Published in IET Control Theory and Applications
Received on 30th October 2010
Revised on 22nd March 2011
doi: 10.1049/iet-cta.2010.0634

ISSN 1751-8644

Game approach to distributed model predictive control

L. Giovanini

National Council for Scientific and Technological Research and the Centre for Signals, Systems and Computational Intelligence, Faculty of Engineering and Water Sciences, Universidad Nacional del Litoral, Santa Fe, Argentina
E-mail: lgiovanini@fich.unl.edu.ar

Abstract: This study introduces a framework for distributed model predictive control (MPC) based on dynamic games, where centralised and decentralised control algorithms can be viewed as dynamical games with coupled control sets. The original optimisation problem is decomposed into smaller coupled optimisation problems in a distributed structure, which is solved iteratively. Then, the resulting dynamic game is analysed using the theory of potential games to derive the properties of the resulting algorithms. This sheds new light on the properties of existing MPC algorithms and allows us to establish a unified framework to analyse them. The control problem of a heat-exchanger network (HEN) is used to illustrate the effectiveness, practicality and limitations of the proposed framework.

1 Introduction

Model predictive control (MPC) is widely recognised as a high performance, yet practical, control technology. This model-based control strategy solves at each sample a discrete-time optimal control problem over a finite horizon, producing a control input sequence. An attractive attribute of MPC technology is its ability to systematically account for system constraints. The theory of MPC for linear systems is well developed; all aspects such as stability, robustness, feasibility and optimality have been extensively discussed in the literature (see e.g. [1–4]). The effectiveness of MPC depends on the model accuracy and the availability of fast computational resources. These requirements limit the application base for MPC. Even though, applications abound in process industries [5], manufacturing [6], supply chains [7], among others, are becoming widespread.

Two common paradigms for solving system-wide MPC calculations are centralised and decentralised strategies. Centralised strategies may arise from the desire to operate the system in an optimal fashion, whereas decentralised MPC control structures can result from the incremental roll-out of the system development. An effective centralised MPC can be difficult, if not impossible to implement in large-scale systems [8, 9]. In decentralised strategies, the system-wide MPC problem is decomposed into subproblems by taking advantage of system structure, and then, they are solved independently. In general, decentralised schemes approximate the interactions and treat inputs from other subsystems as external disturbances. These assumptions lead to poor system performance [10, 11], and stability problems for strongly coupled systems. Therefore there is a need for a cross-functional integration between decentralised controllers, in which a coordination level performs steady-state target calculation [12–16].

Several distributed MPC formulations are available in the literature. A distributed MPC framework was proposed by Dumbar and Murray [17] for the class of systems that have independent subsystem dynamics but link through their cost functions and constraints. Then, Dumbar [18] proposed an extension of this framework that handles systems with weakly interacting dynamics. Stability is guaranteed through the use of a consistency constraint that forces the predicted and assumed input trajectories to be close to each other. The resulting performance is different from centralised implementations in most of cases. Distributed MPC algorithms for unconstrained and linear time-invariant (LTI) systems were proposed in [19–22]. In [19, 20] the evolution of the states of each subsystem is assumed to be only influenced by the states of interacting subsystems and local inputs, whereas these restrictions were removed in [21–23]. This choice of modelling restricts the system where the algorithm can be applied, because in many cases the evolution of states is also influenced by the inputs of interconnected subsystems. More critically for these frameworks is the fact that subsystems-based MPCs only know the cost functions and constraints of their subsystem.

Recent works, [24–28], have been focused on systems dynamically coupled with constraints that share inputs from other subsystems. The authors employed different approaches to deal with coupling constraints. Wakasa *et al.* [25] proposed a decentralised MPC method based on a dual decomposition technique. The centralised MPC problem is formulated as a convex optimisation problem, then the dual problem with decoupled equality constraints is derived using a decomposition technique. A projected subgradient method is used to solve the dual problem, which leads to a decentralised algorithm. On the other side, Venkat *et al.* [26], Venkat [29] and Camponaroga and Olivera [27] decomposed the quadratic MPC problem into a set of local subproblems that are solved iteratively by a network of agents. These works only differ in the way that they solve the local problems: Venkat *et al.* [26] and Venkat [29]

developed an algorithm based on the Jacobi implementation of the best response algorithm whereas Camponaroga and Olivera [27] developed an algorithm based on the method of feasible directions, embedding the computations and communications. Doan *et al.* [28] developed a distributed MPC algorithm based on a distributed version of Han's parallel algorithm. The MPC algorithm relies on local iterative updates only, instead of system-wide information exchange, thus the resulting distributed MPC algorithm is applicable to sparsely coupled linear systems. Finally, Negenborm *et al.* [24] developed a distributed MPC for multi-agent systems. They developed two algorithms to deal with the different degrees of couplings between subsystems: (a) A 'serial implementation' for highly coupled systems resulting from the use of the Block Coordinate Descent algorithm [30] and (b) a 'parallel implementation' for weakly coupled systems resulting from the application of the Auxiliary Problem Principle [31]. The serial approach is based on a Gauss–Seidel scheme that minimises the objective function in a sequential way: one agent after another minimises its local local cost function, assuming that the remaining variables are fixed, and communicate the solutions to other agents. On the other hand, the parallel scheme is based on a Jacobi scheme: the agents perform a series of parallel iterations in which all agents compute the solutions at the same time.

The aim of this work is to develop a theoretical framework that unifies the different approaches to decentralised and distributed MPC for *LTI* systems with general dynamic couplings and convex coupled constraints, allowing an unified analysis of the algorithms reported in literature. We will show that even centralised MPC can be modelled as a non-cooperative game, which further have the potential type structure introduced by Monderer and Sharpley [32]. These type of games have been shown to posses important properties: (i) a Nash equilibrium exist, (ii) it can be attained using greedy best-response type algorithms and (iii) there exist an equivalent centralised implementation. Thus, we use the theory of the potential games to establish the properties of the resulting game. In this way, the original optimisation problem is recasted as a multiobjective optimisation problem, where the contribution of each agent is explicitly measured, then the problem is transformed into a dynamic game of local optimisation problems. Distributed and decentralised MPC algorithms reported in the literature can be obtained for different values of the cost function weights and the shared information constraint, corresponding to potential games with different potential function. Finally, the properties of the resulting algorithms are analysed using the potential function corresponding to each game.

The paper is organised as follows. In Section 2, the distributed MPC framework based on dynamic games is proposed. Then, constrained potential games are studied in Section 3. The conditions for the existence and uniqueness of game's Nash equilibrium are analysed using a Lagrangian framework. The different distributed MPC algorithms emerging from the proposed framework are analysed in Section 4. Depending on network connectivity and the structural information of the performance index and constraints available, different distributed control schemes (distributed MPC, coordinated decentralised MPC and decentralised MPC) emerge. The operation of a heat-exchanger network (HEN) is presented in Section 5 to demonstrate the efficiency and limitations of the proposed framework. Conclusions are given in Section 6.

2 Distributed MPC

2.1 Model predictive control

MPC is formulated as solving an on-line open-loop optimal control problem in a receding horizon style. Using the current state $x(k)$, an input sequence $U(k) \triangleq [u^{\mathrm{T}}(k, k)\Lambda u^{\mathrm{T}}(k+N_u-1, k)]$ is calculated to minimise a performance index $J(x(k), U(k))$ while satisfying some specified constraints. The first element of the sequence $u(k, k)$ is taken as controller output, then the control and the prediction horizons recede ahead by one step at the next sampling time. The new measurements are taken to compensate for unmeasured disturbances that cause the system output to be different from its prediction. At instant k, the controller solves the optimisation problem

$$\min_{U(k)} J(x(k), U(k)) \tag{1a}$$

s.t.

$$x(i+1, k) = Ax(i, k) + Bu(i, k), \quad i = k, \ldots, k+N-1, x(k, k) = x(k) \tag{1b}$$

$$u(i, k) \in \mathcal{U}, \quad i = k, \ldots, k+N_u-1 \tag{1c}$$

$$u(i, k) = 0, \quad i = k+N_u, \ldots, k+N-1 \tag{1d}$$

where $x(k) \in R^{n_x}$ is the state vector, $u(k) \in \mathcal{U} \subseteq R^{n_u}$ is the control input vector, N is the prediction horizon, $N_u < N$ is the control input horizon and $\mathcal{U} \subseteq R^{n_u}$ is the set of global admissible controls

$$\mathcal{U} = \{u \in R^{n_u} | Du \leq d, d > 0\}$$

This set is assumed to be non-empty, compact and convex, containing the origin in its interior. In the optimisation problem (1) the cost function $J(x(k), U(k))$ measures the difference between the predicted and the desired future behaviours. Generally, the quadratic index

$$J(x(k), U(k)) = x^{\mathrm{T}}(N, k)Px(N, k) + \sum_{i=0}^{N-1} x^{\mathrm{T}}(i, k)Qx(i, k) + u^{\mathrm{T}}(i, k)Ru(i, k) \tag{2}$$

is employed because of its connection with optimal control literature. It is assumed that $Q = Q^{\mathrm{T}} \geq 0$, $R = R^{\mathrm{T}} > 0$ are square-weights matrices and the terminal weight $P = P^{\mathrm{T}} > 0$ satisfies the Lyapunov equation

$$A^{\mathrm{T}}PA - P = -Q$$

so that the cost function $J(x(k), U(k)) = \sum_{i=0}^{\infty} x^{\mathrm{T}}(i, k)Qx(i, k) + u^{\mathrm{T}}(i, k)Ru(i, k)$ [4]. In many formulations an extra constraint or extra control modes are included into (1) to ensure the robust stability of the closed-loop system. The optimisation problem (1) with the cost function (2) can be recast as a 'quadratic programming' (QP) problem (see e.g. [33]), whose solution $U^*(k) \triangleq [u^{*\mathrm{T}}(k, k) \cdots u^{*\mathrm{T}}(k+N_u-1)] \in \mathcal{U}$ is a sequence of optimal control inputs.

IET Control Theory Appl., 2011, Vol. 5, Iss. 15, pp. 1729–1739
doi: 10.1049/iet-cta.2010.0634

2.2 Game theoretic framework for distributed MPC

Large-scale and distributed systems are generally composed of several interacting subsystems. The interactions can either be (i) dynamic, in the sense that the states and inputs of each subsystem influence the states of the ones to which it is connected, (ii) because of the fact that subsystems share common goals and constraints or (iii) both. Systems of this type admit a decomposition into m subsystems represented by

$$x_l(k+1) = \sum_{p\in\mathcal{N}} A_{lp}x_p(k) + B_{lj\in\mathcal{N}_p}u_{j\in\mathcal{N}_p}(k), \quad l \in \mathcal{N} \tag{3}$$

where $x_l(k) \in R^{n_{x_l}} \subseteq R^{n_x}$ and $u_{j\in\mathcal{N}_l} \in \mathcal{U}_l \subseteq R^{n_{u_l}} \subset R^{n_u}$ are the vectors of local state and input, respectively, $\mathcal{N} = [1, 2, \ldots, m]$ denotes the index set of subsystems and the set of control inputs indices of subsystem l is denoted by $\mathcal{N}_l$. Each subsystem is assumed to have local convex independent and coupled constraints. The set of local admissible controls

$$\mathcal{U}_l = \{u_{j\in\mathcal{N}_l} \in R^{n_{u_l}} | D_l u_{j\in\mathcal{N}_l} \le d_l, d_l > 0\} \quad \forall l \in \mathcal{N}$$

is also assumed to be non-empty, compact and convex, containing the origin in their interior.

Distributed and decentralised control frameworks are based on a set of m independent agents implementing small-scale optimisations for the subsystems, connected through a communication network such that they can share the common resources and coordinate with each other in order to accomplish the control objectives.

Assumption 1: The local states of each subsystem $x_l(k)$ are accessible.

Assumption 2: The communication between the control agents is synchronous.

These assumptions are not restrictive and it was made to simplify the problem. However, recent works remove some of these assumptions [26–28, 34]. In fact, if the local states are not accessible they can be estimated from local outputs $y_l(k)$ and control inputs using a Kalman filter and then communicate it to other agents, therefore Assumption 1 is reasonable. Assumptions 2 implies the fact that the sampling time interval is long enough to allow several (more than two) iterations of the solutions among the control agents to solve the resulting optimisation problem.

Under these assumptions, the global cost function $J(x(k), U(k))$ can be rewritten in terms of the contributions of m subsystems as a multi-objective cost function

$$J(x(k), U(k)) = \sum_{l\in\mathcal{N}} \gamma_l J_l(x(k), U(k)) = J(x(k), U(k), \Gamma) \tag{4}$$

where $\Gamma = [\gamma_l]$, $\gamma_l \ge 0$, $\sum_{l\in\mathcal{N}} \gamma_l = 1$ is the vector of weights, $U_j(k)$ is the jth system input trajectory such that $U(k) = [U_{j\in\mathcal{N}_l}(k), U_{j\notin\mathcal{N}_l}(k)]$ $l \in \mathcal{N}$ and $J_l(x(k), U(k))$ is the contribution of the lth subsystem to the global cost J. This decomposition of the cost function and input variable leads to a decomposition (1) into m coupled optimisation problems

$$\min_{U_{j\in\mathcal{N}_l}(k)} J(x(k), U(k), \Gamma_l), \quad l \in \mathcal{N} \tag{5a}$$

s.t.

$$x(i+1, k) = Ax(i, k) + Bu(i, k), \quad i = k, \ldots, k+N-1, x(k, k) = x(k) \tag{5b}$$

$$u_{j\in\mathcal{N}_l}(i, k) \in \mathcal{U}_l, \quad i = k, \ldots, k+N_u-1 \tag{5c}$$

$$u_{j\in\mathcal{N}_l}(i, k) \in \tilde{\mathcal{U}}_l(u_{j\notin\mathcal{N}_l}(i, k)) \tag{5d}$$

$$u_{j\notin\mathcal{N}_l}(i, k) = \mathbf{u}_{j\notin\mathcal{N}_l}(i, k) \tag{5e}$$

$$u_{j\in\mathcal{N}_l}(i, k) = 0, \quad i = k+N_u, \ldots, k+N-1 \tag{5f}$$

$$u_{j\notin\mathcal{N}_l}(i, k) = 0 \tag{5g}$$

where $\tilde{\mathcal{U}}_l(U_{j\notin\mathcal{N}_l}(k))$ denotes the set of admissible controls given the controls of other agents, $\mathbf{U}_{j\notin\mathcal{N}_l}(k)$ denotes the assumed controls of others agents, determined by information constraint (12), and Γ_l $\forall l \in \mathcal{N}$ are the vector of weights for the lth optimisation problem.

The goal of the decomposition is to reduce the complexity of (1) by ensuring that subproblems (5) are smaller than the original one (fewer decision variables and constraints) while they try retain its properties (stability, feasibility and optimality). The price paid to achieve this aim is the need of coordination between the subproblems (5) during their solution. In this way, the optimisation problem (1) is transformed into a 'dynamic constrained game' of m agents where each of them searches for their optimal decisions $U^*_{j\in\mathcal{N}_l}(k)$ through a sequence of 'strategic constrained games' in response to decisions of other agents.

Definition 1: A strategic constrained game $\mathcal{G}^C = \langle \mathcal{N}, J(x(k), U(k), \Gamma_l), \mathcal{U}_l, \tilde{\mathcal{U}}_l \rangle$ models the interactions between m agents subject to coupled constraints, where the set $\mathcal{U}_l \subseteq \mathcal{U}$ are the available decisions for each agent, the set $\tilde{\mathcal{U}}_l(u_{j\notin\mathcal{N}_l}(k))$ are the available decisions of agent l given the decisions of others agents and $J(x(k), U(k), \Gamma_l)$: $x(k) \times U(k) \to \mathbb{R}$ is the utility function for each agent.

In general, we are interested in determining the choices that agents will make when faced with a particular game, which is sometimes referred to as the 'solution of the game'. We adopt the most common solution concept, known as 'Nash equilibrium' [35]: a set of choices where no individual agent can improve its utility by unilaterally changing its choice.

Definition 2: A group of control decisions $U^*(k) = [U^*_{j\in\mathcal{N}_l}(k), U^*_{j\notin\mathcal{N}_l}(k)]$ is said to be Nash optimal if for all $U_{j\in\mathcal{N}_l}(k) \in \mathcal{U}_l \cap \tilde{\mathcal{U}}_l(U_{j\notin\mathcal{N}_l}(k))$, such that $U_l(k) = [U_{j\in\mathcal{N}_l}(k), U^*_{j\notin\mathcal{N}_l}(k)]$, verifies

$$J(x(k), U^*(k), \Gamma_l) \le J(x(k), U_l(k), \Gamma_l) \quad \forall l \in \mathcal{N} \tag{6}$$

If the Nash optimal solution is achieved, then each subproblem does not change its decision $U_{j\in\mathcal{N}_l}(k)$ because it has achieved an equilibrium point of the coupling-decision process; otherwise the performance index will degrade. Since every agent is trying to minimise his own objective function, a natural approach is to consider an iterative algorithm based, for example, on the Jacobi (simultaneous) or Gauss–Seidel (sequential) schemes,

Algorithm 1

Data: $Q_l,\ R_l,\ q_{max} > 0,\ \varepsilon_l > 0, \phi \gg 1 \quad \forall l \in \mathcal{N}$

Initialise agent l

$q \leftarrow 1, \rho_l \leftarrow \phi\varepsilon_l, U^0(k) = [u(k,k-1)\ \cdots\ u(k+M,k-1)\ 0]$

Measure and communicate the local state $x_l(k)$

for $p \in \mathcal{N}$ *and* $p \neq l$ **do**

Communicate $x_l(k)$ to agent p

end

Compute local control actions $U_{j\in\mathcal{N}_l}(k)$

while $\rho_l > \varepsilon_l$ and $q < q_{max}$ **do**

$q \leftarrow q + 1$

$\tilde{U}^q_{j\in\mathcal{N}_l}(k) = \arg\min_{U_{j\in\mathcal{N}_l}\in\mathcal{U}_l} J\left(x(k), U^{q-1}(k), \Gamma_l\right)$

for $p \in \mathcal{N}$ and $p \neq l$ **do**

Communicate $\tilde{U}^q_{j\in\mathcal{N}_l}(k)$ to agent p

end

Update the solution iterate $\forall j \in \mathcal{N}_l$

$U^q_j(k) \leftarrow \sum_{p\in\mathcal{N}} \alpha_p \tilde{U}_{\substack{i\in\mathcal{N}_p \\ i=j}} + \left(1 - \sum_{p\in\mathcal{N}} \alpha_p\, card\,(\ j \cap \mathcal{N}_p)\right) U^{q-1}_j(k)$

$\rho_l \leftarrow \left\| U^q_{j\in\mathcal{N}_l}(k) - U^{q-1}_{j\in\mathcal{N}_l}(k) \right\|_\infty$

end

Apply $u_{j\in\mathcal{N}_l}(k,k)$

$k \leftarrow k + 1$

Fig. 1 *Jacobi best response-based algorithm*

where at each iteration every agent, given the solution of the others, updates his own solution by solving optimisation (5). The Jacobi implementation is described in Fig. 1.

At each sample k, $q_{\max}$ represents a design limit on the number of iterates q and ε_l represents the stopping criteria of the iterative process. If the algorithm given in Fig. 1 converges, it can be terminated before it achieves the equilibrium without modifying the stability and feasibility of the system. Therefore if communication between agents is a critical issue, few iterations ($q \ll q_{\max}$) can be implemented to compute the solution of (5) at each sample.

Building on variational inequalities framework, we can prove that the algorithm in Fig. 1, as well as its Gauss–Seidel version, globally converge to a unique Nash equilibrium of the game [17, 20, 26]. These conditions involve the strong monotonicity of the Jacobian of the objective function $\nabla_{U_{j\in\mathcal{N}_l}} J(x(k), U(k), \Gamma_l) \quad \forall l \in \mathcal{N}$ and the compactness of the solution sets $\mathcal{U}_l$ and $\tilde{\mathcal{U}}_l$. The global optimality of the solution provided by algorithm in Fig. 1 depends on the structure of the optimisation problem solved by agents and the information exchanged between them. In the framework proposed these issues are related with the value of weights γ_l, the information constraint (5e) and the structure of $U_{j\in\mathcal{N}_l}$, which determines the structure of constraints that can be handled by MPC schemes. If the subproblems share input variables involved in coupled constraints ($\mathcal{N}_l \cap \mathcal{N}_{p\neq l} \neq \emptyset \quad \forall l, p \in \mathcal{N}$), distributed MPC schemes can solve optimisation problems with coupled constraints. On the other hand, when subproblems do not include the shared input variables ($\mathcal{N}_l \cap \mathcal{N}_{p\neq l} = \emptyset \quad \forall l, p \in \mathcal{N}$), the distributed MPC schemes can only solve optimisation problems with independent constraints [18, 19, 26].

Remark 1: The fact that agents individually achieve Nash optimality does not imply the global optimality of the solution. This relationship will depend on the structure of agents' cost function and constraints, which depends on the value of weights γ_l, the information constraint (5e), the structure of $U_{j\in\mathcal{N}_l}$ and the number of iterations allowed.

2.3 Feasibility

In many reported works distributed MPC schemes are initialised using a centralised feasible solution. However, such initial solution can be constructed in a distributed way from the global initial state. An initial feasible solution input $U^0_{j\in\mathcal{N}_l}(k)$ at $k = 0$ can be computed locally by using an inner approximation of the global feasible set $\mathcal{U}$ based on all the constraints appearing in (1) and the global initial state $x(0)$, which is assumed to be available. Consider an inner-hyperbox approximation Ω, which then takes the form of a Cartesian product

$$\Omega = \Omega_1 \times \cdots \times \Omega_m \subset \mathcal{U} \tag{7}$$

This approximation essentially decomposes and decouples the constraints among subsystems by performing constraint tightening. Each subsystem will have to include Ω_l in their local problem setup. Since the Cartesian product of these local constraint sets are included in the globally feasible set $\mathcal{U}$, any combination of local solutions within Ω_l will be globally feasible as well. The local constraint sets that arise from this inner-hyperbox approximation will be in general quite conservative, but at the same time will allow the construction of a feasible solution locally to initialise the algorithm in Fig. 1.

Calculation of inner-hyperbox approximation can be performed a priori and the local Ω_l constraints distributed to each subsystem. The maximum volume inner box can be computed in polynomial-time using the procedure described in [36]. Obtaining the local component-wise constraints Ω_l is then straightforward. For time steps $k > 0$, we construct a feasible solution in the initialisation of each agent of

doi: 10.1049/iet-cta.2010.0634

algorithm in Fig. 1.

$$\tilde{U}^0_{j\in\mathcal{N}_l}(k) = [u_{j\in\mathcal{N}_l}(k, k-1) \ \cdots \ u_{j\in\mathcal{N}_l}(k+M, k-1) \ 0]$$

The feasibility throughout the iterations is maintained because the new control profile $U^q_{j\in\mathcal{N}_l}(k)$ is built as a convex combination of m feasible solutions $\tilde{U}^q_{j\in\mathcal{N}_l}(k)$. Since problem (5) is a convex constrained QP, any convex combination of $\tilde{U}^q_{j\in\mathcal{N}_l}(k)$ also satisfies the convex constraint set. Therefore $U^q(k)$ is a feasible solution of optimisation problem (5).

3 Constrained potential games

In this section we will show that the game resulting from the decomposition of optimisation problem (1) can be recasted as a constrained-potential game. This result will be used in Section 4 to analyse the distributed MPC schemes using the corresponding potential game.

Definition 3: A strategic game $\mathcal{G}$ is a potential game $\mathcal{G}_P$ if there exist a function $\Phi(x(k), U(k))$: $\mathcal{X}\times\mathcal{U}\rightarrow\mathbb{R}$ such that $\forall l\in\mathcal{N}$ $\forall U(k)=[U_{j\in\mathcal{N}_l}(k), U_{j\notin\mathcal{N}_l}(k)]$ and $U^{\#}_l(k) = [U^{\#}_{j\in\mathcal{N}_l}, U_{j\notin\mathcal{N}_l}(k)]\in\mathcal{U}$ verifies

$$\begin{aligned} &J(x(k), U(k), \Gamma_l) - J(x(k), U^{\#}_l(k), \Gamma_l)) \\ &= \Phi(x(k), U(k) - \Phi(x(k), U^{\#}_l(k)) \end{aligned} \tag{8}$$

A potential function Φ is a function of the control action profiles such that the difference induced by a unilateral deviation equals the change in the deviator's cost function, and the existence of a potential cost function for a game implies a strict joint preferences ordering over game outcomes. This, in turn, ensures that the game posses two desirable properties: (i) every potential game poses at least one pure-strategy equilibrium [37], and (ii) every potential game has the 'finite improvement property' [32]. If the cost function $J(x(k),U(k), \Gamma_l)$ is continuously differentiable, then a strategic game $\mathcal{G}$ is a potential game $\mathcal{G}_\mathcal{P}$ if $\forall l, p\in\mathcal{N}$

$$\begin{aligned} &\frac{\partial^2}{\partial U_j \partial U_i} J(x(k), U(k), \Gamma_l) = \frac{\partial^2}{\partial U_j \partial U_i} J(x(k), U(k), \Gamma_p) \\ &\forall j\in\mathcal{N}_l, i\in\mathcal{N}_p \end{aligned} \tag{9}$$

Definition 4: A constrained potential game $\mathcal{G}^C_P$ is called convex-constrained potential game if $\Phi(x(k), U(k))$ and $g(x(k), U(k))$ are the constraints that describe the admissible control set $\mathcal{U}$.

If J is continuously differentiable, then the game $\mathcal{G}^C$ is a potential game $\mathcal{G}^C_P$ if there exists a potential function $\Phi(x(k), U(k))$ that $\forall l\in\mathcal{N}$ verifies

$$\frac{\partial}{\partial U_j} J(x(k), U(k), \Gamma_l) = \frac{\partial}{\partial U_j}\Phi(x(k), U(k)) \quad \forall j\in\mathcal{N}_l \tag{10}$$

Given the definition of potential game it is possible to develop a Lagrangian characterisation of the constrained potential game $\mathcal{G}^C_P$, where the necessary condition for $U^*(k)$ be a Nash equilibrium solution is [38]

$$\frac{\partial}{\partial U_j}\mathcal{L}_\Phi(x(k), U(k), \nu) = 0 \quad \forall j\in\mathcal{N}_l \tag{11}$$

where $\mathcal{L}_\Phi(x(k), U(k), \nu) = \Phi(x(k), U(k)) + \nu^{\mathrm{T}} g(x(k), U(k))$ is the Lagragian of the game. In terms of the constraints and the potential functions condition (11) becomes

$$\begin{aligned} &\frac{\partial}{\partial U_j}\Phi(x(k), \quad U(k)) + \nu^{\mathrm{T}}\frac{\partial}{\partial U_j} g(x(k), U(k)) = 0, \\ &j\in\mathcal{N}_l, l\in\mathcal{N} \end{aligned} \tag{12a}$$

$$\nu^{\mathrm{T}} g(x(k), U(k)) = 0 \tag{12b}$$

Now, given the conditions for the existence of a Nash solution in a constrained potential game $\mathcal{G}^C_P$, we will explore the relation between the solutions of $\mathcal{G}^C_P$ and $\mathcal{G}^C$.

Theorem 1: Suppose there exist a potential function $\Phi(x(k), U(k))$ of the game $\mathcal{G}^C$. The solution of the optimisation problem

$$\min_{U(k)\in\mathcal{U}} \Phi(x(k), U(k)) \tag{13a}$$

s.t.

$$g(x(k), U(k)) \leq 0 \tag{13b}$$

is a Nash equilibrium of the constrained game $\mathcal{G}^C$.

Proof: Let $U^*(k)$ be the optimal solution of (13). Since $U^*(k)$ minimises $\mathcal{L}_\Phi$, it also minimises the component-wise Lagrangian for all $U_l(k)\in\mathcal{U}_l\cap\tilde{\mathcal{U}}_l$

$$\mathcal{L}_{\Phi l}(x(k), U^*(k), \nu^*) \leq \mathcal{L}_{\Phi l}(x(k), U_l(k), \nu^*) \ \forall l\in\mathcal{N} \tag{14}$$

From condition (12b) we have that the potential Φ function of the game verifies

$$\Phi(x(k), U^*(k)) \leq \Phi(x(k), U_l(k)) \tag{15}$$

or equivalently

$$J(x(k), U^*(k), \Gamma_l) \leq \Phi(x(k), U_l(k)) \quad \forall l\in\mathcal{N} \tag{16}$$

Therefore according to the definition, $U^*(k)$ solution of (13) is a Nash equilibrium of $\mathcal{G}^C$. □

It should be noted that under general cases (non-convexity of Φ and g), not every equilibrium from the constrained game solves (13) because (11) is a necessary condition. Therefore it is essential to use other criteria to choose from candidate solutions that minimises the Lagrangian $\mathcal{L}_\Phi$ or Φ.

Corollary 1: Every potential game whose Φ is continuous and $\mathcal{U}$ is non-empty and compact has a Nash equilibrium.

Proof: Using Weierstrass' Theorem, we can conclude that there always exists an optimal solution $U^*(k)$ to (13), and we know from Theorem 1 that $U^*(k)$ is the corresponding Nash equilibrium to the constrained game. □

This result indicates that compactness and continuity of $\mathcal{U}$ are the necessary condition to ensure the existence of the

Nash equilibrium, rather than convexity. If there is no couple constraints, the result of Corollary 1 is similar to Lemma 4.3 in [32].

Corollary 2: Assume that Φ is continuous and differentiable on $\mathcal{U}$ and Φ and g are convex, then solving (13) is equivalent to solve the constrained game $\mathcal{G}^C$.

Proof: The necessary condition for a Nash equilibrium is given by

$$\frac{\partial \mathcal{L}_\Phi}{\partial U_j}(x(k), U(k), \nu) = 0 \quad \forall j \in \mathcal{N}_l, l \in \mathcal{N} \tag{17}$$

Since Φ and g are convex, (17) becomes a sufficient condition for Nash equilibrium. Therefore the problem of solving a constrained game $\mathcal{G}^C$ is equivalent to solve (13). □

Finally, we can use the results introduced in this section to prove the uniqueness of the Nash equilibrium. Since Φ is convex, the problem (17) has an unique solution. According to Theorem 1, this solution is a Nash equilibrium of the potential game. Owing to the equivalence with optimisation problem (17) from Corollary 1, we can conclude that the solution is also unique for the game.

4 Distributed MPC schemes

There are a growing number of distributed MPC schemes reported in the literature. However they can be classified in three groups, which can be obtained using three different sets of values of $\gamma_l \; \forall l \in \mathcal{N}$ and information exchanged by players [constraint (5e)]. Each of these groups can be characterised through an equivalent potential game, which will be analysed applying classical Lagrangian multiplier theory and dissipative theory to derive their properties (stability, optimality, feasibility, convergence).

4.1 Distributed MPC

In the first case, we will consider the case when the objective function and constraints of the system is available to all agents ($\gamma_l > 0 \; \forall l \in \mathcal{N}$) and all information is shared between agents. Under this conditions the objective function and constraints considered by each agent are

$$J(x(k), U(k), \Gamma_l) = \sum_{l \in \mathcal{N}} \gamma_l J_l(x(k), U(k)) \quad \forall l \in \mathcal{N} \tag{18a}$$

$$U_{j \notin \mathcal{N}_l}(k) = \mathbf{U}_{j \notin \mathcal{N}_l}(k) \tag{18b}$$

$$U_{j \in \mathcal{N}_l}(k) \in \tilde{\mathcal{U}}_{j \in \mathcal{N}_l}(U_{j \notin \mathcal{N}_l}(k)) \tag{18c}$$

Then, the game resulting from decomposition of the centralised MPC has the potential function

$$\Phi(x(k), U(k)) = \sum_{l \in \mathcal{N}} \gamma_l J_l(x(k), U(k)) \tag{19}$$

which requires each agent to observe the decisions of all agents ($U_{j \notin \mathcal{N}_l}(k)$) to evaluate the effect of $U_{j \in \mathcal{N}_l}(k)$, taking in account the effect of interactions between the subsystems. This structure of the optimisation problem (5) leads to a 'complete and perfect information game'. In this case the centralised optimisation problem (1) has been distributed between m independent agents that coordinate their solutions in order to solve it in a distributed way [26, 39], for this reason this control scheme is called 'distributed MPC'.

The agents' objective function and constraints (18) lead to a constrained potential game with a potential function $\Phi(x(k), U(k))$ that is exactly similar to the centralised optimisation problem (1). Therefore it is straightforward to show, using Lagrangian and Lyapunov arguments, that the resulting constrained potential game has the same properties like the centralised MPC. The only difference is how the problems are solved. This means that the existence and uniqueness of the equilibrium point, as well as the convergence of the iterative process, are guaranteed for any system, independently how agents are defined. The location of the of the equilibrium point is exactly the same of the solution of the centralised MPC. Therefore since closed-loop properties (stability, feasibility and optimality) are related with the existence and location of the equilibrium point, the distributed MPC guaranteed the same closed-loop properties like the the centralised MPC, independently of how the system is decomposed.

4.2 Coordinated decentralised MPC

In the second case, let us consider the case when information of each subsystem is only available ($\gamma_l = 1, \gamma_{p \neq l} = 0 \quad l, p \in \mathcal{N}$) and information is shared between agents. Under this condition the objective function and constraints considered by each agent are

$$J(x(k), U(k), \Gamma_l) = J_l(x(k), U(k)) \quad \forall l \in \mathcal{N} \tag{20a}$$

$$U_{j \notin \mathcal{N}_l}(k) = \mathbf{U}_{j \notin \mathcal{N}_l}(k) \tag{20b}$$

$$U_{j \in \mathcal{N}_l}(k) \in \tilde{\mathcal{U}}_{j \in \mathcal{N}_l}(U_{j \notin \mathcal{N}_l}(k)) \tag{20c}$$

Then, the game resulting from decomposition of the centralised MPC has the potential function

$$\Phi(x(k), U(k)) = J(x(k), U(k)) - \sum_{p \neq l} \gamma_p J_p(x(k), U(k)) \tag{21}$$

which requires each agent to observe the decisions of all agents ($U_{j \notin \mathcal{N}_l}(k)$) to evaluate the effect of $U_{j \in \mathcal{N}_l}(k)$, only in subsystem l without taking into account its effects in the remaining agents. This configuration of the distributed problem (5) leads to an 'incomplete and perfect information game' that can achieve Nash optimal solutions for a pure strategy [40]. Therefore each agent minimises its performance index, accommodating the effects of other agents' decisions, without taking in account the effects of its decision on the rest of the system. In this case the centralised optimisation problem (1) has been decentralised between m independent agents that only coordinate the effects of their decisions to minimise the effect of interactions [19, 21, 22, 41, 42]. For this reason the control scheme is called 'coordinated decentralised MPC'.

A coordinated decentralised MPC, with objective functions (20), constitute a constrained potential game with a potential function given by (21), which is different from the original centralised-optimisation problem (1). Using Lyapunov arguments [43, 44] we can prove that the existence of the equilibrium and convergence of the iterative process depend, in this scheme, on how the system is decomposed [45]. The uniqueness of the equilibrium point is guaranteed by the convexity of the cost function and constraints. However, it is located outside of the Paretto set of (13), therefore the solutions are not globally optimal [45]. Therefore since

doi: 10.1049/iet-cta.2010.0634

closed-loop properties (stability, feasibility and optimality) are related to the existence of the equilibrium point, closed-loop properties depend on how the system is decomposed.

4.3 Decentralised MPC

Finally, we consider the case when information of each subsystem is only available ($\gamma_l = 1$, $\gamma_{p\neq l} = 0 \quad l, p \in \mathcal{N}$) and information is not shared between players. Under this condition the objective function and coupled constraints considered by each agent are

$$J(x(k), U(k), \Gamma_l) = J_l(x(k), [U_{j\in\mathcal{N}_l}(k), \mathbf{0}]) \quad \forall l \in \mathcal{N} \tag{22a}$$

$$U_{j\notin\mathcal{N}_l}(k) = \mathbf{0} \tag{22b}$$

This condition evaluates the effect of $U_{j\in\mathcal{N}_l}$ in subsystem l without taking into account its effects on the remaining agents. This configuration of the distributed problem leads to an incomplete and imperfect information game. Therefore each agent minimises its performance index, without taking into account the effects of other agents' decisions. In this case the centralised optimisation problem is decentralised into m independent agents that do not coordinate the effects of their decisions, for this reason this control scheme is called 'decentralised MPC'. The existence of the equilibrium point and convergence of the iterative process depend, in this scheme, on how the system is decomposed [11]. Uniqueness of the equilibrium point is guaranteed by the convexity of the cost function and constraints. Therefore since closed-loop properties (stability, feasibility and optimality) are related with the existence of the equilibrium point, closed-loop properties depends on how the system is decomposed.

The agents objective functions (22) constitute a constrained potential game with the potential function $\Phi(x(k), U(k))$ given by

$$\Phi(x(k), U(k)) = J(x(k), [U_{j\in\mathcal{N}_l}(k), \mathbf{0}]) - \sum_{p\neq l} \gamma_p J_p(x(k), [U_{j\in\mathcal{N}_l}(k), \mathbf{0}])$$

Using Lyapunov arguments [43, 44] we can prove that the existence of the equilibrium and convergence of the iterative process depend, in this scheme, on how the system is decomposed [10, 11]. The uniqueness of the equilibrium point is guaranteed by the convexity of the cost function and constraints. However, it is located outside of the Paretto set of (1), therefore the solutions are not globally optimal. Therefore since closed-loop properties (stability, feasibility and optimality) are related with the existence of the equilibrium point, closed-loop properties depends on how the system is decomposed.

5 Simulation and results

In this section, we will illustrate the applicability and limitations of the theoretical results presented in this paper through the operation of a (HEN. The HEN system studied here is represented schematically in Fig. 2. It is a system with only three recovery exchangers (I_1, I_2 and I_3) and three service (S_1, S_2 and S_3) units. Two hot process streams (h_1 and h_2) and two cold process streams (c_1 and c_2) take part in the heat exchange process. There are also three utility streams (s_1, s_2 and s_3) that can be used to help in reaching the desired outlet temperatures.

The main purpose of a HEN is to recover as much energy as necessary to achieve the system requirements from high-

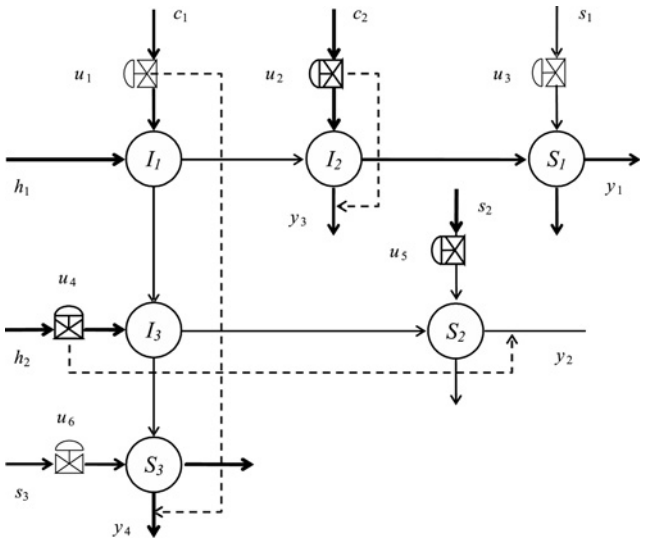

Fig. 2 *Schematic representation of the HEN system*

temperature process streams (h_1 and h_2) and to transfer this energy to cold-process streams (c_1 and c_2). The benefits are savings in fuels needed to produce utility streams s_1, s_2 and s_3. However, the HEN has to also provide the proper thermal conditioning of some of the process streams involved in the heat transfer network. This means that a control system must (i) drive the exit process-stream temperatures (y_1, y_2, y_3 and y_4) to the desired values in presence of external disturbances and input constraints while (ii) minimise the amount of utility energy.

The manipulated variables of a HEN are the flow rates at bypasses around heat exchangers (u_1, u_2 and u_4) and the flow rates of utility streams in service units (u_3, u_5 and u_6), which are constrained

$$0 \leq u_j(k) \leq 1.0, \quad j = 1, \ldots, 6$$

A fraction $0 < u_j < 1$ of bypass j means a fraction u_j of corresponding stream goes through the bypass and a fraction $1 - u_j$ goes through the exchangers, exchanging energy with other streams. If $u_j = 0$ the bypass is 'completely closed' and the whole stream goes through the exchangers, maximising the energy recovery. On the other hand, a value of $u_j = 1$ the bypass is 'completely open' and the whole stream goes through the bypass, minimising the energy recovery.

The HEN studied in this work has more control inputs than outlet temperatures to be controlled, therefore the set of input values satisfying the output targets is not unique. The possible operation points may result in different levels of heat integration and utilities consumption. Under nominal conditions only one utility stream is required (s_1 or s_3) for the operation of the HEN, the others are used to expand the operational region. The inclusion of the control system provides new ways to use the extra utility services (s_2 and s_3) to achieve control objectives by introducing new interactions that allow the redirection of the energy by manipulating the flow rates. For example, any change in the utility stream s_3 (u_6) has a direct effect on output temperature of c_1 (y_4), however the control system will redirect this change (through the modification of u_1) to the output temperature of $h_1(y_1)$, $h_2(y_2)$ and $c_2(y_3)$. In this way, the HEN has energy recycles that induces feedback interaction, whose strength depends on the operational conditions, and leads to a complex dynamic: (i) small-energy recycles induce weak couplings among subsystems, whereas (ii) large-energy recycles induce a time-scale separation, with the dynamics of individual

subsystems evolving in a fast time scale with weak interactions, and the dynamics of the overall system evolving in a slow time scale with strong interactions [8].

A complete definition of this problem can be found in [12]. The controllers were developed using the following linear model

$$Y = G(s) * U \tag{23}$$

where (see equation at the bottom of the page)
and

$$U = [\,u_1 \quad u_2 \quad u_3 \quad u_4 \quad u_5 \quad u_6\,]^{\mathrm{T}}$$
$$Y = [\,y_1 \quad y_2 \quad y_3 \quad y_4\,]$$

The first issue that we need to address in the development of the distributed controllers is selection of the input and output variables associated with each agent. The decomposition was carried after consideration of the multi-loop rules [46]. The resulting decomposition is given in Table 1: Agent 1 corresponds to the first and third rows of (23), whereas agents 2 and 3 correspond to the second and fourth rows of (23) respectively. Agents 1 and 2 will mainly interact between them through the process stream c_1.

For a HEN not only the dynamic performance of the control system is important but also the cost associated with the resulting operating condition must be taken into account. Thus, the performance index (5) is augmented by including an economic term

$$J_U = u_{SS}^{\mathrm{T}} R_U u_{SS} \tag{24}$$

Table 1 Agent definition

	Agent 1	Agent 2	Agent 3
inputs	u_1, u_2, u_3	u_1, u_4, u_5	u_1, u_4, u_6
outputs	y_1, y_3	y_2	y_4

such that the global cost is given by $J + J_U$. In the case of the centralised MPC u_{SS} is given by $u_{SS} = [u_3(k+M, k)u_5(k+M, k)u_6(k+M, k)]$, whereas for the distributed and coordinated decentralised *MPC* u_{SS} is decomposed among the agents [$u_{SS} = u_3(k+M, k)$ for Agent 1, $u_{SS} = u_5(k+M, k)$ for Agent 2 and $u_{SS} = u_6(k+M, k)$ for Agent 3]. Finally, the tuning parameters of the MPC controllers are: $t_s = 0.2$ min; $V_l = 50$; $M_l = 5$; $\varepsilon_l = 0.01$; $q_{\max} = 10$ $l = 1$, 2, 3, the cost function matrices are given in Table 2.

MATLAB-based simulation results are carried out to evaluate the proposed MPC algorithms (coordinated decentralised and distributed MPC) through performance comparison with a centralised and decentralised MPC. The MPC algorithms used the same routines during the simulations that were run in a computer with an Intel Quad-core Q9300 CPU on Linux operating system. One of the processors was used to execute the HEN simulator, whereas the others were used to execute the MPC controllers. Only one processor was used to run the centralised MPC controller. In the case of the distributed algorithms, the controllers were distributed among the other processors. These configurations were adopted in order to make a fair comparison of the computational time employed for each controller.

We consider the responses obtained for disturbance rejection. A sequence of changes is introduced into the system: after stabilising at nominal conditions, the inlet temperature of h_1 ($T^{\mathrm{in}}_{h_1}$) changes from 90 to 80°C; 10 min after the inlet temperature of h_2 ($T^{\mathrm{in}}_{h_2}$) goes from 130 to 140°C and after another 10 min the inlet temperature of c_1 ($T^{\mathrm{in}}_{c_1}$) changes from 30 to 40°C.

Figs. 3–5 show the dynamic responses of the HEN operating with a distributed MPC and a coordinated decentralised MPC. The worse performance is observed during the first and second load changes, most notably on y_1 and y_3. The reasons for this behaviour can be found by observing the manipulated variables. The first fact to be noted is that under nominal steady-state conditions, u_4 is

Table 2 Cost functions matrices

	Agent 1	Agent 2	Agent 3
Coord. Dec.	$Q_i = \mathrm{diag}(1, 1)$ $R_i = \mathrm{diag}(5, 5, 20)$ $R_U = [10]$	$Q_i = [1]$ $R_i = \mathrm{diag}(5, 5, 20)$ $R_U = [40]$	$Q_i = [1]$ $R_i = \mathrm{diag}(5, 5, 20)$ $R_U = [40]$
distributed	$Q_i = \mathrm{diag}(1, 1, 1, 1)$ $R_i = \mathrm{diag}(5, 5, 20)$ $R_U = [10]$	$Q_i = \mathrm{diag}(1, 1, 1, 1)$ $R_i = \mathrm{diag}(5, 5, 20)$ $R_U = [40]$	$Q_i = \mathrm{diag}(1, 1, 1, 1)$ $R_i = \mathrm{diag}(5, 5, 20)$ $R_U = [40]$
centralised		$Q_i = \mathrm{diag}(1, 1, 1, 1)$ $R_i = \mathrm{diag}(5, 5, 20, 5, 20, 20)$ $R_U = \mathrm{diag}(10, 40, 40)$	

$$G(s) = \begin{bmatrix} \dfrac{20.6\,e^{-61.3s}}{38.8s+1} & \dfrac{19.9\,e^{-28.9s}}{25.4s+1} & \dfrac{17.3\,e^{-4.8s}}{23.8s+1} & 0 & 0 & 0 \\ \dfrac{4.6\,e^{-50.4s}}{48.4s+1} & 0 & 0 & 79.1\dfrac{31.4s+0.8}{31.4s+1.0} & \dfrac{20.1\,e^{-4.1s}}{25.6s+1.0} & 0 \\ \dfrac{16.9\,e^{-24.7s}}{39.5s+1} & -39.2\dfrac{22.8s+0.8}{22.8s+1.0} & 0 & 0 & 0 & 0 \\ 24.4\dfrac{48.2s^2+4.0s+0.05}{48.2s^2+3.9s+0.06} & 0 & 0 & -8.4\dfrac{e^{-18.8s}}{27.9s+1} & 0 & \dfrac{16.3\,e^{-3.5s}}{20.1s+1.0} \end{bmatrix}$$

doi: 10.1049/iet-cta.2010.0634

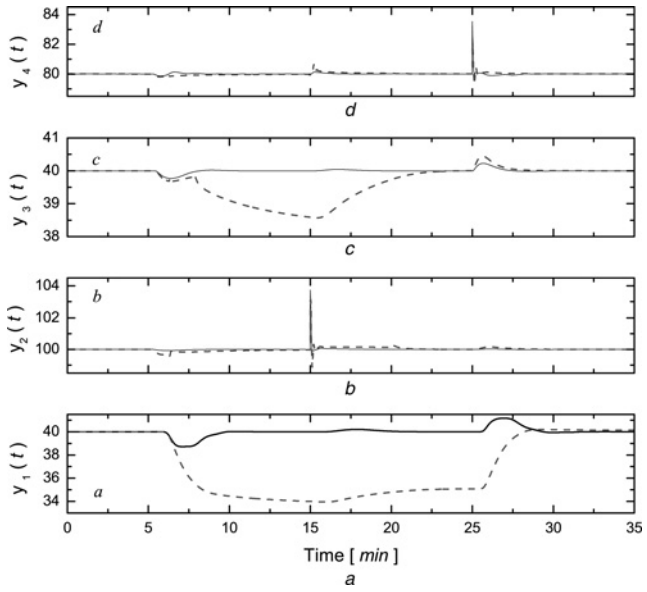

Fig. 3 *Responses of HEN outputs a T_{h1}, b T_{h2}, c T_{c2}, d T_{c1} using (—) distributed MPC and (-.-) coordinated decentralised MPC*

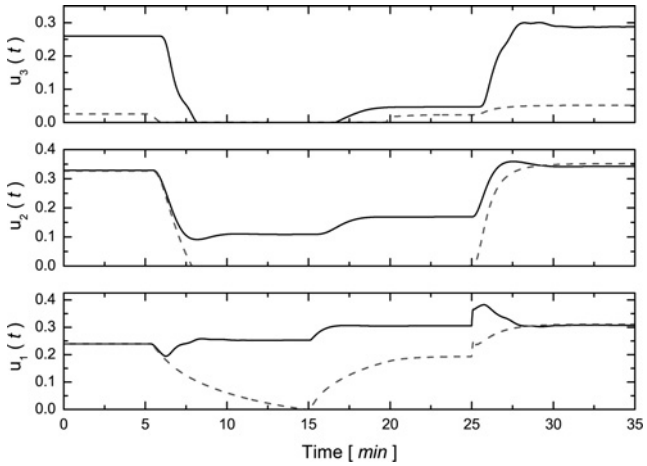

Fig. 4 *Responses of bypasses around exchangers $I_1(u_1)$ and $I_2(u_2)$ and service $S_1(u_3)$ using (—) distributed MPC and (-.-) coordinated decentralised MPC*

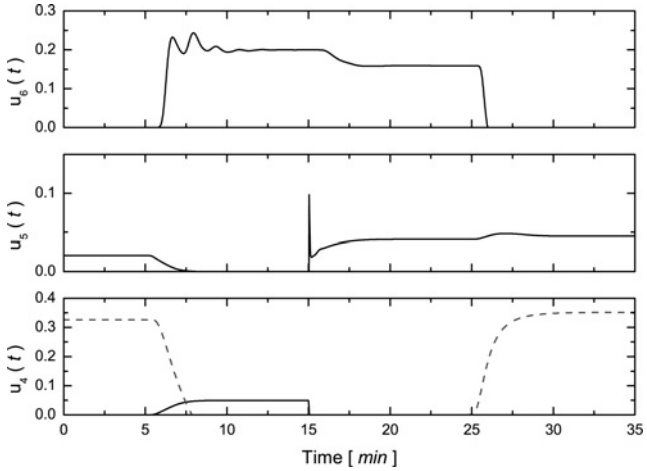

Fig. 5 *Responses of bypasses around exchangers $I_3(u_4)$ and services $S_2(u_5)$ and $S_3(u_6)$ using (—) distributed MPC and (-.-) coordinated decentralised MPC*

completely closed and y_2 is controlled by u_5 (see Fig. 5), achieving the maximum energy recovery. Observe also that u_6 is inactive since no heating service is necessary at this point. After the first load change occurs, both control variables u_2 and u_3 fall rapidly (see Fig. 4). Under this conditions, the system activates the heater flow rate u_6 (see Fig. 5). The dynamic reaction of the heater to the cold disturbance is also stimulated by u_2, while u_6 takes complete control of y_1, achieving the maximum energy recovery. After the initial effect is compensated, y_3 is controlled through u_2 – which never saturates–, while u_6 takes complete control of y_1. Furthermore, Fig. 5 shows that the cold perturbation also affects y_2, where u_5 is effectively taken out of operation by u_4. The ensuing pair of load changes are heat perturbations featuring manipulated movements in the opposite sense to those indicated above.

In these figures we can also see that the coordinated decentralised MPC fails to reject the first and second disturbances on y_1 and y_3 (see Figs. 3*a* and *c*) because it is not able to properly coordinate the use of utility service u_6 to compensate the effects of active constraints on u_2 and u_3. This happens because the coordinated decentralised MPC is only able to address the effect of interactions between agents but it cannot coordinate the use of utility streams s_2 and s_3 to avoid the output-unreachability under input constraint problem. The origin of the problem lies in the cost function employed by the coordinated decentralised MPC which does not include the effect of the local decision variables on the other agents. This fact leads to different steady-state values in the manipulated variables to those obtained by the distributed MPC along the simulation.

Fig. 6 shows the steady-state value of the recovered energy and utility services used by the system for the distributed MPC schemes. As mentioned earlier, the centralised and distributed MPC algorithms have similar steady-state conditions. These solutions are Pareto optimal, hence they achieve the best plant-wide performance for the combined performance index. On the other hand, the coordinated decentralised MPC exhibited a good performance in energy terms, since it employs less service energy, however, it is not able to achieve the control objectives, because it is not able to properly coordinate the use of utility flows u_5 and u_6. As it was pointed out in previous sections, the fact that

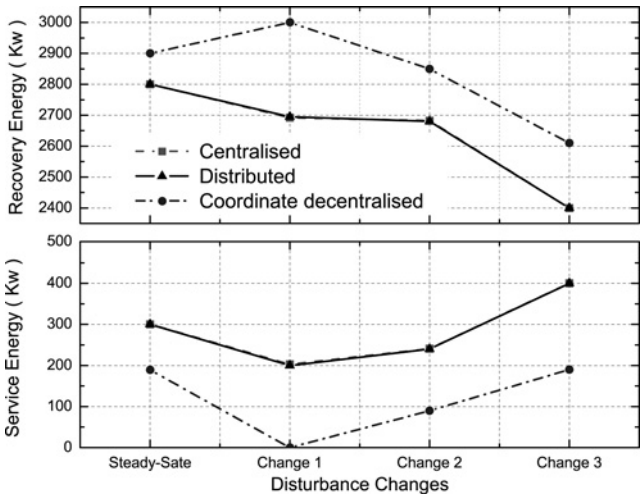

Fig. 6 *Steady-state conditions achieved by the HEN system for different MPC schemes (- -, centralised MPC; —, distributed MPC; and -.-, coordinated decentralised MPC)*

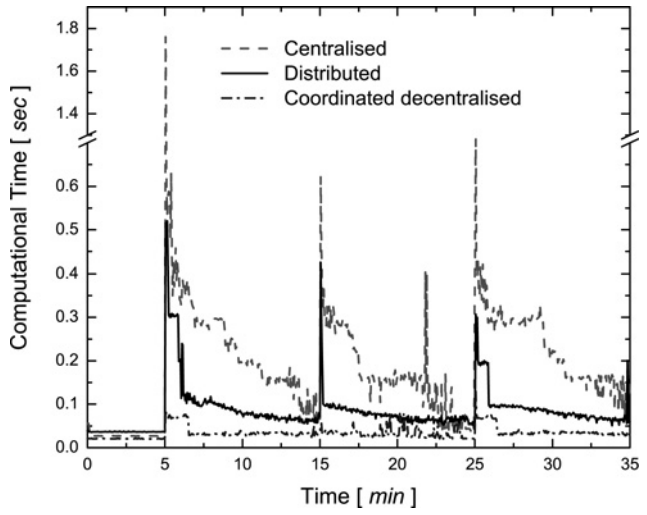

Fig. 7 *CPU times for different MPC schemes (- -, centralised MPC; —, distributed MPC; and -.-, coordinated decentralised MPC)*

the agents achieve the Nash equilibrium does not imply the optimality of the solution.

Fig. 7 shows the CPU time employed for each MPC algorithm during the simulations. As expected, the centralised MPC is the algorithm that uses the CPU more intensively. Its CPU time is always larger than the others along the simulation. This fact is originated on the size of the optimisation problem and the dynamics of the system that forces the centralised MPC to permanently correct the manipulated variables along the simulation because of the system interactions. On the other hand, the coordinated decentralised MPC uses the CPU less intensively than the others algorithms, because of the size of the optimisation problem. However, its CPU time remains almost constant during the entire simulation since it needs to compensate the interactions that had not been taken into account during the computation. In general, all algorithms show larger CPU times after the load changes because of the recalculation of the control law. However, we have to point out that the values of these peaks are smaller than sampling time.

6 Conclusions

In this paper a distributed model of predictive control framework based on dynamic games is presented. The MPC is implemented in distributed way with the inexpensive agents within the network environment. These agents can cooperate and communicate with each other to achieve the objective of the whole system. Coupling effects among the agents are taken into account in this scheme, which is superior to other traditional decentralised control methods. The main advantage of this scheme is that the on-line optimisation 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, sample time etc., can all be designed and tuned separately that provides more flexibility for analysis and applications. The second part of this study is to investigate the convergence, stability, feasibility and performance of the distributed control scheme. These will provide users a better understanding of the developed algorithm and sensible guidance in applications.

7 Acknowledgments

The author wishes to thank the Agencia Nacional de Promoción Científica y Tecnológica, the Universidad Nacional de Litoral (with PAE 37122, PAE-PICT-2007–00052) and the Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) from Argentina, for their support.

8 References

1 Bemporad, A., Morari, M.: 'Robust model predictive control: a survey, in robustness in identification and control', *Lect. Notes Control Inf. Sci.*, 1999, **245**, pp. 207–226
2 Mayne, D., Rawlings, J., Rao, C., Scokaert, P.: 'Constrained model predictive control: stability and optimality', *Automatica*, 2000, **36**, pp. 789–814
3 Kouvaritakis, B., Cannon, M.: 'Nonlinear predictive control: theory and practice' (IET, 2001)
4 Maciejowski, J.: 'Predictive control: with constraints' (Prentice Hall, 2002)
5 Camacho, E., Bordons, C.: 'Model predictive control' (Springer, 2004)
6 Braun, M., Rivera, D., Flores, M., Carlyle, W., Kempf, K.: 'A model predictive control framework for robust management of multi-product, multi-echelon demand networks', *Annu. Rev. Control*, 2003, **27**, (2), pp. 229–245
7 Perea-Lopez, E., Ydstie, B., Grossmann, I.: 'A model predictive control strategy for supply chain optimization', *Comput. Chem. Eng.*, 2003, **27**, (8–9), pp. 1201–1218
8 Kumar, A., Daoutidis, P.: 'Nonlinear dynamics and control of process systems with recycle', *J. Process Control*, 2002, **12**, (4), pp. 475–484
9 Lu, J.: 'Challenging control problems and emerging technologies in enterprise optimization', *Control Eng. Pract.*, 2003, **11**, (8), pp. 847–858
10 Sandell, Jr N., Varaiya, P., Athans, M., Safonov, M.: 'Survey of decentralized control methods for large scale systems', *IEEE Trans. Autom. Control*, 1978, **23**, (2), pp. 108–128
11 Šiljak, D.: 'Decentralized control and computations: status and prospects', *Annu. Rev. Control*, 1996, **20**, pp. 131–141
12 Aguilera, N., Marchetti, J.: 'Optimizing and controlling the operation of heat exchanger networks', *AIChe J.*, 1998, **44**, (5), pp. 1090–1104
13 Zhu, G., Henson, M.: 'Model predictive control of interconnected linear and nonlinear processes', *Ind. Eng. Chem. Res.*, 2002, **41**, (4), pp. 801–816
14 Cheng, R., Forbes, J., Yip, W.: 'Price-driven coordination method for solving plant-wide MPC problems', *J. Process Control*, 2007, **17**, (5), pp. 429–438
15 Cheng, R., Fraser Forbes, J., Yip, W.: 'Dantzig–Wolfe decomposition and plant-wide MPC coordination', *Comput. Chem. Eng.*, 2008, **32**, (7), pp. 1507–1522
16 Aske, E., Strand, S., Skogestad, S.: 'Coordinator MPC for maximizing plant throughput', *Comput. Chem. Eng.*, 2008, **32**, (1–2), pp. 195–204
17 Dunbar, W., Murray, R.: 'Distributed receding horizon control for multi-vehicle formation stabilization', *Automatica*, 2006, **42**, (4), pp. 549–558
18 Dunbar, W.: 'Distributed receding horizon control of dynamically coupled nonlinear systems', *IEEE Trans. Autom. Control*, 2007, **52**, (7), pp. 1249–1263
19 Jia, D., Krogh, B.: 'Distributed model predictive control'. Proc. American Control Conf., 2001, vol. 4
20 Camponogara, E., Jia, D., Krogh, B., Talukdar, S.: 'Distributed model predictive control', *IEEE Control Syst. Mag.*, 2002, **22**, (1), pp. 44–52
21 Zhang, Y., Li, S.: 'Networked model predictive control based on neighbourhood optimization for serially connected large-scale processes', *J. Process Control*, 2007, **17**, (1), pp. 37–50
22 Vaccarini, M., Longhi, S., Katebi, M.: 'Unconstrained networked decentralized model predictive control', *J. Process Control*, 2009, **19**, (2), pp. 328–339
23 Jia, D., Krogh, B.: 'Min–max feedback model predictive control for distributed control with communication'. Proc. American Control Conf., 2002, vol. 6
24 Negenborn, R., De Schutter, B., Hellendoorn, J.: 'Multi-agent model predictive control for transportation networks: serial versus parallel schemes', *Eng. Appl. Artif. Intell.*, 2008, **21**, (3), pp. 353–366

doi: 10.1049/iet-cta.2010.0634

25 Wakasa, Y., Arakawa, M., Tanaka, K., Akashi, T.: 'Decentralized model predictive control via dual decomposition'. 47th IEEE Conf. on Decision and Control, 2008, CDC 2008, 2008, pp. 381–386
26 Venkat, A., Hiskens, I., Rawlings, J., Wright, S.: 'Distributed MPC strategies with application to power system automatic generation control', *IEEE Trans. Control Syst. Technol.*, 2008, **16**, (6), pp. 1192–1206
27 Camponogara, E., Oliveira, L.D.: 'Distributed optimization for model predictive control of linear dynamic networks', *IEEE Trans. Syst., Man Cybern., A Syst. Hum.*, 2009, **39**, (6), pp. 1331–1338
28 Doan, D., Keviczky, T., Necoara, I., Diehl, M., Schutter, B.D.: 'A distributed version of han's method for dMPC using local communications only', *J. Control Eng. Appl. Inform.*, 2009, **11**, (3), pp. 6–15
29 Venkat, A.: 'Distributed model predictive control: theory and applications'. PhD dissertation, University of Wisconsin, Madison, 2006
30 Bertsekas, D., Tsitsiklis, J.: 'Parallel and distributed computation' (Prentice-Hall, Inc, NJ, 1999)
31 Kim, B., Baldick, R.: 'Coarse-grained distributed optimal power flow', *IEEE Trans. Power Syst.*, 1997, **12**, (2), pp. 932–939
32 Monderer, D., Shapley, L.: 'Potential games', *Games Econ. Behav.*, 1996, **14**, pp. 124–143
33 Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.: 'The explicit linear quadratic regulator for constrained systems', *Automatica*, 2002, **38**, (1), pp. 3–20
34 Negenborn, R., Schutter, B.D., Hellendoorn, J.: 'Multi-agent model predictive control for transportation networks: serial versus parallel schemes', *Eng. Appl. Artif. Intell.*, 2008, **21**, (3), pp. 353–366
35 Nash, J.: 'Non-cooperative games', *Ann. Math.*, 1951, **54**, (2), pp. 286–295
36 Bemporad, A., Filippi, C., Torrisi, F.: 'Inner and outer approximations of polytopes using boxes', *Comput. Geom.: Theory Appl.*, 2004, **27**, (2), pp. 151–178
37 Rosenthal, R.: 'A class of games possessing pure-strategy Nash equilibria', *Int. J. Game Theory*, 1973, **2**, (1), pp. 65–67
38 Pavel, L.: 'An extension of duality to a game-theoretic framework', *Automatica*, 2007, **43**, (2), pp. 226–237
39 Rawlings, J., Stewart, B.: 'Coordinating multiple optimization-based controllers: new opportunities and challenges', *J. Process Control*, 2008, **18**, (9), pp. 839–845
40 Osborne, M., Rubinstein, A.: 'A course in game theory' (MIT Press, 1994)
41 Venkat, A., Rawlings, J., Wright, S.L.: 'Plant-wide optimal control with decentralized MPC'. Proc. IFAC Dynamics and Control of Process Systems Conf., Boston, MA, 2004
42 Venkat, A., Rawlings, J., Wright, S.L.: 'Stability and optimality of distributed model predictive control'. 44th IEEE Conf. on Decision and Control, 2005 and 2005 European Control Conf., CDC-ECC'05, 2005, pp. 6680–6685
43 Scattolini, R.: 'Architectures for distributed and hierarchical model predictive control – a review', *J. Process Control*, 2009, **19**, (5), pp. 723–731
44 Alessio, A., Bemporad, A.: 'Stability conditions for decentralized model predictive control under packet drop communication'. American Control Conf., 2008, pp. 3577–3582
45 Balderud, J., Giovanini, L., Katebi, R.: 'Distributed control of underwater vehicles', *Proc. Inst. Mech. Eng., M J. Eng. Maritime Environ.*, 2008, **222**, (2), pp. 95–107
46 Hovd, M., Skogestad, S.: 'Sequential design of decentralized controllers', *Automatica*, 1994, **30**, pp. 1601–1601