Distributed MPC Strategies With Application to Power System Automatic Generation Control

Aswin N. Venkat, Ian A. Hiskens, *Fellow, IEEE*, James B. Rawlings, and Stephen J. Wright

***Abstract*—A distributed model predictive control (MPC) framework, suitable for controlling large-scale networked systems such as power systems, is presented. The overall system is decomposed into subsystems, each with its own MPC controller. These subsystem-based MPCs work iteratively and cooperatively towards satisfying systemwide control objectives. If available computational time allows convergence, the proposed distributed MPC framework achieves performance equivalent to centralized MPC. Furthermore, the distributed MPC algorithm is feasible and closed-loop stable under intermediate termination. Automatic generation control (AGC) provides a practical example for illustrating the efficacy of the proposed distributed MPC framework.**

***Index Terms*—Automatic generation control, distributed model predictive control, power system control.**

I. Introduction

MODEL predictive control (MPC) is widely recognized as a high performance, yet practical, control technology. This model-based control strategy uses a prediction of system response to establish an appropriate control response. An attractive attribute of MPC technology is its ability to systematically account for process constraints. The effectiveness of MPC is dependent on a model of acceptable accuracy and the availability of sufficiently fast computational resources. These requirements limit the application base for MPC. Even so, applications abound in the process industries, and are becoming more widespread [7], [28].

Traditionally, control of large, networked systems is achieved by designing local, subsystem-based controllers that ignore the interactions between the different subsystems. A survey of decentralized control methods for large-scale systems is available in [29]. It is well known that a decentralized control philosophy may result in poor systemwide control performance if the subsystems interact significantly. Centralized MPC, on the other hand, is impractical for control of large-scale, geographically expansive systems, such as power systems. A distributed MPC framework is appealing in this context; the distributed MPC controllers must, however, account for the interactions between the subsystems. These and other issues critical to the success of distributed MPC are examined in this paper.

Each MPC, in addition to determining the optimal current response, also generates a prediction of future subsystem behavior. By suitably leveraging this prediction of future subsystem behavior, the various subsystem-based MPCs can be integrated and the overall system performance improved. A discussion on economic and performance benefits attainable by integrating subsystem-based MPCs is available in [17] and [24]. One of the goals of this paper, however, is to illustrate that a simple exchange of predicted subsystem trajectories (*communication*) does not necessarily improve overall system control performance.

A few distributed MPC formulations are available in the literature. A distributed MPC framework was proposed in [13], for the class of systems that have independent subsystem dynamics but are linked through their cost functions. More recently in [12], an extension of the method described in [13] that handles systems with weakly interacting subsystem dynamics was proposed. Stability is proved through the use of a conservative, consistency constraint that forces the predicted and assumed input trajectories to be close to each other. Also, as pointed out by the author, the performance of the distributed MPC framework in [12] is, in most cases, different from that of centralized MPC. A distributed MPC algorithm for unconstrained, linear time-invariant (LTI) systems was proposed in [8] and [20]. For the models considered in [8] and [20], the evolution of the states of each subsystem is assumed to be influenced only by the states of interacting subsystems and local subsystem inputs. This choice of modeling framework can be restrictive. In many cases, such as the two area power network with FACTS device (see Section V-G3) and most chemical plants, the evolution of the subsystem states is also influenced by the inputs of interconnected subsystems. More crucially for the distributed MPC framework proposed in [8] and [20], the subsystem-based MPCs have no knowledge of each other's cost/utility functions. It is known from noncooperative game theory that if such pure *communication-based strategies* (in which competing agents have no knowledge of each others cost functions) converge, they converge to the Nash equilibrium (NE) [2], [3]. In most cases involving a finite number of agents,

Manuscript received August 30, 2006; revised June 6, 2007. Manuscript received in final form October 10, 2008. First published June 10, 2008; current version published October 22, 2008. Recommended by Associate Editor D. Rivera. This work was supported in part by the Texas-Wisconsin Modeling and Control Consortium and the National Science Foundation (NSF) under Grant CTS-0456694. This paper was presented in part at the *IFAC Symposium on Power Plants and Power Systems Control*, Kananaskis, Canada, June 25-28, 2006.

A. N. Venkat was with the Department of Chemical and Biological Engineering, University of Wisconsin, Madison, WI-53706 USA. He is now with the Process Modeling, Control, and Optimization Group, Shell Global Solutions, Westhollow Technology Center, Houston, TX 77210 USA (e-mail: venkat@bevo.che.wisc.edu).

I. A. Hiskens is with Faculty of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706 USA (e-mail: hiskens@engr.wisc.edu).

J. B. Rawlings is with Faculty of Chemical and Biological Engineering, University of Wisconsin, Madison, WI 53706 USA (e-mail: rawlings@engr.wisc.edu).

S. J. Wright is with Faculty of Computer Sciences, University of Wisconsin, Madison, WI 53706 USA (e-mail: swright@cs.wisc.edu).

Digital Object Identifier 10.1109/TCST.2008.919414

the NE is different from the Pareto optimal (PO) solution [10], [11], [26]. In fact, nonconvergence or suboptimality of pure communication-based strategies may result in unstable closed-loop behavior in some cases. A four area power network example is used here (see Section V-G2) to illustrate instability due to communication-based MPC. Such examples are not uncommon. A distributed MPC framework in which the effect of interconnected subsystems are treated as bounded uncertainties was proposed in [21]. Stability and optimality properties have not been established however.

Most interconnected power systems rely on automatic generation control (AGC) for regulating system frequency and tie-line interchange [37]. These objectives are achieved by controlling the real power output of generators throughout the system, taking into account restrictions on the amount and rate of generator power deviations. To cope with the expansive nature of power systems, a distributed control structure has been adopted for AGC. The current form of AGC may not, however, be well suited to future power systems [1], with various trends set to impact its effectiveness.

Future power systems will see greater use of flexible ac transmission system (FACTS) devices [18]. These devices allow control of power flows over selected paths through a transmission network, offering economic benefits [23] and improved security [14]. However, FACTS controllers must be coordinated with other power system controls, including AGC. On the other hand, greater utilization of intermittent renewable resources, such as wind generation, brings with it power flow fluctuations that are difficult to regulate [27].

These changes provide an opportunity to rethink AGC. Distributed MPC offers an effective means of achieving the desired controller coordination and performance improvements, whilst alleviating the organizational and computational burden associated with centralized control. AGC therefore provides a very relevant example for illustrating the performance of distributed MPC in a power system setting.

This paper is organized as follows. In Section II, a brief description of the different modeling frameworks is presented. Notation used in this paper is introduced in Section III. In Section IV, a description of the different MPC-based systemwide control frameworks is provided. An implementable algorithm for terminal penalty distributed MPC is described in Section V. Properties of this distributed MPC algorithm and closed-loop properties of the resulting distributed controller are established subsequently. Three examples are presented to highlight the performance benefits of terminal penalty distributed MPC. A framework for terminal control distributed MPC is introduced in Section VI. In Section VII, the main contributions of this study are summarized, and various extensions are reported.

II. MODELS

Distributed MPC relies on decomposing the overall system model into appropriate subsystem models. A system comprised of M interconnected subsystems will be used to establish these concepts.

Centralized Model: The overall system model is represented as a discrete, linear time-invariant (LTI) model of the form

$$x(k+1) = Ax(k) + Bu(k), \quad y(k) = Cx(k)$$

in which k denotes discrete time and

$$A = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1M} \\ \vdots & \vdots & \ddots & \vdots \\ A_{i1} & A_{i2} & \dots & A_{iM} \\ \vdots & \vdots & \ddots & \vdots \\ A_{M1} & A_{M2} & \dots & A_{MM} \end{bmatrix}$$
$$B = \begin{bmatrix} B_{11} & B_{12} & \dots & B_{1M} \\ \vdots & \vdots & \ddots & \vdots \\ B_{i1} & B_{i2} & \dots & B_{iM} \\ \vdots & \vdots & \ddots & \vdots \\ B_{M1} & B_{M2} & \dots & B_{MM} \end{bmatrix}$$
$$C = \begin{bmatrix} C_{11} & 0 & \dots & 0 \\ 0 & C_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & C_{MM} \end{bmatrix}$$
$$u = [u_1{}' \quad u_2{}' \quad \dots \quad u_M{}']' \in \mathbb{R}^m$$
$$x = [x_1{}' \quad x_2{}' \quad \dots \quad x_M{}']' \in \mathbb{R}^n$$
$$y = [y_1{}' \quad y_2{}' \quad \dots \quad y_M{}']' \in \mathbb{R}^z.$$

For each subsystem $i = 1, 2, \dots, M$, the triplet (u_i, x_i, y_i) represents the subsystem input, state, and output vector, respectively. The centralized model pair (A, B) is assumed to be stabilizable and (A, C) is detectable.[1]

Decentralized Model: In the decentralized modeling framework, it is assumed that the interaction between the subsystems is negligible. Subsequently, the effect of the external subsystems on the local subsystem is ignored in this modeling framework. The decentralized model for subsystem $i = 1, 2, \dots, M$ is

$$x_i(k+1) = A_{ii}x_i(k) + B_{ii}u_i(k), \quad y_i(k) = C_{ii}x_i(k).$$

Partitioned Model (PM): The PM for subsystem i combines the effect of the local subsystem variables and the effect of the states and inputs of the interconnected subsystems. The PM for subsystem i is obtained by considering the relevant partition of the centralized model and can be explicitly written as

$$x_i(k+1) = A_{ii}x_i(k) + B_{ii}u_i(k) + \sum_{j \neq i} \left(A_{ij}x_j(k) + B_{ij}u_j(k)\right) \tag{1a}$$

$$y_i(k) = C_{ii}x_i(k). \tag{1b}$$

III. NOTATION

For any matrix P, $\lambda_{\max}(P)$, and $\lambda_{\min}(P)$ denote the maximum and minimum (absolute) eigenvalue of P, respectively. For any subsystem $i = 1, 2, \dots, M$, let the predicted state and

[1]In the applications considered here, local measurements are typically a subset of subsystem states. The structure selected for the C matrix reflects this observation. A general C matrix may be used, but impacts possible choices for distributed estimation techniques [35].

input at time instant $k+j, j \geq 0$, based on data at time k be denoted by $x_i(k+j \mid k) \in \mathbb{R}^{n_i}$ and $u_i(k+j \mid k) \in \Omega_i \subset \mathbb{R}^{m_i}$, respectively, where Ω_i is the set of admissible controls for subsystems i. We have the following definitions for the infinite horizon predicted state and input trajectory vectors in the different MPC frameworks:

Centralized state trajectory:
$$\boldsymbol{x}(k)' = [x(k+1 \mid k)', x(k+2 \mid k)', \ldots]$$
Centralized input trajectory:
$$\boldsymbol{u}(k)' = [u(k \mid k)', u(k+1 \mid k)', \ldots]$$
State trajectory $(\text{subsystem } i)$:
$$\boldsymbol{x}_i(k)' = [x_i(k+1 \mid k)', x_i(k+2 \mid k)', \ldots]$$
Input trajectory $(\text{subsystem } i)$:
$$\boldsymbol{u}_i(k)' = [u_i(k \mid k)', u_i(k+1 \mid k)', \ldots].$$

Let N denote the control horizon. Define $\mathcal{U}_i = \Omega_i \times \Omega_i \times \cdots \times \Omega_i \subset \mathbb{R}^{m_i N}$. The following notation is used to represent the finite horizon predicted state and input trajectory vectors in the different MPC frameworks

Centralized state trajectory:
$$\bar{\boldsymbol{x}}(k)' = [x(k+1 \mid k)', x(k+2 \mid k)', \ldots, x(k+N \mid k)']$$
Centralized input trajectory:
$$\bar{\boldsymbol{u}}(k)' = [u(k \mid k)', u(k+1 \mid k)', \ldots, u(k+N-1 \mid k)']$$
State trajectory $(\text{subsystem } i)$:
$$\bar{\boldsymbol{x}}_i(k)' = [x_i(k+1 \mid k)', x_i(k+2 \mid k)', \ldots, x_i(k+N \mid k)']$$
Input trajectory $(\text{subsystem } i)$:
$$\bar{\boldsymbol{u}}_i(k)' = [u_i(k \mid k)', u_i(k+1 \mid k)', \ldots, u_i(k+N-1 \mid k)'].$$

IV. MPC Frameworks for Systemwide Control

Let each set of admissible controls Ω_i be a nonempty, compact, convex set with $0 \in \text{int}(\Omega_i)$. The set of admissible controls for the whole plant Ω is defined to be the Cartesian product of the admissible control sets $\Omega_i, \forall\, i = 1, 2, \ldots, M$.

The *stage cost* at stage $t \geq k$ along the prediction horizon is defined as

$$L_i(x_i(t \mid k), u_i(t \mid k)) = \frac{1}{2}\left[x_i(t \mid k)' Q_i x_i(t \mid k) + u_i(t \mid k)' R_i u_i(t \mid k)\right] \quad (2)$$

in which $Q_i \geq 0, R_i > 0$ are symmetric weighting matrices and $(A_i, Q_i^{1/2})$ is detectable. The *cost function* $\phi_i(\cdot)$ for subsystem i is defined over an infinite horizon and is written as

$$\phi_i(\boldsymbol{x}_i, \boldsymbol{u}_i; x_i(k)) = \sum_{t=k}^{\infty} L_i(x_i(t \mid k), u_i(t \mid k)) \quad (3)$$

with $x_i(k \mid k) \equiv x_i(k)$. For any system, the constrained stabilizable set (also termed Null controllable domain) $\mathcal{X}$ is the set of all initial states $x \subseteq \mathbb{R}^n$ that can be steered to the origin by applying a sequence of admissible controls (see [32, Def. 2]). It is assumed throughout that the initial system state vector $x(k) \in \mathcal{X}$, in which $\mathcal{X}$ denotes the constrained stabilizable set for the overall system. A feasible solution to the corresponding optimization problem, therefore, exists. For notational simplicity, we drop the time dependence of the state and input trajectories in each MPC framework. For instance, in the centralized MPC framework, we write $\boldsymbol{x} \leftarrow \boldsymbol{x}(k)$ and $\boldsymbol{u} \leftarrow \boldsymbol{u}(k)$. In the distributed MPC framework, we use $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i(k)$ and $\boldsymbol{u}_i \leftarrow \boldsymbol{u}_i(k), \forall i = 1, 2, \ldots, M$.

Four MPC-based systemwide control frameworks are described in the following. In each MPC framework, the controller is defined by implementing the first input of the solution to the corresponding optimization problem.

Centralized MPC: In the centralized MPC framework, the MPC for the overall system solves the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{u}} \quad & \phi(\boldsymbol{x}, \boldsymbol{u}; x(k)) = \sum_i w_i \phi_i(\boldsymbol{x}_i, \boldsymbol{u}_i; x_i(k)) \\ \text{subject to} \quad & x(l+1 \mid k) = Ax(l \mid k) + Bu(l \mid k), k \leq l \\ & u_i(l \mid k) \in \Omega_i, k \leq l, \quad i = 1, 2, \ldots, M \end{aligned}$$

where $w_i > 0$, $\sum w_i = 1$.

For any system, centralized MPC achieves the best attainable performance (Pareto optimal) as the effect of interconnections among subsystems are accounted for exactly. Furthermore, any conflicts among controller objectives are resolved optimally.

Decentralized MPC: In the decentralized MPC framework, each subsystem-based MPC solves the following optimization problem:

$$\begin{aligned} & \min_{\boldsymbol{x}_i,\boldsymbol{u}_i} \phi_i(\boldsymbol{x}_i, \boldsymbol{u}_i; x_i(k)) \\ & \text{subject to} \quad x_i(l+1 \mid k) = A_{ii} x_i(l \mid k) + B_{ii} u_i(l \mid k),\ k \leq l \\ & \qquad\qquad\qquad u_i(l \mid k) \in \Omega_i, k \leq l. \end{aligned}$$

Each decentralized MPC solves an optimization problem to minimize its (local) cost function. The effects of the interconnected subsystems are assumed to be negligible and are ignored. In many situations, however, the previous assumption is not valid and leads to reduced control performance.

Distributed MPC: The partitioned model for each subsystem $i = 1, 2, \ldots, M$ is assumed to be available. Two formulations for distributed MPC, namely communication-based MPC and cooperation-based MPC, are considered. Communication-based strategies form the basis for the distributed MPC formulations in [8] and [20]. In the sequel, the suitability of pure communication-based MPC, as a candidate systemwide control formulation, is assessed. For both communication and cooperation-based MPC, several subsystem optimizations and exchanges of variables between subsystems are performed during a sample time. An optimization and exchange of variables is termed an *iterate*. We may choose not to iterate to convergence. The iteration number is denoted by p.

Communication-Based MPC: For communication-based MPC,[2] the optimal state-input trajectory $(\boldsymbol{x}_i^p, \boldsymbol{u}_i^p)$ for subsystem i, $i = 1, 2, \ldots, M$ at iterate p is obtained as the solution to the optimization problem

$$\begin{aligned}
&\min_{\boldsymbol{x}_i, \boldsymbol{u}_i} \phi_i(\boldsymbol{x}_i, \boldsymbol{u}_i; x_i(k)) \\
&\text{subject to} \quad x_i(l+1 \,|\, k) = A_{ii} x_i(l \,|\, k) + B_{ii} u_i(l \,|\, k) \\
&\qquad + \sum_{j \neq i} \left[A_{ij} x_j^{p-1}(l \,|\, k) + B_{ij} u_j^{p-1}(l \,|\, k) \right], \quad k \leq l \\
&\qquad u_i(l \,|\, k) \in \Omega_i, k \leq l.
\end{aligned}$$

Each communication-based MPC utilizes the objective function for that subsystem only. For each subsystem i at iteration p, only that subsystem input sequence $\boldsymbol{u}_i$ is optimized and updated. The other subsystems' inputs remain at $\boldsymbol{u}_j^{p-1}$, $\forall j = 1, 2, \ldots, M$, $j \neq i$. If the communication-based iterates converge, then at convergence, a Nash equilibrium (NE) is achieved. In this work, the term *communication-based MPC* alludes to the previous framework at convergence of the exchanged trajectories. Examples are presented in Section V-G for which communication-based MPC leads to either unacceptable closed-loop performance or closed-loop instability.

Feasible Cooperation-Based MPC (FC-MPC): To arrive at a reliable distributed MPC framework, we need to ensure that the subsystems' MPCs cooperate, rather than compete, with each other in achieving systemwide objectives. The local controller objective $\phi_i(\cdot)$ is replaced by an objective that measures the systemwide impact of local control actions. The simplest choice for such an objective is a strict convex combination of the controller objectives, i.e., $\phi(\cdot) = \sum_i w_i \phi_i(\cdot), w_i > 0, \sum_i w_i = 1$.

In large-scale implementations, the system sampling interval may be insufficient to allow convergence of an iterative, cooperation-based algorithm. In such cases, the cooperation-based algorithm has to be terminated prior to convergence of exchanged trajectories. The final calculated input trajectories are used to define a suitable distributed MPC control law. To enable intermediate termination, it is necessary that all iterates generated by the cooperation-based algorithm are strictly *systemwide feasible* (i.e., satisfy all model and inequality constraints) and the resulting nominal distributed control law is closed-loop stable. Such a distributed MPC algorithm is presented in Section V.

For notational convenience, we drop the k dependence of $\bar{\boldsymbol{x}}_i(k), \bar{\boldsymbol{u}}_i(k), i = 1, 2, \ldots, M$. It is shown in [34] that each $\bar{\boldsymbol{x}}_i$ can be expressed as

$$\bar{\boldsymbol{x}}_i = E_{ii} \bar{\boldsymbol{u}}_i + f_{ii} x_i(k) + \sum_{j \neq i} [E_{ij} \bar{\boldsymbol{u}}_j + f_{ij} x_j(k)]. \tag{6}$$

We consider the more practical case of open-loop stable systems first. A distributed MPC methodology capable of handling large, open-loop unstable systems is described in Section VI.

For open-loop stable systems, the FC-MPC optimization problem for subsystem i, denoted $\mathcal{F}_i$, is defined as

$$\mathcal{F}_i \triangleq \min_{\boldsymbol{u}_i} \sum_{r=1}^{M} w_r \Phi_r \left(\boldsymbol{u}_1^{p-1}, \ldots, \boldsymbol{u}_{i-1}^{p-1}, \boldsymbol{u}_i, \boldsymbol{u}_{i+1}^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}; x_r(k) \right) \tag{7a}$$

$$\text{subject to} \quad u_i(t \,|\, k) \in \Omega_i, \quad k \leq t \leq k+N-1 \tag{7b}$$

$$u_i(t \,|\, k) = 0, \quad k+N \leq t. \tag{7c}$$

The infinite horizon input trajectory $\boldsymbol{u}_i$ is obtained by augmenting $\bar{\boldsymbol{u}}_i$ with the input sequence $u_i(t \,|\, k) = 0, k+N \leq t$. The infinite horizon state trajectory $\boldsymbol{x}_i$ is derived from $\bar{\boldsymbol{x}}_i$ by propagating the terminal state $x_i(k+N \,|\, k)$ using (1) and $u_i(t \,|\, k) = 0, k+N \leq t, \forall i = 1, 2, \ldots, M$. The cost function $\Phi_i(\cdot)$ is obtained by eliminating the state trajectory $\boldsymbol{x}_i$ from (3) using (6) and the input, state parameterization described before. The solution to the optimization problem $\mathcal{F}_i$ is denoted by $\boldsymbol{u}_i^{*(p)}$. By definition

$$\boldsymbol{u}_i^{*(p)} = \left[u_i^{*(p)}(k \,|\, k)', u_i^{*(p)}(k+1 \,|\, k)', \ldots\ldots \right]'$$
$$\bar{\boldsymbol{u}}_i^{*(p)} = [u_i^{*(p)}(k \,|\, k)', u_i^{*(p)}(k+1|k)', \ldots, u_i^{*(p)}(k+N-1|k)']'.$$

V. Terminal Penalty FC-MPC

A. Optimization

For the quadratic form of $\phi_i(\cdot)$ given by (3), the FC-MPC optimization problem (7), for each subsystem $i = 1, 2, \ldots, M$, can be written as

$$\mathcal{F}_i \triangleq \min_{\bar{\boldsymbol{u}}_i} \frac{1}{2} \bar{\boldsymbol{u}}_i' \mathfrak{R}_i \bar{\boldsymbol{u}}_i + \left(\boldsymbol{r}_i(x(k)) + \sum_{j \neq i} \mathcal{H}_{ij} \bar{\boldsymbol{u}}_j^{p-1} \right)' \bar{\boldsymbol{u}}_i \tag{8a}$$

$$\text{subject to} \quad \bar{\boldsymbol{u}}_i \in \mathcal{U}_i \tag{8b}$$

in which

$$\begin{aligned}
\mathfrak{R}_i &= \mathbb{R}_i + \sum_{j=1}^{M} E_{ji}' \mathbb{Q}_j E_{ji} + \sum_{j=1}^{M} E_{ji}' \sum_{l \neq j} \mathbb{T}_{jl} E_{li} \\
\mathbb{Q}_i &= \operatorname{diag}(w_i Q_i(1), \ldots, w_i Q_i(N-1), P_{ii}) \\
\mathbb{T}_{ij} &= \operatorname{diag}(0, \ldots, 0, P_{ij}) \\
\mathbb{R}_i &= \operatorname{diag}(w_i R_i(0), w_i R_i(1), \ldots, w_i R_i(N-1)) \\
\boldsymbol{r}_i(x(k)) &= \sum_{j=1}^{M} E_{ji}' \mathbb{Q}_j \, \boldsymbol{g}_j(x(k)) + \sum_{j=1}^{M} E_{ji}' \sum_{l \neq j} \mathbb{T}_{jl} \boldsymbol{g}_l(x(k)) \\
\mathcal{H}_{ij} &= \sum_{l=1}^{M} E_{li}' \mathbb{Q}_l E_{lj} + \sum_{l=1}^{M} E_{li}' \sum_{s \neq l} \mathbb{T}_{ls} E_{sj} \\
\boldsymbol{g}_i(x(k)) &= \sum_{j=1}^{M} f_{ij} x_j(k)
\end{aligned}$$

and

$$P = \begin{bmatrix} P_{11} & P_{12} & \ldots & \ldots & P_{1M} \\ P_{21} & P_{22} & \ldots & \ldots & P_{2M} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ P_{M1} & P_{M2} & \ldots & \ldots & P_{MM} \end{bmatrix} \tag{9}$$

is a suitable terminal penalty matrix. Restricting attention (for now) to open-loop stable systems simplifies the choice of P. For each $i = 1, 2, \ldots, M$, let $Q_i(0) = Q_i(1) = \cdots = Q_i(N-1) = Q_i$. The terminal penalty P can be obtained as the solution to the centralized Lyapunov equation

$$A' P A - P = -\mathcal{Q} \tag{10}$$

[2]Similar strategies have been proposed by [8] and [20].

in which $\mathcal{Q} = \mathrm{diag}(w_1 Q_1, w_2 Q_2, \ldots, w_M Q_M)$. The centralized Lyapunov equation (10) is solved offline. The solution P to (10) has to be recomputed if the subsystems' models and/or cost functions are altered.

B. Algorithm and Properties

At time k, let $p_{\max}(k)$ represent the maximum number of permissible iterates for the sampling interval. The following algorithm is employed for cooperation-based distributed MPC.

Algorithm 1 (Terminal penalty FC-MPC)

Given $\bar{\boldsymbol{u}}_i^0(k), x_i(k), \mathbb{Q}_i, \mathbb{R}_i, i = 1, 2, \ldots, M$

$\quad p_{\max}(k) \geq 0$ and $\epsilon > 0$

$p \leftarrow 1, \rho_i \leftarrow \Gamma\epsilon, \Gamma \gg 1$

while $\rho_i > \epsilon$ for some $i = 1, 2, \ldots, M$ and $p \leq p_{\max}$

$\quad$ **do** $\forall i = 1, 2, \ldots, M$

$\quad\quad \bar{\boldsymbol{u}}_i^{*(p)} = \arg(\mathcal{F}_i)$, (see (8))

$\quad$ **end (do)**

for each $i = 1, 2, \ldots, M$

$\quad \bar{\boldsymbol{u}}_i^p \leftarrow w_i \bar{\boldsymbol{u}}_i^{*(p)} + (1 - w_i) \bar{\boldsymbol{u}}_i^{p-1}$

$\quad \rho_i \leftarrow \|\bar{\boldsymbol{u}}_i^p - \bar{\boldsymbol{u}}_i^{p-1}\|$

$\quad$ Transmit $\bar{\boldsymbol{u}}_i^p$ to each interconnected subsystem $j = 1, 2, \ldots, M, j \neq i$.

end (for)

$p \leftarrow p + 1$

end (while)

The state trajectory for subsystem i generated by the input trajectories $\bar{\boldsymbol{u}}_1, \bar{\boldsymbol{u}}_2, \ldots, \bar{\boldsymbol{u}}_M$ and initial state z is represented as $\bar{\boldsymbol{x}}_i(\bar{\boldsymbol{u}}_1, \bar{\boldsymbol{u}}_2, \ldots, \bar{\boldsymbol{u}}_M; z)$. At each iterate p in Algorithm 1, the state trajectory for subsystem $i = 1, 2, \ldots, M$ can be calculated as $\bar{\boldsymbol{x}}_i^p(\bar{\boldsymbol{u}}_1^p, \bar{\boldsymbol{u}}_2^p, \ldots, \bar{\boldsymbol{u}}_M^p; x(k))$. At each k, $p_{\max}(k)$ represents a design limit on the number of iterates; the user may choose to terminate Algorithm 1 prior to this limit.

The infinite horizon input and state trajectories $(\boldsymbol{x}_i^p, \boldsymbol{u}_i^p)$ can be obtained following the discussion in Section IV. Denote the cooperation-based cost function after p iterates by

$$\Phi\left(\boldsymbol{u}_1^p, \boldsymbol{u}_2^p, \ldots, \boldsymbol{u}_M^p; x(k)\right) = \sum_{r=1}^{M} w_r \Phi_r\left(\boldsymbol{u}_1^p, \boldsymbol{u}_2^p, \ldots, \boldsymbol{u}_M^p; x_r(k)\right).$$

The following properties can be established for the FC-MPC formulation (8) employing Algorithm 1.

Lemma 1: Given the distributed MPC formulation $\mathcal{F}_i$ defined in (7) and (8), $\forall i = 1, 2, \ldots, M$, the sequence of cost functions $\{\Phi(\boldsymbol{u}_1^p, \boldsymbol{u}_2^p, \ldots, \boldsymbol{u}_M^p; x(k))\}$ generated by Algorithm 1 is nonincreasing with iteration number p.

A proof is given in Appendix A.

Using Lemma 1 and the fact that $\Phi(\cdot)$ is bounded below assures convergence of the sequence of cost functions with iteration number.

Consider the centralized MPC optimization problem obtained by eliminating the subsystem states using the PM equations (1), $\forall i = 1, 2, \ldots, M$

$$\min_{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_M} \Phi(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_M; x(k)) = \sum_{i=1}^{M} w_i \Phi_i(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_M; x_i(k)) \tag{11a}$$

subject to

$$u_i(l \mid k) \in \Omega_i, \quad k \leq l \leq k + N - 1, \tag{11b}$$

$$u_i(l \mid k) = 0, \quad k + N \leq l \tag{11c}$$

$$\forall i = 1, 2, \ldots, M.$$

From the definition of $\phi_i(\cdot)$ given by (3), we have $\mathbb{R}_i > 0$. Hence, $\mathfrak{R}_i > 0, \forall i = 1, 2, \ldots, M$ in (8). It follows that $\Phi_i(\cdot)$ is strictly convex. Using convexity of $\Omega = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_M$ and strict convexity of $\Phi(\cdot)$, the solution $(\boldsymbol{u}_1^*, \ldots, \boldsymbol{u}_M^*)$ to the centralized MPC optimization problem (11) exists and is unique. By definition, $\boldsymbol{u}_i^* = [\bar{\boldsymbol{u}}_i^{*\prime}, 0, 0, \ldots]$.

Lemma 2: Consider $\Phi(\cdot)$ positive definite quadratic and let Ω_i, $\forall i = 1, 2, \ldots, M$ be convex, compact. Assume the solution to Algorithm 1 after p iterates is $(\boldsymbol{u}_1^p, \ldots, \boldsymbol{u}_M^p)$ with an associated cost function value $\Phi(\boldsymbol{u}_1^p, \ldots, \boldsymbol{u}_M^p; x(k))$, in which $\boldsymbol{u}_i^p = [\bar{\boldsymbol{u}}_i^{p\prime}, 0, 0, \ldots]'$. Denote the unique solution to (11) by $(\boldsymbol{u}_1^*, \boldsymbol{u}_2^*, \ldots, \boldsymbol{u}_M^*)$, in which $\boldsymbol{u}_i^* = [\bar{\boldsymbol{u}}_i^{*\prime}, 0, 0, \ldots]'$, and let $\Phi(\boldsymbol{u}_1^*, \boldsymbol{u}_2^*, \ldots, \boldsymbol{u}_M^*; x(k))$ represent the optimal cost function value. The solution obtained at convergence of Algorithm 1 satisfies

$$\lim_{p \to \infty} \Phi\left(\boldsymbol{u}_1^p, \boldsymbol{u}_2^p, \ldots, \boldsymbol{u}_M^p; x(k)\right) = \Phi(\boldsymbol{u}_1^*, \boldsymbol{u}_2^*, \ldots, \boldsymbol{u}_M^*; x(k))$$

$$\lim_{p \to \infty} (\boldsymbol{u}_1^p, \boldsymbol{u}_2^p, \ldots, \boldsymbol{u}_M^p) = (\boldsymbol{u}_1^*, \boldsymbol{u}_2^*, \ldots, \boldsymbol{u}_M^*).$$

A proof is given in Appendix A.

C. Distributed MPC Control Law

At time k, let the FC-MPC algorithm (Algorithm 1) be terminated after $p(k)$ iterates, with

$$\boldsymbol{u}_i^{p(k)}(k; x(k)) = \left[u_i^{p(k)}(k; x(k))', u_i^{p(k)}(k+1; x(k))', \ldots\right]', \quad i = 1, 2, \ldots, M$$

representing the solution to Algorithm 1 after $p(k)$ cooperation-based iterates. The distributed MPC control law is obtained through a receding horizon implementation of optimal control whereby the input applied to subsystem i is

$$u_i(k) = u_i^{p(k)}(k; x(k)). \tag{12}$$

D. Feasibility of FC-MPC Optimizations

Since $x(0) \in \mathcal{X}$, there exists a set of feasible, open-loop input trajectories $(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_M)$, such that $x_i(k) \to 0$, $\forall i = 1, 2, \ldots, M$ and k sufficiently large. Convexity of Ω_i, $\forall i = 1, 2, \ldots, M$ and Algorithm 1 guarantee that given a feasible input sequence at time $k = 0$, a feasible input sequence exists for all future times. One trivial choice for a feasible input sequence at $k = 0$ is $u_i(k+l\,|\,k) = 0, l \geq 0$, $\forall i = 1, 2, \ldots, M$. This choice follows from our assumption that each Ω_i is nonempty and $0 \in \mathrm{int}(\Omega_i)$. Existence of a feasible input sequence for each subsystem i at $k = 0$ ensures that the FC-MPC optimization problem (7), (8) has a solution for each $i = 1, 2, \ldots, M$ and all $k \geq 0$.

E. Initialization

At discrete time $k+1$, define $\forall i = 1, 2, \ldots, M$

$$\boldsymbol{u}_i^0(k+1)' = \Big[u_i^{p(k)}(k+1; x(k))', u_i^{p(k)}(k+2; x(k))', \ldots, u_i^{p(k)}(k+N-1; x(k))', 0, 0, \ldots\Big]. \quad (13)$$

It follows that $\boldsymbol{u}_1^0(k+1), \boldsymbol{u}_2^0(k+1), \ldots, \boldsymbol{u}_M^0(k+1)$ constitute feasible subsystem input trajectories with an associated cost function $\Phi(\boldsymbol{u}_1^0(k+1), \boldsymbol{u}_2^0(k+1), \ldots, \boldsymbol{u}_M^0(k+1); x(k+1))$.

F. Nominal Closed-Loop Stability

Given the set of initial subsystem states $x_i(0)$, $\forall i = 1, 2, \ldots, M$. Define $\tilde{J}_N(x(0))$ to be the value of the cooperation-based cost function with the set of zero input trajectories $u_i(k+j\,|\,k) = 0$, $j \geq 0$, $\forall i = 1, 2, \ldots, M$. At time k, let $J_N^0(x(k))$ represent the value of the cooperation-based cost function with the input trajectory initialization described in (13). For notational convenience we drop the function dependence of the generated state trajectories and write $\boldsymbol{x}_i \equiv \boldsymbol{x}_i(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_M; z)$, $\forall i = 1, 2, \ldots, M$. The value of the cooperation-based cost function after $p(k)$ iterates is denoted by $J_N^{p(k)}(x(k))$. Thus

$$J_N^{p(k)}(x(k)) = \sum_{i=1}^{M} w_i \phi_i\left(\boldsymbol{x}_i^{p(k)}, \boldsymbol{u}_i^{p(k)}; x(k)\right) \quad (14a)$$

$$= \sum_{i=1}^{M} w_i \sum_{j=0}^{\infty} L_i\left(x_i^{p(k)}(k+j|k), u_i^{p(k)}(k+j|k)\right). \quad (14b)$$

At $k = 0$, we have, using Lemma 1, that $J_N^{p(0)}(x(0)) \leq J_N^0(x(0)) = \tilde{J}_N(x(0))$. It follows from (13) and Lemma 1 that

$$0 \leq J_N^{p(k)}(x(k)) \leq J_N^0(x(k)) = J_N^{p(k-1)}(x(k-1)) - \sum_{i=1}^{M} w_i L_i\left(x_i(k-1), u_i^{p(k-1)}(k-1)\right), \quad \forall k > 0. \quad (15)$$

Using the previous relationship recursively from time k to time 0 gives

$$J_N^{p(k)}(x(k)) \leq \tilde{J}_N(x(0)) - \sum_{j=0}^{k-1}\sum_{i=1}^{M} w_i L_i\left(x_i(j), u_i^{p(j)}(j)\right) \leq \tilde{J}_N(x(0)). \quad (16)$$

From (14), we have $\frac{1}{2}\lambda_{\min}(\mathcal{Q})\|x(k)\|^2 \leq J_N^{p(k)}(x(k))$. Using (16), gives $J_N^{p(k)}(x(k)) \leq \tilde{J}_N(x(0)) = \frac{1}{2}x(0)'Px(0) \leq \frac{1}{2}\lambda_{\max}(P)\|x(0)\|^2$. From the previous two cost relationships, we obtain $\|x(k)\| \leq \sqrt{(\lambda_{\max}(P))/(\lambda_{\min}(\mathcal{Q}))}\|x(0)\|$, which shows that the closed-loop system is Lyapunov stable [36, p. 265]. In fact, using the cost convergence relationship (15) the closed-loop system is also attractive, which proves asymptotic stability under the distributed MPC control law.

Lemmas 1 and 2 can be used to establish the following (stronger) exponential closed-loop stability result.

Theorem 1: Given Algorithm 1 using the distributed MPC optimization problem (8) with $N \geq 1$. In Algorithm 1, let $0 < p_{\max}(k) \leq p^* < \infty, \forall k \geq 0$. If A is stable, P is obtained from (10), and

$$Q_i(0) = Q_i(1) = \cdots = Q_i(N-1) = Q_i > 0$$
$$R_i(0) = R_i(1) = \cdots = R_i(N-1) = R_i > 0$$
$$\forall i = 1, 2, \ldots, M$$

then the origin is an exponentially stable equilibrium for the closed-loop system

$$x(k+1) = Ax(k) + Bu(k)$$

in which

$$u(k) = \left[u_1^{p(k)}(k; x(k))', \ldots, u_M^{p(k)}(k; x(k))'\right]'$$

for all $x(k) \in \mathbb{R}^n$ and any $p(k) = 1, 2, \ldots, p_{\max}(k)$.

A proof is given in Appendix A.

Remark 1: If $(A, \mathcal{Q}^{1/2})$ is detectable, then the weaker requirement $Q_i \geq 0$, $R_i > 0$, $\forall i = 1, \ldots, M$ is sufficient to ensure exponential stability of the closed-loop system under the distributed MPC control law.

G. Examples

Power System Terminology and Control Area Model: For the purposes of AGC, power systems are decomposed into control areas, with tie-lines providing interconnections between areas [37]. Each area typically consists of numerous generators and loads. It is common, though, for all generators in an area to be lumped as a single equivalent generator, and likewise for loads. Furthermore, because AGC operation is limited to relatively small system disturbances, use of linearized models is standard [37]. Those modeling simplifications are adopted in all subsequent examples. Some basic power systems terminology is provided in Table I. The notation Δ is used to indicate a deviation from steady state. For example, $\Delta\omega$ represents a deviation in the angular frequency from its nominal operating value (60 Hz).

Consider any control area $i = 1, 2, \ldots, M$, interconnected to control area $j, j \neq i$ through a tie line. A simplified model for such a control area i is given by

Area $\boldsymbol{i}$

$$\frac{d\Delta\omega_i}{dt} + \frac{1}{M_i^a}D_i\Delta\omega_i + \frac{1}{M_i^a}\Delta P_{\mathrm{tie}}^{ij} - \frac{1}{M_i^a}\Delta P_{\mathrm{mech}_i} = -\frac{1}{M_i^a}\Delta P_{\mathrm{L}_i} \quad (17a)$$

$$\frac{d\Delta P_{\mathrm{mech}_i}}{dt} + \frac{1}{T_{\mathrm{CH}_i}}\Delta P_{\mathrm{mech}_i} - \frac{1}{T_{\mathrm{CH}_i}}\Delta P_{\mathrm{v}_i} = 0 \quad (17b)$$

$$\frac{d\Delta P_{\mathrm{v}_i}}{dt} + \frac{1}{T_{\mathrm{G}_i}}\Delta P_{\mathrm{v}_i} - \frac{1}{T_{\mathrm{G}_i}}\Delta P_{\mathrm{ref}_i} + \frac{1}{R_i^f T_{\mathrm{G}_i}}\Delta\omega_i = 0. \quad (17c)$$

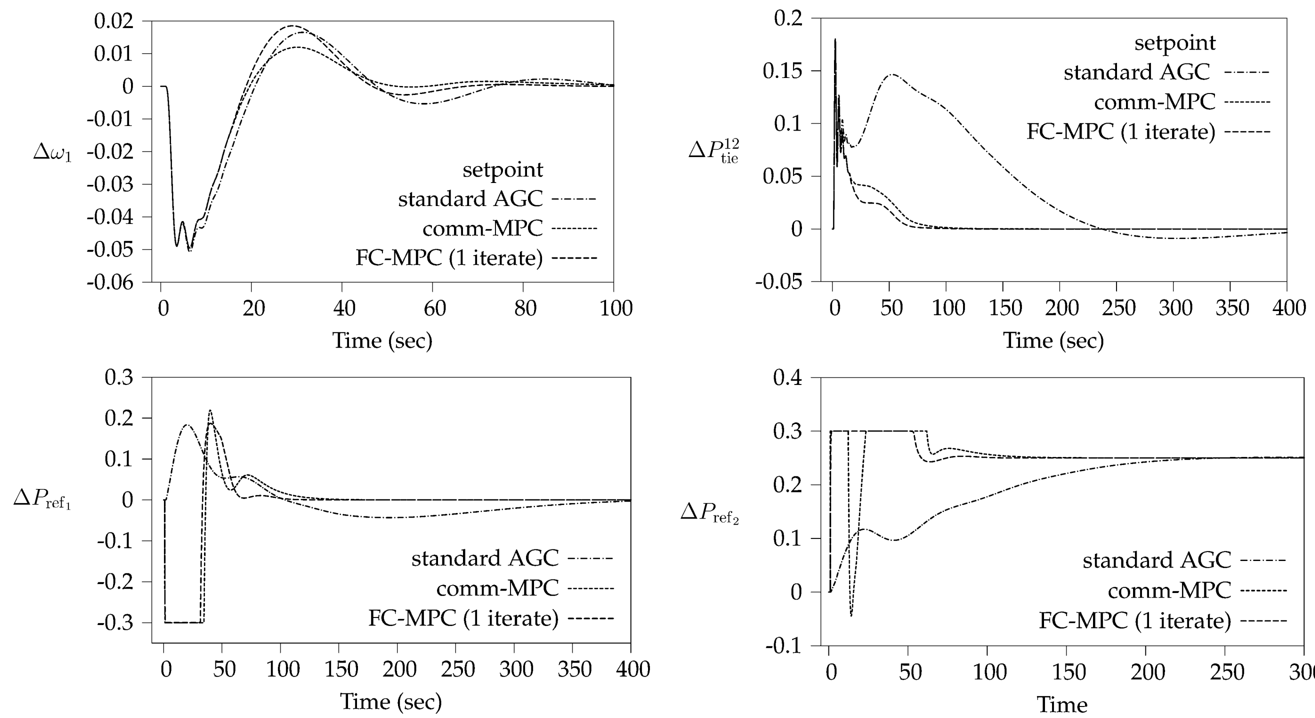


Fig. 1. Performance of different control frameworks rejecting a load disturbance in area 2. Change in frequency $\Delta\omega_1$, tie-line power flow $\Delta P_{\mathrm{tie}}^{12}$, and load reference setpoints $\Delta P_{\mathrm{ref}_1}, \Delta P_{\mathrm{ref}_2}$.

TABLE I
BASIC POWER SYSTEMS TERMINOLOGY

ω	:	angular frequency of rotating mass
δ	:	phase angle of rotating mass
M^a	:	Angular momentum
D	:	$\frac{\text{percent change in load}}{\text{percent change in frequency}}$
P_{mech}	:	mechanical power
P_{L}	:	nonfrequency sensitive load
T_{CH}	:	charging time constant (prime mover)
P_{v}	:	steam valve position
P_{ref}	:	load reference setpoint
R^f	:	$\frac{\text{percent change in frequency}}{\text{percent change in unit output}}$
T_{G}	:	governor time constant
P_{tie}^{ij}	:	tie-line power flow between areas i and j
T_{ij}	:	tie-line (between areas i and j) stiffness coefficient
K_{ij}	:	FACTS device coefficient
	:	(regulating impedance between areas i and j)

Tie-line power flow between areas i and j

$$\frac{d\Delta P_{\mathrm{tie}}^{ij}}{dt} = T_{ij}(\Delta\omega_i - \Delta\omega_j) \tag{17d}$$

$$\Delta P_{\mathrm{tie}}^{ji} = -\Delta P_{\mathrm{tie}}^{ij}. \tag{17e}$$

Performance comparison. The cumulative stage cost Λ is used as an index for comparing the performance of different MPC frameworks. Define

$$\Lambda = \frac{1}{t}\sum_{k=0}^{t-1}\sum_{i=1}^{M} L_i(x_i(k), u_i(k)) \tag{18}$$

where t is the simulation horizon. For each example presented in this paper, the model and controller parameters are omitted for brevity; they are available in [33].

1) Two-Area Power System Network: An example with two control areas interconnected through a tie line is considered initially. A control horizon $N = 15$ is used for each MPC. The controlled variable (CV) for area 1 is the frequency deviation $\Delta\omega_1$ and the CV for area 2 is the deviation in the tie-line power flow between the two control areas $\Delta P_{\mathrm{tie}}^{12}$. From the control area model (17), if $\Delta\omega_1 \to 0$ and $\Delta P_{\mathrm{tie}}^{12} \to 0$ then $\Delta\omega_2 \to 0$.

For a 25% load increase in area 2, the load disturbance rejection performance of the FC-MPC formulation is evaluated and compared against the performance of centralized MPC (cent-MPC), communication-based MPC (comm-MPC), and standard AGC with anti-reset windup. The load reference setpoint in each area is constrained between ± 0.3. In practice, a large load change, such as the one considered above, would result in curtailment of AGC and initiation of emergency control measures such as load shedding. The purpose of this exaggerated load disturbance is to illustrate the influence of input constraints on the different control frameworks.

The relative performance of standard AGC, cent-MPC, and FC-MPC (terminated after one iterate) rejecting the load disturbance in area 2 is depicted in Fig. 1. The closed-loop trajectory of the FC-MPC controller, obtained by terminating Algorithm 1 after one iterate, is almost indistinguishable from the closed-loop trajectory of cent-MPC. Standard AGC performs nearly as well as cent-MPC and FC-MPC in driving the local frequency changes to zero. Under standard AGC, however, the system takes in excess of 400 s to drive the deviational tie-line power flow to zero. With the cent-MPC or the FC-MPC framework, the tie-line power flow disturbance is rejected in about 100 s. A closed-loop performance comparison of the different

TABLE II
PERFORMANCE OF DIFFERENT CONTROL FORMULATIONS W.R.T. CENT-MPC, $\Delta\Lambda\% = (\Lambda_{\text{config}} - \Lambda_{\text{cent}})/(\Lambda_{\text{cent}}) \times 100$

	Λ	$\Delta\Lambda\%$
standard AGC	39.26	189
comm-MPC	15.82	18.26
FC-MPC (1 iterate)	13.42	0.26
FC-MPC (5 iterates)	~ 13.38	~ 0
cent-MPC	13.38	0

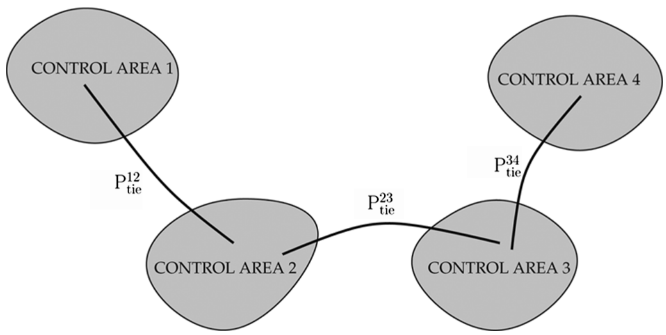


Fig. 2. Four-area power system.

control frameworks is given in Table II. The comm-MPC framework stabilizes the system but incurs a control cost that is nearly 18% greater than that incurred by FC-MPC (one iterate). If five iterates per sampling interval are allowed, the performance of FC-MPC is almost identical to that of cent-MPC.

Notice from Fig. 1 that the initial response of AGC is to increase generation in both areas. This causes a large deviation in the tie-line power flow. On the other hand, under comm-MPC and FC-MPC, MPC_1 initially reduces area 1 generation and MPC_2 orders a large increase in area 2 generation (the area where the load disturbance occurred). This strategy enables a much more rapid restoration of tie-line power flow.

2) Four-Area Power System Network: Consider the four-area power system shown in Fig. 2. The model for each control area follows from (17). In each control area, a change in local power demand (load) alters the nominal operating frequency. The MPC in each control area i manipulates the load reference setpoint P_{ref_i} to drive the frequency deviations $\Delta\omega_i$ and tie-line power flow deviations $\Delta P^{ij}_{\text{tie}}$ to zero. Power flow through the tie lines gives rise to interactions among the control areas. Hence, a load change in area 1, for instance, causes a transient frequency change in all control areas.

The relative performance of cent-MPC, comm-MPC, and FC-MPC is analyzed for a 25% load increase in area 2 and a simultaneous 25% load drop in area 3. This load disturbance occurs at 5 s. For each MPC, we choose a control horizon of $N = 20$. In the comm-MPC and FC-MPC formulations, the load reference setpoint ΔP_{ref_i} in each area is manipulated to reject the load disturbance and drive the change in local frequencies $\Delta\omega_i$ and tie-line power flows $\Delta P^{ij}_{\text{tie}}$ to zero. In the cent-MPC framework, a single MPC manipulates all four ΔP_{ref_i}. The load reference setpoint for each area is constrained between ± 0.5.

The performances of cent-MPC, comm-MPC, and FC-MPC (one iterate) are shown in Fig. 3. Only $\Delta\omega_2$ and $\Delta P^{23}_{\text{tie}}$ are shown as the frequency and tie-line power flow deviations in the other areas display similar qualitative behavior. Likewise, only ΔP_{ref_2} and ΔP_{ref_3} are shown as other load reference setpoints behave similarly. The control costs are given in Table III. Under comm-MPC, the load reference setpoints for areas 2 and 3 switch repeatedly between their upper and lower saturation limits. Consequently, the power system network is unstable under comm-MPC. The closed-loop performance of the FC-MPC formulation, terminated after just one iterate, is within 26% of cent-MPC performance. If the FC-MPC algorithm is terminated after five iterates, the performance of FC-MPC is within 4% of cent-MPC performance. By allowing the cooperation-based iterative process to converge, the closed-loop performance of FC-MPC can be driven to within any prespecified tolerance of cent-MPC performance.

TABLE III
PERFORMANCE OF DIFFERENT MPC FRAMEWORKS RELATIVE TO CENT-MPC, $\Delta\Lambda\% = (\Lambda_{\text{config}} - \Lambda_{\text{cent}})/(\Lambda_{\text{cent}}) \times 100$

	$\Lambda \times 10^{-2}$	$\Delta\Lambda\%$
cent-MPC	7.6	0
comm-MPC	∞	∞
FC-MPC (1 iterate)	9.6	26
FC-MPC (5 iterates)	7.87	3.7

3) Two-Area Power System With FACTS Device: In this example, we revisit the two area network considered in Section V-G1. In this case though, a FACTS device is employed by area 1 to manipulate the effective impedance of the tie line and control power flow between the two interconnected control areas. The control area models follow from (17). In order to incorporate the FACTS device, though, (17a) in area 1 is replaced by

$$\begin{aligned}\frac{d\Delta\delta_{12}}{dt} &= (\Delta\omega_1 - \Delta\omega_2)\\ \frac{d\Delta\omega_1}{dt} &= -\frac{1}{M_1^a}D_1\Delta\omega_1 - \frac{1}{M_1^a}T_{12}\Delta\delta_{12} + \frac{1}{M_1^a}K_{12}\Delta X_{12}\\ &\quad + \frac{1}{M_1^a}\Delta P_{\text{mech}_1} - \frac{1}{M_1^a}\Delta P_{\text{L}_1}\end{aligned}$$

and in area 2 by

$$\begin{aligned}\frac{d\Delta\omega_2}{dt} &= -\frac{1}{M_2^a}D_2\Delta\omega_2 + \frac{1}{M_2^a}T_{12}\Delta\delta_{12} - \frac{1}{M_2^a}K_{12}\Delta X_{12}\\ &\quad + \frac{1}{M_2^a}\Delta P_{\text{mech}_2} - \frac{1}{M_2^a}\Delta P_{\text{L}_2}\end{aligned}$$

where ΔX_{12} is the impedence deviation induced by the FACTS device. The tie-line power flow deviation becomes

$$\Delta P^{12}_{\text{tie}} = -\Delta P^{21}_{\text{tie}} = T_{12}\Delta\delta_{12} - K_{12}\Delta X_{12}.$$

Notice that if $\Delta X_{12} = 0$, the model reverts to (17). The MPC for area 1 manipulates ΔP_{ref_1} and ΔX_{12} to drive $\Delta\omega_1$ and the relative phase difference $\Delta\delta_{12} = \Delta\delta_1 - \Delta\delta_2$ to zero. The MPC for area 2 manipulates ΔP_{ref_2} to drive $\Delta\omega_2$ to zero.

The relative performance of cent-MPC, comm-MPC, and FC-MPC rejecting a simultaneous 25% increase in the load of areas 1 and 2 is investigated. The closed-loop performance of the different MPC frameworks is shown in Fig. 4. The

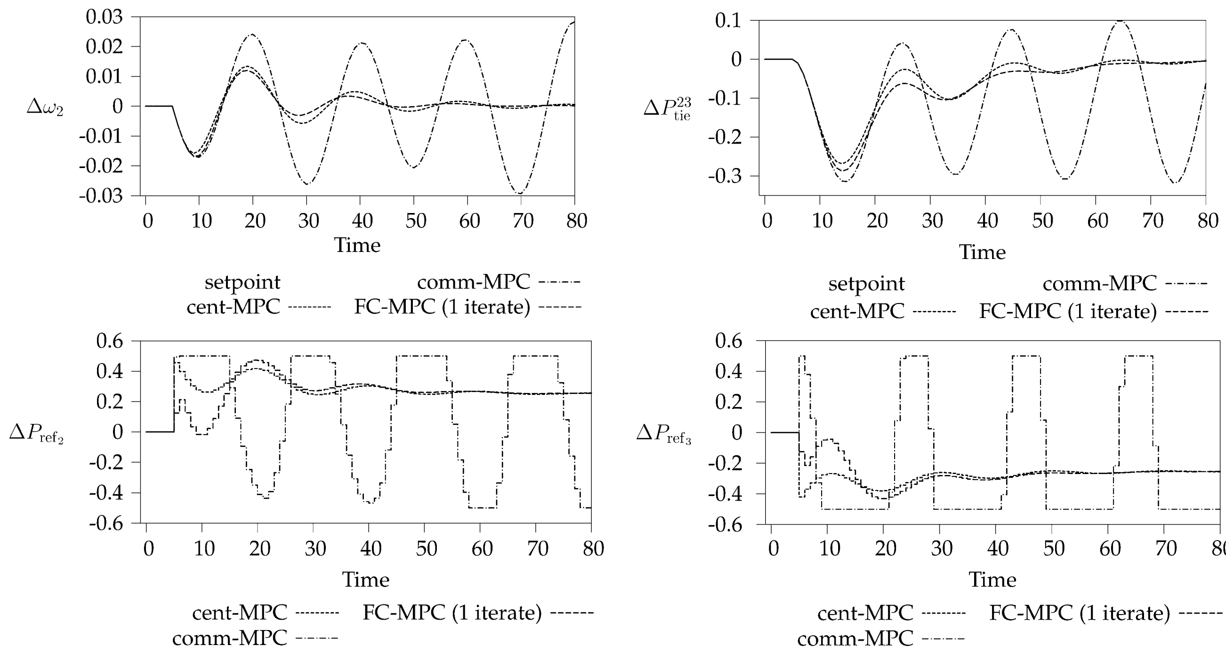


Fig. 3. Performance of different control frameworks rejecting a load disturbance in areas 2 and 3. Change in frequency $\Delta\omega_2$, tie-line power flow $\Delta P_{\text{tie}}^{23}$, and load reference setpoints $\Delta P_{\text{ref}_2}, \Delta P_{\text{ref}_3}$.

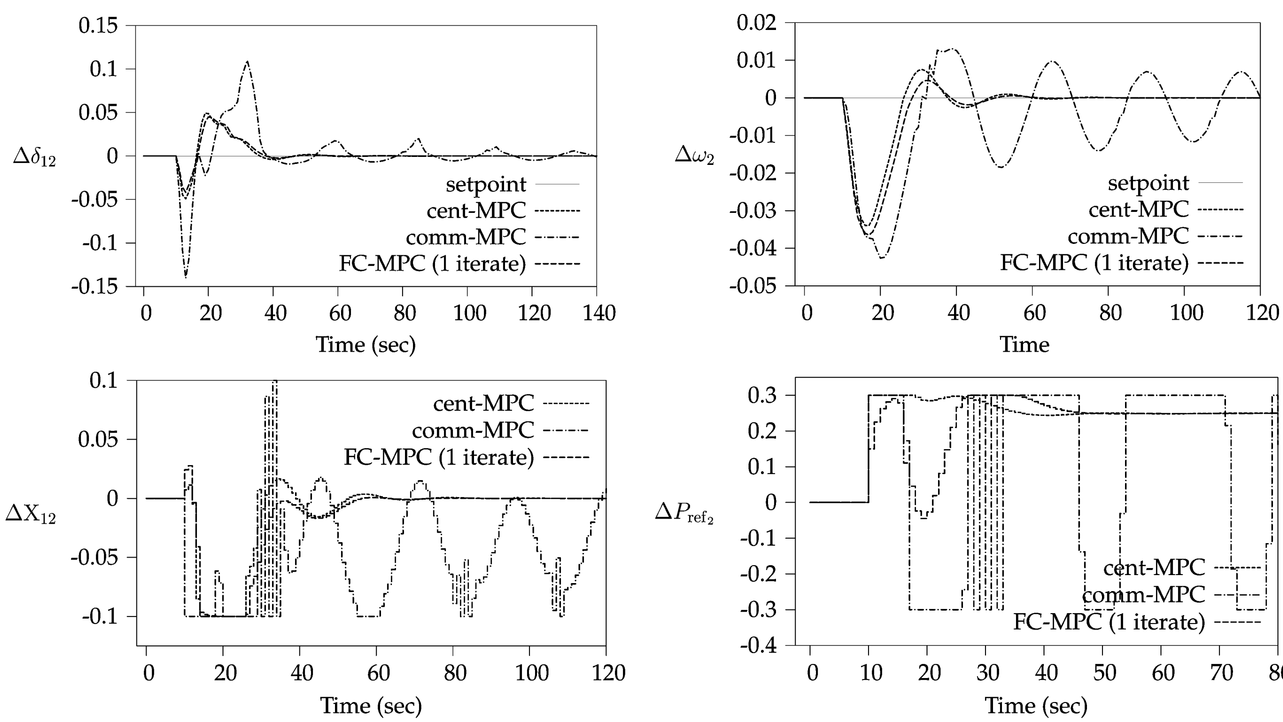


Fig. 4. Performance of different control frameworks rejecting a load disturbance in area 2. Change in relative phase difference $\Delta\delta_{12}$, frequency $\Delta\omega_2$, tie-line impedence ΔX_{12} due to the FACTS device and load reference setpoint ΔP_{ref_2}.

associated control costs are given in Table IV. The performance of FC-MPC (one iterate) is within 28% of cent-MPC performance. The performance of comm-MPC, on the other hand, is highly oscillatory and significantly worse than that of FC-MPC (one iterate). While comm-MPC is stabilizing, the system takes nearly 400 s to reject the load disturbance. With FC-MPC (one iterate), the load disturbance is rejected in less than 80 s. If five iterates per sampling interval are possible, the FC-MPC framework achieves performance that is within 2.5% of cent-MPC performance.

VI. Terminal Control FC-MPC

The terminal penalty-based FC-MPC framework considered earlier utilizes a suboptimal parameterization of the postulated input trajectories. Accordingly, performance is infinite horizon optimal only in the limit as $N \rightarrow \infty$. Otherwise, convergence

TABLE IV
PERFORMANCE OF DIFFERENT MPC FRAMEWORKS RELATIVE TO CENT-MPC, $\Delta\Lambda\% = (\Lambda_{\text{config}} - \Lambda_{\text{cent}})/(\Lambda_{\text{cent}}) \times 100$

	$\Lambda \times 10^{-2}$	$\Delta\Lambda\%$
cent-MPC	3.06	0
comm-MPC	9.53	211
FC-MPC (1 iterate)	3.92	28
FC-MPC (5 iterates)	3.13	2.3

achieves performance that is within a prespecified tolerance of a modified infinite horizon optimal control problem (11). The motivation behind terminal control-based FC-MPC is to achieve infinite horizon optimal (centralized, constrained, LQR [30]) performance at convergence using finite values of N.

For terminal control FC-MPC, the unconstrained centralized feedback law is employed as the terminal feedback law. The idea is to force the collection of subsystem-based MPCs to drive the system state to a neighborhood of the origin in which the unconstrained centralized feedback law is feasible. From [15], we know that such a neighborhood of the origin is well defined and can be computed offline. Following the description in [15], we use $\mathcal{O}_\infty(A,C)$ to denote the maximal output admissible set for the overall system (A,B,C). Since Ω_i, $\forall\, i=1,2,\ldots,M$ and Ω are convex, we have from [15, Th. 2.1] that $\mathcal{O}_\infty$ is convex. We assume that each Ω_i is a polytope, i.e., $\Omega_i = \{u_i \,|\, D_i u_i \le d_i, d_i > 0\}$. The determination of $\mathcal{O}_\infty$, in this case, involves the solution to a set of linear programs. Because (A,C) is detectable only (and not observable), $\mathcal{O}_\infty(A,C)$ is a cylinder with infinite extent along directions in the unobservable subspace.

Let K denote the optimal, centralized linear quadratic regulator (LQR) gain and let Π denote the solution to the corresponding centralized discrete steady-state Riccati equation, i.e.,

$$\Pi = \mathcal{Q} + A'\Pi A - A'\Pi B(\mathcal{R} + B'\Pi B)^{-1}B'\Pi A \quad \text{(19a)}$$
$$K = -(\mathcal{R} + B'\Pi B)^{-1}B'\Pi A \quad \text{(19b)}$$

in which $\mathcal{Q} = \text{diag}(w_1Q_1, w_2Q_2, \ldots, w_MQ_M)$ and $\mathcal{R} = \text{diag}(w_1R_1, w_2R_2, \ldots, w_MR_M)$. Conditions for existence of a solution to (19) are well known [4], [9]. Using a subsystem-wise partitioning for K and Π gives

$$K = \begin{bmatrix} K_{11} & K_{12} & \ldots & K_{1M} \\ K_{21} & K_{22} & \ldots & K_{2M} \\ \vdots & \ddots & \ddots & \vdots \\ K_{M1} & K_{M2} & \ldots & K_{MM} \end{bmatrix}$$
$$\Pi = \begin{bmatrix} \Pi_{11} & \Pi_{12} & \ldots & \Pi_{1M} \\ \Pi_{21} & \Pi_{22} & \ldots & \Pi_{2M} \\ \vdots & \ddots & \ddots & \vdots \\ \Pi_{M1} & \Pi_{M2} & \ldots & \Pi_{MM} \end{bmatrix}.$$

The terminal control law for subsystem $i=1,2,\ldots,M$ at time k is, therefore, $u_i(t\,|\,k) = K_{ii}x_i(t\,|\,k) + \sum_{j\neq i}^{M} K_{ij}x_j(t\,|\,k)$, $k+N \le t$. To arrive at the terminal control FC-MPC optimization problem, we use existing definitions in (8) and redefine

$$\mathbb{Q}_i = \text{diag}\left(w_iQ_i(1), \ldots, w_iQ_i(N-1), \Pi_{ii}\right)$$
$$\mathbb{T}_{ij} = \text{diag}\left(0, \ldots, 0, \Pi_{ij}\right).$$

The terminal control FC-MPC optimization problem is then given by (8), with these modifications. Algorithm 1 is again utilized for terminal control FC-MPC.

Initialization: To initialize Algorithm 1 for terminal control FC-MPC, it is necessary to calculate a set of subsystem input trajectories that steers the terminal system state (i.e., the predicted state at the end of the control horizon of each subsystem-based MPC) inside $\mathcal{O}_\infty(A,C)$. For the initial system state $x(0) \notin \mathcal{O}_\infty(A,C)$, such a set of subsystem input trajectories can be computed by solving a simple quadratic program (QP). One formulation for this initialization QP is described as follows:

$$\mathcal{L}^{\text{N}}(x(k)) = \arg\min_{\tilde{\boldsymbol{u}}(k)} \|\tilde{\boldsymbol{u}}(k)\|^2 \quad \text{(20a)}$$
$$\text{subject to} \quad \mathcal{T}'(\mathbb{E}\tilde{\boldsymbol{u}}(k) + \boldsymbol{g}(x(k))) \in \mathcal{O}_\infty(A,C) \quad \text{(20b)}$$
$$\tilde{\boldsymbol{u}}_i(k) \in \mathcal{U}_i,\ i=1,\ldots,M \quad \text{(20c)}$$

in which

$$\mathbb{E} = \begin{bmatrix} E_{11} & E_{12} & \ldots & E_{1M} \\ E_{21} & E_{22} & \ldots & E_{2M} \\ \vdots & \ddots & \ddots & \vdots \\ E_{1M} & E_{2M} & \ldots & E_{MM} \end{bmatrix}$$
$$\mathcal{T}_i = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ I \end{bmatrix} \quad \tilde{\boldsymbol{u}}(k) = \begin{bmatrix} \bar{\boldsymbol{u}}_1(k) \\ \bar{\boldsymbol{u}}_2(k) \\ \vdots \\ \bar{\boldsymbol{u}}_M(k) \end{bmatrix} \quad \boldsymbol{g}(x) = \begin{bmatrix} \boldsymbol{g}_1(x) \\ \boldsymbol{g}_2(x) \\ \vdots \\ \boldsymbol{g}_M(x) \end{bmatrix}.$$

$\mathcal{T} = \text{diag}(\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_M)$, with E_{ij} defined in (6), and $\boldsymbol{g}_i(x(k))$ defined in (8). The definition of $\mathcal{T}_i$ is such that $x_i(k+N\,|\,k) = \mathcal{T}_i'\bar{\boldsymbol{x}}_i(k)$. The QP (20) is a centralized calculation; distributed versions for this initialization QP can be derived using techniques similar to those presented here, but are not pursued in this paper.

Define the steerable set

$$\mathbb{S}_k = \{x(k) \,|\, \exists\, \bar{\boldsymbol{u}}_i(k) \in \mathcal{U}_i, i=1,\ldots,M, \text{ such that } x(k+N\,|\,k) = \mathcal{T}'\bar{\boldsymbol{x}}(k) \in \mathcal{O}_\infty(A,C)\}.$$

The set $\mathbb{S}_k$ denotes the set of all $x(k)$ for which the initialization QP (20) is feasible for a given N. We have $\mathbb{S}_k \subseteq \mathcal{X}$. Constrained stabilizability, therefore, follows.

At each iterate of the terminal control FC-MPC algorithm, the validity of the terminal set constraint $x^p(k+N\,|\,k) \in \mathcal{O}_\infty(A,C)$ must be verified. Two approaches are available for ensuring the validity of the postulated terminal control law without explicitly enforcing a terminal set constraint. In the first approach, the value of N is altered online to ensure validity of the terminal set constraint. At each iterate, a subsystem-based procedure is used to verify the validity of the postulated terminal control law. If the selected control horizon is not sufficient to ensure feasibility of the terminal control law, N is increased and the subsystems' terminal control FC-MPC optimizations are resolved using the new value of N. Strategies for increasing N online to enable efficient implementation have been investigated in [30] for single MPCs.

Rather than increase N online, a second approach may be adopted. The idea in this case is to restrict the set of permissible initial states to a positively invariant set in which the terminal set constraint is feasible for each subsystem $i=1,\ldots,M$. This positively invariant set depends on the choice of N. For a given $N>0$, we first construct the steerable set $\mathbb{S}$. Next, we determine the set of all possible combinations of system states

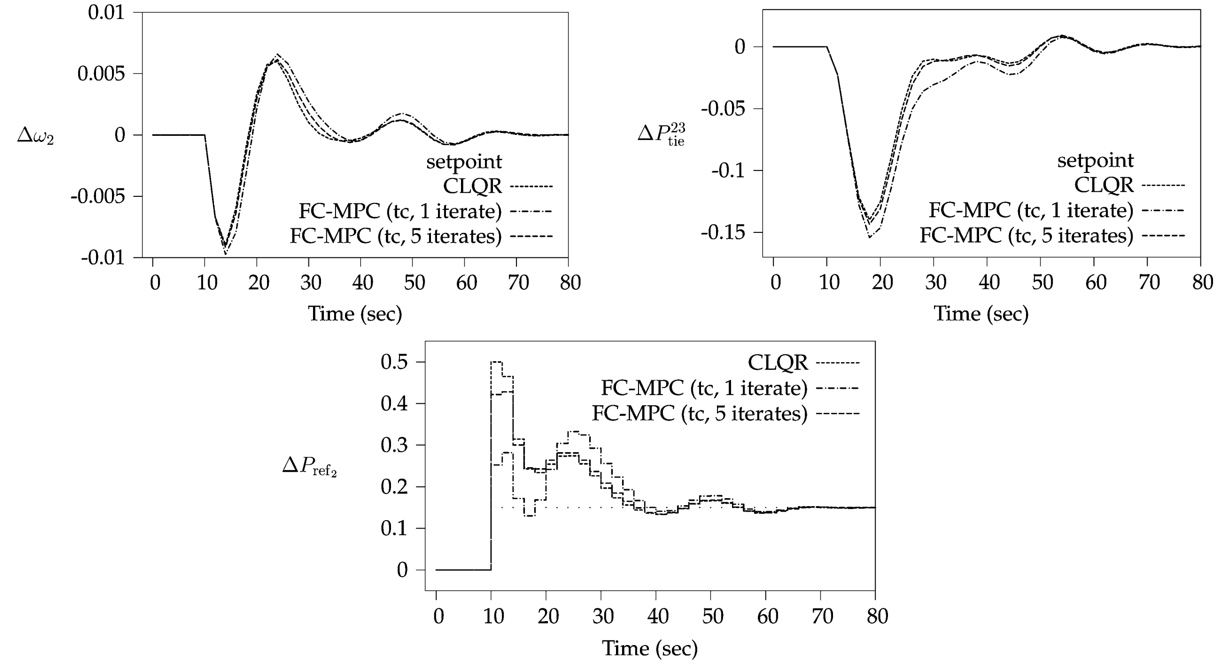


Fig. 5. Performance of FC-MPC (tc) and CLQR, rejecting a load disturbance in areas 2 and 3. Change in local frequency $\Delta\omega_2$, tie-line power flow $\Delta P_{\text{tie}}^{23}$, and load reference setpoint ΔP_{ref_2}.

and assumed subsystem input trajectories for which the solution to the terminal control FC-MPC optimization problem for each subsystem satisfies the terminal set constraint. Finally, the domain of the controller, which is the largest positively invariant set for which the terminal control FC-MPC control law is stabilizing, is constructed. To construct this invariant set, one may employ standard techniques available in the literature for backward construction of polytopic invariant sets under constraints [6], [16], [22]. Space restrictions preclude further development of either approach in this paper; details of both are available in [33].

For the nominal case, the set of shifted input trajectories (13), obtained using the solution to Algorithm 1 for terminal control FC-MPC at time k, is a feasible set of input trajectories at time $k+1$. For this case, therefore, the initialization QP (20) has to be solved only once at $k=0$. Lemmas 1 and 2 established for terminal penalty FC-MPC (see Section V) are also valid for terminal control FC-MPC. At convergence of the exchanged input trajectories, the performance of terminal control FC-MPC is within a prespecified tolerance of the centralized, constrained LQR [30], [32] performance. If (A, B) is stabilizable, (A, C) and $(A, \mathcal{Q}^{1/2})$ are detectable, and $Q_i \geq 0, R_i > 0, \forall i = 1, \ldots, M$, the terminal control FC-MPC control law is nominally asymptotically stable for all values of the iteration number $p(k) > 0$.

Unstable Four-Area Power Network: Consider the four-area power network described in Section V-G2. In this case though, the value of M_4^a was increased to force the system to be open-loop unstable. At time 10 s, the load in area 2 increases by 15% and simultaneously, the load in area 3 decreases by 15%. The load disturbance rejection performance of terminal control FC-MPC [FC-MPC (tc)] is investigated and compared to the performance of the benchmark centralized constrained LQR (CLQR) [30].

TABLE V
PERFORMANCE OF TERMINAL CONTROL FC-MPC RELATIVE TO CENTRALIZED CONSTRAINED LQR (CLQR) FOR CONTROL OF UNSTABLE FOUR AREA NETWORK. $\Delta\Lambda\% = (\Lambda_{\text{config}} - \Lambda_{\text{cent}})/(\Lambda_{\text{cent}}) \times 100$

	$\Lambda \times 10^{-2}$	$\Delta\Lambda\%$
CLQR	4.91	0
FC-MPC (tc, 1 iterate)	5.52	12.4
FC-MPC (tc, 5 iterates)	4.97	1.2

Fig. 5 depicts the stabilizing and disturbance rejection performance of FC-MPC (tc) and CLQR. Only quantities relating to area 2 are shown as variables in other areas displayed similar qualitative behavior. The associated control costs are given in Table V. For terminal control FC-MPC terminated after one iterate, the load disturbance rejection performance is within 13% of CLQR performance. If five iterates per sampling interval are possible, the incurred performance loss drops to <1.5%.

VII. DISCUSSION AND CONCLUSION

Centralized MPC is not well suited for control of large-scale, geographically expansive systems such as power systems. However, performance benefits obtained with centralized MPC can be realized through distributed MPC strategies. For distributed MPC, the overall system is decomposed into interconnected subsystems. Iterative optimization and exchange of information among the subsystems is performed. An MPC optimization problem is solved within each subsystem, using local measurements and the latest available external information (from the previous iterate).

Various forms of distributed MPC have been considered. It is shown that communication-based MPC is an unreliable strategy for systemwide control and may even result in closed-loop instability. Feasible cooperation-based MPC (FC-MPC), on the other hand, precludes the possibility of parochial controller behavior by forcing the MPCs to cooperate towards achieving systemwide control objectives. A terminal penalty version of FC-MPC was initially established. The solution obtained at convergence of the FC-MPC algorithm is identical to the centralized MPC solution (and therefore, Pareto optimal). In addition, the FC-MPC algorithm can be terminated prior to convergence without compromising feasibility or closed-loop stability of the resulting distributed controller. This feature allows the practitioner to terminate the algorithm at the end of the sampling interval, even if convergence is not achieved. The FC-MPC framework allows smooth transitioning from completely decentralized control to completely centralized control. For each subsystem i, by setting $w_i = 1, w_j = 0, A_{ij} = 0, j \neq i$ in the FC-MPC optimization problem, we revert to decentralized MPC. On the other hand, by iterating the FC-MPC algorithm to convergence, centralized MPC performance is realized. Intermediate termination of the FC-MPC algorithm results in performance between decentralized MPC and centralized MPC control limits.

Several extensions for the terminal penalty distributed MPC framework are possible. The proposed distributed MPC framework can be extended to penalize and constrain the rate of change of inputs. The state for subsystem i is augmented with the input from the previous time step (see [25]). Incorporation of the rate of change of input penalty results in additional terms in the FC-MPC cost function and additional input constraints. All established properties apply however. Details can be found in [33, Ch. 10]. To ensure closed-loop stability while dealing with open-loop unstable systems, a terminal state constraint that forces the unstable modes to be at the origin at the end of the control horizon is necessary. The control horizon must satisfy $N \geq l$, in which l is the number of unstable modes for the system. The FC-MPC optimization problem of (8) is solved with an additional coupled input constraint which forces the unstable modes to the origin at the end of the control horizon. The details for the terminal penalty-based FC-MPC optimization problem for open-loop unstable systems are available in [33, Ch. 10]. It follows that all iterates generated by Algorithm 1 (solving the modified FC-MPC optimization problem with coupled input constraints) are systemwide feasible, the cooperation-based cost function $\Phi(\boldsymbol{u}_1^p, \boldsymbol{u}_2^p, \ldots, \boldsymbol{u}_M^p; x(k))$ is a non-increasing function of the iteration number p, and the sequence of iterates converges. An important distinction, which arises due to the presence of the coupled input constraint, is that the limit points of Algorithm 1 need not be optimal. The distributed MPC control law based on any intermediate iterate is feasible and closed-loop stable, but may not achieve centralized MPC performance at convergence of the iterates.

Because terminal penalty FC-MPC is reliant on a suboptimal parametrization of postulated control trajectories, it cannot achieve infinite horizon optimal performance for finite values of N. In Section VI, a terminal control FC-MPC framework, which achieves infinite horizon optimal performance at convergence with finite values of N, was described. Unlike terminal penalty FC-MPC, the proposed terminal control FC-MPC formulation also allows the handling of unstable systems without the need for a coupled input constraint. Consequently for unstable systems, optimality at convergence can be guaranteed with terminal control FC-MPC. For small values of N, the performance of terminal control FC-MPC is observed to be superior to that of terminal penalty FC-MPC. An alternate strategy for terminal control FC-MPC is to explicitly enforce a terminal constraint that forces each subsystem-based estimate of the state vector to be in $\mathcal{O}_\infty(A, C)$. For small N, this strategy typically leads to excessively aggressive controller response, which is undesirable. Enforcing the terminal set constraint explicitly also introduces a coupled input constraint. For this formulation, feasibility and stability of the resulting control law can be shown. Optimality at convergence, however, is not necessarily obtained. Further details are available in [33].

Examples were presented to illustrate the applicability and effectiveness of the proposed distributed MPC framework for AGC. First, a two-area network was considered. Both communication-based MPC and cooperation-based MPC outperformed AGC due to their ability to handle process constraints. The controller defined by terminating Algorithm 1 after five iterates achieved performance that was almost identical to centralized MPC. Next, the performance of the different MPC frameworks was evaluated for a four-area network. For this case, communication-based MPC led to closed-loop instability. FC-MPC (one iterate) stabilized the system and achieved performance that was within 26% of centralized MPC performance. The two-area network considered earlier, with an additional FACTS device to control tie-line impedance, was examined subsequently. Communication-based MPC stabilized the system but gave unacceptable closed-loop performance. The FC-MPC framework was shown to allow coordination of FACTS controls with AGC. The controller defined by terminating Algorithm 1 after just one iterate gave an improvement in performance of around 190% compared to communication-based MPC. For this case, therefore, the cooperative aspect of FC-MPC was very important for achieving acceptable response. Finally, terminal control FC-MPC was employed for control of an open-loop unstable four area network. Terminal control FC-MPC, terminated after five iterates gave performance that was within 1.5% of the infinite horizon optimal control performance. At convergence, the performance of terminal control FC-MPC is always within a prespecified tolerance of the infinite horizon optimal control performance.

APPENDIX A
TERMINAL PENALTY FC-MPC

Lemma 3 (Minimum Principle for Constrained, Convex Optimization): Let $\mathcal{X}$ be a convex set and let f be a convex function over $\mathcal{X}$. A necessary and sufficient condition for x^* to be a global minimum of f over $\mathcal{X}$ is

$$\nabla f(x^*)'(x - x^*) \geq 0, \; \forall x \in \mathcal{X}.$$

A proof is given in [5, p. 194].

Proof of Lemma 1: From Algorithm 1, we know that

$$\Phi\left(\boldsymbol{u}_1^{p-1}, \ldots \boldsymbol{u}_{i-1}^{p-1}, \boldsymbol{u}_i^{*(p)}, \boldsymbol{u}_{i+1}^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}; x(k)\right) \leq \Phi\left(\boldsymbol{u}_1^{p-1}, \boldsymbol{u}_2^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}; x(k)\right) \quad \forall i = 1, 2, \ldots, M \qquad (21)$$

Therefore, from the definition of $\boldsymbol{u}_i^p$ (Algorithm 1), we have

$$\begin{aligned}
&\Phi\left(\boldsymbol{u}_1^p, \boldsymbol{u}_2^p, \ldots, \boldsymbol{u}_M^p; x(k)\right) \\
&\quad = \Phi\Big(w_1 \boldsymbol{u}_1^{*(p)} + (1-w_1)\boldsymbol{u}_1^{p-1}, \ldots, \\
&\qquad\qquad w_M \boldsymbol{u}_M^{*(p)} + (1-w_M)\boldsymbol{u}_M^{p-1}; x(k)\Big) \\
&\quad = \Phi\Big(w_1 \boldsymbol{u}_1^{*(p)} + w_2 \boldsymbol{u}_1^{p-1} + \cdots + w_M \boldsymbol{u}_1^{p-1}, \\
&\qquad\qquad w_1 \boldsymbol{u}_2^{p-1} + w_2 \boldsymbol{u}_2^{*(p)} + \cdots + w_M \boldsymbol{u}_2^{p-1}, \ldots\ldots, \\
&\qquad\qquad w_1 \boldsymbol{u}_M^{p-1} + w_2 \boldsymbol{u}_M^{p-1} + \cdots + w_M \boldsymbol{u}_M^{*(p)}; x(k)\Big)
\end{aligned}$$

By convexity of $\Phi(\cdot)$

$$\begin{aligned}
&\leq \sum_{r=1}^{M} w_r \Phi\left(\boldsymbol{u}_1^{p-1}, \ldots, \boldsymbol{u}_{r-1}^{p-1}, \boldsymbol{u}_r^{*(p)}, \boldsymbol{u}_{r+1}^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}; x(k)\right) \\
&\leq \sum_{r=1}^{M} w_r \Phi\left(\boldsymbol{u}_1^{p-1}, \ldots, \boldsymbol{u}_{r-1}^{p-1}, \boldsymbol{u}_r^{p-1}, \boldsymbol{u}_{r+1}^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}; x(k)\right) \\
&= \Phi\left(\boldsymbol{u}_1^{p-1}, \boldsymbol{u}_2^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}; x(k)\right) \qquad (22)
\end{aligned}$$

in which equality is obtained if $\boldsymbol{u}_i^p = \boldsymbol{u}_i^{p-1}, \forall\, i = 1, 2, \ldots, M$. □

Proof of Lemma 2: Since the level set

$$S_0 = \{(\bar{\boldsymbol{u}}_1, \bar{\boldsymbol{u}}_2, \ldots, \bar{\boldsymbol{u}}_M) \mid \Phi(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_M; x(k)) \leq \Phi(\boldsymbol{u}_1^0, \boldsymbol{u}_2^0, \ldots, \boldsymbol{u}_M^0; x(k))\}$$

is closed and bounded (hence compact), a limit point for Algorithm 1 exists. We know that $(\boldsymbol{u}_1^*, \ldots, \boldsymbol{u}_M^*)$ is the unique solution for the centralized MPC optimization problem (11). Let $\Phi^* = \Phi(\boldsymbol{u}_1^*, \boldsymbol{u}_2^*, \ldots, \boldsymbol{u}_M^*; x(k))$. Define $\boldsymbol{u}_i^\infty = [\bar{\boldsymbol{u}}_i^{\infty\prime}, 0, 0, \ldots]'$. Assume that the sequence $(\bar{\boldsymbol{u}}_1^p, \bar{\boldsymbol{u}}_2^p, \ldots, \bar{\boldsymbol{u}}_M^p)$, generated by Algorithm 1, converges to a feasible subset of the non-optimal level set

$$S_\infty = \{(\bar{\boldsymbol{u}}_1, \bar{\boldsymbol{u}}_2, \ldots, \bar{\boldsymbol{u}}_M) \mid \Phi(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_M; x(k)) = \Phi^\infty\}.$$

Since $\Phi(\cdot)$ is strictly convex and by assumption of non-optimality $\Phi^\infty > \Phi^*$. Let $(\bar{\boldsymbol{u}}_1^\infty, \ldots, \bar{\boldsymbol{u}}_M^\infty) \in S_\infty$ be generated by Algorithm 1 for p large. To establish convergence of Algorithm 1 to a point rather than a limit set, we assume the contrary and show a contradiction. Suppose that Algorithm 1 does not converge to a point. Our assumption here implies that there exists $(\bar{\boldsymbol{v}}_1, \ldots, \bar{\boldsymbol{v}}_M) \in S_\infty$ generated by the next iterate of Algorithm 1 with $(\bar{\boldsymbol{v}}_1, \ldots, \bar{\boldsymbol{v}}_M) \neq (\bar{\boldsymbol{u}}_1^\infty, \ldots, \bar{\boldsymbol{u}}_M^\infty)$. Consider the set of optimization problems

$$\begin{aligned}
\boldsymbol{z}_i^\infty = \arg\min_{\boldsymbol{u}_i} \Phi\big(&\boldsymbol{u}_1^\infty, \ldots, \boldsymbol{u}_{i-1}^\infty, \boldsymbol{u}_i, \boldsymbol{u}_{i+1}^\infty, \ldots, \\
&\boldsymbol{u}_M^\infty; x(k)\big) \qquad (23a)
\end{aligned}$$

$$u_i(l \mid k) \in \Omega_i,\ 0 \leq l \leq N-1 \qquad (23b)$$

$$u_i(l \mid k) = 0,\ N \leq l \qquad (23c)$$

$$\forall i = 1, 2, \ldots, M.$$

We have $\boldsymbol{z}_i^\infty = [\bar{\boldsymbol{z}}_i^{\infty\prime}, 0, 0, \ldots]'$ in which $\bar{z}_i^\infty = [z_i^\infty(0)', \ldots, z_i^\infty(N-1)']'$. By assumption, there exists at least one i for which $\boldsymbol{z}_i^\infty \neq \boldsymbol{u}_i^\infty$. WLOG let $\boldsymbol{z}_1^\infty \neq \boldsymbol{u}_1^\infty$. By definition, $\bar{\boldsymbol{v}}_i = w_i \bar{\boldsymbol{z}}_i^\infty + (1-w_i)\bar{\boldsymbol{u}}_i^\infty, \forall i = 1, 2, \ldots, M$. It follows that $\boldsymbol{v}_i = [\bar{\boldsymbol{v}}_i{}', 0, 0, \ldots]'$. Since $(\bar{\boldsymbol{v}}_1, \bar{\boldsymbol{v}}_2, \ldots, \bar{\boldsymbol{v}}_M) \in S_\infty$, $\Phi(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_M; x(k)) = \Phi^\infty$. Using convexity of $\Phi(\cdot)$, we have

$$\begin{aligned}
\Phi^\infty &= \Phi(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_M; x(k)) \\
&= \Phi(w_1 \boldsymbol{z}_1^\infty + (1-w_1)\boldsymbol{u}_1^\infty, \ldots, \\
&\qquad w_M \boldsymbol{z}_M^\infty + (1-w_M)\boldsymbol{u}_M^\infty; x(k)) \\
&< w_1 \Phi(\boldsymbol{z}_1^\infty, \boldsymbol{u}_2^\infty, \ldots, \boldsymbol{u}_M^\infty; x(k)) + \cdots \\
&\quad + w_M \Phi(\boldsymbol{u}_1^\infty, \ldots, \boldsymbol{u}_{M-1}^\infty, \boldsymbol{z}_M^\infty; x(k)) \\
&< w_1 \Phi^\infty + \cdots + w_M \Phi^\infty \\
&= \Phi^\infty
\end{aligned}$$

in which the strict inequality follows from $\boldsymbol{z}_i^\infty \neq \boldsymbol{u}_i^\infty$ for at least one $i = 1, 2, \ldots, M$. Hence, a contradiction. Suppose now that $(\bar{\boldsymbol{u}}_1^p, \bar{\boldsymbol{u}}_2^p, \ldots, \bar{\boldsymbol{u}}_M^p) \to (\bar{\boldsymbol{u}}_1^\infty, \bar{\boldsymbol{u}}_2^\infty, \ldots, \bar{\boldsymbol{u}}_M^\infty) \neq (\bar{\boldsymbol{u}}_1^*, \bar{\boldsymbol{u}}_2^*, \ldots, \bar{\boldsymbol{u}}_M^*)$. From uniqueness of the optimizer, $\Phi(\boldsymbol{u}_1^*, \boldsymbol{u}_2^*, \ldots, \boldsymbol{u}_M^*; x(k)) < \Phi(\boldsymbol{u}_1^\infty, \boldsymbol{u}_2^\infty, \ldots, \boldsymbol{u}_M^\infty; x(k))$. Since $(\bar{\boldsymbol{u}}_1^p, \bar{\boldsymbol{u}}_2^p, \ldots, \bar{\boldsymbol{u}}_M^p)$, generated using Algorithm 1, converges to $(\bar{\boldsymbol{u}}_1^\infty, \bar{\boldsymbol{u}}_2^\infty, \ldots, \bar{\boldsymbol{u}}_M^\infty)$, we have

$$\begin{aligned}
\boldsymbol{u}_i^\infty = \arg\min_{\boldsymbol{u}_i} \Phi(&\boldsymbol{u}_1^\infty, \ldots, \boldsymbol{u}_{i-1}^\infty, \boldsymbol{u}_i, \boldsymbol{u}_{i+1}^\infty, \ldots, \\
&\boldsymbol{u}_M^\infty; x(k)) \qquad (24a)
\end{aligned}$$

$$u_i(l \mid k) \in \Omega_i, 0 \leq l \leq N-1 \qquad (24b)$$

$$u_i(l \mid k) = 0, N \leq l \qquad (24c)$$

$$\forall i = 1, 2, \ldots, M.$$

From Lemma 3

$$\nabla_{\bar{\boldsymbol{u}}_j} \Phi(\boldsymbol{u}_1^\infty, \ldots, \boldsymbol{u}_M^\infty; x(k))'(\bar{\boldsymbol{u}}_j^* - \bar{\boldsymbol{u}}_j^\infty) \geq 0,\ \forall j = 1, 2, \ldots, M.$$

Define $\Delta\bar{\boldsymbol{u}}_j = \bar{\boldsymbol{u}}_j^* - \bar{\boldsymbol{u}}_j^\infty$ and $\Delta\boldsymbol{u}_j = \boldsymbol{u}_j^* - \boldsymbol{u}^\infty{}_j = [\Delta\bar{\boldsymbol{u}}_j{}', 0, 0, \ldots]'$, $\forall j = 1, 2, \ldots, M$. We have, from our assumption $(\bar{\boldsymbol{u}}_1^\infty, \bar{\boldsymbol{u}}_2^\infty, \ldots, \bar{\boldsymbol{u}}_M^\infty) \neq (\bar{\boldsymbol{u}}_1^*, \bar{\boldsymbol{u}}_2^*, \ldots, \bar{\boldsymbol{u}}_M^*)$, that $\Delta\bar{\boldsymbol{u}}_i \neq 0$ for at least one index i, $1 \leq i \leq M$.

A second-order Taylor series expansion around $(\boldsymbol{u}_1^\infty, \boldsymbol{u}_2^\infty, \ldots, \boldsymbol{u}_M^\infty)$ gives

$$\begin{aligned}
&\Phi(\boldsymbol{u}_1^*, \boldsymbol{u}_2^*, \ldots, \boldsymbol{u}_M^*; x(k)) \\
&\quad = \Phi(\boldsymbol{u}_1^\infty + \Delta\boldsymbol{u}_1, \boldsymbol{u}_2^\infty + \Delta\boldsymbol{u}_2, \ldots, \boldsymbol{u}_M^\infty + \Delta\boldsymbol{u}_M; x(k)) \\
&\quad = \Phi(\boldsymbol{u}_1^\infty, \ldots, \boldsymbol{u}_M^\infty; x(k)) \\
&\qquad + \underbrace{\sum_{j=1}^{M} \nabla_{\bar{\boldsymbol{u}}_j} \Phi(\boldsymbol{u}_1^\infty, \ldots, \boldsymbol{u}_M^\infty; x(k))' \Delta\bar{\boldsymbol{u}}_j}_{\geq 0, \text{Lemma 3}} \\
&\qquad + \underbrace{\frac{1}{2} \begin{bmatrix} \Delta\bar{\boldsymbol{u}}_1 \\ \vdots \\ \Delta\bar{\boldsymbol{u}}_M \end{bmatrix}' \nabla^2 \Phi(\boldsymbol{u}_1^\infty, \ldots, \boldsymbol{u}_M^\infty; x(k)) \begin{bmatrix} \Delta\bar{\boldsymbol{u}}_1 \\ \vdots \\ \Delta\bar{\boldsymbol{u}}_M \end{bmatrix}}_{\geq 0, \text{ since } \Phi(\cdot) \text{ p.d. quadratic}}.
\end{aligned} \qquad (25)$$

Using (25) and optimality of $(\boldsymbol{u}_1^*, \boldsymbol{u}_2^*, \ldots, \boldsymbol{u}_M^*)$ gives

$$\begin{aligned}
\Phi(\boldsymbol{u}_1^*, \ldots, \boldsymbol{u}_M^*; x(k)) &= \Phi\left(\boldsymbol{u}_1^\infty, \ldots, \boldsymbol{u}_M^\infty; x(k)\right) \\
&\quad + \beta(\Delta\boldsymbol{u}_1, \ldots, \Delta\boldsymbol{u}_M) \\
&\leq \Phi\left(\boldsymbol{u}_1^\infty, \ldots, \boldsymbol{u}_M^\infty; x(k)\right) \qquad (26)
\end{aligned}$$

in which $\beta(\cdot)$ is a positive definite function (from (25)). We have from (26) that $\beta(\cdot) \leq 0$, which implies $\beta(\Delta\boldsymbol{u}_1, \ldots, \Delta\boldsymbol{u}_M) = 0$. It follows, therefore, that

$\bar{\boldsymbol{u}}_j^{\infty} = \bar{\boldsymbol{u}}_j^*, \forall j = 1,2,\ldots,M$. Using the previous relation gives $\boldsymbol{u}_j^{\infty} = [\bar{\boldsymbol{u}}_j^{\infty\prime}, 0, 0, \ldots]' = \boldsymbol{u}_j^*$, $\forall j = 1,2,\ldots,M$. Hence, $\Phi(\boldsymbol{u}_1^*, \boldsymbol{u}_2^*, \ldots, \boldsymbol{u}_M^*; x(k)) = \Phi(\boldsymbol{u}_1^{\infty}, \boldsymbol{u}_2^{\infty}, \ldots, \boldsymbol{u}_M^{\infty}; x(k))$. □

Lemma 4: Let the input constraints in (8) be specified in terms of a collection of linear inequalities. Consider the closed ball $B_\varepsilon(0)$, in which $\varepsilon > 0$ is chosen such that the input constraints in each FC-MPC optimization problem (8) are inactive for each $x(k) \in B_\varepsilon(0)$. The distributed MPC control law defined by the FC-MPC formulation of Theorem 1 is a Lipschitz continuous function of $x(k)$, for all $x(k) \in B_\varepsilon(0)$.

A proof is available in [33, Ch. 10].

Proof of Theorem 1: Since $\mathcal{Q} > 0$ and A is stable, $P > 0$ [31]. The constrained stabilizable set $\mathcal{X}$ for the system is $\mathbb{R}^n$. To prove exponential stability, we use the value function $J_N^{p(k)}(x(k))$ as a candidate Lyapunov function. We need to show [36, p. 267] that there exists constants $a, b, c > 0$, such that

$$a\|x(k)\|^2 \le J_N^p(x(k)) \le b\|x(k)\|^2 \tag{27a}$$

$$\Delta J_N^p(x(k)) \le -c\|x(k)\|^2 \tag{27b}$$

in which $\Delta J_N^{p(k)}(x(k)) = J_N^{p(k+1)}(x(k+1)) - J_N^{p(k)}(x(k))$.

Let $\varepsilon > 0$ be chosen such that the input constraints remain inactive for $x \in B_\varepsilon(0)$. Such an ε exists because the origin is Lyapunov stable and $0 \in \mathrm{int}(\Omega_1 \times \cdots \times \Omega_M)$. Since Ω_i is compact $\forall i = 1,2,\ldots,M$, there exists $\sigma > 0$ such that $\|\bar{\boldsymbol{u}}_i\| \le \sigma$. For any x satisfying $\|x\| > \varepsilon$, $\|\bar{\boldsymbol{u}}_i\| < \sigma/\varepsilon\|x\|$, $\forall i = 1,2,\ldots,M$. For $x(k) \in B_\varepsilon(0)$, we have from Lemma 4 that $\bar{\boldsymbol{u}}_i^{p(k)}(x(k))$ is a Lipschitz continuous function of $x(k)$. There exists, therefore, a constant $\rho > 0$, such that $\|\bar{\boldsymbol{u}}_i^{p(k)}(x(k))\| \le \rho\|x(k)\|$, $\forall 0 < p(k) \le p^*$. Define $K_u = \max(\sigma/\varepsilon, \rho)^2$, in which $K_u > 0$ and independent of $x(k)$. The previous definition gives $\|u_i^{p(k)}(k+j; x(k))\| \le \sqrt{K_u}\|x(k)\|$, $\forall i = 1,2,\ldots,M, k \ge 0$ and all $0 < p(k) \le p_{\max}(k)$. For $j \ge 0$, define $u(k+j\,|\,k) = [u_1^{p(k)}(k+j;x(k))', \ldots, u_M^{p(k)}(k+j;x(k))']'$. By definition, $u(k\,|\,k) \equiv u(k)$. We have $\|u(k+j\,|\,k)\| = \sqrt{\sum_{i=1}^M \|u_i^{p(k)}(k+j;x(k))\|^2} \le \sqrt{K_u M}\|x(k)\|$. Similarly, define $x(k+j\,|\,k) = [x_1^{p(k)}(k+j\,|\,k)', \ldots, x_M^{p(k)}(k+j\,|\,k)']'$, $\forall j \ge 0$. By definition $x(k\,|\,k) \equiv x(k)$. Since A is stable, there exists $\bar{c} > 0$, such that $\|A^j\| \le \bar{c}\lambda^j$ [19, Corollary 5.6.13, p. 199], in which $\lambda_{\max}(A) \le \lambda < 1$. Hence

$$\begin{aligned}\|x(k+j|k)\| &\le \|A^j\|\|x(k)\| + \sum_{l=0}^{j-1}\|A^{j-1-l}\|\|B\|\|u(k+l|k)\| \\ &\le \bar{c}\lambda^j\|x(k)\| + \sum_{l=0}^{j-1}\bar{c}\lambda^{j-1-l}\|B\|\|u(k+l\,|\,k)\| \\ &\le \bar{c}\left(1 + \frac{\|B\|}{1-\lambda}\sqrt{MK_u}\right)\|x(k)\|\,\forall j > 0\end{aligned}$$

since $\sum_{l=0}^{j}\lambda^l \le \sum_{l=0}^{\infty}\lambda^l = 1/(1-\lambda), \forall j \ge 0$. Let $\mathcal{R} = \mathrm{diag}(w_1R_1, w_2R_2, \ldots, w_MR_M)$ and $\Gamma = [\bar{c}(1 + \|B\|/(1-\lambda)\sqrt{MK_u})]^2$. Then

$$\begin{aligned}J_N^{p(k)}(x(k)) &= \frac{1}{2}\sum_{i=1}^M w_i \sum_{j=0}^{\infty}\left[\left\|x_i^{p(k)}(k+j\,|\,k)\right\|_{Q_i}^2 + \left\|u_i^{p(k)}(k+j\,|\,k)\right\|_{R_i}^2\right] \\ &= \frac{1}{2}\sum_{j=0}^{N-1}\left[x(k+j\,|\,k)'\mathcal{Q}x(k+j\,|\,k) + u(k+j\,|\,k)'\mathcal{R}u(k+j\,|\,k)\right] + \frac{1}{2}x(k+N\,|\,k)'Px(k+N\,|\,k) \\ &\le \frac{1}{2}\left[\sum_{j=0}^{N-1}\left(\lambda_{\max}(\mathcal{Q})\|x(k+j\,|\,k)\|^2 + \lambda_{\max}(\mathcal{R})\|u(k+j\,|\,k)\|^2\right) + \lambda_{\max}(P)\|x(k+N\,|\,k)\|^2\right] \\ &\le \frac{1}{2}\left[N\lambda_{\max}(\mathcal{Q})\Gamma + N\lambda_{\max}(\mathcal{R})K_uM + \lambda_{\max}(P)\Gamma\right]\|x(k)\|^2 \\ &\le b\|x(k)\|^2\end{aligned}$$

in which $0 < \frac{1}{2}[N\lambda_{\max}(\mathcal{Q})\Gamma + N\lambda_{\max}(\mathcal{R})K_uM + \lambda_{\max}(P)\Gamma] \le b$.

Also, $(1/2)\lambda_{\min}(\mathcal{Q})\|x(k)\|^2 \le J_N^{p(k)}(x(k))$. Furthermore

$$\begin{aligned}J_N^{p(k+1)}(x(k+1)) - J_N^{p(k)}(x(k)) &\le J_N^0(x(k+1)) - J_N^{p(k)}(x(k)) \\ &= -\sum_{i=1}^M w_iL_i\left(x_i(k), u_i^{p(k)}(k;x(k))\right) \\ &\le -\sum_{i=1}^M w_iL_i(x_i(k),0) \\ &= -\frac{1}{2}x(k)'\mathcal{Q}x(k) \\ &\le -\frac{1}{2}\lambda_{\min}(\mathcal{Q})\|x(k)\|^2\end{aligned} \tag{28}$$

which proves the theorem. □

REFERENCES

[1] N. Atić, D. Rerkpreedapong, A. Hasanović, and A. Feliachi, "NERC compliant decentralized load frequency control design using model predictive control," in *Proc. IEEE PES General Meet.*, Jun. 2003, pp. 554–559.

[2] T. Başar, "Asynchronous algorithms in non-cooperative games," *J. Econom. Dyn. Control*, vol. 12, pp. 167–172, 1988.

[3] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory*. Philadelphia, PA: SIAM, 1999.

[4] D. P. Bertsekas, *Dynamic Programming*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[5] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, 1999.

[6] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, pp. 1747–1767, 1999.

[7] E. Camacho and C. Bordons, *Model Predictive Control*, 2nd ed. Berlin, Germany: Springer, 2004.

[8] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar, "Distributed model predictive control," *IEEE Control Syst. Mag.*, vol. 9, no. 1, pp. 44–52, Jan. 2002.

[9] S. W. Chan, G. C. Goodwin, and K. S. Sin, "Convergence properties of the Riccati difference equation in optimal filtering of nonstabilizable systems," *IEEE Trans. Autom. Control*, vol. 29, no. 2, pp. 110–118, Feb. 1984.

[10] J. E. Cohen, "Cooperation and self interest: Pareto-inefficiency of Nash equilibria in finite random games," in *Proc. Nat. Acad. Sci.*, Aug. 1998, vol. 95, pp. 9724–9731.

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

[12] W. B. Dunbar, "Distributed receding horizon control of dynamically coupled nonlinear systems," *IEEE Trans. Autom. Control*, vol. 52, no. 7, pp. 1249–1263, Jul. 2007.

[13] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, no. 4, pp. 549–558, Apr. 2006.

[14] M. Ghandhari, G. Andersson, and I. A. Hiskens, "Control Lyapunov functions for controllable series devices," *IEEE Trans. Power Syst.*, vol. 16, no. 4, pp. 689–694, Nov. 2001.

[15] E. G. Gilbert and K. T. Tan, "Linear systems with state and control constraints: The theory and application of maximal output admissible sets," *IEEE Trans. Autom. Control*, vol. 36, no. 9, pp. 1008–1020, Sep. 1991.

[16] P.-O. Gutman and M. Cwikel, "An algorithm to find maximal state constraint sets for discrete-time linear dynamical systems with bounded controls and states," *IEEE Trans. Autom. Control*, vol. 32, no. 3, pp. 251–254, Mar. 1987.

[17] V. Havlena and J. Lu, "A distributed automation framework for plant-wide control, optimisation, scheduling and planning," presented at the 16th IFAC World Congr., Prague, Czech Republic, Jul. 2005.

[18] N. G. Hingorani and L. Gyugyi, *Understanding FACTS*. Piscataway, NJ: IEEE Press, 2000.

[19] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.

[20] D. Jia and B. H. Krogh, "Distributed model predictive control," in *Proc. Amer. Control Conf.*, Jun. 2001, pp. 2767–2772.

[21] D. Jia and B. H. Krogh, "Min-max feedback model predictive control for distributed control with communication," in *Proc. Amer. Control Conf.*, May 2002, pp. 4507–4512.

[22] S. S. Keerthi and E. G. Gilbert, "Computation of minimum-time feedback control laws for discrete time-systems with state-control constraints," *IEEE Trans. Autom. Control*, vol. 32, no. 5, pp. 432–435, May 1987.

[23] B. H. Krogh and P. V. Kokotovic, "Feedback control of overloaded networks," *IEEE Trans. Autom. Control*, vol. 29, no. 8, pp. 704–711, Aug. 1984.

[24] R. Kulhavý, J. Lu, and T. Samad, "Emerging technologies for enterprise optimization in the process industries," in *Proc. Chem. Process Control—VI: 6th Int. Conf. Chem. Process Control*, J. B. Rawlings, B. A. Ogunnaike, and J. W. Eaton, Eds., Jan. 2001, vol. 98, no. 326, pp. 352–363, AIChE Symp. Series.

[25] K. R. Muske and J. B. Rawlings, "Model predictive control with linear models," *AIChE J.*, vol. 39, no. 2, pp. 262–287, 1993.

[26] R. Neck and E. Dockner, "Conflict and cooperation in a model of stabilization policies: A differential game approach," *J. Econom. Dyn. Control*, vol. 11, pp. 153–158, 1987.

[27] R. Piwko, D. Osborn, R. Gramlich, G. Jordan, D. Hawkins, and K. Porter, "Wind energy delivery issues," *IEEE Power Energy Mag.*, vol. 3, no. 6, pp. 47–56, Nov./Dec. 2005.

[28] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Eng. Practice*, vol. 11, no. 7, pp. 733–764, 2003.

[29] N. R. Sandell, Jr., P. Varaiya, M. Athans, and M. Safonov, "Survey of decentralized control methods for larger scale systems," *IEEE Trans. Autom. Control*, vol. 23, no. 2, pp. 108–128, Feb. 1978.

[30] P. O. M. Scokaert and J. B. Rawlings, "Constrained linear quadratic regulation," *IEEE Trans. Autom. Control*, vol. 43, no. 8, pp. 1163–1169, Aug. 1998.

[31] E. D. Sontag, *Mathematical Control Theory*, 2nd ed. New York: Springer-Verlag, 1998.

[32] M. Sznaier and M. J. Damborg, "Heuristically enhanced feedback control of constrained discrete-time linear systems," *Automatica*, vol. 26, no. 3, pp. 521–532, 1990.

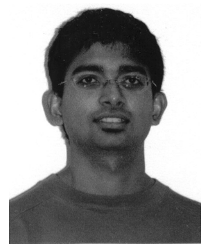
[33] A. N. Venkat, "Distributed model predictive control: Theory and applications" Ph.D. dissertation, Dept. Chem. Biol. Eng., Univ. Wisconsin-Madison, Madison, Oct. 2006 [Online]. Available: http://jbrwww.che.wisc.edu/theses/venkat.pdf

[34] A. N. Venkat, I. A. Hiskens, R. B. Rawlings, and S. J. Wright, "Distributed MPC strategies for automatic generation control," presented at the IFAC Symp. Power Plants Power Syst. Control Kananaskis, Canada, Jun. 2006.

[35] A. N. Venkat, I. A. Hiskens, R. B. Rawlings, and S. J. Wright, "Distributed output feedback MPC strategies for power system control," in *Proc. IEEE Conf. Dec. Control*, Dec. 2006, pp. 4038–4045.

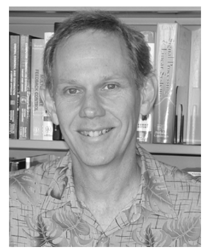
[36] M. Vidyasagar, *Nonlinear Systems Analysis*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[37] A. J. Wood and B. F. Wollenberg, *Power Generation Operation and Control*. New York: Wiley, 1996.



Aswin N. Venkat received the B.Tech. degree in chemical engineering from the Indian Institute of Technology, Mumbai, India, in 2001, and the Ph.D. degree in chemical engineering from the University of Wisconsin, Madison, in 2006.

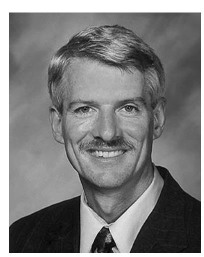
He is currently with the Process Modeling, Control, and Optimization Group, Shell Global Solutions (US) Inc., Westhollow Technology Center, Houston, TX. His research interests include the areas of model predictive control, state estimation, plantwide control and monitoring, real time optimization, and resource scheduling.



Ian A. Hiskens (S'77–M'80–SM'96–F'06) received the B.Eng. degree in electrical engineering and the B.App.Sc. degree in mathematics from the Capricornia Institute of Advanced Education, Rockhampton, Australia, in 1980 and 1983, respectively, and the Ph.D. degree from the University of Newcastle, Newcastle, Australia, in 1991.

He is currently a Professor with the Department of Electrical and Computer Engineering, University of Wisconsin-Madison, Madison. He has held prior appointments with the Queensland Electricity Supply Industry, Australia, from 1980 to 1992, the University of Newcastle, from 1992 to 1999, and the University of Illinois at Urbana-Champaign, from 1999 to 2002. His major research interests include the area of power system analysis, in particular system dynamic performance, security, and numerical techniques. Other research interests include nonlinear and hybrid dynamical systems.

Prof. Hiskens was an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS from 2002 to 2005, and is currently an Associate Editor of the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY. He is also the Treasurer of the IEEE Systems Council.



James B. Rawlings received the B.S. degree from the University of Texas, Austin, in 1979, and the Ph.D. degree from the University of Wisconsin, Madison, in 1985, both in chemical engineering.

He is currently the Paul A. Elfers Professor of Chemical and Biological Engineering, University of Wisconsin, and the Codirector of the Texas-Wisconsin Modeling and Control Consortium (TWMCC). He has held prior appointments at the University of Stuttgart, Stuttgart, Germany, as a NATO Postdoctoral Fellow and as a faculty member at the University of Texas. His research interests include the areas of chemical process modeling, molecular-scale chemical reaction engineering, monitoring and control, nonlinear model predictive control, and moving horizon state estimation.

Stephen J. Wright received the B.Sc. (Hons.) and Ph.D. degrees from the University of Queensland, Queensland, Australia, in 1981 and 1984, respectively.

After holding positions at North Carolina State University, Argonne National Laboratory, and the University of Chicago, he joined the Computer Sciences Department, University of Wisconsin-Madison, Madison, as a Professor in 2001. His research interests include theory, algorithms, and applications of computational optimization.

Prof. Wright is Chair of the Mathematical Programming Society and has served on the editorial boards of *Mathematical Programming, Series A and B*, the *SIAM Journal on Optimization*, the *SIAM Journal on Scientific Computing*, and other journals. He also serves on the Board of Trustees of the Society for Industrial and Applied Mathematics (SIAM).