Proceedings of the 2nd International Workshop on Networked Control Systems
Rende, Italy, November 23-24, 2006

Control and Operation of Heat Exchanger Networks using Model Predictive Control

L. Giovanini and J. Balderud

Abstract— While, during the past decades, many strategies for heat exchanger network synthesis and design have been developed, much less effort have been dedicated to developing online optimal control strategies to tackle their complex and distributed dynamics. Since past optimal control design efforts predominately revolves around centralized control ideas they typically suffer from high computational demands. By contrast, this paper proposes an agent based decentralized predictive control approach, where the computational demand is distributed between several agents. This paper also explores the computational demand associated with the proposed approach and compares it against a traditional, centralized, predictive control approach.

I. Introduction

Many methods for heat exchanger network (HEN) synthesis have been developed over the past decades, which aim to provide HEN designs that yields a reasonable trade-off between capital and operating costs Since the mid 1980s several authors have also investigated the flexibility of HENs, partly for the purpose of analyzing existing HENs [1], and partly for the purpose of improving the synthesis of new ones [2]-[4]. For example, in Papalexandri and Pistikopoulos [2] HEN synthesis and flexibility are considered simultaneously using mathematical programming.

Compared to synthesis of nominal and flexible HENs, much less effort has been dedicated to find methods for the operation of HENs. Mathisen, Skogestad and Wolff [5] investigated bypass selection for control of HENs, without considering the utility consumption. In Mathisen, Morari and Skogestad [6] method for operation of HENs that minimizes utility consumption is proposed. The method is based on structural properties of the network; however, the variable control configuration may result in poor dynamic performance. A method based on repeated steady state optimization is suggested by Boyaci et al. [7], but their focus is not on the control structure for closed loop implementation. Aguilera and Marchetti [8] proposed a method for on-line optimization and control of HENs. At each sample, the optimization problem finds the optimal solution of the HEN at steady-state by specifying the optimal values of some services. They also discussed degrees of freedom with respect to optimization of HENs during operation.

Later, Glemmestad et al. [9] introduced an algorithm for optimal operation of HENs. This is normally done in two steps:

1. Obtaining the optimal solution at regular time intervals, using steady state data.
2. Implementing the optimal solution by specifying the optimal values for some variables (setpoints).

They introduce a simple optimization problem where the bypass fractions do not explicitly appear. This makes the problem less nonlinear. An alternative way of solving the optimization problem, which works in some cases, is to use structural information only [10], [11].

Recently, Giovanini and Marchetti [12] introduced a low-level control structure capable of efficiently handling constraints on the manipulated variables of HENs. The control structure, called *flexible-structure control*, has the capability to modify the closed-loop structure in order to keep regulation and can be configured in most distributed control systems. This control approach is useful to hold the operating point close to an optimum when optimal conditions are located on the constraints.

In this paper centralized and decentralized model predictive control (MPC) algorithms will be employed to operate and control a HEN. In the approach presented in this work, the transient and the steady-state solutions are optimize simultaneously. In this way, a reconfiguration of the system is carried out in order to preserve the controllability of the network.

In the following, it is assumed that the stream data (heat capacity flow rates and supply: target temperatures), network structure and heat exchanger areas are given and the HEN is sufficiently flexible. To manipulate the network it is assumed that utility duties can be adjusted and that a variable bypass is placed across each process-to-process heat exchanger. In case of stream splits, we may also assume that split fractions can be varied.

The remaining part of the paper is organized as follows: First, the model predictive controls in their centralized and decentralized implementations are introduced. Then, in Section 3 the performance of the different algorithms is

Manuscript received January 29, 2006. This work was supported by the EPSRC under Grant BS123456.

L. Giovanini, J Balderud and M. Grimble are with the Industrial Control Centre, Department of Electrical and Electronic Engineering, University of Strathclyde, Glasgow, UK (e-mail: l.giovanini@eee.strath.ac.uk, j.balderud@eee.strath.ac.uk, m.grimble@eee.strath.ac.uk,).

analyzed. Later, the MPC algorithms are applied to an example in Section 4. Finally some conclusions are drawn in Section 5.

II. Model Predictive Control

Model predictive control (MPC) is formulated as resolving an on-line open loop optimal control problem in moving horizon style [13]. Using the current state, an input sequence is calculated to minimize a performance index while satisfying some specified constraints. Only the first element of the sequence is taken as controller output. At the next sampling time, the optimization is resolved with new measurements from the plant. Thus both the control horizon and the prediction horizon move or recede ahead by one step at next sampling time. The purpose of taking new measurements at each sampling time is to compensate for unmeasured disturbances and model inaccuracy, both of which cause the system output to be different from its prediction. At decision instant k, the controller samples the state of the system $x(k)$and then solves an optimization problem of the following form to find the control action:

$$\begin{gathered}\min_{U(k)} J\left(X(k),U(k)\right)\\ st.\\ x(k+i+1,k)=Ax(k+i,k)+Bu(k+i,k) \quad x(k,k)=x(k)\\ 0\geq G\left(X(k),U(k)\right)\end{gathered} \tag{1}$$

where

$$\begin{aligned}X(k)&=\{x(k,k)\cdots x(k+V,k)\} \quad V>M,\\ U(k)&=\{u(k,k)\cdots u(k+M,k)\},\end{aligned} \tag{2}$$

In the preceding formulation, the performance index represents $J(X(k),U(k))$ the measure of the difference between the predicted behavior and the desired future behavior. The variables $x(k+i,k)$ and $u(k+i,k)$ are, respectively, the predicted state and the predicted control at time $k+i$ based on the information at time k and system model

$$\begin{aligned}x(k+1)&=Ax(k)+Bu(k),\\ y(k)&=Cx(k),\end{aligned} \tag{3}$$

where $x\in R^n$, $u\in R^m$ and $y\in R^p$. The constraints $G(X(k),U(k))$ represent physical limits in the system and can also be other constraints to ensure the stability or robustness of the system. The optimization produces an open-loop optimal control sequence in which the first control value is applied to the system $u(k)=u(k,k)$. Then, the controller waits until the next control instant and repeats this process to find the next control action.

Typically, MPC is implemented in a centralized fashion. The complete system is modeled, and all the control inputs are computed in one optimization problem. Many successful MPC applications have been reported in the last two decades [13]. However, solving a single optimization problem for the entire system typically requires significant computation, which scales poorly with the size of the system. To address the computational issue, two research lines have being explored: one approach is to reduce the computational burden by using suboptimal approximations (reducing the number of decision variables or using approximation of the problem) [14] - [18]. More recently Van Antwerp and Braatz [19] developed an iterative ellipsoid algorithm to allow the quick computation of sub-optimal control moves. It should be pointed out that these approaches still take centralized computation and therefore increase the computing burden and need high cost computers.

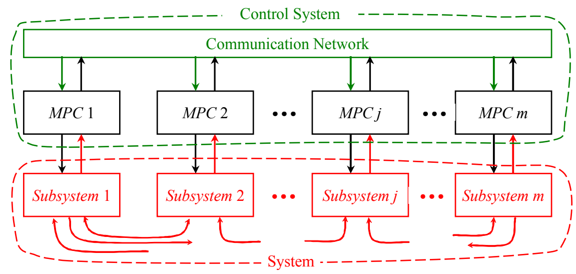

Fig. 1 Distributed control system architecture.

For large-scale systems, because of the effect of control horizon M, the number of optimized control variables $U(k)$ at each sampling time are highly dimensional, the computation is intensive which requires high performance computers or advanced algorithms. To avoid the prohibitively high on-line computational demand a distributed scheme with inexpensive computers under network environment can be employed.

A. Distributed Model Predictive Control based on Nash Optimality

The main idea of the formulation of distributed model predictive control algorithm presented in this work is to decompose the optimization problem (1) into a number of small-scale optimizations connected via dynamic input coupling [20]. These autonomous subsystems are connected via network share the common resources, communicate and co-ordinate each other in order to accomplish the whole objective.

Assumed that the behavior of the whole system is described by m agents, the cost function J can be rewritten as follows

$$J\left(X(k),U(k)\right)=\sum_{l=1}^{m}\gamma_l J_l\left(X(k),U_l(k),U_{n\neq l}(k)\right) \quad n=1,\ldots,m, \tag{4}$$

where the original cost function has been decomposed into m performance indexes related with the local decision variable $U_l(k)=[\,u_l(k)\cdots u_l(k+M)\,]$ $l=1,\ldots,m$. However, the outputs each subsystem are still related to the remaining decision variables $U_{l\neq n}(k)$ $n=1,\ldots,m$. In this way, the optimization (1) can be decomposed into m coupled optimization problem

$$
\begin{aligned}
&\min_{U_l(k)} J_l\left(X(k),U_l(k),U_{n\neq l}(k)\right) \qquad l,n=1,\dots,m\\
&st.\\
&x(k+i+1,k)=Ax(k+i,k)+B_l u_l(k+i,k) \quad x(k,k)=x(k)\\
&\qquad +\textstyle\sum_{\substack{n=1\\ n\neq l}}^{m} B_n u_n(k+i)\\
&0\geq G\left(X(k),U_l(k),U_{n\neq l}(k)\right)
\end{aligned}
\tag{5}
$$

where $U_{l\neq n}(k)$ is assumed given. Given that the communication network is reliable and with capacity that allows the subproblem to exchange information while they solve their local optimization problem, then such distributed problem can be solved by means of the *Nash optimality* concept [21].

Definition 1: A group of control decisions $U(k)=[U_1(k)\cdots U_m(k)]$ is said to be Nash optimal if

$$J_l\left(X(k),U_l^q(k),U_{n\neq l}^q(k)\right)\leq J_l\left(X(k),U_l(k),U_{n\neq l}^q(k)\right) \tag{6}$$

If the Nash optimal solution is achieved, each subproblem does not change its control decision $U_l(k)$ because it has achieved the locally optimal objective under the above conditions; otherwise the local performance index $J_l(X(k),U(k))$ will degrade. Each subsystem optimizes its objective function using its own control decision assuming that other subsystems' solutions are known and optimal. So, if condition (6) is satisfied, the whole system has arrived to an equilibrium point (attractor) in the coupling decision process (see Figure 2).

Since the mutual communication and the information exchange are adequately taken into account, each subsystem solves its local optimization problem provided that the other subsystems' optimal solutions are known. Then, each agent compares the newly optimal solution with that obtained in the previous iteration and checks if the terminal condition is satisfied. If the algorithm is convergent, all the terminal conditions of the *m* agents will be satisfied, and the whole system will arrive at Nash equilibrium at this time. This Nash-optimization process will be repeated at next sampling time.

Algorithm

Step1: At sampling time instant k, each subsystem makes initial estimation of their decision variables and communicates it to the other agents, let the iterative index $q=0$;

$$U_l^q(k)=\left[u_l^q(k)\cdots u_l^q(k+M)\right] \quad l=1,2,\dots,m$$

Step 2: Each agent solves its optimization problem simultaneously to obtain its solution

$$
\begin{aligned}
&\min_{U_l(k)} J_l\left(X(k),U_l(k),U_{n\neq l}(k)\right) \qquad l,n=1,\dots,m\\
&st.\\
&x(k+i+1,k)=Ax(k+i,k)+B_l u_l(k+i) \qquad x(k,k)=x(k)\\
&\qquad +\textstyle\sum_{\substack{n=1\\ n\neq l}}^{m} B_n u_n(k+i)\\
&0\leq G\left(X(k),U_l(k),U_{n\neq l}(k)\right)
\end{aligned}
$$

Step 3: Each agent checks if its terminal iteration condition is satisfied

$$\left\|U_l^q(k)-U_l^{q-1}(k)\right\|_\infty \leq \varepsilon_l \quad l=1,\dots,m$$

If all the conditions are satisfied, then end the iteration and go to step 4; otherwise

$$q=q+1;\ U_l^q(k)=U_l^{q-1}(k) \quad l=1,\dots,m,$$

all agents exchange this information through communication and go to step 2.

Step 4: Computes the instant control law

$$u_l(k)=\left[I\ 0\cdots 0\right]U_l^{q-1}(k) \quad l=1,\dots,m,$$

Step 5: Move horizon to the next sampling time, $k+1\rightarrow k$, and go to step 1, repeat the process.

The solutions of this algorithm are *Nash equilibrium* and the global optimality (*Paretto optimal*) of the solutions will depend on the objective function employed. Two decentralized MPC schemes can be derived from this algorithm, depending on the cost function minimized by the optimization problem [20]. If the cost function only includes local information

$$J_l^*\left(X_l(k),U_l(k),U_{n\neq l}(k)\right)=J_l\left(X(k),U_l(k),U_{n\neq l}(k)\right) \tag{7}$$

the resulting MPC algorithm is called **communication based MPC**. Since the number of interacting agents is finite and only use local costs, the control law will not be global optimal [19]. On the other case, if the cost function includes global information

$$J_l^{\#}\left(X(k),U_l(k),U_{n\neq l}(k)\right)=\sum_{l=1}^{m}\gamma_l J_l\left(X(k),U_l(k),U_{n\neq l}(k)\right), \tag{8}$$

the resulting MPC algorithm is called **cooperation based MPC**. This scheme guarantees the global optimality of the solution [20], [22].

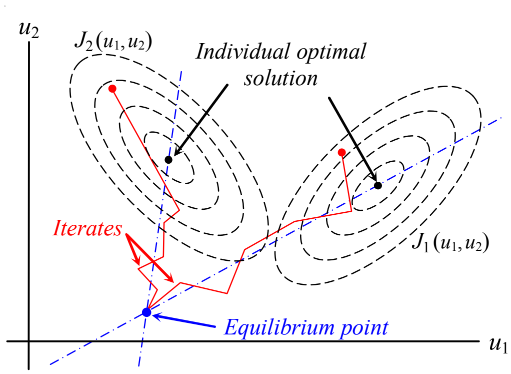

Fig. 2. Trajectories in decision space traced by two subsystems.

III. Application Example

The example network used in this work has been previously proposed as a benchmark by Aguilera and Marchetti [8] (see Figure 3). The nominal stream conditions for this network are given in Table 1, and the heat-exchanger areas are in Table 2.

A justification for the bypass configuration exhibited in Figure 3 is outside the scope of this article. A detailed discussion is given in Aguilera and Marchetti [8], where the configuration is identified as design B. This bypass configuration provides the best dynamical performance but the smaller operational region, which leads to a frequent saturation of the manipulated variables. The focus here is on those manipulated variables that hit limit constraints when the system follows a sequence of setpoint changes, and when it must reject a sequence of load disturbances. A frequent saturation of bypass x_3 is a consequence of the double bypass to the heat exchanger I_3; this means that though direct bypass gives good control performance for temperature T_{h2}, it also reduces the operating space, enhancing the importance of the constrained problem.

It can be seen that the operation and control of this HEN problem is a complex problem which includes many possibly conflicting process requirements that are difficult to satisfy. The traditional QDMC algorithm is computationally intensive, which not only increases the computational burden but also is relatively difficult to implement. By analyzing the dynamic behavior of the network and taking in account the energy efficiency, it is observed that the best pairing of manipulated and controlled variables is to control:

- $T_{h_1}^{out}$ with s_1 as primary variable and s_3 as secondary variable,
- $T_{h_2}^{out}$ with x_3 as primary variable and s_2 and s_3 as secondary variables,
- $T_{c_1}^{out}$ with x_1 as primary variable and s_3 as secondary variable, and
- $T_{c_2}^{out}$ with x_2 as primary variable.

With the proposed distributed MPC algorithms based on Nash optimality (communication and cooperative based MPCs), first of all divide the whole system into four agents, they are

- **Agent 1:** Services S_1, S_2 and S_3,
- **Agent 2:** Heat-exchanger I_3 and service S_3,
- **Agent 3:** Heat-exchangers I_1 and I_3, and service S_2,
- **Agent 4:** Heat-exchangers I_2.

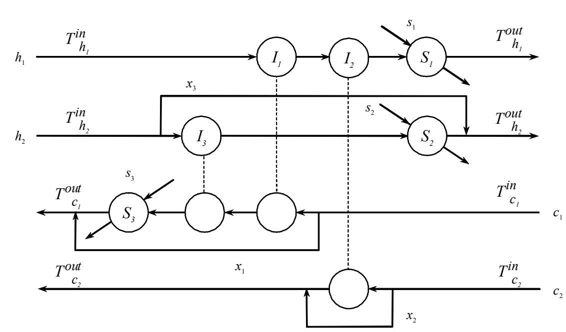

Fig. 3 HEN used for testing and comparing the control strategies.

TABLE I: STREAM CONDITIONS FOR THE NOMINAL OPERATION POINT

Stream	T^{in}	T^{out}	$w\ c\ (Kw/C)$
H_1	90	40	50
H_2	130	100	20
C_1	30	80	40
C_2	20	40	40
S_1	15	---	35
S_2	30	---	30
S_3	200	---	10

An interactive dynamic simulator of heat exchanger networks has been used to test the above example. The simulator is based on a nonlinear model of shell-and-tubes heat exchangers previously reported by Correa and Marchetti [23]. The numerical experience reported here repeats the disturbance and setpoint change sequence used for testing the optimizing and control system presented by Aguilera and Marchetti [8].

Figure 4 shows the dynamic responses obtained using the predictive control schemes for the following scenario: after running at the nominal operating point, $T_{h_1}^{in}$ changes from 90 to 80 °C; 10 min later $T_{h_2}^{in}$ goes from 130 to 140 °C, and after another 10 min $T_{c_1}^{in}$ in changes from 30 to 40 °C. The capability of the decentralized MPC schemes for disturbance rejection can be evaluated by inspection of Figure 4. The worse performance of the communication based MPC is observed during the time period between the first and second load changes, most notably on temperature $T_{h_1}^{out}$. The first fact to be noted in Figure 5 is that under nominal steady-state conditions, the energy integration of centralized and cooperative MPC schemes is similar. However, the communication based MPC provides a poor integration. This is happen due to the characteristics of the algorithm and the objective function employed to build this control scheme, which can not achieve the optimal solution [24]. After the first load change occurs, the system transfer control of h_1 to heater flow rate w_{s_3}. Note that the stationary condition obtained using the decentralized MPC schemes after the first load change yields better heat integration than the centralized MPC. This improvement is due to the elimination of back-off from $w_{s1}=0$ and $x_3=0$.

Under nominal steady-state conditions, the bypass x_3 is completely closed and $T_{h_2}^{out}$ is controlled by the flow rate service w_{s_2}. S3 is also inactive since no heating service is necessary at this point. After the first load change occurs, both control variables w_{s_1} and x_2 fall rapidly. Under this operational condition, the control schemes use the heater w_{s_3}. The dynamic reaction of the heater to the cool disturbance is also stimulated by the bypass x_2 when it falls. After the initial effect is compensated, the control of

TABLE II: EFFECTIVE HEAT TRANSFER AREAS

Equipment	I_1	I_2	I_3	S_1	S_2	S_3
UA (kW/°C)	80	50	20	30	20	10

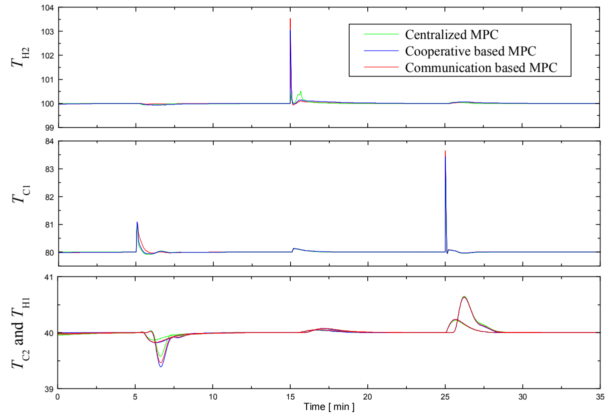

Fig. 4. System responses for disturbances changes.

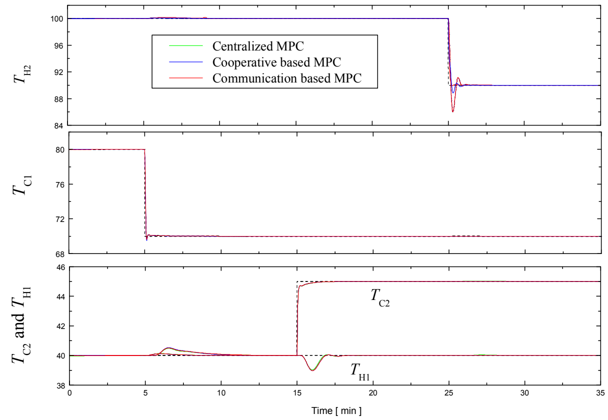

Fig. 6. System responses for setpoint changes.

$T_{c_2}^{out}$ through x_2, which never saturates, while w_{S3} takes complete control of $T_{h_1}^{out}$. Furthermore, the cool perturbation also affects the process stream h_2, where the cooler s_2 is effectively taken out of operation by the bypass x_3. The heat integration of the centralized and cooperative schemes after the second load change appears to be quite similar for both cases being analyzed.

Now consider the sequence of set point changes is as follows: first the target for $T_{c_1}^{out}$ changes from 80 to 70 °C; 10 min later the set point for $T_{c_2}^{out}$ changes from 40 to 45 °C, and after another 10 min $T_{h_2}^{out}$ is taken to 90 from 100 °C. As for the case of load changes, the control performance can be evaluated observing the temperature evolutions in Figure 6. The capability of decentralized MPC schemes for setpoint tracking can be evaluated by inspection of Figure 6. The worse performance is observed in the communication based MPC during the time period after the third setpoint change, most notably on temperature $T_{h_2}^{out}$. The reason for this behavior is a change of the variable in charge of controlling $T_{h_2}^{out}$ and the effect of interactions had not completely take into account by the control scheme. In this case, the control is transferred from x_3 to ws_2, and the communication based MPC only consider the local cost. Similar remarks about the energy integration can be made.

Finally, we analyze the effect of the decentralization scheme on the computational and communication time for each sample. The decentralized MPC schemes are less demanding than the centralized MPC scheme. It should be pointed out that in a networking environment where control schemes are deployed, cooperative and communication based MPC make more efficient use of the resources. Observing the Figure 8 closely, centralized MPC is always computing and communicating for all time of the simulation, this is due to the fact that it is always getting new data and is always computing control actions to mitigate the effect of interactions and disturbance in a centralized fashion. But the decentralized scheme, only use less in communication, mostly at times when there is a significant disturbance or interaction. The communication based MPC computational time can be reduced significantly by modifying the terminating criterion ε of each agent.

To summary, it can be seen that for relatively small structured systems like this example, the centralized MPC is still a strong contender in terms of control performance and cost. But for large-scale systems, the optimization grows in proportion to the cube of the state dimension. This poor scalability makes centralized MPC unattractive for large-

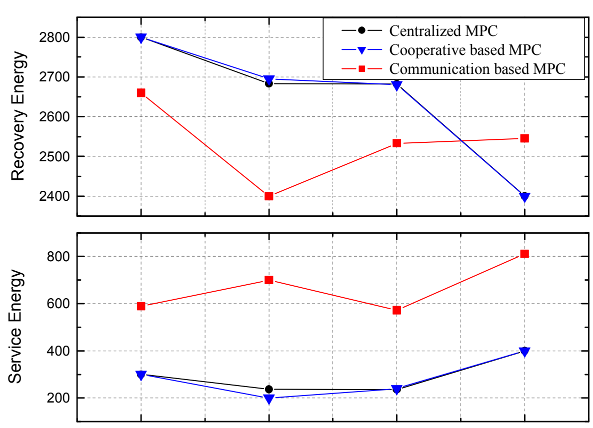

Fig. 5. Steady-state results for disturbance changes.

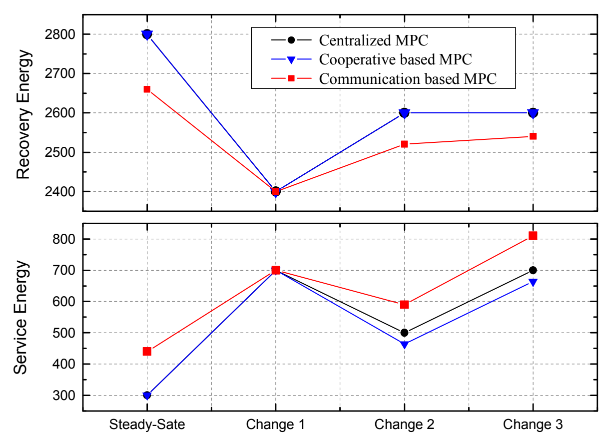

Fig. 7. Steady-state results for setpoint changes.

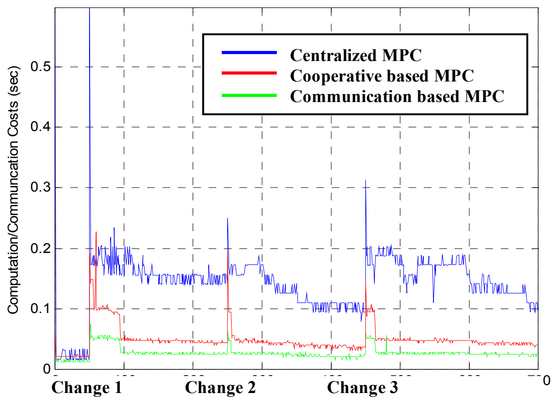

Fig. 8. Computation time for different control schemes.

scale systems.

IV. CONCLUSION

In this study a distributed model predictive control method based on Nash optimality is developed. The MPC is implemented in distributed scheme with the inexpensive agents within the network environment. These agents can co-operate and communicate each other to achieve the objective of the whole system. Coupling effects among the agents are fully taken into account in this scheme, which is superior to other traditional decentralized control methods. The main advantage of this scheme is that the on-line optimization of a large-scale system can be converted to that of several small-scale systems, thus can significantly reduce the computational complexity while keeping satisfactory performance. Furthermore, the design parameters for each agent such as prediction horizon, control horizon, weighting matrix and sample time, etc. can all be designed and tuned separately, which provides more flexibility for the analysis and applications. The second part of this study is to investigate the convergence, stability and performance of the distributed control scheme. These will provide users better understanding to the developed algorithm and sensible guidance in applications.

ACKNOWLEDGMENT

The authors gratefully acknowledge the financial support for this work provided by the Engineering and Physical Science Research Council grant Industrial Non-linear Control and Applications GR/R04683/01.

REFERENCES

[1] E. Kotjabasakis and B. Linnhoff, “Sensitivity tables for the design of flexible processes (1)—how much contingency in heat exchanger networks is cost-effective,” *Chemical Engineering Research*, vol. 64, pp. 199–211, 1986.

[2] J. Cerdá, M. Galli, N. Camussi and M. Isla, “Synthesis of Flexible Heat Exchanger Networks I. Convex Networks,” *Computers & Chemical Engineering*, vol. 14, no. 2, pp. 197-211, 1990.

[3] J. Cerdá, M. Galli, N. Camussi and M. Isla, “Synthesis of Flexible Heat Exchanger Networks. II. Non-Convex Networks for Large Temperature Variations,” *Computers & Chemical Engineering*, vol. 14, no. 2, pp. 213-225, 1990.

[4] K. Papalexandri and E. Pistikopoulos, “Synthesis and retrofit design of operable heat exchanger networks. 1. Flexibility and structural controllability aspects,” *Industrial Engineering and Chemical Research*, vol. 33, no. 7, 1994.

[5] K. Mathisen, S. Skogestad and E. Wolff, “Bypass Selection for Control of Heat Exchanger Networks,” paper presented at ESCAPE-I, Elsinore, Denmark, 1992.

[6] K. Mathisen, M. Morari and S. Skogestad, “Optimal Operation of Heat Exchanger Networks,” presented at Process Systems Engineering, Kyongju, Korea, 1994.

[7] C. Boyaci, D. Uztürk, A. Konukman and U. Akman, “Dynamics and Optimal Control of Flexible Heat exchanger Networks,” *Computers and Chemical Engineering*, vol. 21 Suppl., pp. 775–780, 1996.

[8] N. Aguilera and J. Marchetti, “Optimizing and controlling the operation of heat exchanger networks,” *American Institute of Chemical Engineers Journal*, vol. 44, no. 5, pp. 1090–1104, 1998.

[9] B. Glemmestad, “Optimal Operation of Integrated Processes. Studies on Heat Recovery Systems,” Ph.D. Thesis. Norwegian University of Science and Technology—NTNU, Trondheim, Norway, 1997.

[10] B. Glemmestad, S. Skogestad and T. Gundersen, “Optimal operation of heat exchanger networks,” *Computers and Chemical Engineering*, vol. 23, no 4-5, pp. 509-522, 1999.

[11] K. Mathisen, “Integrated Design and Control of Heat Exchanger Networks,” Ph.D. Thesis. University of Trondheim-NTH, Norway, 1994.

[12] L. Giovanini and J. Marchetti, “Low-level flexible-structure control applied to heat exchanger,” *Computers & Chemical Engineering*, vol. 27, no. 8-9, pp. 1129-1142, 2003.

[13] M. Morari and J. Lee, “Model predictive control: Past, present and future,” *Computers & Chemical Engineering*, vol. 23, no. 4-5, pp. 667-682, 1999.

[14] R. Katebi and M. Johnson, “Predictive control design for large-scale systems,” *Automatica*, vol. 33, no. 3, pp. 421–425, 1997.

[15] N. Ricker and J. Lee, “Nonlinear model predictive control of the Tennessee Eastman challenge process,” *Computer and Chemical Engineering*, vol. 19, no. 9, pp. 961–981, 1995.

[16] A. Zheng, “Reducing on-line computational demands in model predictive control by approximating QP constraints,” *Journal of Process Control*, vol. 9, no. 4, pp. 279–290, 1999.

[17] A. Venkat, J. Rawlings, and S. Wright. “Distributed model predictive control of large-scale systems”, presented at International Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control, Germany, 2005.

[18] T. Keviczky, F. Borrelli and G. Balas. “Stability Analysis of Decentralized RHC for Decoupled Systems”, 44th IEEE Conference on Decision and Control, and European Control Conference, Seville, Spain, December 12-15, 2005.

[19] J. Van Antwerp and R. Braatz, “Model predictive control of large scale processes,” *Journal of Process Control*, vol. 10, no. 1, pp. 1–8, 2000.

[20] L. Giovanini and J. Balderud, “Game Approach to Distributed Model Predictive Control,” *IEEE Control and Decision Conf.*, submitted for publication.

[21] J. Nash, “Non cooperative games,” *Annals of Mathematics*, vol. 54, no. 2, 1951.

[22] R. Neck and E. Dockner, “Conflict and cooperation in a model of stabilization policies: A differential game approach”, *J. Econ. Dyn. Cont.*, vol. 11, pp. 153–158, 1987.

[23] D. Correa and J. Marchetti, “Dynamic simulation of shell and-tube heat exchangers,” *Heat Transfer Engineering*, vol. 8, no. 1, pp. 50-59, 1987.

[24] P. Dubey and J. Rogawski “Inefficiency of smooth market mechanisms”, *J. Math. Econ.*, vol. 19, pp. 285–304, 1990.