2) \\ \vdots \\ \Delta\mathbf{u}(k-nb+1) \end{bmatrix} \in \mathbb{R}^{(nb-1)nu}.$$

This state partition is convenient in order to separate the predicted output states ($\mathbf{x}_y$) from past control signal ones ($\mathbf{x}_{\Delta u}$). Finally, including the integral action to consider the disturbance model of the CARIMA representation 3.1, the nominal state-space model becomes

$$\begin{aligned} \mathbf{x}(k+1+d|k) &= \mathbf{A}\mathbf{x}(k+d|k) + \mathbf{B}\Delta\mathbf{u}(k), \\ \mathbf{y}(k+d|k) &= \mathbf{C}\mathbf{x}(k+d|k) \end{aligned} \tag{3.2}$$

with

$$\mathbf{A} = \begin{bmatrix} A_y & A_{\Delta u} \\ 0 & \underline{I} \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} B_{\Delta u} \\ \bar{I} \end{bmatrix}, \ \mathbf{C} = [C_y \ C_{\Delta u}],$$

$$A_y = \begin{bmatrix} I_{ny} - A_1 & A_1 - A_2 & \dots & A_{na-1} - A_{na} & A_{na} \\ I_{ny} & 0 & \dots & 0 & 0 \\ 0 & I_{ny} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & I_{ny} & 0 \end{bmatrix},$$

$$A_{\Delta u} = \begin{bmatrix} B_2 & \dots & B_{nb-1} & B_{nb} \\ 0 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 0 \end{bmatrix}, \ B_{\Delta u} = \begin{bmatrix} B_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$C_y = \begin{bmatrix} I_{ny} & 0 & \dots & 0 \end{bmatrix}, \ C_{\Delta u} = \begin{bmatrix} 0 & 0 & \dots & 0 \end{bmatrix},$$

$$\underline{I} = \begin{bmatrix} 0 & \dots & 0 & 0 \\ I_{nu} & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & I_{nu} & 0 \end{bmatrix}, \ \bar{I} = \begin{bmatrix} I_{nu} \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Observe that the matrices **A** and **C** are partitioned in $A_{\Delta u}$, A_y, $C_{\Delta u}$ and C_y. Each block can be identified by its dimension where:

$$\begin{aligned} &A_{\Delta u} \in \mathbb{R}^{(na+1)ny \times (nb-1)nu},\ A_y \in \mathbb{R}^{(na+1)ny \times (na+1)ny}, \\ &B_{\Delta u} \in \mathbb{R}^{(na+1)ny \times nu},\ C_y \in \mathbb{R}^{ny \times (na+1)ny}, \\ &C_{\Delta u} \in \mathbb{R}^{ny \times (nb-1)nu},\ \underline{I} \in \mathbb{R}^{(nb-1)nu \times (nb-1)nu}, \\ &\bar{I} \in \mathbb{R}^{(nb-1)nu \times nu}. \end{aligned}$$

3.3 State-space MPC formulation for delayed systems

Consider that the output reference ($\mathbf{y}_r(k)$) is constant over the prediction horizon in order that the complete future state reference is

$$\mathbf{w}(k+i|k) = \left[\mathbf{w}_y(k+i|k)^T \ \ \mathbf{w}_{\Delta u}(k+i|k)^T\right]^T$$

with

$$\begin{aligned} \mathbf{w}_y(k+i|k) &= [\ \mathbf{y}_r(k)\ \mathbf{y}_r(k)\ \dots\ \mathbf{y}_r(k)\ ]^T \in \mathbb{R}^{(na+1)ny}; \\ \mathbf{w}_{\Delta u}(k+i|k) &= [\ 0\ 0\ \dots\ 0\ ]^T \in \mathbb{R}^{(nb-1)nu}. \end{aligned}$$

Moreover, it is considered step-like reference changes applied in steady-state in such way that $w(k+i|k)$ is fixed during a sufficient large period.

The proposed MPC cost function [1] is

$$J_k = \sum_{i=d+1}^{N+d-1} \left\|\mathbf{w}_y(k+i|k) - \mathbf{y}(k+i|k)\right\|_{Q_\delta}^2 + \sum_{i=0}^{N-1} \|\Delta\mathbf{u}(k+i|k)\|_R^2 + \|\mathbf{w}(k+N+d|k)) - \mathbf{x}(k+N+d|k)\|_P^2 \tag{3.3}$$

[1] Norm notation: $\|\cdot\|_X^2 = (\cdot)^T X(\cdot)$

where N is the prediction and control horizon, $Q_\delta > 0$ is the error weighting, $R > 0$ is the control weighting and $P \geq 0$ is a terminal weighting. N, Q_δ and R are tuning parameters and P must be used in order to guarantee stability. Note that, because of the process dead-time, the first point in the horizon is $d+1$ [41].

The Eq. (3.3) function can be written in a vectorial form as follows

$$J_k = [\mathscr{W}(k) - \mathscr{X}(k)]^T \mathscr{Q}[\mathscr{W}(k) - \mathscr{X}(k)] + \mathscr{U}(k)^T \mathbb{R} \mathscr{U}(k) \tag{3.4}$$

where $\mathscr{U}(k)$, $\mathscr{X}(k)$ and $\mathscr{W}(k)$ are

$$\begin{aligned}
\mathscr{U}(k) &= [\Delta\mathbf{u}(k|k)^T \ \Delta\mathbf{u}(k+1|k)^T \ \ldots \ \Delta\mathbf{u}(k+N-1|k)^T]^T, \\
\mathscr{X}(k) &= [\mathbf{x}(k+d+1|k)^T \ \mathbf{x}(k+d+2|k)^T \ \ldots \ \mathbf{x}(k+d+N|k)^T]^T, \\
\mathscr{W}(k) &= [\mathbf{w}(k+d+1|k)^T \ \mathbf{w}(k+d+2|k)^T \ \ldots \ \mathbf{w}(k+d+N|k)^T]^T
\end{aligned}$$

and the augmented weightings matrices are

$$\begin{aligned}
\mathscr{Q} &= \mathrm{diag}(\mathbf{C}^T Q_\delta \mathbf{C}, \mathbf{C}^T Q_\delta \mathbf{C}, \ldots, \mathbf{C}^T Q_\delta \mathbf{C}, P), \\
\mathbb{R} &= \mathrm{diag}(R, R, \ldots, R).
\end{aligned}$$

Now, as the predicted states and the predicted output can be directly related to $\mathbf{x}(k+d|k)$ by

$$\mathscr{X}(k) = \mathscr{A}\mathbf{x}(k+d|k) + \mathscr{B}\mathscr{U}(k) \tag{3.5}$$

where

$$\mathscr{A} = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}^2 \\ \vdots \\ \mathbf{A}^N \end{bmatrix}, \ \mathscr{B} = \begin{bmatrix} \mathbf{B} & 0 & \ldots & 0 \\ \mathbf{AB} & \mathbf{B} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{N-1}\mathbf{B} & \mathbf{A}^{N-2}\mathbf{B} & \ldots & \mathbf{B} \end{bmatrix},$$

it is possible to rewrite Eq. (3.4) as a function of $\mathbf{x}(k+d|k)$, $\mathscr{U}(k)$ and $\mathscr{X}(k)$. In this case, by replacing Eq. (3.5) in Eq. (3.4) it is obtained

$$J_k = \frac{1}{2}\mathscr{U}(k)^T H \mathscr{U}(k) + b\,\mathscr{U}(k) + f_0$$

with

$$\begin{aligned}
H &= 2[\mathscr{B}^T \mathscr{Q} \mathscr{B} + \mathbb{R}] \\
b &= -2[\mathscr{W}^T \mathscr{Q} \mathscr{B} - x(k+d|k)^T \mathscr{A}^T \mathscr{Q} \mathscr{B}] \\
f_0 &= [\mathscr{W} - \mathscr{A}x(k+d|k)]^T \mathscr{Q}[\mathscr{W} - \mathscr{A}x(k+d|k)].
\end{aligned}$$

As consequence, the optimal control signal is obtained by

$$\frac{\partial J}{\partial \mathscr{U}} = 0 \Rightarrow \mathscr{U}^* = -H^{-1}b^T = \mathscr{K}[\mathscr{W} - \mathscr{F}] \tag{3.6}$$

where

$$\begin{aligned}
\mathscr{K} &= [\mathscr{B}^T \mathscr{Q} \mathscr{B} + \mathbb{R}]^{-1} \mathscr{B}^T \mathscr{Q}^T \\
\mathscr{F} &= \mathscr{A}\mathbf{x}(k+d|k).
\end{aligned}$$

Finally, due to the receding horizon principle, the control signal should be

$$\Delta\mathbf{u}(k) = [\, \underbrace{1\ 1\ \dots\ 1\ 1}_{n_u} | \underbrace{0\ 0\ \dots\ 0\ 0}_{(N-1)\times n_u}]\, \mathscr{U}(k).$$

It is important to notice that: (i) $\mathscr{F} = \mathscr{A}\mathbf{x}(k+d|k)$ is the free response in terms of the state-space representation and (ii) $\mathbf{x}(k+d|k)$, which is composed by $\mathbf{y}(k+d|k)$ until $\mathbf{y}(k-na+d|k)$ and $\Delta\mathbf{u}(k-1)$ until $\Delta\mathbf{u}(k-nb+1)$, will be multiplied by the gain $\mathscr{K}\mathscr{A}$.

3.4 Nominal stability

As it is well-known in MPC if the terminal cost function is appropriately chosen, then nominal stability is guaranteed [34]. This allows to use the filtered Smith predictor structure to compute $\mathbf{x}(k+d|k)$ with the aim of providing robustness to the predictive controller.

Theorem. 1 *Consider the matrices* $\mathbf{A}$, $\mathbf{B}$ *and* $\mathbf{C}$ *of the augmented state-space representation and the tuning matrices* $R > 0$ *and* $Q_\delta > 0$ *used in Eq. 3.3. Let the gain K and the semi-positive definite matrix P obtained such that*

$$(\mathbf{A}+\mathbf{B}K)^T P(\mathbf{A}+\mathbf{B}K) - P + K^T RK + \mathbf{C}^T Q_\delta \mathbf{C} = 0. \tag{3.7}$$

If there is a solution for this Ricatti equation in order that P is the terminal weighting used in Eq. (3.3)*, then the MPC control law nominally stabilizes the system represented by* (3.2)*.*

Remark. 1 *If* $P = \mathbf{C}^T Q_\delta \mathbf{C}$*, the proposed formulation turns into the DTCGPC presented in [41] because* J_k *is the same cost function as the DTCGPC one.*

3.5 Prediction up to k+d and robustness

Using the previous results, it is already known that the nominal closed loop system is stable. For the robustness analysis, it is interesting to derive a block diagram representation of the controller structure. The predictor structure is analyzed first.

By considering that $e_i(k+i|k) = 0,\ i > 1$, it is possible to obtain $\mathbf{y}(k+d|k)$ recursively from Eq. (3.1). This approach, used in the GPC strategy, is called optimal predictor. In order to improve robustness by filtering the predicted output, it is proposed to use a filtered Smith predictor formulation. This predictor structure is presented in Fig. 3.1 where $\mathbf{F_r}(z)$ is a diagonal filter $(\mathrm{diag}[f_{r_i}(z)],\ i = 1,..,n_y)$ and $\mathbf{G_n}(z)$ is the nominal transfer function without the effective dead time (d_i). If the process is unstable or integrative, $\mathbf{F_r}(z)$ should be obtained in order that $\mathbf{S}(z) = \mathbf{G_n}(z) - \mathbf{F_r}(z)\mathbf{P_n}(z)$ does not have unstable poles as discussed in [41]. The stability of $\mathbf{S}(z)$ is necessary to guarantee internal stability of the predictor.

Now, to complete the block diagram, it is computed the control action as a function of $\mathbf{y}(k+d|k)$ and $\mathbf{w_y}(k+d+N|k)$. As it was already pointed out, $\mathbf{x}(k+d|k)$ is composed by $\mathbf{y}(k+d|k), ..., \mathbf{y}(k+d-na|k)$ and $\Delta\mathbf{u}(k-1), ..., \Delta\mathbf{u}(k-1-nb)$. As consequence, the optimal unconstrained control law (see Eq. (3.6)) can be expressed by

$$\begin{aligned}\Delta\mathbf{u}(k) =& \mathbf{L_w}(z^{-1})\mathbf{w_y}(k+d+N|k) - \mathbf{L_u}(z^{-1})\Delta\mathbf{u}(k-1)\\ &- \mathbf{L_y}(z^{-1})\mathbf{y}(k+d|k)\end{aligned} \tag{3.8}$$

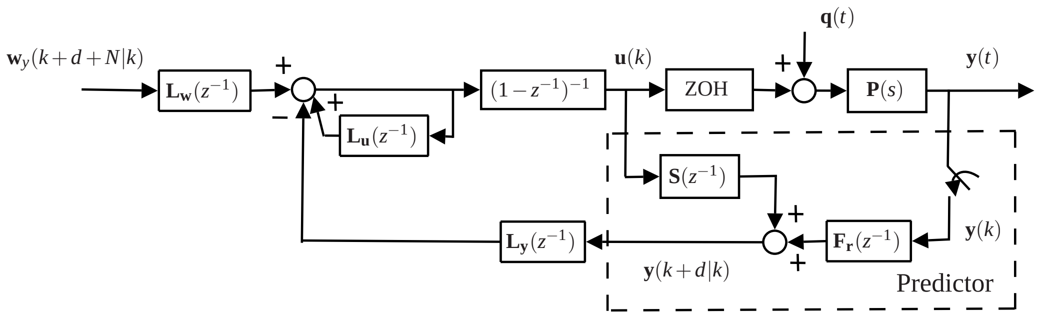

Figure 3.1: Complete control structure diagram

or simply

$$\mathbf{u}(k) = \mathbf{F}(z^{-1})\mathbf{w_y}(k+d+N|k) - \mathbf{C}(z^{-1})\mathbf{y}(k+d|k)$$

where $\mathbf{F}(z^{-1}) = (1+z^{-1})(\mathbf{I}+\mathbf{L_u}(z^{-1}))^{-1}\mathbf{L_w}(z^{-1})$ and $\mathbf{C}(z^{-1}) = (1+z^{-1})(\mathbf{I}+\mathbf{L_u}(z^{-1}))^{-1}\mathbf{L_y}(z^{-1})$. The complete control scheme in terms of transfer function is presented in Fig. 3.1. An important advantage of this formulation comes from the fact that it is possible to establish a relationship between $\mathbf{F_r}(z)$ and robustness as it is going to be analyzed in the following.

3.6 Robustness analysis

The robustness filter should be used to obtain a compromise between robustness and disturbance rejection. As presented in [44], it is possible either to perform an analytical approach or to use intuition as usually done in industrial practice.

3.6.1 Tuning filter using an unstructured uncertainty model

In this report it is used an additive unstructured description of the uncertainty is considered in such way that the real plant $\mathbf{P}$ is in a vicinity of the nominal plant $\mathbf{P_n}$, that is

$$\mathbf{P}(z) = \mathbf{P}_n(z) + \delta\mathbf{P}(z).$$

In general $\delta\mathbf{P}(z)$ can be written as [52]

$$\delta\mathbf{P}(z) = \mathbf{W_2}(z)\Delta(z)\mathbf{W_1}(z),\ \bar{\sigma}(\Delta(z)) < 1,\ \forall\omega \in (0,\pi/T_s)$$

where for this case $\Delta(z)$ is a full matrix, $\bar{\sigma}(X)$ denotes the maximum singular value of X, T_s is the sampling period and $\mathbf{W_1}(z)$ and $\mathbf{W_2}(z)$ are two stable matrix transfer function that characterize spatial and frequency structure of the uncertainty.

The characteristic equation in the presence of uncertainty is

$$\det[\mathbf{I}+\mathbf{C}(\mathbf{G_n}+\mathbf{F_r}\mathbf{W_2}\Delta(z)\mathbf{W_1})] = 0.$$

Using that

$$\begin{aligned}&\det[\mathbf{I}+\mathbf{C}(\mathbf{G_n}+\mathbf{F_r}\mathbf{W_2}\Delta(z)\mathbf{W_1})] = \\ &\det[(\mathbf{I}+\mathbf{C}\mathbf{G_n})]\det[\mathbf{I}+(\mathbf{I}+\mathbf{C}\mathbf{G_n})^{-1}\mathbf{C}\mathbf{F_r}\mathbf{W_2}\Delta(z)\mathbf{W_1})]\end{aligned}$$

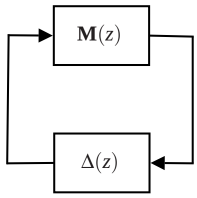

Figure 3.2: Modified plant for robustness analysis

where $\det[\mathbf{I}+\mathbf{CGn}]$ is the nominal characteristic equation which has stable poles only. In this case, a new equivalent plant for robustness analysis is presented in Fig. 3.2 where $\mathbf{M} = -\mathbf{W_1C}(\mathbf{I}+\mathbf{CGn})^{-1}\mathbf{F_rW_2} = -\mathbf{W_1M_0F_rW_2}$ is the equivalent plant.

Thus, assuming $\bar{\sigma}(\Delta(z)) < 1$, it can be concluded that if

$$\bar{\sigma}(\mathbf{M}(e^{j\omega})) < 1,\ \forall\omega \in [0, \pi/T_s],$$

robust stability is guaranteed [52]. It is also possible to state that

$$\bar{\sigma}(\mathbf{M}) \leq \bar{\sigma}(\mathbf{M_0})\bar{\sigma}(\mathbf{F_r})\bar{\sigma}(\mathbf{W_1}).\bar{\sigma}(\mathbf{W_2}) \tag{3.9}$$

In order words, the robustness filter must be chosen in order to verify

$$\bar{\sigma}(\mathbf{F_r}(e^{j\omega})) \leq \frac{1}{\bar{\sigma}(\mathbf{W_1}(e^{j\omega}))\bar{\sigma}(\mathbf{W_2}(e^{j\omega}))\bar{\sigma}(\mathbf{M_0}(e^{j\omega}))} \tag{3.10}$$

$\forall\omega \in [0, \pi/T_s]$.

3.6.2 Tuning filter in industrial practice

One of the most important problems in practice is the derivation of simple rules that can be used by operators to tune the multivariable controller. In general, it is very difficult to model the uncertainties in industrial processes, but it is possible to increase robustness by reducing the singular value of $\mathbf{F_r}(z)$ as can be verifies from Eq. (3.9). A key aspect is that the singular value of $\mathbf{F_r}(z)$ is determined by the magnitude of $f_{ri}(z)$ because $\mathbf{F_r}$ is diagonal transfer function. Thus, the *ny* SISO filters ($f_{ri}(z)$) can be directly tuned as suggested in [42].

It is important to remark that this approach is useful because the robustness filter tuning parameters can be defined to have provide an intuitive interpretation. For instance, if the process is stable, it is possible to choose a second order low pass filter for each output in the form:

$$f_{ri}(z^{-1}) = \left[\frac{1-\beta_i}{1-\beta_i z^{-1}}\right]^2 \tag{3.11}$$

where β_i is the robustness tuning parameter. It can be readily verified from Eq. (3.10) that it is possible to increase robustness by using a greater value of β. On the other hand, the greater is the value of β, the greater is the filtering action and the slower is the disturbance rejection response.

3.7 Simulation Example

To illustrate the properties of the MPC with predictions based on a filtered Smith predictor, this will be applied to the following MIMO delayed system. In this case, the nature of the delay may be communication delay as well as the inner delay of the plant.

$$G(s) = \begin{bmatrix} \frac{9.02}{(2.57s+1)^2}e^{-\theta_{11}s} & \frac{10.01}{2s+1}e^{-\theta_{12}s} \\ \\ \frac{0.495}{42s+1}e^{-\theta_{21}s} & \frac{6.34}{72s+1}e^{-\theta_{22}s} \end{bmatrix},$$

Due to the random nature of the communication network, the delays are assumed to be within the following ranges:

$$4.10 \leq \theta_{11} \leq 4.5, \qquad 5 \leq \theta_{12} \leq 6.8$$
$$5.50 \leq \theta_{21} \leq 6.6, \qquad 4.3 \leq \theta_{22} \leq 5.3.$$

and the nominal dead times were chosen to be $\theta_{11n} = 4.3$, $\theta_{12n} = 5.9$, $\theta_{21n} = 6$ and $\theta_{22n} = 4.8$. The dead-time free model has a Right Half-Plane Zero which may cause internal stability problems. All simulations were performed with the same weighting matrices $Q_\delta = I$, $R = 0.1\text{diag}[9.02^2, 6.34^2]$ and sampling period $T_s = 0.1$s.

The first simulation, presented in Fig. 3.3, is used to illustrate that the state-space DTCMPC and the DTCGPC nominal set-point tracking performances can be similarly good. The simulation test consist of two step changes with amplitude 100 and 10 for outputs 1 and 2 respectively. In these simulations it was used $N = 50$ (prediction horizon) and $N_u = 20$ (control horizon) for the DTCGPC and $N = N_u = 20$ for the DTCMPC. From Fig. 3.3 it can be observed that the DTCMPC and the DTCGPC output responses are similar despite the fact that it was used a prediction horizon more than two times greater in the DTCGPC case. Moreover, if it is tried to use $N = N_u = 20$ in the DTCGPC strategy, the output response is unstable as shown in Fig. 3.4. This unstable behavior is a typical situation where the prediction horizon was not appropriately chosen to control a RHPZ [15].

The increase in the robustness is provided by the filter $F_r(z)$ are illustrated in Figs. 3.5 and 3.6. In both cases, the real dead-time was set to be their maximum values ($\theta_{11r} = 4.5$, $\theta_{12r} = 6.8$, $\theta_{21r} = 6.6$ and $\theta_{22r} = 5.3$) and two input disturbances $\mathbf{q}_1(t) = [-10\ 0]^T$ and $\mathbf{q}_2(t) = [0\ -1]^T$ were added at 150 and 220 seconds respectively. By considering the filter presented in Eq. (3.11), the case without filter ($\beta_1 = \beta_2 = 0$) is presented in Fig. 3.5 and the responses obtained for $F_r(z)$ with $\beta_1 = \beta_2 = 0.9$ are shown in Fig. 3.6. The output response for the case with $\beta_1 = \beta_2 = 0$ is unstable due to the dead-time estimation error. On the other hand the correct tuning of β allows to obtain a stable behavior as shown in Fig. 3.6.

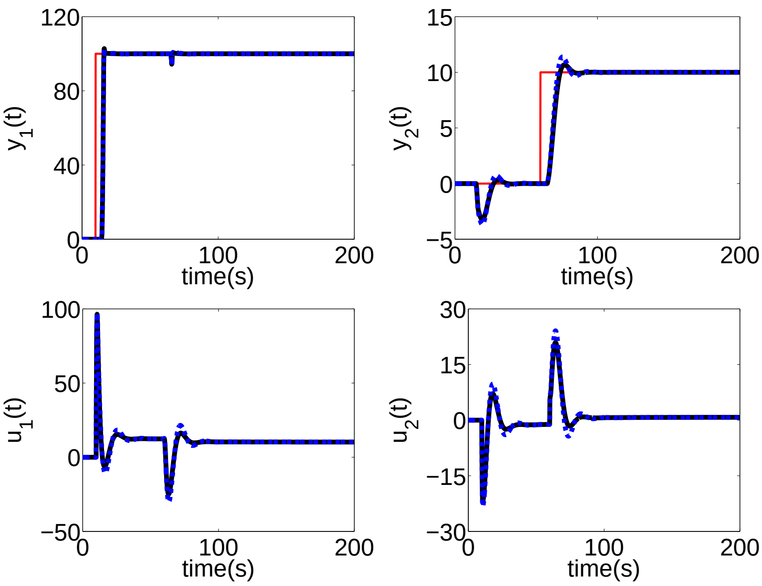

Figure 3.3: Output response and control signal: DTCMPC (solid) and DTCGPC (Dashed). $N^{DTCGPC} = 50$, $N_u^{DTCGPC} = 20$ and $N^{DTCMPC} = 20$.

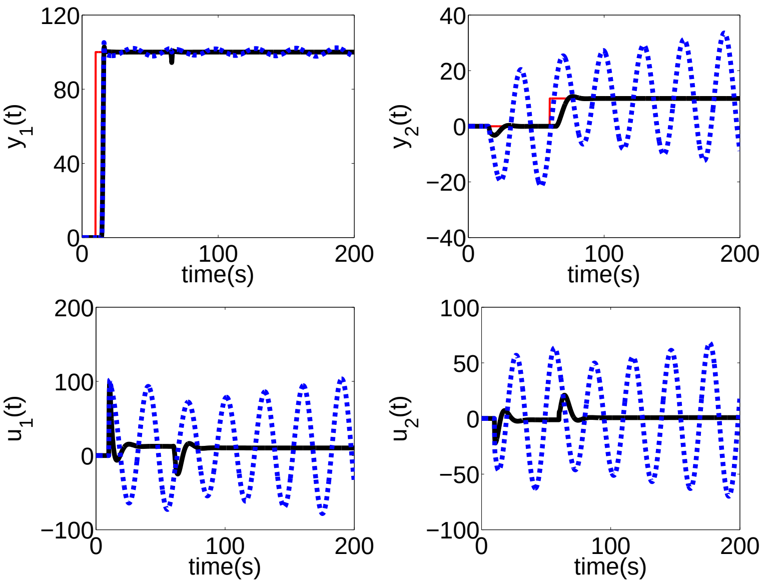

Figure 3.4: Output response and control signal: DTCMPC (solid) and DTCGPC (dashed). $N^{DTCGPC} = 20$, $N_u^{DTCGPC} = 20$ and $N^{DTCMPC} = 20$.

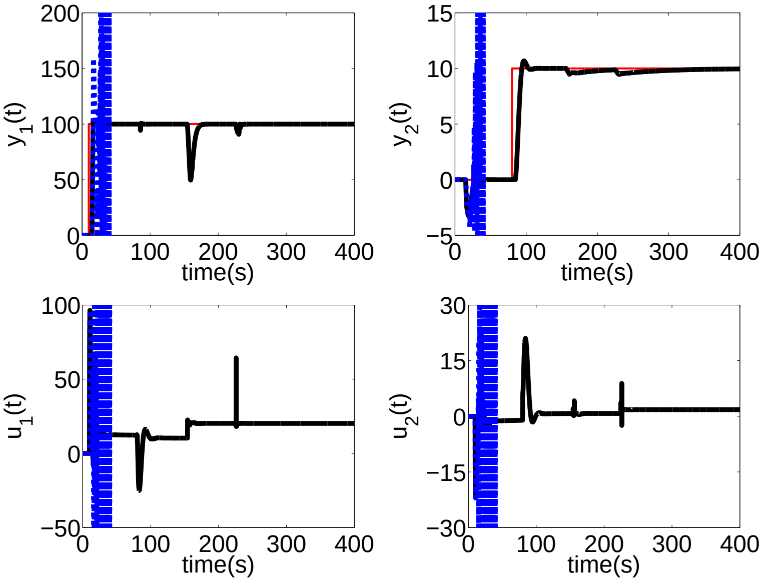

Figure 3.5: Output response and control signal for the DTCMPC with $\beta_1 = \beta_2 = 0$: nominal case (solid) and real case (dashed).

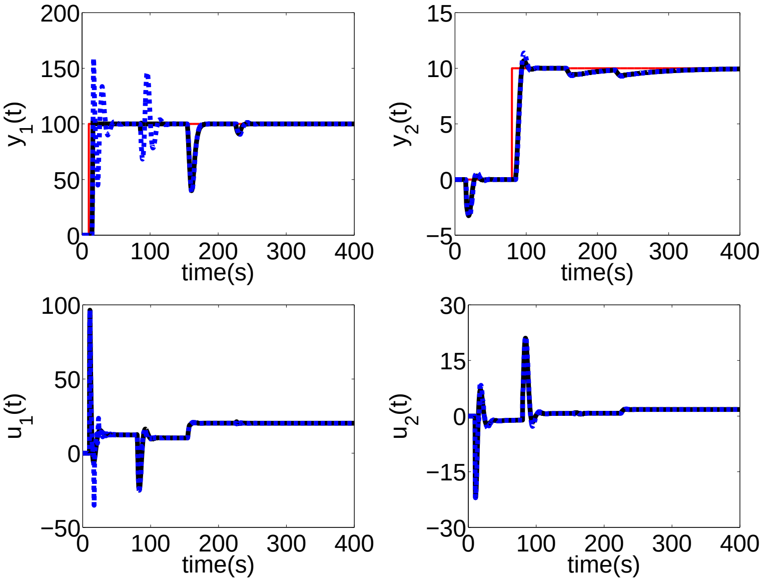

Figure 3.6: Output response and control signal for the DTCMPC with $\beta_1 = \beta_2 = 0.9$: nominal case (solid) and real case (dashed).

Bibliography

[1] M. Alamir. A Framework for Monitoring Control Updating Period In Real-Time NMPC Schemes. In *International Workshop on Assessment and Future Directions of NMPC*, 2008.

[2] M. Alimi, N. Derbel, A. Ouali, and M. Kamoun. A hierarchical Control Algorithm Applied to the Optimization of Large Scale Time-Delay Systems. *Proceedings of the 32nd Conference on Decision and Control*, San Antonio, Texas, 1993.

[3] M. Alimi, N. Derbel, A. Ouali, and M. Kamoun. Hierarchical Control Applied to a large Scale Time-Delay model of a river. *Proceedings of the 2nd IEEE Conference on Control Applications*, September 13-16, Vancouver, Canada, 1993.

[4] R. D'Andrea and G.E. Dullerud. Distributed Control of Spatially Invariant Systems. *IEEE Transactions on Automatic Control*, 48:1478–1495, 2003.

[5] A. Bemporad. Predictive control of teleoperated constrained systems with unbounded communication delays. In *Proceedings of the 37th IEEE Conference on Decision and Control*, 1998.

[6] R. Blind and F. Algower. A Controller Design for Networked Control Systems with Random Delays via the Jump Linear System Approach, which Reduces the Effects of the Delay. *Proceedings of the European Control Conference*, Budapest, Hungary, 2009.

[7] E.F. Camacho and C. Bordons. *Model Predictive Control*. Springer, 2004.

[8] R.S. Chandra, C. Langbort and R. D'Andrea, Distributed Control Design with Robustness to Small Time Delays. *Systems and Control Letters*, 2009.

[9] W.H. Chen, D.J. Ballance, and J. O'Reilly. Model predictive control of nonlinear systems: computational burden and stability. *IEE Proceedings-Control Theory and Applications*, 147(4):387–394, 2000.

[10] A. Cuenca, P. García, K. Årzén, and P. Albertos. A Predictor-Observer for a Networked Control System with Time-Varying Delays and Non-Uniform Sampling. *Proceedings of the European Control Conference*, Budapest, Hungary, 2009.

[11] M. Diehl, R. Findeisen, S. Schwarzkopf, I. Uslu, H. G. Bock, and E. D. Gilles. An Effective Algorithm for Optimization in Nonlinear Model Predictive Control of Large-Scale Systems, 2002.

[12] L. Dritsas, A. Tzes, and G. Nikolakopoulos. Robust Stability analysis of Networked Systems with Varying Delays. *Proceedings of the European Control Conference 2009*, Budapest, Hungary, 2009.

[13] R. Findeisen and F. Allgöwer. Computational Delay in Nonlinear Model Predictive Control. In *Proceedings of the International Symposium on Advanced Control of Chemical Processes (ADCHEM)*, volume 3, 2004.

[14] R Findeisen and P. Varutti. Stabilizing nonlinear predictive control over nondeterministic communication networks. In *Nonlinear Model Predictive Control: Towards New Challenging Applications*, volume 384 of *Lecture Notes in Control and Information Sciences*, pages 167–179. Springer, Berlin, Heidelberg, 2009.

[15] W. Garcia-Gabin and E.F. Camacho. Application of multivariable GPC to a four tank process with unstable transmission zeros. In *Proceedings of the 2002 International Conference on Control Applications*, volume 2, pages 645–650 vol.2, 2002.

[16] E. Gershon and U. Shacked. Robust H_∞ Output-Feedback Control of State-Multiplicative Stochastic Systems with Delay. *Proceedings of the European Control Conference, Budapest*, Hungary, 2009.

[17] W.P.M.H. Heemels, A.R. Teel, N. van de Wouw, and D. Nesic. Networked Control Systems with Communication Constraints: Tradeoffs Between Transmission Intervals and Delays. *Proceedings of the European Control Conference*, Budapest, Hungary, 2009.

[18] J.P. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of Recent Results in Networked Control Systems. *Proceedings of the IEEE*, 95(1):138–162, 2007.

[19] S. Hierche, T. Matiakis, and M. Buss. A Distributed Controller Approach for Delay-Independent Stability of Networked Control System. *Automatica*, 45, 2009.

[20] W. Hu, G-P. Liu, and D.Rees. Networked Predictive Control Over the Internet Using Round-Trip Delay Measurement. *IEEE Transactions on Instrumentation and Measurement*, 57(10):2231–2241, 2008.

[21] G. Huang and S. Wang. Use of uncertainty polytope to describe constraint processes with uncertain time-delay for robust model predictive control applications. *ISA Transactions*, 48:503–511, 2009.

[22] Jian Huang, Yongji Wang, Shuang-Hua Yang, and Qi Xu. Robust stability conditions for remote SISO DMC controller in networked control systems. *Journal of Process Control*, 19(5):743–750, 2009.

[23] S.C. Jeong and P. Park. Constrained MPC algorithm for uncertain time-varying systems with state-delay. *IEEE Transactions on Automatic Control*, 50(2):257–263, 2005.

[24] T. Kaczorek. Stability of positive continuous-time Linear Systems with Delays. *Proceedings of the European Control Conference 2009*, Budapest, Hungary, 2009.

[25] H.R. Karimi, N.A. Diffie, and S. Dashkovskiy. Local H_∞ Control for Production Networks of Autonomous Work Systems with Time-Varing Delay. *Proceedings of the European Control Conference*, Budapest, Hungary, 2009.

[26] D. Kim, Y.S. Lee, W.H. Kwona, and H.S. Park. Maximum allowable delay bounds of networked control systems. , *Control Engineering Practice*, 11:1301–1313, 2003.

[27] M.V. Kothare, V. Balakrishnan, and M. Morari. Robust constrained model predictive control using linear matrix inequalities,. *Automatica*, 32(10):1361–1379, 1996.

[28] W.H. Kwon, Y.S. Lee, and S.H. Han. General receding horizon control for linear time-delay systems. *Automatica*, 40(9):1603–1611, 2004.

[29] W.H. Kwon, J.W. Kang, Y.S. Lee, and Y.S. Moon. A simple receding horizon control for state delayed systems and its stability criterion. *Journal of Process Control*, 13(6):539–551, 2003.

[30] C. Langbort, R.S. Chandra, and R. DÁndrea. Distributed Control Design for Systems Interconnected Over an Arbitrary Graph. *IEEE Transactions on Automatic Control*, 49:1502–1519, 2004.

[31] D. Limon, T. Alamo, and E. F. Camacho. Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties. In *Proceedings of the 41st IEEE CDC*, pages 4619–4624, 2002.

[32] R. Luck and A. Ray. An Observer-based Compensator for Distributed Delays. *Automatica*, 26(5):903–908, 1990.

[33] B. Marinescu and H. Bourles. Robust state-predictive control with separation property: A reduced-state design for control systems with non-equal time delays. *Automatica*, 36(4):555–562, 2000.

[34] 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, 2000.

[35] G. Meinsma, M. Fu, and T. Iwasaki, Robustness of the Stability of Feedback Systems with Respect to Small Time Delays. *Systems and Control Letters*, 36:131–134, 1999.

[36] J. Melin, M. Jungers, J. Daafouz, and C. Iung. On Analysis of Performance for Digitally Controlled and Time-Varying Delayed Systems. , *Proceedings of the European Control Conference*, Budapest, Hungary, 2009.

[37] P. Mendez Monroy, and H. Benitez Perez. Fuzzy Control with Time Delay for Networked Control System. *Proceedings of the European Control Conference*, Budapest, Hungary, 2009.

[38] P. Millan, I. Jurado, C. Vivas, and F.R. Rubio. Networked predictive control of systems with data dropouts. In *Proceedings of the 47th IEEE Conference on Decision and Control*, 2008.

[39] L.A. Montestruque and P.J. Antsaklis. On the model-based control of networked systems. *Automatica*, 39(10):1837 – 1843, 2003.

[40] J. Nilsson, B. Bernhardsson, and B. Wittenmark. Stochastic analysis and control of real-time systems with random time delays. *Automatica*, 34(1):57–64, 1998.

[41] J.E. Normey-Rico and E.F. Camacho. *Control of Dead-time Processes*. Springer-Verlag, London, 2007.

[42] J.E. Normey-Rico and E.F. Camacho. Unified approach for robust dead-time compensator design. *Journal of Process Control*, 19(1):38–47, 2009.

[43] J.E. Normey-Rico and E.F. Camacho. Robustness effect of a prefilter in smith predictor based generalised predictive control. *IEE Proceedings: Control Theory and Applications*, 146(2):179–185, 1999.

[44] J.E. Normey-Rico and E.F. Camacho. Multivariable generalised predictive controller based on the smith predictor. *Control Theory and Applications, IEE Proceedings -*, 147(5):538–546, Sept. 2000.

[45] G. Pin and T. Parisini. Stabilization of networked control systems by nonlinear Model Predictive Control: a set invariance approach. In *Nonlinear Model Predictive Control: Towards New Challenging Applications*, volume 384 of *Lecture Notes in Control and Information Sciences*. Springer, Berlin, Heidelberg, 2009.

[46] I.G. Polushin, P.X. Liu, and C.-H. Lung. On the model-based approach to nonlinear networked control systems. *Automatica*, 44(9):2409–2414, 2008.

[47] M. Razzaghi and M. Razzaghi. Taylor series analysis of time-varying multi-delay systems. *International Journal of Control*, 50, 1980.

[48] C.J. Rapson. *Spatially Distributed Control -Heat Conduction in a Rod-*. Master Thesis. Luleå University of Technology, 2008.

[49] L. Samaranayake. Delay Compensation, Design and Simulation of Controllers for Distributed Control Systems. *Proc. of the 1st International Conference on Industrial and Information Systems ICIIS*. Sri Lanka, 2006.

[50] T.L.M. Santos, J.E. Normey-Rico, and D. Limon. Robust Model Predictive Controller with Terminal Weighting for Multivariable Dead-Time Processes. In *Proceedings of the 2009 IFAC Workshop on Time Delay Systems*, 2009.

[51] Yu-Jing Shi, Tian-You Chai, Hong Wang, and Chun-Yi Su. Delay-dependent Robust Model Predictive Control for Time-delay Systems with Input Constraints. In *Proceedings of the 2009 American Control Conference*, 2009.

[52] S. Skogestad and I. Postlethwaite. *Multivariable feedback control. Analysis and design.* Wiley, 1996.

[53] D. Srinivasagupta, H. Schättler, and B. Joseph. Time-stamped model predictive control: an algorithm for control of processes with random delays. *Computers & Chemical Engineering*, 28(8):1337 – 1346, 2004.

[54] P.L. Tang and C.W. de Silva. Compensation for transmission delays in an ethernet-based control network using variable-horizon predictive control. *IEEE Transactions on Control Systems Technology*, 14(4):707 – 718, 2006.

[55] P. Varutti, B. Kern, T. Faulwasser, and R. Findeisen. Event-based model predictive control for networked control systems. In *Proceedings of the 48th IEEE Conference on Decision and Control*, 2009.

[56] P. Varutti and R.D. Findeisen. Compensating Network Delays and Information Loss by Predictive Control Methods. *Proc. of the 1st International Conference on Industrial and Information Systems ICIIS*, Sri Lanka, 2006.

[57] L. Würth, R. Hannemann, J. Kadam, and W. Marquardt. Neighboring-Extremal Updates for Nonlinear Model-Predictive Control and Dynamic Real-Time Optimization. *Journal of Process Control*, 2009.

[58] T.C. Yang. Networked control systems: a survey. *IEE Proceedings on Control Theory and Applications*, 153(4):403–412, 2006.

[59] F. Yang and H. Fang. Control structure design of networked control systems based on maximum allowable delay bounds. *Journal of the Franklin Institute*, 346, 2009.

[60] J.K. Yook, D.M. Tilbury, and N.R. Soparkar. A Design Methodology for Distributed Control Systems to Optimize Performance in the Presence of time delay. *International Journal of Control*, 74(1), 2001.

[61] V. Zavala and L. Biegler. The advanced-step NMPC controller: Optimality, Stability and Robustness. *Automatica*, 2008.

[62] W. Zhang, S. Branicky, and S.M. Phillips. Stability of Networked Control Systems. *IEEE Control Systems Magazine*, 21(2):84–99, 2001.