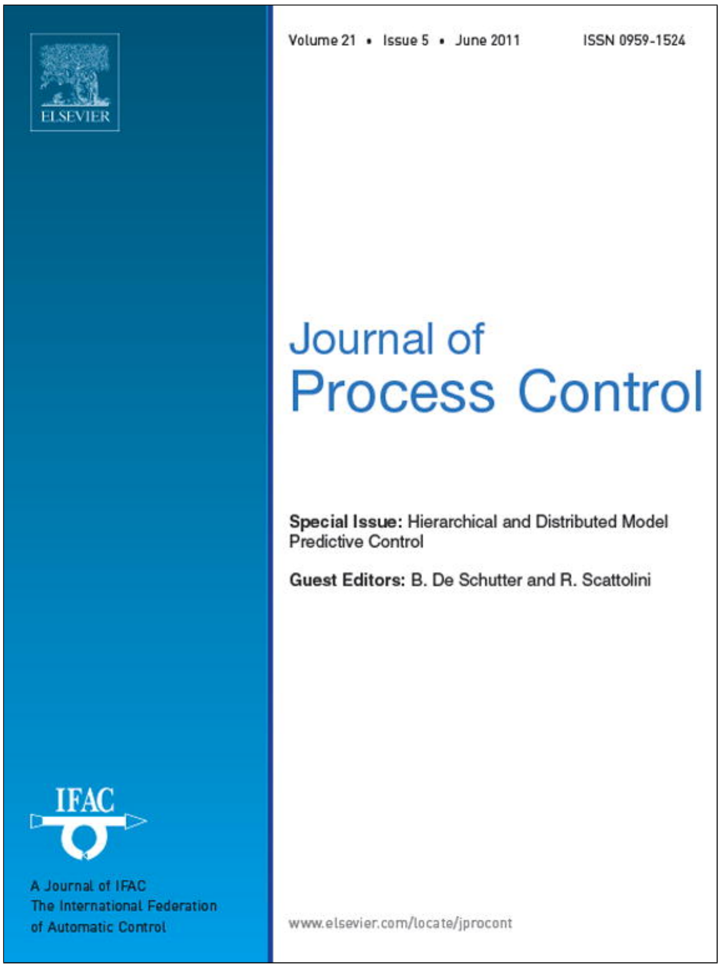

ELSEVIER
Volume 21 • Issue 5 • June 2011
ISSN 0959-1524
Journal of
Process Control
Special Issue: Hierarchical and Distributed Model Predictive Control
Guest Editors: B. De Schutter and R. Scattolini
IFAC
A Journal of IFAC
The International Federation
of Automatic Control
www.elsevier.com/locate/jprocont

Journal of Process Control 21 (2011) 698–704

Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

Cooperative distributed model predictive control for nonlinear systems☆

Brett T. Stewart [a], Stephen J. Wright [b], James B. Rawlings [a,*]

[a] Chemical and Biological Engineering Dept., Univ. of Wisconsin, Madison, WI, USA
[b] Computer Science Dept., Univ. of Wisconsin, Madison, WI, USA

ARTICLE INFO

Article history:
Received 21 June 2010
Received in revised form
13 November 2010
Accepted 14 November 2010
Available online 17 December 2010

Keywords:
Distributed control
Cooperative control
Large scale control
Model predictive control
Nonlinear control

ABSTRACT

In this paper, a distributed controller is presented that can stabilize nonlinear systems. A novel nonlinear nonconvex optimizer is proposed that improves the objective function and is feasible at every iterate. The optimization uses gradient projection and converges to stationary points. The proposed optimization does not require a coordination layer, and hence the controller is truly distributed. Asymptotic stability is established for the controlled system, and an illustrative example is presented.

Published by Elsevier Ltd.

1. Introduction

Model predictive control has become a popular control strategy for many industrial applications because it can handle hard constraints, nonlinear models, and systems with a large number of actuators and measurements [10]. At each sampling time, an open-loop input sequence is obtained from an optimization and the first input is injected into the plant [7]. Large-scale industrial plants usually comprise many subsystems that each contribute to the plantwide goal of converting raw materials into products. These subsystems are connected through material, energy, and information streams, and hence they dynamically interact with one another and affect closed-loop controller performance. Therefore any plantwide control strategy must account for these interactions.

Traditionally plantwide control has been accomplished through decentralized strategies [13]. These methods seek to decompose the plant into weakly interacting subsystems. Controllers are designed for the chosen subsystems in a decentralized fashion, i.e., by ignoring the inter-subsystem interactions. The decentralized controllers may be deliberately tuned for slow, nominal closed-loop performance to maintain closed-loop stability when the neglected interactions are present in the plant operation [16]. It is well known, however, that for plants with strong coupling, decentralized control may not provide good performance and may not even stabilize the system [3]. Cooperative distributed control, on the other hand, has been shown recently to be stabilizing for plants with even strong coupling [17,6]. In cooperative distributed control, the decomposition of the plant into subsystems is not so critical; the strength of the subsystem interactions may influence the closed-loop performance, but not the closed-loop stability. In cooperative distributed control, the subsystems solve optimizations of the plantwide objective independently and exchange information to coordinate their actions as the sample time permits. In previous work, we have proposed a distributed controller for linear plants that does not require a coordinator or hierarchical decomposition [17,18,12].

Recent work in nonlinear distributed control has established plantwide stabilizing properties by the addition of a hierarchy of controllers or through a plantwide coordinator [14,2]. Liu et al. [4] propose a two-tier distributed controller. A plantwide stabilizing controller is assumed to exist a Lyapunov-based controller that stabilizes the plant using only a subset of the plant's inputs. A second tier controller, which accounts for the closed-loop performance of the first tier controller, is used to manipulate the other plant inputs. In further work [5], the authors extend the controller to had a finite number of weakly interacting subsystems and relax the requirement that one controller can stabilize the plantwide system. Necoara et al. [8] develop a nonlinear distributed controller that linearizes the dynamics at each optimization iterate and solves a sequential convex program to optimality. The optimization algorithm uses dual decomposition, and relies upon a central dual optimization to find approximate multipliers at each iteration and does not achieve feasibility until convergence.

☆ This paper was not presented at any IFAC meeting.

* Corresponding author. Tel.: +1 608 263 5859; fax: +1 608 265 8794.
E-mail addresses: btstewart@wisc.edu (B.T. Stewart), swright@cs.wisc.edu (S.J. Wright), rawlings@engr.wisc.edu (J.B. Rawlings).

0959-1524/$ – see front matter. Published by Elsevier Ltd.
doi:10.1016/j.jprocont.2010.11.004

To have a fully distributed controller, we propose the following two criteria for the optimization used to determine the plantwide control action: (i) the optimizers should *not* rely on a central coordinator and (ii) the exchange of information between the subsystems and the iteration of the subsystem optimizations should be able to terminate before convergence without compromising closed-loop properties. The first criterion is motivated by the practicality of industrial distributed control. Distributed control strategies are used for plants in which centralized control is often impractical or undesirable to implement, and a plantwide coordination layer is likely as difficult to implement as centralized control. The second criterion is motivated by the implementation of distributed control. A plantwide control strategy should be robust to communication disruptions and algorithm failures. Therefore these strategies cannot rely on iteration convergence in order to have an implementable input. In the absence of either of these properties, the alternative is usually centralized control.

In this paper, we extend our previous work to nonlinear plants. The main difference is that the plantwide objective function is nonconvex and therefore we propose a novel distributed nonconvex optimization that converges to stationary points without the use of a central coordinating optimization. The statement of this optimization follows in the next section. In Section 3, we present a distributed model predictive controller that uses the nonconvex optimization and show that this controller is asymptotically stabilizing. We then present an illustrative example and follow with conclusions.

1.1. Notation

Given a vector $x \in \mathbb{R}^n$ the symbol $|x|$ denotes the Euclidean 2-norm; given a positive scalar r the symbol $\mathbb{B}_r$ denotes a closed ball of radius r centered at the origin, i.e., $\mathbb{B}_r = \{x \in \mathbb{R}^n, |x| \le r\}$. Given two integers, $l \le m$, we define the set $\mathbb{I}_{l:m} = \{l, l+1, \ldots, m-1, m\}$. The set of all positive integers is denoted $\mathbb{I}_{\ge 0}$. The set of positive reals is denoted $\mathbb{R}_+$. Given an initial state $x(0)$ and an input sequence $\mathbf{u}$, the open-loop response at time k is $x(k) = \phi(k; x(0), \mathbf{u})$.

2. Distributed nonconvex optimization

Consider the optimization

$$\min_u V(u), \quad \text{s.t. } u \in \mathbb{U} \tag{1}$$

in which $u \in \mathbb{R}^m$ and $V(\cdot): \mathbb{R}^m \to \mathbb{R}_+$ is twice continuously differentiable. We assume $\mathbb{U}$ is closed, convex, and can be separated into M orthogonal subspaces such that $\mathbb{U} = \mathbb{U}_1 \times \cdots \times \mathbb{U}_M$, for which $\mathbb{U}_i \in \mathbb{R}^{m_i}$ for all $i \in \mathbb{I}_{1:M}$. We require approximate solutions to the following suboptimizations at iterate $p \ge 0$ for all $i \in \mathbb{I}_{1:M}$

$$\min_{u_i \in \mathbb{U}_i} V(u_i, u_{-i}^p)$$

in which $u_{-i} = (u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_M)$. Let the approximate solution to these optimizations be $\bar{u}_i^p$. In the proposed algorithm, we compute the approximate solutions via line search with gradient projection. At iterate $p \ge 0$

$$\bar{u}_i^p = \mathcal{P}_i(u_i^p - \nabla_i V(u^p)) \tag{2}$$

in which $\nabla_i V(u^p)$ is the i th component of $\nabla V(u^p)$ and the function $\mathcal{P}_i(\cdot)$ denotes the projection onto the set $\mathbb{U}_i$. Define the step $\upsilon_i^p = \bar{u}_i^p - u_i^p$. To choose the stepsize α_i^p, each suboptimizer initializes the stepsize with $\bar{\alpha}_i$ and then uses backtracking with a factor of $\beta \in (0, 1)$ until α_i^p satisfies the Armijo rule ([1], p. 230)

$$V(u^p) - V(u_i^p + \alpha_i^p \upsilon_i^p, u_{-i}^p) \ge -\sigma \alpha_i^p \nabla_i V(u^p)' \upsilon_i^p \tag{3}$$

in which $\sigma \in (0, 1)$. After all suboptimizers finish the backtracking process, they exchange steps. Each suboptimizer forms a candidate step

$$u_i^{p+1} = u_i^p + w_i \alpha_i^p \upsilon_i^p, \quad \forall i \in \mathbb{I}_{1:M} \tag{4}$$

and checks the following inequality, which tests if $V(u^p)$ is convex-like

$$V(u^{p+1}) \le \sum_{i \in \mathbb{I}_{1:M}} w_i V(u_i^p + \alpha_i^p \upsilon_i^p, u_{-i}^p) \tag{5}$$

in which $\sum_{i \in \mathbb{I}_{1:M}} w_i = 1$ and $w_i > 0$ for all $i \in \mathbb{I}_{1:M}$. If condition (5) is not satisfied, then we find the direction with the worst cost improvement $i_{\max} = \arg\max_i \{V(u_i^p + \alpha_i^p \upsilon_i^p, u_{-i}^p)\}$, and eliminate this direction by setting $w_{i_{\max}}$ to zero and repartitioning the remaining w_i so that they sum to 1. We then reform the candidate step (4) and check condition (5) again. We repeat until (5) is satisfied. At worst, condition (5) is satisfied with one direction only. The steps are formalized in Algorithm 1.

Algorithm 1. Distributed gradient projection

Given finite $\bar{p}$, $0 < \sigma < 1$, and $\bar{w}_i > 0$ for all $i \in \mathbb{I}_{1:M}$ such that $\sum_{i \in \mathbb{I}_{1:M}} \bar{w}_i = 1$.

```
for p = 0, 1, ..., p̄ do
    for i ∈ I_{1:M} do
        Compute ū_i^p using (2);
        Find α_i^p satisfying (3);
        V_i^p ← V(u_i^p + α_i^p υ_i^p, u_{-i}^p);
    end for
    υ^p ← (υ_1^p, ..., υ_M^p);
    k ← 1, I_good ← I_{1:M}, w_i ← w̄_i
    while k < M do
    for i ∈ I_{1:M} do
        u_i^{p+1} ← u_i^p + w_i α_i^p υ_i^p;
    end for
    if u^{p+1} satisfies (5) then
        break;
    else
        i_max ∈ argmax_{i ∈ I_{1:M}} {V_i^p};
        I_good ← I_good \ i_max;
        w_{i_max} ← 0;
        w̄ ← Σ_{j ∈ I_good} w_j;
    for i ∈ I_{1:M} do
        w_i ← w_i / w̄;
    end for
    end if
        k ← k + 1;
    end while
end for
```

Remark 1. In previous work, we proposed a similar distributed algorithm for a convex optimization [17]. The main difference in the nonconvex case is that poor suboptimizer steps must be eliminated to ensure the objective function decreases at each iterate.

Lemma 1 (Feasibility). *Given a feasible initial condition, the iterates u^p are feasible for all $p \ge 0$.*

Lemma 2 (Objective decrease). *The objective function decreases at every iterate, that is, $V(u^{p+1}) \le V(u^p)$.*

Lemma 3 (Convergence). *Every accumulation point of the sequence $\{u^p\}$ is stationary.*

The proofs of Lemmas 1 and 2 follow by construction of the algorithm. A proof of Lemma 3 is provided in Appendix A.

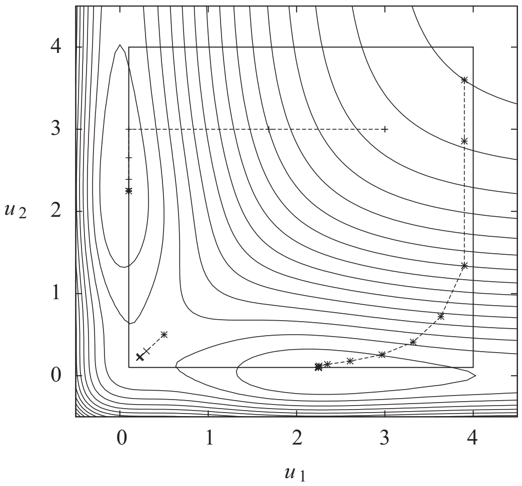


Fig. 1. Nonconvex function presented in [11] optimized with Algorithm 1.

Remark 2 *(Distributed)*. The test of inequality (5) does not need a coordinator. At each optimization iterate the subsystems exchange the solutions of the gradient projection. Each subsystem has a copy of the plantwide model and can evaluate the objection function independently. Therefore the while-loop in Algorithm 1, which is a series of conditional statements without optimization, can be run on each controller. This computation is likely a smaller overhead than a coordinating optimization.

2.1. Example from Rawlings and Mayne

Consider the nonconvex function

$$V(u_1,u_2) = e^{-2u_1} - 2e^{-u_1} + e^{-2u_2} - 2e^{-u_2} + a\exp(-\beta((u_1+0.2)^2 + (u_2+0.2)^2))$$

in which $a = 1.1$ and $\beta = 0.4$ ([11], p.462). There are two global minimum located at (0.007, 2.28) and (2.28, 0,007) and a local minimum at (0.23, 0.23). The inputs are constrained such that $0.1 \le u_i \le 4$ for $i \in \mathbb{I}_{1:2}$. We start the algorithm at three initial conditions (0.5, 0.5), (3.9, 3.6) and (3.5, 3.9). As shown in Fig. 1, each of these points converges to a different local minimum.

3. Distributed nonlinear cooperative control

In this section, we propose a controller that uses the distributed optimization described in the previous section. To facilitate the exposition, we assume the plant comprises only two subsystems.

3.1. Model

We assuming the following models exist

$$x_1^+ = f_1(x_1,x_2,u_1,u_2), \qquad x_2^+ = f_2(x_1,x_2,u_1,u_2) \tag{6}$$

in which $x_i \in \mathbb{R}^{n_i}$, $u_i \in \mathbb{R}^{m_i}$, and $f_i : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \to \mathbb{R}^{n_i}$ is continuous such that $f_i(0) = 0$ for all $i \in \mathbb{I}_{1:2}$. We collect these models to form the plantwide model

$$x^+ = f(x_1,x_2,u_1,u_2) = f(x,u)$$

in which

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \qquad u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \qquad f(x,u) = \begin{bmatrix} f_1(x_1,x_2,u_1,u_2) \\ f_2(x_1,x_2,u_1,u_2) \end{bmatrix}$$

for which $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, and $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$.

3.2. Constraints

At each time step k, we require the inputs to satisfy

$$u_1(k) \in \mathbb{U}_1, \qquad u_2(k) \in \mathbb{U}_2, \quad k \in \mathbb{I}_{0:N-1}$$

in which each $\mathbb{U}_i \in \mathbb{R}^{m_i}$ is compact, convex, and contains the origin in its interior.

3.3. Objective functions

Usually in distributed control implementations an objective function is defined for each subsystem. We construct the plantwide objective function from these objectives. For each subsystem $i \in \mathbb{I}_{1:2}$, we denote the positive definite function $\ell_i(x_i, u_i)$ as the stage cost such that $\ell_i(0, 0) = 0$ and $V_{if}(x)$ as the terminal cost such that $V_{if}(0) = 0$. The objective function for each subsystem $i \in \mathbb{I}_{1:2}$ is defined

$$V_i(x(0),\mathbf{u}_1,\mathbf{u}_2) = \sum_{k=0}^{N-1} \ell_i(x_i(k),u_i(k)) + V_{if}(x(N))$$

in which $\mathbf{u}_i = \{u_i(0), \ldots, u_i(N-1)\} \in \mathbb{R}^{Nm_i}$, $x_i(k) = \phi_i(k; x_i, \mathbf{u}_1, \mathbf{u}_2)$, and $N \ge 1$. Because x_i is a function of both u_1 and u_2, V_i is implicitly a function of both $\mathbf{u}_1$ and $\mathbf{u}_2$. We define the plantwide objective

$$V(x_1(0),x_2(0),\mathbf{u}_1,\mathbf{u}_2) = \rho_1 V_1(x(0),\mathbf{u}_1,\mathbf{u}_2) + \rho_2 V_2(x(0),\mathbf{u}_1,\mathbf{u}_2)$$

in which $\rho_1, \rho_2 > 0$ are weighting factors. To simplify notation we use $V(x, \mathbf{u})$ for the plantwide objective.

Remark 3. Alternatively, the plantwide objective function can be defined without reference to subsystem objective functions.

Assumption 1. For each $i \in \mathbb{I}_{1:2}$, there exists a K_∞ function $\alpha_i(\cdot)$ such that

$$\ell_i(x_i,u_i) \ge \alpha_i(|x_i|), \quad \forall (x_i,u_i) \in \mathbb{R}^{n_i} \times \mathbb{U}_i \tag{7}$$

3.4. Terminal controller

Denote the plantwide terminal penalty $V_f(x) = \rho_1 V_{1f}(x) + \rho_2 V_{2f}(x)$. We define the terminal region $\mathbb{X}_f$ to be a sublevel set of V_f. For $a > 0$, define

$$\mathbb{X}_f = \{x | V_f(x) \le a\}$$

Assumption 2. The plantwide terminal penalty $V_f(\cdot)$ satisfies

$$\alpha_f(|x|) \le V_f(x) \le \gamma_f(|x|), \quad \forall x \in \mathbb{X}_f$$

in which $\alpha_f(\cdot)$ and $\gamma_f(\cdot)$ are K_∞ functions.

Defining $\ell(x, u) = \rho_1 \ell_1(x_1, u_1) + \rho_2 \ell_2(x_2, u_2)$, we require the following stability assumption.

Assumption 3. The terminal cost $V_f(\cdot)$ satisfies

$$\min_{(u_1,u_2)\in\mathbb{U}_1\times\mathbb{U}_2} \left\{ V_f(f(x,u_1,u_2)) + \ell(x,u), \quad \text{s.t.} f(x,u_1,u_2) \in \mathbb{X}_f \right\} \le V_f(x), \quad \forall x \in \mathbb{X}_f$$

This assumption implies that there exists a $\kappa_{if}(x) \in \mathbb{U}_i$ for all $i \in \mathbb{I}_{1:2}$ such that

$$V_f(f(x,\kappa_{1f}(x),\kappa_{2f}(x))) + \ell(x,\kappa_{1f}(x),\kappa_{2f}(x)) \le V_f(x),$$
$$\text{s.t.} f(x,\kappa_{1f}(x),\kappa_{2f}(x)) \in \mathbb{X}_f \tag{8}$$

Each terminal controller $\kappa_{if}(\cdot)$ may be found via a centralized calculation offline. We next provide an example of such a terminal control law.

3.4.1. Distributed terminal control example

In this example, we make a linear approximation of the nonlinear model around the origin and find a stabilizing linear control law. Let $f(\cdot)$ and $\ell(\cdot)$ be Lipschitz continuous in a neighborhood of the origin. Define $A=\nabla_x f(0,0)$, $B=\nabla_u f(0,0)$, $Q=\nabla^2_{xx}\ell(0,0)$, $R=\nabla^2_{uu}\ell(0,0)$, and $S=\nabla^2_{xu}\ell(0,0)$. Denote P_f as the solution to the centralized discrete time Riccati equation

$$P_f = A'P_fA + Q - (A'P_fB+S)(R+B'P_fB)^{-1}(B'P_fA+S')$$

and terminal controller gain K as

$$K = -(R+B'P_fB)^{-1}(B'P_fA+S')$$

In the terminal region, the unconstrained control law $u=Kx$ is used. Defining the stable matrix $A_K=(A+BK)$ and $Q^*=(Q+K'RK)$, let the matrix P satisfy the Lyapunov equation $A_K'PA_K+2Q^*=P$. Following ([11], pp. 135–137), there exists an $a\in(0,\infty)$ such that

$$V_f(f(x,\kappa_{1f}(x),\kappa_{2f}(x)))+\frac{1}{2}x'Q^*x-V_f(x)\le 0, \quad \forall x\in W(a)$$

in which $W(a)=\{x\,|\,V_f(x)\le a\}$

$$\kappa_{1f}(x_1,x_2)=K_{11}x_1+K_{12}x_2$$
$$\kappa_{2f}(x_1,x_2)=K_{21}x_1+K_{22}x_2$$

$$V_{1f}(x_1,x_2)=\frac{1}{2}x_1'P_{11}x_1+\frac{1}{2}x_1'P_{12f}x_2 \tag{9a}$$

$$V_{2f}(x_1,x_2)=\frac{1}{2}x_2'P_{21}x_2+\frac{1}{2}x_2'P_{22}x_2 \tag{9b}$$

and

$$P=\begin{bmatrix}P_{11} & P_{12}\\ P_{21} & P_{22}\end{bmatrix}, \qquad K=\begin{bmatrix}K_{11} & K_{12}\\ K_{21} & K_{22}\end{bmatrix}$$

We then define the terminal set $\mathbb{X}_f=W(a)$.

Remark 4. For systems in which it is undesirable or impossible to calculate the centralized K and P matrices, a decentralized terminal controller can be used with the trade-off that $\mathbb{X}_f$ is smaller.

3.5. Removing the terminal constraint in suboptimal MPC

To show stability, we must ensure that $\phi(N;x,\mathbf{u})\in\mathbb{X}_f$. Imposing a terminal constraint on the state, however, requires the use of coupled input constraints in each suboptimization of cooperative MPC. Such a constraint, in general, does not allow the distributed algorithm to converge to the optimal plantwide control feedback without a coordinator [17]. This terminal constraint can be removed from the control problem by modifying the terminal penalty, however. In the following, we show this feature for the general suboptimal MPC case ([11], pp. 155–158), and note that the proposed distributed controller is of this class.

For some $\beta\ge 1$, let the objective function be defined

$$V^\beta(x,\mathbf{u})=\sum_{k=0}^{N-1}\ell(x(k),u(k))+\beta V_f(x(N)) \tag{10}$$

Define the set of admissible initial $(x,\mathbf{u})$ pairs as

$$\mathbb{Z}_0=\{(x,\mathbf{u})\in\mathbb{X}\times\mathbb{U}^N | V^\beta(x,\mathbf{u})\le\bar{V}, \quad \phi(N;x,\mathbf{u})\in\mathbb{X}_f\} \tag{11}$$

in which $\bar{V}>0$ is an arbitrary constant and $\mathbb{X}=\mathbb{R}^n$. Then the set of initial states $\mathbb{X}_0$ is the projection of $\mathbb{Z}_0$ onto $\mathbb{X}$

$$\mathbb{X}_0=\{x\in\mathbb{X}|\exists\mathbf{u} \quad \text{such that}\,(x,\mathbf{u})\in\mathbb{Z}_0\}$$

Proposition 1 (Terminal constraint satisfaction). *Let $\{(x(k),\mathbf{u}(k))|k\in\mathbb{I}_{\ge0}\}$ denote the set of states and control sequences generated by the suboptimal system. There exists a $\bar{\beta}>1$ such that for all $\beta\ge\bar{\beta}$, if $(x(0),\mathbf{u}(0))\in\mathbb{Z}_0$, then $(x(k),\mathbf{u}(k))\in\mathbb{Z}_0$ with $\phi(N;x(k),\mathbf{u}(k))\in\mathbb{X}_f$ for all $k\in\mathbb{I}_{\ge0}$.*

Proof. The proof is by induction. We show that there is a finite value $\bar{\beta}$ such that the following property holds for all $\beta\ge\bar{\beta}$: For any state and input sequence $(x,u)\in\mathbb{Z}_0$, the successor state and input sequence $(x^+,u^+)\in\mathbb{Z}_0$. The successor state is $x^+=f(x,u(0))$ and the warm start is

$$\tilde{\mathbf{u}}^+=\{u(1),u(2),\ldots,u(N-1),\kappa_f(x(N))\}$$

We know that $\tilde{\mathbf{u}}^+\in\mathbb{U}^N$ because $\kappa(x(N))\in\mathbb{U}$ for $x(N)\in\mathbb{X}_f$. We also have from the properties of $\kappa_f(\cdot)$ that $\phi(N;x^+,\tilde{\mathbf{u}}^+)\in\mathbb{X}_f$ and $V^\beta(x^+,\tilde{\mathbf{u}}^+)\le\bar{V}$ by (8). Next consider any control sequence $\boldsymbol{\upsilon}\in\mathbb{U}^N$ meeting the suboptimal MPC cost requirement

$$V^\beta(x^+,\boldsymbol{\upsilon})\le V^\beta(x^+,\tilde{\mathbf{u}}^+)$$

Expanding the cost function on the left and using the bound on the right gives

$$\sum_{i=0}^{N-1}\ell(z(i),\upsilon(i))+\beta V_f(z(N))\le\bar{V}$$

in which $z(i)=\phi(i;x^+,\boldsymbol{\upsilon})$. This inequality implies

$$\beta V_f(z(N))\le\bar{V}$$

and if we choose

$$\beta\ge\bar{\beta}=\max(1,\bar{V}/a)$$

we obtain $V_f(z(N))\le a$, which implies that $z(N)\in\mathbb{X}_f$. We have found a finite value of $\bar{\beta}$ such that the terminal state corresponding to any admissible $\mathbf{u}^+$ from state x^+ lies in $\mathbb{X}_f$ for $\beta\ge\bar{\beta}$. By induction, since $(x(0),\mathbf{u}(0))\in\mathbb{Z}_0$, $(x(k),\mathbf{u}(k))\in\mathbb{Z}_0$ for all $k\in\mathbb{I}_{\ge0}$, and the result is established. □

For the remainder of the paper, we replace the plantwide objective with the modified objective $V(\cdot)\leftarrow V^{\bar{\beta}}(\cdot)$ and hence the terminal constraint is satisfied.

3.6. Cooperative control algorithm

Let $\tilde{\mathbf{u}}\in\mathbb{U}$ be the initial condition for the cooperative MPC algorithm such that $\phi(N;x(0),\tilde{\mathbf{u}})\in\mathbb{X}_f$. At each iterate p, an approximate solution of the following optimization problem is found

$$\min_{\mathbf{u}} \quad V(x_1(0),x_2(0),\mathbf{u}_1,\mathbf{u}_2) \tag{12a}$$

$$\text{s.t.} \quad x_1^+=f_1(x_1,x_2,u_1,u_2) \tag{12b}$$

$$x_2^+=f_2(x_1,x_2,u_1,u_2) \tag{12c}$$

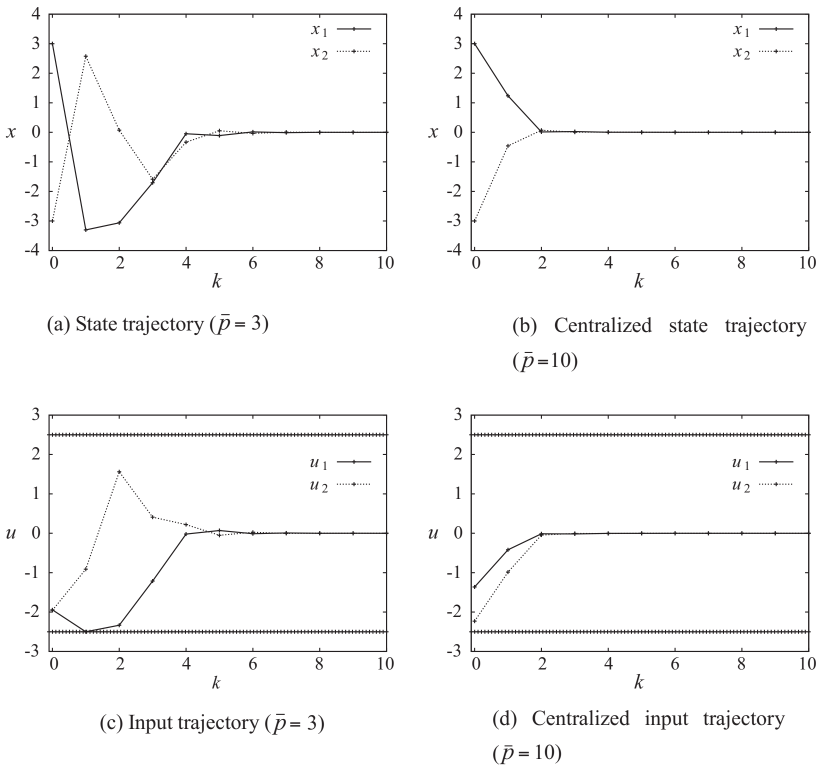
$$\mathbf{u}_i\in\mathbb{U}_i^N, \quad \forall i\in\mathbb{I}_{1:2} \tag{12d}$$

$$|\mathbf{u}_i|\le\delta_i(|x_i(0)|) \quad \text{if}\, x(0)\in\mathbb{B}_r, \quad \forall i\in\mathbb{I}_{1:2} \tag{12e}$$

in which $\delta_i(\cdot)$ is a K_∞ function and $r>0$ can be chosen as small as required. Constraint (12e) is needed for stability and is motivated in the sequel. We can write (12) in the form of (1) by substituting the model equations (12b) and (12c) into the objective function (12a). To achieve distributed control, we use Algorithm 1 to solve (12).

Let the input sequence returned by Algorithm 1 be $\mathbf{u}^{\bar{p}}(x,\tilde{\mathbf{u}})$. The first input of this sequence $\kappa^{\bar{p}}(x(0))=u^{\bar{p}}(0;x(0),\tilde{\mathbf{u}})$ is injected into the plant and the state is moved forward. To reinitialize the algorithm at the next sampling time, we define the warm start

$$\tilde{\mathbf{u}}_1^+=\{u_1(1),u_1(2),\ldots,u_1(N-1),\kappa_{1f}(x(N))\}$$
$$\tilde{\mathbf{u}}_2^+=\{u_2(1),u_2(2),\ldots,u_2(N-1),\kappa_{2f}(x(N))\}$$



(a) State trajectory ($\bar{p} = 3$)

(b) Centralized state trajectory ($\bar{p}=10$)

(c) Input trajectory ($\bar{p} = 3$)

(d) Centralized input trajectory ($\bar{p} = 10$)

Fig. 2. Controller performance with $(x_1(0), x_2(0)) = (3, -3)$. Setting $\bar{p} = 10$ approximates a centralized controller solution. (a) State trajectory ($\bar{p} = 3$); (b) centralized state trajectory ($\bar{p} = 10$); (c) input trajectory ($\bar{p} = 3$); (d) centralized input trajectory ($\bar{p} = 10$).

in which $x(N) = \phi(N; x(0), \mathbf{u}_1, \mathbf{u}_2)$. In general, it is not possible to solve optimization (12) to optimality because of the limited time available between samples. The distributed controller is therefore suboptimal, and the stability of the controller can be established by suboptimal MPC theory.

3.7. *Stability of distributed nonlinear cooperative control*

To establish stability of the control algorithm, we show that the plantwide objective cost decreases between sampling times. Without loss of generality, assume $k = 0$ and the input $u(0)$ is injected into the plant. Using the warm start as the initial condition at the next sampling time, we have

$$\begin{aligned}V(x^+, \tilde{\mathbf{u}}^+) = {} & V(x, \mathbf{u}) - \rho_1 \ell_1(x_1, u_1) - \rho_2 \ell_2(x_2, u_2) - \rho_1 V_{1f}(x(N)) \\ & - \rho_2 V_{2f}(x(N)) + \rho_1 \ell_1(x_1(N), \kappa_{1f}(x(N))) \\ & + \rho_2 \ell_2(x_2(N), \kappa_{2f}(x(N))) \\ & + \rho_1 V_{1f}(f_1(x_1(N), x_2(N), \kappa_{1f}(x(N)), \kappa_{2f}(x(N)))) \\ & + \rho_2 V_{2f}(f_2(x_1(N), x_2(N), \kappa_{1f}(x(N)), \kappa_{2f}(x(N))))\end{aligned}$$

Using (8), the last six terms above are cumulatively nonpositive, giving

$$V(x^+, \tilde{\mathbf{u}}^+) \le V(x, \mathbf{u}) - \rho_1 \ell_1(x_1, u_1) - \rho_2 \ell_2(x_2, u_2)$$

By Lemma 2, the objective function cost decreases from this warm start, so that

$$V(x^+, \mathbf{u}^+) \le V(x, \mathbf{u}) - \rho_1 \ell_1(x_1, u_1) - \rho_2 \ell_2(x_2, u_2)$$

Hence

$$V(x^+, \mathbf{u}^+) - V(x, \mathbf{u}) \le -\alpha(|(x, u)|) \tag{13}$$

in which $\alpha(|(x, u)|) = \rho_1 \alpha_1(|(x_1, u_1)|) + \rho_2 \alpha_2(|(x_2, u_2)|)$.

We now give the main result of the paper. Let $\mathbb{X}_N$ be the forward invariant set of all initial states for which the control optimization (12) is feasible.

Theorem 1 (Asymptotic stability). *Let Assumptions 1–3hold and let $V(\cdot) \leftarrow V^{\bar{\beta}}(\cdot)$ b y Proposition 1. Then for every $x(0) \in \mathbb{X}_N$, the origin is asymptotically stable for the closed-loop system $x^+ = f(x, \kappa^{\bar{p}}(x))$.*

Proof. The proof follows from the stability of suboptimal MPC ([15], Theorem 1), which requires satisfaction of three properties to prove asymptotic stability. (1) There exists a lower bound $\alpha(|x|) \le V(x, \mathbf{u})$ by satisfaction of Assumptions 1 and 2. (2) The descent property has been shown above in (13). (3) The Lyapunov constraint (12e) is explicitly added to the optimization. We have accounted for each required property and the result is established. □

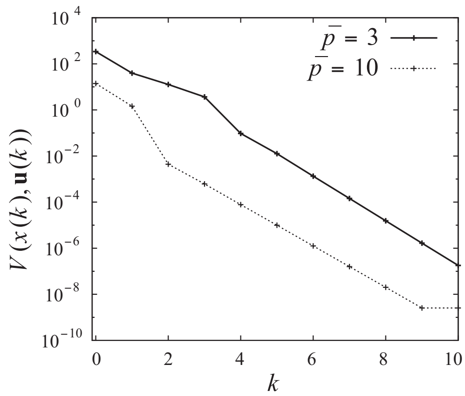


Fig. 3. Open-loop cost to go versus time on the closed-loop trajectory for different numbers of iterations.

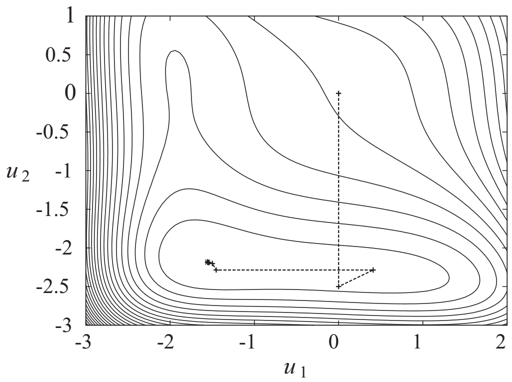


Fig. 4. Contours of V with $N=1$ for $k=0$ with $(x_1(0), x_2(0)) = (3, -3)$. Iterations of the subsystem controllers with initial condition $(u_1^0, u_2^0) = (0, 0)$.

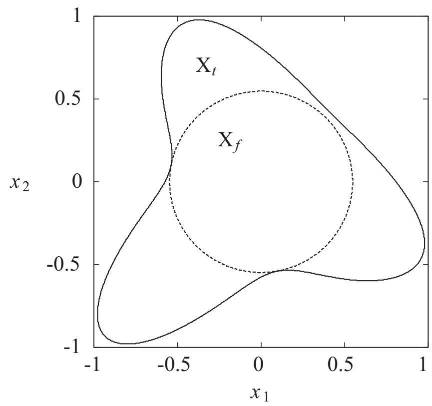


Fig. 5. Terminal region. $\mathbb{X}_t$ are the points in which the terminal controller is stabilizing and $\mathbb{X}_f = \{x \mid V_f(x) \le 0.485\} \subseteq \mathbb{X}_t$ is the terminal region.

Remark 5 *(M subsystems).* The arguments for the controller have been given for the case of two subsystems only, but same arguments apply for any finite $M > 0$ number of subsystems.

4. Illustrative example

For this example, we use the stage cost

$$\ell_1(x_1, u_1) = \frac{1}{2}(x_1' Q_1 x_1 + u_1' R_1 u_1)$$
$$\ell_2(x_2, u_2) = \frac{1}{2}(x_2' Q_2 x_2 + u_2' R_2 u_2)$$

in which $Q_1, Q_2 > 0$ and $R_1, R_2 > 0$. This stage cost gives the objective function

$$V(x, \mathbf{u}) = \frac{1}{2}\sum_{k=0}^{N-1} x(k)' Q x(k) + u(k)' R u(k) + V_f(x(N))$$

in which $Q = \mathrm{diag}(Q_1, Q_2)$, $R = \mathrm{diag}(R_1, R_2)$ and $V_f(\cdot) = V_{1f}(\cdot) + V_{2f}(\cdot)$ is defined by (9). The terminal region is defined as in Section 3.4.1.

4.1. Simulation

Consider the unstable nonlinear system

$$x_1^+ = x_1^2 + x_2 + u_1^3 + u_2$$
$$x_2^+ = x_1 + x_2^2 + u_1 + u_2^3$$

with initial condition $(x_1, x_2) = (3, -3)$. The control objective is to stabilize the system and drive the states to the origin. For the simulation we choose the parameters

$$Q = I, \quad R = I, \quad N = 2, \quad \bar{p} = 3, \quad \mathbb{U}_i = [-2.5, 2.5], \quad \forall i \in \mathbb{I}_{1:2}$$

As shown in Fig. 2, the control scheme is stabilizing. Increasing the maximum number of iterations significantly improves the performance. In Fig. 2, we also show the performance for $\bar{p} = 10$. The cost difference is given in Fig. 3. To elucidate the difficulty in optimizing the nonconvex objective function, the iterations of the zeroth stage control optimization are shown in Fig. 4 for the $N = 1$ case. The terminal region, calculated as in Section 3.4.1, is shown in Fig. 5.

5. Conclusions

In this paper, we present a novel nonlinear controller that solves the control optimization via a distributed optimization. This optimization uses parallel optimizations that correspond to subsystems in a plant. Each iterate is feasible and decreases the objective function, and the iterates converge to stationary points of the plantwide objective function. A unique feature of the optimization is that no coordinating optimization is required. We also show how to design the controller and the terminal controller. Asymptotic stability is established, and an illustrative example is presented showing the stabilizing properties of the controller.

Acknowledgments

The authors gratefully acknowledge the financial support of the industrial members of the Texas-Wisconsin-California Control Consortium. The authors would like to thank Prof. D.Q. Mayne for helpful discussion of Proposition 1.

Appendix A. Proof of convergence

We begin with preliminary propositions, and then give the main proof.

Proposition 2. *Given a closed, convex set $\mathbb{U}$ with any $y \in \mathbb{U}$ and any $z \in \mathbb{R}^m$, for the projection $\mathcal{P}(\cdot)$ onto $\mathbb{U}$*

$$(y - z)'(y - \mathcal{P}(z)) \ge 0$$

with equality if and only if $y = \mathcal{P}(z)$.

Proposition 3. *If $\hat{u}$ is nonstationary*

$$\nabla_i V(\hat{u})'[\hat{u}_i - \mathcal{P}_i(\hat{u}_i - \nabla_i V(\hat{u}))] \ge 0, \quad i \in \mathbb{I}_{1:M}$$

with strict inequality for at least one $i \in \mathbb{I}_{1:M}$.

Proof. Set $y = \hat{u}_i$ and $z = \hat{u}_i - \nabla_i V(\hat{u})$ in Proposition 2 to prove the first claim. To show the second claim, observe that if equality holds for all $i \in \mathbb{I}_{1:M}$, then from Proposition 2 we would have

$$\hat{u}_i = \mathcal{P}_i(\hat{u}_i - \nabla_i V(\hat{u})), \quad \forall i \in \mathbb{I}_{1:M}$$

and therefore $\hat{u} = \mathcal{P}(\hat{u} - \nabla V(\hat{u}))$, and $\hat{u}$ would be stationary. □

Proposition 4. *Suppose $\hat{u}$ is a nonstationary point. Then there are positive constants ρ and ϵ and an index $i \in \mathbb{I}_{1:M}$ such that for all u with $|u - \hat{u}| \le \rho$, the ith suboptimizer chooses stepsize α_i for which*

$$V(u_i, u_{-i}) - V(u_i + \alpha_i \upsilon_i, u_{-i}) \ge \epsilon$$

Proof. Let i be an index such that strict inequality holds in Proposition 3. Using the continuity of $\nabla_i V(\cdot)$ and $\mathcal{P}_i(\cdot)$, define $\rho > 0$ and $\epsilon_i > 0$ such that

$$-\nabla_i V(u)' \upsilon_i = \nabla_i V(u)'[u_i - \mathcal{P}_i(u_i - \nabla_i V(u))] \ge \epsilon_i$$

for all u with $|u - \hat{u}| \leq \rho$. From Taylor's theorem ([9], p. 14), and using continuity of $\nabla_i V(\cdot)$, there is an $\hat{\alpha}_i > 0$ such that for all $\alpha_i \in [0, \hat{\alpha}_i]$

$$\begin{aligned} V(u_i, u_{-i}) - V(u_i + \alpha_i \upsilon_i, u_{-i}) \\ = -\alpha_i \nabla_i V(u_i, u_{-i})' \upsilon_i - \alpha_i [\nabla_i V(u_i + t\alpha_i \upsilon_i, u_{-i}) - \nabla_i V(u_i, u_{-i})]' \upsilon_i \\ = -\alpha_i \nabla_i V(u_i, u_{-i})' \upsilon_i + o(\alpha_i) \geq -\sigma \alpha_i \nabla_i V(u_i, u_{-i})' \upsilon_i \end{aligned} \tag{14}$$

in which $\hat{\alpha}_i$ is small enough to ensure that the remainder term satisfies $o(\alpha_i) \leq -(1-\sigma)\alpha_i \nabla_i V(u_i, u_{-i})' \upsilon_i$, a strictly positive multiple of α_i. Hence the backtracking process terminates at a value α_i greater than or equal to $\underline{\alpha}_i = \min(\bar{\alpha}_i, \beta\hat{\alpha}_i) > 0$. Hence, from (14), we have

$$V(u_i, u_{-i}) - V(u_i + \alpha_i \upsilon_i, u_{-i}) \geq -\sigma \underline{\alpha}_i \nabla_i V(u_i, u_{-i})' \upsilon_i \geq \sigma \underline{\alpha}_i \epsilon_i > 0$$

Therefore the Proposition holds with $\epsilon = \sigma \underline{\alpha}_i \epsilon_i$. □

We now proceed with the proof of the convergence result.

Proof of Lemma 3. Toward a contradiction, suppose that û is a nonstationary point, and let K be a subsequence such that $\{u^p\}_{p \in K} \to \hat{u}$. By taking a further subsequence if necessary, we have from Proposition 4 that there is an index i and a positive constant ϵ such that

$$V(u_i^p, u_{-i}^p) - V(u_i^p + \alpha_i^p \upsilon_i^p, u_{-i}^p) \geq \epsilon$$

for all $p \in K$. Let j^p be the index in $\mathbb{I}_{1:M}$ that attains the *best* decrease on V at iterate p. Since there are only finitely many possible values for j^p, at least one of them must recur infinitely often. By taking a further subsequence we can assure $j^p \equiv j$ for some $j \in \mathbb{I}_{1:M}$. We thus have

$$V(u^p) - V(u_j^p + \alpha_j^p \upsilon_j^p, u_{-j}^p) \geq V(u^p) - V(u_i + \alpha_i^p \upsilon_i^p, u_{-i}^p) \geq \epsilon \tag{15}$$

for all $p \in K$. Moreover, the index j remains in the set $\mathbb{I}_{\text{good}}$ for all inner iterations, at each major iteration $p \in K$. Since all terms in the summation on the right-hand side of (5) are nonnegative and $w_j \geq \bar{w}_j > 0$, using (15), the right-hand side is bounded below by $\bar{w}_j \epsilon > 0$. Therefore

$$V(u^p) - V(u^{p+1}) \geq \bar{w}_j \epsilon > 0, \quad \forall p \in K$$

for which $\bar{w}_j \epsilon$ does not depend on p. This inequality implies that $V(u^p) \to -\infty$ over the *entire* sequence $\{u^p\}$, since $V(u^p)$ decreases at every iteration. This contradicts $\lim_{p \in K} V(u^p) = V(\hat{u})$, and the proof is complete. □

References

[1] D.P. Bertsekas, Nonlinear Programming, second edition, Athena Scientific, Belmont, MA, 1999.

[2] R. Cheng, J.F. Forbes, W.S. Yip, Price-driven coordination method for solving plant-wide MPC problems, J. Process Cont. 17 (5) (2007) 429–438.

[3] H. Cui, E.W. Jacobsen, Performance limitations in decentralized control, J. Process Cont. 12 (2002) 485–494.

[4] J. Liu, D. Muñoz de la Peña, P.D. Christofides, Distributed model predictive control of nonlinear process systems, AIChE J. 55 (5) (2009) 1171–1184.

[5] J. Liu, X. Chen, D. Muñoz de la Peña, P.D. Christofides, Sequential and iterative architectures for distributed model predictive control of nonlinear process systems, AIChE J. 56 (5) (2010) 2137–2149.

[6] J.M. Maestre, D. Muñoz de la Peña, E.F. Camacho, Distributed model predictive control based on a cooperative game, Optimal Cont. Appl. Meth., n/a. doi:10.1002/oca.940.

[7] D.Q. Mayne, J.B. Rawlings, C.V. Rao, P.O.M. Scokaert, Constrained model predictive control: stability and optimality, Automatica 36 (6) (2000) 789–814.

[8] I. Necoara, C. Savorgnan, Q.T. Dinh, J. Suykens, M. Diehl, Distributed nonlinear optimal control using sequential convex programming and smoothing techniques, in: Proceedings of the Conference on Decision and Control, Shanghai, China, December, 2009.

[9] J. Nocedal, S.J. Wright, Numerical Optimization, second edition, Springer, New York, 2006.

[10] S.J. Qin, T.A. Badgwell, A survey of industrial model predictive control technology, Control Eng. Pract. 11 (7) (2003) 733–764.

[11] J.B. Rawlings, D.Q. Mayne, Model Predictive Control: Theory and Design, Nob Hill Publishing, Madison, WI, 2009, ISBN 978-0-9759377-0-9, 576 pp.

[12] J.B. Rawlings, B.T. Stewart, Coordinating multiple optimization-based controllers: new opportunities and challenges, J. Process Cont. 18 (2008) 839–845.

[13] N.R. Sandell, P. Varaiya, M. Athans, M. Safonov Jr., Survey of decentralized control methods for large scale systems, IEEE Trans. Autom. Cont. 23 (2) (1978) 108–128.

[14] R. Scattolini, Architectures for distributed and hierarchical model predictive control—a review, J. Process Cont. 19 (May 5) (2009) 723–731, ISSN 0959-1524.

[15] P.O.M. Scokaert, D.Q. Mayne, J.B. Rawlings, Suboptimal model predictive control (feasibility implies stability), IEEE Trans. Autom. Cont. 44 (March 3) (1999) 648–654.

[16] D.D. Šiljak, Decentralized Control of Complex Systems, Academic Press, London, 1991, ISBN 0-12-643430-1.

[17] B.T. Stewart, A.N. Venkat, J.B. Rawlings, S.J. Wright, G. Pannocchia, Cooperative distributed model predictive control, Syst. Cont. Lett. 59 (2010) 460–469.

[18] A.N. Venkat, I.A. Hiskens, J.B. Rawlings, S.J. Wright, Distributed MPC strategies with application to power system automatic generation control, IEEE Cont. Syst. Technol. 16 (November 6) (2008) 1192–1206.