Delft University of Technology

Delft Center for Systems and Control

Technical report 11-025

A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark*

I. Alvarado, D. Limon, D. Muñoz de la Peña, J.M. Maestre, M.A. Ridao, H. Scheu, W. Marquardt, R.R. Negenborn, B. De Schutter, F. Valencia, and J. Espinosa

If you want to cite this report, please use the following reference instead:
I. Alvarado, D. Limon, D. Muñoz de la Peña, J.M. Maestre, M.A. Ridao, H. Scheu, W. Marquardt, R.R. Negenborn, B. De Schutter, F. Valencia, and J. Espinosa, "A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark," *Journal of Process Control*, vol. 21, no. 5, pp. 800–815, June 2011. doi:10.1016/j.jprocont.2011.03.003

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark☆

I. Alvarado[a], D. Limon[a], D. Muñoz de la Peña[a], J.M. Maestre[a], M.A. Ridao[a], H. Scheu[b], W. Marquardt[b], R.R. Negenborn[c], B. De Schutter[c], F. Valencia[d], J. Espinosa[d]

[a]*Dept. Ingeniería de Sistemas y Automática, Universidad de Sevilla. Sevilla, Spain.*
[b]*AVT Process Systems Engineering, RWTH Aachen University. Aachen, Germany.*
[c]*Delft Center for Systems and Control, Delft University of Technology. Delft, The Netherlands.*
[d]*Universidad Nacional de Colombia. Medellín, Colombia.*

Abstract

Recently, there has been a renewed interest in the development of distributed model predictive control (MPC) techniques capable of inheriting the properties of centralized predictive controllers, such as constraint satisfaction, optimal control, closed-loop stability, etc. The objective of this paper is to design and implement in a four-tank process several distributed control algorithms that are under investigation in the research groups of the authors within the European project HD-MPC. The tested controllers are centralized and decentralized model predictive controllers schemes for tracking and several distributed MPC schemes based on *(i)* cooperative game theory, *(ii)* sensivity-based coordination mechanisms, *(iii)* bargaining game theory, and *(iv)* serial decomposition of the centralized problem. In order to analyze the controllers, a control test is proposed and a number of performance indices are defined. The experimental results of the benchmark provide an overview of the performance and the properties of several state-of-the-art distributed predictive controllers.

Keywords: Distributed control, Predictive control, Optimal control, Benchmark examples, Control applications.

1. Introduction

Distributed model predictive control (DMPC) is an important control methodology in current control engineering for large-scale or networked systems, mainly to overcome computational (and possibly communication) limitations of centralized approaches. These distributed algorithms are based on a wide range of techniques. Systematic studies of these techniques require the analysis of benchmark problems to assess the performance of the different algorithms and to characterize their properties.

The use of benchmarks is useful for evaluating the capabilities of different approaches to control systems for real problems. Benchmarks allow one to test, evaluate, and compare different control solutions on real or simulated plants. The research and

☆This research has been supported by the European 7th framework STREP project "Hierarchical and distributed model predictive control (HD-MPC)", contract number INFSO-ICT-223854.

the industry community benefit from these activities since the design of a good simulation test-bed is often time and resource consuming. However, many simulation test-beds are often subject to harsh criticism as they either cover only a narrow part of the problem or they are purposely designed to get biased rather than objective performance results. Suitable benchmark problems would effectively overcome these problems by (a) allowing an objective evaluation of alternative control technologies, by (b) reducing resources and time spent on developing validation models, by (c) giving researchers the possibility to evaluate their proposals on a variety of cases, and by (d) opening up a forum to compare the performance of various solutions and to discuss the quality of the results.

The objective of this paper is to design and implement several distributed control algorithms, to analyze the algorithms, and to compare them on a common real benchmark process, namely a four-tank plant located in the Department of Ingeniería de Sistemas y Automática of the University of Seville. This plant is based on the quadruple-tank process [8]. This process has proven to be a very interesting system for control education and research despite its simplicity, since the system is a highly coupled system that can exhibit transmission zero dynamics, the dynamics are nonlinear and the states and inputs are subject to hard constraints. Furthermore, the four-tank plant is implemented using industrial instrumentation and is safe to use. The quadruple-tank process has been used to illustrate various control strategies including internal model control [6], dynamic matrix control [5], multi-variable robust control [24] and distributed MPC [12]. In addition, it has also been utilized as an educational tool to teach advanced multi-variable control techniques.

For the proposed benchmark, four-tank plant has been divided into two subsystems coupled through the inputs. The objective of the distributed controllers is to minimize a quadratic tracking performance index of the whole plant, which adds objective coupling between the controllers. In order to evaluate the controllers to be tested, a collection of indices will be proposed. These mainly measure two aspects: the closed-loop performance and the communication requirement of the controllers. Timing and communication delay issues are negligible in this benchmark due to the implementation of the controller. The controllers tested are a centralized MPC scheme for tracking, a decentralized MPC scheme for tracking [9], a distributed MPC scheme based on a cooperative game [11], a sensitivity-driven distributed MPC scheme [21, 22], a feasible-cooperation distributed MPC scheme based on bargaining game theory concepts, and a serial DMPC scheme [16, 17]. The distributed MPC algorithms have been developed by partners of the project HD-MPC [1], which aims at the development of new and efficient methods and algorithms for distributed and hierarchical model-based predictive control of large-scale, complex, networked systems.

This paper is organized as follows. In Section 2, a description and a dynamic model of the four-tank plant are provided and the benchmark control problem is presented. In Section 3 the controllers applied to the four-tank plant are briefly introduced and the experimental results are shown. The results of the benchmark are compared and discussed in Section 4. Finally, Section 5 contains conclusions.

Notation

Throughout this paper, $(z_1, z_2, \cdots, z_N)$ stands for $[z_1^T, z_2^T, \cdots, z_N^T]^T$, that is, the column vector resulting from stacking the column vectors z_1, z_2, etc. As usual, for a vector

[1]For more information, see the HD-MPC web-site `http://www.ict-hd-mpc.eu/`.

$z \in \mathbb{R}^n$, $||z||_M$ denotes the weighted Euclidean norm, i.e. $||z||_M = \sqrt{z^T M z}$. I stands for the unitary matrix which dimension is derived from the context. $M^{(i)}$ denotes the i-th column of the matrix M

2. Description of the benchmark

In this section, the control benchmark with which the designed distributed predictive controllers will be tested is presented. This benchmark is executed in the four-tank plant which is inspired by the educational quadruple-tank process proposed by Johansson in [8]. Johansson's process has also been used as a suitable test-bed for distributed controllers [6, 12] and it has been proposed as one of the case studies in the European project HD-MPC [2]. This is due to the following interesting properties: *(i)* the dynamics of the plant exhibit large coupling between the subsystems and the degree of coupling can be manually adjusted, *(ii)* the dynamics of the plant are nonlinear, *(iii)* the state can be measured, *(iv)* the states and inputs of the plant are subject to hard constraints, and *(v)* the plant can be safely operated. A detailed description of the plant, the Simulink simulation model used in the design of the controllers as well as the experimental and simulation results of the controllers tested in this benchmark are available at the HD-MPC website[1].

2.1. The four-tank plant

The four-tank plant is a laboratory plant that has been designed to test process control techniques using industrial instrumentation and control systems. The plant consists of a hydraulic process of four interconnected tanks inspired by the educational quadruple-tank process (see Figure 2(a)) proposed by Johansson in [8]. A photograph of the four-tank plant is shown in Figure 1 and a schematic plot of the plant is given in Figure 2(b). As it can be noticed, the four-tank plant retains the structure of Johansson's process, see Figure 2(a), but has been modified to enable different configurations and interconnections of the tanks.

The inlet flow of each tank is measured by an electro-magnetic flow-meter (Siemens Sitrans FM Flow sensor 711/S and transmitters Intermag/transmag) and regulated by a pneumatic valve (Siemens VC 101 with a positioner Sipart PS2 PA). This allows the plant to emulate the three-way valve of Johansson's quadruple-tank process by providing suitable set-points to the flow controllers. The level of each tank is measured by means of a pressure sensor (Siemens Sitrans P 7MF4020 and 7MF4032). All the measurements and commands are 4-20 mA current signals transmitted from/to a PLC Siemens S7-200. In order to achieve a safe operation of the plant and to prevent the overflow of tanks, each tank has a high-level switching sensor used as an alarm to switch off the pumps.

As in the quadruple tank process shown in Figure 2(a), the four tanks of the real plant are filled from a storage tank located at the bottom of the plant. The tanks at the top (tanks 3 and 4) discharge into the corresponding tank at the bottom (tanks 1 and 2, respectively). The three-way valves are emulated by a proper calculation of the set-points of the flow control loops according to the considered ratio of the three-way valve. Thus, the inlet flows of the three-way valves q_a and q_b in Figure 2(a) can be also considered to be the manipulated variables of the real plant.

Some of the parameters of the plant, such as the cross section of the outlet hole a_i and the ratio γ of each three-way valve, can be manually adjusted by the user. Hence, the dynamics of the plant can be tuned by the user. Furthermore, the inlet flows as

Figure 1: The four-tank plant.

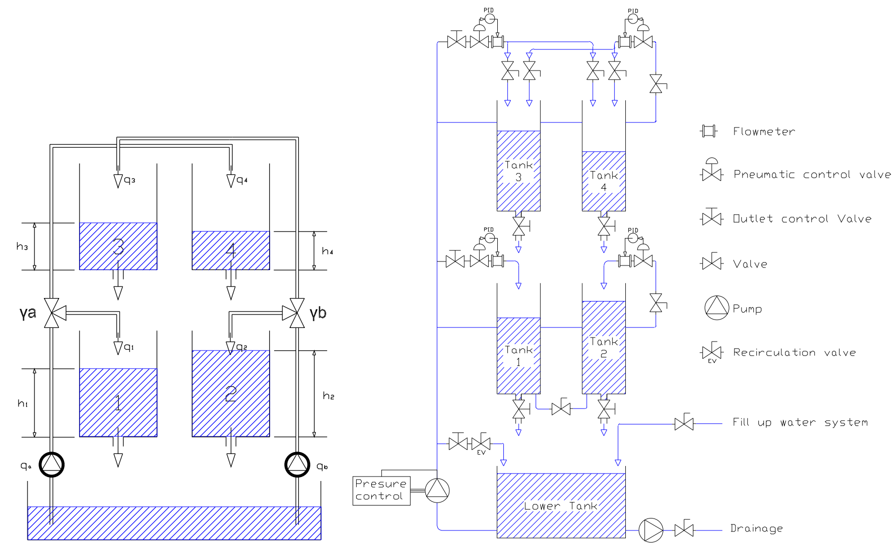
well as the levels of the tanks are physically constrained. Table 1 shows the values of the adjustable parameters, the physical limits of the levels and flows, and the operating point of the plant chosen for this benchmark.

The sampling of each sensor as well as the command of each manipulated variable is carried out by the PLC. This device stores the data and facilitates the implementation of low-level (e.g. PID) controllers, sequential controllers, and plant supervisors. All the data are continuously available through an OPC server installed on a remote PC connected to the PLC (via RS-232). The controllers to be tested are implemented and executed in Matlab/Simulink connected to the OPC Server using the OPC protocol. The total time that the transmission of the signals takes is negligible with respect to the sampling time (in this benchmark chosen as 5 seconds). Note that for this reason, timing and coordination issues are not relevant in this benchmark.

2.2. *Simulation and prediction model of the four-tank plant*

In order to design the controllers to be tested, a simulation model has been developed. This model is based on the simplified model of the quadruple-tank process proposed in [8]. This model is given by the following differential equations:

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



(a) Johansson's quadruple-tank process diagram.

(b) The real plant diagram.

Figure 2: The four-tank process diagram.

where h_i, and a_i with $i \in \{1, 2, 3, 4\}$ refer to the water level and the discharge constant of tank i, respectively, S is the cross section of the tanks, q_j and γ_j with $j \in \{a, b\}$ denote the flow and the ratio of the three-way valve of pump j, respectively, and g is the gravitational acceleration. The discharge constant a_i has been experimentally estimated and the ratios of three-way valves γ_a and γ_b are defined by the user. These parameters can be found in Table 1.

Notice that this model exhibits mismatches with the real behavior of the plant, since this model does not take into account the evolution of the real inlet flows of each tank (controlled by the control valves to emulate the three-way valves), the turbulence in the tanks or the variation of the level of the tank due to the inlet water flow. Nevertheless, these equations provide a satisfactory model of the four-tank process whenever the levels of the tanks are over 0.2 m. When the levels of the tanks are below 0.2 m, eddy effects in the discharges of the tanks render the model inaccurate.

For the predictive controllers to be tested in this benchmark, a linear prediction model will be derived based on the simulation model. This linear model is obtained by linearizing the simulation model at an operating point given by the equilibrium levels and flows as shown in Table 1. Defining the deviation variables

$$x_i = h_i - h_i^0, \quad i \in \{1, 2, 3, 4\}, \tag{2}$$
$$u_1 = q_a - q_a^0, \tag{3}$$
$$u_2 = q_b - q_b^0, \tag{4}$$

we obtain the following continuous-time linear model:

$$\begin{aligned} \frac{dx}{dt} &= A_c x + B_c u, \\ y &= C_c x, \end{aligned} \tag{5}$$

	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	Maximum flow of q_a
q_{bmax}	4	m^3/h	Maximum flow of q_b
q_{min}	0	m^3/h	Minimum flow of q_a and q_b
a_1	1.31e-4	m^2	Discharge constant of tank 1
a_2	1.51e-4	m^2	Discharge constant of tank 2
a_3	9.27e-5	m^2	Discharge constant of tank 3
a_4	8.82e-5	m^2	Discharge constant of tank 4
S	0.06	m^2	Cross-section of the tanks
γ_a	0.3		Parameter of the 3-way valve
γ_b	0.4		Parameter of the 3-way valve
h_1^0	0.65	m	Linearization level of tank 1
h_2^0	0.66	m	Linearization level of tank 2
h_3^0	0.65	m	Linearization level of tank 3
h_4^0	0.66	m	Linearization level of tank 4
q_a^0	1.63	m^3/h	Linearization flow of q_a
q_b^0	2.00	m^3/h	Linearization flow of q_b

Table 1: Parameters of the plant

where $x = (x_1, x_2, x_3, x_4)$, $u = (u_1, u_2)$, $y = (x_1, x_2)$,

$$A_c = \begin{bmatrix} \frac{-1}{\tau_1} & 0 & \frac{1}{\tau_3} & 0 \\ 0 & \frac{-1}{\tau_2} & 0 & \frac{1}{\tau_4} \\ 0 & 0 & \frac{-1}{\tau_3} & 0 \\ 0 & 0 & 0 & \frac{-1}{\tau_4} \end{bmatrix},$$

$$B_c = \begin{bmatrix} \frac{\gamma_a}{S} & 0 \\ 0 & \frac{\gamma_b}{S} \\ 0 & \frac{(1-\gamma_b)}{S} \\ \frac{(1-\gamma_a)}{S} & 0 \end{bmatrix},$$

$$C_c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

where $\tau_i = \frac{S}{a_i}\sqrt{\frac{2h_i^0}{g}} \geq 0$, with $i \in \{1, 2, 3, 4\}$, is the time constant of tank i. For the chosen parameters the linear system shows four real stable poles and two non-minimum phase zeros. Based on this model, the discrete-time model

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \qquad (6)$$

has been obtained using the Tustin method [3] with a sampling time of 5 seconds.

The linear model (6) is used to design a centralized predictive controller. In order to design the different distributed MPC controllers, the different subsystems of the plant to be controlled must be defined first. This is carried out in the following section.

2.3. Proposed partition of the plant and subsystems definition

The main objective of the benchmark is to study and evaluate distributed predictive controllers in a common framework, allowing a comparison among themselves and with a centralized as well as a decentralized controller scheme.

In this paper, a partition of the plant into two coupled subsystems as proposed in [2, 12] is considered. Subsystem 1 consists of tanks 1 and 3 while subsystem 2 consists of tanks 2 and 4. Then the state and output of each subsystem is defined as follows:

$$\begin{aligned} x_{s1} &= (x_1, x_3) \\ y_{s1} &= x_1 \\ x_{s2} &= (x_2, x_4) \\ y_{s2} &= x_2 \end{aligned}$$

The continuous-time models of subsystems 1 and 2 are given by

$$\begin{aligned} \frac{dx_{s1}}{dt} &= A_{c1}x_{s1} + B_{c1}u, \\ y_{s1} &= C_{c1}x_{s1}, \end{aligned} \quad (7)$$

and

$$\begin{aligned} \frac{dx_{s2}}{dt} &= A_{c2}x_{s2} + B_{c2}u, \\ y_{s2} &= C_{c2}x_{s1}, \end{aligned} \quad (8)$$

respectively. The matrices A_{c1}, B_{c1}, C_{c1}, A_{c2}, B_{c2}, and C_{c2} are easily derived from (5). The linear prediction model for the distributed predictive controllers tested in this benchmark are the discrete-time model of each subsystem for a sampling time of 5 seconds derived from (7) and (8) by means of the Tustin method. These discrete-time models will be denoted as

$$\begin{aligned} x_{s1}(k+1) &= A_1x_{s1}(k) + B_1u(k), \\ y_{s1}(k) &= C_1x_{s1}(k), \end{aligned} \quad (9)$$

and

$$\begin{aligned} x_{s2}(k+1) &= A_2x_{s2}(k) + B_2u(k), \\ y_{s2}(k) &= C_2x_{s2}(k). \end{aligned} \quad (10)$$

Notice that the subsystems of this partition are coupled through the inputs, but not through the states. This class of coupling is common in the process industry and has been widely studied in the design of distributed predictive controllers [19, 23, 25]. Furthermore, as proved in [23, Appendix B], every system can be split into a collection of subsystems only coupled through the inputs, and hence, the derived results of this benchmark are relevant despite the absence of direct coupling through the states.

For the design of decentralized and distributed MPC schemes, it is interesting to analyze the correlation between manipulable variables and controlled variables. This allows one to choose the manipulable variable to be used to control every controlled variable of the corresponding subsystem. This analysis has been done by means of the relative gain array (RGA) method [4]. The RGA matrix calculated for the linearized model (5) results in

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

From these matrix it is inferred that the main interaction are given for the pairing y_{s1}-u_2 in subsystem 1, and for the pairing y_{s2} - u_1 in subsystem 2. Denoting by v_{si} the coupling signal of subsystem i, namely, $v_{s1} = u_1$ and $v_{s2} = u_2$, the model of each subsystem can be rewritten as

$$\begin{aligned} \frac{dx_{s1}}{dt} &= A_{c1}x_{s1} + B_{c_1}^{(2)}u_{s1} + B_{c_1}^{(1)}v_{s1} \\ y_{s1} &= C_{c1}x_{s1}, \end{aligned} \tag{11}$$

and

$$\begin{aligned} \frac{dx_{s2}}{dt} &= A_{c2}x_{s2} + B_{c_2}^{(1)}u_{s2} + B_{c_2}^{(2)}v_{s2}, \\ y_{s2} &= C_{c2}x_{s1}, \end{aligned} \tag{12}$$

This model has been discretized by means of the Tustin method with a sampling time resulting the following model:

$$\begin{aligned} x_{s1}(k+1) &= A_1x_{s1}(k) + B_1^{(2)}u_{s1}(k) + B_1^{(1)}v_{s1}(k), \\ y_{s1}(k) &= C_1x_{s1}(k), \end{aligned} \tag{13}$$

and

$$\begin{aligned} x_{s2}(k+1) &= A_2x_{s2}(k) + B_2^{(1)}u_{s2}(k) + B_2^{(2)}v_{s2}(k), \\ y_{s2}(k) &= C_2x_{s2}(k). \end{aligned} \tag{14}$$

The decentralized model of the plant is derived from (13) and (14) making the coupling signals v_{s1} and v_{s2} equal to 0, that is,

$$\begin{aligned} x_{s1}(k+1) &= A_1x_{s1}(k) + B_1^{(2)}u_{s1}(k), \\ y_{s1}(k) &= C_1x_{s1}(k), \end{aligned} \tag{15}$$

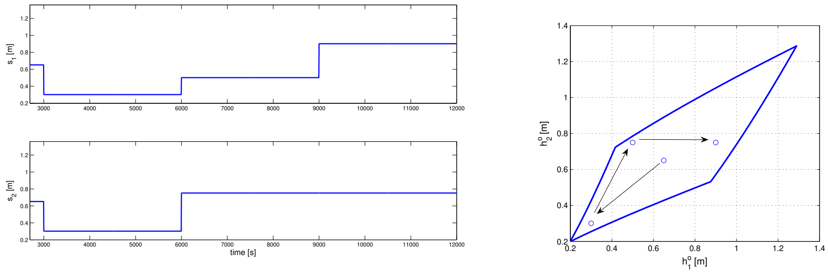
and

$$\begin{aligned} x_{s2}(k+1) &= A_2x_{s2}(k) + B_2^{(1)}u_{s2}(k), \\ y_{s2}(k) &= C_2x_{s2}(k). \end{aligned} \tag{16}$$

2.4. Control problem

To compare centralized, decentralized, and distributed predictive controllers under similar operation conditions a tracking experiment is defined where a set of reference changes in the levels of tanks 1 and 2, h_1 and h_2, has to be followed by manipulating the inlet flows q_a and q_b based on the measured levels of the four tanks:

- Initially, the set-points are set to $s_1 = s_2 = 0.65$ m. These set-points are aimed to steer the plant to the operating point and to guarantee identical initial conditions for each controller. Once the plant has reached the operating point the benchmark starts maintaining the operating point for 300 seconds.
- In the first step, the set-points are changed to $s_1 = s_2 = 0.3$ m. These values are kept for 3000 seconds.



(a) Set-point signals for the benchmark. (b) Set of admissible set-points.

Figure 3: Set-points for the real plant.

- Then, the set-points are changed to $s_1 = 0.5$ m and $s_2 = 0.75$ m. These values are kept for 3000 seconds.
- Finally, the set-points are changed to $s_1 = 0.9$ m and $s_2 = 0.75$ m. Again, these values are kept for 3000 seconds. To perform this change tanks 3 and 4 have to be emptied and filled respectively.

The set-point signals are shown in Figure 3(a). The control test duration is 3 hours and 20 minutes. It is important to remark that the set-points have been chosen in such a way that large changes in the different equilibrium points are involved. This is illustrated in Figure 3(b), where the region of reachable set-points is depicted together with the proposed set-points. Notice that some of the proposed set-points are close to the physical limits of the plant in terms of inputs or level of the tanks 3 and 4. This will allow us to check how the designed controllers behave when the system is close to the constraints.

The objective of the benchmark is to design distributed MPC controllers to optimize the performance index

$$J = \sum_{k=0}^{N_{test}-1} \Big((h_1(k) - s_1(k))^2 + (h_2(k) - s_2(k))^2 + 0.01(q_a(k) - q_a^s(k))^2 + 0.01(q_b(k) - q_b^s(k))^2 \Big),$$

where q_a^s and q_b^s are the steady manipulable variables of the plant for the set-points s_1 and s_2 calculated from steady conditions of the proposed model of the plant (1). The tested controllers have been designed using a sampling time of 5 seconds. The performance index measures the response of the plant once it has been steered to the operating point. Then J is calculated over the time interval $[2700, 12000]$ seconds, that is, for a total of $N_{test} = 1860$ samples.

Notice that the resulting distributed control problem to be solved in this benchmark exhibits two classes of couplings: *(i)* coupled dynamics, since the subsystems to be controlled are coupled through the inputs and *(ii)* coupled objectives, since the distributed controllers must optimize a global cost function. This fact makes that the four-tank benchmark is appropriate for comparing the closed-loop performance of distributed MPC controllers. However, other aspects such as the network communication and timing issues of the controllers are not relevant since the way the controllers have been implemented (see section 2.1) makes their effects negligible. Nevertheless, these aspects can be studied by evaluating the data communication requirements between controllers. Therefore, the evaluation and comparison between the different controllers will be performed according to the following collection of indices:

- Controller properties

  1. Modeling requirements: the class of models considered by each of the controllers, for instance linear/nonlinear, plant model or subsystem model, etc.
  2. Controller objectives: the properties that may be addressed by the tested controllers, for instance optimality, constraint satisfaction, stabilizing design, recursive feasibility, etc.
  3. Auxiliary software needed: optimization routines, simulation routines, etc.

- Performance evaluation

  1. Performance index J: a measure of the performance of the controlled plant.
  2. Performance index during the transient J_t: a measure of the performance during the transient to remove the effect of steady offset.
  3. Settling time: a measure of the speed of the controlled plant calculated by summing the settling times (defined as 95% achievement) after each step in the reference signal.
  4. Number of floating-point reals transmitted between the controllers per sampling period.
  5. Number of data packets transmitted during a sampling period.

3. Tested predictive controllers

For designing and tuning the DMPC controllers, all the participants have used the same Simulink model of the nonlinear continuous-time system (1) which has been identified using real data at the University of Seville. Each controller has been implemented as a Simulink block and integrated in a Simulink control model similar to the simulation model used in the design stage. This Simulink control model communicates with the PLC of the real plant via the OPC protocol to receive the measured level of the tanks and to send the calculated manipulated variables.

In the following subsections, the different control techniques are presented together with the results of the control test in the real plant.

3.1. Centralized MPC for tracking

A centralized predictive controller based on the linearized prediction discrete-time model (6) has been tested on the plant. Since the reference is changed throughout the benchmark, the MPC scheme for tracking proposed in [9] has been chosen. This controller is capable of steering the plant to any admissible set-point ensuring constraint satisfaction. We present next a description of this MPC controller.

The system to be controlled is subject to hard constraints on states and inputs,

$$(x(k), u(k)) \in Z \quad = \quad \{z \in \mathbb{R}^{n+m} : A_z z \leq b_z\}, \ \forall k \geq 0, \tag{17}$$

where the set Z is a suitable compact convex polyhedron containing the origin in its interior, n denotes the number of states of the plant, and m is the number of inputs.

This set consists of the limits of the levels and the flows of the plant and it can be easily calculated from the description of the plant in the previous section.

The MPC controller to be designed has to track a piece-wise constant sequence of set-points or references $s(k)$ while guaranteeing that the constraints are satisfied at all times.

The MPC scheme for tracking is based on the addition of the steady state and input (x_r, u_r) as decision variables (defined by the variable r), the use of a modified cost functions and an extended terminal constraint.The proposed cost function is

$$\begin{aligned} V_N(x,s,U,r) &= \sum_{i=0}^{N-1} \|y(i)-r\|_Q^2 + \|u(i)-u_r\|_R^2 + \|x(N)-x_r\|_P^2 \\ &\quad + \|r+h^0-s\|_T^2 \end{aligned}$$

where Q, R, P and T are matrices of appropriate dimensions, U is a sequence of N future control inputs, i.e. $U = \{u(0), \dots, u(N-1)\}$, (x_r, u_r) is the steady state and the input associated with r, $h^0 = (h_1^0, h_2^0)$, s is the set-point to be reached, and $y(i)$ is the predicted state of the system at time i given by $x(i+1) = Ax(i) + Bu(i)$, $y(i) = Cx(i)$, with $x(0) = x$. Based on this prediction model, there exists a matrix M_s such that $(x_r, u_r) = M_s r$. 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

$$\begin{aligned} V_N^*(x,s) \quad &= \quad \min_{U,r} V_N(x,s,U,r) \\ s.t. \quad & x(0) = x, \\ & x(i+1) = Ax(i) + Bu(i), \\ & (x(i), u(i)) \in Z, \qquad i = 0, \cdots, N-1\,, \\ & (x_r, u_r) = M_s r, \\ & (x(N), r) \in \Omega_{t,K}^a, \end{aligned}$$

where the set $\Omega_{t,K}^a \subseteq \mathbb{R}^{n+p}$ defining the terminal constraint is a polyhedral set and p is the number of outputs of the plant. Applying the receding horizon strategy, the control law is given by $K_N(x, s) = u^*(0; x, s)$. 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 and the controller is well-defined.

If the terminal ingredients K, P and $\Omega_{t,K}^a$ satisfy the assumptions provided in [9], the proposed controller stabilizes the plant, ensures constraint satisfaction, and guarantees that the controlled variable y converges to the set-point s if it is admissible. In case it is not admissible, the controller steers the plant to the closest admissible steady state (according to the offset cost function $\|r + h^0 - s\|_T^2$). Moreover, given that the evolution of the system remains in X_N, the system can be steered to any admissible set-point even in the case that the set-point changes along the time. Another interesting property is that thanks to the properties of the terminal region, the proposed controller provides a larger domain of attraction than a standard MPC scheme for regulation.

This controller ensures zero-offset in the nominal case. But, if there is model mismatch, a nonzero offset of the controller can result. In order to remove this offset, the offset cancellation technique proposed in [10] can be used. To this aim, the following disturbance estimator has been implemented

$$d(k+1) = \lambda d(k) + (1-\lambda)(x(k) - Ax(k-1) - Bu(k-1))$$

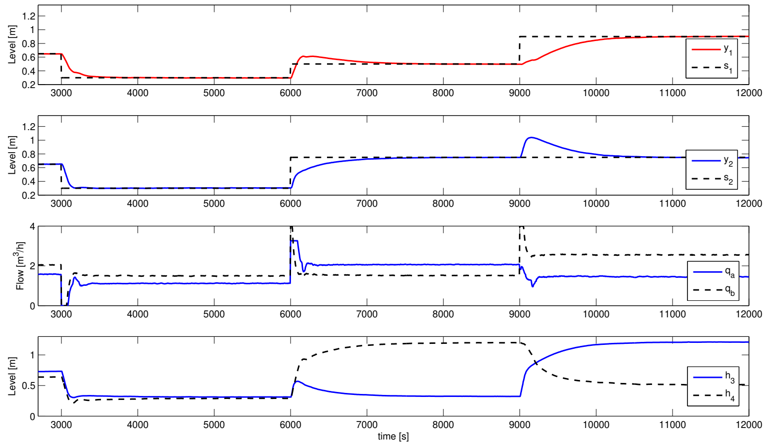


Figure 4: Evaluation of the control test in the real plant of the centralized MPC scheme for tracking.

where matrices A, B are defined in (6) and $\lambda = 0.99458$. Notice that this estimator converges to the steady value of the actual disturbance. Then, taking a modified set-point

$$\hat{s}(k) = s(k) + Hd(k)$$

where $H = C(I - (A + BK))^{-1}$, the effect of the disturbances is counteracted in steady state [10].

The weighting matrices of the controller designed for the four-tank plant are chosen to minimize the performance indices defined in the previous section, that is, $Q = I$ and $R = 0.01I$. The terminal control gain K is the corresponding LQR gain and the matrix P is derived from the Riccati equation, and the terminal set $\Omega^a_{t,K}$ is calculated as proposed in [9]. The offset cost weighting matrix has been chosen as $T = 100I$ and the chosen prediction horizon is $N = 5$.

This controller has been successfully tested on the real plant and the results are shown in Figure 4. The performance index for this test is $J = 28.4091$.

The MPC scheme for tracking may exhibit a possible optimality loss due to the addition of the artificial reference as a decision variable. In this benchmark, a standard MPC controller for regulation has also been applied to the four-tank plant. This controller is also based on the optimization problem $P_N(x, s)$ but adding a constraint to force y_s to be equal to s, hence reducing the degrees of freedom of the controller. Figure 5 shows the results obtained. The performance index for this test is $J = 25.4655$, hence better than performance of the MPC scheme for tracking. It is important to remark that this controller does not guarantee feasibility, stability, or constraint satisfaction when the set-point is changed, although for this particular case, these have been achieved.

3.2. *Decentralized MPC for tracking*

The second control technique tested has been a decentralized predictive controller based on the decentralized model of the system (15) and (16). The proposed cost

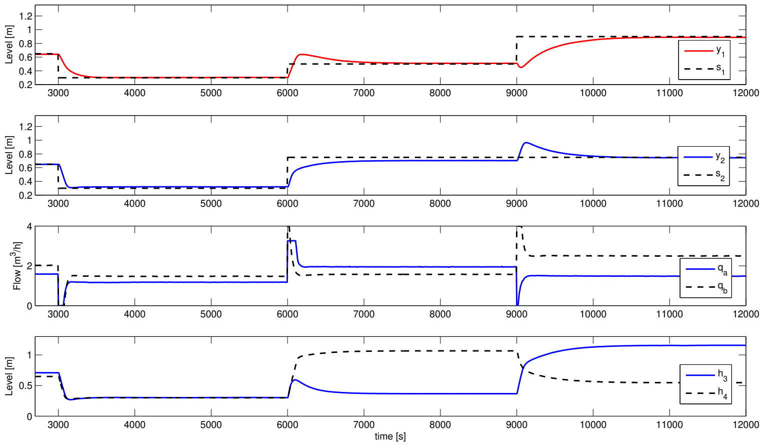


Figure 5: Evaluation of the control test in the real plant of the centralized MPC scheme for regulation.

function for the subsystem j is

$$\begin{aligned} V_{N,j}(x_{sj}, s_j, U_j, r_j) &= \sum_{i=0}^{N-1} \|y_{sj}(i) - r_j\|_{Q_j}^2 + \|u_{sj}(i) - u_{rj}\|_{R_j}^2 \\ &\quad + \|x_{sj}(N) - x_{rj}\|_{P_j}^2 + \|r_j + h_j^0 - s_j\|_{T_j}^2 \end{aligned}$$

where $U_j = \{u_{sj}(0), \ldots, u_{sj}(N-1)\}$, (x_{rj}, u_{rj}) is the steady state and the input associated with r_j for the j-th subsystem according to the models (15) and (16), and s_j is the setpoint to be reached. M_{sj} is the matrix such that $(x_{rj}, u_{rj}) = M_{sj} r_j$. Then the proposed MPC optimization problem for each subsystem j is $P_{N,j}(x_{sj}, s_j)$ is given by

$$\begin{aligned} V_{N,j}^*(x_{sj}, s_j) \quad &= \quad \min_{U_j, r_j} V_{N,j}(x_{sj}, s_j, U_j, r_j) \\ s.t. \quad & x_{sj}(0) = x_{sj}, \\ & x_{sj}(i+1) = A_j x_{sj}(i) + B_j^{(l)} u_{sj}(i), \quad l \neq j \\ & (x_{sj}(i), u_{sj}(i)) \in Z_j, \qquad i = 0, \cdots, N-1, \\ & (x_{rj}, u_{rj}) = M_{sj} r_j, \\ & (x_{sj}(N), r_j) \in \Omega_{t,K_j}^a. \end{aligned}$$

The set Z_j defines the constraints on the the states and the input of subsystem j. The weighting matrices are $Q_j = I$ and $R_j = 0.01I$. The terminal control gain of each subsystem j, K_j, is the corresponding LQR gain, the matrix P_j is derived from the Riccati equation and the terminal set Ω_{t,K_j}^a is calculated as proposed in [9]. The offset cost weighting matrix has been chosen as $T_j = 100$ and the chosen prediction horizon is $N = 5$. The results of the experiments can be seen in Figure 6; the performance index is $J = 39.5421$.

3.3. Distributed MPC based on a cooperative game

In this section we present the distributed MPC scheme based on a cooperative game approach presented in [11]. This control scheme considers a class of distributed linear

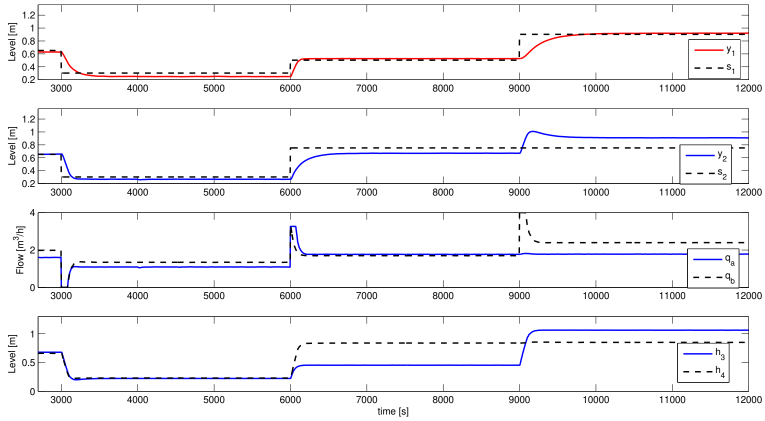


Figure 6: Evaluation of the control test in the real plant of the decentralized MPC scheme.

systems composed of subsystems coupled with the neighboring subsystem through the inputs. The four-tank plant belongs to this class of models. In particular, the linear models (13), and (14) are used to design the distributed controller.

The control objective of this controller is to regulate the system to the given set-points while guaranteeing that a given set of state and input constraints are satisfied. The proposed distributed scheme assumes that for each subsystem, there is a controller that has access to the model and the state of that subsystem. The controllers do not have any knowledge of the dynamics of their neighbor, but can communicate freely among them in order to reach an agreement on the value of the inputs applied to the system. The proposed strategy is based on negotiation between the controllers on the basis of a global performance index. At each sampling time, agents make proposals to improve an initial feasible solution on the basis of their local cost function, state, and model. This initial feasible solution is obtained from the optimal solution of the previous time step and two stabilizing controllers defined by feedback gains K_1 and K_2. These proposals are accepted if the global cost improves the corresponding cost of the current solution. The trajectories chosen are denoted by U_1^d and U_2^d.

The MPC controllers minimize the sum of two local performance indices J_1 and J_2 that depend on the future evolution of both states and inputs:

$$
\begin{aligned}
J_1(x_1, U_1, U_2) &= \sum_{i=1}^{N} \|y_{s1}(i) - r_1\|_{Q_1}^2 + \|u_{s1}(i) - u_{r1}\|_{R_1}^2, \\
J_2(x_2, U_2, U_1) &= \sum_{i=1}^{N} \|y_{s2}(i) - r_2\|_{Q_2}^2 + \|u_{s2}(i) - u_{r2}\|_{R_2}^2,
\end{aligned}
$$

where $r_1 = s_1 - h_1^0$, and $r_2 = s_2 - h_2^0$. The target for the inputs u_{r1} and u_{r2} are given by the steady input u corresponding to the set-point (s_1, s_2) calculated using (1). $U_1 = (u_{s1}(0), \cdots, u_{s1}(N-1))$ and $U_2 = (u_{s2}(0), \cdots, u_{s2}(N-1))$. The predicted outputs are calculated using the model (13) and (14) taking into account $v_{s1} = u_{s2}$ and $v_{s2} = u_{s1}$.

For this benchmark , the weighting matrices were chosen to minimize the benchmark objective function, that is, $Q_1 = Q_2 = I$, $R_1 = R_2 = 0.01$. The prediction horizon N was chosen as $N = 5$. The local controller gains for each controller were $K_1 = [0.17, 0.21]$ and $K_2 = [-0.16, -0.14]$. These gains were designed with LMI techniques based on the full model of the system in order to stabilize both subsystems independently while assuring the stability of the centralized system. The role of these gains is important to guarantee closed-loop stability (see [11] for more details).

Each controller solves a sequence of reduced-dimension optimization problems to determine the future input trajectories U_1 and U_2 based on the model of its subsystem. We summarize the DMPC algorithm proposed in [11] as follows:

1. At time step k, each controller l receives its corresponding partial state measurement $x_{sl}(k)$.
2. Both controllers communicate. Controller 1 sends $K_1 x_{s1}(N)$ and controller 2 sends $K_2 x_{s2}(N)$, where $x_{s1}(N)$ and $x_{s2}(N)$ are the N-steps ahead predicted states obtained from the current states applying $U_1^d(k-1), U_2^d(k-1)$ shifted one time step. This information is used to generate the shifted trajectories $U_l^s(k)$, which is the initial solution.
3. Each controller l minimizes J_l assuming that the neighbor keeps applying the shifted optimal trajectory evaluated at the previous time step $U_{nl}^s(k)$. The optimal solution is denoted by $U_l^*(k)$.
4. Each controller l minimizes J_l optimizing the neighbor input assuming that it applies the shifted input trajectory U_{nl}^s. Solving this optimization problem, controller l defines an input trajectory denoted by $U_{nl}^w(k)$ for its neighbor that optimizes its local cost function J_l.
5. Both controllers communicate. Controller 1 sends $U_1^*(k)$ and $U_2^w(k)$ to controller 2 and receives $U_2^*(k)$ and $U_1^w(k)$.
6. Each controller evaluates the local cost function J_l for each of the nine possible combinations of input trajectories, i.e.,
$$U_1 \in \{U_1^s(k), U_1^w(k), U_1^*(k)\},$$
$$U_2 \in \{U_2^s(k), U_2^w(k), U_2^*(k)\}.$$
7. Both controllers communicate and share the information of the value of their local cost function for each possible combination of input trajectories. In this step, both controllers receive enough information to take a cooperative decision.
8. Each controller applies the input trajectory that minimizes $J = J_1 + J_2$. Because both controllers have access to the same information after the second communication cycle, both controllers choose the same optimal input sets $U_1^d(k), U_2^d(k)$.
9. The first input of each optimal sequence is applied and the procedure is repeated at the next sampling time.

From a game-theoretical point of view, at each time step both controllers 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 controller 1, and each column represents one of the three possible decisions of controller 2. The cells contain the sum of the cost functions of both controllers for a particular choice of future inputs. At each time step, the option that yields a lower global cost is chosen. Note that both controllers share this information, so they both choose the same option. The nine possibilities are shown in Table 2.

The proposals made are suboptimal because each controller has an incomplete view of the system and proposes the best solutions from its own point of view. The proposed algorithm has low communication and computational burdens and provides a feasible solution to the centralized problem assuming that a feasible solution is available to

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

Table 2: Cost function table used for the decision making.

initialize the controller. In addition, an optimization-based procedure to design the controller such that practical stability of the closed-loop is guaranteed is provided in [11]. In this benchmark, the controller has not been designed to guarantee closed-loop stability because neither a terminal region nor a terminal cost has been considered in the controller formulation.

Note that this control scheme is designed for systems controlled only by two agents because the number of options of the cooperative game for more than two controllers grows in a combinatorial manner.

At this point we have to remark the fact that when the reference is switched from one working point to another it is necessary to reset the value of U_s to a feasible solution. This solution is obtained by solving a feasibility problem, in particular a linear programming (LP) problem, based on the full model of the system.

The proposed distributed MPC controller only needs three 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 designed controller has been successfully tested on the real plant; the trajectories are shown in Figure 7. The performance index of the test is $J = 29.5787$. The performance index is close to the performance index of the centralized MPC for regulation. Note however that the input trajectories are not smooth because the controllers switch between different modes.

3.4. Sensitivity-driven DMPC

A novel sensitivity-driven DMPC (S-DMPC) scheme [22] is considered in this subsection. S-DMPC is based on a new distributed dynamic optimization method employing a sensitivity-based coordination mechanism [21]. For the distributed controllers the four-tank system is decomposed first according to (11) and (12). On each prediction horizon $[t_0(k), t_f(k)]$, the continuous-time optimal control problem can be formulated

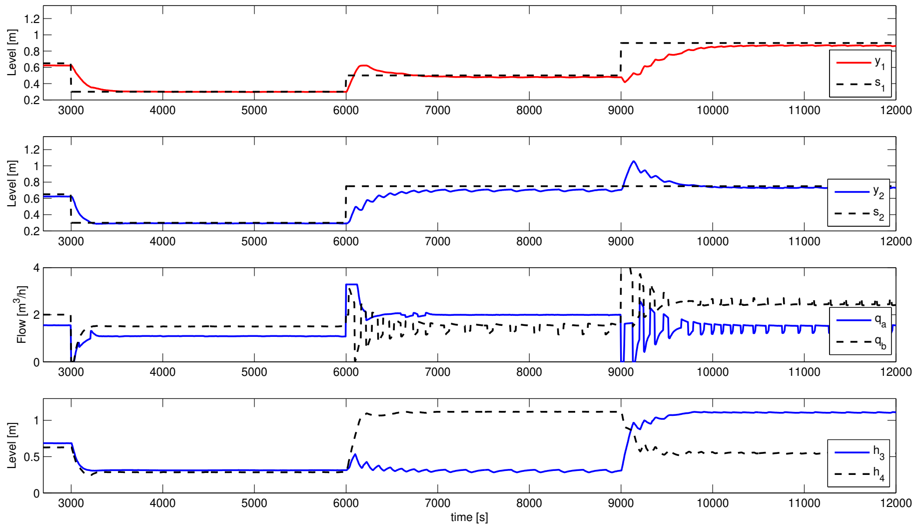


Figure 7: Evaluation of the control test in the real plant of the DMPC scheme based on a cooperative game.

as

$$\min_{u_{sj}} J_j \tag{18a}$$

$$\text{s.t. } J_j = \int_{t_0(k)}^{t_f(k)} (y_{sj} - r_j)^2 + 0.01(u_{sj} - u_{rj})\, dt, \tag{18b}$$

$$\frac{d\, x_{sj}}{dt} = A_{cj} x_{sj} + B_{cj}^{(l)} u_{sj} + B_{cj}^{(j)} v_{sj} + \hat{d}_j,\ l \neq j,\ x_{sj}(0) = x_{sj,0}(k), \tag{18c}$$

$$y_{sj} = C_{cj} x_{sj}, \tag{18d}$$

$$0 \leq D_j(x_{sj}, u_{sj}) + e_j, \qquad j = 1, 2. \tag{18e}$$

where $r_j = s_j - h_j^0$ and the target for the inputs u_{r1} and u_{r2} are given by the steady input u corresponding to the set-point (s_1, s_2) calculated using (1). $x_{sj,0}(k)$ denotes the measured initial conditions at time sample k, and $\hat{d}_j$ denotes additive disturbances to be estimated. If no disturbance estimation is available, $\hat{d}_j,\ j = 1, 2$, are assumed to be zero. The linear state and input constraints are described by matrices D_j and vectors $e_j,\ j = 1, 2$, in equation (18e). In order to solve the continuous-time optimal control problems, they are transcribed into quadratic parametric programming problems by means of control vector parametrization [20], i.e., a discretization of the input variables $u_{sj}(t)$ using parameter vectors p_j with $p = (p_1, p_2)$. As a result the quadratic programs (QPs)

$$\min_{p_j} J_j(p) \tag{19a}$$

$$J_j(p) = \frac{1}{2} p^T \boldsymbol{A}^j p + p^T \boldsymbol{B}^j + \boldsymbol{C}^j, \tag{19b}$$

$$c_j(p) = \boldsymbol{D}^{j\,T} p + \boldsymbol{E}^j \geq 0, \tag{19c}$$

$j = 1, 2$, can be derived, being $\boldsymbol{A}^j$, $\boldsymbol{B}^j$, $\boldsymbol{C}^j$, $\boldsymbol{D}^j$, and $\boldsymbol{E}^j$ appropriate matrices. In order to achieve global optimality, the QP (19) for $j = 1, 2$ are coordinated based on sensitivities [21, 22]. In particular, the objective functions are modified as follows:

$$
\begin{aligned}
J_j^*(p, p^{[\kappa]}) = J_j(p) &+ \left[\sum_{\substack{l=1 \\ l \neq j}}^{2} \left. \frac{\partial J_l}{\partial p_j} \right|_{p^{[\kappa]}} - \left. \frac{\partial c_l}{\partial p_j} \right|_{p^{[\kappa]}}^T \lambda_l^{[\kappa]} \right] (p_j - p_j^{[\kappa]}) \\
&+ \frac{1}{2} (p_j - p_j^{[\kappa]})^T \Omega_j (p_j - p_j^{[\kappa]}), \\
&j = 1, 2 \,.
\end{aligned}
\tag{20}
$$

The first term of the objective function is a copy of the subsystem's objective function. To relate the local optimization problems to the overall objective, all nonlocal contributions are accounted for by linear approximations to result in the second term of the objective function. The third term of the objective function J_i^* is added to improve convergence of the method by means of Wegstein's method [22, 28]. The index $[\kappa]$ indicates variables of the κ-th iteration, and λ_j denotes the Lagrange multipliers associated to the corresponding constraint functions c_j.

The S-DMPC algorithm at control step k comprises the following steps:

1. Transcribe the optimal control problem to compute $\boldsymbol{A}^j$, $\boldsymbol{B}^j(k)$, $\boldsymbol{C}^j(k)$, $\boldsymbol{D}^j$, and $\boldsymbol{E}^j(k)$; $\boldsymbol{A}^j$ and $\boldsymbol{D}^j$ do not depend on the initial state $x_0(k) = x_0(t_0(k))$ and need to be computed only once.
2. Select initial parameters $p^{[0]}(k)$ and an estimate of the initial Lagrange multipliers $\lambda^{[0]}(k)$ based on the solution $(p^*(k-1), \lambda^*(k-1))$ of the last sampling time $k-1$ and set $\kappa := 0$.
3. Send the control parameters $p_j^{[\kappa]}(k)$ and the Lagrange multipliers $\lambda_j^{[\kappa]}(k)$, $j = 1, 2$, to the distributed controllers.
4. Solve the following QP to obtain the minimizer $p_j^{[\kappa+1]}$ and the Lagrange multiplier $\lambda_j^{[\kappa+1]}$:

$$
\begin{aligned}
&\min_{p_j} J_j^* \\
&\text{s.t. } c_j(p) \geq 0, \\
&\qquad j = 1, 2.
\end{aligned}
$$

5. Increase $\kappa := \kappa + 1$ and go back to 3.
6. Stop iteration, if $p^{[\kappa]}$ satisfies a predefined convergence criterion.

The method is implemented with a prediction horizon of 500 seconds, in order to achieve a stable closed-loop control. The input variables u_i have been discretized using 3 parameters for each input. One parameter has been chosen to reflect the steady-state values, while the others have been chosen to approximate the transient part within the first 10 seconds of the horizon by piece-wise constant representations, i.e.

$$u_{si}(t) = \sum_{j=1}^{3} p_{i,j} \cdot \phi_j(t), \text{ with} \tag{21a}$$

$$\phi_1(t) = \begin{cases} 1, & t_0(k) < t < t_0(k) + 5 \\ 0, & \text{else} \end{cases}, \tag{21b}$$

$$\phi_2(t) = \begin{cases} 1, & t_0(k) + 5 < t < t_0(k) + 10 \\ 0, & \text{else} \end{cases}, \tag{21c}$$

$$\phi_3(t) = \begin{cases} 1, & t_0(k) + 10 < t \\ 0, & \text{else} \end{cases} \tag{21d}$$

We have tested the controller for three different configurations:

(a) With a fixed number of 3 iterations, i.e., an implementation without convergence leading to suboptimal control,
(b) with a fixed number of 10 iterations for optimal control, and
(c) with a fixed number of 10 iterations and an additional Kalman filter to eliminate the steady-state offset.

The design of the Kalman filter in configuration (c) aims at improving control performance. The linear Kalman filter

$$\begin{bmatrix} \frac{d\hat{x}_{sj}}{dt} \\ \frac{d\hat{d}_j}{dt} \end{bmatrix} = \begin{bmatrix} A_{cj} & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_{sj} \\ \hat{d}_j \end{bmatrix} + \begin{bmatrix} B_{cj} \\ 0 \end{bmatrix} u_s + K(x_{sj} - \hat{x}_{sj}), \quad \begin{bmatrix} \hat{x}_{sj}(0) \\ \hat{d}_j(0) \end{bmatrix} = \begin{bmatrix} \hat{x}_{sj,0} \\ \hat{d}_{j,0} \end{bmatrix}, \tag{22}$$
$$j = 1, 2 ,$$

is added for each of the subsystems for combined state and disturbance estimation. The additive disturbances $\hat{d}_j \in \mathbb{R}^2$ are introduced to model plant-model mismatch. They are assumed to be constant (or slowly time-varying). The Kalman gain $K \in \mathbb{R}^{4\times 2}$ is calculated using the algebraic Riccati equation.

Due to the strong coupling of the subsystems, convergence of the method is rather slow. It is possible to achieve optimality in approximately 10 iterations. However, already with only three iterations, good performance can be achieved. The performance index in the real plant for the configurations investigated are $J = 45.072$ for configuration (a), $J = 35.525$ for configuration (b), and $J = 28.616$ for configuration (c). The trajectories for configuration (b) are shown in Figure 8, while the trajectories for configuration (c) are given in Figure 9. The Kalman filter in configuration (c) is able to estimate the steady-state disturbances d_j of the plant successfully, such that the steady-state control errors vanish. A non-smooth behavior of the controlled flow rates q_a and q_b can be observed, which is induced by the Kalman filter and could be reduced by a better tuning of the filter. So far, the controllers have only been tuned in a simulation environment and applied to the real plant without further tuning.

3.5. Feasible-cooperation DMPC based on bargaining game theory concepts

In this section, a distributed predictive control scheme based on bargaining game theory is presented. A game is defined as the tuple $(T, \{\Omega_j\}_{j\in T}, \{\phi_j\}_{j\in T})$, where $T = \{1, \ldots, M\}$ is the set of players, Ω_j is a finite set of possible actions of player i, and

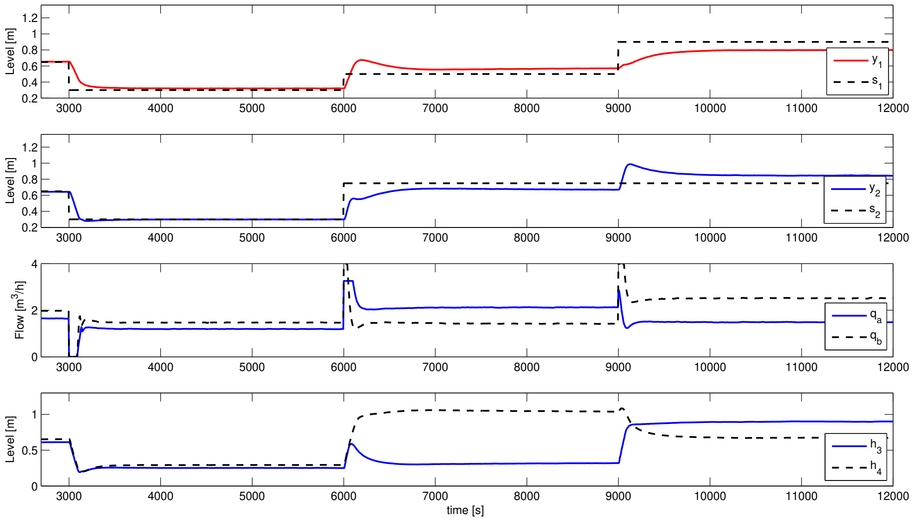


Figure 8: Evaluation of the control test in the real plant of the S-DMPC scheme.

$\phi_j : \Omega_1 \times \ldots \times \Omega_M \longrightarrow R$ is the payoff function of the j-th player [1].
Based on the definition of a game, a DMPC problem can be defined as a tuple $G = (T, \{\Omega_j\}_{j\in T}, \{\phi_j\}_{j\in T})$, where $T = \{1, \ldots, M\}$ is the set of subsystems, Ω_j is the non-empty set of feasible control actions for subsystem j, and $\phi_j : \Omega_1 \times \ldots \times \Omega_M \longrightarrow \mathbb{R}$, where ϕ_j is the cost function of the j-th subsystem. From this point of view, DMPC is a game in which the players are the subsystems, the actions are the control inputs, and the payoff of each subsystem is given by the value of its cost function.

In the specific case of the four-tank plant, the whole system model has been decomposed into two subsystems modeled by (9) and (10). Based on these prediction models, the cost functions used to measure the performance of subsystem j, $j = 1, 2$, is

$$\ell_j(\widetilde{y}_j(k) - \widetilde{r}_j(k), \widetilde{u}_j(k) - \widetilde{u}_{rj}(k)) = \|\widetilde{y}_j(k) - \widetilde{r}_j(k)\|^2 + \|\widetilde{u}_j(k) - \widetilde{u}_{rj}(k)\|^2, \tag{23}$$

where $\widetilde{y}_j(k) = (y_{sj}(k|k), \ldots, y_{sj}(k+N|k))$, $\widetilde{u}_j(k) = (u_{sj}(k|k), \ldots, u_{sj}(k+N|k))$, $\widetilde{r}_j(k) = (r_j(k), \cdots, r_j(k))$ and $\widetilde{u}_{rj}(k) = (u_{srj}(k), \cdots, u_{srj}(k))$. The target $r_j(k)$ is given by $r_j(k) = s_j(k) - h_j^0$ and the target for the inputs $u_{rj}(k)$ is given by the steady input $r_j(k)$ calculated using (9) or (10).
Therefore, the DMPC of the four-tank system is a game with $T = \{1, 2\}$, in which the feasible set Ω_j is determined by the constraint, the state, and the input of the j-th system Z_j. The feasible cost function for a given sequence of predicted inputs $\widetilde{u}(k)$, $\phi_j(\widetilde{u}(k))$, is a quadratic function obtained from (23) by calculating the predictions $\widetilde{y}_j(k)$ using the following recursion (derived from the models (9) and (10)):

$$\begin{aligned} x_{sj}(k+i+1|k) &= A_j x_{sj}(k+i|k) + B_j u_s(k+i|k), \\ y_{sj}(k+i|k) &= C_j x_{sj}(k+i|k), \end{aligned}$$

with $x_{sj}(k|k) = x_{sj}(k)$.

Following the cooperative game theory introduced in [14, 15, 18], the formulation of the DMPC as a game is completed by introducing the concept of a disagreement

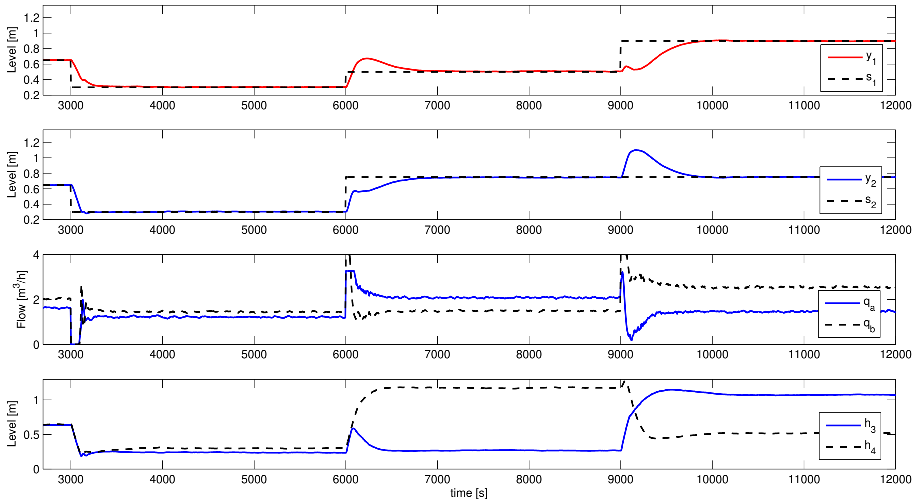


Figure 9: Evaluation of the control test in the real plant of the S-DMPC scheme with Kalman filter.

point. The disagreement point, $\delta_j(k)$, at time step k, is defined as the benefit that the j-th player receives when no agreement is achieved among the players. In the case of DMPC, the disagreement point can be computed as follows

$$\begin{aligned} \delta_j(k) &= \arg \min_{\widetilde{u}_j(k)} \max_{\widetilde{u}_{-j}(k)} \phi_j(\widetilde{u}(k)) \\ &\text{s.t. } \widetilde{u}_j(k) \in \Omega_j\,, \\ &\qquad \widetilde{u}_{-j}(k) \in \Omega_{-j}\,, \end{aligned} \tag{24}$$

where $\widetilde{u}_j(k)$ denotes the solution of the j-th player at time step k, $\widetilde{u}_{-j}(k) = (\widetilde{u}_1(k), \ldots, \widetilde{u}_{j-1}(k), \widetilde{u}_{j+1}(k), \ldots, \widetilde{u}_M(k))$ and $\Omega_{-j} = \Omega_1 \times \ldots \times \Omega_{j-1} \times \Omega_{j+1} \times \ldots \times \Omega_M$.
Note that the optimization problem (24) defines the worst case for subsystem j. Then, $\delta_j(k)$ is the best benefit that the j-th subsystem can achieve given the worst case.

According to [13, 15], the solution of the cooperative game associated with the DMPC problem can be computed as the solution of the optimization problem [7, 18]

$$\begin{aligned} &\max_{\widetilde{u}(k)} \sum_{j=1}^{M} w_j \log\left(\delta_j(k) - \phi_j(\widetilde{u}(k))\right) \\ &\text{s.t. } \delta_j(k) > \phi_j(\widetilde{u}(k)), \text{ for } j = 1, \ldots, M \\ &\quad \widetilde{u}_j(k) \in \Omega_j, \text{ for } j = 1, \ldots, M, \end{aligned} \tag{25}$$

where w_j are weights with $w_j > 0$ and $\sum_{j=1}^{M} w_j = 1$. This problem can be solved in a distributed fashion using the feasible-cooperation approach presented in [26, 27].

Let $\phi_j(\widetilde{u}(k)) = \sigma_j(\widetilde{u}_l(k), \widetilde{u}_{-l}(k))$. Then assuming $\widetilde{u}_{-l}(k)$ fixed, the maximization problem

$$\begin{aligned}
&\max_{\widetilde{u}_l(k)} \sum_{j=1}^{M} w_j \log\left[\delta_j(k) - \sigma_j(\widetilde{u}_l(k), \widetilde{u}_{-l}(k))\right] \\
&\text{s.t. } \delta_j(k) > \sigma_j(\widetilde{u}_l(k), \widetilde{u}_{-l}(k)), \text{ for } j = 1, \ldots, M \\
&\widetilde{u}_l(k) \in \Omega_l
\end{aligned} \tag{26}$$

defines the maximum profit that the whole system can achieve while the control actions of the other subsystems are fixed at $\widetilde{u}_{-i}(k)$. Thus, the maximization problem (25) can be solved in a distributed (and cooperative) way by letting each subsystem i solve (26). It is easy to verify that (26) corresponds to a convex minimization problem, for which efficient solvers are accessible.

With the purpose of implementing the DMPC controller described in this section, the following steps have been proposed:

1. Given the initial conditions, $x(k)$, all subsystems compute their disagreement points $d_i(k)$ according to (24) in a separated way.
2. After computing the disagreement points, each subsystem sends its disagreement point to the other subsystems.
3. Each subsystem solves the optimization problem (26). If (26) is feasible, let $\widetilde{u}^*_{i,q}(k)$ be an optimal solution (so it satisfies the constraints, i.e., $\delta_r(k) > \sigma_r(\widetilde{u}^*_{i,q}(k), \widetilde{u}_{-i,q-1}(k))$, for $r = 1, \ldots, M$). If (26) is not feasible, subsystem i decides not to cooperate. In this step, if $q = 1$, then $\widetilde{u}^{\mathrm{d}}_i(k)$ is considered as initial condition for subsystem i, for solving (26). Otherwise, $\widetilde{u}_{i,q-1}(k)$ is considered as initial condition for subsystem i, for solving (26).
4. The subsystems that decide to cooperate update their control actions by a convex combination $\widetilde{u}_{i,q}(k) = w_i\widetilde{u}^*_{i,q}(k) + (1 - w_i)\widetilde{u}_{i,q-1}(k)$. The subsystems that decide not to cooperate select their control actions equal to $\widetilde{u}_{i,q}(k) = w_i\widetilde{u}^{\mathrm{d}}_i(k) + (1 - w_i)\widetilde{u}_{i,q-1}(k)$, where $0 < w_i < 1$.
5. Each subsystem sends its control actions to the other subsystems. If $\|\widetilde{u}_{i,q}(k) - \widetilde{u}_{i,q-1}(k)\| \leqslant \xi$ ($\xi > 0$) for all subsystems, or if $q = q_{\max}$, or if the maximum allowable time for the computation of the optimal control input $\widetilde{u}^*(k) = (\widetilde{u}^*_1(k), \ldots, \widetilde{u}^*_M(k))$ has been reached, the first element of the control sequence $\widetilde{u}_{i,q}(k)$ is applied and each subsystem returns to step 1. Else, each subsystem returns to step 3.

At time step $k + 1$ the initial conditions for subsystem i for solving (24) are determined by the shifted control sequence $\widetilde{u}_{i,0}(k + 1) = (u^*_{i,q_{\text{end}}}(k + 1, k), \ldots, u^*_{i,q_{\text{end}}}(k + N_u, k), 0)$, where $u^{*T}_{i,q_{\text{end}}}(k + 1, k)$ denotes the optimal value of the control inputs for subsystem i at iteration q_{end} at time step $k + 1$ given the conditions at time step k.

Figure 10 shows the behavior of the four-tank system, when the DMPC controller based on game-theoretical concepts computes the optimal control inputs. The performance index calculated for the control test is $J = 46.3177$. This result was obtained considering $q_{\max} = 1$.

Note that the aim of the game-theoretical formulation of the DMPC problem is that the subsystems cooperate while obtaining some benefit. From Figure 10, it is possible to conclude that this aim has been achieved, because the pumps are working jointly in order to reach the reference values for the levels h_1 and h_2, which is the global control objective. Also, the control decisions are taken in a cooperative way. Therefore, when

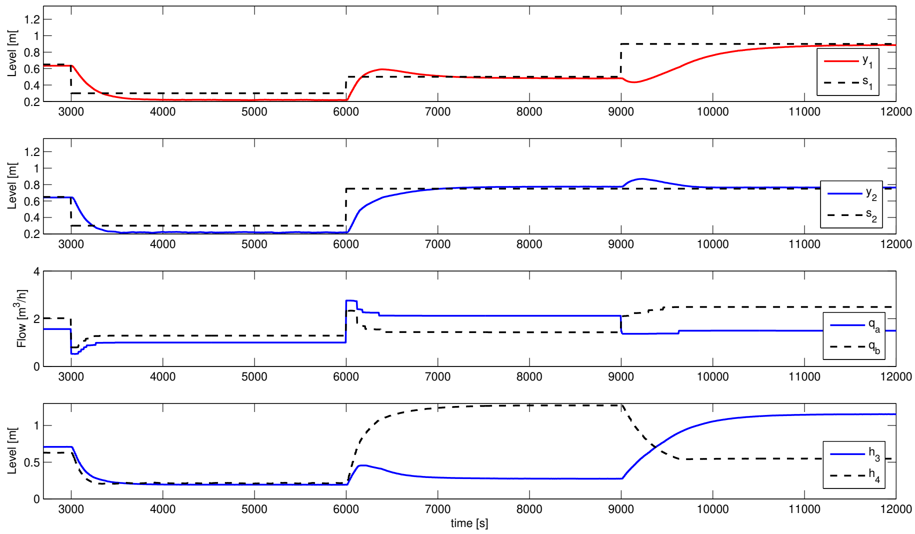


Figure 10: Evaluation of the control test in the real plant of the DMPC scheme based on a bargaining game.

the changes in the reference values were applied, the pumps react with the purpose of achieving the new operating point in a cooperative fashion without sacrificing the local performance.

3.6. Serial DMPC scheme

We have implemented the scheme proposed in [16, 17] for the four-tank system. This scheme is derived from a serial decomposition of an augmented Lagrangian formulation of the centralized overall MPC problem. This results in a scheme in which controllers perform at each control step a number of iterations to obtain agreement on which actions should be taken. The goal of the iterations is to obtain actions that are optimal from a system-wide point of view using only local models and measurements and communicating only with neighboring agents on values of interconnecting variables.

Below we first summarize the assumptions and characteristics of the serial DMPC scheme. Then we describe how this scheme can be used to control the four-tank system.

3.6.1. Local dynamics and objectives

In general, consider a system divided into $n = 2$ subsystems. The dynamics of subsystem $j \in \{1, \ldots, n\}$ are assumed to be adequately modeled by the deterministic linear discrete-time time-invariant model (13) and (14), where the $v_{sj}(k)$ variables represent the influence of other subsystems on subsystem j. For variables $x_{sj}(k), u_{sj}(k), y_{sj}(k)$ upper and lower bounds are specified.

So-called interconnecting input variables $\mathbf{w}_{\text{in},lj}(k)$ represent the variables of subsystem j that are influenced by subsystem l, i.e., a selection of $v_{sj}(k)$. So-called interconnecting output variables $\mathbf{w}_{\text{out},lj}(k)$ are the variables of subsystem j that influence a neighboring subsystem l, i.e., a selection of $x_{sj}(k)$, $u_{sj}(k)$, and $y_{sj}(k)$. Define the interconnecting inputs and outputs for the control problem of controller j at control step k

as

$$\mathbf{w}_{\text{in},j}(k) = y_{s\,j}(k), \quad \mathbf{w}_{\text{out},j}(k) = E_j(x_{s\,j}(k), u_{s\,j}(k), y_{s\,j}(k)), \tag{27}$$

where E_j is an interconnecting output selection matrix that contains zeros everywhere, except for a single 1 per row corresponding to a local variable that corresponds to an interconnecting output variable. The variables $\mathbf{w}_{\text{in},j}(k)$, $\mathbf{w}_{\text{out},j}(k)$ are partitioned such that

$$\mathbf{w}_{\text{in},j}(k) = (\mathbf{w}_{\text{in},l_{j,1}j}(k), \ldots, \mathbf{w}_{\text{in},l_{j,m_j}j}(k)), \tag{28}$$

$$\mathbf{w}_{\text{out},j}(k) = (\mathbf{w}_{\text{out},l_{j,1}j}(k), \ldots, \mathbf{w}_{\text{out},l_{j,m_j}j}(k)), \tag{29}$$

where $\mathcal{N}_j = \{l_{i,1}, \ldots, l_{j,m_j}\}$ is the set of indexes of the m_j neighbors of subsystem j. The interconnecting inputs to the control problem of controller j with respect to controller l must be equal to the interconnecting outputs from the control problem of controller l with respect to controller j, since the variables of both control problems model the same quantity. For controller j this thus gives rise to the following interconnecting constraints

$$\mathbf{w}_{\text{in},lj}(k) = \mathbf{w}_{\text{out},jl}(k), \quad \mathbf{w}_{\text{out},lj}(k) = \mathbf{w}_{\text{in},lj}(k). \tag{30}$$

The controllers are assumed to be striving for the best overall network performance in a distributed way. In addition, the common assumption is made that the objectives of the controllers can be represented by convex functions $J_{\text{local},j}$, for $j \in \{1, \ldots, n\}$, which are typically linear or quadratic.

3.6.2. Scheme outline

The distributed MPC scheme for n agents comprises at control step k the following steps:

1. For $j = 1, \ldots, n$, controller j makes a measurement of the current state of the subsystem $x_{s\,j}(k)$.
2. The controllers cooperatively solve their control problems in the following serial iterative way[1]:
   (a) The iteration counter s is set to 1 and the Lagrange multipliers $\tilde{\lambda}^{(s)}_{\text{in},lj}(k)$, $\tilde{\lambda}^{(s)}_{\text{out},jl}(k)$ are initialized arbitrarily.
   (b) For $j = 1, \ldots, n$, one controller j after another determines $\tilde{x}_{s\,j}^{(s)}(k+1)$, $\tilde{u}_{s\,j}^{(s)}(k)$, $\tilde{\mathbf{w}}^{(s)}_{\text{in},lj}(k)$, $\tilde{\mathbf{w}}^{(s)}_{\text{out},lj}(k)$ as solutions of the optimization problem

$$\min \; J_{\text{local},j}\left(\tilde{x}_{s\,j}(k+1), \tilde{u}_{s\,j}(k), \tilde{y}_{s\,j}(k)\right) + \sum_{l \in \mathcal{N}_j} J^{(s)}_{\text{inter},j}\left(\tilde{\mathbf{w}}_{\text{in},lj}(k), \tilde{\mathbf{w}}_{\text{out},lj}(k)\right), \tag{31}$$

   subject to the local dynamics (13)–(14) (including the bound constraints) and (27) of subsystem j over the horizon and the current state $x_{s\,j}(k)$. The

[1]The tilde notation is used to represent the predicted variables over the prediction horizon N.

additional performance criterion $J_{\text{inter},j}$ in (31) at iteration s is defined as

$$J^{(s)}_{\text{inter},j}\left(\tilde{\mathbf{w}}_{\text{in},lj}(k), \tilde{\mathbf{w}}_{\text{out},lj}(k)\right) =$$
$$\begin{bmatrix} \tilde{\lambda}^{(s)}_{\text{in},lj}(k) \\ -\tilde{\lambda}^{(s)}_{\text{out},jl}(k) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \tilde{\mathbf{w}}_{\text{in},lj}(k) \\ \tilde{\mathbf{w}}_{\text{out},lj}(k) \end{bmatrix} + \frac{\gamma_{\text{c}}}{2} \left\| \begin{bmatrix} \tilde{\mathbf{w}}_{\text{in,prev},jl}(k) - \tilde{\mathbf{w}}_{\text{out},lj}(k) \\ \tilde{\mathbf{w}}_{\text{out,prev},jl}(k) - \tilde{\mathbf{w}}_{\text{in},lj}(k) \end{bmatrix} \right\|^2,$$

where $\tilde{\mathbf{w}}_{\text{in,prev},jl}(k) = \tilde{\mathbf{w}}^{(s)}_{\text{in},jl}(k)$ and $\tilde{\mathbf{w}}_{\text{out,prev},jl}(k) = \tilde{\mathbf{w}}^{(s)}_{\text{out},jl}(k)$ is the information computed at the current iteration s for each controller $l \in \mathcal{N}_j$ that has solved its problem *before* controller j in the *current* iteration s. In addition, $\tilde{\mathbf{w}}_{\text{in,prev},jl}(k) = \tilde{\mathbf{w}}^{(s-1)}_{\text{in},jl}(k)$ and $\tilde{\mathbf{w}}_{\text{out,prev},jl}(k) = \tilde{\mathbf{w}}^{(s-1)}_{\text{out},jl}(k)$ is the information computed at the *previous* iteration $s-1$ for the other controllers. The constant γ_{c} is a positive scalar that penalizes the deviation from the interconnecting variable iterates that were computed by the controllers before controller j in the current iteration and by the other controllers during the last iteration. The results $\tilde{\mathbf{w}}^{(s)}_{\text{in},lj}(k)$ and $\tilde{\mathbf{w}}^{(s)}_{\text{out},lj}(k)$ of the optimization are sent to controller l.

(c) Update the Lagrange multipliers,

$$\tilde{\lambda}^{(s+1)}_{\text{in},lj}(k) = \tilde{\lambda}^{(s)}_{\text{in},lj}(k) + \gamma_{\text{c}}\left(\tilde{\mathbf{w}}^{(s)}_{\text{in},lj}(k) - \tilde{\mathbf{w}}^{(s)}_{\text{out},jl}(k)\right). \tag{32}$$

Send $\tilde{\lambda}^{(s+1)}_{\text{in},lj}(k)$ to controller l and receive the multipliers from controller l to be used as $\tilde{\lambda}^{(s+1)}_{\text{out},jl}(k)$.

(d) Move on to the next iteration $s+1$ and repeat steps 2b–2c. The iterations stop when the infinity norm for each $\tilde{\lambda}^{(s+1)}_{\text{in},lj}(k) - \tilde{\lambda}^{(s)}_{\text{in},lj}(k)$ is smaller than a small positive scalar γ_{ϵ}.

3. The controllers implement the actions until the beginning of the next control step.

The scheme just presented does not guarantee stability; however, as the interaction between controllers is taken into account via the objective function only, the local optimization problems remain feasible over the iterations. In addition, under the assumptions on the objective functions and prediction models the solution of this scheme converges to the solution that a centralized MPC controller would have obtained for a sufficiently small γ_{ϵ} and given sufficient time for performing iterations.

3.6.3. The four-tank system

For control of the four-tank system two subsystems are defined, according to the partition proposed in Section 2.3, but in this case, $\mathbf{w}_{\text{in},21} = u_1$, $\mathbf{w}_{\text{out},12} = u_1$, $\mathbf{w}_{\text{in},12} = u_2$, $\mathbf{w}_{\text{out},21} = u_2$. The local control objectives are defined as follows:

$$J_{\text{local},1} = \sum_{i=0}^{N-1}\left((y_{s1}(i+1) - r_1)^2 + 0.01(u_{s1}(i) - u_{r1})^2\right)$$
$$J_{\text{local},2} = \sum_{i=0}^{N-1}\left((y_{s2}(i+1) - r_2)^2 + 0.01(u_{s2}(i) - u_{r2})^2\right).$$

where $r_1 = s_1 - h_1^0$, and $r_2 = s_2 - h_2^0$. The target for the inputs u_{r1} and u_{r2} are given by the steady input u corresponding to the set-point (s_1, s_2) calculated using (1). The output of each subsystem is predicted using the models given by (13) and (14). The

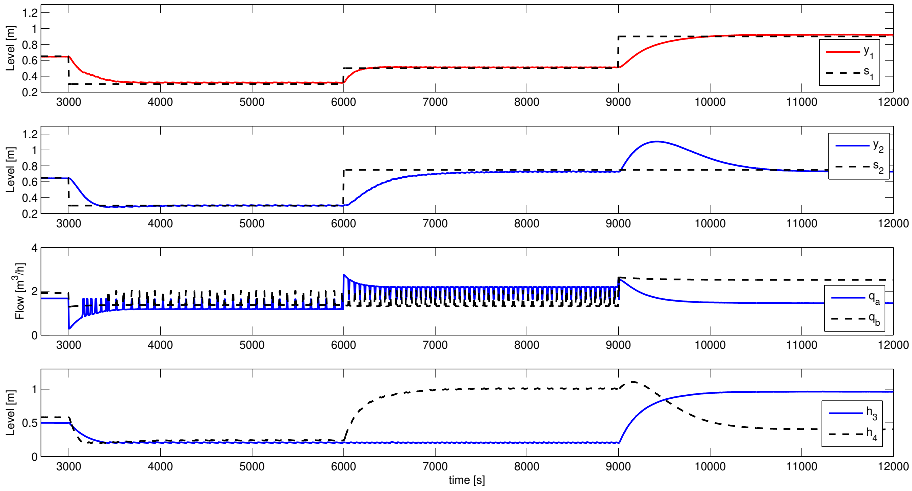


Figure 11: Evaluation of the control test on the real plant of the Serial DMPC scheme.

control test performed is done using as parameters $\gamma_c = 1$, $\gamma_\epsilon = 1e^{-2}$, and $N = 5$. Controller 1 starts the iterations.

Figure 11 shows the trajectories resulting from the control test. The calculated performance index for this controller is $J = 38.18$.

4. Evaluation and comparison of the results of the benchmark

4.1. Evaluation of the controllers

In this paper, eight different MPC controllers have been considered. Table 3 shows some qualitative properties of these controllers. The entry *Model Requirements* shows whether the controllers need full or partial knowledge of the system and whether the model used is linear or nonlinear. The entry *Control Objectives* shows whether the controller is optimal from a centralized point of view (i.e., provides the same solution as the centralized MPC for regulation), guarantees constraint satisfaction if a feasible solution is obtained and whether it can be designed to guarantee closed-loop stability in a regulation problem. The *Auxiliary Software* entry shows which type of additional software is needed by each controller of the distributed scheme.

The two centralized controllers are based on a linear prediction model of the full plant and are included as a reference for the performance of the distributed MPC schemes. Note that if the controllers could communicate without limits, they would be able to obtain the optimal centralized solution for the linear model of the plant. Notice that the real optimal centralized controller should consider an accurate nonlinear prediction model of the plant. This has not been implemented in the benchmark and could be considered as future work.

On the other hand, the decentralized controller provides a reference on what can be achieved with no communication among the controllers at all. All the distributed predictive controllers are based on linear prediction models and assume that each agent has access only to its local state and model.

It is worth noting that the centralized MPC scheme for tracking can designed to guarantee closed-loop stability not only for regulation problems, but also for tracking problems with any given reference at the cost of optimality. In this benchmark, all the ingredients needed to provide stability guarantees for the nominal case were taken into account. The decentralized controller considered cannot guarantee optimality, constraint satisfaction, nor stability. Note that in order to guarantee closed-loop stability, the DMPC scheme based on a cooperative game needs full model knowledge in order to design the optimization problem (including the terminal region, the terminal cost function, and the corresponding local controllers, see [11]) of each agent. In this benchmark, this controller was not designed to guarantee closed-loop stability.

The distributed controllers that guarantee optimality (provided sufficient evaluation time) are the serial DMPC scheme and the S-DMPC scheme. Note that these controllers are also the ones with a larger communication and computational burden.

Another key issue in distributed schemes is the class of computational capabilities that each controller must have. In particular, for the schemes considered each controller must be able to solve either QP problems or general nonlinear optimization problems. In the experiments, the controllers used MATLAB's optimization toolbox, in particular `quadprog` and `fmincon`.

The properties of each of the proposed controllers are discussed and studied in the previous works which have been included in the references. In this paper, we comment these theoretical properties in order to compare these controllers. Note however, that in general, these properties may not hold in the proposed benchmark because the theoretical properties often assume that there are no modeling errors or disturbances and that a given set of assumptions hold. We have carried out all the experiments with the real plant, so there are modeling errors and disturbances. In addition, although most of the controllers are defined for regulation, the benchmark is a reference tracking problem. Issues such as steady-state error and disturbance estimation play a relevant role in this benchmark. These issues may cause the designed controllers to not satisfy the design conditions established in the original work.

4.2. Evaluation of the experimental results

The experimental results demonstrate how centralized solutions provide the best performance while the performance of a fully decentralized controller is worse. Distributed schemes in which the controllers communicate in general improve this performance, although the experimental results also demonstrate that a distributed MPC scheme is not necessarily better (according to a certain performance index) than a decentralized scheme and it depends on the formulation of the controller and its design.

It is also clear how those controllers that incorporate offset-free techniques (the MPC scheme for tracking, the MPC scheme for regulation and the S-DMPC scheme with Kalman filter) provide a better performance index. In order to obtain a measure of the performance without the effect of the steady offset, the transient performance index J_t has been calculated. This index is evaluated computing the cumulated cost during the transient. The entry t_s shows the cumulated settling times of the three reference changes. This shows that those offset-free controllers have a transient performance index similar to the total performance index while for the rest of the controllers, the transient index is better. Note that this index only evaluates the performance during the transient and does not take into account steady-state errors. It can be seen that the decentralized scheme shows the shortest settling time t_s and the best transient performance J_t, although this controller exhibits the worst overall performance J. This is

Qualitative properties	Model Requirements	Control Objectives	Auxiliary Software
Centralized Tracking MPC	Linear system Full model	Suboptimal Constraints Stability	QP
Centralized Regulation MPC	Linear system Full model	Optimal Constraints Stability	QP
Decentralized MPC	Linear system Local model	Suboptimal	QP
DMPC Cooperative game	Linear system Local model (*Full model*)	Suboptimal Constraints (*Stability*)	QP
S-DMPC	Linear system Local model	Optimal Constraints	QP
DMPC Bargaining game	Linear system Local model	Suboptimal Constraints	NLP
Serial DMPC	Linear system Local model	Optimal Constraints	QP

Table 3: Table of qualitative properties of each tested controller.

due to the fact that the controller rapidly reaches an equilibrium point of the controlled system that's far from the real set-point (see the third step response in Figure 6).

All the controllers were implemented using a MATLAB function and were not designed to optimize the computational time. For this reason, the computation time has not been taken into account. These computation times were lower than the sampling time chosen for each controller and, moreover, they could be dramatically reduced using an appropriate implementation framework.

Motivated by these issues, the computational burden is best measured by the number and size of the optimization problems solved at each sampling time. The centralized schemes solve a single QP problem with $2N$ optimization variables while the decentralized controller solves 2 QP problems with N optimization variables. The difference in the computational burden between these schemes grows with the prediction horizon and the number of subsystems. Distributed schemes try to find a trade-off between the burden of computation and communication, and optimality. The DMPC scheme based on a cooperative game and the DMPC scheme based on a bargaining game solve a fixed number of low-complexity optimization problems. The S-DMPC scheme and the serial DMPC scheme provide optimality at the cost of a higher computational burden.

On the other hand, the communicational burden of each controller is measured by the average number of floating point numbers that have to be transmitted each sampling time by each agent and the number of communication cycles involved. It can be seen that iterative DMPC schemes (S-DMPC and Serial DMPC) in general need to transmit a larger amount of information, while the two controllers based on game theory reach suboptimal cooperative solutions with a lower communicational burden.

The centralized and distributed predictive controllers tested can potentially deal with the satisfaction of hard constraints in the inputs and states of the plant under

Control performance	J	J_t	t_s	N	# floats	# trans
Centralized Tracking MPC	28.4	28.12	3280	5	N.D	N.D.
Centralized Regulation MPC	25.46	23.78	2735	5	N.D	N.D.
Decentralized MPC	39.54	21.2	1685	5	0	0
DMPC Cooperative game	30.71	28.19	2410	5	20	3
S-DMPC (w/o KF)	35.65	23.28	2505	100	33	10
S-DMPC (with KF)	28.61	28.26	1895	100	33	10
DMPC Bargaining game	46.32	39.52	3715	5	6	2
Serial DMPC	38.18	35.96	2800	5	[20,70]†	[2,7]†

†: [a,b] denotes a possible value in this interval.

Table 4: Table of the quantitative benchmark indexes of each tested controller. # floats stands for the number of floating-point reals transmitted between the controllers during a sampling period. # trans denotes the number of data packets transmitted during a sampling period.

appropriate assumptions. However, state constraints are not active throughout the evolution of the controlled system although there exist states close to the physical limits of the plant. All the controllers considered have demonstrated good properties in the closed-loop experiments carried out, exhibiting stable and feasible trajectories in spite of the disturbances and mismatches between the prediction model and the plant.

5. Conclusions

In this paper the results of the HD-MPC four-tank benchmark have been presented. In this benchmark, eight different MPC controllers were applied to the four-tank process plant. These controllers were based on different models and assumptions and provide a broad view of the different distributed MPC schemes developed within the HD-MPC project. The results obtained show how distributed strategies can improve the results obtained by decentralized strategies using the information shared by the controllers. Future work will focus on benchmarking of more complex systems involving more than two subsystems and on testing on the four-tank process a centralized nonlinear MPC controller with a sufficiently large prediction horizon in order to measure the loss of performance due to the linear nature of the prediction model.

References

[1] O. Akira. A noncooperative approach to general *n*-person cooperative games. Discussion Papers 2005-01, Graduate School of Economics, Hitotsubashi University, 2005.

[2] I. Alvarado, D. Limon, and Miguel A. Ridao. Documentation for benchmark cases-part I (HD-MPC D6.3.1 Part I). Public deliverable, HD-MPC Consortium, 2010. This deliverable and a delaited description of the plant and the simulation model can be downloaded from `http://www.ict-hd-mpc.eu`.

[3] K. J. ström and B. Wittenmark. *Computer-Controlled Systems: Theory and Design*. Prentice Hall, 1997.

[4] E. Bristol. On a new measure of interaction for multivariable process control. *IEEE Transactions on Automatic Control*, 11:133–134, 1966.

[5] L. Dai and K. J. ström. Dynamic matrix control of a quadruple tank process. In *IFAC World Congress*, 1999.

[6] E. P. Gatzke, E. S. Meadows, C. Wang, and F. J. Doyle III. Model based control of a four-tank system. *Computers & Chemical Engineering*, 24:1503–1509, 2000.

[7] J. C. Harsanyi. A simplified bargaining model for the *n*-person cooperative game. *International Economic Review*, 4(2):194–220, 1963.

[8] K. H. Johansson. The quadruple-tank process. *IEEE Transactions on Control Systems Technology*, 8:456–465, 2000.

[9] D. Limon, I. Alvarado, T. Alamo, and E. F. Camacho. MPC for tracking of piecewise constant references for constrained linear systems. *Automatica*, 44(9):2382–2387, 2008.

[10] D. Limon, I. Alvarado, T. Alamo, and E. F. Camacho. Robust tube-based MPC for tracking of constrained linear systems with additive disturbances. *Journal of Process Control*, 20(3):248–260, 2010.

[11] J. M. Maestre, D. Muñoz de la Peña, and E. F. Camacho. Distributed model predictive control based on a cooperative game. *Optimal Control Applications and Methods, Accepted for publication*, 2010.

[12] M. Mercangöz and F. J. Doyle III. Distributed model predictive control of an experimental four-tank system. *Journal of Process Control*, 17:297–308, 2007.

[13] J. F. Nash. Equilibrium points in *n*-persons games. *Proceedings of the National Academy of Sciences of United States of America*, 36(1):48–40, 1950.

[14] J. F. Nash. The bargaining problem. *Econometrica*, 18(2):155–162, 1950.

[15] J. F. Nash. Two-person cooperative games. *Econometrica*, 21(1):128–140, 1953.

[16] R. R. Negenborn, B. De Schutter, and J. Hellendoorn. Multi-agent model predictive control for transportation networks: Serial versus parallel schemes. *Engineering Applications of Artificial Intelligence*, 21(3):353–366, 2008.

[17] R. R. Negenborn, P. J. van Overloop, T. Keviczky, and B. De Schutter. Distributed model predictive control for irrigation canals. *Networks and Heterogeneous Media*, 4(2):359–380, 2009.

[18] H. J. Peters. *Axiomatic Bargaining Game Theory*. Springer, 1992.

[19] J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design*. Nob-Hill Publishing, 1st edition, 2009.

[20] R. Sargent and G. Sullivan. The development of an efficient optimal control package. In J. Stoer, editor, *Proceedings of the 8th IFIP Conference on Optimization Technique*, pages 158–168. Springer, 1978.

[21] H. Scheu, J. Busch, and W. Marquardt. Nonlinear distributed dynamic optimization based on first order sensitivities. In *American Control Conference*, Baltimore, Maryland, USA, July 2010.

[22] H. Scheu and W. Marquardt. Sensitivity-based coordination in Distributed Model Predictive Control. *Journal of Process Control, accepted for publication*, 2010.

[23] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia. Cooperative distributed model predictive control. *Systems & Control Letters*, 59:460–469, 2010.

[24] R. Vadigepalli, E. P. Gatzke, E. S. Meadows, C. Wang, and F. J. Doyle III. Robust control of a multivariable experimental 4-tank system. *Computers & Chemical Engineering*, 24:1503–1509, 2000.

[25] A. Venkat, J. Rawlings, and S. Wright. Distributed model predictive control of large-scale systems. In *Assessment and Future Directions of Nonlinear Model Predictive Control*, pages 591–605. Springer, 2007.

[26] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright. Distributed MPC strategies for automatic generation control. In *Proceedings of the IFAC Symposium on Power Plants and Power Systems Control*, pages 25–28, 2008.

[27] A. N. Venkat, J. B. Rawlings, and S. J. Wright. Stability and optimality of distributed, linear model predictive control. Part I: State feedback. *Texas-Wisconsin Modeling and Control Consortium Technical Report*, 3, 2006.

[28] Joseph Henry Wegstein. Accelerating convergence of iterative processes. *Communications of the ACM*, 1(6):9–13, 1958.