Preprint, 11th IFAC Symposium on Dynamics and Control of Process Systems,
including Biosystems
June 6-8, 2016. NTNU, Trondheim, Norway

Reference Tracking using a Non-Cooperative Distributed Model Predictive Control Algorithm

Anca Maxim *,** **Clara M. Ionescu** **
Constantin F. Caruntu * **Corneliu Lazar** * **Robin De Keyser** **

* *Department of Automatic Control and Applied Informatics, Gheorghe Asachi Technical University of Iasi, Blvd. Mangeron 27, 700050 Iasi, Romania*
(e-mails: {anca.maxim, caruntuc, clazar}@ac.tuiasi.ro)
** *Department of Electrical energy, Systems and Automation, Ghent University, Technologiepark 914, Gent B9052, Belgium*
(e-mails: {claramihaela.ionescu, robain.dekeyser}@ugent.be)

Abstract: In this paper, a non-cooperative distributed model predictive control (DMPC) algorithm for tracking constant references is developed and evaluated. As such, an augmented model is employed (i.e. the control loop is embedded with integrators) and the augmented state contains the state increments and the error between the reference and the predicted output. The algorithm is tested in real life experiments on the quadruple tank process with non-minimum phase behaviour. The experimental results show acceptable performance index for the DMPC method when compared with the centralized approach.

Keywords: non-cooperative DMPC, reference tracking, two agent systems, MIMO systems, non-minimum phase systems

1. INTRODUCTION

In process industry, the distributed model predictive control (DMPC) methodology is an alternative control choice when trying to avoid the burden of the centralized control strategy (Venkat et al., 2007). According to (Christofides et al., 2013), the DMPC methods can be divided into: i) *cooperative* DMPC, in which each local controller optimizes a global cost function or ii) $non-cooperative$ DMPC, if a local cost function is minimized. Furthermore, in (Scattolini, 2009) the DMPC algorithms are classified depending on the topology of the communication network into: *fully connected* or *partially connected* algorithms. From the point of view of the amount of information exchanged between the local controllers, the algorithms can be: i) $non-iterative$, if the information is exchanged only once per sampling period or ii) *iterative*, if the local controllers exchange information more than once within one sampling period.

The offset-free tracking of a reference signal is one of the most common control requirements. In (Maeder and Morari, 2007; Faanes and Skogestad, 2005; Borrelli and Morari, 2007) a centralized method for tracking constant references is presented. The idea is to use an integrator as the disturbance model to account for disturbances and plant-model mismatch. Furthermore, the steady-state values for the states and the given disturbance are used to ensure an offset-free tracking. This approach, although efficient, requires a centralized model to compute the steady-state values and is not available in the distributed MPC context.

In (Betti, 2013), there are several solutions for the offset-free tracking problem, all using an integrator in the control loop. The solutions are based on: i) introducing an artificial disturbance which will be estimated together with the model states (Pannocchia and Rawlings, 2003) ii) using an integrator as an internal model for the reference and being directly introduced in the control loop (Magni and Scattolini, 2007) or iii) using a *velocity-form model* in which the augmented state is composed from the state increments and the error (Wang, 2009).

The method proposed in this paper is a sequel of the work presented in (Maxim et al., 2015) which introduced a non-cooperative DMPC strategy for two agent systems. The regulatory problem (i.e. constant reference tracking) was solved by means of a centralized approach. That consisted in using the centralized model to obtain the steady-state values for the states and inputs imposing the outputs steady-state values. This method, although efficient in simulation, starts from the premises that the model is identical with the system, which is not the case in real life experiments.

Thus, in this paper another non-cooperative DMPC approach, suitable for tracking constant output references is suggested. The idea emerged from the centralized MPC methodology given in (Wang, 2009), where a 'velocity-form' model is employed introducing an artificial integrator in the system. In this way, the modelling errors will cease to influence the closed-loop results, yielding a zero steady state error in the real-time experiment. The performance of the method is tested in a reference tracking experiment on the quadruple tank process from Quanser.

The paper is organized as follows. The next section describes the proposed state-space DMPC strategy. The third section presents the quadruple-tank system (Quanser) and the experimental conditions, along with the real-life comparative test. A conclusion section summarizes the main outcome of this paper.

2. DISTRIBUTED MODEL-BASED PREDICTIVE CONTROL ALGORITHM

In this section, the non-cooperative DMPC formulation is given. To this end, the preliminaries are first settled (i.e. the process model is written in a *velocity form* using state and input increments). Afterwards, the predictor formulation is derived and the optimal solution is obtained.

2.1 Plant 'velocity-form' model for predictive controller design

Consider a linear-time-invariant system composed of two non-overlapping sub-systems, dynamically coupled through inputs and subject to state and control constraints. Each sub-system is described by the following dynamics:

$$\begin{aligned} x_{pi}(k+1) &= A_{pi}x_{pi}(k) + B_{pii}u_i(k) + B_{pij}u_{ni}(k) \\ y_i(k) &= C_{pi}x_{pi}(k) \end{aligned} \tag{1}$$

where k is the time instant, $u_{pi} \in R^{m_i}$ and $y_i \in R^{q_i}$ are the input/output variables of the sub-systems and $x_{pi} \in R^{n_i}$ are the states of the sub-systems, with $i = \overline{1,2},\ j = \overline{1,2},\ i \neq j$. The subscript ni denotes the neighboring agent of agent i (i.e. $u_{n1} = u_2$ and $u_{n2} = u_1$). The following linear constraints in the state and inputs are considered:

$$x_{pi} \in X_i,\ u_i \in U_i,\ i = \overline{1,2} \tag{2}$$

where X_i and U_i with $i = \overline{1,2}$ are defined by a set of linear inequalities.

Aiming for implementing an offset-free constant reference tracking algorithm, an integrator, which is an internal model of constant reference, is included in the control structure similar to the Internal Model Control principle. The integral action is applied to the control increments obtained at the controller output.

Using the methodology described in (Wang, 2009), the difference operation is applied on both sides of (1) resulting:

$$\begin{aligned} \Delta x_{pi}(k+1) = A_{pi}\Delta x_{pi}(k) + B_{pii}\Delta u_i(k) + B_{pij}\Delta u_{ni}(k), \\ \Delta y_i(k+1) = \\ C_{pi}A_{pi}\Delta x_{pi}(k) + C_{pi}B_{pii}\Delta u_i(k) + C_{pi}B_{pij}\Delta u_{ni}(k) \end{aligned} \tag{3}$$

with $i = \overline{1,2},\ j = \overline{1,2},\ i \neq j$.
The model (3) introduces the increments of the variables x_{pi}, u_i, u_{ni} and y_i. A new state variable $x_i(k) = \left[\Delta x_{pi}(k)^T y_i(k)\right]^T$, $i = \overline{1,2}$ is introduced for each sub-system, resulting the augmented model:

$$\underbrace{\begin{bmatrix} \Delta x_{pi}(k+1) \\ y_i(k+1) \end{bmatrix}}_{x_i(k+1)} = \underbrace{\begin{bmatrix} A_{pi} & O^T_{q_i \times n_i} \\ C_{pi}A_{pi} & 1 \end{bmatrix}}_{A_i} \underbrace{\begin{bmatrix} \Delta x_{pi}(k) \\ y_i(k) \end{bmatrix}}_{x_i(k)} + \underbrace{\begin{bmatrix} B_{pii} \\ C_{pi}B_{pii} \end{bmatrix}}_{B_{ii}} \Delta u_i(k) + \underbrace{\begin{bmatrix} B_{pij} \\ C_{pi}B_{pij} \end{bmatrix}}_{B_{ij}} \Delta u_{ni}(k)$$

$$y_i(k) = \underbrace{\begin{bmatrix} O_{q_i \times n_i} & I_{q_i} \end{bmatrix}}_{C_i} \begin{bmatrix} \Delta x_{pi}(k) \\ y_i(k) \end{bmatrix} \tag{4}$$

which will be used to design the predictive controller. Note that the new inputs of the state-space model in *velocity-form* are $\Delta u_i(k)$ and $\Delta u_{ni}(k)$.
The superscript notation T defines the transpose operation, I is the identity matrix and O is a zero matrix. The A_i, B_{ii}, B_{ij} and C_i with $i = \overline{1,2},\ j = \overline{1,2},\ i \neq j$ are matrices with adequate dimensions of the discrete-time sub-systems model in *velocity-form*. The model (4) can be written in a compact form as:

$$\begin{cases} x_i(k+1) = A_i x_i(k) + B_{ii}\Delta u_i(k) + B_{ij}\Delta u_{ni}(k) \\ y_i(k) = C_i x_i(k) \end{cases} \tag{5}$$

with $i = \overline{1,2},\ j = \overline{1,2},\ i \neq j$.

2.2 Non-Cooperative DMPC optimization formulation

For a linear-time-invariant system composed of two sub-systems described by equation (1), a non-cooperative DMPC architecture is defined with two agents communicating between them. Each sub-system i is controlled by a local agent (a predictive controller) which has available at any time instant k the information regarding its own sub-system.

Based on the *velocity-form* model of sub-system i (5) and its corresponding state x_i, the control variable Δu_i is computed. At each sampling time, each agent solves a reduced dimension optimization problem assuming a given fixed input trajectory for its neighbour. The control objective of each agent is to regulate the sub-system to a constant reference while guaranteeing that the constraints are satisfied.

Using the sub-systems *velocity-form* model (5) for designing the predictive controllers, the integral action is naturally embedded in the control algorithms. Thus, the constant reference tracking is obtained for two agent systems.

The main idea is that each sub-system solves a MPC optimization problem considering that its dynamics is given by (5) and the neighbour future input sequence $\Delta u_{ni}(k+l)$ of sub-system i is known in advance over the control horizon N_c, $l = \overline{0, N_c - 1}$. Hence, the only information that agent i must receive from its neighbour ni at time k is $\Delta u_{ni}(k+l)$. In order to obtain a computationally efficient non-cooperative DMPC algorithm, we constructed the neighbour future input sequence at time instant k as in (Maxim et al., 2015) using the last $N_c - 1$ elements of the precedent optimal control sequence computed at instant $k-1$. To preserve the same length for the input vectors, the last element from the vector is doubled. This is similar with an one unit shorter control horizon for the neighbour (i.e. $N_c - 1$ instead of N_c).

$$\Delta U_{ni}(k) = \begin{bmatrix} \Delta u^*_{ni}(k|k-1) \\ \vdots \\ \Delta u^*_{ni}(k+N_c-2|k-1) \\ \Delta u^*_{ni}(k+N_c-2|k-1) \end{bmatrix} \quad (6)$$

Therefore, at time $k-1$ each agent $i, i=\overline{1,2}$ transmits to the other the optimal sequence $\Delta U^*_i(k-1)$ which is used at time k to create $\Delta U_{ni}(k)$ (i.e. equation (6)).

For each agent i, a local cost function is defined as:

$$J_i(x_i(k), \Delta U_i(k), \Delta U_{ni}(k)) = (R_{si} - Y_i)^T (R_{si} - Y_i) + \Delta U_i(k)^T R_i \Delta U_i(k) \quad (7)$$

based on the future output predictor:

$$Y_i = [y_i(k+1|k) \ \ldots \ y_i(k+N_p|k)]^T$$

the own future input sequence

$$\Delta U_i(k) = [\Delta u_i(k|k) \ \ldots \ \Delta u_i(k+N_c-1|k)]^T$$

with $i = \overline{1,2}$, N_p the prediction horizon and N_c the control horizon ($N_c \leq N_p$). In this paper, $N_c = N_p$ and the predicted reference trajectory $R_{si} \in R^{N_p}$, $i = \overline{1,2}$ is assumed constant and equal with the setpoint at time instant k. The weight matrices have the form $R_i = \alpha I_{N_c}$, $i = \overline{1,2}$, $\alpha \geqslant 0$.

Using the *velocity-form* model (5), the prediction of state and input variables are computed:

$$x_i(k+l|k) = A_i^l x_i(k) + A_i^{l-1} B_{ii} \Delta u_i(k|k) + \ldots + A_i^{Np-l} B_{ii} \Delta ui(k+l-1|k) + A_i^{l-1} B_{ij} \Delta u_{ni}(k|k) + \ldots + A_i^{Np-l} B_{ij} \Delta u_{ni}(k+l-1|k) \qquad l = \overline{1, N_p} \quad (8)$$

and using the output equation given in (5) the output predictor is achieved:

$$y_i(k+l|k) = C_i A_i^l x_i(k) + C_i A_i^{l-1} B_{ii} \Delta u_i(k|k) + \ldots + C_i A_i^{Np-l} B_{ii} \Delta ui(k+l-1|k) + C_i A_i^{l-1} B_{ij} \Delta u_{ni}(k|k) + \ldots + C_i A_i^{Np-l} B_{ij} \Delta u_{ni}(k+l-1|k) \qquad l = \overline{1, N_p} \quad (9)$$

Note that the cost function (7) depends on the current measured state $x_i(k)$ and on the neighbour future input sequence $\Delta U_{ni}(k)$ because the output predictor Y_i, $i = \overline{1,2}$ is computed in the following matrix form using (9):

$$Y_i = \tilde{A}_i x_i(k) + \tilde{B}_{ii} \Delta U_i(k) + \tilde{B}_{ij} \Delta U_{ni}(k) \qquad i = \overline{1,2}, \ j = \overline{1,2}, \ i \neq j \quad (10)$$

where

$$\tilde{A}_i = \begin{bmatrix} C_i A_i & C_i A_i^2 & \ldots & C_i A_i^{N_p} \end{bmatrix}^T$$

$$\tilde{B}_{ii} = \begin{bmatrix} C_i B_{ii} & 0 & \ldots & 0 \\ C_i A_i B_{ii} & C_i B_{ii} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ C_i A_i^{N_p-1} B_{ii} & C_i A_i^{N_p-2} B_{ii} & \ldots & C_i A_i^{N_p-N_c} B_{ii} \end{bmatrix}$$

and B_{ij} is obtained similar with B_{ii}.

After the substitution of (10) into the cost function (7) and some matrix manipulation, the following explicit form for the cost function is obtained:

$$\begin{aligned} J_i(x_i(k), \Delta U_i, \Delta U_{ni}) &= \\ (R_{si} - \tilde{A}_i x_i(k))^T (R_{si} - \tilde{A}_i x_i(k)) &+ 2\Delta U_i^T \tilde{B}_{ii}^T \tilde{B}_{ij} \Delta U_{ni} \\ -2\Delta U_i^T \tilde{B}_{ii}^T [R_{si} - \tilde{A}_i x_i(k)] &- 2\Delta U_{ni}^T \tilde{B}_{ij}^T [R_{si} - \tilde{A}_i x_i(k)] \\ +2\Delta U_i^T (\tilde{B}_{ii}^T \tilde{B}_{ii} + R_i) \Delta U_i &+ \Delta U_{ni}^T (\tilde{B}_{ij}^T \tilde{B}_{ij}) \Delta U_{ni} \end{aligned} \quad (11)$$

with $i = \overline{1,2}, \ j = \overline{1,2}, \ i \neq j$.
As mentioned before, in (11) the free variable is $\Delta U_i(k), i = \overline{1,2}$. If there are no constraints imposed on the control variable or the outputs, the optimal solution is obtained having the first derivative of (11) equal to zero yielding:

$$\begin{aligned} \Delta U_i^*(k) = (\tilde{B}_{ii}^T \tilde{B}_{ii} + R_i)^{-1} \tilde{B}_{ii}^T [R_{si} - \tilde{A}_i x_i(k)] \\ -(\tilde{B}_{ii}^T \tilde{B}_{ii} + R_i)^{-1} \tilde{B}_{ii}^T \tilde{B}_{ij} \Delta U_{ni}, \end{aligned} \quad (12)$$

with $i = \overline{1,2}, \ j = \overline{1,2}, \ i \neq j$.

On the other hand, dealing with constraints directly in the optimization problem is one of the MPC features. The constraints are imposed on the control variables incremental variation:

$$\Delta U_i^{min} \leq \Delta U_i(k) \leq \Delta U_i^{max} \quad (13)$$

on the amplitude of the control variables:

$$U_i^{min} \leq U_i(k) \leq U_i^{max} \quad (14)$$

and on the outputs:

$$Y_i^{min} \leq Y_i(k) \leq Y_i^{max} \quad (15)$$

with $i = \overline{1,2}$, where $U_i^{min} = [u_i^{min} \ldots \ u_i^{min}]^T$ is a column vector with N_c elements which contains the lower limit imposed on each input element in the prediction window. Similar definition are used for U_i^{max},ΔU_i^{min},ΔU_i^{max},Y_i^{min}, Y_i^{max}.

According to the quadratic programming formulation, (14) is split into two inequalities:

$$\begin{aligned} -U_i(k) &\leq -U_i^{min} \\ U_i(k) &\leq U_i^{max} \end{aligned} \quad (16)$$

with $i = \overline{1,2}$. The same quadratic notation is also used for (13) and (15).

Next, the control variable constraints are parametrized in the same parameter vector ΔU_i, $i = \overline{1,2}$ obtaining:

$$\underbrace{\begin{bmatrix} u_i(k|k) \\ u_i(k+1|k) \\ \vdots \\ u_i(k+N_c-1|k) \end{bmatrix}}_{U_i(k)} = \underbrace{\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}}_{M_{ij}} u_i(k-1) + \underbrace{\begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 1 & 1 & 0 & \ldots & 0 \\ \vdots \\ 1 & 1 & 1 & \ldots & 1 \end{bmatrix}}_{M_{ii}} \underbrace{\begin{bmatrix} \Delta u_i(k|k) \\ \Delta u_i(k+1|k) \\ \vdots \\ \Delta u_i(k+N_c-1|k) \end{bmatrix}}_{\Delta U_i} \quad (17)$$

$i = \overline{1,2},\ j = \overline{1,2},\ i \neq j.$
Substituting (17) in (16) a compact form is obtained:

$$-(M_{ij}u_i(k-1) + M_{ii}\Delta U_i(k)) \leq -U_i^{min}$$
$$(M_{ij}u_i(k-1) + M_{ii}\Delta U_i(k)) \leq U_i^{max} \qquad (18)$$

The output constraints are also parametrized using the definition given in (10) obtaining:

$$-(\tilde{A}_i x_i(k) + \tilde{B}_{ii}\Delta U_i(k)) \leq -Y_i^{min} + \tilde{B}_{ij}\Delta U_{ni}(k)$$
$$(\tilde{A}_i x_i(k) + \tilde{B}_{ii}\Delta U_i(k)) \leq Y_i^{max} - \tilde{B}_{ij}\Delta U_{ni}(k) \qquad (19)$$

where $i = \overline{1,2},\ j = \overline{1,2},\ i \neq j.$
All the constraints (13),(18),(19) can be written in a matrix form as:

$$\begin{bmatrix} -I \\ I \\ -M_{ii} \\ M_{ii} \\ -\tilde{B}_{ii} \\ \tilde{B}_{ii} \end{bmatrix} \Delta U_i = \begin{bmatrix} -\Delta U_i^{min} \\ \Delta U_i^{max} \\ -U_i^{min} + M_{ij}u_i(k-1) \\ U_i^{max} - M_{ij}u_i(k-1) \\ -Y_i^{min} + \tilde{A}_i x_i(k) + \tilde{B}_{ij}\Delta U_{ni} \\ Y_i^{max} - \tilde{A}_i x_i(k) - \tilde{B}_{ij}\Delta U_{ni} \end{bmatrix} \qquad (20)$$

with $i = \overline{1,2},\ j = \overline{1,2},\ i \neq j$ and I the identity matrix. Finally, the constrained optimization problem is formulated as: find the minimum of the cost function (11) subject to (20). The proposed non-cooperative DMPC algorithm for tracking is described as follows:

- Step 1: *Initialization*: the states $x_{pi}(k)$ are measured and the augmented states $x_i(k)$ are computed. Then, using the optimal solution from time $k-1$ (i.e. $\Delta U_{ni}^*(k-1)$), the neighbour future input sequence $\Delta U_{ni}(k)$ is created;
- Step 2: *Optimization*: solve the optimal control problem:
$$\min_{\Delta U_i(k)} J_i$$
with J_i defined in equation (11) subject to the constraints (20);
- Step 3: *Implementation*: apply to the sub-system control signal using the first element of the integrated optimal solution (i.e. $u(k) = u(k-1) + \Delta u_i^*(k|k)$)
- Step 4: *Communication*: send out the optimal solution found at time k (i.e. $\Delta U_i^*(k)$) to the other agent, increment the current time to $k+1$ and return to Step 1 at the next sampling time.

To conclude, each agent $i, i = \overline{1,2}$ solves a local optimization problem using: i) the current state measurement $x_{pi}(k)$ and ii) the optimal input strategy $\Delta U_{ni}^*(k-1)$ received at the previous sampling instant from the neighbour agent to whom it is connected.

3. EXPERIMENTAL TESTS

Of all applications existing in literature, the four-tank system seems to be a representative proof-of-concept benchmark used to illustrate the efficacy of the proposed MPC and DMPC algorithms. The seminal work of Johansson (Johansson, 2000) initiated a series of system variations and MPC algorithms (Gatzke et al., 2000; Mercangoz and Doyle III, 2007; Alvarado et al., 2011) and robust control strategies (Vadigepalli et al., 2001). A comparative study of DMPC strategies is given in (Alvarado et al., 2011).

3.1 Process description

In this section the quadruple tank system from Quanser depicted in Fig. 1 is described. The control objective is to regulate the level of the water in the lower tanks (L_2, L_4) by manipulating the water flows (i.e. the voltages of the two pumps V_{p1}, V_{p2}). There is a strong coupling effect between the inputs and the outputs, (e.g. in Tank 2 there are two inputs: the flow from Pump 2 through Out 2, marked with red line, and the flow from Pump 1 through Out 1, denoted with green line, that is the output flow from Tank 1).

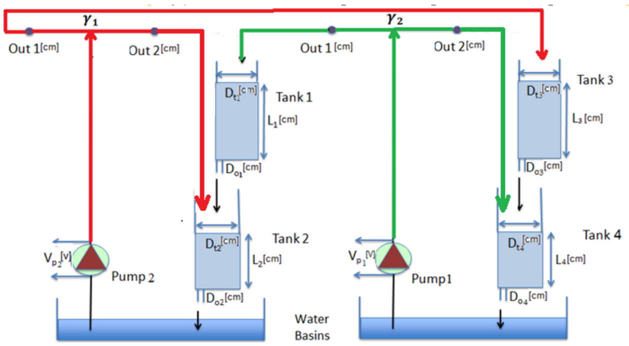

Fig. 1. Schematic diagram of the quadruple tank process from Quanser with non-minimum phase dynamics.

In Tank 2 there is a greater flow coming from Pump 1, via Tank 1, than the flow coming directly from Pump 2. This is due to the fact that the outlet diameter Out 1 is bigger than the diameter Out 2, while the outgoing orifices from each tank D_{o_i}, $i = 1...4$ have all the same diameter. The same conditions applies for Tank 4.

The state-space linearised mathematical model is obtained from the non-linear model derived in (Johansson, 2000) using the setup parameters (i.e. the cross-sections of the tanks) provided by Quanser. The model has been rewritten with the tanks notations used in Fig. 1:

$$\begin{aligned} x_p(k+1) &= Ax_p(k) + Bu(k) \\ y(k) &= Cx_p(k) \end{aligned} \qquad (21)$$

with

$$A = \begin{bmatrix} 0.69 & 0 & 0 & 0 \\ 0.24 & 0.89 & 0 & 0 \\ 0 & 0 & 0.29 & 0 \\ 0 & 0 & 0.11 & 0.92 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0.08 \\ 0.13 & 0.01 \\ 0.18 & 0 \\ 0.02 & 0.09 \end{bmatrix} \quad C = \begin{bmatrix} 0 & 1.05 & 0 & 0 \\ 0 & 0 & 0 & 0.75 \end{bmatrix} \qquad (22)$$

where $x_p = [x_1\ x_2\ x_3\ x_4]^T$, $u = [u_1\ u_2]^T$ and $y = [x_2\ x_4]^T$ are the state, input and output vector respectively. The numerical values in (22) were obtained for the sampling period $T_s = 1s$.

Starting from (21), the *velocity-form* centralized model is obtained, using the same procedure described in (3-5) with the difference that the input vectors are not split. After that, the centralized cost function is defined like in (7) and by minimizing it, the unconstrained explicit solution is obtained.

For the distributed control purpose, the quadruple tank system model (21) can be decomposed into two sub-systems coupled through the inputs (i.e. sub-system #1 composed of Tank 1 and Tank 2 and sub-system #2 obtained from Tank 3 and Tank 4). Each sub-system has the following states and outputs variables:

$$\begin{cases} x_{p1} = [x_1 \; x_2]^T \\ y_1 = x_2 \end{cases} \begin{cases} x_{p2} = [x_3 \; x_4]^T \\ y_2 = x_4 \end{cases} \tag{23}$$

where x_i, $i = \overline{1,4}$ are the states corresponding with the levels in each tank.

The sub-systems are interconnected through the inputs because each pump feeds water in both sub-systems depending on the valves configuration (i.e. parameters γ_1, γ_2). For example, if γ_1=0.25 than one quarter of Pump 2 water flow is fed into Tank 2 through Out 2 and three quarters supply Tank 3 through Out 1. The current setup exhibits non-minimum phase behaviour (i.e Out 2 flow<Out 1 flow).

The numerical values for the *velocity-form* sub-system models given in (5) are:

$$\begin{cases} A_1 = \begin{bmatrix} 0.69 & 0 & 0 \\ 0.24 & 0.89 & 0 \\ 0.25 & 0.94 & 1 \end{bmatrix} B_{11} = \begin{bmatrix} 0 \\ 0.13 \\ 0.14 \end{bmatrix} \\ B_{12} = \begin{bmatrix} 0.08 \\ 0.01 \\ 0.01 \end{bmatrix} C_1 = [\,0 \; 0 \; 1\,] \\ A_2 = \begin{bmatrix} 0.29 & 0 & 0 \\ 0.11 & 0.92 & 0 \\ 0.08 & 0.69 & 1 \end{bmatrix} B_{21} = \begin{bmatrix} 0.18 \\ 0.02 \\ 0.01 \end{bmatrix} \\ B_{22} = \begin{bmatrix} 0 \\ 0.09 \\ 0.07 \end{bmatrix} C_2 = [\,0 \; 0 \; 1\,] \end{cases} \tag{24}$$

In the next section the results of the non-cooperative DMPC algorithm presented in subsection 2.2 are given.

3.2 Results

In this subsection the experimental results obtained on the real life four-tank system from Quanser with non-minimum phase behaviour are presented. The results were achieved in a reference tracking experiment structured as follows: first, each sub-system was brought to steady state from 0 to 10 cm. Then, the setpoint for the first sub-system was changed from 10 cm to 12 cm while the reference for sub-system #2 remained unchanged at 10 cm. After stabilization, the reference was changed for the second sub-system from 10 cm to 12 cm while the reference for sub-system #1 remained at 12 cm.

In Fig 2 the comparative results between the centralized MPC (plotted with dashed red line) and the proposed DMPC algorithm (depicted with continuous blue line) for both sub-systems outputs (i.e. the water level in the lower tanks denoted with y_1 and y_2 for sub-system #1 and #2 respectively) are presented. The water levels in the upper tanks are not controlled.

The centralized and distributed algorithms were implemented using the input weight matrix parameter $\alpha = 1200$ and $\alpha = 3000$ respectively. The value was chosen after several tests, as to obtain the best DMPC result when compared with centralized MPC. To assess the performance of centralized MPC versus the DMPC algorithm the following performance indices (i.e. mean squared error MSE and control effort Eff) were computed:

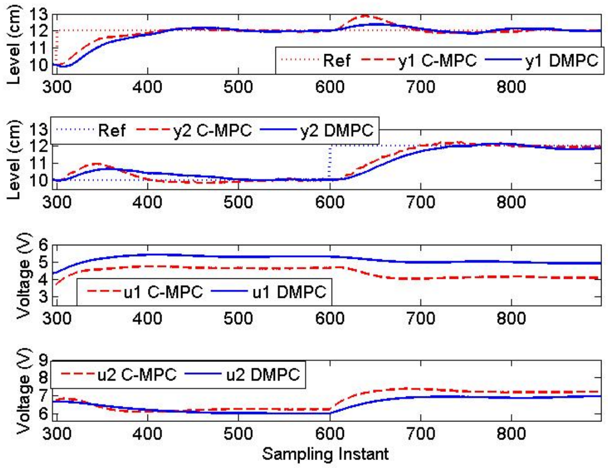

Fig. 2. Real time results for **reference tracking** on the quadruple tank process from Quanser with non-minimum phase dynamics.

Control Algorithm	MSE(cm^2)	ST (s)	Eff (V/s)
Centralized MPC(Y_1)	0.1997	200	4.3742
DMPC (Y_1)	0.2694	200	5.1220
Centralized MPC (Y_2)	0.2864	200	6.7276
DMPC (Y_2)	0.3619	200	6.4882

Table 1. MSE, ST (settling time) and Eff for centralized MPC and DMPC in **reference tracking** experiment

$$\begin{aligned} MSE_i &= \frac{1}{N}\sum [R_{si} - \tilde{Y}_i]^T [R_{si} - \tilde{Y}_i] \\ Eff_i &= \frac{1}{N}\sum \tilde{U}_i^T \tilde{U}_i \end{aligned} \tag{25}$$

where $i = \overline{1,2}$ is the sub-system index and N is the total number of discrete-time steps (i.e from time 295s until 900s). Note that $\tilde{Y}_i$ denotes a vector containing the output measurements from sub-system i and $\tilde{U}_i$ is a vector composed with the first element from the optimal solution which was sent at each sampling instant to the process. Table 1 corresponds to the results given in Fig. 2. The results indicate that DMPC performs similar to centralized MPC strategy in the initial part of the experiment, when the setpoint changes for each sub-system (i.e. y_1 analysed starting from 300 s until 500 s and y_2 in the time range 600 s - 800 s). On the other hand, regarding the interaction between the sub-systems, it is clear that under centralized MPC sub-system #1 has a more severe reaction when sub-system's #2 reference is changed, compared with the DMPC case.

The performance between the centralized MPC and the DMPC methods was tested on another sepoint tracking test, in which the references of both sub-systems were moved simultaneously in different directions. In Fig. 3 the results are shown.

The practical benefits of the non-cooperative DMPC are best experienced in large-scale systems where a centralized

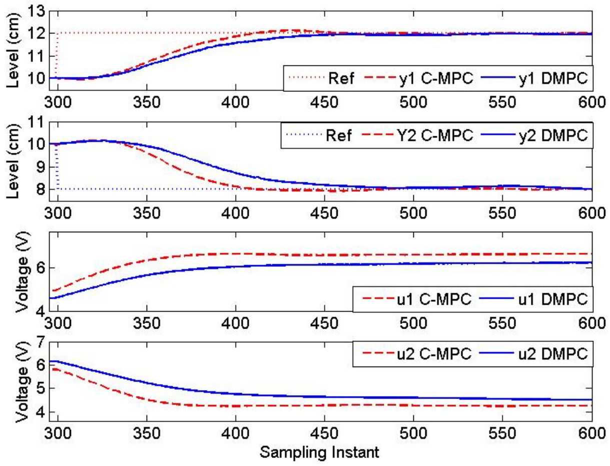

Fig. 3. Real time results for **reference tracking** on the quadruple tank process from Quanser with non-minimum phase dynamics when the references change simultaneously.

approach is impossible due to the large numbers of systems variables and the computational load on the computer in charge. Our aim was to implement both centralized and distributed control on a test-bench process and to compare the performances.

4. CONCLUSION

In this paper a non-cooperative DMPC algorithm for reference tracking is proposed. The results obtained are compared with the centralized MPC strategy. The tests were performed on the same real life plant (the quadruple tank process from Quanser) and reveal acceptable performance for the DMPC algorithm with a higher MSE index and a lower implementation complexity. Further work will focus on a robustness/stability analysis and will compare the proposed DMPC method with an input-output DMPC method. Also a more complex system (i.e. with sub-systems connected through states and inputs) will be used in experiments.

5. ACKNOWLEDGEMENT

The work by A. Maxim and C.F. Caruntu was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS UEFISCDI, project number PN-II-RU-TE-2014-4-0970.

REFERENCES

Alvarado, I., Limon, D., de la Pena, D., Maestre, J., Ridao, M., Scheu, H., Marquardt, W., Negenborn, R., De Schutter, B., Valencia, F., and Espinosa, J. (2011). A comparative analysis of distributed mpc techniques applied to the hd-mpc four-tank benchmark. *J. of Process Control*, 21, 800–815. doi: 10.1016/j.jprocont.2011.03.003.

Betti, G. (2013). *Methods and applications of distributed and decentralized model predictive control.* PhD thesis, Politecnico di Milano, Milan, Italy.

Borrelli, F. and Morari, M. (2007). Offset free model predictive control. *Proceedings of the 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, Dec. 12-14*, 1245–1250.

Christofides, P., Scattolini, R., Munoz de la Pena, D., and Liu, J. (2013). Distributed model predictive control: A tutorial review and future research directions. *Computers and Chemical Engineering*, 51(-), 21–41. doi: 10.1016/j.compchemeng.2012.05.011.

Faanes, A. and Skogestad, S. (2005). Offset-free tracking with mpc with model mismatch: Experimental results. *Ind. Eng. Chem. Res*, 44(11), 3966–3972. doi: 10.1021/ie049422y.

Gatzke, E., Meadows, E., Wang, C., and Doyle III, F. (2000). Model based control of a four-tank system,. *Computers and Chemical Engineering*, 22, 1503–1509.

Johansson, K.H. (2000). The quadruple-tank process: A multivariable laboratory process with an adjustable zero. *IEEE Trans Ctrl Syst Tech*, 8, 456–465.

Maeder, U. and Morari, M. (2007). Offset-free reference tracking for predictive controllers. *Proceedings of the 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, Dec. 12-14*, 5252–5257.

Magni, L. and Scattolini, R. (2007). Tracking of non-square nonlinear continuous time systems with piecewise constant model predictive control. *J. of Process Control*, 17(8), 631–640. doi:10.1016/j.jprocont.2007.01.007.

Maxim, A., Lazar, C., and Caruntu, C. (2015). A computationally efficient non-cooperative distributed model predictive control algorithm for two agent systems. *Proceedings of the 20th International Conference on Control Systems and Computer Science, Bucharest, Romania, May. 27-29*, 673 – 678.

Mercangoz, M. and Doyle III, F.J. (2007). Distributed model predictive control of an experimental four-tank system. *J of Process Control*, 17(-), 297–308. doi: 10.1016/j.jprocont.2006.11.003.

Pannocchia, G. and Rawlings, J. (2003). Disturbance models for offset-free model predictive control. *AIChE Journal*, 49, 426–437.

Scattolini, R. (2009). Architectures for distributed and hierarchical model predictive control a review. *J. of Process Control*, 19(5), 723–731. doi: 10.1016/j.jprocont.2009.02.003.

Vadigepalli, R., Gatzke, E., and Doyle III, F. (2001). Robust control of a multivariable experimental four-tank system. *Industrial Engineering Chemical Research*, 40, 1916–1927.

Venkat, A., Rawlings, J., and Wright, S. (2007). Distributed model predictive control of large-scale systems. *LNCIS*, 358(5), 591–605. doi:10.1007/978-3-540-72699-950.

Wang, L. (2009). *Control System Design and Implementation Using MATLAB.* 'Springer-Verlag.