GAME APPROACH TO DISTRIBUTED MODEL PREDICTIVE CONTROL

Leonardo Giovanini and Jonas Balderud
Industrial Control Centre
University of Strathclyde
50 George Street, Glasgow G1 1QE, UK
leonardo.giovanini@eee.strath.ac.uk, j.balderud@eee.strath.ac.uk

ABSTRACT
In this paper, a novel distributed model predictive control scheme based on Nash optimality is presented for large-scale processes, in which the on-line optimization of the whole system is decomposed into that of several small co-operative subsystems in distributed structures. The relevant computational convergence, closed-loop performance and the nominal stability for distributed model predictive control are analyzed. The control problem is illustrated to verify the effectiveness and practicality of the proposed control algorithm

KEY WORDS
Model predictive control, distributed control system, Nash optimality

1. Introduction

In model predictive control (MPC), also called receding horizon control, the control input is obtained by solving a discrete-time optimal control problem over a finite horizon, producing an optimal open-loop control input sequence. The first control in that sequence is applied. At the next sampling instant, a new optimal control problem is formulated and solved based on the new measurements. The theory of MPC for linear systems is well developed; nearly all aspects, such as stability, feasibility, optimality, robustness and nonlinearity, have been discussed in the literature (see, e.g., [5], [15], [14], [16], [17]). MPC is very popular in the process control industry because the control objectives and operating constraints can be represented explicitly in the optimization problem that is solved at each control instant. 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 [17]. 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) [12], [21], [27]. More recently Van Antwerp and Braatz [25] 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.

With the rapid development of communication network and the field-bus technology, centralized control has not been a sole structure in applications and gradually replaced by distributed control. Distributed control structure brings new requirements to the traditional control field and allows the conceivability of new challenging control applications. For economic consideration and also no degrading performance, it is desirable to use several inexpensive microcomputers to replace a very high performance computer in control systems. In distributed or decentralized control schemes the local control inputs are computed using local measurements and reduced-order models of the local dynamics [24]. Previous work on distributed MPC is reported in [1], [2], [4], [8], [9], [20], [22]. The proposed algorithms use a wide variety of approaches, including multi-loop ideas [20], decentralized computation using standard coordination techniques [8], [9], [22], robustness to the actions of others [10], [11], [6], penalty functions [23], [26], and partial grouping of computations [13]. The key point is that, when decisions are made in a decentralized fashion, the actions of each subsystem must be consistent with those of the other subsystems, so that decisions taken independently do not lead to a violation of the coupling constraints. The decentralization of the control is further complicated when disturbances act on the subsystems making the prediction of future behavior uncertain.

In this paper, we consider situations where the distributed controllers can exchange information several times every sample. The objective is to achieve some degree of coordination among agents that are solving MPC problems with locally relevant variables, costs, and constraints, but without solving a centralized MPC problem. Such coordination schemes are useful when the local optimization problems are much smaller than a centralized problem. These schemes are also useful in applications where a centralized controller is not appropriate or feasible because, although some degree of coordination is desired, the subsystems cannot divulge all the information about their local models and objectives (e.g. deregulated power markets). In distributed control, the type of coordination that can be realized is determined by the information structure; that is, the connectivity and capacity of the communication network. Here we assume that the connectivity of the communication network is sufficient for the subsystems to obtain information regarding all the variables that appear in their local problems. In this case, we are interested in identifying conditions under which the agents can perform multiple iterations to find solutions to their local optimization problems that are consistent in the sense that all shared variables converge to the same values for all the agents. We also show that when convergence is achieved using this type of coordination, the solutions to the local problems collectively solve an equivalent, global, multiobjective optimization problem. In other words, the coordinated distributed computations solve an equivalent centralized MPC problem. This means that properties that can be proved for the equivalent centralized MPC problem (e.g., stability) are enjoyed by the solution obtained using the coordinated distributed MPC implementation. The significance of proposed distributed control scheme is to reduce the computational burden in complex large-scale systems. Also it can be extended to the remote control and multi-agent systems.

The paper is organized as follows. In Section 2, a distributed MPC algorithm based on Nash optimality is proposed. In Section 3, the convergent condition of the distributed predictive control algorithm for linear models is analyzed. The nominal stability and the performance deviation under communication failure are analyzed, respectively, in Sections 4 and 5. A simulation example is provided to demonstrate the efficiency of the distributed MPC algorithm in Section 6. Conclusions are given in Section 7.

2. Distributed Model Predictive Control

2.1 Model Predictive Control

Model predictive control (MPC) is formulated as resolving an on-line open loop optimal control problem in moving horizon style. 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:

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

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.

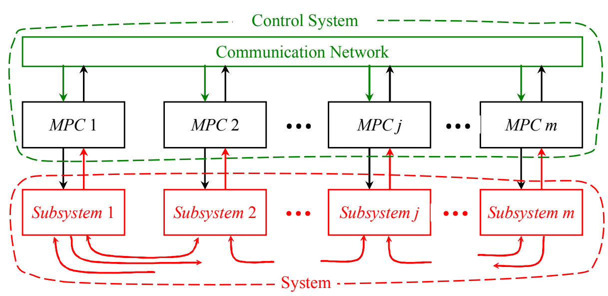

Figure 1: Distributed control system architecture

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, this work proposes a distributed scheme with inexpensive computers under network environment.

2.2 Distributed Model Predictive Control based on Nash optimality

The main idea of the distributed model predictive control algorithm is the on-line optimization of MPC. The distributed control system architecture diagram is shown in Figure 1. Since an optimization formulation can be decomposed into a number of small-scale optimizations. These autonomous subsystems are connected via network with dynamic input coupling among them, 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

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

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

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

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 [18].

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

$$\begin{aligned}&J_l\left(X(k),U_l^q(k),U_{n\neq l}^q(k)\right)\le\\ &J_l\left(X(k),U_l(k),U_{n\neq l}^q(k)\right), n=1,\dots,m\end{aligned} \tag{6}$$

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.

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 $q=0$;

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

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

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

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

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

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

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

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

Step 4: Computes the instant control law

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

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

Two decentralized MPC schemes can be derived from this algorithm, depending on the cost function minimized by the optimization problem. If the cost function only includes local information

$$J_l^*\left(X_l(k),U_l(k),U_{n\neq l}(k)\right) \quad l,n=1,\dots,m, \tag{7}$$

the resulting MPC algorithm is called **communication based MPC**. On the other case, if the cost function includes global information

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

the resulting MPC algorithm is called **cooperation based MPC**.

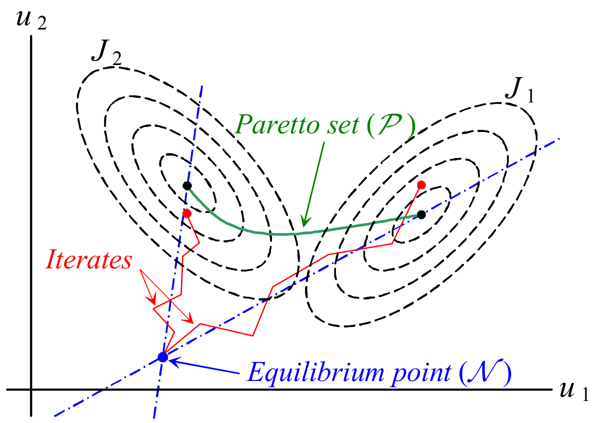

Figure 2. Trajectories in decision space traced by two subsystems.

3. Convergence analysis

Consider this distributed model predictive control of linear dynamic plants. At sampling time instant k, the output prediction model of the lth agent can be described as

$$\hat{y}(k+j,k) = C_l A_l^j x(k) + \sum_{i=1}^{j} h_{il} u_l(k+j-i) + \sum_{\substack{n=1 \\ n\neq l}}^{m} \sum_{i=1}^{j} h_{in} u_n(k+j-i) \tag{9}$$

and the predicted trajectories $\hat{Y}(k) = [\,\hat{y}(k+1,k) \cdots \hat{y}(k+V,k)]^T$ is given by

$$\hat{Y}(k) = \Gamma x(k) + \mathcal{A}_l U_l(k) + \sum_{\substack{n=1 \\ n\neq l}}^{m} \mathcal{A}_n U_n(k) \tag{10}$$

where $\mathcal{A}_l \quad l = 1,\ldots,m$ is the dynamic matrices [15]. Then, the performance index of the lth agent can be written as

$$J_l\left(X(k), U_l(k)\right) = \left\| R(k) - \hat{Y}(k) \right\|_{\mathcal{Q}_l}^2 + \left\| U_l(k) \right\|_{\mathcal{R}_l}^2 \tag{11}$$

where $R(k) = [\,r(k) \cdots r(k+V)\,]^T$ is the setpoint trajectory. According to extremum condition for Nash optimality, a necessary condition

$$\frac{\partial J\left(X(k), U(k)\right)}{\partial U_l(k)} = 0 \quad l = 1,\ldots,m,$$

leads to the optimal solution for the l:th agent which gives,

$$U_l(k) = \mathcal{K}_l\left[R(k) - \Gamma x(k) - \sum_{\substack{n=1 \\ n\neq l}}^{m} A_{ln} U_n(k) \right] \tag{12}$$

with,

$$K_l = \left(A_l^T \mathcal{Q}_l A_l + \mathcal{R}_l \right)^{-1} A_l^T \mathcal{Q}_l \quad l = 1,\ldots,m. \tag{13}$$

the overall solution is ,

$$U^q(k) = \mathcal{K}_1\left[R(k) - \Gamma x(k) \right] + \mathcal{K}_0 U^{q-1}(k), \tag{14}$$

where,

$$\mathcal{K}_1 = \begin{bmatrix} K_1 & & 0 \\ & \ddots & \\ 0 & & K_m \end{bmatrix} \tag{15}$$

At each sample, the setpoint $R(k)$ and the system state $x(k)$ are known and do not depend on $U(k)$. Therefore, the first term of (14) is irrelevant to convergence property of the solution. Then, the solution is given by

$$U^q(k) = \mathcal{K}_0 U^{q-1}(k) + \mathcal{C}_0. \tag{16}$$

From this expression is easy to see that the behavior of the distributed problem during the iterative process is determined by the behavior difference equation (16). Then, the convergence of the distributed MPC algorithm is related with stability of (16) given as

$$\left\| \rho\left(\mathcal{K}_0\right) \right\|_1 \leq 1. \tag{17}$$

Therefore, the way that the global problem is split and the parameters $M, V, \mathcal{Q}_l$ and $\mathcal{R}_l \quad l = 1,\ldots,m$ should be tuned to ensure the convergence of the decentralized algorithm. This convergence analysis can be extended to constrained systems using Lyapunov arguments.

4. Performance Analysis

In distributed control, each agent can work independently to achieve its local objective, but can not accomplish the entire objective on its own. To this purpose, the subsystems communicate, coordinate and negotiate with each other, exchanging information through a communication network. The question that naturally emerges is how the decomposition of the global objective affects the performance compared with the solution of the global cost function.

Applying the necessary conditions for extremum for the global cost,

$$\frac{\partial J\left(X(k), U(k)\right)}{\partial U(k)} = 0$$

which implies,

$$\frac{\partial J}{\partial U_l(k)} = -2\frac{\partial U(k)}{\partial U_l(k)} \mathcal{A}^T \mathcal{Q} \hat{E}^0(k) + 2\frac{\partial U(k)}{\partial U_l(k)} \mathcal{K} U(k) = 0 \qquad l = 1 \ldots m \tag{18}$$

The elements of $\partial J / \partial \mathrm{U}_l$ are zero for all elements except the l : th element,

$$\frac{\partial U(k)}{\partial U_l(k)} = [0 \cdots 0\; I\; 0 \cdots 0] \quad l = 1,\ldots,m,$$

then, equation (18) can be rewritten as follows,

$$-2\left[\mathcal{A}_{1l} \cdots \mathcal{A}_{ml}\right] \mathcal{Q} \hat{E}^0(k) + 2\left[k_{1l} \cdots k_{ml}\right] U(k) = 0 \quad l = 1,\ldots,m. \tag{19}$$

The solution of this system of linear equation is the optimal solution, which belongs to the Pareto set.

Applying the necessary conditions to cooperation based MPC where

$$\frac{\partial J_l^{\#}\left(X(k),U_l(k),U_{n\neq l}(k)\right)}{\partial U_l(k)}=0 \quad l,n=1,\ldots,m$$

which is the same condition for optimality of the global cost. Therefore, if the global cost $J_l(X(k),U(k))$ is optimized in each agent the solution of the decentralized scheme is globally optimal and belongs to the Pareto set (see Figure 3).It should be pointed out that the centralized solution is always Pareto optimal.

However, if the communication based MPC is employed, the necessary condition leads to

$$\begin{aligned}&\frac{\partial J_l^{*}}{\partial U_l(k)}=-2A_l^T Q_l \hat{E}_l^0(k)+2K_l U_l(k)=0\\&l=1,\ldots,m\end{aligned} \tag{20}$$

Comparing conditions (18.*a*) and (20), it is clear that when the agents minimize the local cost $J(X(k),U_l(k))$ $l=1,\ldots,m$ the obtained solution is different from the optimal solution and they do not belong to the Pareto set but rather to a Nash equilibrium set.

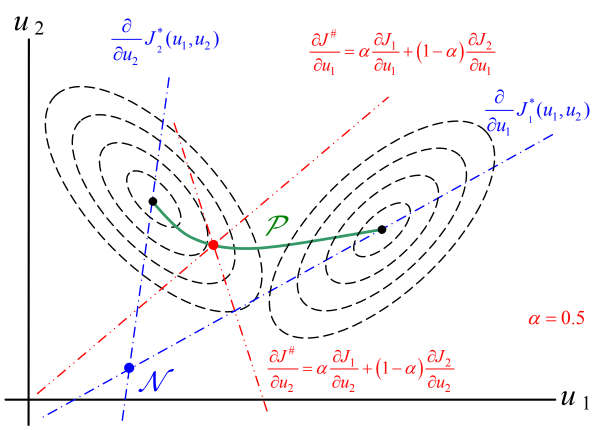

Figure 3: Position of attractors for different cost functions.

Depending on the cost function minimized, $J_l^*(\cdot)$ or $J_l^{\#}(\cdot)$ (equations **Error! Reference source not found.** and**Error! Reference source not found.**), the Nash equilibrium point $\mathcal{N}$ can be located on the Paretto set $\mathcal{P}$,if the global cost function is minimizes, or not if the local cost function is minimizes [7], [19] (see Figure 3).

5. Nominal Stability

In order to analyze the nominal stability, integral Nash optimal solution of the whole system, provided that the algorithm is convergent at each sampling time, can be written as follows,

$$U(k)=\mathcal{K}\left[R(k)-\Gamma x(k)\right] \tag{21}$$

which leads to the closed-loop state-space model

$$x(k+1)=\left(A-B\mathcal{K}\Gamma\right)x(k)+B\mathcal{K}R(k) \tag{22}$$

This expression shows the state mapping relationship of the distributed system between the time instant k and k+1. The nominal stability of the whole distributed system can be guaranteed, if and only if

$$\left\|\lambda\left(A-BS\left(I-\mathcal{K}_0\right)^{-1}\mathcal{K}_1\Gamma\right)\right\|\leq 1. \tag{23}$$

That is, the norms of eigenvalues of state mapping are less than 1. This stability analysis can be extended to constrained systems using Lyapunov arguments.

6. Simulation and Results

Let consider the following transfer function matrix,

$$\begin{bmatrix}17.3\frac{e^{-4.8s}}{23.8s+1} & 20.6\frac{e^{-61.3s}}{38.8s+1} & 19.9\frac{e^{-28.9s}}{25.4s+1} & 0\\ 0 & 4.6\frac{e^{-50.4s}}{48.4s+1} & 0 & 79.1\frac{31.4s+0.8}{31.4s+1}\\ 0 & 24.4\frac{48.2s^2+4.0s+0.05}{48.2s^2+3.9s+0.06} & 0 & -8.4\frac{e^{-18.79s}}{27.9s+1}\\ 0 & 16.9\frac{e^{-24.7s}}{39.5s+1} & -39.2\frac{22.8s+0.8}{22.8s+1} & 0\end{bmatrix}$$

obtained from the linearization of heat exchange network (HEN) employed by Aguilera and Marchetti [3] to test centralized optimization schemes for HENs (for more details on HEN design and control see [3] and the references in it). The main characteristic of this system are interactions between different inputs, which propagate across the systems. The system inputs are constrained to

$$\begin{aligned}-1&\leq u(k)\leq 1,\\-1&\leq \Delta u(k)\leq 1,\end{aligned}$$

For this system, the decomposition was carried after consideration of the multi-loop and determining which input affects the corresponding output most. The parameters of the MPC controllers are

$$\begin{aligned}&T_S=1\,\text{sec}; N=300; V=50; M=5;\\&\mathcal{Q}_l=I_{V\times V};\mathcal{R}_l=r_l I_{M\times M};\quad l=1,2,3,4;\\&r_1=20; r_2=25; r_3=10; r_4=5.\end{aligned}$$

The simulation was run for *10* min with the following setpoint changes: Firstly, T_{C_1} goes from 0 to –5 at 1 min; then T_{C_2} goes from 0 to 5 at 4 min. Finally, T_{H_2} goes from 0 to –10 at 7 min. In this set of simulations three MPC schemes are compared: the classical centralized MPC (**CPMC**), the *communication based MPC* (**DMPC** $_1$) and the *cooperation based MPC* (**DMPC** $_2$).

Figure 4 shows the behavior of each agents cost function during the iterative process, which is compared with the constant cost of CMPC. It can be seen that the agents modify their cost dynamically as they approach the global cost (CMPC). As the terminal criterion is reduced more iterations occurs amongst the agents and the cost converges with the cost of CMPC and vice versa (Figure 4.*a*). The key point here is that this iteration process can be terminated even sooner (intermittently) and still the agent s will have almost reached the global cost. This becomes another tuning parameter especially for improving the quality of the control and also for speed. The oscillatory nature of the DMPC2 cost function is due to the interactions amongst the agents as the find a common objective.

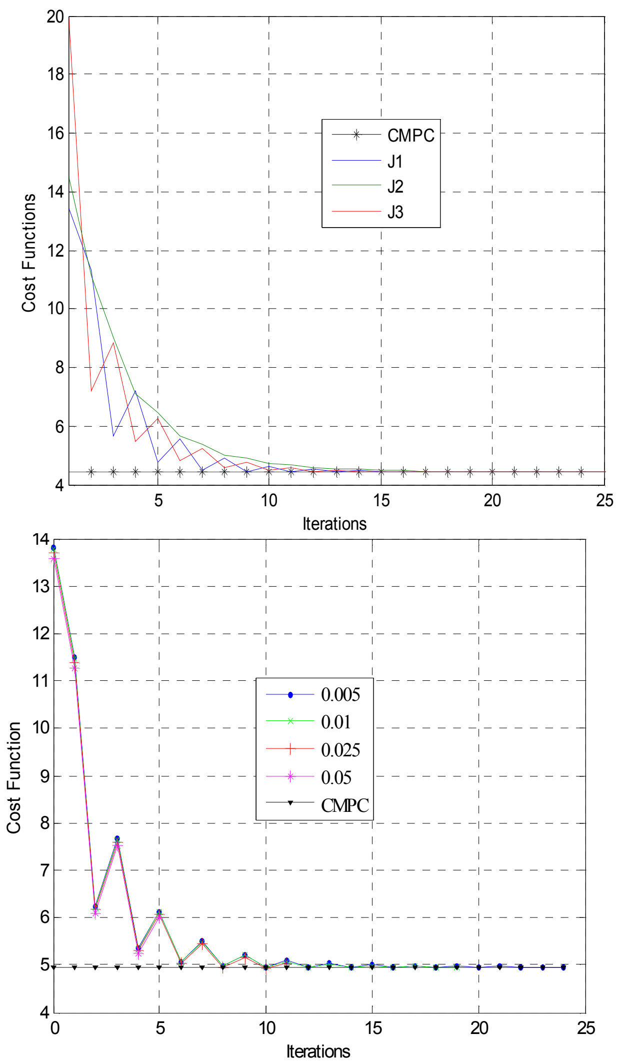

Figure 4: *a*) Behavior of each agent's cost function at time 50 *sec*, b) for different stopping criteria.

Finally, we analyze the effect of the decentralization scheme on the computational and communication time. Considering Figure 5 and based on the definitions given earlier, $DMPC_2$ is seen to have the highest demands for communication as well as computation. $DMPC_1$ is slightly more expensive that CMPC. Well having said that; it should be pointed out that in a networking environment where these schemes are deployed, $DMPC_2$ and DMPC1 make more efficient use of the communication resource. Observing the Figure closely, CMPC 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 DMPC2 computational time can be reduced significantly by modifying the terminating criterion ε_l $l = 1, \ldots, m$ of each agent.

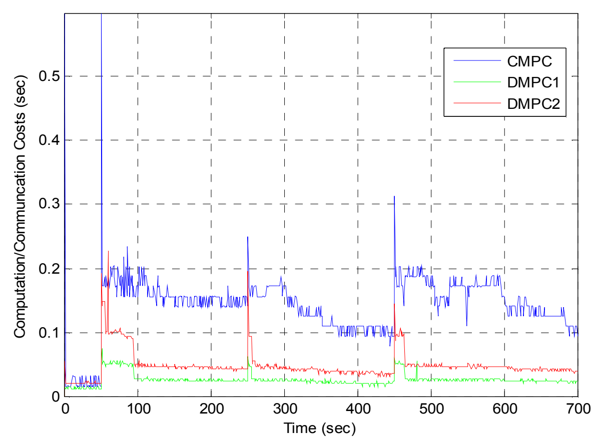

Figure 5: Computation time for different control schemes

To summary, it can be seen that for relatively small structured systems like this example, the CMPC is still a strong contender in terms of control performance and cost. But as the size of the system increases, the scaling become very poor as will be seen later.

7. Conclusions

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.

To summary, it can be seen that for relatively small structured systems like this example, the CMPC is still a strong contender in terms of control performance and cost. But as the size of the system increases, the scaling become very poor as will be seen later.

References

[1] L. Acar, Some examples for the decentralized receding horizon control, *Proc. 31st Conf. Decision and Control*, Tucson, AZ, 1992, pp. 1356-1359.

[2] M. Aicardi, G. Casalino, R. Minciardi, and R. Zoppoli, On the existence of stationary optimal receding-horizon strategies for dynamic teams with common past information structures, *IEEE Trans. Automat. Contr.*, vol. 37, pp. 1767-1771, Nov. 1992.

[3] N. Aguilera and J. Marchetti. Optimizing and controlling the operation of heat exchanger networks, *AIChe Journal*, vol. 44(5), pp. 1090–1104, 1998.

[4] M. Baglietto, T. Parisini, and R. Zoppoli, Neural approximators and team theory for dynamic routing: A receding-horizon approach, *Proc. 38th Conf. Decision and Control*, Phoenix, AZ, 1999, pp. 3283-3288.

[5] A. Bemporad and M. Morari, Robust model predictive control: A survey, *Robustness in Identification and Control* (Lecture Notes in Control and Information Sciences), vol. 245. New York: Springer-Verlag, pp. 207-226, 1999.

[6] E. Camponogara, E. Jia, B. Krogh and S. Talukdar, Distributed model predictive control, IEEE Control System Magazine, pp. 44-52, 2002.

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

[8] H. El Fawal, D. Georges, and G. Bornard, Optimal control of complex irrigation systems via decomposition-coordination and the use of augmented Lagrangian, *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, vol. 4, San Diego, pp. 3874-3879, 1998.

[9] M. Gomez, J. Rodellar, F. Vea, J. Mantecon, and J. Cardona, Decentralized predictive control of multi reach canals, *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, San Diego, pp. 3885-3890, 1998.

[10] D. Jia and B. Krogh, Distributed model predictive control, *Proc. American Control Conf.*, Arlington, pp. 2767-2772, 2001.

[11] D. Jia and B. Krogh, Min-max feedback model predictive control for distributed control with communications, *Proc. American Control Conf.*, Anchorage, pp. 4507-4512, 2002.

[12] R. Katebi and M. Johnson, Predictive control design for large-scale systems, *Automatica*, vol. 33(3), pp. 421–425, 1997.

[13] T. Keviczky, F. Borrelli and G. J. Balas, Model Predictive Control for Decoupled Systems: A Study on Decentralized Schemes, *Proc. American Control Conf.*, 2004.

[14] B. Kouvaritakis and M. *Cannon. Non-linear Predictive Control: theory and practice*, Control Engineering Series, vol. 61, 2001.

[15] J. Maciejowski, *Predictive Control with Constraints*. Prentice Hall, England, 2002.

[16] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, Constrained model predictive control: Stability and optimality, *Automatica*, vol. 36(6), pp. 789-814, 2000.

[17] M. Morari and J. Lee, Model predictive control: Past, present and future, *Comput. Chem. Eng.*, vol. 23(4-5), pp. 667-682, 1999.

[18] J. Nash, Non cooperative games, Annals of Mathematics, vol. 54(2), 1951.

[19] 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.

[20] S. Oschs, S. Engell, and A. Draeger, Decentralized vs. model predictive control of an industrial glass tube manufacturing process, in *Proc. 1998 IEEE Int. Conf. Control Applications*, Italy, pp. 16-20, 1998.

[21] N. Ricker and J. Lee, Nonlinear model predictive control of the Tennessee Eastman challenge process, *Computer and Chemical Engineering*, vol. 19(9), pp. 961–981, 1995.

[22] S. Sawadogo, R. Faye, P. Malaterre, and F. Mora-Camino, Decentralized predictive controller for delivery canals, in *Proc. 1998 IEEE Int. Conf. Systems, Man, and Cybernetics*, San Diego, CA, pp. 3880-3884, 1998.

[23] D. Shim, H. Kim and S. Sastry, Decentralized Nonlinear Model Predictive Control of Multiple Flying Robots, *42nd IEEE CDC*, Maui, p 3621-3625, December 2003.

[24] D. Siljak, Decentralized control and computations: Status and prospects, *Annu. Rev. Contr.*, vol. 20, pp. 131-141, 1996.

[25] J. Van Antwerp and R. Braatz, Model predictive control of large scale processes, *Journal of Process Control*, vol. 10(1), pp. 1–8, 2000.

[26] S. Waslander, G. Inalhan and C. Tomlin, Decentralized Optimization via Nash Bargaining, *Conf. on Cooperative Control and Optimization*, 2003.

[27] A. Zheng, Reducing on-line computational demands in model predictive control by approximating QP constraints, *Journal of Process Control*, vol. 9(4), pp. 279–290, 1999.