Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

A nonlinear distributed model predictive scheme for systems based on Hammerstein model

Degachi, H. * **Chanfreut, P.** ** **Maestre, J.M.** **

* *National Engineering school of Tunis, Laboratory of Analysis, Conception and Control of Systems, Tunis El Manar University, Tunisia (e-mail: degachihajer@gmail.com)*
** *Systems and Automation Engineering Department, University of Seville, Spain (e-mail: pchanfreut,pepemaestre@us.es).*

Abstract: A distributed model predictive control (DMPC) scheme for systems based on the Hammerstein structure is proposed in this work. To deal with the nonlinearity of the Hammerstein model, the DMPC problem is reduced to an optimization problem on the intermediate variable and, subsequently, the inverse of the nonlinear block is considered to find the corresponding control inputs. Also, a sub-optimality bound of the method is presented. To illustrate our approach, we consider a nonlinear system controlled in a distributed manner by two agents that exchange information and make cooperative decisions. The efficiency of the proposed DMPC is demonstrated through a simulation example where it is compared to the corresponding centralized approach.

Keywords: Distributed control, Distributed feedback, Linear Control Systems, Network topologies, Control Systems Design.

1. INTRODUCTION

Model predictive control (MPC) is an optimization based control framework that has gained special relevance in the last decades (Garcia et al. (1989); Mayne et al. (2000)). MPC controllers make use of a model of the system and calculate the control actions through iterative optimizations of a certain performance criterion (Camacho and Bordons (2013)). In particular, each time instant, a sequence of actions for a future window of time is computed and the first input is implemented in the system. Subsequently, the procedure is repeated based on the receding horizon principle.

In its earliest form, the MPC strategy was proposed in a centralized manner, where a single optimization problem was defined for the overall system. However, this approach hinders its applicability in large scale systems due to the high computation burden required to solve the optimization problems, especially in the case of nonlinear systems. To overcome this issue, non-centralized forms of MPC have emerged in the past years. In this context, the global system is partitioned into a set of subsystems that are governed by local MPC controllers. Within this framework, decentralized MPC comprises those approaches where the set of controllers do not exchange any information and, therefore, base their computations solely on local data (Siljak (2011)). On the other hand, distributed MPC considers the possibility of interactions among controllers (Christofides et al. (2013); Scattolini (2009)), thus increasing the information available to the local entities and allowing the coordination of their actions. When the inter-subsystems interactions are strong, the global performance of decentralized controllers is far from optimal and, hence, distributed architectures offer an efficient alternative to deal with the latter while reducing the computation burden entailed by the centralized formulation.

Several DMPC methods have been proposed for a wide range of linear and nonlinear processes (see Maestre et al. (2014)). Nevertheless, the development of distributed control schemes to deal with nonlinear control problems still demands additional research efforts, as there are few approaches capable of dealing with this class of problems. In Rocha et al. (2018); Rocha and Oliveira-Lopes (2016), a nonlinear system is controlled by a set of cooperative agents that use local linearized models. Also, in Dunbar (2007), the authors deal with a dynamically coupled nonlinear system where the local controllers minimize their own objective and iteratively exchange data. In Stewart et al. (2011), a distributed gradient projection algorithm is proposed to optimize the nonconvex objective function of the DMPC problem. A different alternative is proposed in Necoara et al. (2009), where the nonlinear optimization problem is reduced to a linear convex problem using sequential convex programming.

⋆ This work is partially supported by the Spanish Training programme for Academic Staff (FPU17/02653), the H2020 ADG-ERC project OCONTSOLAR (ID 789051), the MINECO-Spain project C3PO (DPI2017-86918-R), and the FEDER-Andalucía project GESVIP (US-1265917).

For many industrial processes, it is suitable to describe their dynamics by a static nonlinearity and a linear dynamic model. This representation characterizes the Hammerstein structure. In particular, Hammerstein models allow a representation of nonlinear systems through a series of interconnected blocks, such that, between the real system's input and output, a new virtual variable is defined, commonly called *internal* or *intermediate* variable. Hammerstein models have been successfully used to describe a variety of systems such as neutralization reactors (Ławryńczuk (2010, 2011)), solid oxide fuel cells (Huo et al. (2008)), or distillation columns (Marusak (2010)). Several works deal with the application of MPC to Hammerstein multiple inputs-multiple outputs (MIMO) models, either in a centralized or decentralized way. In this context, the block-oriented structure is exploited to generalize the good features of the linear MPC to nonlinear MPC problems by using the inverse of the nonlinear block or by linearizing it around steady-state values. See for example Zheng et al. (2014), where an adaptive decentralized MPC is established for a large-scale system presented by a set of small-scale Hammerstein models, and Ławryńczuk (2010), where a centralized MPC implementation.

In this work, we propose a distributed MPC method tailored for the Hammerstein model. In particular, we develop a DMPC approach for two independent agents coupled through their outputs and inputs, which cooperate to jointly minimize a global objective function. The control problem is posed in terms of the internal variable, from which the real inputs are then derived according to the Hammerstein model. In addition to the novel DMPC scheme, the paper also proposes a sub-optimality bound on the value attained for the overall cost function.

The rest of the paper is organized as follows: the Hammerstein model is described in Section II. The proposed distributed MPC for the Hammerstein model is presented in Section III. Simulation results are presented in Section IV. Finally, conclusions are given in Section V.

2. PROBLEM SETTING

In this section, we present the Hammerstein model and introduce the underlying assumptions on which our results are based.

2.1 Hammerstein model

The Hammerstein model allows a representation of nonlinear systems through a series of interconnected blocks. In particular, it consists of a dynamic linear block, connected in series to one or more static nonlinear blocks whose outputs are *virtual* variables of the system (see Fig.1).

Hereon, let $u_i(k) \in \mathbb{R}^{r_i}$ (for $i = 1, ..., n$) and $y_i(k) \in \mathbb{R}^{q_i}$ (for $i = 1, ..., m$) represent respectively inputs and outputs of the Hammerstein model at time instant k. Likewise, the intermediate variables will be denoted by $s_i(k) \in \mathbb{R}^{p_i}$ (with $i = 1, ..., n$). Also, let us use vectors $u(k) = [u_1(k), u_2(k), \dots u_n(k)]^T$, $s(k) = [s_1(k), s_2(k), \dots s_n(k)]^T$ and $y(k) = [y_1(k), y_2(k), \dots, y_m(k)]$, which aggregate accordingly these variables for the overall system.

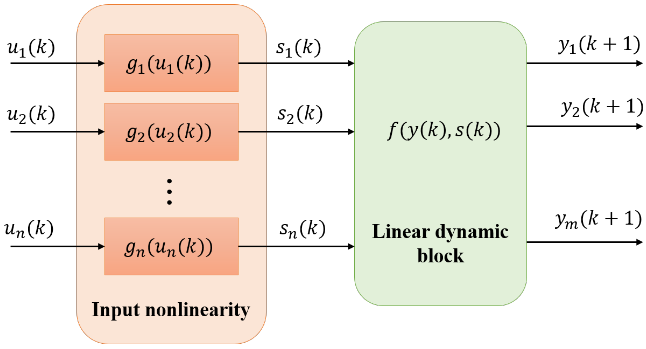

Fig. 1. Hammerstein model with decoupled nonlinearities

Following the Hammerstein structure, the outputs of the nonlinear sub-blocks, i.e., the intermediate variables, are related to real inputs as follows

$$s_i(k) = g_i(u_i(k)), \tag{1}$$

where $g_i(\cdot)$ is a non linear static function that maps input $u_i(k)$ to a cetain $s_i(k)$.

Assumption 1. In this work, we consider decoupled nonlinearities, i.e, each intermediate variable $s_i(k)$ is expressed as a continuous function of its input $u_i(k)$. In particular, we consider the following relation:

$$\begin{aligned} s_i(k) &= g_i(u_i(k)) = \\ & c_{i,1}u_i(k) + c_{i,2}u_i^2(k) + \cdots + c_{i,n}u_i^n(k), \end{aligned} \tag{2}$$

where $c_{i,j}$, with $j = 1, ..., n$, are real coefficients.

Assumption 2. The nonlinear map of the Hammerstein model is bijective, i.e, functions $g_i(u_i(k)) : \mathbb{R} \to \mathbb{R}$ have an inverse $g_i^{-1}(s_i(k)) : \mathbb{R}^{p_i} \to \mathbb{R}^{r_i}$. Hence,

$$u_i(k) = g_i^{-1}(s_i(k)). \tag{3}$$

Assumption 3. The input variables evolves in set $\mathcal{U}_i = [u_i^{\min}, u_i^{\max}]$. As consequence, the intermediate variables will take values in

$$\mathcal{S}_i = \left[s_i^{\min}, s_i^{\max}\right] = g_i\left(\left[u_i^{\min}, u_i^{\max}\right]\right) \tag{4}$$

where s_i^{min} and s_i^{max} are respectively the minimum and the maximum of function $g_i(u_i(k))$ over the input space.

Likewise, the outputs of the linear block are calculated by linear model:

$$y(k+1) = Ay(k) + Bs(k), \tag{5}$$

where A and B are respectively the output transition and the intermediate variable-to-output matrices, whose dimensions are $\sum_{i=1}^{m} q_i \times \sum_{i=1}^{m} q_i$ and $\sum_{i=1}^{m} q_i \times \sum_{i=1}^{n} p_i$.

2.2 Two subsystems model

In this work, the Hammerstein structure is exploited to develop a DMPC algorithm based on two coupled agents, such that:

$$\begin{aligned} y_1(k+1) &= A_{11}y_1(k) + B_{11}s_1(k) + A_{12}y_2(k) + B_{12}s_2(k), \\ y_2(k+1) &= A_{22}y_2(k) + B_{21}s_2(k) + A_{21}y_1(k) + B_{21}s_1(k), \end{aligned} \tag{6}$$

where $y_i(k)$ and $s_i(k)$ are respectively the output and virtual variable of subsystem $i = 1, 2$. Note that, by aggregating both relations in (6), the linear dynamic block can be modeled as (5), where global matrices A and B aggregate accordingly matrices A_{ij} and B_{ij} for $i, j = 1, 2$. Also, the input nonlinearity is modeled as (1), i.e.,

$$s_1(k) = g_1(u_1(k)) \quad \text{and} \quad s_2(k) = g_2(u_2(k)). \tag{7}$$

As a result, there exists a global nonlinear function $g(u(k))$ such that $s(k) = g(u(k))$, where $g(u(k))$ considers accordingly $g_1(u_1(k))$ and $g_2(u_2(k))$.

3. DMPC FOR HAMMERSTEIN MODELS

The proposed method is based on Maestre et al. (2011a) and Maestre et al. (2011b), where the authors propose a heuristic mechanism for fast decision-making in DMPC problems. In this section, we present the proposed DMPC algorithm and introduce an index to evaluate its performance.

3.1 Agents objective

Let $J_i(k) = J_i^y(k) + J_i^u(k)$ represent the local cost functions of each control agent i, where

$$J_i^y(k) = \sum_{t=1}^{N_p} \|y_i^{sp}(k+t) - y_i(k+t)\|^2_{Q_i} \tag{8}$$

and

$$J_i^u(k) = \sum_{t=0}^{N_u-1} \|\Delta u_i(k+t)\|^2_{R_i}, \tag{9}$$

Here, N_u and N_p are respectively the control and prediction horizons, Q_i and R_i are positive weighting matrices whose dimensions are $N_p \times N_p$ and $N_u \times N_u$, $y_i^{sp}(\cdot)$ represents the set-point of subsystem i. Additionally, the control inputs come into play in an incremental form, i.e.,

$$\Delta u_i(k) = u_i(k+t) - u_i(k+t-1), \tag{10}$$

and it is assumed that $\Delta u_i(k+t) = 0$ for $t \geq N_u$.

In this work, the agents cooperate to provide the lowest value of the global cost function $J(k) = J_1(k) + J_2(k)$, which considers in an equitable manner both local costs. Mathematically, the global problem can be posed as

$$\begin{aligned} \min_{\Delta \mathbf{u}_1(k), \Delta \mathbf{u}_2(k)} \quad & J(k) \\ \text{s.t.} \quad & (6), (10), (14), \\ & u_1(k+t) \in \mathcal{U}_1, \\ & u_2(k+t) \in \mathcal{U}_2, \\ & \forall t = 0, ..., N_\mathrm{p} - 1. \end{aligned} \tag{11}$$

where $\Delta \mathbf{u}_i$ represents the sequence of $\Delta u_i(k+t)$ for $t = 0$ to $t = N_\mathrm{p} - 1$.

Note that, given (6), eq. (8) can be rewritten as a function on current input $y(k)$ and sequence

$$\mathbf{s}(k) = [\, s(k) \;\; s(k+1) \;\; \cdots \;\; s(k+N_p-1)\,]^\mathrm{T}. \tag{12}$$

Likewise, given (3) and (10), eq. (9) can be expressed in terms of $\mathbf{s}_i(k)$ and $s_i(k-1)$, where

$$\mathbf{s}_i(k) = [\, s_i(k) \;\; s_i(k+1) \;\; \cdots \;\; s_i(k+N_p-1)\,]^\mathrm{T}. \tag{13}$$

The latter allows us to transform the nonlinear optimization problem (11) into a sum of a linear MPC problem, derived from (5), and a nonlinear one which is related to (1). In particular, function $J(\cdot)$ can be decomposed as follows

$$J(y(k), s(k-1), \mathbf{s}(k)) = J^y(y(k), \mathbf{s}(k)) + J^u(s(k-1), \mathbf{s}(k)), \tag{14}$$

where

$$J^y(y(k), \mathbf{s}(k)) = J_1^y(y(k), \mathbf{s}(k)) + J_2^y(y(k), \mathbf{s}(k)), \tag{15}$$

and

$$J^u(s(k-1), \mathbf{s}(k)) = J_1^u(s_1(k-1), \mathbf{s}_1(k)) + J_2^u(s_2(k-1), \mathbf{s}_2(k)). \tag{16}$$

Note that $J^y(\cdot)$ is a quadratic function on $\mathbf{s}(k)$ while $J^u(\cdot)$ is nonlinear since it deals with the inverse of the nonlinear block. Also, let us remark that the dependence on variable $\mathbf{s}(k)$ involves, by its definition, the dependence on $\mathbf{s}_1(k)$ and $\mathbf{s}_2(k)$.

3.2 DMPC algorithm

The proposed DMPC approach uses linear MPC on function $J^y(\cdot)$ to determine the first sequence of virtual variables. Next, a nonlinear optimization of $J^u(\cdot)$ is solved, which also provides a new solution for $\mathbf{s}(k)$. The information obtained in these two steps is exchanged and used to define a searching set that is explored by the two agents with the aim of finding the sequence $\mathbf{s}^*(k)$ that minimizes (14). Finally, inputs $u^*(k)$ are derived from $s^*(k)$.

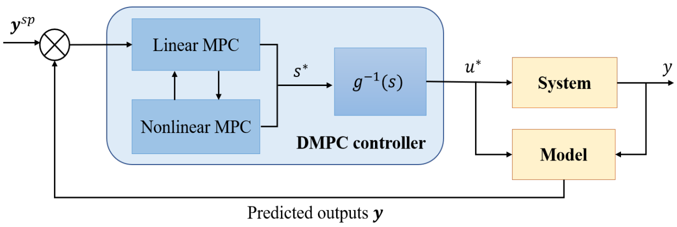

Fig. 2. DMPC for the Hammerstein model

The proposed DMPC algorithm for the Hammerstein model is described below.

(1) At time step k, measure the system output $y(k)$ and determine the optimal sequences of intermediate variables that minimizes (15), that is,

$$\begin{aligned} (\mathbf{s}_1^{y*}(k), \mathbf{s}_2^{y*}(k)) = \arg\min_{\mathbf{s}(k)} \quad & J^y(y(k), \mathbf{s}(k)) \\ \text{s.t.} \quad & (6), \\ & s_1(k+t) \in \mathcal{S}_1, \\ & s_2(k+t) \in \mathcal{S}_2, \\ & \forall t = 0, ..., N_\mathrm{p} - 1. \end{aligned} \tag{17}$$

Note that (17) is a convex optimization problem where the model is linear and the objective function is quadratic. Hence, it can be easily solved in a distributed fashion, e.g, by using dual decomposition.

(2) Optimize function (16) to determine sequences $\mathbf{s}_1^{u*}$ and $\mathbf{s}_2^{u*}$. To this end, each agent i solves

$$\begin{aligned} \mathbf{s}_i^{u*}(k) = \arg\min_{\mathbf{s}_i^u(k)} \quad & J_i^u(s_i(k-1), \mathbf{s}_i(k)) \\ \text{s.t.} \quad & s_i(k+t) \in \mathcal{S}_i, \\ & \forall t = 0, ..., N_\mathrm{p} - 1. \end{aligned} \tag{18}$$

where $s_i(k-1)$ is the solution at iteration $k-1$.

Remark 1. Problem (18) can be solved locally by each controller, i.e, the optimization of (9) is decoupled. Nevertheless, given the nonlinear relationship between $u_i(k)$ and $s_i(k)$, it is a nonlinear optimization problem.

(3) At this stage, the two agents exchange information to make a cooperative decision. Using $[\mathbf{s}_1^{y*}(k), \mathbf{s}_2^{y*}(k)]$ and $[\mathbf{s}_1^{u*}(k), \mathbf{s}_2^{u*}(k)]$ obtained in steps (1) and (2), we will determine in a distributed manner the sequence of intermediate variables that minimizes (14). In particular, the agents browse randomly set

$$\mathcal{S}^* = [\mathbf{s}_1^{u*}(k), \mathbf{s}_2^{u*}(k)] \times \left[\mathbf{s}_1^{y*}(k), \mathbf{s}_2^{y*}(k)\right], \quad (19)$$

where $\mathcal{S}^*$ is a subset of $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2$, and evaluate global index $J(\cdot)$ for different alternatives (see Table 1). After this exhaustive search, the sequence $\mathbf{s}^*(k) \in \mathcal{S}^*$ that provides a lowest value of (14) is taken as solution.

Remark 2. The extreme points of the searching set are also evaluated, as they are necessary to derive the sub-optimality bound that will be introduced later.

(4) Based on the values of $\mathbf{s}^*(k)$, which contains accordingly sequences $\mathbf{s}_1^*(k)$ and $\mathbf{s}_2^*(k)$, the agents use functions $g_1^{-1}(s_1(k))$ and $g_2^{-1}(s_2(k))$ to determine their control inputs. Only $u_1^*(k)$ and $u_2^*(k)$ are applied to the system and the whole procedure is repeated for time instant $k = k + 1$.

Remark 3. If the computation burden becomes an issue, it may be preferable to switch to a non-incremental formulation of the control problem, as it makes problem (18) not dependent on the time instant, so that they become static and can be simply solved off-line before starting the algorithm.

Table 1. Proposals evaluation for decision making [1].

	$(\mathbf{s}_1^{u*}, \mathbf{s}_2^{u*})$	$\dots$	$(\mathbf{s}_1^{y*}, \mathbf{s}_2^{y*})$
$(\mathbf{s}_1^{u*}, \mathbf{s}_2^{u*})$	$J_1(\mathbf{s}_1^{u*}, \mathbf{s}_2^{u*})$ $+J_2(\mathbf{s}_1^{u*}, \mathbf{s}_2^{u*})$	$\dots$	$J_1(\mathbf{s}_1^{y*}, \mathbf{s}_2^{y*})$ $+J_2(\mathbf{s}_1^{u*}, \mathbf{s}_2^{u*})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$(\mathbf{s}_1^{y*}, \mathbf{s}_2^{y*})$	$J_1(\mathbf{s}_1^{u*}, \mathbf{s}_2^{u*})$ $+J_2(\mathbf{s}_1^{y*}, \mathbf{s}_2^{y*})$	$\dots$	$J_1(\mathbf{s}_1^{y*}, \mathbf{s}_2^{y*})$ $+J_2(\mathbf{s}_1^{y*}, \mathbf{s}_2^{y*})$

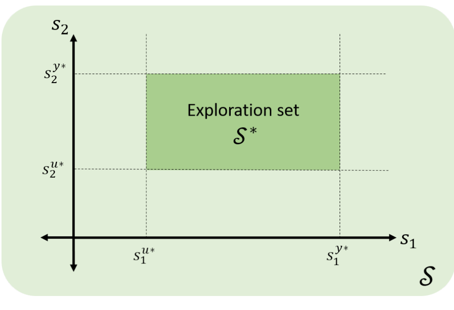

Fig. 3. Illustrative sketch of sets $\mathcal{S}$ and $\mathcal{S}^*$ for the case of $N_\mathrm{p} = 1$ and $s_1, s_2 \in \mathbb{R}^1$.

3.3 Performance evaluation

The nonlinear nature and the decomposition of the optimization problem lead to a loss of performance regarding the global criterion. To quantify the latter, we consider performance index $\rho(k)$, which is defined as follows:

$$\rho(k) = \frac{J(y(k), s(k-1), \mathbf{s}^*(k)) - J^*(k)}{J^*(k)}. \quad (20)$$

where $J(y(k), s(k-1), \mathbf{s}^*(k))$ is the cost obtained with solution $\mathbf{s}^*(k) = [\mathbf{s}_1^*(k), \mathbf{s}_2^*(k)]^\mathrm{T}$ provided by the algorithm proposed in Section II, and $J^*(k)$ is the minimum possible cost.

Note that $J^*(k)$ must satisfy

$$J^*(k) \geq J^u(s(k-1), \mathbf{s}^*(k)) + J^y(y(k), \mathbf{s}^*(k)). \quad (21)$$

For the sake of clarity, let us use

$$J^{u*}(k) = J^u(s(k-1), \mathbf{s}^*(k)) \text{ and } J^{y*}(k) = J^y(y(k), \mathbf{s}^*(k)). \quad (22)$$

Then, the difference between $J(y(k), s(k-1), \mathbf{s}^*(k))$ and $J^*(k)$ is unknown, but it is upper bounded by the difference between $J(y(k), s(k-1), \mathbf{s}^*(k))$ and $J^{y*}(k) + J^{u*}(k)$, i.e.,

$$J^*(k) - (J^{u*}(k) + J^{y*}(k)) \leq J(y(k), s(k-1), \mathbf{s}^*(k)) - (J^{u*}(k) + J^{y*}(k)) \quad (23)$$

The right hand side of the inequality above can be calculated during the execution of the algorithm in steps (1) and (2). Then, it is possible to compute an upper bound on the actual value of $\rho(k)$, such that

$$0 \leq \rho(k) \leq \rho_{\max}(k)$$

where

$$\rho_{\max}(k) = \frac{J(y(k), s(k-1), \mathbf{s}^*(k)) - (J^{y*}(k) + J^{u*}(k))}{J^{y*}(k) + J^{u*}(k)}.$$

4. SIMULATION RESULTS

In this section, we illustrate the results obtained with the proposed DMPC algorithm. To this end, we consider a Hammerstein model described by the following equations:

$$\begin{bmatrix} y_1(k+1) \\ y_2(k+1) \end{bmatrix} = A \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} + B \begin{bmatrix} s_1(k) \\ s_2(k) \end{bmatrix} \quad (24)$$

where A and B are defined as:

$$A = \begin{bmatrix} 0.025 & 0.1 \\ -0.19 & 0.05 \end{bmatrix}, \; B = \begin{bmatrix} 1.73 & -0.58 \\ -0.17 & 0.28 \end{bmatrix}, \quad (25)$$

and the output of the nonlinear block is given by

$$\begin{bmatrix} s_1(k) \\ s_2(k) \end{bmatrix} = \begin{bmatrix} u_1(k) + 0.5u_1(k)^2 \\ u_2(k) + 0.5u_2(k)^2 \end{bmatrix}. \quad (26)$$

The prediction and control horizons of the MPC controller are respectively $N_\mathrm{p} = 3$ and $N_\mathrm{u} = 1$, and the weighting matrices are $Q_i = 1$ and $R_i = 0.1$ for agents $i = 1, 2$. The control objective is to track an output reference while respecting constraints on the manipulated variable. In particular, the desired trajectories for y_1 and y_2 are shown in red dashed lines in Figs. 4 and 5, and the constraints on u are the following:

$$-10 \leq u_1(k) \leq 5 \text{ and } -10 \leq u_2(k) \leq 5, \; \forall k. \quad (27)$$

Note that function $g_i(u_i(k)) = u_i(k) + 0.5u_i(k)^2$ is a continuous function that decreases monotonically in interval $[-10, -1]$ and increases monotonically in interval $[-1, 5]$. Therefore, it is easy to determine a piecewise analytic expression of the inverse $g_i^{-1}(s_i(k))$.

[1] Time index k and the dependence on $s(k-1)$ and $y(k)$ of functions $J_1(\cdot)$ and $J_2(\cdot)$ have been omitted in Table 1 for the sake of clarity.

To assess the performance of our approach, the proposed DMPC algorithm is compared with nonlinear centralized MPC. To solve the corresponding optimization problem, we use MATLAB Optimization Toolbox, and in particular, the genetic algorithm solver. Figs. 4 to 7 show the output tracking and the evolution of the control inputs for both approaches. It can be observed that the DMPC algorithm provides similar results to the centralized approach in terms of outputs' behavior. However, a notable difference is present in the evolution of the control signals, which reflects the different nature of the optimization problems that are respectively solved to find u_1 and u_2.

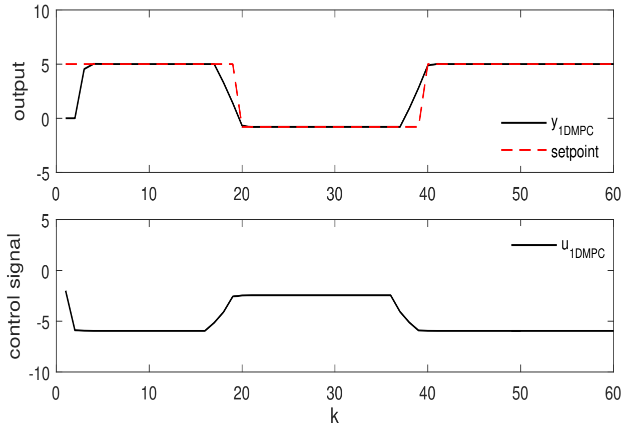

Fig. 4. Output tracking of y_1 for the proposed DMPC

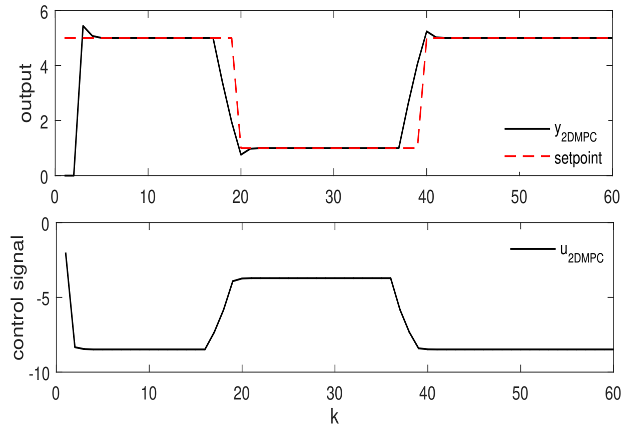

Fig. 5. Output tracking of y_2 for the proposed DMPC

Additionally, in Fig. 8, we show the evolution of the cost function $J(\cdot)$. It can be seen that, especially at the middle of the simulation, the cost provided by nonlinear centralized MPC is lower than the one obtained with the DMPC algorithm. Note that there some time instants where the DMPC approach seems to outperform the solution of nonlinear centralized MPC, however, the latter is due to a lack of accuracy of the solution provided by the genetic algorithm. Also, the evolution of performance index $\rho(\cdot)$ is illustrated in Figure 9. Note that when $\rho(\cdot)$ is approximately 0, then the solution found with the DMPC method is nearly optimal. Hence, this index can be used in real time as a criterion to keep on searching for better results or terminate the optimization. Finally, Fig. 10

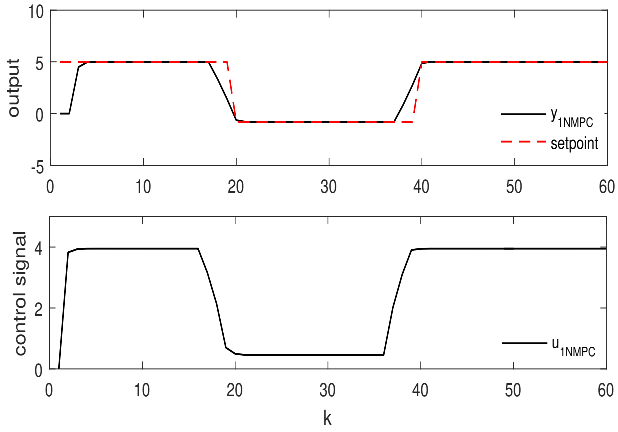

Fig. 6. Output tracking of y_1 for the centralized MPC

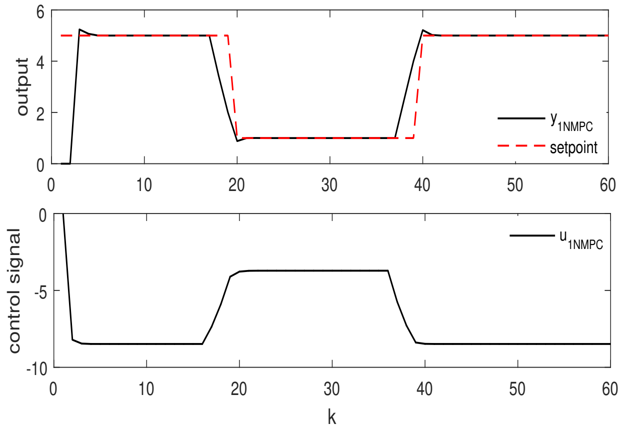

Fig. 7. Output tracking of y_2 for centralized MPC

illustrate the benefits of the DMPC approach in terms of computation time.

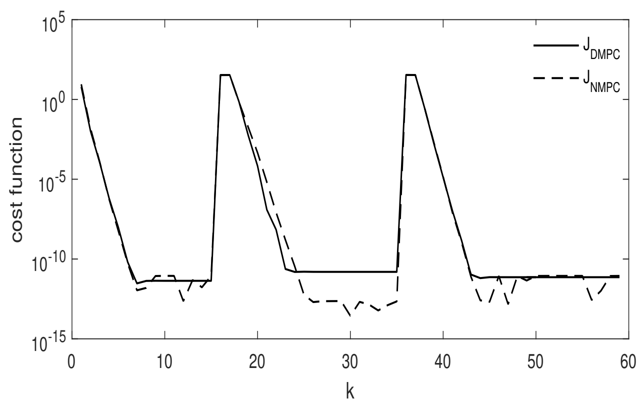

Fig. 8. Cost functions for the centralized and distributed MPC

5. CONCLUSION

A DMPC approach for the class of block-oriented models composed of a nonlinear steady-state block and a linear dynamic block is presented in this work. In particular, the proposed DMPC algorithm uses the Hammerstein model and formulates the control problem in terms of its internal variable. The main advantage of this idea is to

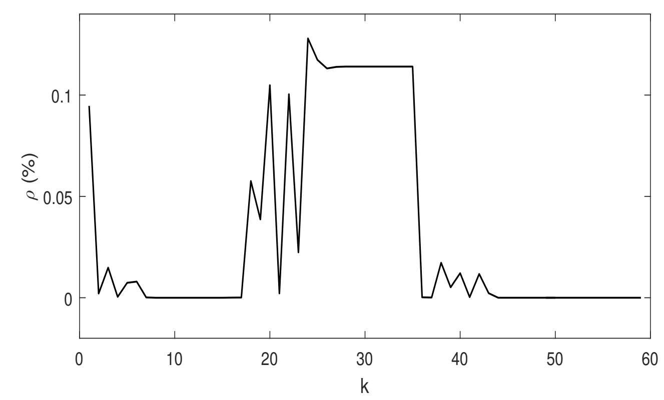

Fig. 9. Loss of performance

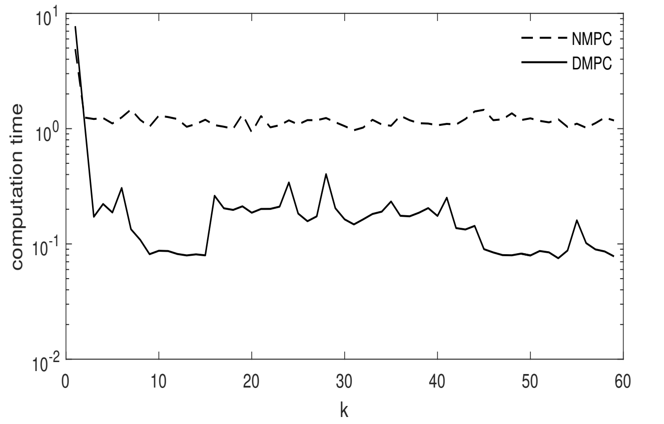

Fig. 10. Computation time for the proposed and the centralized MPC

exploit the linear features of the dynamic block instead of considering a single nonlinear optimization problem. The final step of the proposed DMPC algorithm involves a coordinated search through the space of overall solutions. However, the variables obtained in a distributed fashion are used to shrink the search space and thus to reduce the computational time. Also, a criterion for evalutating the loss of performance in comparison to nonlinear centralized MPC is provided. The simulation results show a level of performance close to the optimum and a notable decrease of the computational burden.

Future work will extend these results to consider nonlinear systems with a greater number of agents and a more realistic scenario. Additionally, the case of coupled nonlinearities will be studied.

REFERENCES

Camacho, E.F. and Bordons, C. (2013). *Model predictive control*. Springer Science & Business Media.

Christofides, P.D., Scattolini, R., de la Pena, D.M., and Liu, J. (2013). Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, 51, 21–41.

Dunbar, W.B. (2007). Distributed receding horizon control of dynamically coupled nonlinear systems. *IEEE Transactions on Automatic Control*, 52(7), 1249–1263.

Garcia, C.E., Prett, D.M., and Morari, M. (1989). Model predictive control: theory and practice—a survey. *Automatica*, 25(3), 335–348.

Huo, H.B., Zhu, X.J., Hu, W.Q., Tu, H.Y., Li, J., and Yang, J. (2008). Nonlinear model predictive control of sofc based on a hammerstein model. *Journal of Power Sources*, 185(1), 338–344.

Ławryńczuk, M. (2010). Suboptimal nonlinear predictive control based on multivariable neural hammerstein models. *Applied Intelligence*, 32(2), 173–192.

Ławryńczuk, M. (2011). On-line set-point optimisation and predictive control using neural hammerstein models. *Chemical Engineering Journal*, 166(1), 269–287.

Maestre, J., De La Pena, D.M., Camacho, E., and Alamo, T. (2011a). Distributed model predictive control based on agent negotiation. *Journal of Process Control*, 21(5), 685–697.

Maestre, J., Munoz De La Pena, D., and Camacho, E. (2011b). Distributed model predictive control based on a cooperative game. *Optimal Control Applications and Methods*, 32(2), 153–176.

Maestre, J.M., Negenborn, R.R., et al. (2014). *Distributed model predictive control made easy*, volume 69. Springer.

Marusak, P.M. (2010). Efficient predictive control algorithm based on fuzzy hammerstein models: A case study. In *Soft Computing in Industrial Applications*, 11–20. Springer.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.

Necoara, I., Savorgnan, C., Tran, D.Q., Suykens, J., and Diehl, M. (2009). Distributed nonlinear optimal control using sequential convex programming and smoothing techniques. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 543–548. IEEE.

Rocha, R.R. and Oliveira-Lopes, L.C. (2016). A cooperative distributed model predictive control for nonlinear systems with automatic partitioning. In *Computer Aided Chemical Engineering*, volume 38, 2205–2210. Elsevier.

Rocha, R.R., Oliveira-Lopes, L.C., and Christofides, P.D. (2018). Partitioning for distributed model predictive control of nonlinear processes. *Chemical Engineering Research and Design*, 139, 116–135.

Scattolini, R. (2009). Architectures for distributed and hierarchical model predictive control–a review. *Journal of process control*, 19(5), 723–731.

Siljak, D.D. (2011). *Decentralized control of complex systems*. Courier Corporation.

Stewart, B.T., Wright, S.J., and Rawlings, J.B. (2011). Cooperative distributed model predictive control for nonlinear systems. *Journal of Process Control*, 21(5), 698–704.

Zheng, L., Xiao-qi, P., and Yan-po, S. (2014). Modelling and decentralized adaptive model predictive control for alumina trihydrate precipitation process. *IFAC Proceedings Volumes*, 47(3), 8534–8539.