DISTRIBUTED MODEL PREDICTIVE CONTROL OF A FOUR-TANK SYSTEM

Mehmet Mercangöz
Francis J. Doyle III

Department of Chemical Engineering,
University of California, Santa Barbara, CA 93106, USA

Abstract: A distributed model predictive control (DMPC) framework is proposed. The physical plant structure and the plant mathematical model are used to partition the control duties over self-sufficient estimation and control nodes. Linear models and local measurements at the nodes are used to estimate the relevant plant states. This information is then used in the model predictive control calculations. Communication among relevant nodes during estimation and control calculations provides improvement over the performance of completely decentralized controllers. The DMPC framework is demonstrated for the level control of an interacting four-tank system. The performance of the DMPC system for disturbance rejection is compared with other control configurations. The results indicate that the performance of the proposed framework provides significant improvement over completely decentralized MPC controllers, and approaches the performance of a fully centralized design.

Keywords: Distributed decentralized estimation and control, model based control, network control, plantwide control.

1. INTRODUCTION

Efficient plantwide control of chemical processing plants provides a significant economic advantage by enabling closer operation to optimization constraints, decreasing the number of shut-downs and by reducing the amount of off-specification products. Efficient control systems can also provide environmental and operational safety for these chemical plants.

Control systems in typical modern chemical plants are built in a hierarchical structure, where a large number of digital PI, PID and other simple controllers enable stable operation of most unit operations. These controllers are then connected to multivariate systems spanning several unit operations to control the important quality variables or to achieve more sophisticated tasks such as waste minimization, economic optimization or production scheduling as shown in figure 1 (Skogestad, 2004). The information flow in these hierarchical structures is in a vertical direction and the systems at the same level are not aware of the existence of their neighbors even though they may be interacting.

The objective of this paper is to develop a framework for a horizontal connection among the different control systems in a chemical plant at the multivariate control level. The industry standard for multivariate control is model predictive control (MPC) and the proposed framework provides a communication structure for estimation and control among different distributed MPC applications.

Decentralized estimation and control problems have attracted attention from several different fields. The Control of vehicle formations or a group of robots in mechanical or aerospace engineering, control of power grids in electrical engineering and coordination of wireless systems in computer science are examples of these problems. Achievements, especially in the field of multivariate estimation and control, include the development of parallel partially decentralized controllers (Siljak, 1991). Multi-level hierarchical control systems have been designed based on decomposition and coordination strategies (Findeisen *et al*., 1980). A fully distributed decentralized estimation and control structure (DDEC) to achieve the same performance of a centralized algorithm under certain conditions have also been developed (Mutambara, 1998). This method has been successfully applied to a chemical engineering plantwide control problem for a state-feedback control law (Vadigepalli and Doyle III, 2003). Distributed approaches to MPC applications have also been investigated (Jia and Krogh, 2001). However, these approaches involved assumptions on the worst-case interactions to design a stabilizing hierarchical control strategy.

In the present study, a nodal estimation network is designed as an extension on the scalable DDEC methodology of Mutambara (1998), however the simple state feedback based control structure of the DDEC is replaced with an MPC algorithm. The nodal communication of state information in the original DDEC methodology is preserved and extended to include the communication and renewal of MPC results among relevant controllers before the implementation of control action. Additional requirements regarding model decomposition are also considered for the design of a DMPC network.

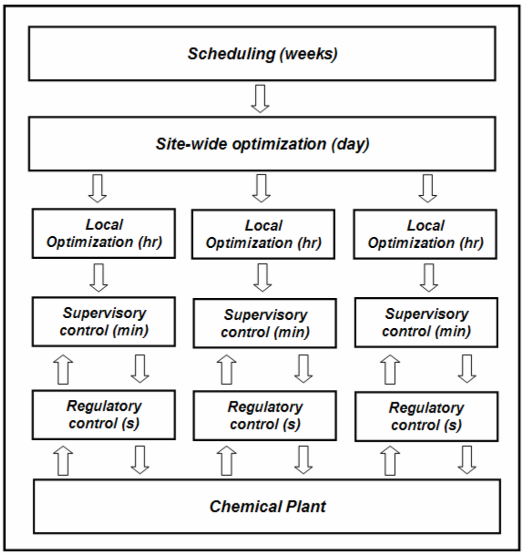

Fig.1. Hierarchical organization of process control systems in chemical plants.

In the following sections the DDEC methodology is revisited and the DMPC framework is outlined. The application of the proposed framework to an interacting four-tank system is provided. The simulation results to compare the efficiency of the DMPC algorithm to conventional centralized and completely decentralized MPC formulations precede a discussion of these results and a conclusion section.

2. DDEC METHODOLOGY

The DDEC methodology is based on a linear discrete-time plant with n_u inputs and n_y outputs of the following state–space form:

$$x(k)=\Phi x(k-1)+Bu(k-1)+w(k-1) \qquad (1)$$

$$y(k)=Hx(k)+v(k) \qquad (2)$$

where $x(k)\in\Re^n$ is the n-dimensional state of interest at time k; $\Phi:\Re^n\rightarrow\Re^n$ is the state transition matrix from time $(k-1)$ to k; $u(k)\in\Re^{n_u}$ and $B:\Re^n\rightarrow\Re^{n_u}$ are the input vector and matrix, respectively; $y(k)\in\Re^{n_y}$ is the measurements vector at time k; $H:\Re^n\rightarrow\Re^{n_y}$ is the observation matrix; and, $w(k)\sim N(0,Q)$ and $v(k)\sim N(0,R)$ are the associated process and measurement noise vectors, respectively, and are modelled as uncorrelated, zero mean sequences with covariance matrices Q and R, respectively.

2.1 Model Decomposition

The plant model given in (1) and (2) is partitioned according to either physical structure or based on an analysis of the mathematical model. At this stage the number of DDEC nodes is determined along with the allocation of measurements and control inputs among different nodes. Even though two nodes can share measurements, a control input cannot be assigned to more than a single node. The number of nodes should also be chosen carefully to evenly distribute the computational requirements and the communication load due to overlapping states. Linear transformations T_i for each node i are then designed to obtain the local state transition and observations given by:

$$x_i(k)=\Phi_i x_i(k-1)+B_i u_i(k-1)+w_i(k-1) \qquad (3)$$

$$y_i(k)=C_i x_i(k)+v_i(k) \qquad (4)$$

where $u_i(k)$ are the inputs affecting the local states. Φ_i is related to the global state transition matrix $\Phi(k)$ as $\Phi_i=T_i\Phi(k)T_i^\dagger$, where $T_i^\dagger$ is the generalized inverse of T_i. The local state vector at node i, $x_i(k)$, is related to the global state vector $x(k)$ by $x_i(k)=T_i x(k)$.

2.2 Distributed Prediction and Estimation

The prediction and estimation calculations at every time step are done according to the distributed and decentralized Kalman filter (DDKF). The state $x_i(k)$ at node i, is predicted according to:

$$\hat{x}_i(k|k-1)=\Phi_i\,\hat{x}_i(k-1|k-1)+B_i\,u_i(k-1) \qquad (5)$$

$$P_i(k|k-1)=\Phi_i P_i(k-1|k-1)\Phi_i^{\mathrm{T}}+Q_i \qquad (6)$$

where Q_i and R_i represent the local covariance matrices of process and measurement noise, respectively. The estimation step follows in three stages: (i) local estimation, (ii) internodal communication and (iii) assimilation to produce a global estimate. The Local covariance and state estimates are computed from local measurements as follows:

$$P_i(k|y_i(k))=[C_i^T R_i^\dagger C_i]^\dagger \qquad (7)$$

$$\hat{x}_i(k|y_i(k))=P_i(k|y_i(k))[C_i^T R_i^\dagger]\,y_i(k) \qquad (8)$$

The relevant subset of local estimates of the state and prediction error covariances are communicated to relevant nodes and the information at each node is transformed into the local state subspace. The transformed state and covariance estimates are given by

$$P_i^\dagger(k|y_j(k))=T_i[T_j^T P_j^\dagger(k|y_j(k))T_j]^\dagger T_i^T \qquad (9)$$

$$\hat{x}_i(k|y_j(k))=T_i T_j^\dagger\,\hat{x}_j(k|y_j(k)) \qquad (10)$$

The transformed states are assimilated locally to produce state and covariance estimates according to:

$$\hat{x}_i(k|k) = P_i(k|k)[P_i^{-1}(k|k-1)\hat{x}_i(k|k-1) + \sum_{j=1}^{N} P_i^{\dagger}(k|y_j(k))\hat{x}_i(k|y_j(k))] \quad (11)$$

$$P_i(k|k) = [P_i^{-1}(k|k-1) + \sum_{j=1}^{N} P_i^{\dagger}(k|y_j(k))]^{\dagger} \quad (12)$$

This combined process of local prediction, inter-nodal communication and assimilation among N nodes produces estimates identical to those obtained from an equivalent centralized Kalman filter algorithm.

2.3 Distributed Control

A nodal control law obtained as a cost minimizing control function is given by

$$u_i(k) = K_{ci}[x_{ri}(k) - \hat{x}_i(k|k)] \quad (13)$$

where $x_{ri}(k)$ is the local state reference, $\hat{x}_i(k|k)$ is the local optimal state estimate, and K_{ci} is the optimal control gain computed from the solution to a distributed and decentralized backward Riccati recursion. The prediction, estimation and control stages of the DDEC algorithm are shown in figure 2.

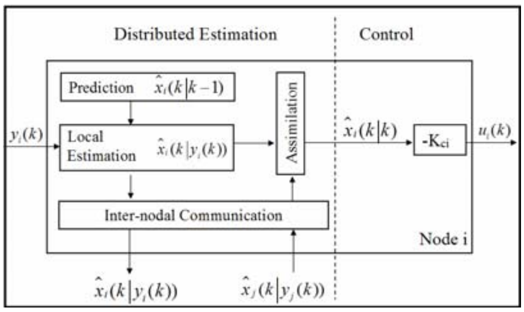

Fig.2. Organization of the distributed estimation and state-feedback based control in the DDEC algorithm.

3. DMPC FRAMEWORK

The DMPC framework relies on a similar model decomposition structure as given in (1) to (4). However, local models are developed by explicitly indicating interactions due to control moves from n neighboring nodes given as

$$x_i(k) = \Phi_i x_i(k-1) + B_i u_i(k-1) + \sum_{j=1}^{n} B_j u_j(k-1) + w_i(k-1) \quad (14)$$

$$y_i(k) = C_i x_i(k) + v_i(k) \quad (15)$$

The DMPC framework also requires "self-sufficiency" of local subsystems, meaning that every subsystem will be able to estimate the local states and achieve the local control objectives with the measurements and control inputs allocated to that node. This translates into the requirement that every subsystem will be observable with the allocated measurements and controllable with the assigned control inputs. "Self-sufficiency" will enable successful operation of a node in the case of a failure in other nodes or during an intermission in the communication structure. In the present study, it is assumed that an off-line Kalman filter with innovation gain M_i is available at every node prior to the start of the algorithm based on the local nodal models (14) and (15), and also on the expected disturbances. There are no requirements on the connectivity of the nodal DMPC network. The states of the local models x_i are separated into two sets, x_i^o and x_i^{no}, denoting the overlapping and non-overlapping components respectively. Moreover, every node i has reliability factors $r_{i(l)}^j$ for any overlapping state l of x_i^o coming from a relevant node j as determined during the design of the local Kalman filters.

3.1 DMPC estimation

The DMPC algorithm is initiated when a node i obtains its corresponding measurements $y_i(k)$. These local measurements are used with the predictions from the previous time step $\hat{x}_i(k|k-1)$ to produce local state estimates according to

$$\hat{x}_i(k|y_i(k)) = \hat{x}_i(k|k-1) + M_i[y_i(k) - C_i\hat{x}_i(k|k-1)] \quad (16)$$

Subsets of these local state estimates among x_i^o are broadcasted to relevant nodes and, in return, external state information is received back. The estimate for state l sent by a node j to another node i is denoted by $\hat{x}_{i(l)}^j(k|y_i(k))$, where l denotes a certain state in x_i^o. A given node i can interact with multiple other nodes and the shared states can be completely different or have common elements between different interconnections.

After the communication step, the received state estimates are weighted and fused together with the local estimates at node i, according to the pre-assigned reliability factors as:

$$\hat{x}_{i(l)}(k|k) = \sum_{j=1}^{q} \hat{x}_{i(l)}^j(k|y_j(k)) r_{i(l)}^j \quad (17)$$

where q is the number of overlapping nodes for the state $x_{i(l)}$, including node i itself and the corresponding reliability factors for the different estimates are related by

$$\sum_{j=1}^{q} r_{i(l)}^{j} = 1 \tag{18}$$

This step ensures that the information about the shared states x_i^o is distributed throughout the network. In the DDEC scheme, the assimilation step uses the inverses of the estimation error covariances to weight the information coming from all other nodes. Since the DMPC framework is based on an off-line suboptimal Kalman filter, inverses of the steady-state error covariances can be used as reliability factors for the corresponding local estimators. However, because of the generality of the DMPC framework in terms of the interconnection structure and the number of overlapping states, there is no restriction on the choice of $r_{i(l)}^{j}$ besides (18). One potential negative aspect of the DMPC estimation scheme is the loss of equivalence to an optimal centralized estimator, as is the case with the DDEC methodology.

3.2 DMPC prediction and control

State estimates obtained in the previous section are used to initialize the local models (14) and (15). At each time step the nodes also receive information about the control moves of the neighboring nodes in the previous time step. This information is then used in the state transition equations (14) to include the effects of input interactions by assuming constant values throughout the prediction horizon. The prediction and control stages are conducted locally at each node *i*, according to the MPC algorithm by solving a numerical optimization problem given as

$$\begin{aligned} \min_{u_i(k|k),\ldots,u_i(m-1+k|k)} & \sum_{t=0}^{p-1}\Big(\sum_{s=1}^{n_{iy}}[w_{t+1,s}^{y}(y_{it}(k+t+1|k) \\ & -y_{is}^{ref}(k+t+1))]^2 + \sum_{s=1}^{n_{iu}}[w_{t,s}^{u_i}u_{is}(k+t|k)]^2\Big) \\ s.t. \quad & u_i^{low} \le u_i \le u_i^{high} \\ & \Delta u_i^{low} \le \Delta u_i \le \Delta u_i^{high} \\ & u_i(k+h|k)=0 \quad for \quad h=m,m+1,\ldots,p-1 \end{aligned} \tag{19}$$

where s denotes the s^{th} component of a vector, $(k+t|k)$ denotes the t steps ahead prediction using the local models, based on information available at time k and finally, y_i^{ref} denotes the output reference for sample time k.

When a node obtains the solution of the local MPC problem, it sends the control moves for the next time step to its interacting neighbors before implementing it in the actual system. The nodes then use this information to update the input interactions in the prediction models and repeat the MPC calculations with the same initial states. The MPC calculations can be repeated for a certain number of iterations or based on a convergence criterion. The convergence properties of the DMPC algorithm will be reported in a subsequent publication. After a satisfactory solution is obtained from the MPC calculations, local nodes implement the control moves for the current time step and the predictions for state information is send to the next estimation stage which starts again as new measurements are received.

The prediction and control structure of the DMPC framework introduces cooperation for control calculations in the nodal network, whereas in the DDEC scheme the network only cooperates for state estimation and the control calculations are performed locally. Even though repeated MPC calculations are computationally more cumbersome compared to LQG based local controllers, the ability to include constraints, and the flexibility in the design of controllers considerably improve the performance of a DMPC network over a DDEC counterpart. The prediction estimation and control stages of the DMPC algorithm are depicted in figure 3.

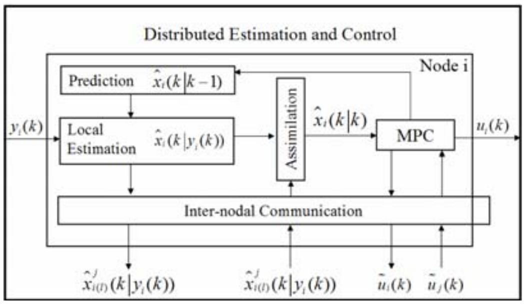

Fig.3. Organization of the distributed estimation, MPC based control and the inter-nodal communication stages in the DMPC framework.

4. CASE STUDY

The DMPC framework is applied in a case study involving level controls in an interacting four-tank system. This problem provides a simple, illustrative example for system decomposition and yet has challenging dynamic behaviour that can distinguish the performance of different control strategies.

4.1 System Description

The simulated four-tank problem considered here is a variant of the experimental system described by Gatzke *et al*. (2000). A schematic of this process is shown in figure 4. The system has two inputs (pump speeds) which can be manipulated to control the two outputs (levels in tanks 3 and 4). The multivariate dynamics is created by the cross-recycle streams feeding the two different overhead tanks 1 and 2. In this case study the dynamics are enriched by adding first order lags between the control signals and the pump throughput, and the system is simulated based on a full nonlinear mass balance model given in 20. Bernoulli's law is used for the flows out of the tanks, A_i stands for the cross sectional area, h_i for the liquid level and k_i for the flow factors from tank *i*. F_{in} stands for the input flows to the overhead tanks, d_i for the flow disturbances, γ_i for the recycle flow ratios, v_i for actual pump throughput and finally u_i for the controller output.

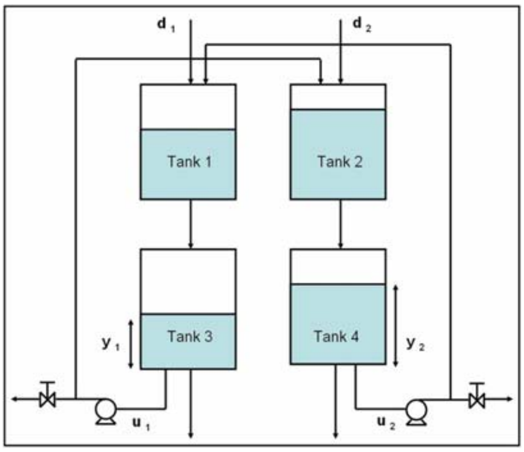

Fig.4. Schematic description of the four-tank system.

$$
\begin{aligned}
A_1 \frac{dh_1}{dt} &= F_{in} + d_1 + \gamma_1 v_2 - k_1 \sqrt{h_1} \\
A_2 \frac{dh_2}{dt} &= F_{in} + d_2 + \gamma_2 v_1 - k_2 \sqrt{h_2} \\
A_4 \frac{dh_4}{dt} &= k_2 \sqrt{h_2} - v_2 - k_4 \sqrt{h_4} \\
A_3 \frac{dh_3}{dt} &= k_1 \sqrt{h_1} - v_1 - k_3 \sqrt{h_3} \\
\tau_1 \frac{dv_1}{dt} &= -v_1 + u_1 \\
\tau_2 \frac{dv_2}{dt} &= -v_2 + u_2
\end{aligned} \tag{20}
$$

By omitting all information about the flow disturbances, the nonlinear model equations with specified parameter values can be linearized and rearranged to give the following discrete-time system with a sampling frequency of 1 s^{-1}. This system will serve as the starting centralized model in the DMPC framework, corresponding to equations (1) and (2).

$$
x(k+1) = \begin{bmatrix} 0.89 & 0.26 & -0.2 & 0.03 & 0 & 0 \\ 0 & 0.34 & 0 & 0.08 & 0 & 0 \\ 0 & 0 & 0.37 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.45 & 0 & 0 \\ 0 & 0 & 0.03 & -0.1 & 0.82 & 0.27 \\ 0 & 0 & 0.09 & 0 & 0 & 0.25 \end{bmatrix} x(k) + \begin{bmatrix} -0.1 & 0.01 \\ 0 & 0.06 \\ 0.63 & 0 \\ 0 & 0.69 \\ 0.01 & -0.1 \\ 0.07 & 0 \end{bmatrix} u(k) \tag{21}
$$

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

4.2 Nodal Decomposition

The control objective in the four-tank system is to keep the levels at tanks 3 and 4 at the specified reference values in the face of flow disturbances. This objective and an examination of the rest of the process lead to a clear division of the system into two physical subsystems with each subsystem containing a controlled tank 3 or 4 and their matching overhead tanks 1 or 2. The physical separation of the process into two subsystems also allocates the measurements (tank levels 3 or 4) and the manipulated variables corresponding to each controlled tank as well (pump speeds 1 or 2).

The states of the system described in (21) are pre-arranged for the demonstration of mathematical decomposition. The measured states 1 and 5 are the focal points for the distribution of states among the two subsystems and they are positioned diagonally on the two sides of the state transition matrix. This arrangement forms a disconnected system, except for states 3 and 4, which are shared between the two subsystems leading to nodal models given below. These models correspond to equations (14) and (15) of the DMPC framework. In this distribution node 1 contains states 1 to 4, output 1 and input 1, whereas node 2 contains states 3 to 6, output 2 and input 2. This decomposition also satisfies the "self-sufficiency" requirement for the nodes in a DMPC network.

$$
x_1(k+1) = \begin{bmatrix} 0.89 & 0.26 & -0.2 & 0.03 \\ 0 & 0.34 & 0 & 0.08 \\ 0 & 0 & 0.37 & 0 \\ 0 & 0 & 0 & 0.45 \end{bmatrix} x_1(k) + \begin{bmatrix} -0.1 \\ 0 \\ 0.63 \\ 0 \end{bmatrix} u_1(k) + \begin{bmatrix} 0.01 \\ 0.06 \\ 0 \\ 0.69 \end{bmatrix} u_2(k) \tag{22}
$$

$$
y_1(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} x_1(k)
$$

$$
x_2(k+1) = \begin{bmatrix} 0.37 & 0 & 0 & 0 \\ 0 & 0.45 & 0 & 0 \\ 0.03 & -0.1 & 0.82 & 0.27 \\ 0.09 & 0 & 0 & 0.25 \end{bmatrix} x_2(k) + \begin{bmatrix} 0 \\ 0.69 \\ -0.1 \\ 0 \end{bmatrix} u_2(k) + \begin{bmatrix} 0.63 \\ 0 \\ 0.01 \\ 0.07 \end{bmatrix} u_1(k) \tag{23}
$$

$$
y_2(k) = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} x_2(k)
$$

4.3 DMPC design

The next stage in the construction of a DMPC network is to add the anticipated disturbance and noise models on the different subsystems. In this case, a single unmeasured disturbance and output noise channels are added to both nodes. Kalman filters, based on these models are then designed with expected covariance values for the process and measurement noises.

The overlapping states between the subsystems were determined during the nodal decomposition step, however for the DMPC design, the reliability factors for these states at each node has to be specified. For simplicity, in this case study the reliability of state estimation at both nodes are assumed to be the same and factors $r^2_{1(3)}$, $r^2_{1(4)}$, $r^1_{2(3)}$, $r^1_{2(4)}$ are all taken as 0.5 in accordance with (18).

For the MPC design, upper and lower limits for pump speeds are taken as 5 and 0, and only output weights of 100 are used in both subsystems. The prediction horizons are specified as 8 and the move horizons are specified as 3. The DMPC framework does not require symmetry for the design of the nodal MPC controllers, however similar computational loads will create a balanced network and nodal communication will proceed without long delays. In this case study, the performance of the DMPC network is compared to two other control strategies. The first one employs a centralized MPC controller for the whole system and the second one has two completely decentralized MPC controllers. Both of these strategies have the same input constraints, output weights, prediction and control horizons as in the DMPC design. As a final design parameter, the MPC solutions are repeated only once by the DMPC nodes.

4.4 Performance Comparison

The performance of the DMPC network is compared with the centralized and completely decentralized MPC controllers in a simulation study using the nonlinear four-tank system. Concurrent feed flow disturbances were considered in the simulation scenarios in the form of 1 and 1.5 m^3/s steps for disturbances 1 and 2 respectively. Different control strategies were compared based on the maximum deviations from the set-points, the integral absolute errors and the settling times for the two outputs.

The simulation results are shown in figure 5 and the performance measures are listed in table 1. The results show that in the present configuration the concurrent disturbances effect output 1 more profoundly and in return the controllers have more difficulty managing this output. According to the results, the DMPC formulation outperforms the fully decentralized MPC controllers by a large margin. For both outputs all three performance measures are in favour of the DMPC but for output 1 the difference is more pronounced. Comparison of the DMPC formulation with the centralized MPC controller reveals that the DMPC comes fairly close to the performance of the centralized MPC. Moreover, even though the overall performance of the centralized MPC is better than the DMPC scheme, the DMPC has better results for the settling time and IAE for output 2.

The short oscillations in output 1 after the disturbance are caused by the distributed state estimation due to the initial differences in the state estimates for the shared states. The oscillations disappear as both estimators converge to the correct state estimates. This behaviour is not observed in output 2.

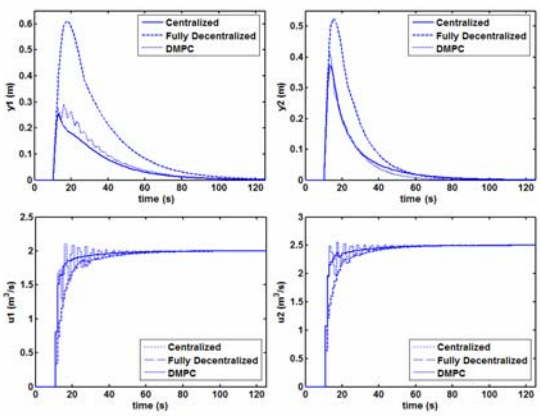

Fig.5. Dynamic response of the four-tank system to concurrent feed flow disturbances.

Table 1 Performance measures for different control formulations

	Settling time (s)	Maximum Deviation (m)	IAE (m·s)
Centralized MPC (y1)	90	0.24	6.1
Centralized MPC (y2)	90	0.37	5.7
DMPC (y1)	90	0.28	7.5
DMPC (y2)	70	0.42	5.2
Decentralized MPC (y1)	110	0.61	16.3
Decentralized MPC (y2)	90	0.52	9.2

5. CONCLUSION

In the present study, a distributed model predictive control framework is presented. The methodology is demonstrated on a four-tank level control problem. The results show that the new methodology performs significantly better than a completely decentralized set of controllers. In terms of computational parallelization, the size of the individual problems is reduced by more than 33 % compared to a completely centralized formulation. Iterative solutions of the DMPC provide feed-forward anticipation of the interactions due to control inputs, bringing the performance of the DMPC closer to that of a completely centralized MPC. In addition, the “self-sufficiency” criterion required for the DMPC enables the subsystems to stay functional in case of failures in other subsystems. With these properties, the DMPC framework can be a viable candidate to provide a connecting link between existing MPC applications in chemical engineering systems.

REFERENCES

Findeisen, W., Brdys, M., Malinowski, K., Tatjewski, P. and Wozniak, A. (1980). *Control and coordination in hierarchical systems*, Wiley Inc., New York, NY.

Gatzke, E.P., Meadows, E.S., Wang, C. and Doyle III, F. J. (2000). Model based control of a four-tank system. *Comput. Chem. Eng.,* **24**, 1503-1509.

Jia, D., & Krogh, B. H. (2001). Distributed model predictive control. In: *Proceedings of the American Control Conference*, pp. 2767–2772. Arlington, VA.

Mutambara, A.G.O. (1998). *Decentralized estimation and control for multisensor systems*, CRC Press, Boca Raton, FL.

Siljak, D.D. (1991). Decentralized control of complex systems. *Mathematics in Series and Engineering,* **184**, Academic Press, San Diego, CA.

Skogestad, S. (2004). Control structure design for complete chemical plants. *Comput. Chem. Eng.,* **28**, 219-234.

Vadigepalli, R. and Doyle III, F. J. (2003). A distributed state estimation and control algorithm for plantwide processes. *IEEE Transactions Contr. Syst. Technol.*, **11**, 119-127.