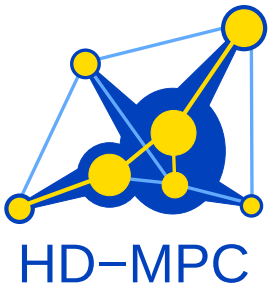
SEVENTH FRAMEWORK PROGRAMME
THEME – ICT
[Information and Communication Technologies]

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



Deliverable Number:	D6.4.1
Deliverable Type:	Report
Contractual Date of Delivery:	March 1, 2010
Actual Date of Delivery:	**March 1, 2010**
Title of Deliverable:	**Report on implementation for selected benchmarks**
Dissemination level:	Public
Workpackage contributing to the Deliverable:	WP6
WP Leader:	USE
Partners:	TUD, EDF, K.U.Leuven, RWTH, USE, UNC, SUPELEC, INOCSA
Author(s):	I. Alvarado, D. Muñoz, D. Limon, M.A. Ridao, J. Espinosa, J. García, F. Valencia, Z. Hidayat, B. De Schutter, R. Babuška, A. Nuñez

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 deliverable describes and analyzes the exercises on the proposed benchmark cases in the period of time M12-M18. These benchmark cases are described in Deliverable D6.3.1 and models and additional information can be found in the Virtual Portal: the heat system, the four tank system, an electric power system, and the chemical benchmark case.
The two-dimensional heat system benchmark is used to compare various decentralized Kalman filters in Chapter 1. The methods that are compared are:

- Centralized Kalman filter (CKF),
- Parallel information filter (PIF),
- Decentralized information filter (DIF),
- Decoupled hierarchical Kalman filter (DHKF),
- Distributed Kalman filter with weighted averaging (DKFWA),
- Distributed Kalman filter with consensus filters (DKFCF),
- Distributed Kalman filter with bipartite fusion graphs (DKFBFG).

Another distributed state estimation scheme (DDKF: Distributed and Decentralized Kalman Filter)is applied to a one-dimensional heat system. The observer performance under additive and structural disturbance is also studied. Then, in Chapter 3, a combined DDKF and MPC formulation is tested on the same benchmark.
Concerning the 4-tanks benchmark, different control approaches have been tested and compared both in simulation and on the real plant. These approaches are:

- Tracking Control. Control that allows changes in the reference.
  - Centralized control for tracking.
  - Decentralized control for tracking. Two MPC for tracking are used, the same as in the previous case, but applied to each subsystem. The pairing procedure between the inputs is done based on the Relative Gain Array. Two examples are done, one with the correct pairing, and the second with the wrong one.
- Regulation controller. To perform the reference changes, one controller for each reference is designed.
  - Centralized control.
  - Distributed control. *Distributed MPC based on a cooperative game*

Finally, the last exercise is related to the electrical power system benchmark and is described in Chapter 5. A centralized MPC is formulated for the control of generation units . Due to different time scale of machines dynamics, a two levels time-response-based hierarchical structure is proposed. The proposed control structure involves the interaction among the centralized MPC and classical voltage and speed regulators.

Part I

Heat conduction and convection benchmark

Chapter 1

Decentralized Kalman filters for the heated plate

1.1 Introduction

This chapter presents a comparison of various decentralized Kalman filters to estimate the discretized states of the heated plate, which is one of the benchmarks within HD-MPC work package WP6. There are seven methods that are being compared, they are:

1. Centralized Kalman filter (CKF),
2. Parallel information filter (PIF),
3. Decentralized information filter (DIF),
4. Decoupled hierarchical Kalman filter (DHKF),
5. Distributed Kalman filter with weighted averaging (DKFWA),
6. Distributed Kalman filter with consensus filters (DKFCF),
7. Distributed Kalman filter with bipartite fusion graphs (DKFBFG).

This chapter is organized as follows. In Section 1.2 we give a concise overview of the seven decentralized Kalman filters considered in the comparison. Next, we briefly present the set-up and model of the heated plate in Section 1.3. The results of the simulations are then given in Section 1.4. Section 1.5 concludes the chapter.

1.2 Short overview of the decentralized Kalman filter methods

We will first briefly describe the methods that will be applied in the comparison; for a more extensive description and details the interested reader is referred to the references cited below.

Given a linear system and its measurement model and a set of noisy measurement data from the system, a *Kalman filter* [6] computes the best estimate of the states for the system. Kalman filters are widely used in many areas of engineering. There are two types of the Kalman filters: the continuous-time Kalman filter and the discrete-time Kalman filter. In this chapter the term Kalman filter refers to the discrete-time version.

In the Kalman filter the measurements can be obtained from several sensors, which form together a sensor network. Each sensor node in the network can measure either the same states or different ones. For our application, we are interested in the second case. The measurements from all sensors at discrete time instant k can be collected into a measurement matrix and used to compute the estimate. In this case, the method is called the Centralized Kalman Filter (CKF). This is the original form introduced in [6].

The Kalman filter has a variant which is called the information filter. The information filter is also often be used in decentralized applications. In the information filter the estimates are computed based on the information update which can be determined in a decentralized way. In the Parallel Information Filter (PIF) proposed by Speyer [11], each sensor node has the global system model and computes the whole state of the system or the global state. The estimates from the individual nodes are then sent to a central processor to be summed stochastically with the estimates from the other nodes in order to obtain the fused estimates.

The Decentralized Information Filter (DIF) was proposed by Rao and Durrant-Whyte [10]. In a similar way as in the PIF, each sensor node in the DIF also estimates the global state. This method was derived by decentralizing the information update computation of the centralized approach to all sensor nodes. By this approach, the estimation errors of the DIF are guaranteed to be equal to the ones obtained by centralized approach, provided that the sensor nodes are all connected.

In the Decoupled Hierarchical Kalman Filter (DHKF) [4, 3], each sensor node has only a part of the global system model. This also means that each node only computes a part of the global state. The estimate is then communicated to the other nodes so that all nodes have the global state.

The Distributed Kalman Filter with Weighted Averaging (DKFWA) was proposed by Alriksson and Rantzer [1]. Similarly to the DIF, each node in the DKFWA also estimates the global state. The difference is that in the DKFWA the Kalman gain is computed only once at the beginning and used for the estimation process. In this way, the computation load at each node is reduced.

The Distributed Kalman Filter with Consensus Filters (DKFCF) was proposed by Olfati-Saber [9]. Basically this method is the same as the DIF but with an additional consensus step. In the consensus step, each node exchanges information with its neighbors so as to equalize the estimates.

The Distributed Kalman Filter with Bipartite Fusion Graphs (DKFBFG) proposed by Kahn and Moura [7] divides the network into some connected partitions, each of which can contain one or more sensors. In each partition, the global state is divided into two parts: the estimated states and the unestimated states. The unestimated states that are needed to compute the estimated states are then obtained from the other partitions.

1.3 Heated plate model and simulation set-up

The model of the heated plate can be expressed as a two-dimensional conduction model. Such models can be found in standard text books about heat transfer and can be written as follows [5, 8]:

$$\frac{\partial T}{\partial t} = \frac{1}{\rho C_{\mathrm{p}}}\left[\kappa\frac{\partial^2 T}{\partial x^2} + \kappa\frac{\partial^2 T}{\partial y^2} + \frac{\dot{Q}_{\mathrm{s}}^{*}(x,y,t)}{K}\right] \quad , \tag{1.1}$$

where T is the temperature of the plate, ρ the density of the plate, C_{p} the heat capacity per unit mass, κ the thermal conductivity, $\dot{Q}_{\mathrm{s}}^{*}$ the heater power per unit area, and x and y are spatial coordinates of length and width respectively, and K is a constant. The K is used to take into account the effect of the spatial discretization.

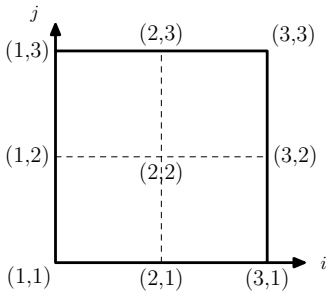


Figure 1.1: Example of the segmentation of the plate for spatial discretization.

For transient analysis it is necessary to have boundary conditions. For our problem, we use convective boundary conditions, which are modeled as Robin or third-kind boundary conditions. The model (1.1) with convective boundary conditions can be written as [5, 8]

$$\frac{\partial T}{\partial t} = \frac{1}{\rho C_\mathrm{p}} \left[\kappa \frac{\partial^2 T}{\partial x^2} + \kappa \frac{\partial^2 T}{\partial y^2} + \frac{\dot{Q}_\mathrm{s}(x,y,t)}{K} + \frac{h}{K}(T_\mathrm{env} - T) \right] \ , \tag{1.2}$$

where T_env is the temperature of the environment and h is the convection coefficient.

To simulate the heated plate, we use a state space model that is discretized in space and time. Firstly, the model (1.2) is discretized spatially by taking sample points of the plate with regular distance. The temperature at the sample points is taken as the discretized state of the plate. We refer to the sample points as nodes.

The spatial discretization results a set of ordinary differential equations (ODEs) in which each ODE corresponds to a node. The nodes are indexed as shown in Fig. 1.1. Note that in the spatial discretization it is important to take into account the numbering or order of the nodes. In our case the node (i, j) is assigned a number l according to the following relation

$$l = (i-1) \cdot N_i + j \ , \tag{1.3}$$

where N_i is the number of nodes in each column i. For example, in Fig. 1.1 node (1,3) has node number 3 and node (3,1) has node number 7.

The states of the plate are defined as

$$\begin{aligned} x_1(t) &= T_{(1,1)}(t) & x_4(t) &= T_{(2,1)}(t) & x_7(t) &= T_{(3,1)}(t) \\ x_2(t) &= T_{(1,2)}(t) & x_5(t) &= T_{(2,2)}(t) & x_8(t) &= T_{(3,2)}(t) \\ x_3(t) &= T_{(1,3)}(t) & x_6(t) &= T_{(2,3)}(t) & x_9(t) &= T_{(3,3)}(t) \ . \end{aligned}$$

Using these states definition, the system equation can be written as

$$\rho C_\mathrm{p} \frac{dx_1}{dt} = \left(-\frac{\kappa}{\Delta x} - \frac{\kappa}{\Delta y} - \frac{h}{K} \right) x_1 + \frac{\kappa}{\Delta y} x_2 + \frac{\kappa}{\Delta x} x_4 + \left(\frac{\dot{Q}_\mathrm{s}(1)}{K} + \frac{h}{K} T_\mathrm{env} \right) \tag{1.4}$$

$$\rho C_\mathrm{p} \frac{dx_2}{dt} = \frac{\kappa}{(\Delta y)^2} x_1 + \left(-\frac{\kappa}{\Delta x} - \frac{2\kappa}{(\Delta y)^2} + \frac{h}{K} \right) x_2 + \frac{\kappa}{(\Delta y)^2} x_3 + \frac{\kappa}{\Delta x} x_5 + \left(\frac{\dot{Q}_\mathrm{s}(2)}{K} + \frac{h}{K} T_\mathrm{env} \right) \tag{1.5}$$

$$\rho C_\mathrm{p} \frac{dx_3}{dt} = -\frac{\kappa}{\Delta y} x_2 + \left(-\frac{\kappa}{\Delta x} + \frac{\kappa}{\Delta y} - \frac{h}{K} \right) x_3 + \frac{\kappa}{\Delta x} x_6 + \left(\frac{\dot{Q}_\mathrm{s}(3)}{K} + \frac{h}{K} T_\mathrm{env} \right) \tag{1.6}$$

$$\rho C_\mathrm{p} \frac{dx_4}{dt} = \frac{\kappa}{(\Delta x)^2} x_1 + \left(-\frac{2\kappa}{(\Delta x)^2} - \frac{\kappa}{\Delta y} - \frac{h}{K} \right) x_4 + \frac{\kappa}{\Delta y} x_5 + \frac{\kappa}{\Delta y} x_7 + \left(\frac{\dot{Q}_\mathrm{s}(4)}{K} + \frac{h}{K} T_\mathrm{env} \right) \tag{1.7}$$

$$\rho C_{\mathrm{p}}\frac{dx_5}{dt} = \frac{\kappa}{(\Delta x)^2}x_2 + \frac{\kappa}{(\Delta y)^2}x_4 + \left(-\frac{2\kappa}{(\Delta x)^2} - \frac{2\kappa}{(\Delta y)^2} - \frac{h}{K}\right)x_5 + \frac{\kappa}{(\Delta y)^2}x_6 + \frac{\kappa}{(\Delta x)^2}x_8 + \left(\frac{\dot{Q}_{\mathrm{s}}(5)}{K} + \frac{h}{K}T_{\mathrm{env}}\right) \tag{1.8}$$

$$\rho C_{\mathrm{p}}\frac{dx_6}{dt} = \frac{\kappa}{(\Delta x)^2}x_3 - \frac{\kappa}{\Delta y}x_5 + \left(-\frac{2\kappa}{(\Delta x)^2} + \frac{\kappa}{\Delta y} - \frac{h}{K}\right)x_6 + \frac{\kappa}{(\Delta x)^2}x_9 + \left(\frac{\dot{Q}_{\mathrm{s}}(6)}{K} + \frac{h}{K}T_{\mathrm{env}}\right) \tag{1.9}$$

$$\rho C_{\mathrm{p}}\frac{dx_7}{dt} = -\frac{\kappa}{\Delta x}x_4 + \left(\frac{\kappa}{\Delta x} - \frac{\kappa}{\Delta y} - \frac{h}{K}\right)x_7 + \frac{\kappa}{\Delta y}x_8 + \left(\frac{\dot{Q}_{\mathrm{s}}(7)}{K} + \frac{h}{K}T_{\mathrm{env}}\right) \tag{1.10}$$

$$\rho C_{\mathrm{p}}\frac{dx_8}{dt} = -\frac{\kappa}{\Delta x}x_5 + \frac{\kappa}{(\Delta y)^2}x_7 + \left(\frac{\kappa}{\Delta x} - \frac{2\kappa}{(\Delta y)^2} - \frac{h}{K}\right)x_8 + \frac{\kappa}{(\Delta y)^2}x_9 + \left(\frac{\dot{Q}_{\mathrm{s}}(8)}{K} + \frac{h}{K}T_{\mathrm{env}}\right) \tag{1.11}$$

$$\rho C_{\mathrm{p}}\frac{dx_9}{dt} = -\frac{\kappa}{\Delta x}x_6 - \frac{\kappa}{\Delta y}x_8 + \left(\frac{\kappa}{\Delta x} + \frac{2\kappa}{\Delta y} - \frac{h}{K}\right)x_9 + \left(\frac{\dot{Q}_{\mathrm{s}}(9)}{K} + \frac{h}{K}T_{\mathrm{env}}\right)\ , \tag{1.12}$$

where $\dot{Q}_{\mathrm{s}}(i)$ is the heat input at node i. In matrix form, the continuous time state equations above can be written as

$$\dot{x} = Ax + Bu$$

where

$$A = \frac{1}{\rho C_{\mathrm{p}}}\begin{pmatrix}
-\frac{\kappa}{\Delta x} - \frac{\kappa}{\Delta y} - \frac{h}{K} & \frac{\kappa}{\Delta y} & 0 & \frac{\kappa}{\Delta x} & 0 & 0 & 0 & 0 & 0 \\
\frac{\kappa}{(\Delta y)^2} & -\frac{\kappa}{\Delta x} - \frac{2\kappa}{(\Delta y)^2} + \frac{h}{K} & \frac{\kappa}{(\Delta y)^2} & 0 & \frac{\kappa}{\Delta x} & 0 & 0 & 0 & 0 \\
0 & -\frac{\kappa}{\Delta y} & -\frac{\kappa}{\Delta x} + \frac{\kappa}{\Delta y} - \frac{h}{K} & 0 & 0 & \frac{\kappa}{\Delta x} & 0 & 0 & 0 \\
\frac{\kappa}{(\Delta x)^2} & 0 & 0 & -\frac{2\kappa}{(\Delta x)^2} - \frac{\kappa}{\Delta y} - \frac{h}{K} & \frac{\kappa}{\Delta y} & 0 & \frac{\kappa}{(\Delta x)^2} & 0 & 0 \\
0 & \frac{\kappa}{(\Delta x)^2} & 0 & \frac{\kappa}{(\Delta y)^2} & -\frac{2\kappa}{(\Delta x)^2} - \frac{2\kappa}{(\Delta y)^2} - \frac{h}{K} & \frac{\kappa}{(\Delta y)^2} & 0 & \frac{\kappa}{(\Delta x)^2} & 0 \\
0 & 0 & \frac{\kappa}{(\Delta x)^2} & 0 & -\frac{\kappa}{\Delta y} & -\frac{2\kappa}{(\Delta x)^2} + \frac{\kappa}{\Delta y} - \frac{h}{K} & 0 & 0 & \frac{\kappa}{(\Delta x)^2} \\
0 & 0 & 0 & -\frac{\kappa}{\Delta x} & 0 & 0 & -\frac{\kappa}{\Delta x} & \frac{\kappa}{\Delta y} & 0 \\
0 & 0 & 0 & 0 & -\frac{\kappa}{\Delta x} & 0 & \frac{\kappa}{\Delta y} & \frac{\kappa}{\Delta x} - \frac{\kappa}{\Delta y} - \frac{h}{K} & \frac{\kappa}{(\Delta y)^2} \\
0 & 0 & 0 & 0 & 0 & -\frac{\kappa}{(\Delta x)^2} & 0 & \frac{\kappa}{(\Delta y)^2} & \frac{\kappa}{\Delta x} + \frac{2\kappa}{\Delta y} - \frac{h}{K}
\end{pmatrix},$$

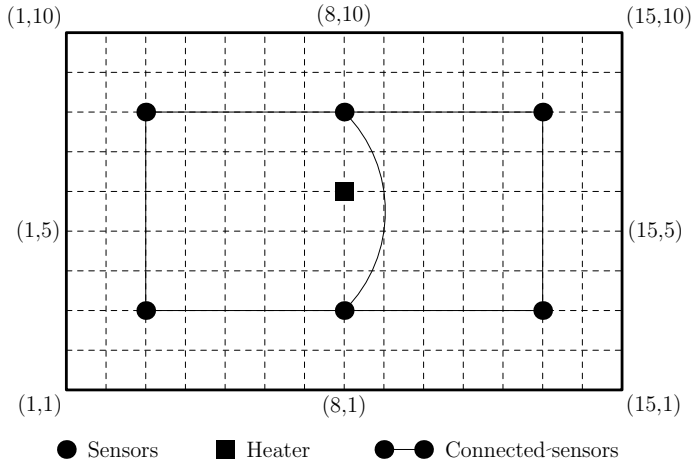


Figure 1.2: Plate set-up for the simulations.

$$B = \frac{1}{\rho C_{\mathrm{p}}} \begin{pmatrix} \frac{1}{K} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{h}{K} \\ 0 & \frac{1}{K} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{h}{K} \\ 0 & 0 & \frac{1}{K} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{h}{K} \\ 0 & 0 & 0 & \frac{1}{K} & 0 & 0 & 0 & 0 & 0 & \frac{h}{K} \\ 0 & 0 & 0 & 0 & \frac{1}{K} & 0 & 0 & 0 & 0 & \frac{h}{K} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{K} & 0 & 0 & 0 & \frac{h}{K} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{K} & 0 & 0 & \frac{h}{K} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{K} & 0 & \frac{h}{K} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{K} & \frac{h}{K} \end{pmatrix} \quad \text{and} \quad u = \begin{pmatrix} \dot{Q}_{\mathrm{s}}(1) \\ \vdots \\ \dot{Q}_{\mathrm{s}}(9) \\ T_{\mathrm{env}} \end{pmatrix}.$$

The input matrix B above assumes a heat source at each node i. To remove the heater at node i we can just set the element of B which corresponds to $\dot{Q}_{\mathrm{s}}(i)$ to zero. The continuous-time state model above can be discretized using available discretization approaches like zero-order hold.

The heated plate used for the simulations is a 1 m by 1.5 m plate that is discretized spatially into a 10-by-15 grid in the same way as in the example above. The model of this heated plate is then discretized in time using a zero-order hold approach with a sampling period of 0.2 s and simulated for 20 min. The plate is heated in node (8,6). Six sensors are placed in node (3,3), (3,8), (3,13), (8,3), (8,8), and (8,13) to measure the temperature at those nodes. The other parameters of the simulated states are listed in Table 1.1. An illustration of the plate set-up is shown in Fig. 1.2. In the figure, the connection between two sensors means they can share information to each other. The connection topology of the network is represented by the Laplacian matrix $\mathscr{L}$. The Laplacian $\mathscr{L}$ for the simulation set-up as shown in Fig. 1.2 is

$$\mathscr{L} = \begin{pmatrix} 1 & -0.5 & -0.5 & 0 & 0 & 0 \\ -0.5 & 1 & 0 & -0.5 & 0 & 0 \\ -0.5 & 0 & 1 & -0.5 & -0.5 & 0 \\ 0 & -0.5 & -0.5 & 1 & 0 & -0.5 \\ 0 & 0 & -0.5 & 0 & 1 & -0.5 \\ 0 & 0 & 0 & -0.5 & -0.5 & 1 \end{pmatrix}. \tag{1.13}$$

To simulate the decentralized Kalman filters in this setup, it is necessary to mention the system model that is used in each sensor node. In the network, each sensor node has its own system model.

Table 1.1: The parameters of the simulated plate used for the simulations.

Parameters	Notations	Values	Units
Density of the plate	ρ	2700	[kg/m^3]
Thermal conductivity	κ	300	[W/m·K]
Heat capacity per area unit	C_p	150	[J/K·kg]
Power of the heaters	$\dot{Q}_\mathrm{s}$	7200	[W/m^2]
Convection coefficient	h	10	[W/m^2·K]
Environment temperature	T_env	298	[K]
Constant	K	0.0067	[m]

Table 1.2: Connectivity of the nodes in the various estimation methods.

Methods	Connectivity
PIF	All nodes are connected to central processor
DHKF	All nodes are neighbors to the others
DIF	All nodes are neighbors to the others
DKFCF	Using Laplacian matrix
DKFBFG	Unestimated states are accessible
DKFWA	Using Laplacian matrix

The system model in a node can be the same or different from that of the others and it depends on the estimation methods. However, the measurement matrix in each node is different from one to the other, namely each node measures only the state on which it is located on the plate.

Not all of the decentralized Kalman filters use the topology of the network in their computations. The methods that explicitly use the Laplacian matrix are the DIF, DKFCF, and DKFWA. For the PIF, the most important assumption is that all nodes are connected to the central processor where the estimates of the nodes are added up stochastically. The DHKF also does not use the Laplacian matrix. Since the estimates are computed separately in each node, then it is assumed the nodes are connected to communicate the estimates. In other words, all nodes are neighbors to the others. The DIF is originally derived with the assumption that all nodes are neighbors to the others. This means there are no zero elements in the Laplacian matrix. In the simulation we do not use the original assumption and use the connectivity based on (1.13) instead. As a result we expect that the error of the DIF will be higher. The DKFBFG also does not use the Laplacian matrix in the computations, but it assumes that the unestimated states are accessible from the other nodes. See Table 1.2 for a summary.

1.4 Simulation results

The temperature profile of the plate at the end of simulation time is shown in Fig. 1.3 and the estimation errors, also at the end of simulation time, of all the compared methods are shown in Figs. 1.5 and 1.6. The estimation errors of the states are plotted for each row for clarity reason. In the figures, the sensor locations are marked with the letter 's' in the x-axis. We also present the estimation errors of the DKFBFG separately since they are much larger than that of the other methods. Adding the errors of the DKFBFG to the same plot as the others would make the errors of the other methods less visible. Fig. 1.4 shows the estimation error of the DIF at node 2 as a function of the discrete time step k.

From the estimation errors plots, we can see that the estimation errors of all methods are com-

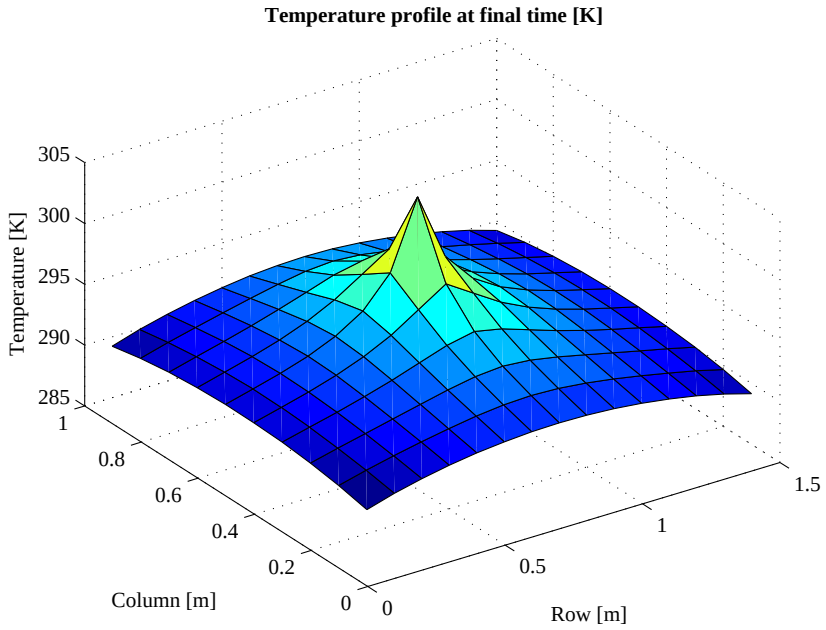


Figure 1.3: Temperature profile at the end of the simulation.

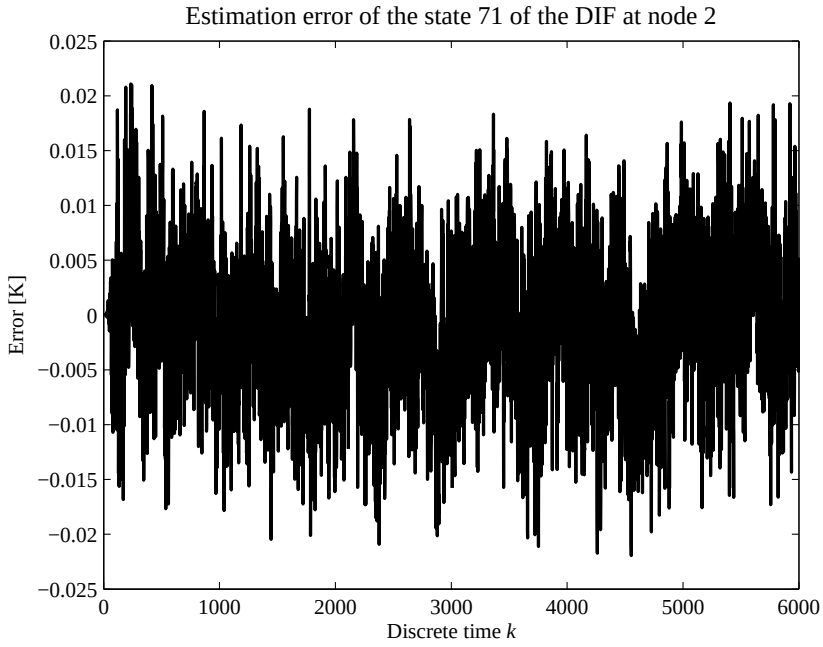


Figure 1.4: Estimation error of state 71 at node 2 for the DIF as a function of the time step k.

parable and also have a similar tendency. In general, the DKFCF and the DHKF give smaller errors compared to the others. But the DHKF yields more variations than the DKFCF does due to the consensus process, which tries to equalize the estimates of the nodes.

Interestingly for all methods except the DKFBFG, it seems that the errors are larger at the measurement point. We can see this in Fig. 1.5c and 1.6b at state index 3, 8, and 15. Those locations are the locations of the sensor nodes. For the DKFBFG however, the errors are larger for larger indices. But it can be seen that the errors are smaller at the sensor nodes, see Figs. 1.7c and 1.8b. It is still an open question why this is happening. One possible direction of explanation could maybe be found in the way the information from the neighbors is collected and added to obtain the overall state.

1.5 Summary

In this chapter we have compared several decentralized Kalman filter approaches using the heated plate as benchmark. In general, the DKFCF and the DHKF gives the smallest errors. Of these two, the DHKF yields more variation than the DKFCF.

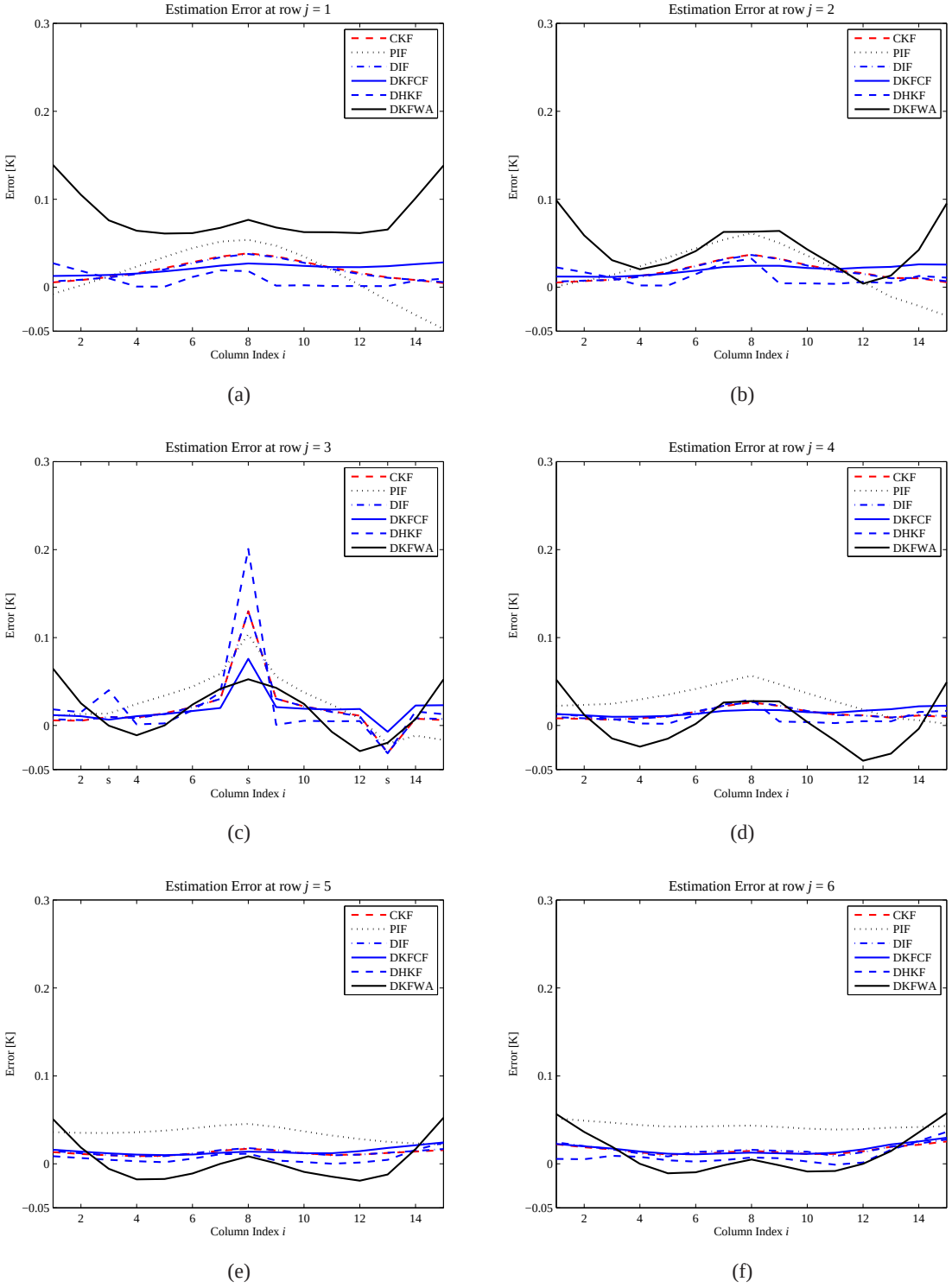


Figure 1.5: The estimation errors in row $j = 1$ to $j = 6$ for all methods except the DKFBFG at time step $k = 6000$. In subfigure (c) the sensor positions are indicated by the letter s on the x-axis.

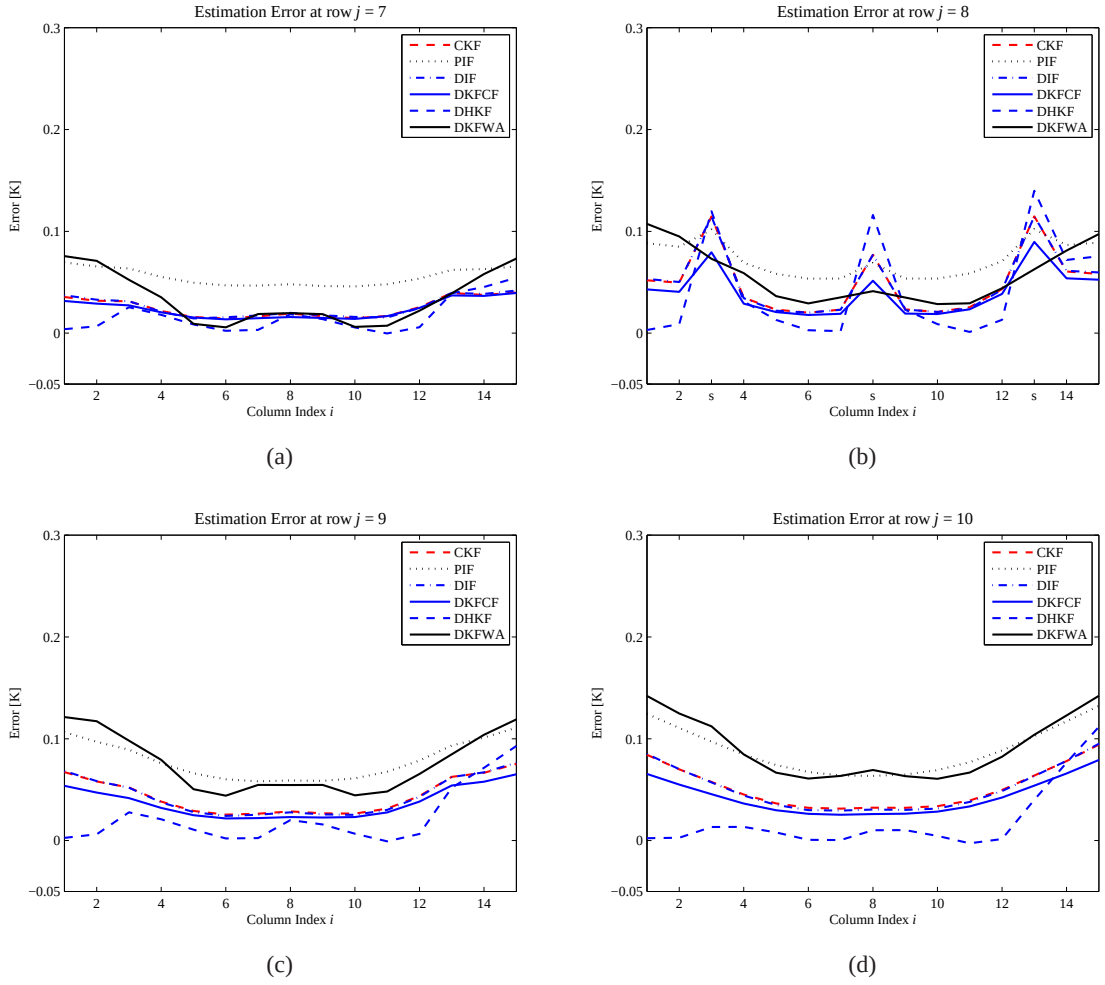


Figure 1.6: The estimation errors in row $j = 7$ to $j = 10$ for all methods except the DKFBFG at time step $k = 6000$. In subfigure (b) the sensor positions are indicated by the letter s on the x-axis.

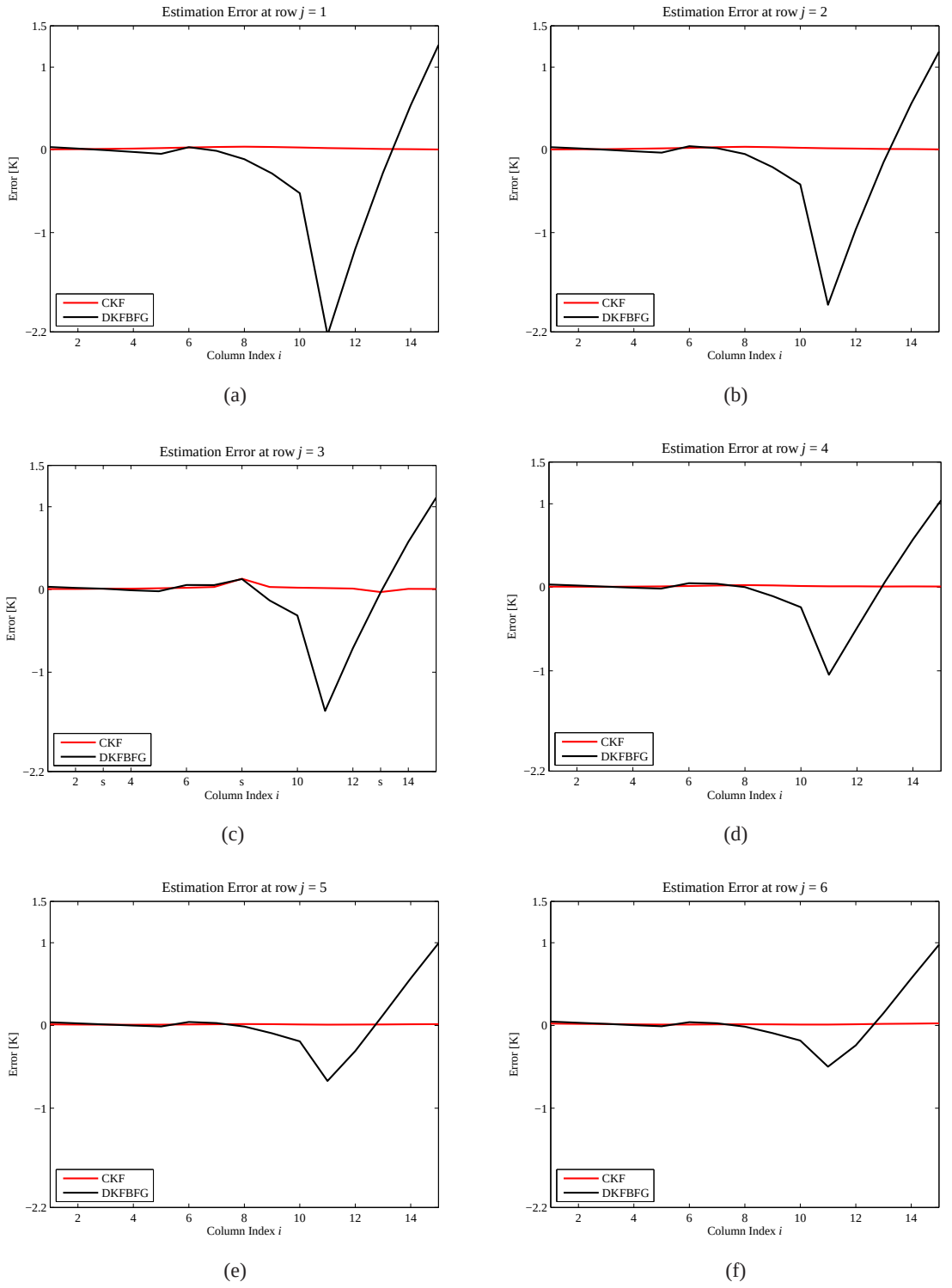


Figure 1.7: The estimation errors in row $j = 1$ to $j = 6$ for the DKFBFG at time step $k = 6000$. In subfigure (c) the sensor positions are indicated by the letter s on the x-axis.

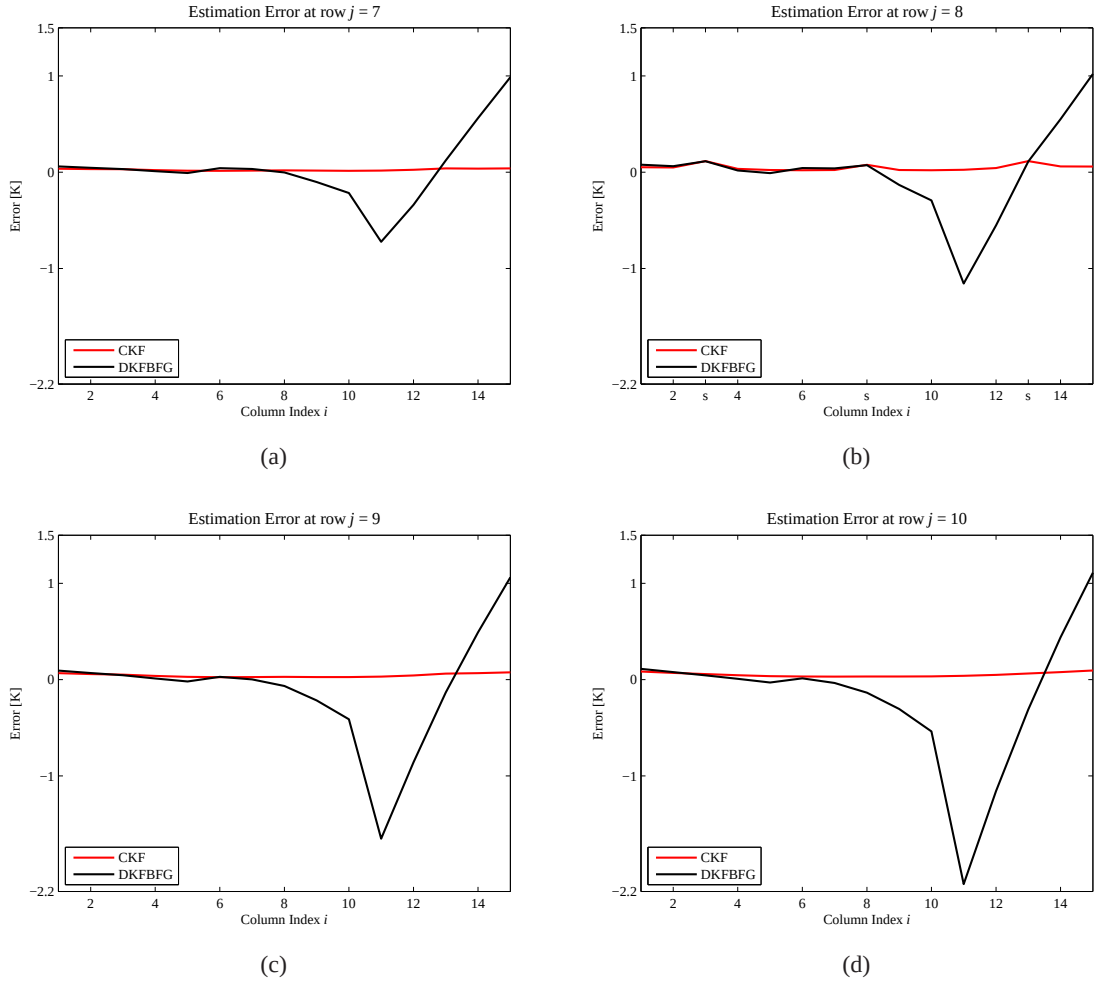


Figure 1.8: The estimation errors in row $j = 7$ to $j = 10$ for the DKFBFG at time step $k = 6000$. In subfigure (b) the sensor positions are indicated by the letter s on the x-axis.

Chapter 2

A distributed state estimation scheme applied to the heat conduction and convection benchmark

In this Section, a distributed state estimation scheme (DDKF-Distributed and Decentralized Kalman Filter) based on the contribution of [29] is applied to one of the proposed benchmarks [49]. An extension to spatially-distributed systems is proposed. This DDKF is originally conceived as a counterpart of a DKF (Decentralized Kalman Filter) which is a set of fully decentralized filters such that they estimate the whole dynamics at each node.

If the DDKF scheme is seen as it was proposed in [48], the first step consist in choosing a set of computational nodes such that each resulting local filter achieves a local prediction and at the same time the communication among nodes does not become a bottleneck for the estimation scheme. Once the computational nodes are defined, the following step deals with finding the internodal transformations such that each local model can be derived from the centralized one.

Hence, the DDKF online computations are performed as the classic Kalman filter: prediction step, and estimation step. However, these computations must be made taking into account the local estimation, internodal communication, and assimilation in order to produce a right estimation at each node.

2.1 Prediction step

Consider a linear state-space model for the whole large-scale system as:

$$\begin{aligned} x(k) &= F(k)x(k-1) + B(k)u(k-1) + w(k-1) \\ z(k) &= H(k)x(k) + v(k) \end{aligned} \tag{2.1}$$

where $x \in \Re^n$ is the system state, $F \in \Re^{n\times n}$ is the transition matrix, $B \in \Re^{n\times m}$ is the input matrix, $u \in \Re^m$ is the input vector and $w \in \Re^n$ is the state uncertainty vector. Moreover, $z \in \Re^p$ is the observation vector, $H \in \Re^{n\times p}$ is the observation matrix and $v \in \Re^p$ is the measurement noise.

The prediction of the state and error covariance is performed at each node i as in the classic Kalman filter using reduced models. Hence, the state and covariance predictions can be computed as follows:

$$\begin{aligned} \hat{x}_i(k|k-1) &= F_i(k)\hat{x}_i(k-1|k-1) + B_i(k)u_i(k-1) \\ P_i(k|k-1) &= F_i(k)P_i(k-1|k-1)F_i^T(k) + Q_i(k) \end{aligned} \tag{2.2}$$

where Q_k is the covariance matrix associated with the state uncertainty.

2.2 Estimation step

Once the prediction step is performed, the interconnected nodes send their state estimations of the neighbor nodes based on its own local measurements. Also the error covariance and state estimation of each node due only to its sensor information must be computed and properly distributed. Finally, when the estimation is exchanged, each node must assimilate the received estimations in order to perform the local estimation.

Consider the error covariance and state estimation of node i due to the information of each neighbor node as:

$$\begin{aligned} P_i(k|z_j(k)) &= T_i(k)[T_j^T(k)P_j^+(k|z_j(k))T_j(k)]^+T_i^T(k) \\ \hat{x}_i(k|z_j(k)) &= V_{ji}(k)\hat{x}_j(k|z_j(k)) \end{aligned} \tag{2.3}$$

where $P_i(k|z_j(k))$ is the error covariance at node i due only to the information concerned to sensor j, T_i, and T_j are transformation matrices, and $P_j^+(k|z_j(k))$ is the Moore-Penrose generalized inverse of the error covariance at node j due only to the availability of the information provided by the sensor j. Note that $(\cdot)^+$ is referred as the Moore-Penrose generalized inverse [29]. Finally $V_{ji}(k)$ is an internodal transformation matrix defined as:

$$V_{ji}(k) = T_i(k)T_j^+(k) \tag{2.4}$$

Then, assuming only orthonormal transformation matrices, the error covariances and state estimation of each node due to its local information can be defined as:

$$\begin{aligned} P_j(k|z_j(k)) &= \left[H_j^T R_j^+ H_j\right]^+ \\ \hat{x}_j(k|z_j(k)) &= P_j(k|z_j(k))\left[H_j^T R_j^+\right] z_j(k) \end{aligned} \tag{2.5}$$

Once the communications are performed, the assimilation procedure takes place as the following equations:

$$\begin{aligned} P_i(k|k) &= \left[P_i^{-1}(k|k-1) + \sum_{j=1}^{N} P_i^+(k|z_j(k))\right]^+ \\ \hat{x}_i(k|k) &= P_i(k|k)\left\{P_i^{-1}(k|k-1)\hat{x}_i(k|k-1) + \sum_{j=1}^{N}\left[P_i^+(k|z_j(k))\hat{x}_i(k|z_j(k))\right]\right\} \end{aligned} \tag{2.6}$$

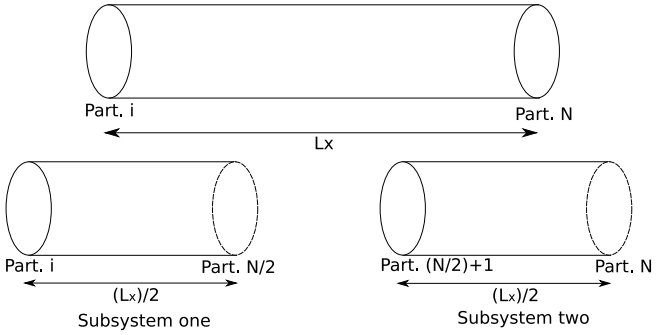


Figure 2.1: A Rod. One dimensional heat exchange.

2.3 A note on the application of the DDKF on spatially distributed systems

The methodology presented in [48], [37], and [28] is mainly focused on large-scale systems which are not distributed in space. The heuristic procedure for the partition of the large-scale system in a given number of subsystems uses a separation of the state variables into "states of interest" and "overlapping" states. States of interest are those present at each node and which are not communicated to other node(s). Instead overlapping states are communicated to other node(s) in order to calculate their own local estimates. Consider a partition of the rod into two rods as it is shown in Fig. 2.1. There is no explicit relationship between the subsystem one and two. Then the conduction and convection phenomena in one side can not be seen in the other side. This deficiency is overcame adding to the prediction model of each node the term associated to the interaction between subsystems:

$$\begin{aligned}\hat{x}_i(k) = F_i(k)\hat{x}_i(k-1) + B_i(k)u_i(k-1) + w_i(k-1) \\ + \sum_{j=1}^{N_i} A_j(k)\hat{x}_j(k-1)\end{aligned} \tag{2.7}$$

where N_i is the number of interacting nodes, and $A_j(k)$ with $j = 1,\ldots,N_i$ the interaction matrices.

2.4 Simulation results

In order to test the modified DDKF formulation for the Heat Conduction and Convection System, the following parameters and assumptions are used in order to perform the simulations: First, it is assumed a solid rod of aluminum whose parameters are presented in [49]. This rod is inside a room with a given environment temperature of $298K$. The length of the rod is $2m$, and 20 partitions are considered. Then a state-space model composed by 20 states is obtained. It is assumed only measurements from partitions 5 and 15. These measurements are corrupted by white noise with unknown standard deviation. On the other hand two heated points are assumed inside the bar at points 5, and 15 as manipulated variables. The continuous time system is discretized with a sampling time of $T_s = 0.1s$. The discrete time matrices do not change their sparse structure and then the distribution process can be applied.

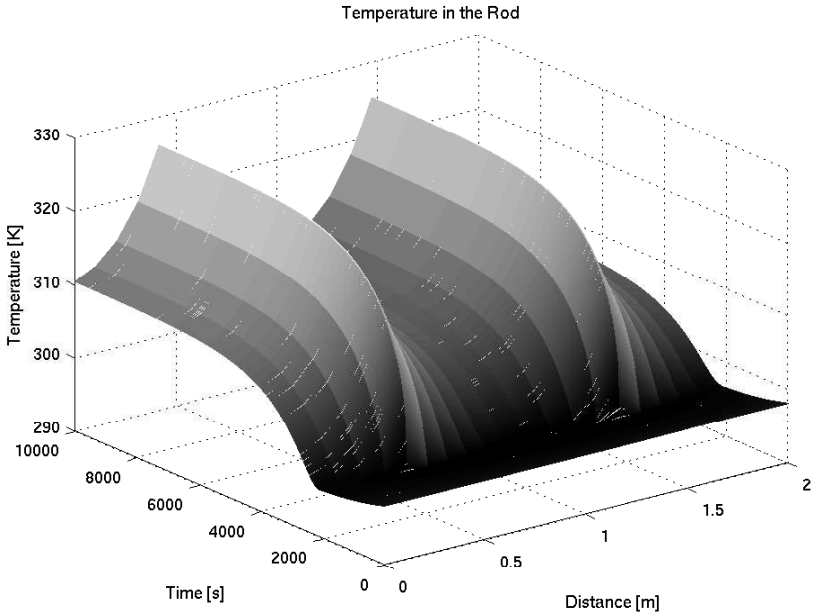


Figure 2.2: Dynamic response of the plant

As there is a spatially distributed system, the methodology presented in [48] can not be applied directly to make a partition of the system. In this case two nodes of 10 partitions are assumed due to the available sensors, that is, there is a node from the beginning to the 10th partition and the other from this partition to the end of the rod. It can be easily demonstrated that the observability of each subsystem is guaranteed even if there is only one temperature sensor at each node. The control actions are distributed in the same way. The original and the extended algorithms are applied to show the deficiency of the first one with respect to spatially-distributed systems.

As statistical parameters of the observers it is considered a model uncertainty covariance of $Q_1 = Q_2 = 0.05 * I_{n/2}$, and a noise covariance of $R_1 = R_2 = 100$, where the subindex 1 and 2 correspond to each node, and $I_{n/2}$ is the identity matrix of order $n/2$, with n the global number of states. The initial conditions of the observer and the controller are set to be a random number around the environment temperature.

Consider the temperature profile of the rod as it is shown in Fig. 2.2. The plant is excited by means of the heaters applying step inputs. The estimated profile of temperature using the badly conditioned observer is shown in Fig. 2.3. The bad estimation is caused at each node by the lack of important information as it was stated in Section IV. The complete estimate, using the modified procedure is fused and it is shown in the Fig. 2.4, where an improved estimation is achieved. The behavior of the distributed observers are shown in Fig. 2.5. Note the good performance of the filters even with measurement noise.

As it can be seen from Figs. 2.2 and 2.4 the modified DDKF tackles the estimation problem in a distributed procedure, filtering the measurement noise and reaching the desired performance. The error in the estimate can be seen in Fig. 2.6. The bad estimation at the beginning of the simulation is

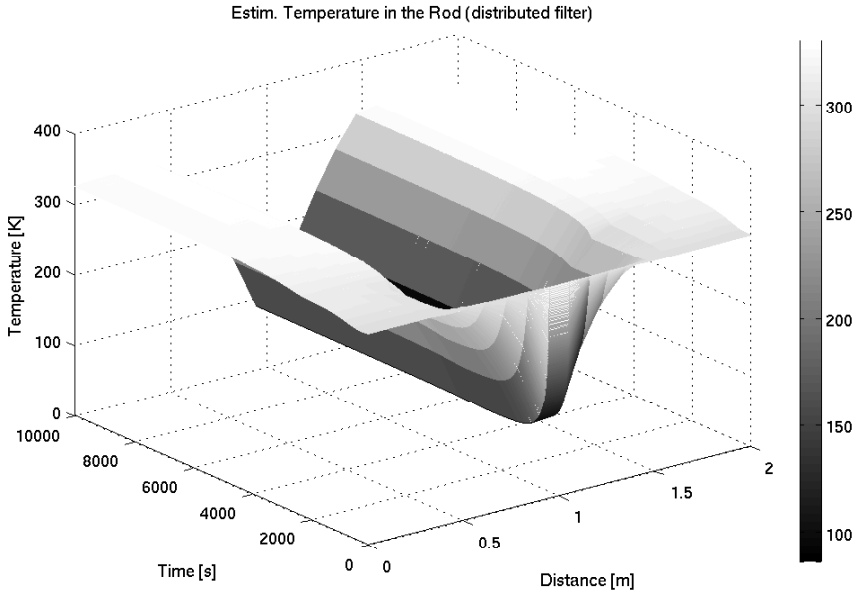


Figure 2.3: Distributed Observer, complete reconstruction of the system state (with the original procedure)

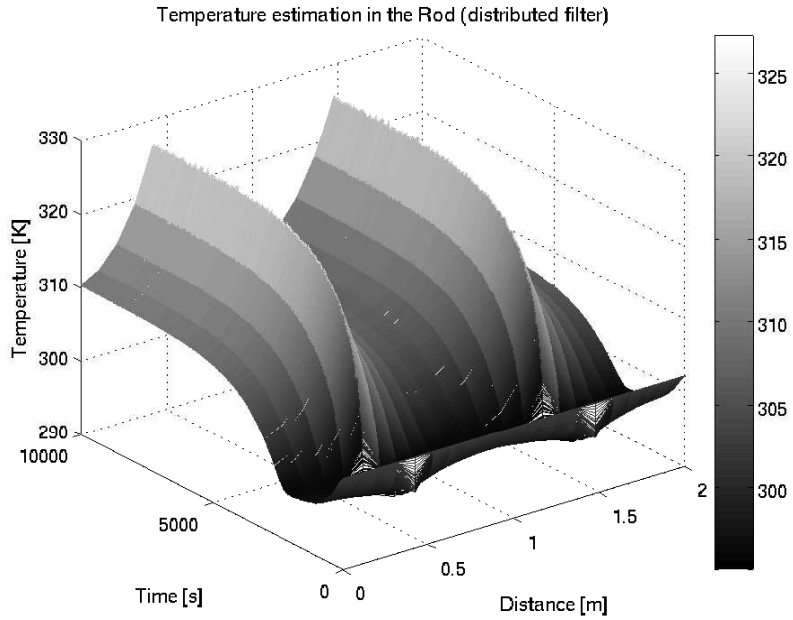


Figure 2.4: Distributed Observer. Complete reconstruction of the system state (procedure with modifications)

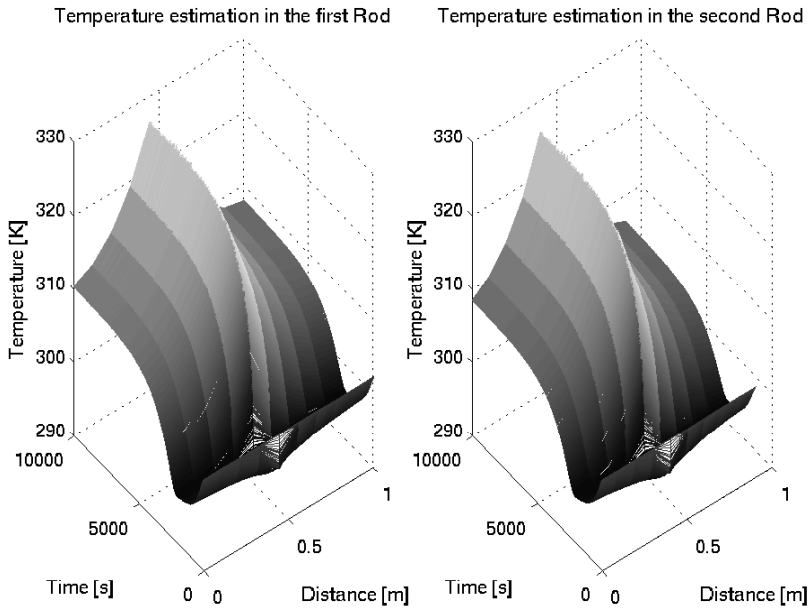


Figure 2.5: Distributed Behavior of the Observer

due to the different initial conditions. Hence, the errors are lower than $1K$ in $320K$, showing the good performance of the designed scheme.

2.5 About disturbance rejection

In this section, the observer performance under disturbances is studied.

2.5.1 Additive disturbances

Consider an additive disturbance to the plant input. The disturbance consists in a variation of the 10% of the input at $5000s$. The plant and the observer responses are shown in Fig. 2.7 and 2.8. In Fig. 2.9 the estimation error is shown. Notice an error fluctuating between $0K$ and $12K$, that is, a maximal error of approximately 4%.

2.5.2 Structural disturbances

Structural or model disturbances deal with inaccuracies on the state model. This kind of disturbances are simulated as an added term to the state equation (in this case is a term which modifies the prediction model in the prediction step). This term is a white noise with an unknown mean and covariance. This kind of disturbance is quite important for two reasons: the modeling inaccuracies are taken into account with this term, and second it can emulate some error in the communication step. As stated before, any inaccuracy in the communication step meaningfully deteriorate the estimation process. A set of simulations are made with a set of white noises added to the observer models. Setting the covariances with a magnitude of 0.1 of the random value the estimation task becomes inaccurate. The

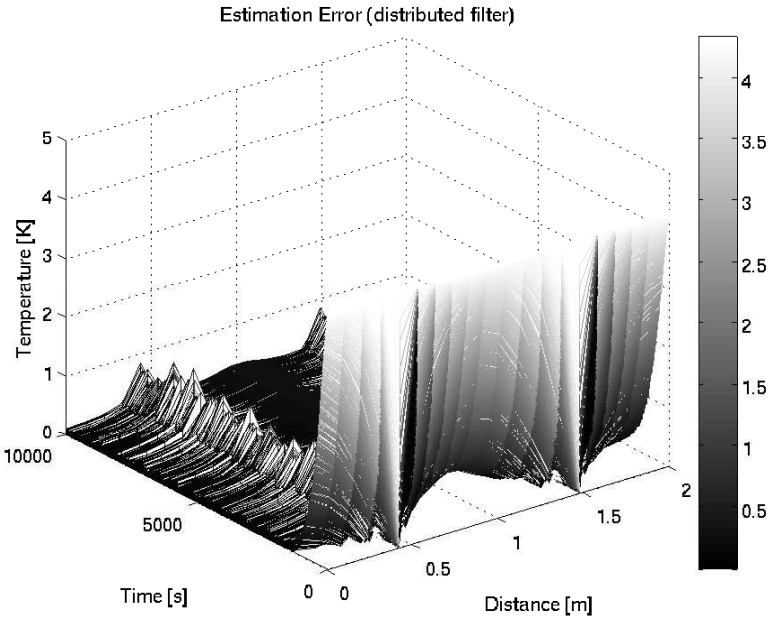


Figure 2.6: Distributed behavior of the Observer

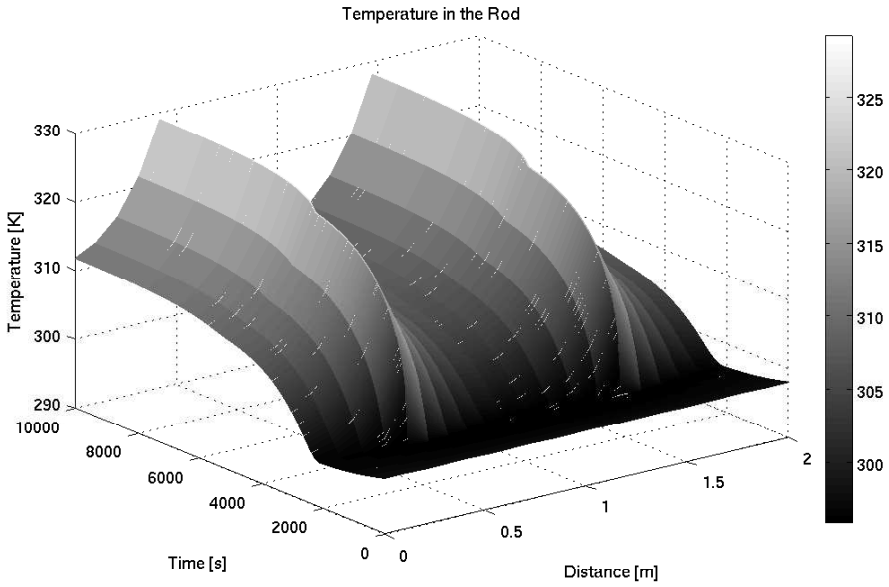


Figure 2.7: Plant response

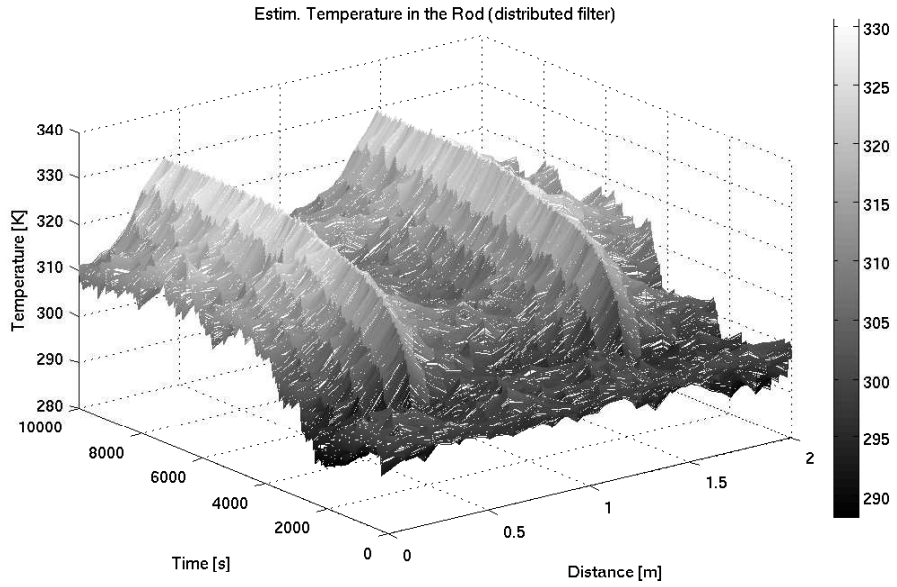


Figure 2.8: Observer response

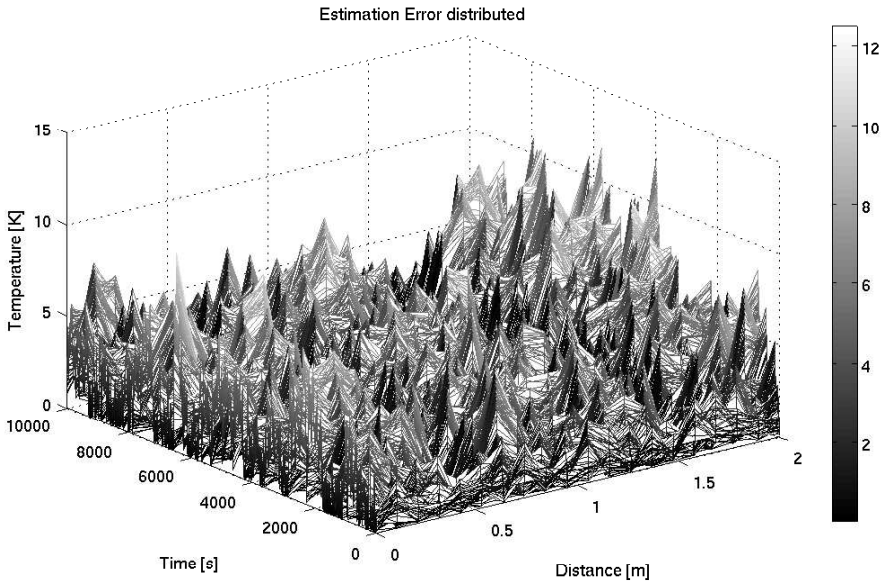


Figure 2.9: Estimation Error

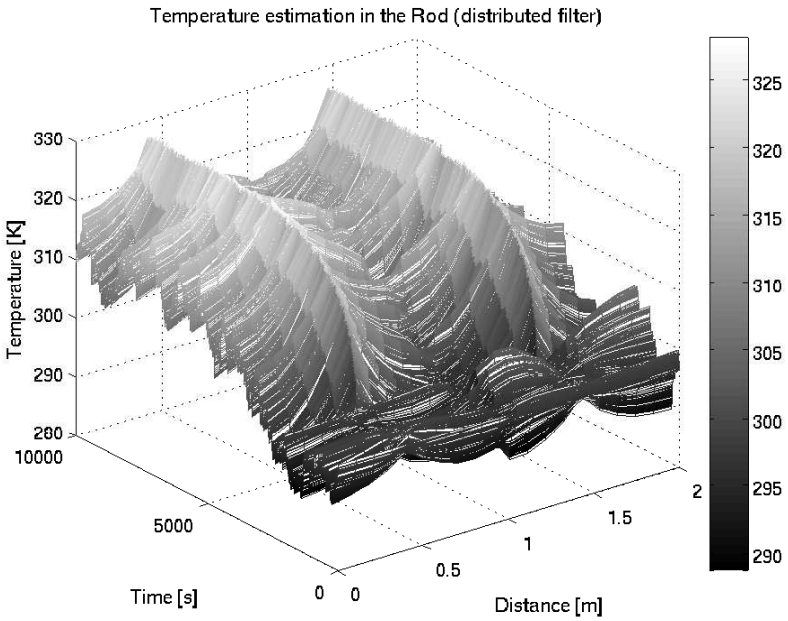


Figure 2.10: Observer response with a model uncertainty.

estimation response and the estimation error with this covariances can be seen in Figs. 2.10 and 2.11. Note that the largest error is about $15K$ that produces an estimation error of 5%.

2.5.3 Noise filtering

In this subsection the noise tolerance of the filtering scheme is tested. The simulation performed to test the distributed filter performance was made considering a Gaussian measurement noise with zero mean and a covariance of one. Simulations were performed with covariances $C_{N1} = C_{N2} = 10$ for each noise respectively, with similar results as presented before. Then, an impractical noise whose covariances are $C_{N1} = C_{N2} = 100$ are applied. Notice, although the noise magnitude is high, the filter was designed to overcome this kind of noises. In Figs. 2.12 and 2.13, the observer response and the observation error are shown. The last Figure shows an error of $4K$ in $305K$, or in other words an error of 1.31% of observation.

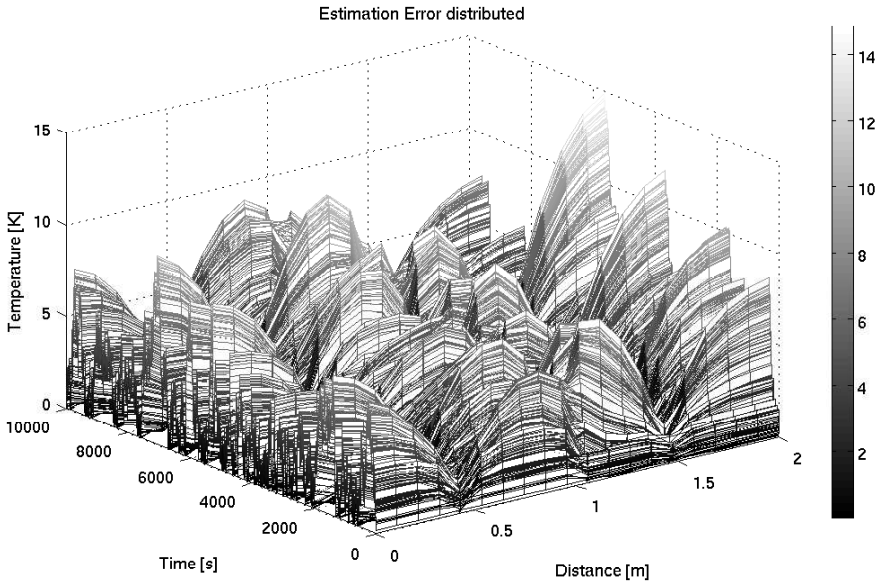


Figure 2.11: Observer Error with a model uncertainty

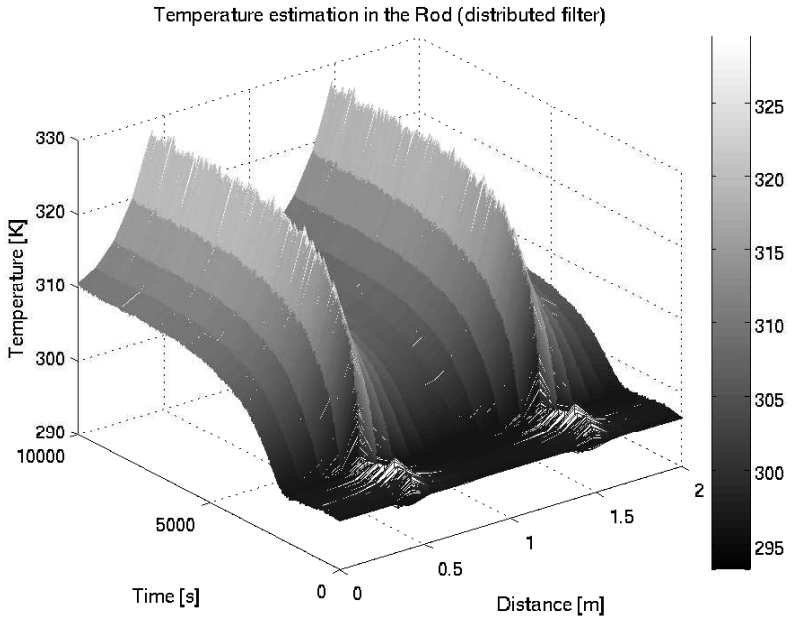


Figure 2.12: Observer response when a Gaussian noise with $C_{N1} = C_{N2} = 10$

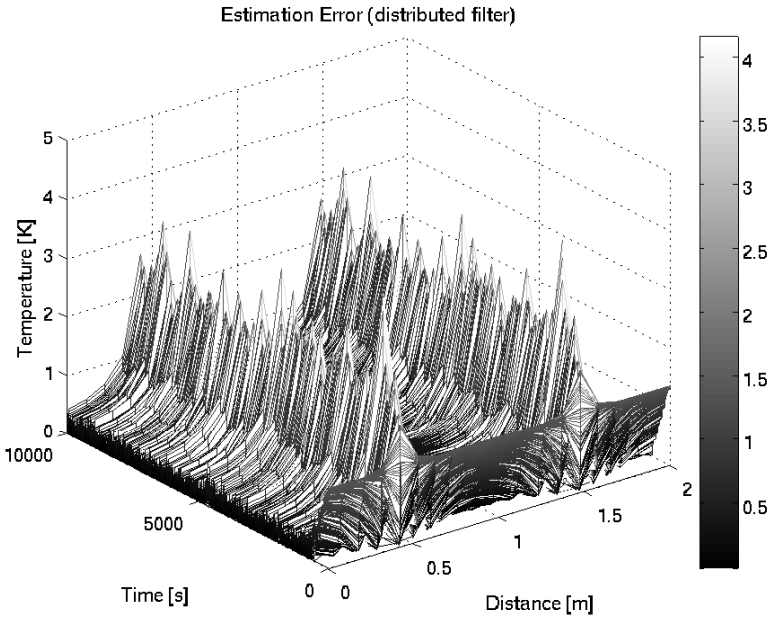


Figure 2.13: Observer Error when a Gaussian noise with $C_{N1} = C_{N2} = 10$

Chapter 3

A distributed model predictive control formulation

Consider a system with n subsystems, where each of them is influenced by other subsystems. Then the linear approximation to each subsystem is given by:

$$\begin{aligned} x_i(k+1) &= \sum_{j=1}^{n} [A_{ij}x_j(k) + B_{ij}u_j(k)] \\ y_i(k) &= C_i x_i(k) + D_i u_i(k) \end{aligned} \tag{3.1}$$

where $x_i \in \mathbb{R}^{n_i}$, $u_i \in \mathbb{R}^{m_i}$, $i = 1,2,\ldots,n$, being k the sample time. In this Equation $x_i(k)$ is the current state, $y_i(k)$ is the output, $u_i(k)$ is the input at node i, A_{ij}, and B_{ij} the interaction matrices. From the subsystems representation (3.1), the model of the whole system becomes

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{3.2}$$

where $A = [A_1, A_2, \ldots, A_n]^T$, $B = [B_1, B_2, \ldots, B_n]^T$, $C = [C_1, C_2, \ldots, C_n]^T$, $D = [D_1, D_2, \ldots, D_n]$, being $A_i = [A_{i1}, A_{i2}, \ldots, A_{in}]$, $B_i = [B_{i1}, B_{i2}, \ldots, B_{in}]$. Based on the model of the whole system (3.2), the formulation of the model predictive control in a centralized fashion can be presented as:

$$\begin{aligned} &\min_{u(k)} J[x(k), u(k)] \\ s.t &: x(k+1) = Ax(k) + Bu(k) \\ &y(k) = Cx(k) + Du(k) \\ &\text{operational constraints} \end{aligned} \tag{3.3}$$

Commonly, it is used as cost function $J(.)$ the following quadratic expression:

$$\begin{aligned} J[x(k,t), u(k,t)] = \sum_{t=0}^{N_p} [y_{ref}(k+t+1) - y(k+t+1)]^T Q [y_{ref}(k+t+1) - y(k+t+1)] \\ + \sum_{t=0}^{N_u} [u(k+t)^T R u(k+t)] \end{aligned} \tag{3.4}$$

where $N_p, N_u \in \Re$ are the prediction and control horizons respectively, $N_p \geq N_u$, $y_{ref} \in \Re^p$ is the output of the system, and $Q, R > 0$.
Taking into account each subsystem model (3.1), the quadratic cost function (3.4) becomes:

$$J[x(k), u(k)] = \sum_{i=1}^{n} [J_i(x_i(k,t), u_i(k,t))] \tag{3.5}$$

with $J_i[x_i(k,t), u_i(k,t)] = \sum_{t=0}^{N_{pi}} [y_{iref}(k+t+1) - y_i(k+t+1)]^T Q_i [y_{iref}(k+t+1) - y_i(k+t+1)] + \sum_{t=0}^{N_{ui}} [u_i(k+t)^T R_i u_i(k+t)]$.

Since the global cost function can be decomposed as the sum of local cost functions, then the centralized model predictive control problem can be solved as a sum of distributed ones. Thus model predictive control can be written as:

$$\begin{aligned} &\min_{u_i(k)} \sum_{i=1}^{n} J_i[x_i(k,t), u_i(k,t)] \\ s.t: &\; x_i(k+t+1) = \sum_{j=1}^{n} [A_{ij} x_j(k+t) + B_{ij} u_j(k+t)] \\ &\; y_i(k+t) = C_i x_i(k+t) + D_i u_i(k+t) \\ &\; \text{operational constraints} \end{aligned} \tag{3.6}$$

3.1 Simulation results

Now, in order to test the combined DDKF and DMPC formulation for the Heat Conduction and Convection System, the following parameters and assumptions are used in order to perform the simulations: the system is controlled in a constant set point of $305K$. Five heated points are assumed inside the bar at points 1, 5, 10, 15, and 20 as manipulated variables. The controller considers the following constraint on the manipulated variables: $(0 \leq u_i \leq 2000W/m^2)$. The continuous time system is discretized with a sampling time of $T_s = 0.1s$. The discrete time matrices do not change their sparse structure and then the distribution process can be applied.

The designed DMPC are composed by two local MPC with $H_p = 20$, and $H_c = 2$, where H_p, and H_c are the prediction and control horizons respectively. As it was pointed out, constraints are imposed over the control actions.

The temperature profile of the rod can be seen in Fig. 3.1. Notice that the temperature is controlled around the set point at those points in which there is a heater. The figure shows that the temperature goes down between two nodes caused by the heat transferred to the environment by means of the convection phenomenon. On the other hand, Fig. 3.2 shows the controller performance assuming all the states are measured without noise. Note that the performance with and without observer becomes indistinguishable, even with measurement noise.

The behavior of each observer is shown in Figs. 3.3 and 3.4, and the complete observer in Fig. 3.5. Note, the performance of the filters even with measurement noise. Once the filters has their estimate, they are used by the DMPC to find the control that minimizes the proposed cost function. The applied control actions are shown in Fig. 3.6. In these figures the control actions of each local controller are presented. Note that the constraints of the control inputs are satisfied.

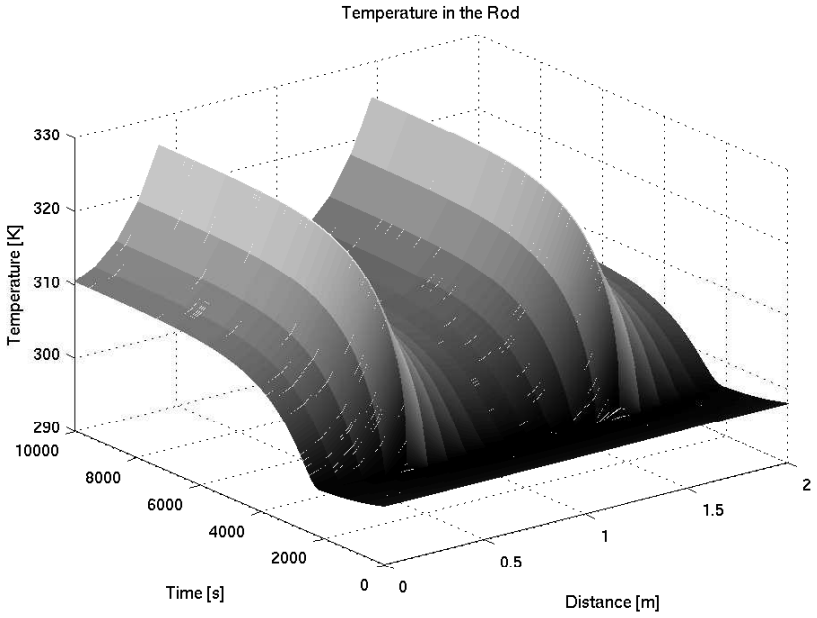


Figure 3.1: Controlled plant with observer. Measures with noise.

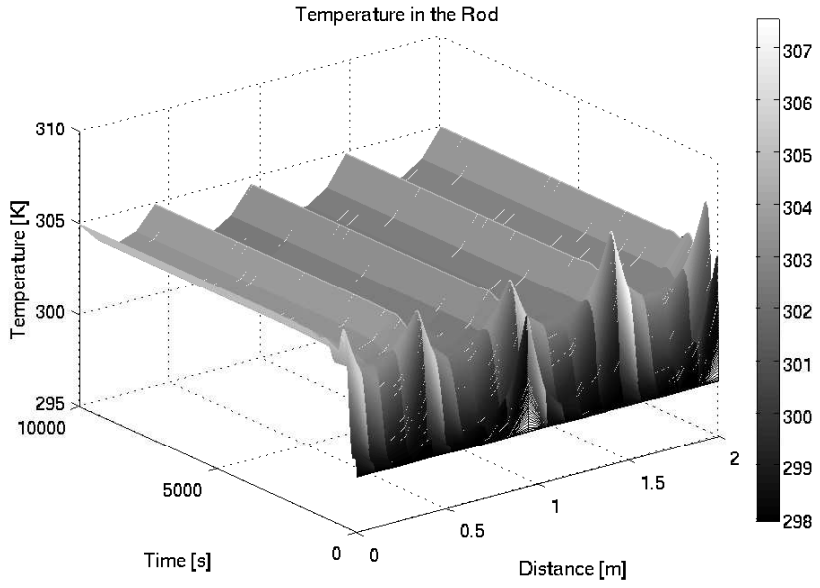


Figure 3.2: Controlled plant assuming that all the states are measured. Measure without noise.

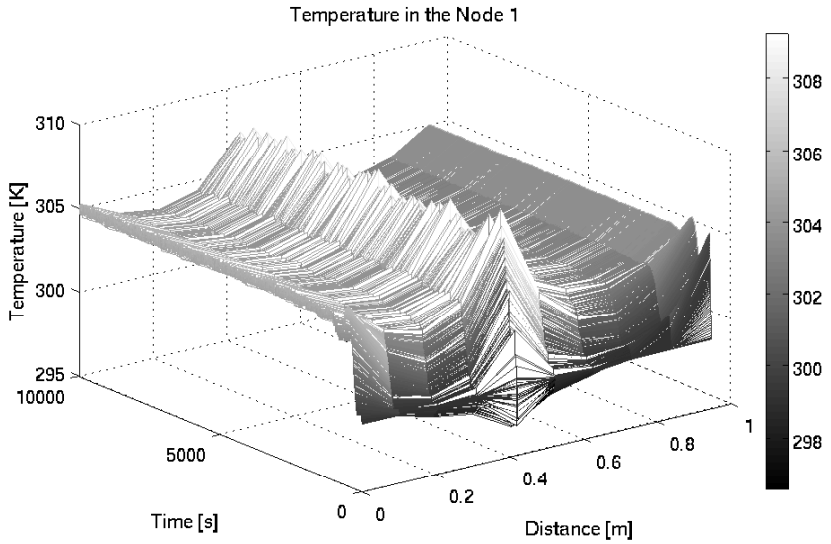


Figure 3.3: Observer behavior of the node one

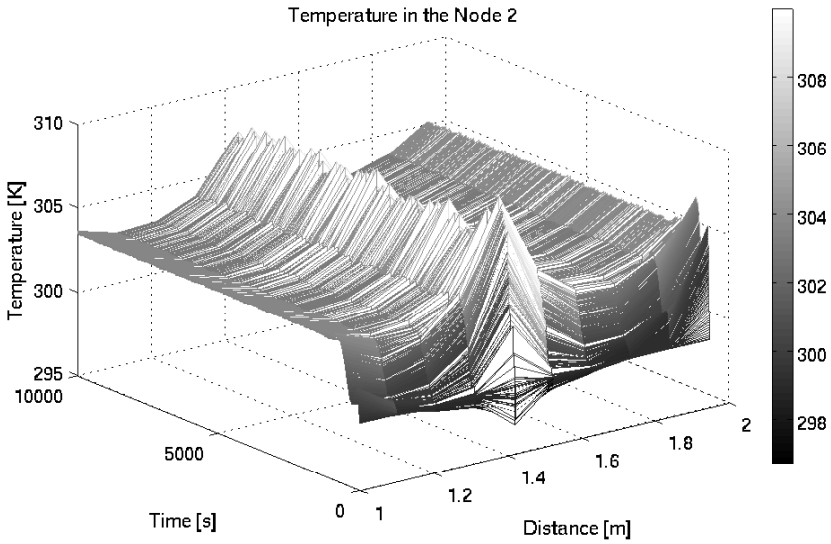


Figure 3.4: Observer behavior of the node two

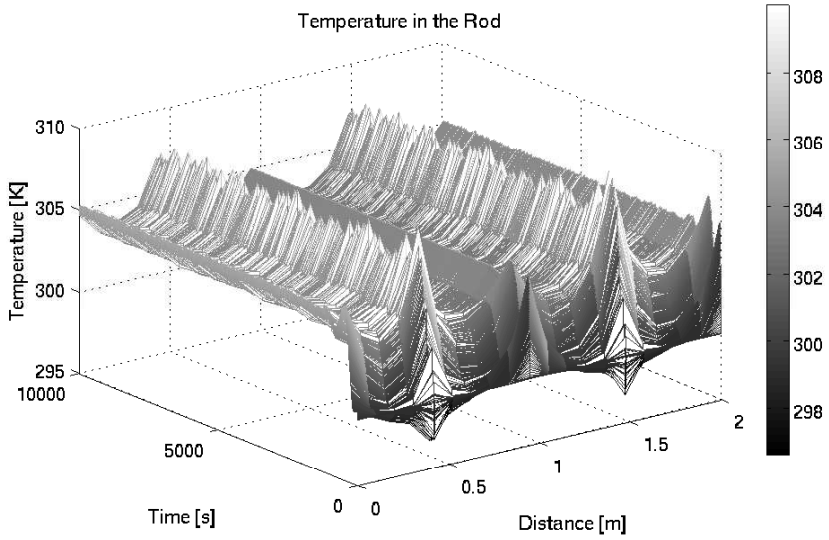


Figure 3.5: Distributed Observer (complete reconstruction of the system state)

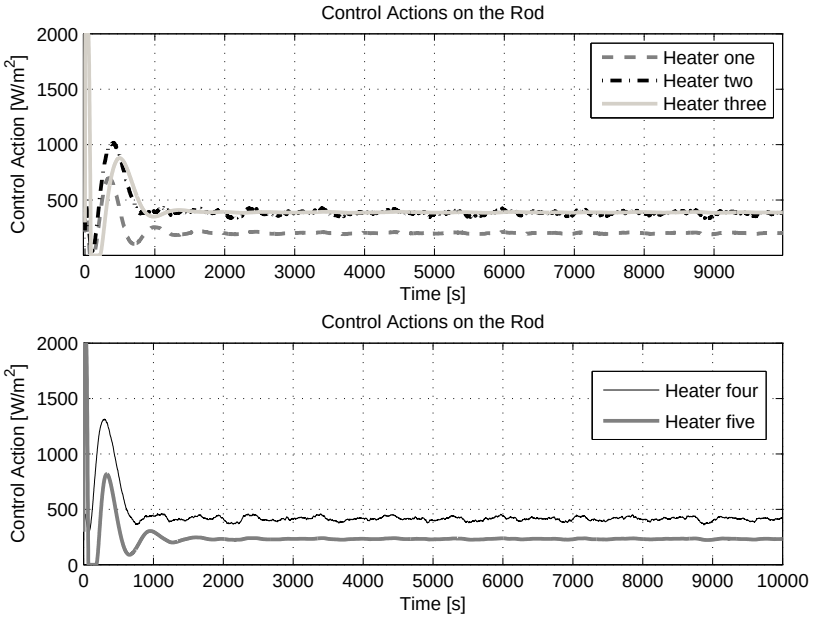


Figure 3.6: Control Actions at each node executed by the DMPC. Each node has one heater as it is presented

3.2 About this implementation

A distributed estimation procedure is applied to a benchmark case due to its intuitive application and its reported results. It is also shown that the scheme is able to estimate the state in large-scale systems with explicitly interaction. When this procedure is applied to a spatially distributed system, there is certain missing information that hinders the local and hence the global performance of the estimation scheme. In this paper a modification to the selected scheme is proposed to overcome the discussed drawback and then it is demonstrated in simulation by means of an example. The results show a good performance of the distributed filters, even if the available measurements are corrupted by noise. A qualitative analysis is made to different kind of disturbances: structural and additive to the input. As future work, there is a need to extrapolate the linear results on distributed observers to the nonlinear framework, using the tools as the unscented transformation or particle filters, among others. Moreover, computational and communication issues must be discussed in those strategies, and the approach to the centralized optimal as it has been published in the linear case.

In this work a coupled estimation and control system is tested in simulation in a spatially distributed system (benchmark system). It can be seen that the global estimation and control problem can be partitioned in a lower number of subsystems without loss of system performance, this is due to the fact that the global cost function can be decomposed in several number of local cost functions. It was also shown that the controller hold its global optimality once the problem is partitioned. Moreover, the modified DDKF was able to tackle the global estimation problem leading to the controller an acceptable performance based on the noisy information available.

Part II

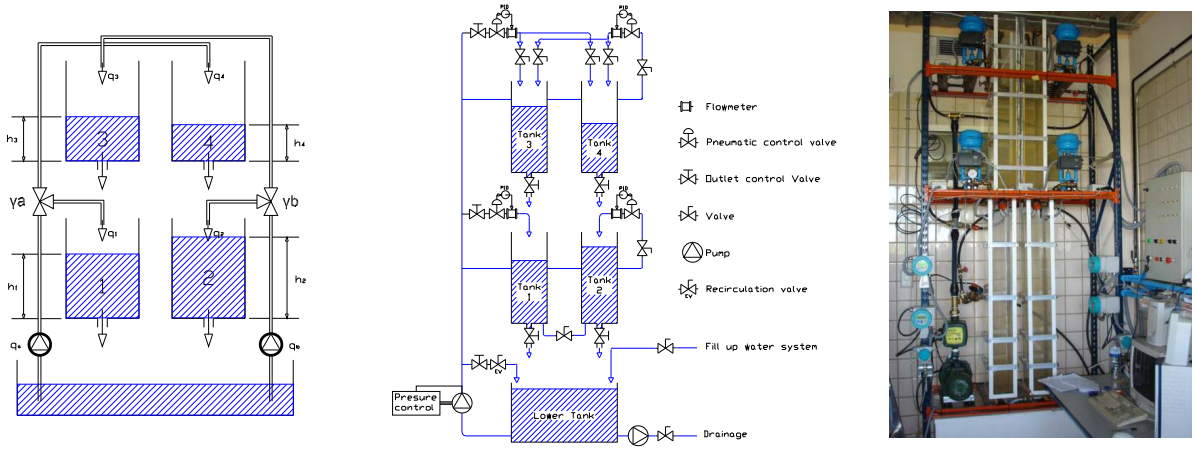
Four-tanks system benchmark

Chapter 4

Control schemas applied to the four-tanks real plant

4.1 Introduction

The four tanks plant is a multivariable laboratory plant of interconnected tanks with nonlinear dynamics and subject to state and input constraints. One important property of this plant is that the dynamics present multivariable transmission zeros which can be located in the right-hand side of the *s* plane for some operating conditions. This plant is based on the well known quadruple-tank process [12], and its scheme can be seen in Fig. 4.1(a). In the original plant, the inputs are the voltages of the two pumps and the outputs are the water levels in the lower two tanks. Fig. 4.1(b) shows the scheme of the real plant. The main difference is that a control valve regulates the inlet flow of each tank. The three-way valve ratio is imposed by a suitable choice of the references of the flows.



(a) (b) (c)

Figure 4.1: (a) Scheme of the quadruple tank process, (b) scheme of the real plant and (c) picture of the real plant.

A state space continuous time model of the quadruple tank process system [12] can be derived from first principles as follows

$$\begin{aligned}
\frac{dh_1}{dt} &= -\frac{a_1}{A_1}\sqrt{2gh_1}+\frac{a_3}{A_1}\sqrt{2gh_3}+\frac{\gamma_a}{A_1}q_a \\
\frac{dh_2}{dt} &= -\frac{a_2}{A_2}\sqrt{2gh_2}+\frac{a_4}{A_2}\sqrt{2gh_4}+\frac{\gamma_b}{A_2}q_b \\
\frac{dh_3}{dt} &= -\frac{a_3}{A_3}\sqrt{2gh_3}+\frac{(1-\gamma_b)}{A_3}q_b \\
\frac{dh_4}{dt} &= -\frac{a_4}{A_4}\sqrt{2gh_4}+\frac{(1-\gamma_a)}{A_4}q_a
\end{aligned} \tag{4.1}$$

The estimated parameters of the real plant and the considered intervals of admissible variation of the levels and flows are shown in the following table:

	Value	Unit	Description
H_{1max}	1.36	m	Maximum level of the tank 1
H_{2max}	1.36	m	Maximum level of the tank 2
H_{3max}	1.30	m	Maximum level of the tank 3
H_{4max}	1.30	m	Maximum level of the tank 4
H_{min}	0.2	m	Minimum level in all cases
Q_{amax}	3.26	m^3/h	Maximal flow of pump 'a'
Q_{bmax}	4.00	m^3/h	Maximal flow of pump 'b'
Q_{min}	0	m^3/h	Minimal flow of both pumps
Q_a^0	1.63	m^3/h	Equilibrium flow
Q_b^0	2.00	m^3/h	Equilibrium flow
a_1	1.310e-4	m^2	Discharge constant of tank 1
a_2	1.507e-4	m^2	Discharge constant of tank 2
a_3	9.267e-5	m^2	Discharge constant of tank 3
a_4	8.816e-5	m^2	Discharge constant of tank 4
A	0.06	m^2	Cross-section of all tanks
γ_a	0.3		Parameter of the 3-ways valve
γ_b	0.4		Parameter of the 3-ways valve
h_1^0	0.6534	m	Equilibrium level of tank 1
h_2^0	0.6521	m	Equilibrium level of tank 2
h_3^0	0.6594	m	Equilibrium level of tank 3
h_4^0	0.6587	m	Equilibrium level of tank 4
T_m	5	s	Sample time

The minimum level of the tanks has been taken greater than zero to prevent eddy effects in the discharge of the tank. The values of γ_a and γ_b have been chosen in order to obtain a system with non-minimum phase multivariable zeros.

Linearizing the model at an operating point given by h_i^0 and defining the deviation variables $x_i = h_i - h_i^o$ and $u_j = q_j - q_j^o$ where $j = a,b$ and $i = 1,\cdots,4$ we have:

$$\frac{dx}{dt} = \begin{bmatrix} \frac{-1}{\tau_1} & 0 & \frac{A_3}{A_1\tau_3} & 0 \\ 0 & \frac{-1}{\tau_2} & 0 & \frac{A_4}{A_2\tau_4} \\ 0 & 0 & \frac{-1}{\tau_3} & 0 \\ 0 & 0 & 0 & \frac{-1}{\tau_4} \end{bmatrix} x + \begin{bmatrix} \frac{\gamma_a}{A_1} & 0 \\ 0 & \frac{\gamma_b}{A_2} \\ 0 & \frac{(1-\gamma_b)}{A_3} \\ \frac{(1-\gamma_a)}{A_4} & 0 \end{bmatrix} u.$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x$$

where $\tau_i = \frac{A_i}{a_i}\sqrt{\frac{2h_i^0}{g}} \geq 0$, $i = 1, \cdots, 4$, are the time constants of each tank.

The main sources of deviation between the nonlinear model and the real plant are (i) the linearization error; (ii) the hypothesis that parameters a_i do not depend on the levels of the tank; (iii) the actuator dynamics since the modeled input to the plant is the reference of the PID that controls the flow of each pipe.

4.1.1 Control configurations

Four different control configurations are going to be used that can be grouped in two classes:

- Tracking Control. Control that allow changes in the reference.
  - Centralized control for tracking. In this case the control decides both flows and reads all the variables of the process. The used controller is a Centralized MPC for tracking [13].

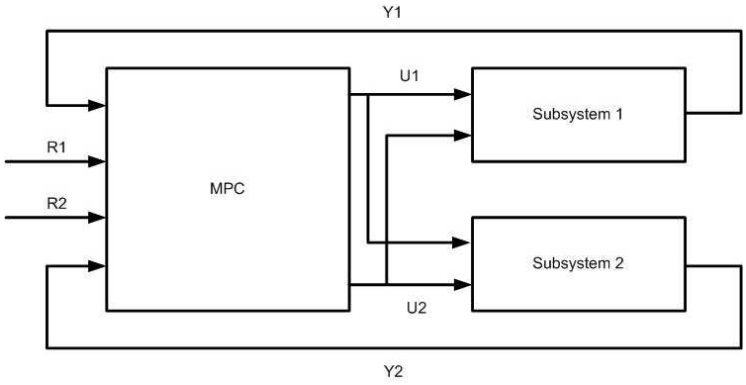


Figure 4.2: Centralized Control.

  - Decentralized control for tracking. Two MPC for tracking are used, the same as in the previous case, but applied to each subsystem. The pairing procedure between the inputs is done based on the Relative Gain Array. Two examples are done, one with the correct pairing, and the second with the wrong one.
- Regulation controller. To perform the reference changes, One controller for each reference is designed.
  - Centralized control.
  - Distributed control. *Distributed MPC based on a cooperative game* [16]

In a future work, a distributed controller for tracking will be developed and, a decentralized MPC controller for regulation will be implemented over the plant to compare the performance with the distributed and centralized schemes.

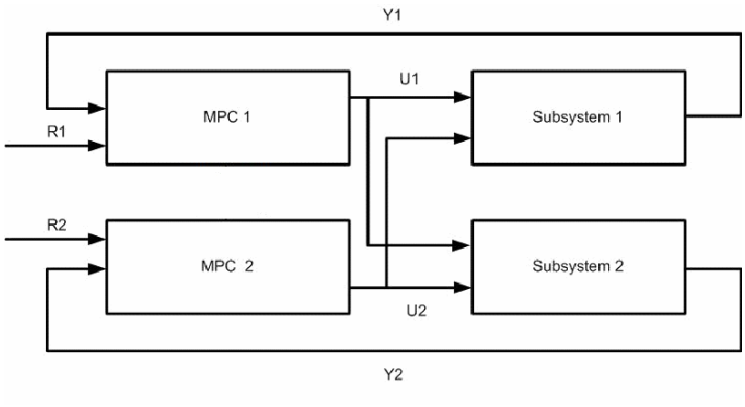


Figure 4.3: Decentralized Control.

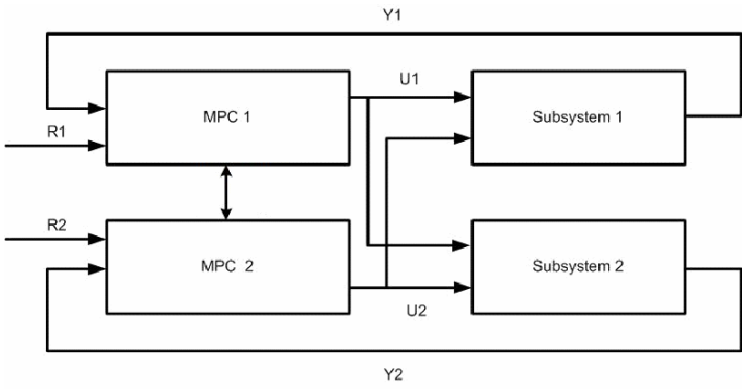


Figure 4.4: Distributed Control.

4.1.2 Benchmark

Four reference changes are performed during the benchmark experiment. These references are chosen from the available set of references provided by the Centralized MPC for tracking. The references are the following:

- $ref_1 = [0.65; 0.65]$. The first reference is provided to setup the plant at the working point.
- $ref_2 = [0.30; 0.30]$. This reference is close to lower limit of the levels of the plant.
- $ref_3 = [0.50; 0.75]$. This reference is provided to perform the following change of setpoint (where only one reference level is changed).
- $ref_4 = [0.90; 0.75]$. To perform this change the tank 3 and 4 have to be emptied and filled.

The performance criterion is the integral of the square error of the outputs SEI.

$$SEI = \int_0^{t_f} ((h_1(\tau) - h_{1r}(\tau))^2 + (h_2(\tau) - h_{2r}(\tau))^2) d\tau$$

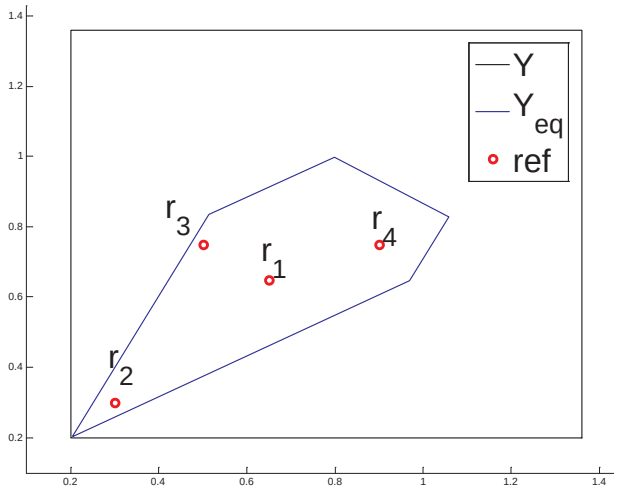


Figure 4.5: References.

4.2 Centralized MPC for tracking

In this section we present the MPC for tracking proposed in [13].

Notation: vector (x,t,r) denotes $[x^T,t^T,r^T]^T$; for a given λ, $\lambda X = \{\lambda x : x \in X\}$; $int(X)$ denotes the interior of set X; a matrix T definite positive is denoted as $T > 0$ and $T > P$ denotes that $T - P > 0$. For a given symmetric matrix $P > 0$, $\|x\|_P$ denotes the weighted Euclidean norm of x, i.e. $\|x\|_P = \sqrt{x^T P x}$. Matrix $\mathbf{0}_{n,m} \in \mathbb{R}^{n\times m}$ denotes a matrix of zeros. Consider $a \in \mathbb{R}^{n_a}$, $b \in \mathbb{R}^{n_b}$, and set $\Gamma \subset \mathbb{R}^{n_a+n_b}$, then projection operation is defined as $Proj_a(\Gamma) = \{a \in \mathbb{R}^{n_a} : \exists b \in \mathbb{R}^{n_b}, (a,b) \in \Gamma\}$.

Let a discrete-time linear system be described by:

$$\begin{aligned} x^+ &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{4.2}$$

where $x \in \mathbb{R}^n$ is the current state of the system, $u \in \mathbb{R}^m$ is the current input, $y \in \mathbb{R}^p$ is the current output and x^+ is the successor state. The state of the system and the control input applied at sampling time k are denoted as $x(k)$ and $u(k)$ respectively. The system is subject to hard constraints on state and control:

$$(x(k),u(k)) \in Z \quad = \quad \{z \in \mathbb{R}^{n+m} : A_z z \leq b_z\}, \forall k \geq 0 \tag{4.3}$$

where the set Z is a compact convex polyhedron containing the origin in its interior.

The problem we consider is the design of an MPC controller to track a piece-wise constant sequence of set points or references $s(k)$ in such a way that the constraints are satisfied at all times.

Characterization of the steady states

Consider the nominal model of the plant (4.2) subject to the constraints on the nominal state and input given by (4.3). Every nominal steady state and input $z_s = (x_s,u_s)$ is a solution of the equation

$$\begin{bmatrix} A - I_n & B \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \mathbf{0}_{n,1} \tag{4.4}$$

and hence it is an element of the null space of the linear transformation given by matrix $[A-I_n\ B]$. Since it is assumed that (A,B) is controllable, the dimension of this null space is equal to m. Therefore, there exists a matrix $M_\theta \in \mathbb{R}^{(n+m)\times m}$ such that every nominal steady state and input can be posed as

$$z_s = M_\theta \theta \tag{4.5}$$

for certain $\theta \in \mathbb{R}^m$. The subspace of nominal steady outputs is then given by

$$y_s = N_\theta \theta \tag{4.6}$$

where $N_\theta \triangleq [C\ D]M_\theta$.

The existence of constraints (4.3) limits the set of admissible nominal steady states and inputs and the set of admissible nominal controlled variables, which are given by

$$\begin{aligned} \mathscr{Z}_s &\triangleq \{(x_s,u_s) \in \mathscr{Z} : (A-I_n)x_s + Bu_s = \mathbf{0}_{n,1}\} \\ \mathscr{Y}_s &\triangleq \{Cx_s + Du_s : (x_s,u_s) \in \mathscr{Z}_s\} \end{aligned}$$

Invariant set for tracking

Consider that the nominal system (4.2) is controlled by the following control law:

$$u = K(x-x_s) + u_s = Kx + L\theta \tag{4.7}$$

where $L = [-K\ I_m]M_\theta$. If K is such that matrix $A+BK$ is Hurwitz then this control law steers the system to the steady state and input $(x_s,u_s) = M_\theta\theta$. The existence of constraints limits the set of initial states and steady states and inputs that can admissibly be stabilized. This leads to the following definition.

Definition 1 (Invariant set for tracking) *An invariant set for tracking is the set of initial states and steady states and inputs (characterized by θ) that can be stabilized by the control law (4.7) fulfilling the constraints* (4.3) *throughout its evolution.*

This set can be computed as an admissible invariant set for the augmented system $x^a \triangleq (x,\theta) \in \mathbb{R}^{n+m}$. Then the closed-loop system can be posed as:

$$\underbrace{\begin{bmatrix} x \\ \theta \end{bmatrix}^+}_{x_a^+} = \underbrace{\begin{bmatrix} A+BK & BL \\ 0 & I_m \end{bmatrix}}_{A_a} \underbrace{\begin{bmatrix} x \\ \theta \end{bmatrix}}_{x_a} \tag{4.8}$$

subject to the set of constraints (4.3), that can be posed as

$$\mathscr{X}^a = \{x^a = (x,\theta) : (x, Kx+L\theta) \in \mathscr{Z}, M_\theta\theta \in \mathscr{Z}\}$$

Set $\Omega^a_{t,K} \subset \mathscr{X}^a$ is an admissible invariant set for tracking, for system (4.8) constrained to $\mathscr{X}^a$, if $A_a\Omega^a_t \subseteq \Omega^a_t$ and $\Omega^a_{t,K} \subseteq \mathscr{X}^a$. See that for any $(x(0),\theta) \in \Omega^a_{t,K}$, the trajectory of the system $x(i+1) = Ax(i)+Bu(i)$ controlled by $u(i) = Kx(i)+L\theta$ is confined in $\Omega_{t,K} = Proj_x(\Omega^a_{t,K})$[1] and tends to $(x_s,\bar{u_s}) = M_\theta\theta$.

[1] In what follows, superscript a denotes that set $\Omega^a_{t,K}$ is defined in the augmented state, while no superscript denotes that set $\Omega_{t,K}$ is defined in the state vector space x, i.e. $\Omega_{t,K} = Proj_x(\Omega^a_{t,K})$.

Although the maximal invariant set is not needed, it is convenient in order to provide a bigger region of attraction. The maximal admissible invariant set for system (4.8) may not be finitely determined due to the unitary eigenvalues of the plant. Fortunately, in this case, taking as constraints $\mathscr{X}_\lambda^a = \{x^a = (x,\theta) : (x, Kx+L\theta) \in \mathscr{Z}, M_\theta\theta \in \lambda\mathscr{Z}\}$, the associated maximal admissible invariant set is finitely determined for any $\lambda \in (0,1)$, resulting in a polyhedral region [15, 14]. Thus taking a λ arbitrarily close to 1, the resulting invariant set is arbitrarily close (in the Hausdorff sense) to the maximal one.

It is interesting to characterize what will be the set of the steady states, inputs and controlled variables that could be reached from an initial state contained $\Omega_{t,K}$. This can be done by defining the following set of parameters Θ

$$\Theta \stackrel{\Delta}{=} \{\theta : (x_s,u_s) = M_\theta\theta \in \mathscr{Z}, \quad x_s \in \Omega_{t,K}\} \tag{4.9}$$

This set is equal to the projection of $\Omega_{t,K}^a$ onto θ. Then the set of reachable steady controlled variables s is given by

$$\mathscr{Y}_t = N_\theta\Theta \tag{4.10}$$

Notice that if the calculation method proposed in [15] is used to compute $\Omega_{t,K}^a$, then this set $\mathscr{Y}_t$ is potentially equal to the maximal one $\mathscr{Y}_s$ since $\mathscr{Y}_t \subseteq \lambda\mathscr{Y}_s$ and λ can be chosen arbitrarily close to 1.

4.2.1 Optimization problem

In this section the proposed MPC for tracking is presented. As was previously stated, this predictive controller is based on the addition of the steady state and input as decision variables, the usage of a modified cost function and an extended terminal constraint. To this end, the following assumption is considered.

Assumption 1

1. *Let $Q \in \mathbb{R}^{n\times n}$, $R \in \mathbb{R}^{m\times m}$ and $T \in \mathbb{R}^{n\times n}$ be positive definite matrices.*
2. *Let $K \in \mathbb{R}^{m\times n}$ be a stabilizing control gain such that $(A+BK)$ is Hurwitz.*
3. *Let $P \in \mathbb{R}^{n\times n}$ be a positive definite matrix such that*

$$(A+BK)^T P(A+BK) - P = -(Q+K^T RK)$$

4. *Let $\Omega_{t,K}^a \subseteq \mathbb{R}^{n+n_\theta}$ be an admissible polyhedral invariant set for tracking for system* (4.2) *subject to* (4.3) *and a gain controller K.*

Consider that the current state of the system is x and the desired steady output to be reached is s, then the proposed cost function is

$$\begin{aligned} V_N(x,s,\mathbf{u},\theta) \quad = \quad & \sum_{i=0}^{N-1} \|x(i)-x_s\|_Q^2 + \|u(i)-u_s\|_R^2 \\ & + \|x(N)-x_s\|_P^2 + \|y_s - s\|_T^2 \end{aligned}$$

where $\mathbf{u}$ is a sequence of N future control inputs, i.e. $\mathbf{u} = \{u(0),\ldots,u(N-1)\}$, $z_s = (x_s,u_s) = M_\theta\theta$, $y_s = N_\theta\theta$, $x(i)$ is the predicted state of the system at time i given by $x(i+1) = Ax(i)+Bu(i)$, with $x(0) = x$. Note that this cost can be posed as a quadratic function of the decision variables.

The proposed MPC optimization problem $P_N(x,s)$ is given by

$$\begin{aligned} V_N^*(x,s) \quad &= \quad \min_{\mathbf{u},\theta} V_N(x,s,\mathbf{u},\theta) \\ &s.t. \quad x(0) = x, \\ &\quad x(j+1) = Ax(j) + Bu(j), \\ &\quad (x(j),u(j)) \in Z, \qquad j = 0,\cdots,N-1 \\ &\quad (x_s,u_s) = M_\theta \theta, \\ &\quad (x(N),\theta) \in \Omega_{t,K}^a. \end{aligned}$$

It is worth noting that $\mathbf{u}$ and θ are the decision variables and x and s are parameters of the proposed optimization problem $P_N(x,s)$. Moreover, it turn out to be a standard (parametric) Quadratic Programming problem that can be efficiently solved by specialized algorithms.

Given that the constraints of $P_N(x,s)$ do not depend on s, there exists a (polyhedral) region $X_N \subset \mathbb{R}^n$ such that for all $x \in X_N$, $P_N(x,s)$ is feasible (for any $s \in \mathscr{Y}_t$). Applying the receding horizon strategy, the control law is given by $K_N(x,s) = u(0)$, where $u(0)$ is a function of x and s.

Theorem 1 (Stability) *Consider that assumption 1 hold. Given $\lambda \in (0,1)$, suppose that $\Omega_{t,K}^a$ is an admissible invariant set for tracking. Then, for any feasible initial state $x_0 \in X_N$ and for any desired steady state $s \in \mathscr{Y}_t$, the proposed MPC controller $K_N(x,s)$ asymptotically steers the system to s in an admissible way.*

Property 1 *The set of admissible steady outputs that can be tracked without offset is $\mathscr{Y}_t$. Since the evolution of the system remains in X_N, the system can be steered to any admissible reference. Then, any sequence of piecewise admissible references can be tracked without offset.*

If the desired steady output s is not admissible, then it cannot be tracked without offset and the controller steers the system to a close admissible steady output.

Property 2 *Consider a desired admissible set point $s \in \mathscr{Y}_t$ and design a standard MPC i.e. translating the system to the corresponding equilibrium steady state x_s to s, using a linear stabilizing local control law and the maximal admissible invariant set $\mathscr{O}_\infty(x_s)$ as terminal set. Also consider the proposed MPC where the local controller gain is the same and the set $\Omega_{t,K}^a$ is used as terminal cost. Then:*

1. *Since $\mathscr{O}_\infty(x_s) \subseteq \Omega_{t,K}^a$, the domain of attraction of the proposed MPC is larger than that of the standard MPC.*

2. *A desirable property of the MPC controllers is that if the unconstrained optimal control law is used as terminal controller, then the MPC is locally optimal. In the proposed MPC for tracking, this property is lost due to the term $\|y_s - s\|_T$ added in the cost function. However, it can be proved that if this term is more heavily penalized, then the local optimality of the controller is enhanced.*

Thus, taking an arbitrarily large matrix T, the MPC for tracking provides a larger domain of attraction and a control law which is locally nearly optimal.

Property 3 *The proposed controller stabilizes the system for any suboptimal solution such that*

- *The suboptimal cost at each sample time is lower that the one at the previous instant.*

- *If the system is in $\Omega^q_{t,K}$, the suboptimal cost must be lower than the one obtained by the local linear controller.*

Property 4 *If the set point to track is time varying convergent to a steady value, then the proposed controller makes the system follow it in an admissible way ultimately reaching the steady set point.*

4.2.2 Application to the quadruple tank process

The discrete-time model of the plant for the aforementioned parameters is:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}^+ = \begin{bmatrix} 0.9705 & 0 & 0.0205 & 0 \\ 0 & 0.9661 & 0 & 0.0195 \\ 0 & 0 & 0.9792 & 0 \\ 0 & 0 & 0 & 0.9802 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0.0068 & 0.0001 \\ 0.0002 & 0.0091 \\ 0 & 0.0137 \\ 0.0160 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

The defining matrices of the stage cost of the performance criterion have been chosen as

$$Q = 100 \times I_4 \quad R = 1 \times I_2 \tag{4.11}$$

The terminal control gain K has been chosen as the LQR gain for the matrices (4.11) and matrix P is the solution of the Ricatti equation. The offset cost weighting matrix T has been chosen as $T = 10^4 \times I_2$. Finally the control horizon has been chosen as $N = 5$.

$$K = \begin{bmatrix} -2.2525 & -1.6731 & 0.8413 & -8.0546 \\ -2.0469 & -3.1698 & -7.7111 & 0.6285 \end{bmatrix}$$

The resulting regions are shown in Fig. 4.6.

The derived controller has been tested on the nominal model, and then applied on the real plant. Fig. 4.7 shows the simulated evolution of levels h_1 and h_2, and the references, the evolution of the control actions and the evolution of the levels h_3 and h_4

Fig. 4.8 shows the same as the previous figure, but applied to the real plant. It can be appreciate that:

- The behavior of the plant is quite similar to the model, but the model it is not well identified because there is offset respect the first reference, which is, more or less, the linearizing point of the plant. The MPC for tracking may present offset in any change of reference due to the error between the model and the real plant, but not in the linearizing point.

- The integral of the square error SEI is

  $$SEI = 174.2441$$

  This value will be decreased when we get an more accurate model of the plant and when we introduce the offset cancellation loop that will remove the offset in permanent regime.

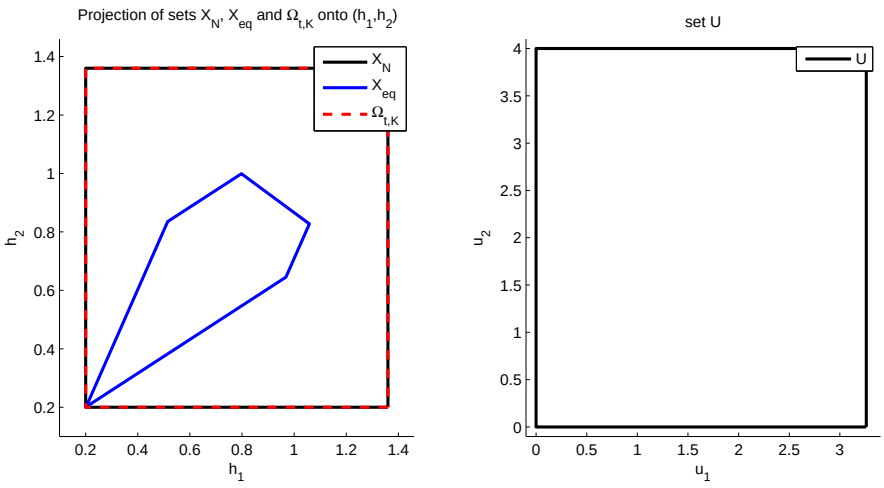


Figure 4.6: Different sets of the MPC for tracking applied to the quadruple tank process

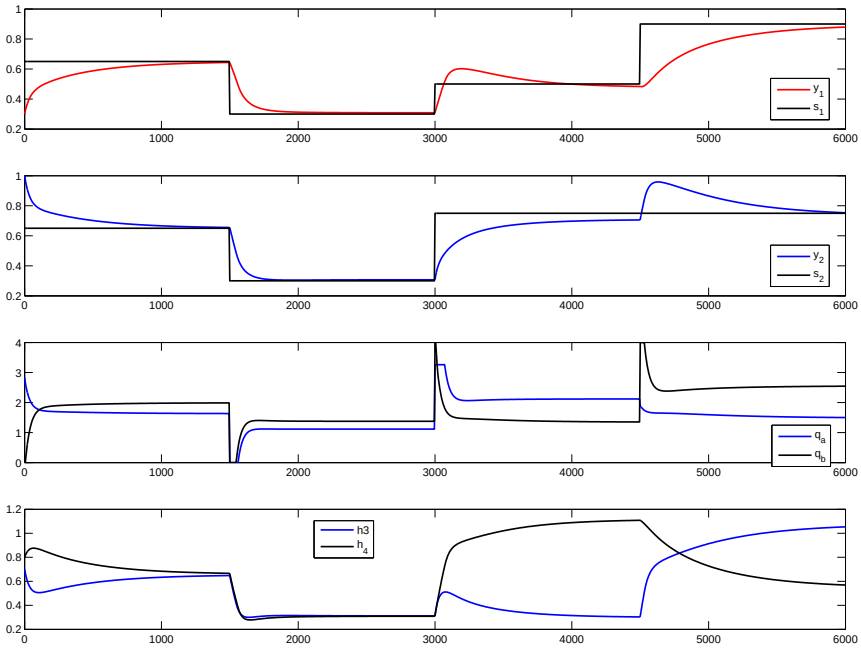


Figure 4.7: Simulation of the plant controlled by the centralized MPC for tracking

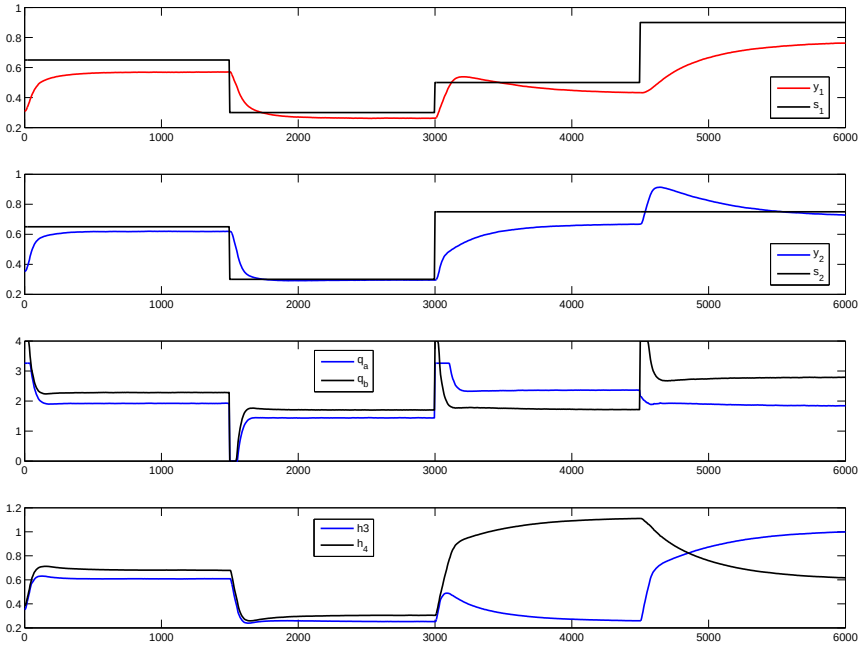


Figure 4.8: Evolution of the plant controlled by the centralized MPC for tracking

4.3 Decentralized MPC for tracking

4.3.1 Pairing based on the relative gain array

Two controllers are going to be used to control the plant. The pairing between inputs and outputs in decided based on the relative gain array (*RGA*)

$$RGA = \begin{bmatrix} -0.4 & 1.38 \\ 1.38 & -0.4 \end{bmatrix}$$

Considering the values of the RGA the correct pairing is control h_1 with q_b (y_1 with u_2) and h_2 with q_a (y_2 with u_1). The models of the subsystems are:

- Subsystem 1 (y_1 with u_2)

$$\begin{aligned} \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}^+ &= \begin{bmatrix} 0.9705 & 0.0205 \\ 0 & 0.9792 \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0.0001 \\ 0.0137 \end{bmatrix} u_2 \\ y_1 &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} \end{aligned} \tag{4.12}$$

- Subsystem 2 (y_2 with u_1)

$$\begin{bmatrix} x_2 \\ x_4 \end{bmatrix}^+ = \begin{bmatrix} 0.9661 & 0.0195 \\ 0 & 0.9802 \end{bmatrix} \begin{bmatrix} x_2 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0.0002 \\ 0.0160 \end{bmatrix} u_1 \qquad (4.13)$$

$$y_2 = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ x_4 \end{bmatrix}$$

The matrices that define the stage cost of the performance criterion have been chosen in both cases as:

$$Q = 100 \times I_2 \quad R = 1 \times I_1 \qquad (4.14)$$

The terminal control gains K_1 and K_2 have been chosen as the LQR gain for the matrices (4.14) and matrix P is the solution of the Ricatti equation. The offset cost weighting matrix T has been chosen as $T = 10^4 \times I_1$. Finally the control horizon has been chosen as $N = 5$.

$$K_1 = \begin{bmatrix} -2.3445 & -8.3902 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} -1.9719 & -8.4094 \end{bmatrix}$$

The resulting regions are shown in Fig. 4.9.

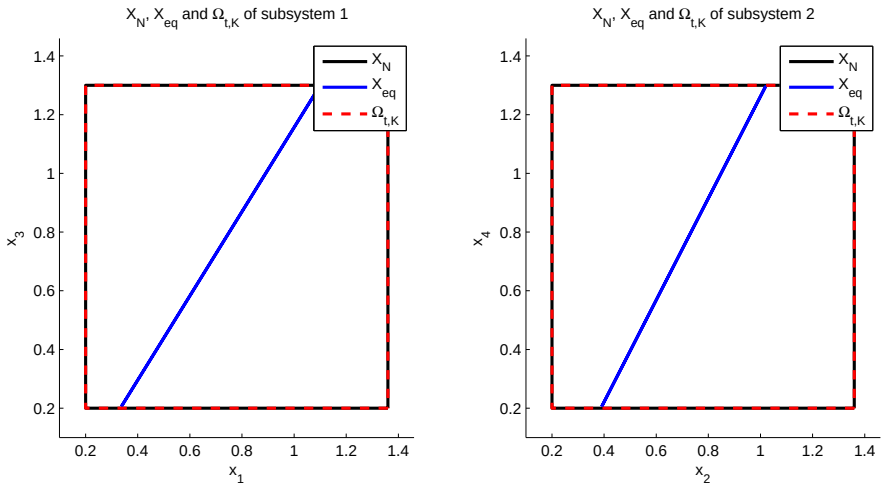


Figure 4.9: Different sets of the MPC for tracking

where

$$0 \leq u_1 \leq 4.00 \quad 0 \leq u_2 \leq 3.26$$

The derived controllers has been tested on the nominal model, and then applied on the real plant. Fig. 4.10 shows the simulated evolution of levels h_1 and h_2, and the references, the evolution of the control actions and the evolution of the levels h_3 and h_4

Fig. 4.11 shows the same as the previous figure, but applied to the real plant. The integral of the square error SEI is

$$SEI = 276.8838$$

This value will be decreased when we get an more accurate model of the plant and when we introduce the offset cancellation loop that may remove the offset in permanent regime.

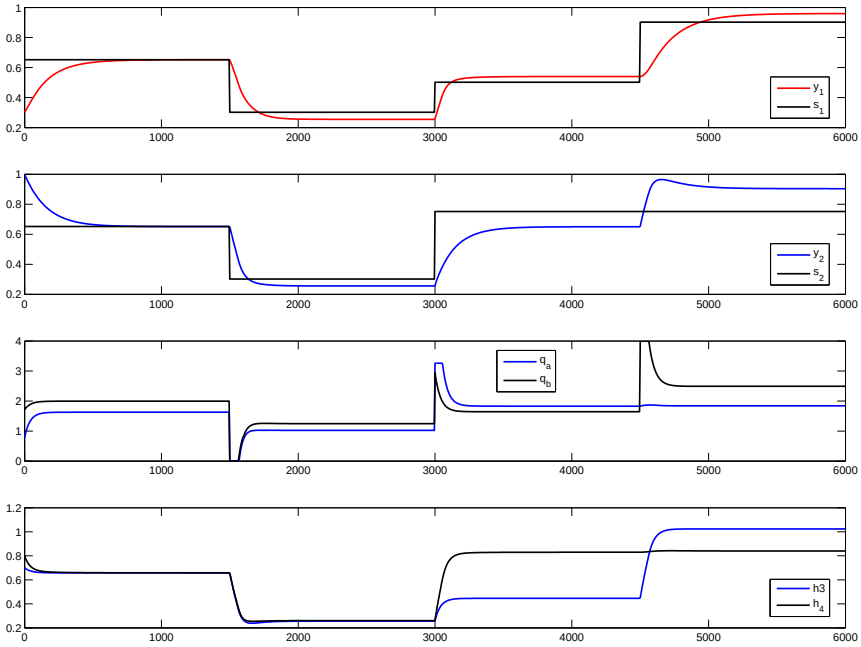


Figure 4.10: Simulation of the plant controlled by 2 MPC for tracking with the correct pairing

4.3.2 Wrong pairing

Considering the values of the RGA the wrong pairing is control h_1 with q_a (y_1 with u_1) and h_2 with q_b (y_2 with u_2). The models of the subsystems are:

- Subsystem 1 (y_1 with u_1)

$$\begin{bmatrix} x_1 \\ x_3 \end{bmatrix}^+ = \begin{bmatrix} 0.9705 & 0.0205 \\ 0 & 0.9792 \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0.0068 \\ 0 \end{bmatrix} u_1 \qquad (4.15)$$
$$y_1 = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}$$

- Subsystem 2 (y_2 with u_2)

$$\begin{bmatrix} x_2 \\ x_4 \end{bmatrix}^+ = \begin{bmatrix} 0.9661 & 0.0195 \\ 0 & 0.9802 \end{bmatrix} \begin{bmatrix} x_2 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0.0091 \\ 0 \end{bmatrix} u_2 \qquad (4.15)$$
$$y_2 = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ x_4 \end{bmatrix}$$

The defining matrices of the stage cost of the performance criterion have been chosen in both cases as:

$$Q = 100 \times I_2 \quad R = 1 \times I_1 \qquad (4.14)$$

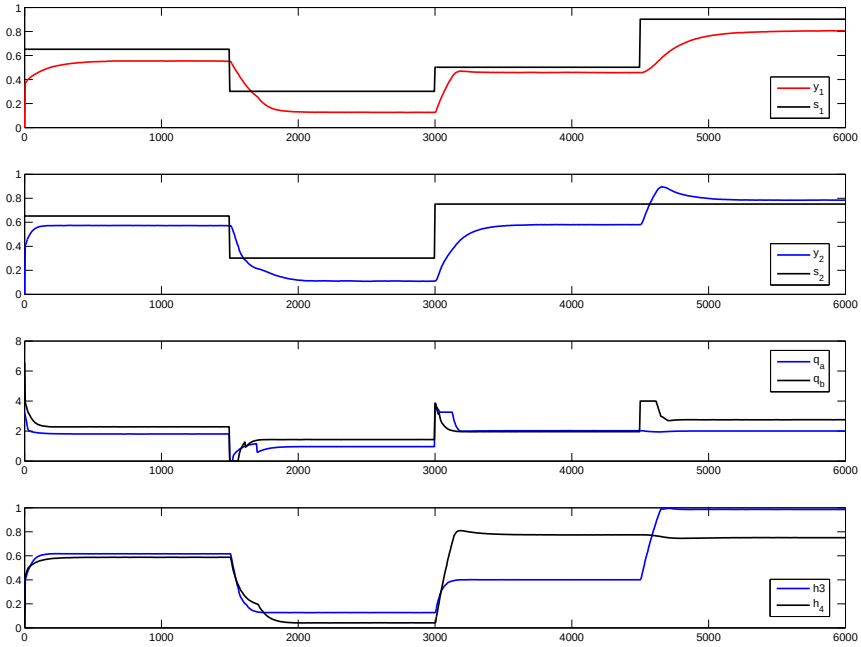


Figure 4.11: Evolution of the plant controlled by 2 MPC for tracking with the correct pairing

The terminal control gains K_1 and K_2 have been chosen as the LQR gain for the matrices (4.14) and matrix P is the solution of the Ricatti equation. The offset cost weighting matrix T has been chosen as $T = 10^4 \times I_1$. Finally the control horizon has been chosen as $N = 5$.

$$K_1 = \begin{bmatrix} -6.3344 & -1.4545 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} -6.6115 & -1.1918 \end{bmatrix}$$

The resulting regions are shown in Fig. 4.12.

where

$$0 \leq u_1 \leq 4.00 \quad 0 \leq u_2 \leq 3.26$$

The derived controllers has been tested on the nominal model, and then applied on the real plant. Fig. 4.13 shows the simulated evolution of levels h_1 and h_2, and the references, the evolution of the control actions and the evolution of the levels h_3 and h_4

Fig. 4.14 shows the same as the previous figure, but applied to the real plant. The integral of the square error *SEI* is not calculated because the test is not finished due to the maximum level of tank 3 was reached and the security system stopped the pump. It is clear that the performance of centralized controller is better than the decentralized one. Once we develop the tracking formulation of the distributed controller the performance of it must be between the centralized and the decentralized ones.

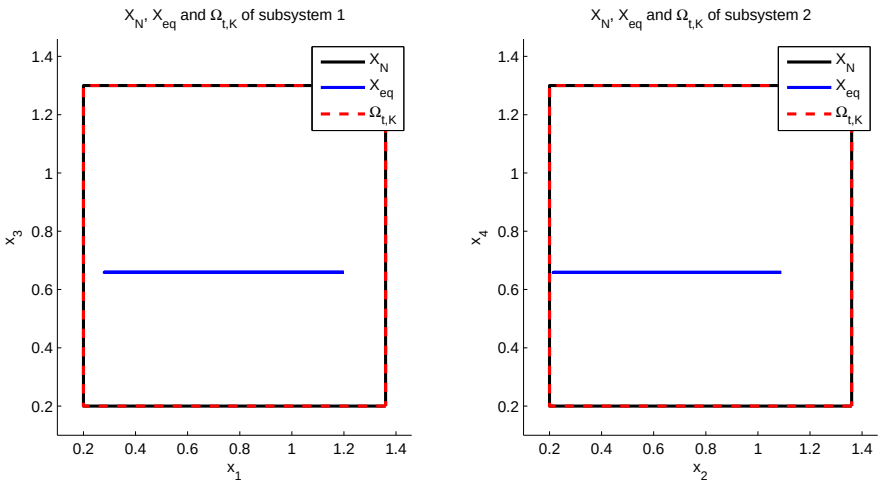


Figure 4.12: Different sets of the MPC for tracking

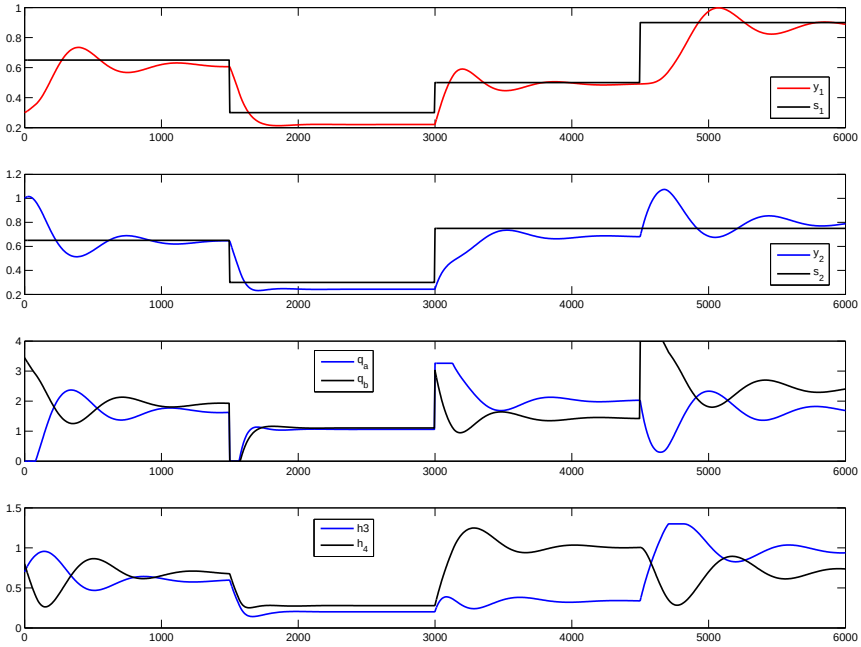


Figure 4.13: Simulation of the plant controlled by 2 MPC for tracking with the wrong pairing

4.4 Distributed MPC based on a cooperative game

In this section we compare a centralized MPC with the distributed MPC based on a cooperative game scheme presented in [16]. We consider the following class of distributed linear systems in which two

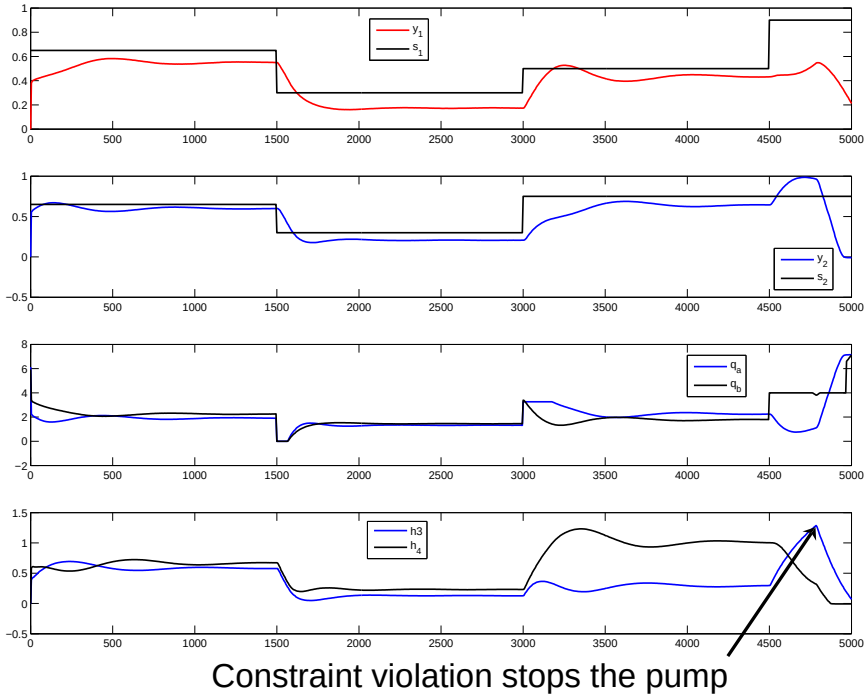


Figure 4.14: Evolution of the plant controlled by 2 MPC for tracking with the wrong pairing

subsystems coupled with the neighbor subsystem through the inputs are defined

$$\begin{aligned} x_1(t+1) &= A_1 x_1(t) + B_{11} u_1(t) + B_{12} u_2(t) \\ x_2(t+1) &= A_2 x_2(t) + B_{21} u_1(t) + B_{22} u_2(t) \end{aligned} \tag{4.15}$$

where $x_i \in \mathbb{R}^{n_i}$, $i = 1,2$ are the states of each subsystem and $u_i \in \mathbb{R}^{m_i}$, $i = 1,2$ are the different inputs. This class of systems are of relevance when identifications techniques are used to obtain the transfer function of a process. We consider the following linear constraints in the state and the inputs

$$x_i \in \mathscr{X}_i, u_i \in \mathscr{U}_i,\ i = 1,2$$

where $\mathscr{X}_i$ and $\mathscr{U}_i$ with $i = 1,2$ are defined by a set of linear inequalities.

The control objective is to regulate the system to the origin while guaranteeing that the constraints are satisfied. Centralized MPC solves a single optimization problem to decide the optimal sequences of the inputs u_1 and u_2 with respect to a given performance index based on the full model of the system and on measurements from all the sensors. In distributed and decentralized schemes two independent controllers (hereby denoted agents) are defined. Agent 1 has access to the model of subsystem 1, its state x_1 and decides the value of u_1. On the other hand, agent 2 has access to the model of subsystem 2, its state x_2 and decides the value of u_2. This implies that neither agent has access to the full model or state information and that in order to find a cooperative solution, they must communicate. The proposed controller guarantees practical stability of the closed-loop system.

The proposed benchmark is based on a nonlinear model and consists of several reference steps of the levels of tanks 1 and 2. In order to test the proposed DMPC scheme and compare its performance

with a centralized controller we propose to design a different controller for each set-point; that is, we consider four different linear models with the following state and input variables

$$x_1 = \begin{bmatrix} h_1 - h_{1r} \\ h_3 - h_{3r} \end{bmatrix} u_1 = \begin{bmatrix} q_a - q_{ar} \end{bmatrix}$$

$$x_2 = \begin{bmatrix} h_2 - h_{2r} \\ h_4 - h_{4r} \end{bmatrix} u_2 = \begin{bmatrix} q_b - q_{br} \end{bmatrix}$$

where $h_{1r}, h_{2r}, h_{3r}, h_{4r}, q_{ar}$ and q_{br} define the level and flow of each of the four different set-points of the benchmark. The linear model for each set-point is obtained linearizing the nonlinear model of the quadruple tank process. This implies that during the simulation, we switch between four different controllers for regulation.

The objective of the MPC controllers is to minimize a performance index that depends on the future evolution of both states and inputs based on the following local cost functions

$$J_1(x_1, U_1, U_2) = \sum_{k=0}^{N-1} L_1(x_{1,k}, u_{1,k}) + F_1(x_{1,N})$$
$$J_2(x_2, U_2, U_1) = \sum_{k=0}^{N-1} L_2(x_{2,k}, u_{2,k}) + F_2(x_{2,N})$$

where $L_i(x,u) = x^T Q_i x + u^T R_i u$ and $F_i(x) = x^T P_i x$ with $i = 1,2$ are the stage and terminal cost functions respectively.

Centralized MPC solves a single large-scale problem based on the model of the whole system. In the example section we will compare the performance of the proposed approach with a centralized MPC controller based on the following optimization problem:

$$\begin{aligned} \{U_1^c(t), U_2^c(t)\} = \arg\min_{U_1,U_2} \quad & J_1(x_1(t), U_1, U_2) + J_2(x_1(t), U_2, U_1) \\ & x_{1,k+1} = A_1 x_{1,k} + B_{11} u_{1,k} + B_{12} u_{2,k} \\ & x_{1,0} = x_1(t) \\ & x_{1,k} \in \mathscr{X}_1,\ k = 0, \dots N \\ & u_{1,k} \in \mathscr{U}_1,\ k = 0, \dots N-1 \\ & x_{1,N} \in \Omega_1 \\ & x_{2,k+1} = A_2 x_{2,k} + B_{22} u_{2,k} + B_{21} u_{1,k} \\ & x_{2,0} = x_2(t) \\ & x_{2,k} \in \mathscr{X}_2,\ k = 0, \dots N \\ & u_{2,k} \in \mathscr{U}_2,\ k = 0, \dots N-1 \\ & x_{2,N} \in \Omega_2 \end{aligned} \tag{4.16}$$

where U_i are the decision variables of the optimization problems solved by both agents. The centralized MPC provides in general the best closed-loop performance, but can only be applied when it is possible to control the system with a single controller that has access to the full model and state of the same. Note that this formulation takes into account both a terminal cost and a terminal region which can be designed to guarantee closed-loop stability. In the benchmark we have used the design procedure proposed in [16] to obtain matrices the P_1, P_2 and the sets Ω_1, Ω_2.

The objective of the DMPC scheme is to minimize the performance index in a distributed manner. At each sampling time, each agent solves a sequence of reduced dimension optimization problems based on the model of its subsystem and assuming a given fixed input trajectory for its neighbor. The proposed DMPC algorithm is the following:

1. At time step t, each agent i receives its corresponding partial state measurement $x_i(t)$.

2. Each agent i minimizes J_i assuming that the neighbor keeps applying the optimal trajectory evaluated at the previous time step; that is, $U_{ni} = U_{ni}^s(t)$. These trajectories are obtained from the optimal input sequence of agent i at time $t-1$, denoted $U_i^d(t-1)$, as follows:

$$U_1^s(t) = \begin{bmatrix} u_{1,1}^d \\ u_{1,2}^d \\ \vdots \\ u_{1,N-1}^d \\ K_1 x_{1,N} \end{bmatrix}, \; U_2^s(t) = \begin{bmatrix} u_{2,1}^d \\ u_{2,2}^d \\ \vdots \\ u_{2,N-1}^d \\ K_2 x_{2,N} \end{bmatrix}$$

where $x_{1,N}, x_{2,N}$ are the N-steps ahead predicted state obtained from $x_1(t-1), x_2(t-1)$ respectively applying the input trajectories $U_1^d(t-1), U_2^d(t-1)$ and K_1, K_2 are two known feedback gains. Agent 1 solves the following optimization problem:

$$\begin{aligned} U_1^*(t) = \arg\min_{U_1} \quad & J_1(x_1(t), U_1, U_2^s(t)) \\ & x_{1,k+1} = A_1 x_{1,k} + B_{11} u_{1,k} + B_{12} u_{2,k} \\ & x_{1,0} = x_1(t) \\ & x_{1,k} \in \mathscr{X}_1, \; k = 0, \dots N \\ & u_{1,k} \in \mathscr{U}_1, \; k = 0, \dots N-1 \\ & x_{1,N} \in \Omega_1 \end{aligned} \tag{4.17}$$

Agent 2 solves the following optimization problem:

$$\begin{aligned} U_2^*(t) = \arg\min_{U_2} \quad & J_2(x_2(t), U_2, U_1^s(t)) \\ & x_{2,k+1} = A_2 x_{2,k} + B_{22} u_{2,k} + B_{21} u_{1,k} \\ & x_{2,0} = x_2(t) \\ & x_{2,k} \in \mathscr{X}_2, \; k = 0, \dots N \\ & u_{2,k} \in \mathscr{U}_2, \; k = 0, \dots N-1 \\ & x_{2,N} \in \Omega_2 \end{aligned} \tag{4.18}$$

The sets Ω_1 and Ω_2 define the terminal region constraints that are necessary to prove closed-loop practical stability following a terminal region/terminal cost approach. Note that in both optimization problems the free variable is U_i while the neighbor input trajectory U_{ni} is fixed.

3. Each agent i minimizes J_i optimizing the neighbor input assuming that it applies the input trajectory computed in the previous optimization problem U_i^*. Agent 1 solves the following optimization problem:

$$\begin{aligned} U_2^w(t) = \arg\min_{U_2} \quad & J_1(x_1(t), U_1^*(t), U_2) \\ & x_{1,k+1} = A_1 x_{1,k} + B_{11} u_{1,k} + B_{12} u_{2,k} \\ & x_{1,0} = x_1(t) \\ & x_{1,k} \in \mathscr{X}_1, \; k = 0, \dots N \\ & u_{2,k} \in \mathscr{U}_2, \; k = 0, \dots N-1 \\ & x_{1,N} \in \Omega_1 \end{aligned} \tag{4.19}$$

Table 4.1: Cost function table used for the decision making.

	$U_2^s(t)$	$U_2^*(t)$	$U_2^w(t)$
$U_1^s(t)$	$J_1(x_1(t),U_1^s(t),U_2^s(t))$ $+J_2(x_2(t),U_2^s(t),U_1^s(t))$	$J_1(x_1(t),U_1^s(t),U_2^*(t))$ $+J_2(x_2(t),U_2^*(t),U_1^s(t))$	$J_1(x_1(t),U_1^s(t),U_2^w(t))$ $+J_2(x_2(t),U_2^w(t),U_1^s(t))$
$U_1^*(t)$	$J_1(x_1(t),U_1^*(t),U_2^s(t))$ $+J_2(x_2(t),U_2^s(t),U_1^*(t))$	$J_1(x_1(t),U_1^*(t),U_2^*(t))$ $+J_2(x_2(t),U_2^*(t),U_1^*(t))$	$J_1(x_1(t),U_1^*(t),U_2^w(t))$ $+J_2(x_2(t),U_2^w(t),U_1^*(t))$
$U_1^w(t)$	$J_1(x_1(t),U_1^w(t),U_2^s(t))$ $+J_2(x_2(t),U_2^s(t),U_1^w(t))$	$J_1(x_1(t),U_1^w(t),U_2^*(t))$ $+J_2(x_2(t),U_2^*(t),U_1^w(t))$	$J_1(x_1(t),U_1^w(t),U_2^w(t))$ $+J_2(x_2(t),U_2^w(t),U_1^w(t))$

Agent 2 solves the following optimization problem:

$$\begin{aligned} U_1^w(t) = \arg\min_{U_1} \quad & J_2(x_2(t),U_2^*(t),U_1) \\ & x_{2,k+1} = A_2 x_{2,k} + B_{22}u_{2,k} + B_{21}u_{1,k} \\ & x_{2,0} = x_2(t) \\ & x_{2,k} \in \mathscr{X}_2,\ k = 0,\dots N \\ & u_{1,k} \in \mathscr{U}_1,\ k = 0,\dots N-1 \\ & x_{2,N} \in \Omega_2 \end{aligned} \tag{4.20}$$

In this optimization problem the free variable is U_{ni} (the input trajectory U_i is fixed). Solving this optimization problem, agent i defines an input trajectory for its neighbor that optimizes its local cost function J_i.

4. Both agents communicate. Agent 1 sends $U_1^*(t)$ and $U_2^w(t)$ to agent 2 and receives $U_2^*(t)$ and $U_1^w(t)$.

5. Each agent evaluates the local cost function J_i for each the nine different possible combination of input trajectories; that is $U_1 \in \{U_1^s(t),U_1^w(t),U_1^*(t)\}$ and $U_2 \in \{U_2^s(t),U_2^w(t),U_2^*(t)\}$.

6. Both agents communicate and share the information of the value of local cost function for each possible combination of input trajectories. In this step, both agents receive enough information to take a cooperative decision.

7. Each agent applies the input trajectory that minimizes $J = J_1 + J_2$. Because both agents have access to the same information after the second communication cycle, both agents choose the same optimal input sets. We denote the chosen set of input trajectories as $U_1^d(t),U_2^d(t)$.

8. The first input of each optimal sequence is applied and the procedure is repeated the next sampling time.

From a game theory point of view, at each time step both agents are playing a cooperative game. This game can be synthesized in strategic form by a three by three matrix. Each row represents one of the three possible decisions of agent 1, and each column represents one of the three possible decisions of agent 2. The cells contain the sum of the cost functions of both agents for a particular choice of future inputs. At each time step, the option that yields a lower global cost is chosen. Note that both agents share this information, so they both choose the same option. The nine possibilities are shown in table 4.1.

At each sampling time, the controllers decide among three different options. The shifted optimal input trajectory $U_i^s(t)$ keeps applying the latest optimal trajectory. The *selfish* option $U_i^*(t)$ provides the best improvement in J_i if the rest of the system's manipulated variables stay unchanged. The *altruist* option $U_i^w(t)$ provides the best improvement for the neighbor agent cost function J_2. In this case, the agent i sacrifices its own welfare in order to improve the overall performance.

4.4.1 Design procedure

The distributed controller that has been presented is designed for regulation problems. Given that the benchmark puts to test the controller in four different working points four different versions of the controller were designed, each one linearized in a working point. Whenever the reference changes the controller is switched. Next we will reproduce here the results for the first working point for the plant. We omit the results for the rest of working points because the design procedure is the same and the results are similar.

In this point we have to remark the fact that when the reference is switched from one working point to another one it is precise to reset the value of U_s to a feasible solution. This is necessary in order to assure a decreasing cost and the stability.

The reference is given by (0.65, 0.65). The linearized model of the first agent is:

$$A_1 = \begin{bmatrix} 0.9705 & 0.0205 \\ 0 & 0.9792 \end{bmatrix}, B_{11} = \begin{bmatrix} 0.0068 \\ 0 \end{bmatrix}, B_{12} = \begin{bmatrix} 0.0001 \\ 0.0137 \end{bmatrix}$$

The model of the second agent is given by:

$$A_1 = \begin{bmatrix} 0.9661 & 0.0195 \\ 0 & 0.9802 \end{bmatrix}, B_{11} = \begin{bmatrix} 0.0002 \\ 0.016 \end{bmatrix}, B_{12} = \begin{bmatrix} 0.0091 \\ 0 \end{bmatrix}$$

The controller gains were designed for the following weighting matrices $Q_1 = Q_2 = diag(100, 0)$, $R_1 = R_2 = I$. The local controller gains for each agent were,

$$K_1 = \begin{bmatrix} 0.17 & 0.21 \\ 0.00 & 0.00 \end{bmatrix}, K_2 = \begin{bmatrix} 0.00 & 0.00 \\ -0.16 & -0.14 \end{bmatrix}$$

These gains and the terminal cost matrices P_1, P_2 were designed with LMI techniques in order to stabilize both subsystems independently while assuring the stability of the centralized system. Following the procedure detailed previously it is possible to calculate a distributed invariant set corresponding to this gain. We show in Figs. 4.15 and 4.16 the sets corresponding to both agents. It is remarkable that the Chebyshev radius of the distributed invariant set is the same in this case that the centralized one.

The centralized gain is given by $K = diag(K_1, K_2)$. The role of the gain K is important because the option in the game that allows to guarantee closed-loop stability is constructed shifting the last centralized control action; that is, the first element is dropped after it is applied in the system and a term with the value Kx_N is added at the end of the horizon control vector. Although we talk here about the centralized gain K note that centralized here does not require a cooperation between the two agents because it is constructed in such a way that this cooperation can be avoided.

4.4.2 Simulation and experiment results

The derived controllers has been tested on the nominal model, and then applied on the real plant. Fig. 4.17 shows the simulated evolution of levels h_1 and h_2, and the references, the evolution of the control actions and the evolution of the levels h_3 and h_4 for the centralized MPC controller.

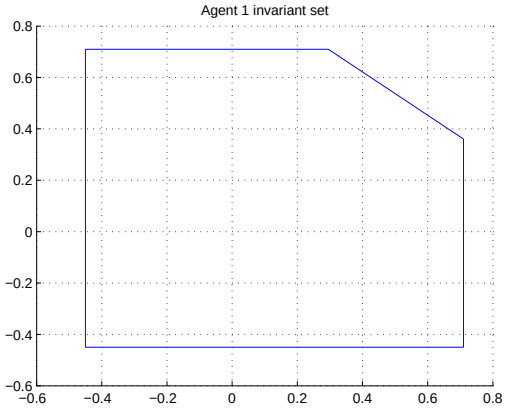


Figure 4.15: Invariant set for agent 1.

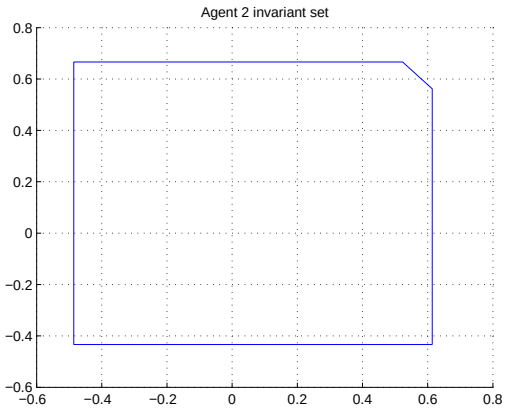


Figure 4.16: Invariant set for agent 2.

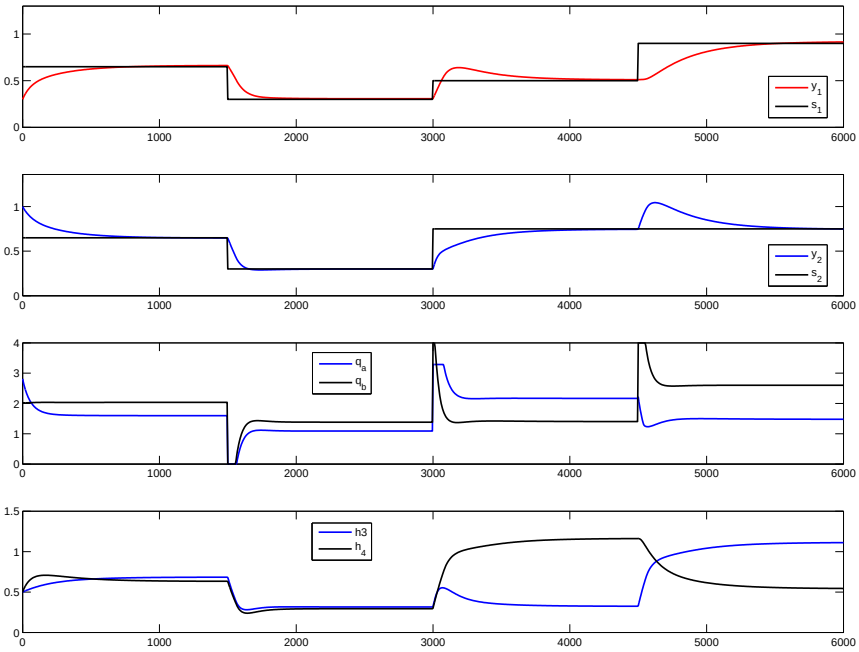


Figure 4.17: Simulation of the plant controlled by the centralized MPC for regulation

Fig. 4.18 shows the same as the previous figure, but applied to the real plant. The integral of the square error *SEI* is 92.9616.

Fig. 4.19 shows the simulated evolution of levels h_1 and h_2, and the references, the evolution of the control actions and the evolution of the levels h_3 and h_4 for the distributed MPC controller based on a cooperative game.

Fig. 4.20 shows the same as the previous figure, but applied to the real plant. The integral of the square error *SEI* is 200.3820.

During the simulations the distributed controller switches between the different control options. Fig. 4.21 shows the different option chosen at each time step.

The proposed distributed MPC controller only needs two communication steps in order to obtain a cooperative solution to the centralized optimization problem, has low communication and computational burdens and provides a feasible solution to the centralized problem. The simulation and experimental results show that the distributed scheme is able to control the system. Note that in this case, because the control input is decided by consensus, the pairing does not affect the performance of the distributed control scheme if the states are grouped correctly.

In a future work we will apply a distributed MPC scheme for tracking including an integral term in order to reduce the steady state error and the effect of the model mismatch.

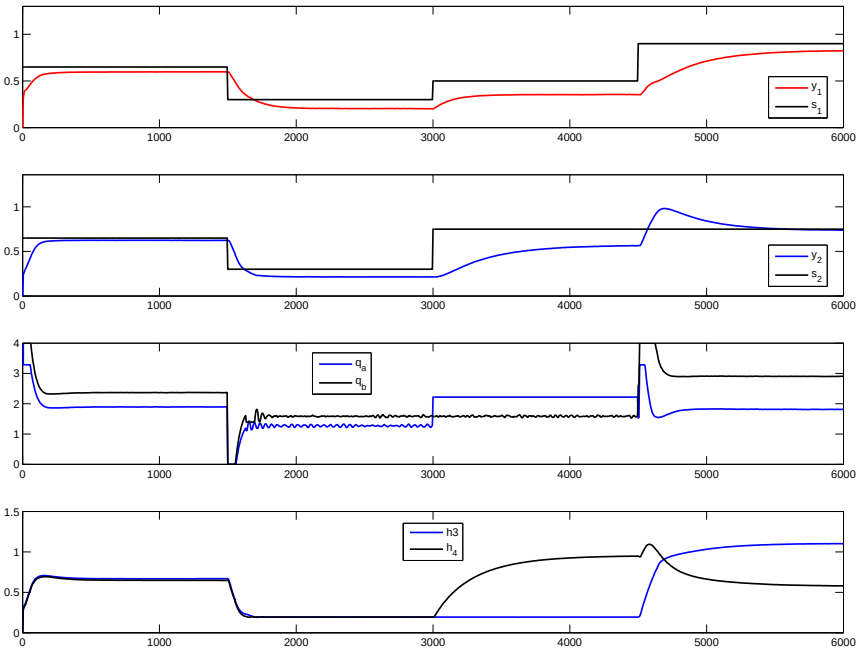


Figure 4.18: Evolution of the plant controlled by the centralized MPC for regulation

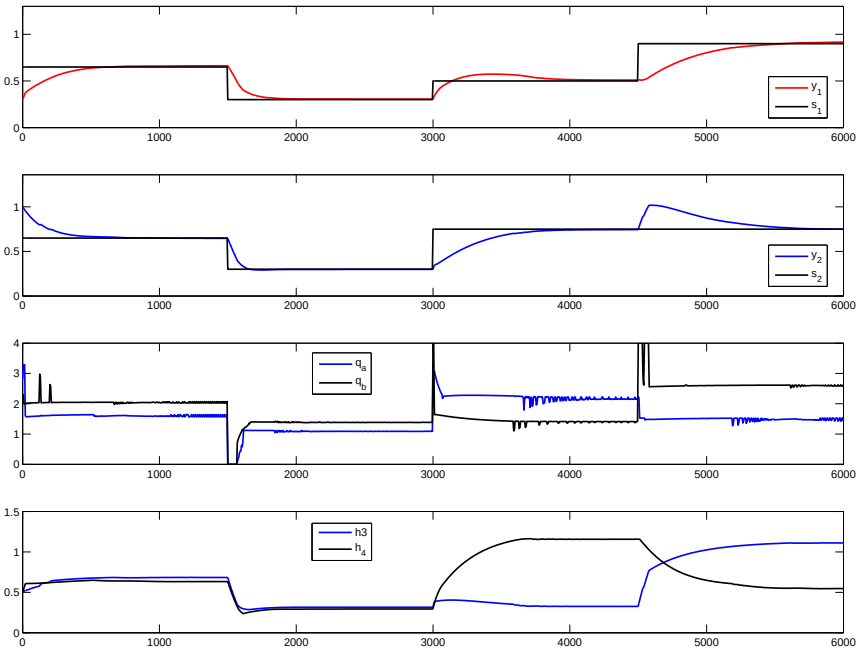


Figure 4.19: Simulation of the plant controlled by the distributed MPC for regulation

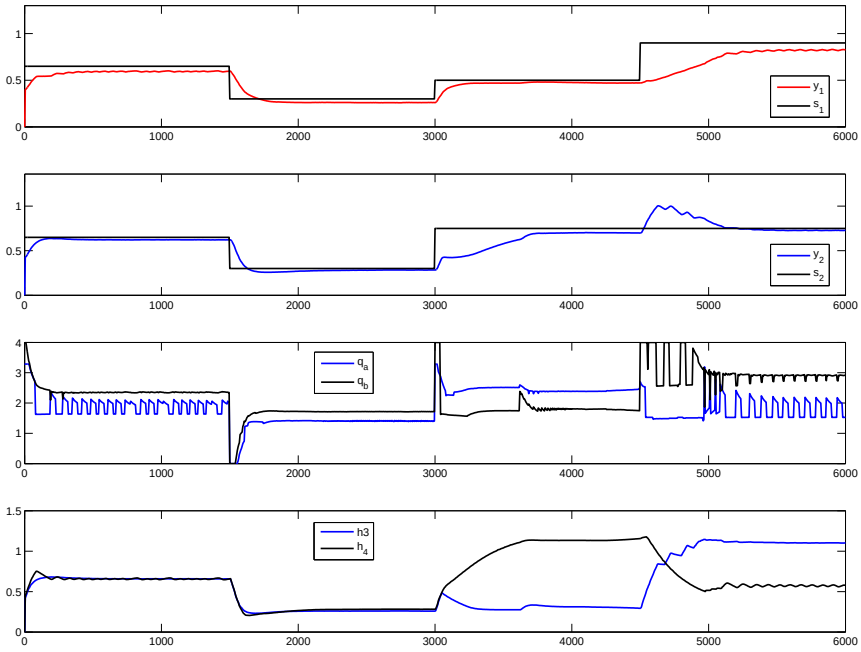


Figure 4.20: Evolution of the plant controlled by the distributed MPC for regulation

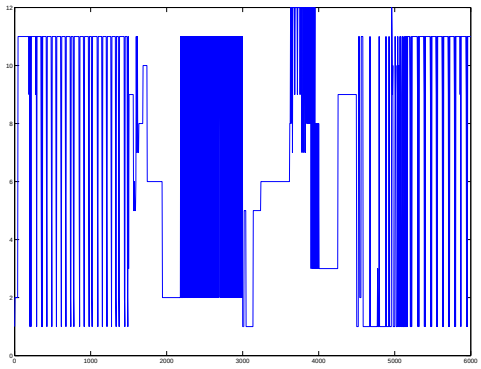


Figure 4.21: Options chosen at each time step by the distributed MPC controller.

Part III

Control scheme applied to the electrical generation units benchmark

Chapter 5

Model predictive control for generation units

5.1 Introduction

It is well know that generation units has two main dynamics:

1. Mechanical Dynamics: these dynamics depict the load angle, the mechanical power and the rotor speed behavior.
2. Electrical Dynamics: these dynamics depict the voltage at terminal bus and the field voltage behavior

The mechanical and electrical dynamics have different time-scale responses, being mechanical dynamics slower than electric ones. Then in order to design a MPC for generation units, a multi-model controller is necessary. Moreover taking into account that in the real systems classic speed and voltage regulators cannot be modified, these dynamics must be included in the model used to predict voltage and speed trajectories.

In this report, a centralized MPC is formulated for the control of generation units. Due to different time scale of machines dynamics, a two levels time-response-based hierarchical structure is proposed. The proposed control structure involves the interaction among the centralized MPC and classical voltage and speed regulators.

5.2 Controller design

In order to implement an MPC for generation units, a third order model depicted by equation (5.1) was selected.

$$\begin{aligned} \dot{\delta} &= \omega_0(\omega - 1) \\ \dot{\omega} &= \frac{1}{M}[-D(\omega - 1) + P_m - P_e] \\ \dot{E}'_q &= \frac{1}{\tau'_{d0}}[-E_q - (x_d - x'_d)I_d + E_f] \end{aligned} \tag{5.1}$$

where δ, ω, E'_q denote the angle, speed and electromotive force in the quadrature axis of the generator, respectively; $\omega_0, M, D, \tau'_{d0}, x_d, x'_d$ represent the nominal speed, the inertia, the damping, the transient time constant, the reactance and the transient reactance, of the generation unit, respectively; P_m, P_e, I_d, E_f are the mechanical and electric power, the field current in the direct axis and the field voltage, respectively. All values of the variables listed before are in $p.u$.

This model was used because it is a reduced order model that describes the speed and voltage behavior in terms of measurable and estimable variables, and its parameters can be provided by the machines manufacturers or can be knew from machine tests.

Additional to the load angle, the speed and the voltage dynamics, the prediction model must include the dynamics of voltage and speed regulators. Then it is assumed that these controllers have a first order system behavior given by

$$\begin{aligned} \dot{E}_f &= \frac{1}{\tau_e}[-E_f + k_e(V_{ref} - E'_q)] \\ \dot{P}_m &= \frac{1}{\tau_g}[-P_m + k_g(\omega_{ref} - \omega)] \end{aligned} \tag{5.2}$$

where E_f and P_m are the field voltage and the mechanical power respectively; τ_e, τ_g are the time constants of the field voltage regulator and the speed regulator respectively; k_e, k_g are the gains of the field voltage and speed controllers respectively; V_{ref}, ω_{ref} denote the voltage and speed reference values. Thus the model used to predict the behavior of each machine becomes

$$\begin{aligned} \dot{\delta} &= \omega_0(\omega - 1) \\ \dot{\omega} &= \frac{1}{M}[-D(\omega - 1) + P_m - P_e] \\ \dot{E}'_q &= \frac{1}{\tau'_{d0}}[-E_q - (x_d - x'_d)I_d + E_f] \\ \dot{E}_f &= \frac{1}{\tau_e}[-E_f + k_e(V_{ref} - E'_q)] \\ \dot{P}_m &= \frac{1}{\tau_g}[-P_m + k_g(\omega_{ref} - \omega)] \end{aligned} \tag{5.3}$$

Lets define $x = [\delta, \omega, P_m, E'_q, E_f]^T$, $u = [V_{ref}, \omega_{ref}]^T$, $d = [P_e, I_d]^T$, and $y = [\omega, E'_q]^T$, then the model given by (5.3) can be written in matrix form as

$$\begin{aligned} \dot{x} &= Ax + Bu + Dd \\ y &= Cx \end{aligned} \tag{5.4}$$

where A, B, C, D are the matrices associated with the continuous time model (5.3), each matrix is defined as follows:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & -\frac{D}{M} & -\frac{1}{M} & 0 & 0 \\ 0 & -\frac{k_g}{\tau_g} & -\frac{1}{\tau_g} & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau'_{d0}} & \frac{1}{\tau'_{d0}} \\ 0 & 0 & 0 & -\frac{k_e}{\tau_e} & -\frac{1}{\tau_e} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \frac{k_g}{\tau_g} \\ 0 & 0 \\ \frac{k_e}{\tau_e} & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{M} \\ 0 \\ -(x_d - x'_d) \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The A matrix shown before has a block diagonal form, then each machine model can be decomposed into two independent subsystems, each one associated with mechanical and electric dynamics. To implement the MPC, all the mechanical and electric dynamics are grouped and discretized by the Tustin approximation, using two different sampling times (one for each model depicted before). Thus MPC for generation units become

$$\min_{u(t)} \sum_{k=1}^{Np} l_{1m}[x_m(t+k), y_{refm}(t+k)] + \sum_{k=1}^{Nc} l_{2m}[u_m(t+k)] \tag{5.5}$$

$$s.t. \quad : \quad x_m(t+k+1) = A_m x_m(t+k) + B_m u_m(t+k) \tag{5.6}$$

$$operational\ constraints \tag{5.7}$$

$$\min_{u(t)} \sum_{k=1}^{Np} l_{1e}[x_m(t+k), y_{refe}(t+k)] + \sum_{k=1}^{Nc} l_{2e}[u_e(t+k)] \tag{5.8}$$

$$s.t. \quad : \quad x_e(t+k+1) = A_e x_e(t+k) + B_e u_e(t+k) \tag{5.9}$$

$$operational\ constraints \tag{5.10}$$

where $l_{1m}[x_m(t+k), y_{mref}(t+k)] = (y_{mref}(t+k) - y_m(t+k+1))^T Q_m (y_{mref}(t+k) - y_m(t+k+1))$, $l_{1e}[x_e(t+k), y_{eref}(t+k)] = (y_{eref}(t+k) - y_e(t+k+1))^T Q_e (y_{eref}(t+k) - y_e(t+k+1))$, $l_{2m}[u_m(t+k)] = u_m(t+k)^T R_m u_m$, $l_{2e}[u_e(t+k)] = u_e(t+k)^T R_e u_e$, $x_m(k) = [\delta(k), \omega(k), P_m(k)]^T$, $x_e = [E'_q, E_f]^T$, being $y_m(t+k+1) = \omega(t+k+1)$, $y_e = E'_q(t+k+1)$, A_m, A_e, B_m, B_e the matrices associated with each submodel discretization and $Q_m, Q_e, R_m, R_e > 0$.

Each controller, presented before, works independently and has its own sampling time. Also the information exchange between controllers does not exists due to the block form of the A matrix of the prediction model.

To compute the initial conditions for the states, the steady-state equations was used. Thus

$$\begin{aligned} \omega_0 &= \frac{k_g \omega_{ref} - P_{med}}{D + k_g} \\ E'_{q0} &= -(x_d - x'_d) I_d + E_f \\ E_f &= k_e (V_{ref} - V_{med}) \\ P_m &= k_g (\omega_{ref} - \omega_0) \end{aligned} \tag{5.11}$$

where P_{med} and V_{med} are the measured values at generation buses of active power and voltage magnitude respectively. The direct axis current, I_d was estimated using the following equations

$$\begin{aligned} S &= \sqrt{P_{med}^2 + Q_{med}^2} \\ I_g &= \frac{S}{\sqrt{3} V_{med}} \\ I_d &= \frac{2}{3} I_g (\cos(\delta_0) + \cos(\delta_0 - \frac{2}{3}\pi) + \cos(\delta_0 + \frac{2}{3}\pi)) \end{aligned} \tag{5.12}$$

Since there is no equation available for the computation of the initial condition of the load angle, δ_0, we implemented an angle estimator.

Let define the phasor current as (5.13)

$$\bar{I} = (\frac{P_{med} + jQ_{med}}{\overline{V}})^* \tag{5.13}$$

and applying the Ohm law to the transmission system

$$\bar{I} = Y_{bus} \overline{V} + v \tag{5.14}$$

then knowing the active and reactive power demands at generation units, and the Y_{bus} matrix, an objective function can be formulated to minimize the error v. Thus the initial condition of the load angle can be computed as the solution of the optimization problem given by (5.15).

$$\begin{aligned} &\min_{\overline{V}} [(\bar{I} - Y\overline{V})^T Q (\bar{I} - Y\overline{V})] \\ s.t: &\bar{I} = (\frac{P_{med} + jQ_{med}}{\overline{V}})^* \\ &V_{med} = \|\overline{V}\| \end{aligned} \tag{5.15}$$

where $\bar{I}$ and $\overline{V}$ are the complex values associated with current and voltage phasors, and Y is the Y_{bus} matrix of the system.

5.3 Simulation results

The control scheme presented in the previous section was implemented in the Benchmark of Electric Power System (see Fig. 5.1).

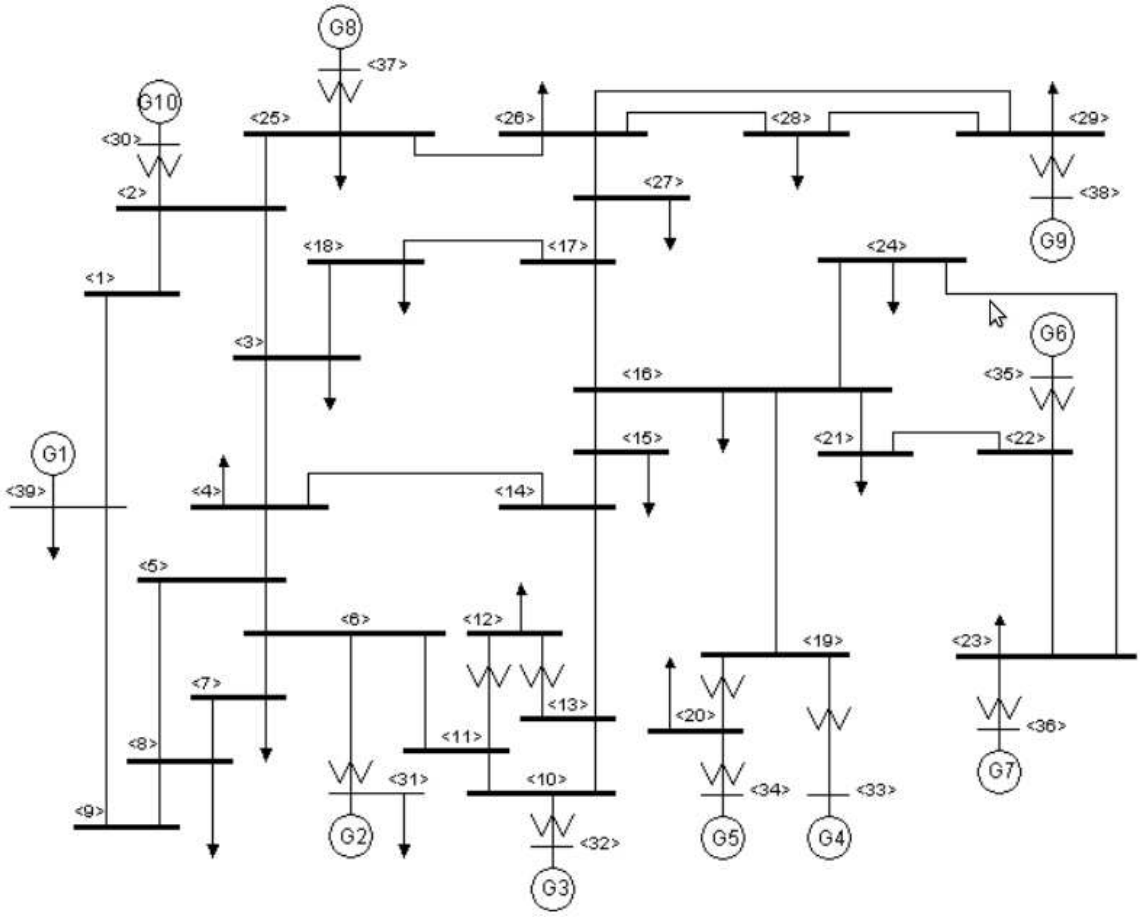


Figure 5.1: New England Electric Power System

The electric power system presented in Fig. 5.1 has ten generation units $G1,\ldots,G10$, thirty-nine transmission nodes or buses, and nineteen load centers. Each generation unit has its own speed and voltage regulator. Also it has its own power system stabilizer (PSS). The system was simulated using Matlab/Simulink software, integration routine ODE23stiff/trapezoidal. More details about the system of Fig. 5.1 can be found in the documentation sent by the Universidad Nacional de Colombia. In this documentation it is possible to see the used parameters to tune the controllers. Moreover as it is stated the performance of the controllers becomes acceptable, leading field voltages and mechanical powers of each generation unit to stable values.

After simulate the power system with the classic controllers, the multi-model MPC depicted in the previous sections was added in order to compute the optimal speed and voltage references. The sample times used for mechanical and electrical models was $0.1s$ and $0.001s$ respectively. The prediction and control horizons were 200 and 90 for the model predictive controller associated with the mechanical dynamics and 500 and 60 for the mode predictive controller associated with the electrical dynamics.

Figs. 5.2 to 5.4 show the field voltage, the mechanical power and the speed behavior after including the proposed model predictive controller. In this simulation at 20*s* a three phase fault and at 50*s* a line outage was introduced.

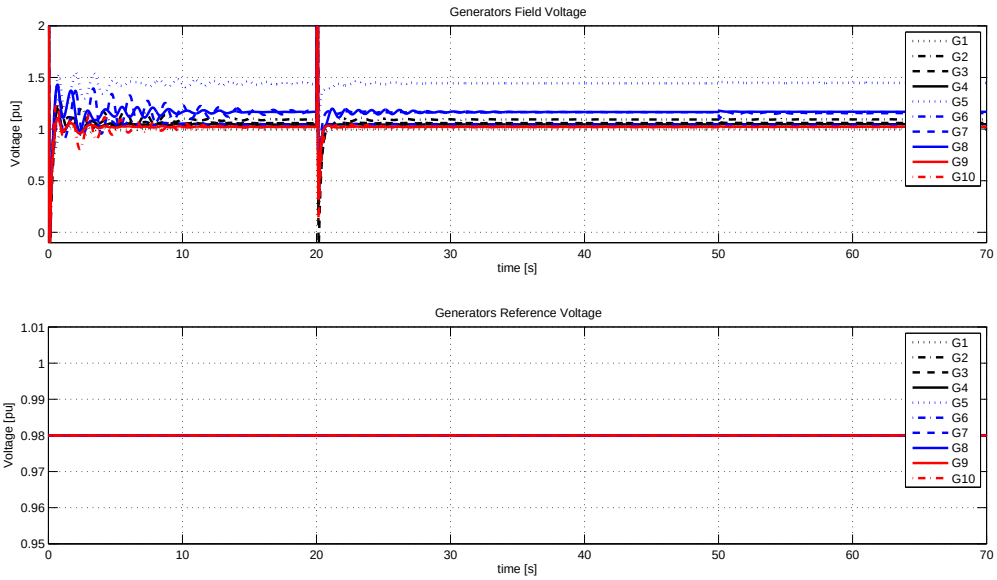


Figure 5.2: Field voltage of generation units with MPC

In Figs. 5.2 to 5.4 it is possible to see that the addition of an MPC reduce the start-up transient behavior of the machines. also with the introduction of the MPC, it is possible to obtain a higher damping of oscillations due to disturbances in the network, and achieve a faster system balance. This allows to improve the customers quality of service, because the rejection of disturbances is carry out using soft control actions. It is shown in Figs. 5.5 and 5.6.
Moreover MPC gives the possibility to take a better mechanical power control than the obtained using only the speed regulator. This fact allows to synchronize the machines of the system in a shorter time.

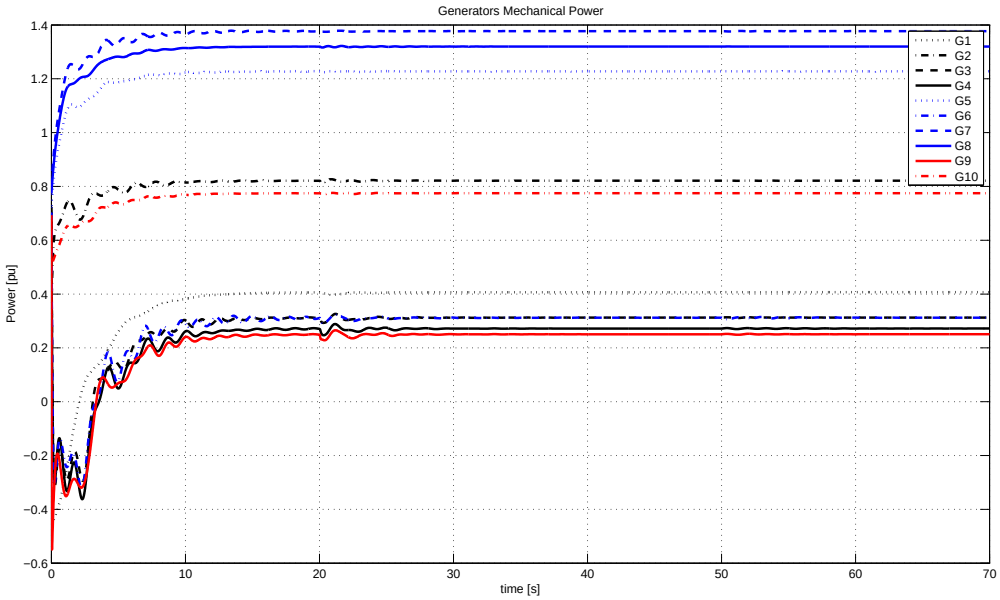


Figure 5.3: Mechanical power of generation units with MPC

5.4 About this implementation

From this implementation, it is possible to conclude that MPC is a promising control structure for the control of power systems. From the results, notice that the optimal calculation of the reference of the generation units allows to give and additional damping to the oscillations due to disturbances in the transmission network and during the start-up of the machines. This improves the quality service given to the customers. Also the proposed control scheme allows to be closer to real time optimal dispatch.

Moreover the system decomposition shown in this work presents a procedure to implement an MPC in real complex systems, as it was demonstrated in the study case and results presented before. However it is necessary to improve the performance of the control scheme proposed before, in order to carry out real-time applications in real large-scale systems.

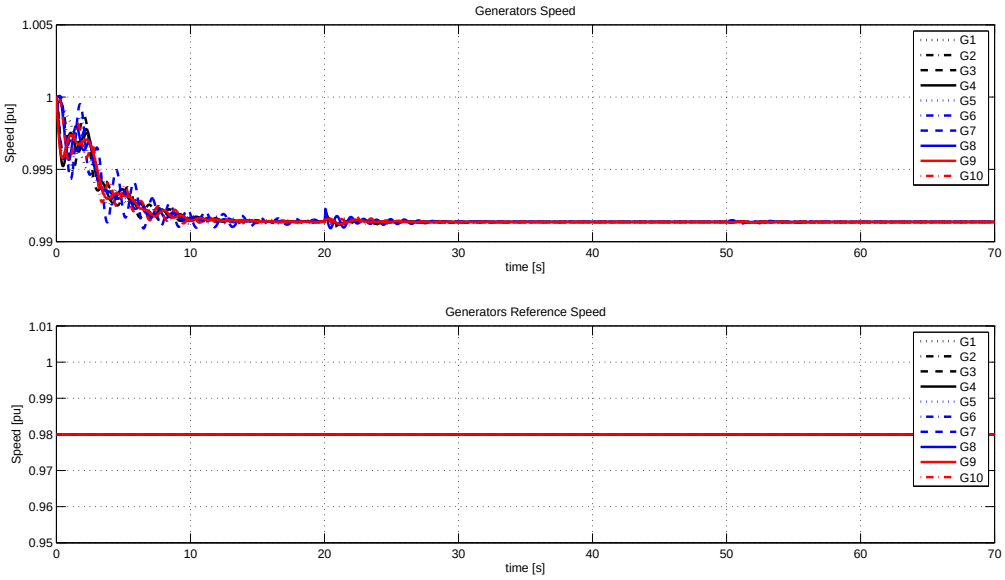


Figure 5.4: Speed of generation units with MPC

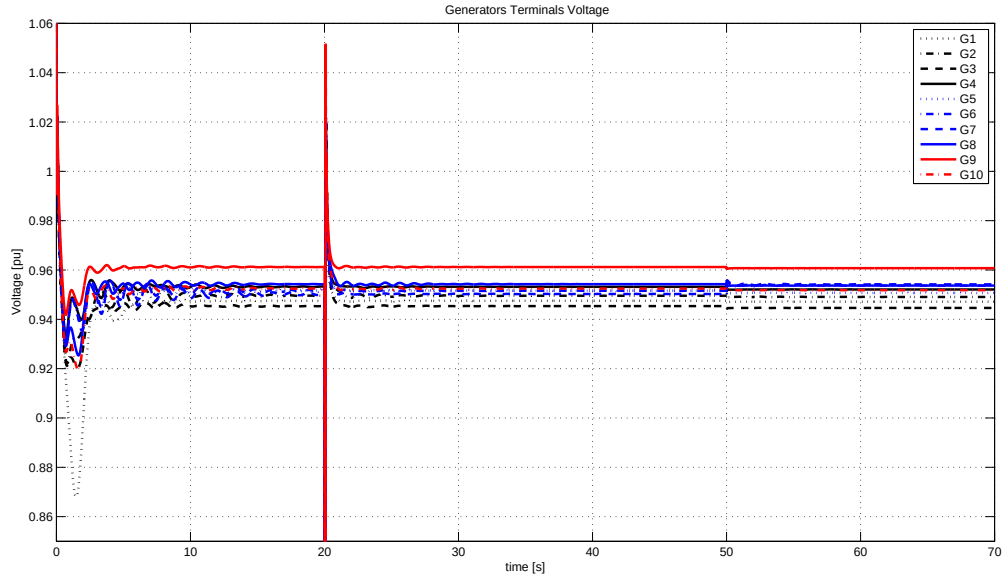


Figure 5.5: Generation buses voltage

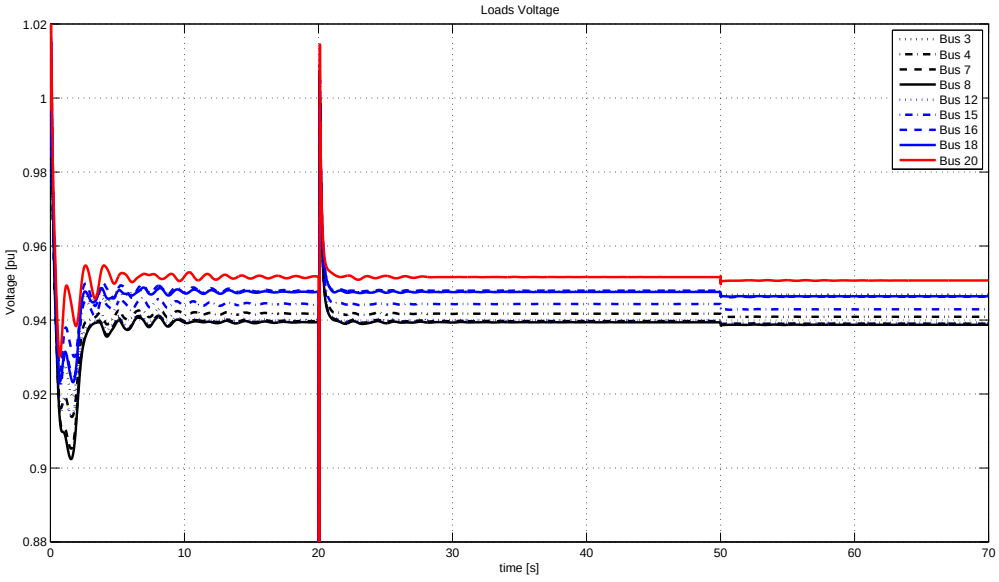


Figure 5.6: Load buses voltage

Bibliography

[1] Alriksson, P. and Rantzer, A. ”Model based information fusion in sensor networks”. In *Proceedings of the 17th IFAC World Congress*, pages 4150–4155, Seoul, Korea, July 2008.

[2] García, J. and Espinosa, J. ”A benchmark for distributed control and estimation: A generic model of the heat conduction and convection in a rod, a plate and a cube”. Internal Report No.5, HD-MPC of Large Scale Systems – V4, 2009.

[3] Hashemipour, H.R., Roy, S., and Laub, A.J. ”Decentralized structures for parallel Kalman filtering”. *IEEE Transactions on Automatic Control*, 33(1):88–94, January 1988.

[4] Hassan, M.F.,Salut, G., Singh, M.G., and Titli, A. ”A decentralized computational algorithm for the global Kalman filter”. *IEEE Transactions on Automatic Control*, 23(2):262–268, April 1978.

[5] Holman, J.P. *Heat Transfer*. McGraw-Hill Higher Education, 10th edition, 2008.

[6] Kalman, R.E. ”A new approach to linear filtering and prediction problems”. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[7] Khan, U.A. and Moura, J.M.F. ”Distributing the Kalman filter for large-scale systems”. *IEEE Transactions on Signal Processing*, 56(10):4919–4935, October 2008.

[8] Lienhard IV, J.H. and Lienhard V, J.H. *A Heat Transfer Textbook*. Phlogiston Press, 3 edition, 2008.

[9] Olfati-Saber, R. ”Distributed Kalman filtering for sensor networks”. In *Proceedings of the 46th IEEE Conference on Decision and Control 2007 (CDC ’07)*, pages 5492–5498, New Orleans, Louisiana, USA, December 2007.

[10] Rao, B. S. and Durrant-Whyte, H. F. ”Fully decentralised algorithm for multisensor Kalman filtering”. *IEE Proceedings D: Control Theory and Applications*, 138(5):413–420, September 1991.

[11] Speyer, J.L. ”Computation and transmission requirements for a decentralized linear-quadratic-gaussian control problem”. *IEEE Transactions on Automatic Control*, 24(2):266–269, April 1979.

[12] Johansson K.J. ”The Quadruple-Tank Process” *IEEE Trans. Cont. Sys. Techn.*, 8:456–465, 2000.

[13] Limon D., Alvarado I., Álamo T. and Camacho E.F. ”MPC for tracking of Piece-Wise Constant References for Constrained Linear Systems” *Automatica*, 44(9), 2382–2387, 2008.

[14] Gilbert, E.G. and Tan. K. ”Linear systems with state and control constraints: The theory and application of maximal output admissible sets” *IEEE Transations on Automatic Control*, 36: 1008–1020, 1991.

[15] Alvarado I. ”Model Predictive Control for Tracking of Constarined Linear Systems” *PhD Thesis*, Universidad deSevilla, 2007.

[16] Maestre, J.M., Muñoz de la Peña, D. and Camacho E.F. ”Distributed MPC based on a cooperative game” *Proceedings of the CDC*, 2009.

[17] Alanyani,M., and Saligrama, V. “Distributed Tracking in multi-hop Networks with Communication Delays and Packet Losses”. *Proc. of IEEE Workshop on Statistical Sig. Proc.*, Bordeaux, France, pp.1190-1195, Jul. 2005.

[18] Alighanbari, M., and How, J.P. “An Unbiased Kalman Consensus Algorithm”. *Proc. of the ACC*, Minneapolis, USA, June 2006.

[19] Berg,T., and Durrant-Whyte, H. “Model Distribution in decentralized multi-sensor data fusion”. Tech, Report., University of Oxford, 1990.

[20] Cattivelli, F.D., Lopez, C.G., and Sayed, A.H. “Diffusion Strategies for Distributed Kalman Filtering: Folrmulation and Performance Analysis”. *Proc. of the IAPR Workshop on Cognitive Information Processing*, Thessalonik, Greece, June 2008.

[21] Feltes, J.W.; Lima, L.T.G.; Senthil, J. “IEEE 421.5 standard - perspective of a user and supplier of simulation software”, Power Engineering Society General Meeting, IEEE, 12-16 June, Page(s): 995 - 998, Vol. 1, 2005.

[22] Garcia, J., and Espinosa, J. “A Benchmark for Distributed Estimation and Control: a Generic Model for the Heat Conduction and Convection in a Rod, a Plate, and a Cube”. *Internal Report 5*. Universidad Nacional de Colombia.

[23] Hashemipour,H.R., Roy, S., and Laub, A.J “Decentralized Structures for Parallel Kalman Filtering”. *IEEE Trans. on Automatic Control*,vol 33, no. 1, pp. 88-94. Jan 1988.

[24] Khan, U.A., and Moura, J.M.F. “Model Distribution for Distributed Kalman Filters: a Graph Theoretic Approach”. *In Conf on Signals, Systems and Computers*. Vol 4, Issue 7, pp:611-615.

[25] Khan, U.A., and Moura, J.M.F. “Distributed Kalman Filters in Sensor Networks: Bipartite Fusion Graphs”. *In Proc. of the 14th Workshop on Statistical Signal Processing, SSP’07*, Madison, WI, Aug. 2007, pp. 700-704.

[26] Khan, U.A., and Moura, J.M.F. ”Distributing the Kalman Filter for Large-Scale Systems”. *IEEE Trans. on Signal Processing*, vol 56, no. 10, pp.4919-4935, 2008.

[27] Kundur, P., “Power System Stability and Control”McGraw-Hill, New York, 1994.

[28] Mergangoz, M., and Doyle III, F.J. “Distributed Model Predictive Control of an Experimental four Tank System”. *Journal of Process Control*,17, pp.297-308, 2007.

[29] Mutambara, A.G.O. *Decentralized Estimation and Control for Multisensor Systems*. Boca Raton, FL:CRC, 1998.

[30] Olfati-Saber,R. “Distributed Kalman Filters with Embedded Consensus Filters”. *Proc. of 44th Conf. on Decision and Control*,Seville, Spain, pp.8179-8184, Dec. 2005.

[31] Oruç, S., Sijs, J., and van de Bosch, P.P.J. “Optimal Decentralized Kalman Filter”. *17th Mediterranean Conference on Control and Automation*, Thessaloniki, Grecce, June 2009.

[32] Pacific Gas and Electric Company, “Apendix H: Power System Stabilizers for Generation Entities”PGE Interconnnection Handbooks.

[33] Rao, B., and Durrant-Whyte, H.J. “Fully Decentralized Algorithm for Multisensor Kalman Filter”. *Proc. on IEEE Control Theory and Applications*, vol 138, pp.413-420, Sep. 1991.

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

[35] Speyer,J.L. “Computation and Transmission requirements for a decentralized linear-quadratic gaussian control problem”. *IEEE Trans. on Automatic Control*,vol 24, no. 2,pp. 266-269. Feb 1979.

[36] Sijs, J., Lazar, M., van de Bosch, P.P.J., and Papp, Z. “An Overview of non-centralized Kalman Filters”. *17th IEEE International Conference on Control Applications*. San Antonio, Texas, USA (2008), pp.739-744.

[37] Vadigepalli, R., and Doyle III, F.J. “A Distributed State Estimation and Control Algorithm for Plantwide Processes”. *IEEE Trans. on Control Systems Technology*, vol 11, no. 1, pp.119-127, 2003.

[38] Camponogara, E., Jia, D., Krogh, B. H., and Talukdar, S. “Distributed Model Predictive Control”. *IEEE Control Systems Magazine*, February, 2002.

[39] Camponogara, E., and Talukdar, S. “Distributed Model Predictive Control: Synchronous and Asynchronous Computation”. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, Vol. 37, No. 5, September, 2007.

[40] Di Palma, F., and Magni, L.“A Multi-Model Structure for Model Predictive Control”. *Annual Reviews in Control*. vol 28, pp. 47-52, 2004.

[41] Doan, D., Keviczky, T., Necoara, I., and Diehl, M. “A Jacobi algorithm for distributed model predictive control of dynamically coupled systems”, arXiv:0809.3647v1, Submitted on 22 Sep 2008.

[42] Du, X., Xi, Y., and Li, S. “Distributed Model Predictive Control for Large-scale Systems”. *Proceedings of the American Control Conference*, Arlington, VA, June 2001.

[43] Dunbar, W. B., and Desa, S, “Distributed Model Predictive Control for Dynamic Supply Chain Management”. *Int. Workshop on Assessment and Future Directions of NMPC*, Freudenstadt-Lauterbad, Germany, August 26-30, 2005.

[44] Li, S., Zhang, Y., and Zhu, Q. “Nash-Optimization Enhanced Distributed Model Predictive Control Applied to the Shell Benchmark Problem”. *Information Sciences*. vol 170, issue 2-4,pp. 329-349, Feb 2005.

[45] Necoara, I., Doan, D., and Suykens, J. A. K, "Application of the proximal center decomposition method to distributed model predictive control". Internal Report 08-32, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2008. Published in the IEEE Conference on Decision and Control (CDC 2008) Cancun.

[46] Negenborn, R. R. "Multi-Agent Model Predictive Control with Applications to Power Networks". Dutch Institute of Systems and Control, NGInfra PhD thesis series on infrastructures Nr.13, Netherlands, 2007.

[47] Rawlings, J. B., Venkat, A. N., and Wright, S. J. "Distributed model predictive control of large-scale systems". *2005 NMPC Workshop, Assessment and Future Directions of Nonlinear Model Predictive Control*. Zollernblick, Freudenstadt-Lauterbad, Germany, August, 2005.

[48] Vadigepalli, R., and Doyle III, F.J. "Structural Analysis of Large-scale Systems for Distributed State Estimation and Control Applications". *Cont. Eng. Practice*, 11, pp.895-905, 2003.

[49] Valencia, F., Calderon, C., Garcia, J., Scattolini, R., and Espinosa, J. "Documentation for Benchmark Cases -Part II". *Deliverable 6.3.1*, HD-MPC project.

[50] Venkat, A. N., Hiskens, I. A., Rawlings, J. B., and Wright,S. J. "Distributed MPC Strategies with Application to Power System Automatic Generation Control". *IEEE Trans. on Control Systems Technology*. vol 16 , iss. 6 , pp. 1192-1206, 2008.

[51] Wang, X., Jia, D., Lu, C., and Koutsoukos, X, "Decentralized Utilization Control in Distributed Real-Time Systems". *Proceedings of the 26th IEEE International Real-Time Systems Symposium (RTSS'05)*. Dec 2005.