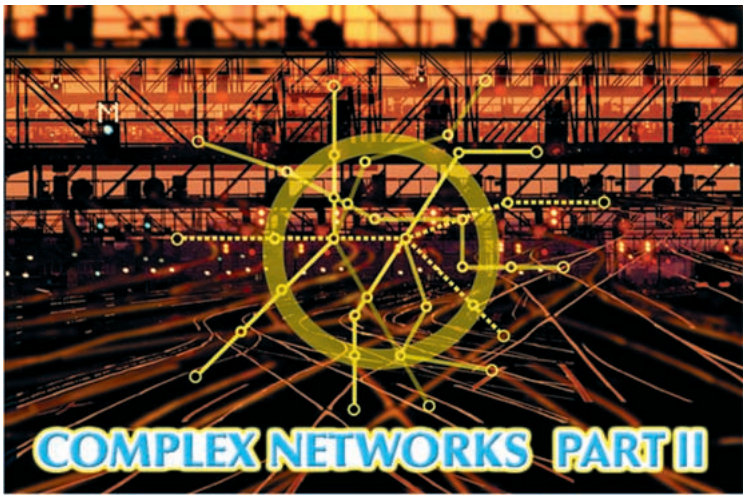

Distributed Model Predictive Control

By Eduardo Camponogara, Dong Jia, Bruce H. Krogh, and Sarosh Talukdar

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 given 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 is well developed; nearly all aspects, such as stability, nonlinearity, and robustness, have been discussed in the literature (see, e.g., [1]-[4]). MPC is very popular in the process control industry because the actual control objectives and operating constraints can be represented explicitly in the optimization problem that is solved at each control instant. Many successful MPC applications have been reported in the last two decades [2], [4].

Typically, MPC is implemented in a centralized fashion. The complete system is modeled, and all the control inputs are computed in one optimization problem. In large-scale applications, such as power systems, water distribution systems, traffic systems, manufacturing systems, and economic systems, it is useful (sometimes necessary) to have distributed or decentralized control schemes, where local control inputs are computed using local measurements and reduced-order models of the local dynamics [5], [6]. The goal of the research described here is to realize the attractive features of MPC (meaningful objective functions and constraints) in a decentralized implementation.

Previous work on distributed MPC is reported in [7]-[14]. In some applications, multiple low-level controllers are simply implemented using MPC, just as one might use proportional-integral-derivative (PID) controllers to close local feedback loops [13]. For water distribution systems, full-scale centralized MPC computations have been decomposed for decentralized computation, using standard coordi-

Krogh (krogh@ece.cmu.edu), Camponagara, Jia, and Talukdar are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A.

0272-1708/02/$17.00©2002IEEE

nation techniques (e.g., augmented Lagrangian) or estimation and prediction schemes to obtain solutions at each control instant [10]-[12], [14]. Decentralized team strategies for linear-quadratic problems were considered in [8] and [9]. This method is suitable for linear time-invariant systems. If the optimal control problem is a linear quadratic Gaussian (LQG) problem, an analytical solution can be found. Some researchers [9] suggest using a neural network to approximate stationary controllers for nonlinear systems, but this approach is not suitable for large systems.

In this article, we consider situations where the distributed controllers, or agents, can exchange information. 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, as in network control applications where the number of local state and control variables for each agent and the number of variables shared with other agents are a small fraction of the total number of variables in the system. 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 agents cannot divulge all the information about their local models and objectives. This is the case, for example, in the newly deregulated power markets in the United States.

Our objective is to achieve some degree of coordination among agents that are solving MPC problems with locally relevant variables, costs, and constraints.

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 interagent communication network. Here we assume that the connectivity of the communication network is sufficient for the agents to obtain information regarding all the variables that appear in their local problems. Regarding the network capacity, we consider two different situations. First, we consider situations where it is possible for the agents to exchange information several times while they are solving their local optimization problems at each control instant. 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.

We then consider a situation where the capacity of the communication network does not allow the agents to exchange information while they are solving their local optimization problems. In particular, we consider the case when the agents can exchange information only once after each local MPC optimization problem is solved, while the current, local control actions are being applied to the system. Consequently, there is a one-step delay in the information available from the other agents when an agent formulates and solves its local MPC optimization problem at each step. For this new situation, we focus on the fundamental issue of stability since sufficient conditions from the literature for centralized MPC do not apply. For a class of linear time-invariant systems, we develop an extension of the stability-constraint method in [17] and [21] to the distributed MPC problem with one-step communication delays. We present sufficient conditions for guaranteeing stability of the closed-loop system and illustrate the approach by an example of load-frequency control in a two-area power system.

Model Predictive Control

In MPC, control decisions $u(k)$ are made at discrete time instants $k = 0,1,2,...$, which usually represent equally spaced time intervals. 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:

$$\min_{X(k),U(k)} J(X(k),U(k))$$

where

$$X(k) = \{x(k+1|k),...,x(k+N|k)\}$$

$$U(k) = \{u(k|k),...,u(k+N-1|k)\}$$

s.t.

$$x(k+i+1|k) = F(x(k+i|k),u(k+i|k)) \qquad (i = 0,...,N-1)$$

$$G(X(k),U(k)) \le 0$$

$$x(k|k) = x(k).$$

In the preceding formulation, the performance index represents the measure of the difference between the predicted behavior and the desired future behavior: the lower the value, the better the performance. 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. Predictions are based on the system model $x(k+1)=F(x(k),u(k))$. The constraints 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; that is, $u(k)=u(k|k)$. Then, the controller waits until the next control instant and repeats this process to find the next control action.

Distributed Model Predictive Control Problem

The standard MPC formulation in the previous section can be summarized as a series of static optimization problems, $\{SP_k|k=0,1,2,...\}$, each of the form:

$$SP_k: \min_S J(S)$$
$$\text{s.t.} \quad G(S) \le 0$$
$$H(S)=0,$$

where S is the vector of the decision variables, including state variables X and control variables U, over the prediction horizon. The equality constraint in the problem includes the prediction model and other equality operation constraints.

Distributed MPC is a decomposition of SP_k into a set of M subproblems, $\{SP_{ki}|i=1,2,...,M\}$, and each subproblem is assigned to a different agent. The goals of the decomposition are twofold: first, to ensure that each subproblem is much smaller than the overall problem (that is, to ensure that SP_{ki} has far fewer decision variables and constraints than SP_k), and second, to ensure that SP_{ki} is coupled to only a few other subproblems (that is, SP_{ki} shares variables with only a few other subproblems).

To be more specific, consider the ith subproblem and the corresponding agent. From the ith agent's point of view, the goals of the decomposition are to partition the set of variables S into three subsets: $S=S_i\cup S_i^{\text{nei}}\cup S_i^{\text{rem}}$, where S_i, called local variables, is the set of decision variables allocated to the ith agent; S_i^{nei}, called neighbor variables, is the set of decision variables allocated to agent i's neighbors (the agents with which agent i can cooperate—exchange data); and S_i^{rem}, called remote variables, is the set of all other decision variables in the system. Problem SP_{ki} can then be formulated as follows:

$$SP_{ki}: \min_{S_i} J_i\left(S_i,S_i^{nei}\right)$$
$$\text{s.t.} \quad G_i\left(S_i,S_i^{nei}\right) \le 0$$
$$H_i\left(S_i,S_i^{nei}\right)=0,$$

where G_i and H_i are components of G and H related to agent i, and the performance index J_i represents the interests of agent i.

We assume that the network of interactions between the subsystems is sparse, which means $\dim(S_i)+\dim(S_i^{\text{nei}}) << \dim(S)$, and require that some agent handles each decision variable and constraint, i.e., $\cup_i S_i=S$, $\cup_i G_i=G$, $\cup_i H_i=H$. Either the objective functions sum to a given global objective function, $\Sigma J_i=J$, or the global problem can be thought of as a multiobjective optimization problem with the vector objective function, $J=\left[J_1,...,J_M\right]^T$. We believe that it is possible to develop systematic procedures for performing the above decomposition for many, if not all, complex networks.

Cooperative Iteration

In this section we consider the case when the communication network allows the agents to exchange information while they solve their local optimization problems. In particular, we consider a scheme in which each agent computes a solution to its local problem assuming values for the variables of its neighbors. The agent then broadcasts the values of its own variables to its neighbors and resolves its optimization problem with the updated values for the shared variables. The objective of the coordination is to achieve convergence in the values of the variables shared by multiple agents.

For $i=1,...,M$, let $s_{ki}^0, s_{ki}^1, s_{ki}^2, ...$ be the resulting sequence of the iterative search at time k for a solution to subproblem SP_{ki}. Two important questions are:

- Under what conditions will these iterations converge to a solution of SP_{ki}?
- Under what conditions will the solutions of $SP_{k1}, SP_{k2}, ..., SP_{kM}$ compose a solution of the overall problem SP_k?

The following theorem provides answers to these two questions.

Theorem 1 [15]: If, for all i:

1) $\Sigma J_i=J$ when J is a scalar, or $\Sigma J_i=W^TJ$ when J is a multiobjective vector, where W is a vector of nonnegative weights;
2) J_i and G_i are convex;
3) H_i is linear;
4) The agents within each neighborhood work sequentially;
5) The equality constraints can be relaxed without emptying the feasible region of SP_k. In other words, there exists a positive number δ such that

$$\left\{s|G(s)<0, |H(s)|<\delta\right\}\neq\varnothing;$$

6) J_i is bounded from below in the feasible region;
7) The starting point is in the interior of the feasible region;
8) Each agent cooperates with its neighbors in that it broadcasts its latest iteration to these neighbors;

9) Each agent uses the same interior-point method (barrier method) with the same Lagrange multipliers to generate its iterations.

Then: a) in the case where J is a single objective, SP_k has a unique solution and $\{\cup s_{ki}^l | l = 0,1,2,...\}$ will converge to this solution; b) in the case where J is a multiobjective vector, $\{\cup s_{ki}^l | l = 0,1,2,...\}$ will converge to a point on the Pareto surface of SP_k. (When there are multiple, conflicting objectives, the Pareto surface is the set of the best possible tradeoffs among these objectives.)

In essence, this theorem indicates when computational advantages can be obtained by tackling a network of subproblems with a network of agents that has the same structure. To be more specific, consider two sparse graphs: Γ, whose nodes represent subproblems and whose arcs represent couplings between (or variables shared by) these subproblems; and Γ', whose nodes represent agents and whose arcs represent communication channels between these agents. If a large optimization problem can be decomposed into a network of much smaller subproblems represented by Γ, and if conditions 1 through 9 are met, then solving the subproblems with agents arranged so that $\Gamma' = \Gamma$ provides the following advantages: locally optimal solutions (those found by the agents) are coincident with the globally optimal solution; each agent needs to cooperate (exchange data) only with its neighbors (adjacent nodes in Γ'); and agents not in the same neighborhood may work in parallel.

Conditions 2 and 3, however, are unrealistic (real problems are often nonconvex and have nonlinear equality constraints), and condition 4 is overly constraining (we would like the agents to be able to work asynchronously: all in parallel, each at its own speed).

Experiments on a number of small but prototypical networks [15], [16] indicate that conditions 2 and 3 are not necessary and can often be relaxed to allow for nonconvex problems with nonlinear equality constraints. This suggests that when distributed controls are to be designed for a real network, experiments should be conducted to see if the network would allow conditions 2 and 3 to be relaxed.

We do not, as yet, have a procedure for relaxing condition 4 to allow for at least some periods of asynchronous work. The next section suggests some promising directions.

Heuristics for Asynchronous Work

Consider the formulation of SP_{ki}, the subproblem to be solved by the ith agent. Two classes of heuristics for tackling this subproblem asynchronously are:

- Using models to predict the reactions of agent i's neighbors, so agent i does not have to wait to be informed of these reactions, but rather can proceed with its iterations on the basis of predictions of S_i^{nei}. These models can either be developed from first principles or learned from historical records of S_i^{nei}.
- Tightening the constraints on some agents so they leave some maneuvering room for other agents.

Both classes show promise. For instance, the inequality constraints can be tightened as follows:

$$G_i\left(S_i, S_i^{\text{nei}}\right) \le -R_i,$$

where the resource margin, $R_i > 0$, can either be fixed, a priori, or adjusted dynamically.

Fig. 1 presents some experimental results for a "forest of pendulums." This forest consists of an array of up to nine frictionless pendulums [15], [16]. Each pendulum is connected to its adjacent pendulums by linear springs and is controlled by an agent that can exert two orthogonal and horizontal forces on the pendulum. At the start of the experiment, the pendulums are oscillating in synchronism. Subsequently, they are subjected to a set of random disturbances. The objective is to return them to synchronism as soon as possible while expending as little control energy as possible [16]. The mechanical connections of the pendulums make this optimization problem profoundly nonconvex with equality constraints that are profoundly nonlinear. Nevertheless, asynchronous calculation schemes for the agents produce convergence to solutions not very different from the optimal solution, as shown in Fig. 1.

Coordination for Stability

The coordination scheme in the previous sections allows the agents to compute a set of solutions to the local problems that also solves a global optimization problem, thereby making it possible to emulate centralized MPC through distributed computations. We now consider a second scenario in which it is only possible for the agents to communicate the solutions to their local problems once during each control interval. In this case, the collection of local solutions is not equivalent to the solution of a global problem because the agents are using information from their neighbors that is delayed by one time step. Consequently, the stability results for

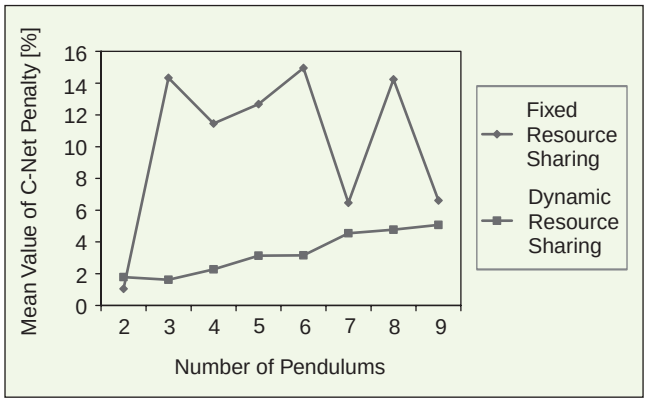

Figure 1. *Typical results of asynchronous iterations with a resource-margin heuristic. The "C-Net Penalty" is the percentage deviation from the optimal solution. The upper curve corresponds to fixed and equal resource margins for all pendulums. The lower curve was obtained with resource margins that were smaller for the pendulums suffering the greatest deviations from the desired behavior.*

the centralized MPC cannot be used here directly; new conditions for stability are required. In this article, we study linear systems without constraints as the first attempt in this direction for distributed MPC.

In the literature, to ensure stability, some constraints or conditions are added to the MPC optimization problem. There are two kinds of schemes for applying stability constraints [1], [3]. In the dominant scheme, stability constraints are applied to the end state in the prediction. The prediction horizon must then be set long enough so that a feasible solution exists. This method is not suitable for distributed MPC because it is unclear how long the prediction horizon should be when only the local information is used.

The second method is a recently proposed scheme by Cheng and Krogh called stability-constrained model predictive control (SC-MPC) [17]-[21]. In this approach, a contractive constraint computed online (rather than offline, as in most previous schemes) is imposed on the first state in the prediction. The selection of the prediction horizon does not affect the stability of the system.

In this section, we apply the SC-MPC approach to our distributed MPC scheme. We present sufficient conditions for closed-loop system stability using distributed MPC with contractive stability constraints, called stability-constrained distributed model predictive control (SC-DMPC).

A distributed linear time-invariant system with each subsystem controllable and coupling only in state variables can be modeled as

$$\begin{bmatrix} x_1(k+1) \\ \vdots \\ x_M(k+1) \end{bmatrix} = \begin{bmatrix} A_{11} & \cdots & A_{1M} \\ \vdots & \ddots & \vdots \\ A_{M1} & \cdots & A_{MM} \end{bmatrix} \begin{bmatrix} x_1(k) \\ \vdots \\ x_M(k) \end{bmatrix} + \begin{bmatrix} B_1 & & \\ & \ddots & \\ & & B_M \end{bmatrix} \begin{bmatrix} u_1(k) \\ \vdots \\ u_M(k) \end{bmatrix}, \tag{1}$$

where $x_i \in R^{n_i}$ and $u_i \in R^{m_i}$ are the state vector and the control vector of the ith subsystem, respectively. For a system of this type, we propose the following scheme to achieve coordination among agents. During each step, each agent only broadcasts its solution of the local MPC problem after it applies its control action to the local subsystem. In the computation, agents use the information they get from neighbor agents to estimate the effect from neighbor subsystems, meaning that each agent uses the predictions of neighbor agents at the previous step to estimate the effect from neighbor subsystems. For the jth agent, we denote the information from the other agents by the vector

$$v_j(k+i|k) = \left[x_1^T(k+i|k-1) \cdots x_{j-1}^T(k+i|k-1) \; x_{j+1}^T(k+i|k-1) \cdots x_M^T(k+i|k-1) \right]^T,$$

where $x_s(k+i|k-1)$ is the state prediction by the agent s at the control step $k-1$. The agent j uses the following model to predict the future states of the local part:

$$\begin{aligned} x_j(k+i+1|k) &= A_{jj}x_j(k+i|k) + B_j u_j(k+i|k) \\ &\quad + \sum_{\substack{s=1 \\ s\neq j}}^{M} A_{js}x_s(k+i|k-1) \\ &= A_{jj}x_j(k+i|k) + B_j u_j(k+i|k) + K_j v_j(k+i|k), \end{aligned}$$

where $K_j = [A_{j1} \cdots A_{j,j-1} A_{j,j+1} \cdots A_{jM}]$. The SC-DMPC algorithm is based on the following lemmas. Proofs of all the results in this section can be found in [22].

Lemma 1: Consider a system in (1). Suppose (A_{ii}, B_i) $(i = 1,\ldots,M)$ are controllable and B_i is of full rank. If the system satisfies the condition that $\text{rank}([B_i A_{i1} \cdots A_{i,i-1} A_{i,i+1} \cdots A_{iM}]) = m_i$, there exists a similarity transform matrix $P = \text{diag}(P_1, P_2, \ldots, P_M)$ such that the system can be represented in the controllable companion form given by

$$\begin{bmatrix} x_1^1(k+1) \\ x_1^2(k+1) \\ \vdots \\ x_M^1(k+1) \\ x_M^2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & I_{11} & \cdots & 0 & 0 \\ A_{11}^1 & A_{11}^2 & \cdots & A_{1M}^1 & A_{1M}^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & I_{MM} \\ A_{M1}^1 & A_{M1}^2 & \cdots & A_{MM}^1 & A_{MM}^2 \end{bmatrix} \times \begin{bmatrix} x_1^1(k) \\ x_1^2(k) \\ \vdots \\ x_M^1(k) \\ x_M^2(k) \end{bmatrix} + \begin{bmatrix} 0 & \cdots & 0 \\ B_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \\ 0 & \cdots & B_M^2 \end{bmatrix} \begin{bmatrix} u_1(k) \\ \vdots \\ u_M(k) \end{bmatrix} \tag{2}$$

where

$$Px = \begin{bmatrix} x_1^1 \\ x_1^2 \\ \vdots \\ x_M^1 \\ x_M^2 \end{bmatrix}, \quad x_i^1 \in R^{m_i} \text{ and } x_i^2 \in R^{n_i - m_i},$$

$$PAP^{-1} = \begin{bmatrix} 0 & I_{11} & \cdots & 0 & 0 \\ A_{11}^1 & A_{11}^2 & \cdots & A_{1M}^1 & A_{1M}^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & I_{MM} \\ A_{M1}^1 & A_{M1}^2 & \cdots & A_{MM}^1 & A_{MM}^2 \end{bmatrix}$$

$$I_{ii} \in R^{(n_i - m_i)\times(n_i - m_i)}, \; A_{ij}^1 \in R^{m_i \times m_j} \text{ and } A_{ij}^2 \in R^{m_i \times (n_j - m_j)},$$

$$PB = \begin{bmatrix} 0 & \cdots & 0 \\ B_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \\ 0 & \cdots & B_M^2 \end{bmatrix}, \; B_i^2 \in R^{m_i \times m_i}.$$

Lemma 2: Consider a system in controllable companion form (2). Suppose states are measurable. To each subsystem, for any $x_i(k)$ and $0 < \beta_i < 1$, there exists $u_i(k)$ such that

$$\|x_i(k+1)\|^2 \le \|x_i(k)\|^2 - \beta_i \|x_i^1(k)\|^2. \tag{3}$$

Moreover, if

$$u(k) = \begin{bmatrix} u_1(k) \\ \vdots \\ u_M(k) \end{bmatrix},$$

where $u_i(k)$ satisfies (3) and $\beta = \min(\beta_1, \cdots, \beta_M)$, then

$$\|x(k+1)\|^2 \le \|x(k)\|^2 \beta \|x^1(k)\|^2,$$

where

$$x^1(k) = \begin{bmatrix} x_1^1(k) \\ \vdots \\ x_M^1(k) \end{bmatrix}.$$

To ensure the stability of the system, each agent also adds a contractive constraint into its local MPC problem. For the jth agent, the algorithm is described as follows.

SC-DMPC Algorithm

Step 1. *Communication:* Send out its previous predictions $X_j(k-1)$ to other controllers and also get information $V_j(k) = \{v_j(k|k), \ldots, v_j(k+N-1|k)\}$ from other controllers.

Step 2. *Initialization:* Given the measured $x_j(k)$ and $l_j(k)$ from the previous iteration (set $l_j(0)$ to be an arbitrary number), and $0 < \beta_j \le 1$, define

$$\hat{l}_j(k) = \max\left\{ l_j(k), \|x_j(k)\|^2 \right\} - \beta_j \|x_j^1(k)\|^2.$$

Set $\hat{x}_j(k|k) = x_j(k)$.

Step 3. *Optimization:* Solve the following optimal control problem.

$$\min_{X_j(k), U_j(k)} J_j(X_j(k), U_j(k))$$

subject to

$$x_j(k+i+1|k) = A_{jj} x_j(k+i|k) + B_j u_j(k+i|k) + K_j v_j(k+i|k), \quad i = 0, 1, \ldots, N-1$$

$$\|x_j(k+1|k)\|^2 \le \hat{l}_j(k).$$

Step 4. *Assignment:* Let

$$u_j(k) = u_j(k|k), \; l_j(k+1) = \|x_j(k+1|k)\|^2.$$

Step 5. *Implementation:* Apply the control $u(k)$. Set $k = k+1$ and return to step 1 at the next sample time.

Note that each controller does not need to communicate with all the other controllers. Controllers i and j would communicate with each other only if the subsystems i and j have direct interaction with each other. If the system is loosely coupled, the amount of communication is not that great.

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 interagent communication network.

Lemma 2 shows that when the contractive constraint $\|x_j(k+1|k)\|^2 \le \hat{l}_j(k)$ is added, the feasible control set is still nonempty. It also guarantees that at each step, the collection of solutions for the local problems comprises a feasible solution for the overall system. The following theorem gives a sufficient condition for stability of the closed-loop system.

Theorem 2: Consider a system in the controllable companion form (2). The control is computed at each control instant using SC-DMPC. The system is asymptotically stable if the following matrix is stable:

$$\tilde{A} = \begin{bmatrix} 0 & A_{12}^2 & \cdots & A_{1M}^2 \\ A_{21}^2 & 0 & \cdots & A_{2M}^2 \\ \vdots & \vdots & \ddots & \vdots \\ A_{M1}^2 & A_{M2}^2 & \cdots & 0 \end{bmatrix}.$$

A Power System Application

In a power system with two or more independently controlled areas, the generation within each area must be con-

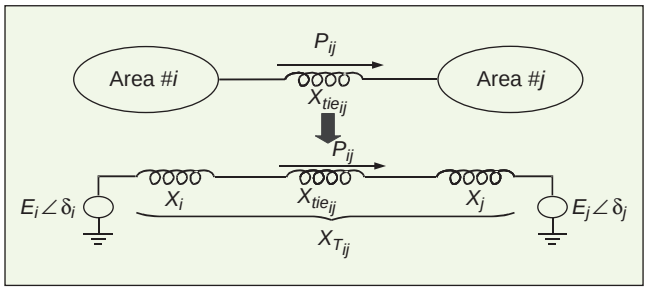

Figure 2. *A simplified description of a power system for studying load frequency control (LFC).*

trolled so as to maintain the system frequency and the scheduled power exchange. This function is commonly referred to as load-frequency control (LFC), for which centralized control is not practical. Many decentralized control schemes have been proposed for the LFC problem [23]-[25]. Here we handle the LFC problem by using the SC-DMPC scheme. We assign MPC controllers to control the generator power output directly. Thus, each area can be described as one equivalent generator in series with impedance, as shown in Fig. 2.

The dynamic equation model of each area can be expressed as

$$\Delta\dot{\delta}_i(t) = 2\pi\Delta f_i(t)$$
$$\Delta\dot{f}_i(t) = \frac{-\Delta f_i(t)}{T_{P_i}} + \frac{K_{P_i}\Delta P_{g_i}(t)}{T_{P_i}} - \frac{K_{P_i}}{2\pi T_{P_i}} \times \left(\sum_{j\in N} K_{S_{ij}}[\Delta\delta_i(t) - \Delta\delta_j(t)]\right) - \frac{K_{P_i}\Delta P_{d_i}(t)}{T_{P_i}},$$

where

$\Delta\delta_i(t)$: incremental phase angle deviation of the area bus in rad;

$\Delta f_i(t)$: incremental frequency deviation in Hz;

$\Delta P_{g_i}(t)$: incremental change in the generator output in p.u.;

$\Delta P_{d_i}(t)$: load disturbance for the ith area in p.u.;

T_{P_i}: system model time constant in s;

K_{P_i}: system gain;

$K_{S_{ij}}$: synchronizing coefficient of the tie-line between ith and jth areas.

The distributed MPC controllers coordinate the generator outputs by providing set points to the turbine controllers. Here, the dynamics of turbines will not be included. The objective of LFC is to keep the frequency deviation of the system at zero and to maintain the deviation of the power flow through the tie-line at zero. The deviation of power flow between areas is proportional to the difference of phase angle deviation between areas. Thus, the local performance index can be selected as

$$J_i(t) = \int_t^{t+T} \left\{ \sum_{\substack{j=1 \\ j\neq i}}^{M} p_{ij}(\Delta\delta_i(\tau) - \Delta\delta_j(\tau))^2 + q_i\Delta f_i^2(\tau) + r_i\Delta P_{g_i}^2(\tau) \right\} d\tau.$$

In the simulation study, a two-area LFC scenario is constructed. Each controller is assumed to know the load disturbance in its local area. The parameters used in the simulation are: area #1: $T_{P_1} = 25$ s, $K_{P_1} = 112.5$, $K_{S_{12}} = 0.5$; area #2: $T_{P_2} = 20$ s,

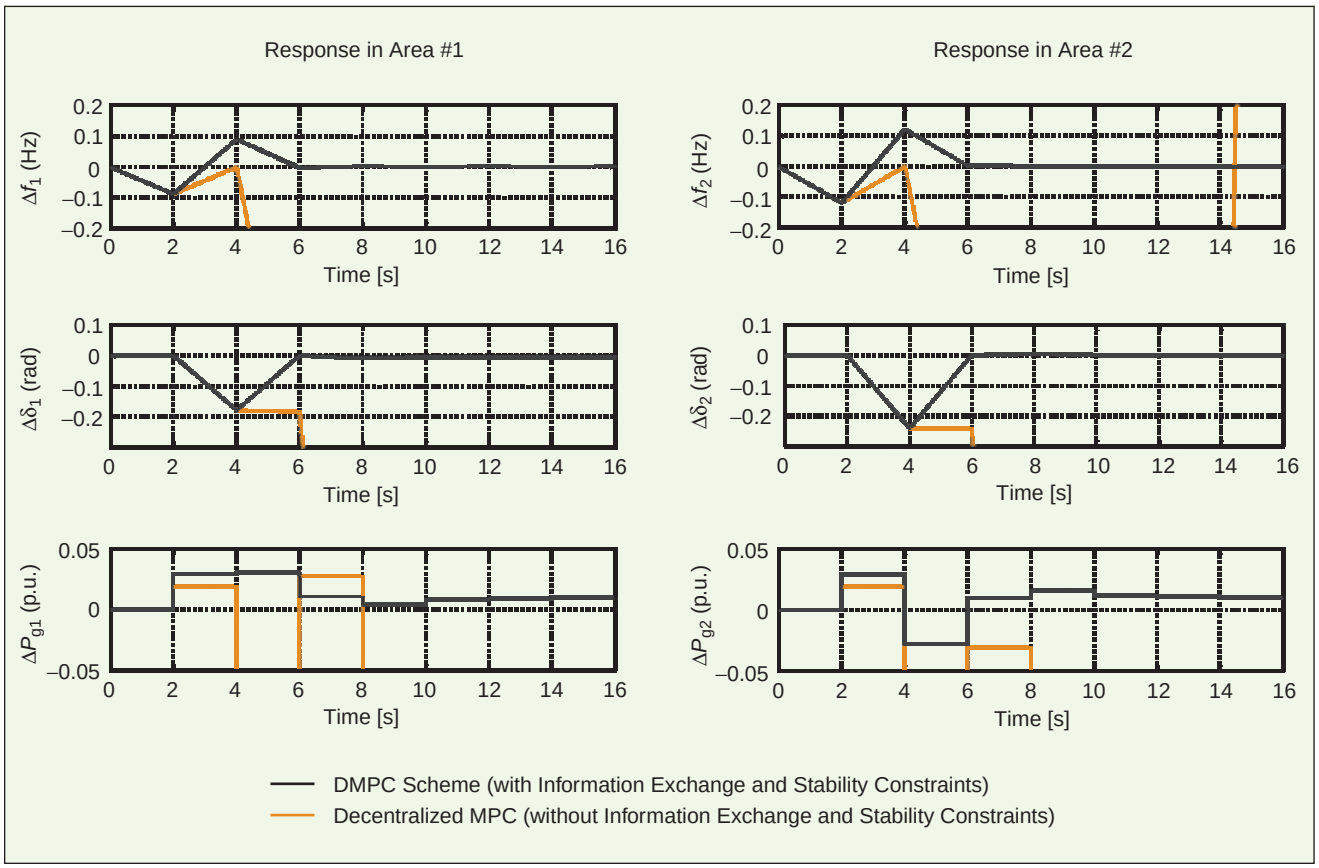

Figure 3. *Response of the LFC system for SC-DMPC and decentralized MPC without stability constraints or information exchange (simulation of the discrete-time model).*

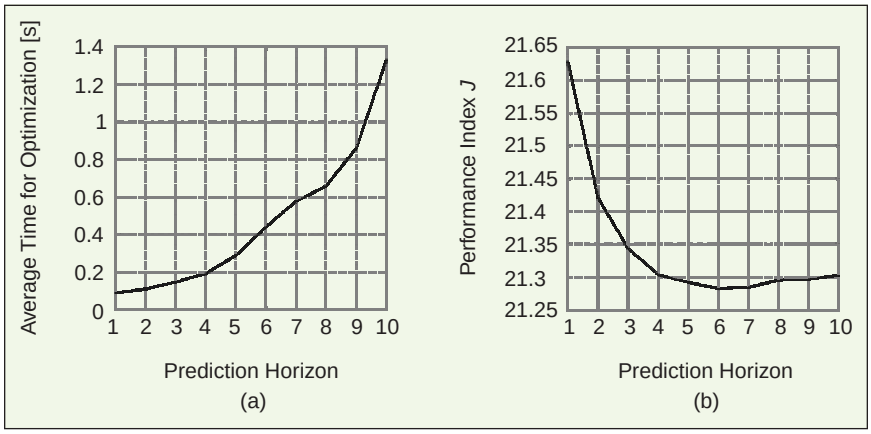

Figure 4. *The computation time and the performance achieved for different prediction horizons. The performance is evaluated over the simulation time. The computation time is the average time used in optimization in one agent.*

$K_{P_2} = 120$, $K_{S_{21}} = 0.5$. Other parameters for optimization are $p_{12} = p_{21} = 10, q_1 = q_2 = 100$, and $r_1 = r_2 = 10$. The constants for stability constraints are $\beta_1 = \beta_2 = 0.8$. The load disturbance is 0.01 p.u. load increment in each area at time 0. The control interval is set to be 2.0 s. The prediction horizon is selected to be unity to minimize the amount of computation and communication. By the Euler method, a discrete-time state-space model of the system can be obtained. The model satisfies the condition for stability in Theorem 2.

Fig. 3 shows the simulation results. The blue curves are system behavior by the SC-DMPC algorithm. The frequency variations go to zero, and each subsystem can provide enough power for its local load increment. The red curves are system behavior by MPC in a decentralized fashion, but there is no information exchange among the controllers and no stability constraint either. The system is unstable under the disturbance. Centralized MPC with stability constraints is also tried. The system performance is not much better than that using SC-DMPC. They are nearly the same. From the simulation we find that SC-DMPC can work as well as centralized MPC for systems with canonical structure. Although the model using the Euler method is only a rough approximation of the real system, this example provides some understanding about how the distributed MPC scheme performs.

Another notable point of this scheme is that, for interconnected linear time-invariant systems satisfying the conditions of Lemma 1 and Theorem 2, the scheme imposes no constraint on the selection of prediction horizon, providing more flexibility in selecting this value. Obviously, a short prediction horizon would require a smaller amount of computation time, but a properly chosen longer prediction horizon could improve the performance of the system. As shown in Fig. 4(a), the average time for optimization increases exponentially as the prediction horizon increases; but, as shown in Fig. 4(b), the performance is improved as the prediction horizon increases until it reaches six. We see that too long a prediction horizon can degrade the performance because the errors in the prediction are very large. Thus, by applying this distributed MPC scheme, one can obtain a compromise between the potential improvement in performance and the prediction errors.

Conclusion

This article has presented new results for distributed model predictive control, focusing on i) the coordination of the optimization computations using iterative exchange of information and ii) the stability of the closed-loop system when information is exchanged only after each iteration. Current research is focusing on general methods for decomposing large-scale problems for distributed MPC and methods for guaranteeing stability when multiple agents are controlling systems subject to abrupt changes.

Acknowledgment

This research was supported by the EPRI/DoD Complex Interactive Networks/Systems Initiative.

References

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

[2] E.F. Camacho and C. Bordons, *Model Predictive Control in the Process Industry*. New York: Springer, 1995.

[3] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, 2000, pp. 789-814.

[4] M. Morari and J.H. Lee, "Model predictive control: Past, present and future," *Comput. Chem. Eng.*, vol. 23, no. 4-5, 1999, pp. 667-682.

[5] N.R. Sandell, P. Varaiya, M. Athans, and M.G. Safonov, "Survey of decentralized control methods for large scale systems," *IEEE Trans. Automat. Contr.*, vol. AC-23, pp. 108-128, Feb. 1978.

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

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

[8] 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.

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

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

[11] G. Georges, "Decentralized adaptive control for a water distribution system," in *Proc. 3rd IEEE Conf. Control Applications*, Glasgow, UK, 1994, pp. 1411-1416.

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

[13] 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*, Trieste, Italy, 1998, pp. 16-20.

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

[15] E. Camponogara, "Controlling networks with collaborative nets," Ph.D. dissertation, Carnegie Mellon Univ., 2000.

[16] S.N. Talukdar and E. Camponogara, "Collaborative nets," in *Proc. 33rd Hawaii Int. Conf. System Sciences*, Hawaii, 2000.

[17] X. Cheng, "Model predictive control and its application to plasma enhanced chemical vapor deposition," Ph.D. dissertation, Carnegie Mellon Univ., 1996.

[18] X. Cheng and B.H. Krogh, "Stability-constrained model predictive control with state estimation," in *Proc. American Control Conf.*, Albuquerque, NM, 1997, pp. 2493-2498.

[19] X. Cheng and B.H. Krogh, "Stability constrained model predictive control for nonlinear systems," in *Proc. American Control Conf.*, San Diego, CA, 1998, pp. 2091-2096.

[20] X. Cheng and B.H. Krogh, "Nonlinear stability-constrained model predictive control with input and state constraints," in *Proc. 36th Conf. Decision and Control*, Philadelphia, PA, 1998, pp. 1674-1678.

[21] X. Cheng and B.H. Krogh, "Stability-constrained model predictive control," *IEEE Trans. Automat. Contr.*, vol. 46, pp. 1816-1820, Nov. 2001.

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

[23] P.K. Chawdhry and S.I. Ahson, "Application of interaction-prediction approach to load-frequency control (LFC) problem," *IEEE Trans. Syst., Man, Cyberneti.*, vol. SMC-12, no. 1, 1982, pp. 66-71.

[24] K.Y. Lim, Y. Wang, G. Guo, and R. Zhou, "A new decentralized robust controller design for multiarea load-frequency control via incomplete state feedback," *Opt. Contr. Applicat. Methods*, vol. 19, 1998, pp. 345-361.

[25] A. Rubaai and J.A. Momoh, "Three-level control scheme for load-frequency control of interconnected power systems," in *Proc. TENCON'91 IEEE Region 10 Int. Conf. EC3-Energy, Computer, Communication and Control Systems*, 1991, pp. 63-70.

Eduardo Camponogara received the B.Sc. and M.Sc. degrees in computer science in Brazil and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University in 2000. He is currently with the Department of Automation and Systems at the Federal University of Santa Catarina, Brazil, where he is conducting research on distributed decision making, optimization, and control with applications in the operation of large, dynamic networks.

Dong Jia received the B.S. degree in automatic control and the M.S. degree in control theory and applications from Xi'an Jiaotong University, China, in 1996 and 1999, respectively. He is currently a Ph.D. candidate in electrical and computer engineering at Carnegie Mellon University. His research interests include model predictive control and distributed coordination of controllers in large- scale systems.

Bruce H. Krogh received the B.S. degree in mathematics and physics from Wheaton College, Wheaton, IL, in 1975 and the Ph.D. degree in electrical engineering from the University of Illinois, Urbana, in 1983. He joined Carnegie Mellon University in 1983, where he is currently a Professor of Electrical and Computer Engineering. His research interests include discrete-event and hybrid dynamic systems, distributed control strategies, and the interfaces between control and computer science.

Sarosh N. Talukdar studied at the IIT-Madras and Purdue University. After earning his doctorate, he worked at McGraw-Edison for several years, where he developed simulation and optimization software for electric power circuits. In 1974, he joined Carnegie Mellon University, where he is a Professor of Electrical and Computer Engineering. He works in the areas of engineering design, electric power systems, distributed decision making, and complex networks.