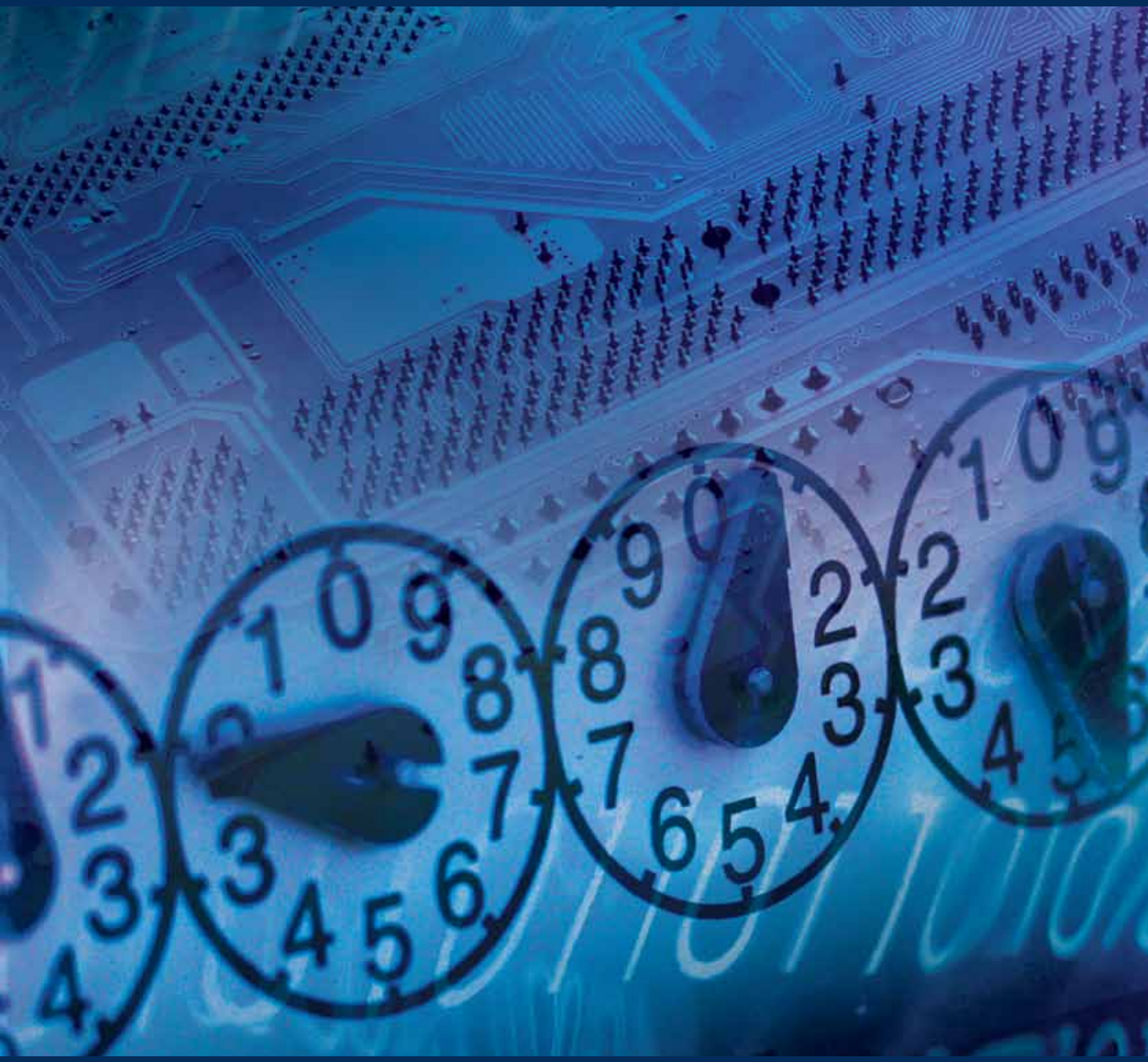
Journal of Control Science and Engineering

Model Predictive Control

Guest Editors: Baocang Ding, Marcin T. Cychowski, Yugeng Xi, Wenjian Cai, and Biao Huang



Model Predictive Control

Journal of Control Science and Engineering

Model Predictive Control

Guest Editors: Baocang Ding, Marcin T. Cychowski, Yugeng Xi, Wenjian Cai, and Biao Huang

Editorial Board

Contents

Hindawi Publishing Corporation
Journal of Control Science and Engineering
Volume 2012, Article ID 240898, 2 pages
doi:10.1155/2012/240898

Editorial
Model Predictive Control

Baocang Ding,[1] Marcin T. Cychowski,[2] Yugeng Xi,[3] Wenjian Cai,[4] and Biao Huang[5]

[1] *Ministry of Education Key Lab For Intelligent Networks and Network Security (MOE KLINNS Lab), Department of Automation, School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China*
[2] *NIMBUS Centre for Embedded Systems Research, Cork Institute of Technology, Rossa Avenue, Cork, Ireland*
[3] *Department of Automation, Shanghai Jiaotong University, Shanghai 200240, China*
[4] *School of Electronic and Electrical Engineering, Nanyang Technological University, BLK S2, Singapore 639798*
[5] *Department of Chemical and Materials Engineering, University of Alberta, Edmonton, AB, Canada T6G 2G6*

Correspondence should be addressed to Baocang Ding, baocang.ding@gmail.com

Received 9 February 2012; Accepted 9 February 2012

Model predictive control (MPC) has been a leading technology in the field of advanced process control for over 30 years. At each sampling time, MPC optimizes a performance cost satisfying the physical constraints, which is initialized by the real measurements, to obtain a sequence of control moves or control laws. However, MPC only implements the control move corresponding to the current sampling time. At the next sampling time, the same optimization problem is solved with renewed initialization. In order to obtain offset-free control, the system model is augmented with a disturbance model to capture the effect of model-pant mismatch. The state and disturbance estimates are utilized to initialize the optimization problem. A steady-state target calculation unit is constructed to compromise between what is desired by real-time optimization (RTO) and what is feasible for the dynamic control move optimization. As a result, MPC exhibits innumerous research issues from both theoretical and practical aspects. Nowadays, there is big gap between the theoretical investigations and industrial applications. The main focus of this special issue is on the new research ideas and results for MPC.

A total number of 14 papers were submitted for this special issue. Out of the submitted papers, 6 contributions have been included in this special issue. The 6 papers consider 6 rather different, yet interesting topics.

A synthesis approach of MPC is the one with guaranteed stability. The technique of linear matrix inequality (LMI) is often used in the synthesis approach to address the satisfaction of physical constraints. However, the existing LMI formulations are for symmetric constraints. M. S. M. Cavalca et al. treat asymmetric output constraints in integrating SISO systems based on pseudoreferences.

Usually, the steady-state target calculation, followed with the dynamic move calculation, is implemented in each sampling time. This is not practical for fast-sampling systems. Y. C. Chu and M. Z. Q. Chen propose a method to tackle this issue. In their method, the steady-state target calculation works in parallel to, with a period longer than and a scale of optimization larger than, the dynamic move calculation. It is shown that their method is particularly suitable for tracking the periodic references.

The nonlinear system represented by a Hammerstein model has always been a good platform for control algorithm research. D. F. He and L. Yu revisit this topic by invoking the pole-placement method on the linear subsystem. They propose the algorithm which consists of three online steps, instead of the two-step MPC. They also propose to use their algorithm in the grade transition control of industrial polypropylene plants, via a simulation study.

Usually, MPC has its own paradigm for robust control. However, combing MPC with $\mathcal{H}_2/\mathcal{H}_\infty$ control could be a good topic for improving the robustness of MPC. P. E. Orukpe applies the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ method in MPC where the system uncertainties are modeled by perturbations in a linear fractional transform (LFT) representation and unknown bounded disturbances.

Interests in the cooperative control of multi-agent systems have been growing significantly over the last years. MPC has the ability to redefine cost functions and constraints as needed to reflect changes in the environment, which makes

it a nice choice for multiagent systems. S. B. Wei et al. give a method for distributed MPC which punishes, in the cost function, the deviation between what an agent optimizes and what other related agents think of it. The deviation weight matrix at the end of the control horizon is specially discussed for improving the control performance.

Besides, C. H. F. Silva et al. report some experimental studies for two classical algorithms: infinite horizon MPC and MPC with reference system. The pilot plant is for level and pH control, which has physical constraints and nonlinear dynamics.

We hope the readers of *Journal of Control Science and Engineering* will find the special issue interesting and stimulating, and expect that the included papers contribute to further advance the area of MPC.

Acknowledgments

We would like to thank all the authors who have submitted papers to the special issue and the reviewers involved in the refereeing of the submissions.

Baocang Ding
Marcin T. Cychowski
Yugeng Xi
Wenjian Cai
Biao Huang

Hindawi Publishing Corporation
Journal of Control Science and Engineering
Volume 2012, Article ID 313716, 8 pages
doi:10.1155/2012/313716

Research Article

Distributed Model Predictive Control of the Multi-Agent Systems with Improving Control Performance

Wei Shanbi, Chai Yi, and Li Penghua

College of Automation, Chongqing University, Chongqing 400044, China

Correspondence should be addressed to Wei Shanbi, wsbmei@yahoo.com.cn

Received 8 October 2011; Revised 22 December 2011; Accepted 19 January 2012

Academic Editor: Yugeng Xi

This paper addresses a distributed model predictive control (DMPC) scheme for multiagent systems with improving control performance. In order to penalize the deviation of the computed state trajectory from the assumed state trajectory, the deviation punishment is involved in the local cost function of each agent. The closed-loop stability is guaranteed with a large weight for deviation punishment. However, this large weight leads to much loss of control performance. Hence, the time-varying compatibility constraints of each agent are designed to balance the closed-loop stability and the control performance, so that the closed-loop stability is achieved with a small weight for the deviation punishment. A numerical example is given to illustrate the effectiveness of the proposed scheme.

1. Introduction

Interests in the cooperative control of multiagent systems have been growing significantly over the last years. The main motivation is the wide range of military and civilian applications, including formation flight of UAV and automated traffic systems. Compared with the traditional approach, model predictive control (MPC), or receding horizon control (RHC) has the ability to redefine cost functions and constraints as needed to reflect changes in the system and/or the environment. Therefore, MPC is extensively applied to the cooperative control of multiagent systems, which makes the agents operate close to the constraint boundaries and obtain better performance than traditional approaches [1–3]. Moreover, due to the computational advantages and the convenience of communication, distributed MPC (DMPC) is recognized as a nature technique to address trajectory optimization problems for multiagent systems.

One of the challenges for distributed control is to ensure that local control actions keep consistent with the actions of others agents [4, 5]. For the coupled systems, the local optimization problem is solved based on the states of its neighbors' at sample time instant using Nash-optimization technique in [6]. As the local controllers lack of communication and cooperation, the local control actions cannot keep consistent [7, 8]. require each local controller exchange information with all other local controllers to improve optimality and consistency based on sufficient communication. For the decoupled systems [9], exploits the estimation of the prediction state trajectories of the neighbors' [10]; treats the prediction state trajectories of the neighbor agents as bounded disturbance where a min-max optimal problem is solved for each agent with respect to the worst-case disturbance. In [11, 12], the optimal variables of the local optimization problem contain the control action of its own and its neighbors' which are coupled in collision avoidance constraints and cost function. Obviously, the deviation between the actions of what the agent is actually doing and of what its neighbor estimates for it affects the control performance. Sometimes the consistency and collision avoidance cannot be achieved, and the feasibility and stability of this scheme cannot be guaranteed [13]. proposes a distributed MPC with a fixed compatibility constraint to restrict the deviation. When the bound of this constraint is sufficiently small, the closed-loop system state enter a neighborhood of the objective state [14, 15] give an improvement over [13] by adding *deviation punishment term* to penalize the deviation of the computed state trajectory

from the assumed state trajectory. Closed-loop exponential stability follows if the weight on the deviation function term is large enough. But the large weight leads to the loss of the control performance.

A contribution in this paper is to propose an idea to reduce the adverse effect of the deviation punishment on the control performance. At each sample time, the value of compatibility constraint is set as the maximum value of the deviation of the previous sample time. We give the stability condition to guarantee the exponential stability of the global closed-loop system with a small weight on the deviation punishment term, which is obtained by dividing the centralize stability constraint as the manner of [16, 17]. The effectiveness of the scheme is also demonstrated by a numerical example.

Notations. x_k^i is the value of vector x^i at time $k \cdot x_{k,t}^i$ is the value of vector x^i at a future time $k+t$, predicted at time $k \cdot |x| = [|x_1|, |x_2|, \ldots, |x_N|]$ is the absolute value for each component of x. For a vector x and positive-definite matrix Q, $\|x\|_Q^2 = x^T Q x$.

2. Problem Statement

Let us consider a system which is composed of N_a agents. The dynamics of agent i [11] is

$$x_{k+1}^i = f^i\left(x_k^i, u_k^i\right), \tag{1}$$

where $u_k^i \in \mathbb{R}^{m_i}$, $x_k^i \in \mathbb{R}^{n_i}$, and $f^i : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \mapsto \mathbb{R}^{n_i}$, are the input, state, and state transition function of agent i, respectively. $u_k^i = [u_k^{i,1}, \ldots, u_k^{i,m_i}]^{\mathrm{T}}$, $x_k^i = [x_k^{i,1}, \ldots, x_k^{i,n_i}]^{\mathrm{T}}$. The sets of feasible input and state of agent i are denoted as $\mathcal{U}^i \subset \mathbb{R}^{m_i}$ and $\mathcal{X}^i \subset \mathbb{R}^{n_i}$, respectively, that is,

$$u_k^i \in \mathcal{U}^i, x_k^i \in \mathcal{X}^i, \quad k \geq 0. \tag{2}$$

At each time k, the control objective is [18] to minimize

$$J_k = \sum_{t=0}^{\infty} \left[\|x_{k,t}\|_Q^2 + \|u_{k,t}\|_R^2 \right] \tag{3}$$

with respect to $u_{k,t}$, $t \geq 0$, where $x = [(x^1)^{\mathrm{T}}, \ldots, (x^{N_a})^{\mathrm{T}}]^{\mathrm{T}}$, $u = [(u^1)^{\mathrm{T}}, \ldots, (u^{N_a})^{\mathrm{T}}]^{\mathrm{T}}$; $x_{k,t+1}^i = f^i(x_{k,t}^i, u_{k,t}^i)$, $x_{k,0}^i = x_k^i$; $Q = Q^{\mathrm{T}} > 0$, $R = R^{\mathrm{T}} > 0$. $u \in \mathbb{R}^m$, $m = \sum_i m_i$, and $x \in \mathbb{R}^n$, $n = \sum_i n_i$. Then,

$$x_{k+1} = f(x_k, u_k), \tag{4}$$

where $f = [f^1, f^2, \ldots, f^{N_a}]^{\mathrm{T}}$, $f : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$. (x_e^i, u_e^i) is the equilibrium point of agent i, and (x_e, u_e) is the corresponding equilibrium point of all agents. $\mathcal{X} = \mathcal{X}^1 \times \mathcal{X}^2 \times \cdots \times \mathcal{X}^{N_a}$, $\mathcal{U} = \mathcal{U}^1 \times \mathcal{U}^2 \times \cdots \times \mathcal{U}^{N_a}$. The models for all agents are completely decoupled. The coupling between agents arises due to the fact that they operate in the same environment, and that the "cooperative" objective is imposed on each agent by the cost function. Hence, there are the coupling cost function and coupling constraints [19]. The coupling constraints can be transformed to coupling cost function term directly or handled as decoupling constraints using the technique of [15]. In the present paper we will not consider this issue.

The *control objective* for all system is to cooperatively asymptotically stabilize all agents to an equilibrium point (x_e, u_e) of (4). In this paper we assumed that the $(x_e, u_e) = (0, 0)$, $f(x_e, u_e) = 0$. The corresponding equilibrium point for each agent is $(x_e^i, u_e^i) = (0, 0)$, $f^i(x_e^i, u_e^i) = 0$. Assumption $f^i(0, 0) = 0$ is not restrictive, since if $(x_e^i, u_e^i) \neq (0, 0)$, one can always shift the origin of the system to it.

The resultant control law for minimization of (3) can be implemented in a centralized way. However, the existing methods for centralized MPC are only computationally tractable for small-scale system. Furthermore, the communication cost of implementing a centralized receding horizon control law may be costly. Hence, by means of decomposition, J_k is divided as J_k^i's such that the minimization of (3) is implemented in distributed manner, with

$$J_k^i = \sum_{t=0}^{\infty} \left[\|z_{k,t}^i\|_{\overline{Q}_i}^2 + \|u_{k,t}^i\|_{\overline{R}_i}^2 \right], \qquad J_k = \sum_{i=1}^{N_a} J_k^i, \tag{5}$$

where $z_{k,t}^i = [(x_{k,t}^i)^{\mathrm{T}} (x_{k,t}^{-i})^{\mathrm{T}}]^{\mathrm{T}}$; $x_{k,t}^{-i}$ includes the states of the neighbors. The set of neighbors' of agent i is denoted as $\mathcal{N}_i$. $x_k^{-i} = \{x_k^j \mid j \in \mathcal{N}_i\}$, $x_k^{-i} \in \mathbb{R}^{n^{-i}}$, $n^{-i} = \sum_{j \in \mathcal{N}_i} n^j$. For each agent i, the control objective is to stabilize it to the equilibrium point $(x_e^i, u_e^i) \cdot \overline{Q}_i = \overline{Q}_i^{\mathrm{T}} > 0$, $\overline{R}_i = \overline{R}_i^{\mathrm{T}} > 0$. $\overline{Q}_i$ is obtained by dividing Q using the technique of [19]. For the agents that have decoupled dynamics, the couplings of control moves for all system are not considered. R is a diagonal matrix and $\overline{R}_i$ is directly obtained.

Under the networked environment, the bandwidth limitation can restrict the amount of information exchange [17]. It is thus appropriate to allow agents to exchange information only once in each sampling interval. We assume that the connectivity of the interagent communication network is sufficient for agents to obtain information regarding all the variables that appear in their local problems.

In the receding horizon control manner, a finite-horizon cost function is exploited to approximate J_k^i. According to the (5), the evolution of the control moves with predictive horizon for agent i is based on the estimation of the state trajectories $x_{k,t}^{-i}, t \leq N$ of the neighbors', which are substituted by the assumed state trajectories $\hat{x}_{k,t}^{-i}, t \leq N$ as [11]. In each control interval, the transmitted information between agents is the assumed state trajectories. As the cooperative consistency and efficiency of distributed control moves is affected for the existence of the deviation of the computed state trajectory from the assumed state trajectory, it is appreciate to penalize it by adding the deviation punishment term into the local cost function.

Define

$$u_{k,t}^i = F_i(k) x_{k,t}^i, \qquad \forall t \geq N. \tag{6}$$

$F_i(k)$ is the gain of distributed state feedback controller.

Consider

$$\breve{J}_k^i = \sum_{t=0}^{N-1}\left[\|\hat{z}_{k,t}^i\|_{\overline{Q}_i}^2 + \|u_{k,t}^i\|_{\overline{R}_i}^2 + \|x_{k,t}^i - \hat{x}_{k,t}^i\|_{\overline{T}_i}^2\right] + \sum_{t=N}^{\infty}\left[\|x_{k,t}^i\|_{Q_i}^2 + \|u_{k,t}^i\|_{R_i}^2\right], \tag{7}$$

where

$$\hat{z}_{k,t}^i = \left[\left(x_{k,t}^i\right)^{\mathrm{T}}\left(\hat{x}_{k,t}^{-i}\right)^{\mathrm{T}}\right]^{\mathrm{T}}, \quad \hat{x}_{k,0}^i = x_k^i, \tag{8}$$

$\hat{x}_{k,t}^{-i}$ includes the assumed states of the neighbors. $Q_i = Q_i^{\mathrm{T}} > 0$ and $R_i = R_i^{\mathrm{T}} = \overline{R}_i$ satisfy

$$\mathrm{diag}\{Q_1, Q_2, \ldots, Q_{N_a}\} \geq Q, \qquad \mathrm{diag}\{R_1, R_2, \ldots, R_{N_a}\} = R. \tag{9}$$

Obviously, Q_i is designed to stabilize the agent i to the local equilibrium point, independently. $\overline{Q}_i$ is designed to stabilize the agent i to the local equilibrium point with neighbor agents, cooperatively. $\overline{T}_i$ is the weight on the deviation punishment term, to penalize the deviation of the computed state trajectory from the assumed state trajectory.

At each time k, the optimization problem for distributed MPC is transformed as:

$$\min_{\overline{U}_k^i, F_i(k)} \breve{J}_k^i, s.t. (1), (2), (6), (7). \tag{10}$$

$\overline{U}_k^{*i} = [(u_{k,0}^{*i})^{\mathrm{T}}, (u_{k,1}^{*i})^{\mathrm{T}}, \ldots, (u_{k,N-1}^{*i})^{\mathrm{T}}]^{\mathrm{T}}$, only when $u_k^{*i} = u_{k,0}^{*i}$ is implemented, and the problem (9) is solved again at time $k+1$.

Remark 1. The local deviate punishment by each agent effects the control performance, that is, incurs the loss of optimality.

3. Stability of Distributed MPC

The stability of distributed MPC by simply applying the procedure as in the centralized MPC will be discussed. The compact and convex terminal set Ω^i is defined

$$\Omega^i = \left\{x^i \in \mathbb{R}^{n_i} \mid \left(x^i\right)^{\mathrm{T}} P_i x^i \leq \alpha^i\right\}, \tag{11}$$

where $P_i > 0$, $\alpha^i > 0$ are specified such that Ω^i is a control invariant set. So using the idea of [20, 21], one simultaneously determines a linear feedback such that Ω^i is a positively invariant under this feedback.

Define the local linearization at the equilibrium point

$$A_i = \frac{\partial f^i}{\partial x^i}(0,0), \qquad B_i = \frac{\partial f^i}{\partial u^i}(0,0). \tag{12}$$

and assume that (A_i, B_i) is stabilizable. When $x_{k,N+t}^i$, $t \geq 0$ enters into the terminal set Ω^i, the local linear feedback control law is assumed as $u_{k,N+t}^i = F_i(k)x_{k,N+t}^i = K_i x_{k,N+t}^i$. K_i is a constant which is calculated off line as follows.

3.1. Design of the Local Control Law. The following equation follows for achieving closed-loop stability:

$$\left\|x_{k,N+t+1}^i\right\|_{P_i}^2 - \left\|x_{k,N+t}^i\right\|_{P_i}^2 \leq -\left\|x_{k,N+t}^i\right\|_{Q_i}^2 - \left\|u_{k,N+t}^i\right\|_{R_i}^2, \quad t \geq 0. \tag{13}$$

Lemma 1. *Suppose that there exist $Q_i > 0$, $R_i > 0$, $P_i > 0$, which satisfy the Lyapunov-equation:*

$$(A_i + B_i K_i)^T P_i (A_i + B_i K_i) - P_i = -\kappa_i P_i - Q_i - K_i^T R_i K_i, \tag{14}$$

for some $\kappa_i > 0$. Then, there exists a constant $\alpha^i > 0$ such that Ω_i defined in (11) satisfies (13).

Remark 2. Lemma 1 is directly obtained by referring to "*Lemma* 1" in [21]. For MPC, the stability margin can be adjusted by turning the value of κ_i according to Lemma 1. With regard to DMPC, [11] adjusts the stability margin by tuning the weight in the local cost function. The control objective is to asymptotically stabilize the closed-loop system, so that $x_{k,\infty}^i = 0$ and $u_{k,\infty}^i = 0$. For $t = 0, \ldots, \infty$, summing (13) obtains

$$\sum_{t=N}^{\infty}\left[\left\|x_{k,t}^i\right\|_{Q_i}^2 + \left\|u_{k,t}^i\right\|_{R_i}^2\right] \leq \left\|x_{k,N}^i\right\|_{P_i}^2. \tag{15}$$

Considering both (7) and (15), yields

$$\breve{J}_k^i \leq \bar{J}_k^i = \sum_{t=0}^{N-1}\left[\left\|\hat{z}_{k,t}^i\right\|_{\overline{Q}_i}^2 + \left\|u_{k,t}^i\right\|_{\overline{R}_i}^2 + \left\|x_{k,t}^i - \hat{x}_{k,t}^i\right\|_{\overline{T}_i}^2\right] + \left\|x_{k,N}^i\right\|_{P_i}^2, \tag{16}$$

where $\bar{J}_k^i$ is a finite-horizon cost function, which consists of a finite horizon standard cost, to specify the desired control performance and a terminal cost, to penalize the states at the end of the finite horizon.

The terminal region Ω^i for agent i is designed, so that it is invariant for nonlinear system controlled by a local linear state feedback. The quadratic terminal cost $\|x_{k,N}^i\|_{P_i}^2$ bounds the infinite horizon cost of the nonlinear system starting from Ω^i and controlled by the local linear state feedback.

3.2. Compatibility Constraint for Stability. As in [18], we define two terms, $\xi^{-i} = x^{-*i} - \hat{x}^{-i}$, $\xi^i = x^{*i} - \hat{x}^i$,

$$\overline{Q}_i = \begin{bmatrix} \overline{Q}_i^1 & \overline{Q}_i^{12} \\ \left(\overline{Q}_i^{12}\right)^{\mathrm{T}} & \overline{Q}_i^3 \end{bmatrix},$$

$$C_x^*(k) = \sum_{i=1}^{N_a}\sum_{t=1}^{N-1}\left\{2\left(x_{k,t}^{*i}\right)^{\mathrm{T}}\overline{Q}_i^{12}\xi_{k,t}^{-i} + 2\left(\hat{x}_{k,t}^{-i}\right)^{\mathrm{T}}\overline{Q}_i^3\xi_{k,t}^{-i} + \left(\xi_{k,t}^{-i}\right)^{\mathrm{T}}\overline{Q}_i^3\xi_{k,t}^{-i}\right\},$$

$$C_\xi^*(k) = \sum_{i=1}^{N_a}\sum_{t=1}^{N-1}\left(\xi_{k,t}^i\right)^{\mathrm{T}}\overline{T}_i\xi_{k,t}^i, \tag{17}$$

Lemma 2. *Suppose that* (9) *holds and there exits* $\rho(k)$ *such that, for all* $k > 0$,

$$\begin{aligned} &0 \le \rho(k) \le 1, \\ &-\rho(k)\sum_{i=1}^{N_a}\left\{\left\|\left(x^i(k)\right)^T, \left(\hat{x}_k^{-i}\right)^T\right\|_{\overline{Q}_i}^2 + \left\|u^{*i}(0 \mid k)\right\|_{\overline{R}_i}^2\right\} \\ &+ C_x^*(k) - C_\xi^*(k) \le 0. \end{aligned} \tag{18}$$

Then, by solving the receding-horizon optimization problem

$$\min_{\overline{U}^i(k)} \overline{J}_k^i, s.t. (1), (2), (14), (16), \qquad u_{k,N}^i = K_i x_{k,N}^i, x_{k,N}^i \in \Omega^i, \tag{19}$$

and implementing $u_{k,0}^{*i}$, the stability of the global closed-loop system is guaranteed, once a feasible solution at time $k = 0$ is found.

Proof. Define $\overline{J}(k) = \sum_{i=1}^{N_a} \overline{J}_k^i$. Suppose, at time k, there are optimal solution $\overline{U}_k^{*i}$, $i \in \{1, \ldots, N_a\}$, which yields

$$\begin{aligned} \overline{J}^*(k) = &\sum_{i=1}^{N_a}\left\{\left\|\left(x_k^i\right)^{\mathrm{T}}, \left(\hat{x}_k^{-i}\right)^{\mathrm{T}}\right\|_{\overline{Q}_i}^2 + \left\|u_{k,0}^{*i}\right\|_{\overline{R}_i}^2\right\} \\ &+ \sum_{i=1}^{N_a}\sum_{t=1}^{N-1}\left\{\left\|\left(x_{k,t}^{*i}\right)^{\mathrm{T}}, \left(\hat{x}_{k,t}^{-i}\right)^{\mathrm{T}}\right\|_{\overline{Q}_i}^2 + \left\|u_{k,t}^{*i}\right\|_{\overline{R}_i}^2 \right. \\ &\left. + \left\|\left(x_{k,t}^{*i}\right)^{\mathrm{T}} - \left(\hat{x}_{k,t}^{i}\right)^{\mathrm{T}}\right\|_{\overline{T}_i}^2\right\} + \sum_{i=1}^{N_a}\left\|x_{k,N}^{*i}\right\|_{P_i}^2. \end{aligned} \tag{20}$$

At time $t + 1$, according to Lemma 2, $\overline{U}_{k+1}^i = \{u_{k,1}^{*i}, \ldots, u_{k,N-1}^{*i}, K_i x_{k,N}^{*i}\}$ is feasible, which yields

$$\begin{aligned} \overline{J}(k+1) &= \sum_{i=1}^{N_a}\sum_{t=1}^{N}\left\{\left\|\left(x_{k,t}^{*i}\right)^{\mathrm{T}}, \left(x_{k,t}^{-*i}\right)^{\mathrm{T}}\right\|_{\overline{Q}_i}^2 + \left\|u_{k,t}^{*i}\right\|_{\overline{R}_i}^2\right\} \\ &\quad + \sum_{i=1}^{N_a}\left\|x_{k,N+1}^{*i}\right\|_{P_i}^2 \\ &= \sum_{i=1}^{N_a}\sum_{t=1}^{N-1}\left\{\left\|\left(x_{k,t}^{*i}\right)^{\mathrm{T}}, \left(x_{k,t}^{-*i}\right)^{\mathrm{T}}\right\|_{\overline{Q}_i}^2 + \left\|u_{k,t}^{*i}\right\|_{\overline{R}_i}^2\right\} \\ &\quad + \left\|x_{k,N}^*\right\|_Q^2 + \left\|u_{k,N}^*\right\|_R^2 + \left\|x_{k,N+1}^*\right\|_P^2, \end{aligned} \tag{21}$$

where $P = \mathrm{diag}\{P_1, P_2, \ldots, P_{N_a}\}$. By applying (9) and Lemma 2, (11) guarantees that

$$\left\|x_{k,N+1}^*\right\|_P^2 - \left\|x_{k,N}^*\right\|_P^2 \le -\left\|x_{k,N}^*\right\|_Q^2 - \left\|u_{k,N}^*\right\|_R^2. \tag{22}$$

Substituting (22) into $\overline{J}(k+1)$ yields

$$\begin{aligned} \overline{J}(k+1) \le &\sum_{i=1}^{N_a}\sum_{t=1}^{N-1}\left\{\left\|\left(x_{k,t}^{*i}\right)^{\mathrm{T}}, \left(x_{k,t}^{-*i}\right)^{\mathrm{T}}\right\|_{\overline{Q}_i}^2 + \left\|u_{k,t}^{*i}\right\|_{\overline{R}_i}^2\right\} \\ &+ \sum_{i=1}^{N_a}\left\|x_{k,N}^{*i}\right\|_{P_i}^2. \end{aligned} \tag{23}$$

By applying (17)–(19),

$$\begin{aligned} &\overline{J}(k+1) - \overline{J}^*(k) \\ &\quad \le -(1-\rho(k))\sum_{i=1}^{N_a}\left\{\left\|\left(x_k^i\right)^{\mathrm{T}}, \left(\hat{x}_k^{-i}\right)^{\mathrm{T}}\right\|_{\overline{Q}_i}^2 + \left\|u_{k,0}^{*i}\right\|_{\overline{R}_i}^2\right\} \\ &\quad \le -(1-\rho(k))\|x_k\|_Q^2. \end{aligned} \tag{24}$$

At time $k + 1$, by reoptimization, $\overline{J}^*(k+1) \le \overline{J}(k+1)$. Hence, it leads to

$$\begin{aligned} \overline{J}^*(k+1) - \overline{J}^*(k) &\le -(1-\rho(k))\|x_k\|_Q^2 \\ &\le -(1-\rho(k))\lambda_{\min}(Q)\|x(k)\|_Q^2, \end{aligned} \tag{25}$$

where $\lambda_{\min}(Q)$ is the minimum eigenvalue of Q. This indicates that the closed-loop system is exponentially stable.

Satisfaction of (18) indicates that all $x_{k,t}^i$ should not deviate too far from their assumed values $\hat{x}_{k,t}^i$ [13]. Hence, (18) can be taken as a new version of the compatibility condition. This compatibility condition is derived from a single compatibility condition that collects all the states (whether predicted or assumed) with in the switching horizon and is disassembled to each agent in distributed manner, which results in local compatibility constraint for each agent. □

3.3. Synthesis Approach of Distributed MPC. In the synthesis approach, the local optimization problem incorporates the above compatibility condition. Since $x_{k,t}^*$ for all agent i is coupled with other agents through (18), it is necessary to assign the constraint to each agent so as to satisfy (18) along the optimization. The continued discussion on stability depends on handling of (18).

Denote $\xi_k^i = [\xi_k^{i,1}, \ldots, \xi_k^{i,n_i}]^{\mathrm{T}}$, $\xi_k^{-i} = \{\xi_k^j \mid j \in \mathcal{N}_i\}$. At time $k > 0$, by solving the optimization problem, there exits a parameter $\varepsilon_k^{i,l}$, $l = 1, \ldots, n_i$, for each element of $\xi_k^{i,l}$, $l = 1, \ldots, n_i$.

Define

$$\varepsilon_k^{i,l} = \max_t \left|\xi_{k-1,t}^{i,l}\right|, \tag{26}$$

and denote $\varepsilon_k^i = [\varepsilon_k^{i,1}, \ldots, \varepsilon_k^{i,n_i}]^{\mathrm{T}}$, $\varepsilon_k^{-i} = \{\varepsilon_k^j \mid j \in \mathcal{N}_i\}$. At time $k + 1 > 0$, set following constraint for each agent i:

$$\left|\left(x_{k+1,t}^i\right)^{\mathrm{T}} - \left(\hat{x}_{k+1,t}^i\right)^{\mathrm{T}}\right| < \varepsilon_k^i. \tag{27}$$

From (26) and (27), it is shown that $\xi_{k+1,t}^i < \varepsilon_k^i$ and $\xi_{k+1,t}^{-i} < \varepsilon_k^{-i}$.

Denote

$$\begin{aligned} C_x^{*i}(k) = \sum_{t=1}^{N-1}\left\{2\left(x_{k,t}^{*i}\right)^{\mathrm{T}}\overline{Q}_i^{12}\varepsilon_k^{-i}\right. \\ \left. + 2\left(\hat{x}_{k,t}^{-i}\right)^{\mathrm{T}}\overline{Q}_i^3\varepsilon_k^{-i} + \left(\varepsilon_k^{-i}\right)^{\mathrm{T}}\overline{Q}_i^3\left(\varepsilon_k^{-i}\right)^{\mathrm{T}}\right\}, \end{aligned} \tag{28}$$

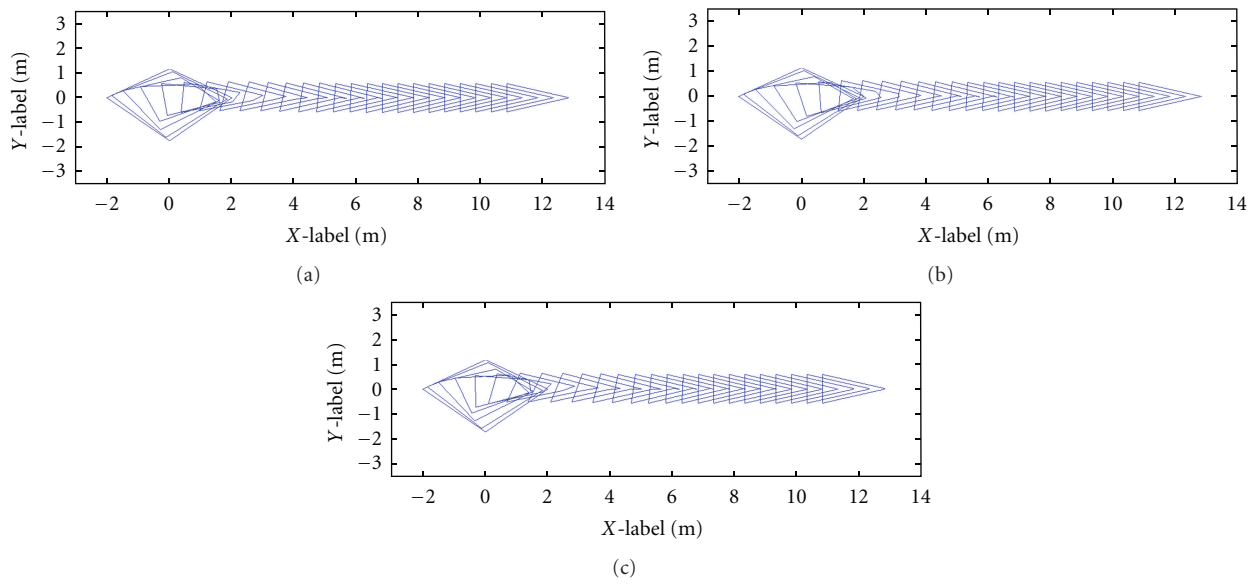


Figure 1: Evolutions of the formation with different control schemes.

$$C_{\xi}^{*i}(k) = \sum_{t=1}^{N-1} \left(\xi_{k,t}^{i}\right)^{\mathrm{T}} \overline{T}_i \xi_{k,t}^{i}. \tag{29}$$

Then $C_x^*(k) \le \sum_{i=1}^{N_a} C_x^{*i}(k)$, $C_\xi^*(k) = \sum_{i=1}^{N_a} C_\xi^{*i}(k)$.

By applying (26)–(29), it is shown that (18) is guaranteed by assigning

$$\begin{aligned} &0 \le \rho_i(k) \le 1, \\ &\sum_{i=1}^{N_a} -\rho_i(k)\left\{\left\|\left(x_k^i\right)^{\mathrm{T}}, \left(\hat{x}_k^{-i}\right)^{\mathrm{T}}\right\|_{\overline{Q}_i}^2 + \left\|u_{k,0}^{*i}\right\|_{\overline{R}_i}^2\right\} \\ &\quad + \sum_{i=1}^{N_a} C_x^{*i}(k) - \sum_{i=1}^{N_a} C_\xi^{*i}(k) \le 0. \end{aligned} \tag{30}$$

is dispensed to agent i:

$$\begin{aligned} &0 \le \rho_i(k) \le 1, \\ &\sum_{t=1}^{N-1} \left\|\xi_{k,t}^{i}\right\|_{\overline{T}_i}^2 \ge -\rho_i(k)\left\{\left\|\left(x_k^i\right)^{\mathrm{T}}, \left(\hat{x}_k^{-i}\right)^{\mathrm{T}}\right\|_{\overline{Q}_i}^2 \right. \\ &\qquad \left. + \left\|u_{\mathrm{k},0}^{*i}\right\|_{\overline{R}_i}^2\right\} + C_x^{*i}(k). \end{aligned} \tag{31}$$

By using (26)–(28), conservativeness is introduced. Hence, (31) is more stringent than (18).

Remark 3. By adding the deviation punishment term in the local cost function, the closed-loop stability follows with a large weight. The larger weight means the more loss of the performance [14, 19]. For a small value of $\overline{T}_i$, we can adjust the value of $\rho_i(k)$ to obtain exponential stability. As the $\rho_i(k)$ is set by optimization, this scheme has more freedom to tuning parameters, to balance the closed-loop stability and control performance.

Remark 4. According to (31), the maximum value and minimum value of $\overline{T}_i$ can be calculated by considering the range of each variable. We choose the middle value for $\overline{T}_i$. Obviously, the $\overline{T}_i$ is time varying and denoted as $\overline{T}_i(k)$.

4. Control Strategy

For practical implementation, distributed MPC is formulated in the following algorithm.

Algorithm. Off-line stage:

(i) Set the value of the prediction horizon N.

(ii) According to (3), (5) and (9), find Q_i, R_i, $\overline{Q}_i, \overline{R}_i$, $t = 0, \ldots, N-1$, for all agents.

(iii) Set the value of the compatibility constraint for all agents $\mathcal{E}_i(0) = +\infty$, $j \in \mathcal{N}_i$.

(iv) Calculate the terminal weight P_i, local linear feedback control gain K_i and the terminal set Ω^i.

On-line stage: For agent i, perform the following steps at $k = 0$:

(i) Take the measurement of x_0^i. Set $\overline{T}_i = 0$.

(ii) Send x_0^i to its neighbor j, $j \in \mathcal{N}_i$ of agent i. Receive x_0^j.

(iii) Set $\hat{x}_{t,0}^j = x_{0,0}^j$, $j \in \mathcal{N}_i$, $t = 0, \ldots, N-1$ and $\hat{x}_{0,t}^i = x_0^i$.

(iv) Solve problem (19).

(v) Implement $u_0^i = u_{0,0}^{*i}$.

(vi) Get $\hat{x}_{t,0}^i$ and the value of compatibility constraint $\mathcal{E}_i(1)$.

(vii) Send $\hat{x}^i_{0,t}$ and $\mathcal{E}_i(1)$ to its neighbor j, $j \in N_i$. Receive $\hat{x}^j_{0,t}$ and $\mathcal{E}_j(1)$. Calculate $\overline{T}_i(k)$.

For the agent i, perform the following steps at $k > 0$:

(i) Take the measurement of x^i_k.

(ii) Solve problem (19).

(iii) Implement $u^i_k = u^{*i}_{0,k}$.

(iv) Get $\hat{x}^i_{k,t}$ and the new value of compatibility constraint $\mathcal{E}_i(k+1)$.

(v) Send $\hat{x}^i_{k,t}$ and $\mathcal{E}_i(k+1)$ to its neighbor j, $j \in \mathcal{N}_i$. Receive $\hat{x}^j_{k,t}$ and $\mathcal{E}_j(k+1)$.

(vi) Calculate $\overline{T}_i(k)$.

5. Numerical Example

We consider the model of agent i [22] as

$$x^i_{k+1} = \begin{bmatrix} I_2 & I_2 \\ 0 & I_2 \end{bmatrix} x^i_k + \begin{bmatrix} 0.5I_2 \\ I_2 \end{bmatrix} u^i_k, \tag{32}$$

which is obtained by discretizing the continuous-time model

$$\dot{x}^i = \begin{bmatrix} 0 & I_2 \\ 0 & 0 \end{bmatrix} x^i + \begin{bmatrix} 0 \\ I_2 \end{bmatrix} u^i. \tag{33}$$

($x^i_k = [q^{i,x}_k, q^{i,y}_k, v^{i,x}_k, v^{i,y}_k]^{\mathrm{T}}$, $q^{i,x}_k$ and $q^{i,y}_k$ are positions in the horizontal and vertical directions, resp. $v^{i,x}_k$ and $v^{i,y}_k$ are velocities in the horizontal and vertical directions, resp.) with sampling time interval of 0.5 second. There are four agents. A set of positions of the four agents constitute a formation. The initial positions of the four agents are

$$\left[q^{1,x}_o, q^{1,y}_o\right] = [0, 2], \qquad \left[q^{2,x}_o, q^{2,y}_o\right] = [-2, 0], \tag{34}$$

$$\left[q^{3,x}_o, q^{3,y}_o\right] = [0, -3], \qquad \left[q^{4,x}_o, q^{4,y}_o\right] = [2, 0]. \tag{35}$$

Linear constraints on states and input are

$$\left|x^i\right| \le \begin{bmatrix} 100 & 100 & 15 & 15 \end{bmatrix}^{\mathrm{T}}, \qquad \left|u^i\right| \le \begin{bmatrix} 2 & 2 \end{bmatrix}^{\mathrm{T}}. \tag{36}$$

The agent i, $i = 1, 2, 3$ are selected as the core agents of the formation. $\mathcal{A}_0$ is designed as $\mathcal{A}_0 = \{(1,2); (1,3); (2,4)\}$. If all systems achieve the desire formation and the core agents cooperatively cover the virtue leader, then $u^{i,x}_k(k) = 0$, $u^{i,y}_k = 0$. The global cost function is obtained as

$$\begin{aligned} J(k) = \sum_{t=0}^{\infty} \Big[& \left\| q^1_{k,t} - q^2_{k,t} + c_{12} \right\|^2 + \left\| q^1_{k,t} - q^3_{k,t} + c_{13} \right\|^2 \\ & + \left\| q^2_{k,t} - q^4_{k,t} + c_{24} \right\|^2 + \frac{1}{9} \left\| \left(q^1_{k,t} + q^2_{k,t} + q^3_{k,t} \right) - q_c \right\|^2 \\ & + \left\| v^1_{k,t} \right\|^2 + \left\| v^2_{k,t} \right\|^2 + \left\| v^3_{k,t} \right\|^2 + \left\| v^4_{k,t} \right\|^2 + \left\| u_{k,t} \right\|^2 \Big]. \end{aligned} \tag{37}$$

They cooperatively track the virtual leader whose reference is $q_c = (0.5 * k, 0)$. The distance between agents is defined as $c_{12} = (-2, 1), c_{13} = (-2, -1), c_{24} = (-2, 1)$. Choose $\mathcal{N}_1 = \{2\}$, $\mathcal{N}_2 = \{1\}$, $\mathcal{N}_3 = \{1\}$, $\mathcal{N}_4 = \{2\}$. Then,

$$Q = \begin{bmatrix} 2\frac{1}{9}I_2 & 0 & -\frac{8}{9}I_2 & 0 & -\frac{8}{9}I_2 & 0 & 0 & 0 \\ 0 & I_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{8}{9}I_2 & 0 & 2\frac{1}{9}I_2 & 0 & \frac{1}{9}I_2 & 0 & -I_2 & 0 \\ 0 & 0 & 0 & I_2 & 0 & 0 & 0 & 0 \\ -\frac{8}{9}I_2 & 0 & \frac{1}{9}I_2 & 0 & 1\frac{1}{9}I_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_2 & 0 & 0 \\ 0 & 0 & -I_2 & 0 & 0 & 0 & I_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_2 \end{bmatrix}, \qquad R = I_8.$$

$$\overline{Q}_1 = \begin{bmatrix} \frac{7}{9}I_2 & 0 & -\frac{4}{9}I_2 & 0 \\ 0 & \frac{1}{3}I_2 & 0 & 0 \\ -\frac{4}{9}I_2 & 0 & I_2 & 0 \\ 0 & 0 & 0 & \frac{1}{3}I_2 \end{bmatrix},$$

$$\overline{Q}_2 = \begin{bmatrix} 1\frac{1}{9}I_2 & 0 & -\frac{4}{9}I_2 & 0 \\ 0 & \frac{1}{3}I_2 & 0 & 0 \\ -\frac{4}{9}I_2 & 0 & \frac{4}{9}I_2 & 0 \\ 0 & 0 & 0 & \frac{1}{3}I_2 \end{bmatrix},$$

$$\overline{Q}_3 = \begin{bmatrix} 1\frac{1}{9}I_2 & 0 & -\frac{8}{9}I_2 & 0 \\ 0 & \frac{1}{2}I_2 & 0 & 0 \\ -\frac{8}{9}I_2 & 0 & \frac{8}{9}I_2 & 0 \\ 0 & 0 & 0 & \frac{1}{3}I_2 \end{bmatrix},$$

$$\overline{Q}_4 = \begin{bmatrix} I_2 & 0 & -I_2 & 0 \\ 0 & I_2 & 0 & 0 \\ -I_2 & 0 & I_2 & 0 \\ 0 & 0 & 0 & \frac{1}{3}I_2 \end{bmatrix}, \tag{38}$$

and $\overline{R}_i = I_2$, $i \in \{1, 2, 3, 4\}$. Choose $Q_i = 6.85 * I_4$ and $R_i = I_2$, $i \in \{1, 2, 3, 4\}$, $N = 10$. The terminal set is $\alpha_i = 0.22$. The above choice of model, cost, and constraints allow us to rewrite problem (19) as a quadratic programming with quadratic constraint. To solve the optimal control problems numerically, the package NPSOL 5.02 is used. From top to bottom, the first subgraph of Figure 1 is the evolution of the formation with central MPC; the second sub-graph of Figure 1 is the evolution of the formation with distributed MPC with time-varying compatible constraint; the third sub-graph of Figure 1 is the evolution of the formation with distributed MPC with a fixed compatibility constraint.

With the three control schemes, the formation of all agents can be achieved. The obtained $J_{\text{true's}}$ are 2.5779×10^6, 4.8725×10^6, and 5.654×10^6, respectively. Compared with the second sub-graph, the third sub-graph have a

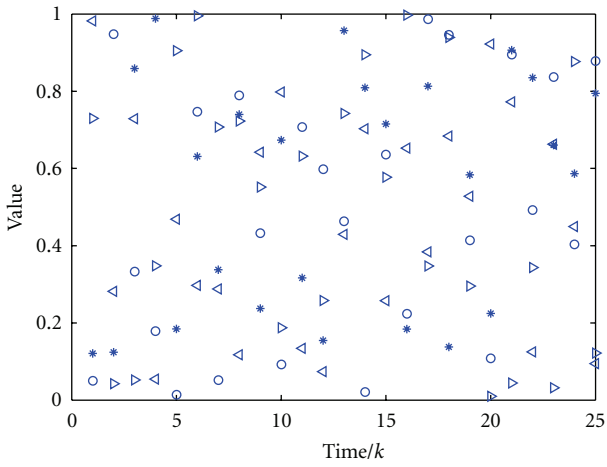


FIGURE 2: The value of $\rho_i(k)$.

large overshoot at the time-instant $k = 9$ (nearby the position $(3, 0)$). The distributed MPC with the time-varying compatible constraint has a better control process comparing to the one with fixed compatible constraint. The value of $\rho_i(k)$ is shown in Figure 2. "$*$" for agent 1; "O" for agent 2; ">" for agent 3; "<" for agent 4.

Remark 5. For the second simulation, the value of the fixed compatible constraint is 0.2. For the third simulation, the values of the time-varying compatible constraint is calculated according to the states deviation of the previous horizon.

6. Conclusions

In this paper, we have proposed an improved distributed MPC scheme for multiagent systems based on deviation punishment. One of the features of the proposed scheme is that the cost function of each agent penalizes the deviation between the predicted state trajectory and the assumed state trajectory, which improves the consistency and optimal control trajectory. At each sample time, the value of compatibility constraint is set by the deviation of previous sample time-instant. The closed-loop stability is guaranteed with a small value for the weight of the deviation function term. Furthermore, the effectiveness of the scheme has been investigated by a numerical example. One of the future works will focus on feasibility of optimization.

Acknowledgment

This work is supported by a Grant from the Fundamental Research Funds for the Central Universities of China, no. *CDJZR10170006*.

References

[1] J. A. Primbs, "The analysis of optimization based controllers," *Automatica*, vol. 37, no. 6, pp. 933–938, 2001.

[2] J. M. Maciejowski, *Predictive Control with Constraints*, Prentice Hall, Englewood Cliffs, NJ, USA, 2002.

[3] J. A. Rossiter, *Model-Based Predictive Control: A Practical*, CRC, Boca Raton, Fla, USA, 2003.

[4] Y. Kuwata, A. Richards, T. Schouwenaars, and J. P. How, "Distributed robust receding horizon control for multivehicle guidance," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 627–641, 2007.

[5] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar, "Distributed model predictive control," *IEEE Control Systems Magazine*, vol. 22, no. 1, pp. 44–52, 2002.

[6] S. Li, Y. Zhang, and Q. Zhu, "Nash-optimization enhanced distributed model predictive control applied to the Shell benchmark problem," *Information Sciences*, vol. 170, no. 2–4, pp. 329–349, 2005.

[7] A. N. Venkat, J. B. Rawlings, and S. J. Wright, "Stability and optimality of distributed model predictive control," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference (CDC-ECC '05)*, vol. 2005, pp. 6680–6685, Seville, Spain, December 2005.

[8] A. N. Venkat, J. B. Rawlings, and S. J. Wright, "Distributed model predictive control of large-scale systems," in *Assessment and Future Directions of Nonlinear Model Predictive Control*, vol. 358 of *Lecture Notes in Control and Information Sciences*, pp. 591–605, 2007.

[9] M. Mercangöz and F. J. Doyle III, "Distributed model predictive control of an experimental four-tank system," *Journal of Process Control*, vol. 17, no. 3, pp. 297–308, 2007.

[10] D. Jia and B. Krogh, "Min-max feedback model predictive control for distributed control with communication," in *Proceedings of the American Control Conference*, pp. 4507–4512, May 2002.

[11] T. Keviczky, F. Borrelli, and G. J. Balas, "Decentralized receding horizon control for large scale dynamically decoupled systems," *Automatica*, vol. 42, no. 12, pp. 2105–2115, 2006.

[12] F. Borrelli, T. Keviczky, G. J. Balas, G. Stewart, K. Fregene, and D. Godbole, "Hybrid decentralized control of large scale systems," in *Hybrid Systems: Computation and Control*, vol. 3414 of *Lecture Notes in Computer Science*, pp. 168–183, Springer, 2005.

[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, 2006.

[14] W. B. Dunbar, "Distributed receding horizon control of cost coupled systems," in *Proceedings of the 46th IEEE Conference on Decision and Control (CDC '07)*, pp. 2510–2515, December 2007.

[15] S. Wei, Y. Chai, and B. Ding, "Distributed model predictive control for multiagent systems with improved consistency," *Journal of Control Theory and Applications*, vol. 8, no. 1, pp. 117–122, 2010.

[16] B. Ding, "Distributed robust MPC for constrained systems with polytopic description," *Asian Journal of Control*, vol. 13, no. 1, pp. 198–212, 2011.

[17] B. Ding, L. Xie, and W. Cai, "Distributed model predictive control for constrained linear systems," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 11, pp. 1285–1298, 2010.

[18] B. Ding and S. Y. Li, "Design and analysis of constrained nonlinear quadratic regulator," *ISA Transactions*, vol. 42, no. 2, pp. 251–258, 2003.

[19] W. Shanbi, B. Ding, C. Gang, and C. Yi, "Distributed model predictive control for multi-agent systems with coupling constraints," *International Journal of Modelling, Identification and Control*, vol. 10, no. 3-4, pp. 238–245, 2010.

[20] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.

[21] T. A. Johansen, "Approximate explicit receding horizon control of constrained nonlinear systems," *Automatica*, vol. 40, no. 2, pp. 293–300, 2004.

[22] W. B. Dunbar, *Distributed receding horizon control for multiagent systems*, Ph.D. thesis, California Institute of Technology, Pasadena, Calif, USA, 2004.

Hindawi Publishing Corporation
Journal of Control Science and Engineering
Volume 2012, Article ID 402948, 12 pages
doi:10.1155/2012/402948

Research Article

Model Predictive Control of Uncertain Constrained Linear System Based on Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Control Approach

Patience E. Orukpe

Department of Electrical and Electronic Engineering, University of Benin, P.M.B 1154, Benin City, Edo State, Nigeria

Correspondence should be addressed to Patience E. Orukpe, patience.orukpe01@imperial.ac.uk

Received 30 June 2011; Revised 13 October 2011; Accepted 5 November 2011

Academic Editor: Marcin T. Cychowski

Uncertain constrained discrete-time linear system is addressed using linear matrix inequality based optimization techniques. The constraints on the inputs and states are specified as quadratic constraints but are formulated to capture hyperplane constraints as well. The control action is of state feedback and satisfies the constraints. Uncertainty in the system is represented by unknown bounded disturbances and system perturbations in a linear fractional transform (LFT) representation. Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ method is applied in a model predictive control strategy. The control law takes account of disturbances and uncertainty naturally. The validity of this approach is illustrated with two examples.

1. Introduction

Model predictive control (MPC) is a class of model-based control theories that use linear or nonlinear process models to forecast system behaviour. MPC is one of the control techniques that is able to cope with model uncertainties in an explicit way [1]. MPC has been used widely in practical applications to industrial process systems [2] and active vibration control of railway vehicles [3]. One of the methods used in MPC when uncertainties are present is to minimise the objective function for the worst possible case. This strategy is known as minimax and was originally proposed [4] in the context of robust receding control, [5] in the context of feedback and feedforward control and [6] in the context of $\mathcal{H}_\infty$ MPC. MPC has been applied to $\mathcal{H}_\infty$ problems in order to combine the practical advantage of MPC with the robustness of the $\mathcal{H}_\infty$ control, since robustness of MPC is still being investigated for it to be applied practically.

This work is motivated by the work in [7, 8] where uncertainty in the system was modeled by perturbations in a linear fractional representation. In [9], model predictive control based on a mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control approach was considered. The designed controller has the form of state feedback and was constructed from the solution of a set of feasibility linear matrix inequalities. However, the issue of handling both uncertainty and disturbances simultaneously was not considered. In this paper, we extend the result of [9] to constrained uncertain linear discrete-time invariant systems using a mixed $\mathcal{H}_2/\mathcal{H}_\infty$ design approach and the uncertainty considered is norm-bounded additive. This is more suitable as both performance and robustness issues are handled within a unified framework.

The method presented in this paper develops an LMI design procedure for the state feedback gain matrix F, allowing input and state constraints to be included in a less conservative manner. A main contribution is the accomplishment of a prescribed disturbance attenuation in a systematic way by incorporating the well-known robustness guarantees through $\mathcal{H}_\infty$ constraints into the MPC scheme. In addition, norm-bounded additive uncertainty is also incorporated. A preliminary version of some of the work presented in this paper was presented in [10].

The structure of the work is as follows. After defining the notation, we describe the system and give a statement of the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ problem in Section 2. In Section 3, we derive sufficient conditions, in the form of LMIs, for the existence of a state feedback control law that achieves the design specifications. In Section 4, we consider two examples that illustrate our algorithm. Finally, we conclude in Section 5.

The notation we use is fairly standard. $\mathcal{R}$ denotes the set of real numbers, $\mathcal{R}^n$ denotes the space of n-dimensional (column) vectors whose entries are in $\mathcal{R}$ and $\mathcal{R}^{n\times m}$ denoting the space of all $n \times m$ matrices whose entries are in $\mathcal{R}$. For $A \in \mathcal{R}^{n\times m}$, we use the notation A^T to denote transpose. For $x, y \in \mathcal{R}^n$, $x < y$ (and similarly $\leq$, $>$, and $\geq$) is interpreted element wise. The identity matrix is denoted as I and the null matrix by 0 with the dimension inferred from the context.

2. Problem Formulation

We consider the following discrete-time linear time invariant system:

$$\begin{aligned} x_{k+1} &= Ax_k + B_w w_k + B_u u_k + B_p p_k, \\ q_k &= C_q x_k + D_{qu} u_k + D_{qw} w_k, \\ p_k &= \Delta_k q_k, \\ z_k &= \begin{bmatrix} C_z x_k \\ D_{zu} u_k \end{bmatrix}, \\ & x_0 \text{ given}, \end{aligned} \tag{1}$$

where x_0 is the initial state, $x_k \in \mathcal{R}^n$ is the state, $w_k \in \mathcal{R}^{n_w}$ is the disturbance, $u_k \in \mathcal{R}^{n_u}$ is the control, $z_k \in \mathcal{R}^{n_z}$ is the controlled output, $A \in \mathcal{R}^{n\times n}$, $B_w \in \mathcal{R}^{n\times n_w}$, $B_u \in \mathcal{R}^{n\times n_u}$, $C_z \in \mathcal{R}^{n_{z_1}\times n}$, and $D_{zu} \in \mathcal{R}^{n_{z_2}\times n_u}$, and where $n_z = n_{z_1} + n_{z_2}$. The signals q_k and p_k model uncertainties or perturbations appearing in the feedback loop.

The operator, Δ_k, is block diagonal:

$$\Delta_k \in \mathbf{\Delta}_k = \left\{ \Delta_k = \begin{bmatrix} \Delta_{1k} & & 0 \\ & \ddots & \\ 0 & & \Delta_{tk} \end{bmatrix} : \|\Delta_{ik}\| \leq 1 \ \forall i \right\}, \tag{2}$$

and is norm bounded by one. Scalings can be included in C_q and B_p, thus generalizing the bound. Δ_k can represent either a memoryless time-varying matrix with $\overline{\sigma}(\Delta_{ik}) \leq 1$, for $i = 1, \ldots, t$, $k \geq 0$, or the constraints:

$$p_{ik}^T p_{ik} \leq q_{ik}^T q_{ik}, \quad i = 1, \ldots, t, \tag{3}$$

where $p_k = [p_{1k}, \ldots, p_{tk}]^T$, $q_k = [q_{1k}, \ldots, q_{tk}]^T$, and the partitioning is induced by Δ_k. Each Δ_k is assumed to be either a full block or a repeated scalar block, and models a number of factors, such as dynamics or parameters, nonlinearities, that are unknown, unmodeled or neglected. In this work, we only consider full blocks for simplicity.

In terms of the state space matrices, this formulation can be viewed as replacing a fixed (A, B_u, B_w) by $(A, B_u, B_w) \in (\mathcal{A}, \mathcal{B}_u, \mathcal{B}_w)$, where

$$\begin{aligned} (\mathcal{A}, \mathcal{B}_u, \mathcal{B}_w) = \Big\{ \Big[& A + B_p \Delta_k C_q, B_u + B_p \Delta_k D_{qu}, B_w \\ & + B_p \Delta_k D_{qw} \Big] \mid \Delta_k \in \mathbf{\Delta}_k \Big\}. \end{aligned} \tag{4}$$

In robust model predictive control, we consider norm-bounded uncertainty and define stability in terms of quadratic stability [11] which requires the existence of a fixed quadratic Lyapunov function ($V(\zeta) = \zeta^T P \zeta$, $P > 0$) for all possible choices of the uncertainty parameters.

In the case of norm-bounded uncertainties:

$$\begin{aligned} &\begin{bmatrix} A & B_w & B_u \end{bmatrix} \\ &\quad \in \left\{ \begin{bmatrix} A^o & B_w^o & B_u^o \end{bmatrix} + F_A \Delta_H \begin{bmatrix} E_A & E_w & E_u \end{bmatrix} : \Delta \in \mathbf{\Delta} \right\}, \end{aligned} \tag{5}$$

where $[A^o \ B_w^o \ B_u^o]$ represents the nominal model, $\Delta_H = \Delta(I - H\Delta)^{-1}$, with

$$\begin{aligned} \Delta \in \mathbf{\Delta} := \Big\{ & \Delta = \operatorname{diag}\left(\delta_1 I_{q_1}, \ldots, \delta_l I_{q_l}, \Delta_{l+1}, \ldots, \Delta_{l+f} \right) : \|\Delta\| \\ & \leq 1, \delta_i \in \mathcal{R}, \Delta_i \in \mathcal{R}^{q_i \times q_i} \Big\} \end{aligned} \tag{6}$$

and where F_A, E_A, E_w, E_u, and H are known and constant matrices with appropriate dimensions. This linear fractional representation of uncertainty, which is assumed to be well posed over $\mathbf{\Delta}$ (i.e., $\det(I - H\Delta) \neq 0$ for all $\Delta \in \mathbf{\Delta}$), has great generality and is used widely in robust control theory [12].

We use the following lemma, which is a slight modification of a result in [13] and which uses the fact that $\Delta \in \mathbf{\Delta}$ to remove explicit dependence on Δ for the solution with norm bounded uncertainties.

Lemma 1. *Let $\mathbf{\Delta}$ be as described in* (6) *and define the subspaces*

$$\begin{aligned} \mathbf{\Sigma} = \Big\{ & \operatorname{diag}\left(S_1, \ldots, S_l, \lambda_1 I_{ql+1}, \ldots, \lambda_s I_{ql+f} \right) \\ & : S_i = S_i^T \in \mathcal{R}^{q_i \times q_i}, \quad \lambda_j \in \mathcal{R} \Big\}, \\ \mathbf{\Gamma} = \Big\{ & \operatorname{diag}\left(G_1, \ldots, G_l, 0_{ql+1}, \ldots, 0_{ql+f} \right) : G_i = -G_i^T \in \mathcal{R}^{q_i \times q_i} \Big\}. \end{aligned} \tag{7}$$

Let $T_1 = T_1^T, T_2, T_3, T_4$ be matrices with appropriate dimensions. We have $\det(I - T_4\Delta) \neq 0$ *and* $T_1 + T_2\Delta(I - T_4\Delta)^{-1}T_3 + T_3^T(I - \Delta^T T_4^T)^{-1}\Delta^T T_2^T < 0$ *for every $\Delta \in \mathbf{\Delta}$ if there exist $S \in \mathbf{\Sigma}$ and $G \in \mathbf{\Gamma}$ such that $S > 0$ and*

$$\begin{bmatrix} T_1 + T_2 S T_2^T & T_3^T + T_2 S T_4^T + T_2 G \\ T_3 + T_4 S T_2^T + G^T T_2^T & T_4 S T_4^T + T_4 G + G^T T_4^T - S \end{bmatrix} < 0. \tag{8}$$

If $\mathbf{\Delta}$ is unstructured, then (8) *becomes*

$$\begin{bmatrix} T_1 + \lambda T_2 T_2^T & T_3^T + \lambda T_2 T_4^T \\ T_3 + \lambda T_4 T_2^T & \lambda\left(T_4 T_4^T - I \right) \end{bmatrix} < 0, \tag{9}$$

for some scalar $\lambda > 0$. In this case, condition (9) *is both necessary and sufficient.*

We also use the following Schur complement result [14].

Lemma 2. *Let $X_{11} = X_{11}^T$ and $X_{22} = X_{22}^T$. Then*

$$\begin{aligned} &\begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix} \geq 0 \Longleftrightarrow X_{22} \geq 0, \\ &X_{22} - X_{12}^T X_{11}^+ X_{12} \geq 0, \qquad X_{12}(I - X_{11}X_{11}^+) = 0, \end{aligned} \tag{10}$$

where X_{11}^+ denotes the Moore-Penrose pseudo-inverse of X_{11}.

We assume that the pair (A, B_u) is stabilizable and that the disturbance is bounded as

$$\|w\|_2 := \sqrt{\sum_{k=0}^{\infty} w_k^T w_k} \le \overline{w}, \tag{11}$$

where $\overline{w} \ge 0$ is know.

The aim is to find a state feedback control law $\{u_k = Fx_k\}$ in $\mathcal{L}_2$, where $F \in \mathcal{R}^{n_u \times n}$, such that the following constraints are satisfied for all $\Delta_k \in \mathbf{\Delta}_k$.

(1) *Closed-loop stability*: the matrix $A + B_u F$ is stable.

(2) *Disturbance rejection*: for given $\gamma > 0$, the transfer matrix from w to z, denoted as T_{zw}, is quadratically stable and satisfies the $\mathcal{H}_\infty$ constraint

$$\|z\|_2 < \gamma \|w\|_2, \tag{12}$$

for $x_0 = 0$.

(3) *Regulation*: for given $\alpha > 0$, the controlled output satisfies the $\mathcal{H}_2$ constraint:

$$\|z\|_2 := \sqrt{\sum_{k=0}^{\infty} z_k^T z_k} < \alpha. \tag{13}$$

(4) *Input constraints*: for given $H_1, \ldots, H_{m_u} \in \mathcal{R}^{n_u \times n_u}$, $H_j = H_j^T \ge 0$, $h_1, \ldots, h_{m_u} \in \mathcal{R}^{n_u \times 1}$, and $\overline{u}_1, \ldots, \overline{u}_{m_u} \in \mathcal{R}$, the inputs satisfy the quadratic constraints:

$$u_k^T H_j u_k + 2h_j^T u_k \le \overline{u}_j, \quad \forall k; \text{ for } j = 1, \ldots, m_u. \tag{14}$$

(5) *State/output constraints*: for given $G_1, \ldots, G_{m_x} \in \mathcal{R}^{n \times n}$, $G_j = G_j^T \ge 0$, $g_1, \ldots, g_{m_x} \in \mathcal{R}^{n \times 1}$, and $\overline{x}_1, \ldots, \overline{x}_{m_x} \in \mathcal{R}$ the states/outputs satisfy the quadratic constraints:

$$x_{k+1}^T G_j x_{k+1} + 2g_j^T x_{k+1} \le \overline{x}_j, \quad \forall k; \text{ for } j = 1, \ldots, m_x. \tag{15}$$

An $F \in \mathcal{R}^{n_u \times n}$ satisfying these requirements will be called an admissible state feedback gain.

3. LMI Formulation of Sufficiency Conditions

The next theorem, which is the main result of this paper, derives sufficient conditions, in the form of LMIs, for the existence of an admissible F.

Theorem 3. *Let all variables, definitions, and assumptions be as above. Then there exists an admissible state feedback gain matrix F if there exists solutions $Q = Q^T \in \mathcal{R}^{n \times n}$, $Y \in \mathcal{R}^{n_u \times n}$, $\delta_j \ge 0$, $\mu_j \ge 0$, $\nu_j \ge 0$, $\overline{\Lambda} = \operatorname{diag}(\overline{\lambda}_1 I, \ldots, \overline{\lambda}_t I) > 0$, and $\overline{\Psi}_j = \operatorname{diag}(\overline{\psi}_1 I, \ldots, \overline{\psi}_t I) > 0$ to the LMIs shown in* (16)–(19).

$$\begin{bmatrix} -Q & \star & \star & \star & \star & \star & \star \\ 0 & -\alpha^2\gamma^2 I & \star & \star & \star & \star & \star \\ 0 & 0 & -\overline{\Lambda} & \star & \star & \star & \star \\ AQ + B_u Y & \alpha^2 B_w & B_p\overline{\Lambda} & -Q & \star & \star & \star \\ C_q Q + D_{qu} Y & \alpha^2 D_{qw} & 0 & 0 & -\overline{\Lambda} & \star & \star \\ C_z Q & 0 & 0 & 0 & 0 & -\alpha^2 I & \star \\ D_{zu} Y & 0 & 0 & 0 & 0 & 0 & -\alpha^2 I \end{bmatrix} < 0, \tag{16}$$

$$\begin{bmatrix} 1 & \star & \star \\ \gamma^2\overline{w}^2 & \alpha^2\gamma^2\overline{w}^2 & \star \\ x_0 & 0 & Q \end{bmatrix} \ge 0, \tag{17}$$

$$\begin{bmatrix} Q & \star & \star & \star \\ H_j^{1/2} Y & \mu_j I & \star & \star \\ -h_j^T Y & 0 & \mu_j \overline{u}_j & \star \\ 0 & 0 & \mu_j & 1 \end{bmatrix} \ge 0, \quad j = 1, \ldots, m_u, \tag{18}$$

$$\begin{bmatrix} Q & \star & \star & \star & \star & \star & \star \\ 0 & \delta_j I & \star & \star & \star & \star & \star \\ 0 & 0 & \overline{\Psi}_j & \star & \star & \star & \star \\ G_j^{1/2}(AQ + B_u Y) & \nu_j G_j^{1/2} B_w & G_j^{1/2} B_p \overline{\Psi}_j & \nu_j I & \star & \star & \star \\ C_q Q + D_{qu} Y & \nu_j D_{qw} & 0 & 0 & \overline{\Psi}_j & \star & \star \\ -g_j^T(AQ + B_u Y) & -\nu_j g_j^T B_w & -g_j^T B_p \overline{\Psi}_j & 0 & 0 & \nu_j \overline{x}_j - \delta_j \overline{w}^2 & \star \\ 0 & 0 & 0 & 0 & 0 & \nu_j & 1 \end{bmatrix} \ge 0, \quad j = 1, \ldots, m_x. \tag{19}$$

Here, $\star$ represents terms readily inferred from symmetry and the partitioning of $\overline{\Lambda}$ and $\overline{\Psi}_j$ is induced by the partitioning of Δ_k. If such solutions exist, then $F = YQ^{-1}$.

Remark 4. The variables in the LMI minimization of Theorem 3 are computed online at time k, the subscript k is omitted for convenience.

Proof. Using $u_k = Fx_k$, the dynamics in (1) become

$$x_{k+1} = \overbrace{(A + B_u F)}^{A_{cl}} x_k + B_w w_k + B_p p_k, \qquad z_k = \overbrace{\begin{bmatrix} C_z \\ D_{zu}F \end{bmatrix}}^{C_{cl}} x_k. \tag{20}$$

Consider a quadratic function $V(x) = x^T P x$, $P > 0$ of the state x_k. It follows from (20) that

$$\begin{aligned} &V(x_{k+1}) - V(x_k) \\ &= x_k^T \left[A_{cl}^T P A_{cl} - P\right] x_k + x_k^T A_{cl}^T P B_w w_k + x_k^T A_{cl}^T P B_p p_k \\ &\quad + w_k^T B_w^T P A_{cl} x_k + w_k^T B_w^T P B_w w_k + w_k^T B_w^T P B_p p_k \\ &\quad + p_k^T B_p^T P A_{cl} x_k + p_k^T B_p^T P B_w w_k + p_k^T B_p^T P B_p p_k \\ &= \begin{bmatrix} x_k^T & w_k^T & p_k^T \end{bmatrix} K \begin{bmatrix} x_k \\ w_k \\ p_k \end{bmatrix} - x_k^T C_{cl}^T C_{cl} x_k + \gamma^2 w_k^T w_k, \end{aligned} \tag{21}$$

where

$$K = \begin{bmatrix} A_{cl}^T P A_{cl} - P + C_{cl}^T C_{cl} & A_{cl}^T P B_w & A_{cl}^T P B_p \\ B_w^T P A_{cl} & B_w^T P B_w - \gamma^2 I & B_w^T P B_p \\ B_p^T P A_{cl} & B_p^T P B_w & B_p^T P B_p \end{bmatrix}. \tag{22}$$

Using $q_k = (C_q + D_{qu}F)x_k + D_{qw}w_k$,

$$\begin{aligned} q_k^T \Lambda q_k &= x_k^T \left(C_q + D_{qu}F\right)^T \Lambda \left(C_q + D_{qu}F\right) x_k \\ &\quad + x_k^T \left(C_q + D_{qu}F\right)^T \Lambda D_{qw} w_k \\ &\quad + w_k^T D_{qw}^T \Lambda \left(C_q + D_{qu}F\right) x_k \\ &\quad + w_k^T D_{qw}^T \Lambda D_{qw} w_k, \end{aligned} \tag{23}$$

where $\Lambda = \operatorname{diag}(\lambda_1 I, \ldots \lambda_t I)$.

Substituting (23) into (21), it can be verified that we can write

$$\begin{aligned} V(x_{k+1}) - V(x_k) &= \begin{bmatrix} x_k^T & w_k^T & p_k^T \end{bmatrix} \overline{K} \begin{bmatrix} x_k \\ w_k \\ p_k \end{bmatrix} + p_k^T \Lambda p_k \\ &\quad - q_k^T \Lambda q_k - x_k^T C_{cl}^T C_{cl} x_k + \gamma^2 w_k^T w_k, \end{aligned} \tag{24}$$

where $\overline{K}$ is defined in (25) and $C_{pw} := C_q + D_{qu}F$.

$$\overline{K} = \begin{bmatrix} A_{cl}^T P A_{cl} - P + C_{cl}^T C_{cl} + C_{pw}^T \Lambda C_{pw} & A_{cl}^T P B_w + C_{pw}^T \Lambda D_{qw} & A_{cl}^T P B_p \\ B_w^T P A_{cl} + D_{qw}^T \Lambda C_{pw} & B_w^T P B_w - \gamma^2 I + D_{qw}^T \Lambda D_{qw} & B_w^T P B_p \\ B_p^T P A_{cl} & B_p^T P B_w & B_p^T P B_p - \Lambda \end{bmatrix}. \tag{25}$$

Assuming that $\lim_{k\to\infty} x_k = 0$ we have

$$\sum_{k=0}^{\infty} \left[x_{k+1}^T P x_{k+1} - x_k^T P x_k\right] = -x_0^T P x_0. \tag{26}$$

We write the $\mathcal{H}_2$ cost function as

$$\|z\|_2^2 = \sum_{k=0}^{\infty} \left(x_k^T C_{cl}^T C_{cl} x_k - \gamma^2 w_k^T w_k\right) + \gamma^2 \sum_{k=0}^{\infty} w_k^T w_k. \tag{27}$$

Adding (26) and (27) and carrying out a simple manipulation gives

$$\begin{aligned} \|z\|_2^2 &= x_0^T P x_0 + \gamma^2 \|w\|_2^2 \\ &\quad + \sum_{k=0}^{\infty} \begin{bmatrix} x_k^T & w_k^T & p_k^T \end{bmatrix} \overline{K} \begin{bmatrix} x_k \\ w_k \\ p_k \end{bmatrix} + \sum_{k=0}^{\infty} \left(p_k^T \Lambda p_k - q_k^T \Lambda q_k\right), \end{aligned} \tag{28}$$

where $\overline{K}$ is defined in (25).

Setting $x_0 = 0$, it follows from (3), (12), and (28) that $\|z\|_2 < \gamma \|w\|_2$ if $\overline{K} < 0$ and $\Lambda \geq 0$. In this work, we will take $\Lambda > 0$ to simplify our solution [8]. Using (2) and Lemma 1 it can be shown that

$$\overline{K} < 0, \tag{29}$$

is also sufficient for quadratic stability of T_{zw}.

Next, we linearize the matrix inequality (29) by applying a Schur complement, to give

$$\begin{bmatrix} -P & \star & \star & \star & \star & \star & \star \\ 0 & -\gamma^2 I & \star & \star & \star & \star & \star \\ 0 & 0 & -\Lambda & \star & \star & \star & \star \\ A_{cl} & B_w & B_p & -P^{-1} & \star & \star & \star \\ C_{pw} & D_{qw} & 0 & 0 & -\Lambda^{-1} & \star & \star \\ C_z & 0 & 0 & 0 & 0 & -I & \star \\ D_{zu}F & 0 & 0 & 0 & 0 & 0 & -I \end{bmatrix} < 0. \tag{30}$$

Pre- and post-multiplying the equation above by diag$(P^{-1}, I, I, I, I, I, I)$ gives

$$\begin{bmatrix} -P^{-1} & \star & \star & \star & \star & \star & \star \\ 0 & -\gamma^2 I & \star & \star & \star & \star & \star \\ 0 & 0 & -\Lambda & \star & \star & \star & \star \\ A_{cl}P^{-1} & B_w & B_p & -P^{-1} & \star & \star & \star \\ C_{pw}P^{-1} & D_{qw} & 0 & 0 & -\Lambda^{-1} & \star & \star \\ C_zP^{-1} & 0 & 0 & 0 & 0 & -I & \star \\ D_{zu}FP^{-1} & 0 & 0 & 0 & 0 & 0 & -I \end{bmatrix} < 0, \tag{31}$$

setting $Q = \alpha^2 P^{-1}$, $F = YP\alpha^{-2} = YQ^{-1}$, $C_{pw} = C_q + D_{qu}F$ and multiplying through by α^2, the equation above becomes

$$\begin{bmatrix} -Q & \star & \star & \star & \star & \star & \star \\ 0 & -\alpha^2\gamma^2 I & \star & \star & \star & \star & \star \\ 0 & 0 & -\alpha^2\Lambda & \star & \star & \star & \star \\ AQ + B_uY & \alpha^2 B_w & \alpha^2 B_p & -Q & \star & \star & \star \\ C_qQ + D_{qu}Y & \alpha^2 D_{qw} & 0 & 0 & -\alpha^2\Lambda^{-1} & \star & \star \\ C_zQ & 0 & 0 & 0 & 0 & -\alpha^2 I & \star \\ D_{zu}Y & 0 & 0 & 0 & 0 & 0 & -\alpha^2 I \end{bmatrix} < 0. \tag{32}$$

Pre- and post-multiplying the equation above by diag$(I, I, \Lambda^{-1}, I, I, I, I)$ gives

$$\begin{bmatrix} -Q & \star & \star & \star & \star & \star & \star \\ 0 & -\alpha^2\gamma^2 I & \star & \star & \star & \star & \star \\ 0 & 0 & -\alpha^2\Lambda^{-1} & \star & \star & \star & \star \\ AQ + B_uY & \alpha^2 B_w & \alpha^2 B_p\Lambda^{-1} & -Q & \star & \star & \star \\ C_qQ + D_{qu}Y & \alpha^2 D_{qw} & 0 & 0 & -\alpha^2\Lambda^{-1} & \star & \star \\ C_zQ & 0 & 0 & 0 & 0 & -\alpha^2 I & \star \\ D_{zu}Y & 0 & 0 & 0 & 0 & 0 & -\alpha^2 I \end{bmatrix} < 0. \tag{33}$$

The equation above is a bilinear matrix inequality, thus by defining $\alpha^2\Lambda^{-1}$ as a variable $\overline{\Lambda}$, we get the LMI in (16).

Now, it follows from (3), (11), (28), and (29) that,

$$\|z\|_2^2 \le x_0^T P x_0 + \gamma^2 \|w\|_2^2 \le x_0^T P x_0 + \gamma^2 \overline{w}^2. \tag{34}$$

Thus the $\mathcal{H}_2$ constraint in (13) is satisfied if

$$x_0^T P x_0 + \gamma^2 \overline{w}^2 < \alpha^2. \tag{35}$$

Dividing by α^2, rearranging and using a Schur complement give (17) as an LMI sufficient condition for (13).

To turn (14) and (15) into LMIs, we first show that $x_k^T P x_k \le \alpha^2 \ \forall k > 0$. Since $\overline{K} < 0$, it follows from (3) and (24) that

$$x_{k+1}^T P x_{k+1} - x_k^T P x_k \le \gamma^2 w_k^T w_k. \tag{36}$$

Applying this inequality recursively, we get

$$\begin{aligned} x_k^T P x_k &\le x_0^T P x_0 + \gamma^2 \sum_{j=0}^{k-1} w_j^T w_j \\ &\le x_0^T P x_0 + \gamma^2 \overline{w}^2 < \alpha^2. \end{aligned} \tag{37}$$

It follows that

$$\left\| P^{1/2} x_k \right\|^2 < \alpha^2, \tag{38}$$

or equivalently,

$$x_k^T Q^{-1} x_k < 1, \quad \forall k > 0. \tag{39}$$

Next, we transform the constraints in (14) to a set of LMIs. Setting $F = YQ^{-1} = YP\alpha^{-2}$ and $u_k = Fx_k$,

$$\begin{aligned} e_j(u_k) &:= u_k^T H_j u_k + 2h_j^T u_k - \overline{u}_j \\ &= x_k^T Q^{-1} Y^T H_j Y Q^{-1} x_k + 2h_j^T Y Q^{-1} x_k - \overline{u}_j. \end{aligned} \tag{40}$$

Now for any $\mu_j \in \mathcal{R}$, we can write

$$\begin{aligned} e_j(u_k) = -\mu_j\left(1 - x_k^T Q^{-1} x_k\right) - \begin{bmatrix} x_k \\ 1 \end{bmatrix}^T \\ \times \left(\begin{bmatrix} \mu_j Q^{-1} & -Q^{-1} Y^T H_j Y Q^{-1} & -Q^{-1} Y^T h_j \\ & -h_j^T Y Q^{-1} & -\mu_j + \overline{u}_j \end{bmatrix} \right) \\ \times \begin{bmatrix} x_k \\ 1 \end{bmatrix}. \end{aligned} \tag{41}$$

Therefore a sufficient condition for $e_j(u_k) \leq 0$ is $\mu_j \geq 0$ and

$$\begin{bmatrix} \mu_j Q^{-1} & -Q^{-1}Y^T H_j Y Q^{-1} & -Q^{-1}Y^T h_j \\ & -h_j^T Y Q^{-1} & \overline{u}_j - \mu_j \end{bmatrix} \geq 0. \tag{42}$$

Pre- and post-multiplying by $\mathrm{diag}(Q, I)$ gives a bilinear matrix inequality and applying a Schur complement, we get

$$\begin{bmatrix} \mu_j Q & Y^T H_j^{1/2} & -Y^T h_j \\ H_j^{1/2} Y & I & 0 \\ -h_j^T Y & 0 & \overline{u}_j - \mu_j \end{bmatrix} \geq 0. \tag{43}$$

Pre- and post-multiplying the above bilinear matrix inequality by $\mathrm{diag}(\mu_j^{-1/2}, \mu_j^{1/2}, \mu_j^{1/2})$ and applying a Schur complement, this is equivalent to the LMI in (18).

Finally, to obtain an LMI formulation of the state constraints (15), the following analogous steps are carried out:

$$\begin{aligned} f_j(x_{k+1}) := & \begin{bmatrix} x_k \\ w_k \\ p_k \end{bmatrix}^T \begin{bmatrix} A_{cl}^T \\ B_w^T \\ B_p^T \end{bmatrix} G_j \begin{bmatrix} A_{cl} & B_w & B_p \end{bmatrix} \begin{bmatrix} x_k \\ w_k \\ p_k \end{bmatrix} \\ & + 2 g_j^T \begin{bmatrix} A_{cl} & B_w & B_p \end{bmatrix} \begin{bmatrix} x_k \\ w_k \\ p_k \end{bmatrix} - \overline{x}_j. \end{aligned} \tag{44}$$

Now for any $\nu_j,\ \rho_j \in \mathcal{R}$, we can write

$$\begin{aligned} f_j(x_{k+1}) = & -\nu_j \left(1 - x_k^T Q^{-1} x_k\right) - \rho_j \left(\overline{w}^2 - w_k^T w_k\right) \\ & - \begin{bmatrix} x_k \\ w_k \\ 1 \end{bmatrix}^T \left(\begin{bmatrix} \nu_j Q^{-1} - \left(A_{cl} + B_p \Delta_k C_{pw}\right)^T G_j \left(A_{cl} + B_p \Delta_k C_{pw}\right) & \star & \star \\ -\left(B_w + B_p \Delta_k D_{qw}\right)^T G_j \left(A_{cl} + B_p \Delta_k C_{pw}\right) & \rho_j I & \star \\ -g_j^T \left(A_{cl} + B_p \Delta_k C_{pw}\right) & -g_j^T \left(B_w + B_p \Delta_k D_{qw}\right) & -\nu_j - \rho_j \overline{w}^2 + \overline{x}_j \end{bmatrix} \right) \begin{bmatrix} x_k \\ w_k \\ 1 \end{bmatrix}. \end{aligned} \tag{45}$$

Therefore a sufficient condition for $f_j(x_{k+1}) \geq 0$ is $\nu_j \geq 0$, $\rho_j \geq 0$ and

$$\begin{aligned} & \begin{bmatrix} \nu_j Q^{-1} & \star & \star \\ 0 & \rho_j I & \star \\ -g_j^T \left(A_{cl} + B_p \Delta_k C_{pw}\right) & -g_j^T \left(B_w + B_p \Delta_k D_{qw}\right) & \overline{x}_j - \nu_j - \rho_j \overline{w}^2 \end{bmatrix} \\ & - \begin{bmatrix} \left(A_{cl} + B_p \Delta_k C_{pw}\right)^T \\ \left(B_w + B_p \Delta_k D_{qw}\right)^T \\ 0 \end{bmatrix} G_j \begin{bmatrix} \left(A_{cl} + B_p \Delta_k C_{pw}\right) & \left(B_w + B_p \Delta_k D_{qw}\right) & 0 \end{bmatrix} \geq 0. \end{aligned} \tag{46}$$

Applying Schur complement to the above equation, we get

$$\begin{bmatrix} \nu_j Q^{-1} & \star & \star & \star \\ 0 & \rho_j I & \star & \star \\ -g_j^T \left(A_{cl} + B_p \Delta_k C_{pw}\right) & -g_j^T \left(B_w + B_p \Delta_k D_{qw}\right) & \overline{x}_j - \nu_j - \rho_j \overline{w}^2 & \star \\ G_j^{1/2} \left(A_{cl} + B_p \Delta_k C_{pw}\right) & G_j^{1/2} \left(B_w + B_p \Delta_k D_{qw}\right) & 0 & I \end{bmatrix} \geq 0. \tag{47}$$

When Δ is structured we proceed as follows. For norm-bounded uncertainty, we first separate the terms involving modeling uncertainties from the other terms as

$$\underbrace{\begin{bmatrix} \nu_j Q^{-1} & \star & \star & \star \\ 0 & \rho_j I & \star & \star \\ -g_j^T A_{cl} & -g_j^T B_w & \overline{x}_j - \nu_j - \rho_j \overline{w}^2 & \star \\ -G_j^{1/2} A_{cl} & -G_j^{1/2} B_w & 0 & I \end{bmatrix}}_{-T_1} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ -g_j^T B_p \\ G_j^{1/2} B_p \end{bmatrix}}_{-T_2} \Delta_k \underbrace{\begin{bmatrix} C_{pw} & D_{qw} & 0 & 0 \end{bmatrix}}_{T_3} + \underbrace{\begin{bmatrix} C_{pw}^T \\ D_{qw}^T \\ 0 \\ 0 \end{bmatrix}}_{T_3^T} \Delta_k^T \underbrace{\begin{bmatrix} 0 & 0 & -B_p^T g_j & B_p^T G^{1/2} \end{bmatrix}}_{-T_2^T} \geq 0. \tag{48}$$

Equation (48) is equivalent to $-T_1 - T_2 \Delta T_3 - T_3^T \Delta^T T_2^T > 0$, where $T_4 = 0$. By using (8) from Lemma 1, we have

$$\begin{bmatrix} -T_1 - T_3^T S T_3 & -T_2 \\ -T_2^T & S \end{bmatrix} > 0. \tag{49}$$

Applying Schur complement to (49), we get

$$\begin{bmatrix} -T_1 & T_3^T & -T_2 \\ T_3 & S^{-1} & 0 \\ -T_2^T & 0 & S \end{bmatrix} > 0. \tag{50}$$

Substituting the variables from (48) into (50) and swapping the third and sixth diagonal elements, we get

$$\begin{bmatrix} \nu_j Q^{-1} & \star & \star & \star & \star & \star \\ 0 & \rho_j I & \star & \star & \star & \star \\ 0 & 0 & S & \star & \star & \star \\ G_j^{1/2} A_{cl} & G_j^{1/2} B_w & G_j^{1/2} B_p & I & \star & \star \\ C_{pw} & D_{qw} & 0 & 0 & S^{-1} & \star \\ -g_j^T A_{cl} & -g_j^T B_w & -g_j^T B_p & 0 & 0 & \overline{x}_j - \nu_j - \rho_j \overline{w}^2 \end{bmatrix} \geq 0. \tag{51}$$

Pre- and post-multiplying by $\mathrm{diag}(Q, I, I, I, I)$ gives

$$\begin{bmatrix} \nu_j Q & \star & \star & \star & \star & \star \\ 0 & \rho_j I & \star & \star & \star & \star \\ 0 & 0 & S & \star & \star & \star \\ G_j^{1/2} A_{cl} & G_j^{1/2} B_w & G_j^{1/2} B_p & I & \star & \star \\ C_{pw} & D_{qw} & 0 & 0 & S^{-1} & \star \\ -g_j^T A_{cl} & -g_j^T B_w & -g_j^T B_p & 0 & 0 & \overline{x}_j - \nu_j - \rho_j \overline{w}^2 \end{bmatrix} \geq 0. \tag{52}$$

The above equation is bilinear and thus we pre- and post-multiply it by $\mathrm{diag}(\nu_j^{-1/2}, \nu_j^{1/2}, \nu_j^{1/2}, \nu_j^{1/2}, \nu_j^{1/2}, \nu_j^{1/2})$ to obtain

$$\begin{bmatrix} Q & \star & \star & \star & \star & \star \\ 0 & \nu_j \rho_j I & \star & \star & \star & \star \\ 0 & 0 & \nu_j S & \star & \star & \star \\ G_j^{1/2} A_{cl} & \nu_j G_j^{1/2} B_w & \nu_j G_j^{1/2} B_p & \nu_j I & \star & \star \\ C_{pw} & \nu_j D_{qw} & 0 & 0 & \nu_j S^{-1} & \star \\ -g_j^T A_{cl} & -\nu_j g_j^T B_w & -\nu_j g_j^T B_p & 0 & 0 & \nu_j \overline{x}_j - \nu_j^2 - \nu_j \rho_j \overline{w}^2 \end{bmatrix} \geq 0. \tag{53}$$

From the above equation, we can see that the variable S and its inverse appear in the matrix inequality, thus to make it uniform, we pre- and post-multiply by $\mathrm{diag}(I, I, S^{-1}, I, I, I)$ to get

$$\begin{bmatrix} Q & \star & \star & \star & \star & \star \\ 0 & \nu_j\rho_j I & \star & \star & \star & \star \\ 0 & 0 & \nu_j S^{-1} & \star & \star & \star \\ G_j^{1/2}A_{cl} & \nu_j G_j^{1/2}B_w & \nu_j G_j^{1/2}B_p S^{-1} & \nu_j I & \star & \star \\ C_{pw} & \nu_j D_{qw} & 0 & 0 & \nu_j S^{-1} & \star \\ -g_j^T A_{cl} & -\nu_j g_j^T B_w & -\nu_j g_j^T B_p S^{-1} & 0 & 0 & \nu_j\overline{x}_j - \nu_j^2 - \nu_j\rho_j\overline{w}^2 \end{bmatrix} \geq 0. \tag{54}$$

Thus the above equation is a nonlinear matrix inequality in ν_j^2 and bilinear in $\nu_j\rho_j$ and $\nu_j S^{-1}$, hence we define new variables $\overline{\Psi}_j = \nu_j S^{-1}$ and $\delta_j = \nu_j\rho_j$ and finally applying a Schur complement, we obtain the LMI of (19). □

Remark 5. The input and state constraints used in this paper are more general than those used in [9] in that we allow linear terms and so this makes it possible to include asymmetric or hyperplane constraints.

Remark 6. When there is no uncertainty, the problem reduces to disturbance rejection technique considered in [9].

Remark 7. When there is no disturbance, the results reduce to those of [7].

Remark 8. The method used in this paper guarantees recursive feasibility (see [15, Chapter 4]). Also see [16] for a different approach.

4. Numerical Examples

In this section, we present two examples that illustrate the implementation of the proposed scheme. In the first example we consider a solenoid system, and in the second example we consider the coupled spring-mass system. The solution to the linear objective minimization was computed using LMI Control Toolbox in the MATLAB® environment and α^2 was set as a variable.

4.1. Example 1. We consider a modified version of the solenoid system adapted from [17]. The system (see Figure 1) consists of a central object wrapped with coil and is attached to a rigid surface via a spring and damper, which forms a passive vibration isolator. The solenoid is one of the common actuator components. The basic principle of operation involves a moving ferrous core (a piston) that moves inside a wire coil. Normally, the piston is held outside the core by a spring and damper. When a voltage is applied to the coil and current flows, the coil builds up a magnetic field that attracts the piston and pulls it into the center of the coil. The piston can be used to supply a linear force. Application of this includes pneumatic valves and car door openers.

The system is modeled by

$$\begin{aligned} \begin{bmatrix} x_{k+1}^1 \\ x_{k+1}^2 \end{bmatrix} &= \begin{bmatrix} 0.6148 & 0.0315 \\ -0.3155 & -0.0162 \end{bmatrix} \begin{bmatrix} x_k^1 \\ x_k^2 \end{bmatrix} + \begin{bmatrix} 0.0385 \\ 0.0315 \end{bmatrix} u_k \\ &\quad + \begin{bmatrix} 0.00385 \\ 0.00315 \end{bmatrix} w_k + \begin{bmatrix} 0 \\ 10 \end{bmatrix} p_k, \\ q_k &= C_q x_k + D_{qu} u_k + D_{qw} w_k, \\ p_k &= \Delta q_k, \\ z_k &= \begin{bmatrix} C_z x_k \\ D_{zu} u_k \end{bmatrix}, \end{aligned} \tag{55}$$

where

$$C_q = \begin{bmatrix} 1 & 0 \end{bmatrix}, \qquad D_{qu} = 1, \qquad D_{qw} = 0, \tag{56}$$

where x^1 and x^2 are the position and the velocity of the plate. The cost function is specified using $C_z = \text{diag}(1, 1)$ and $D_{zu} = 10$. The magnetic force u is the control variable, and w is the external disturbance to the system, which is bounded in the range $[-1, 1]$. The initial state is given as $x_0 = [1 \quad 0]^T$.

We choose $\gamma^2 = 0.01$ and $\gamma^2 = 1$. Figures 2 and 3 compare the closed-loop response for the high and low disturbance rejection levels, respectively, for randomly generated Δ's. The optimization is feasible, the response is stable, and the performance is good. A control constraint of $|u_k| \leq 0.5$ is imposed, which is satisfied. The computation time for 100 samples was about 10 s, making 0.1 s per sample.

4.2. Example 2. We revisit a modified version of Example 2 reported in [7]. The system consists of a two-mass-spring model whose discrete-time equivalent is obtained using Euler first-order approximation with a sampling time of 0.1 s.

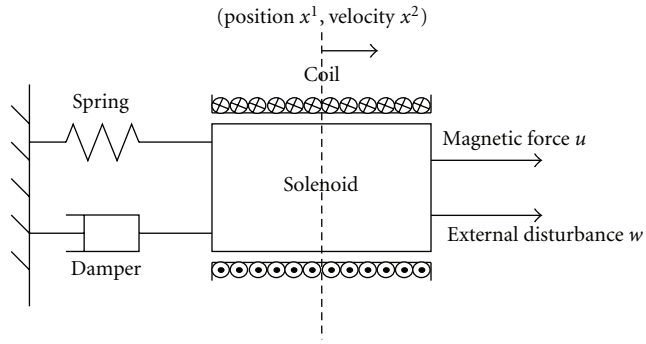


Figure 1: Solenoid system.

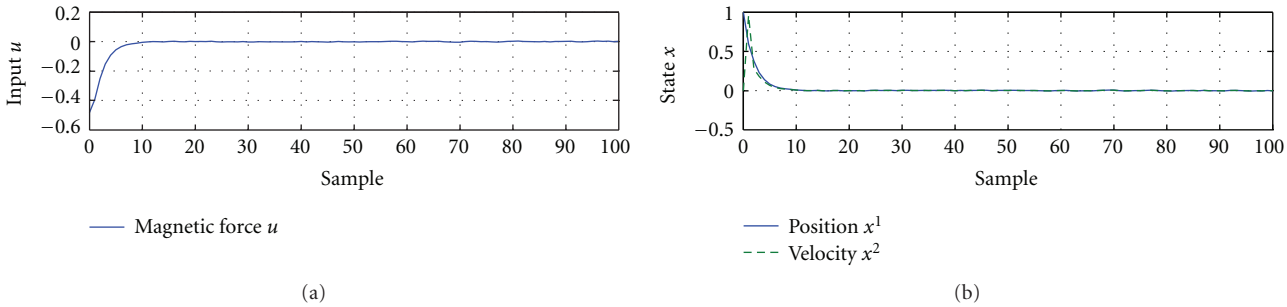


Figure 2: Closed-loop response of the solenoid system with $\gamma^2 = 0.01$.

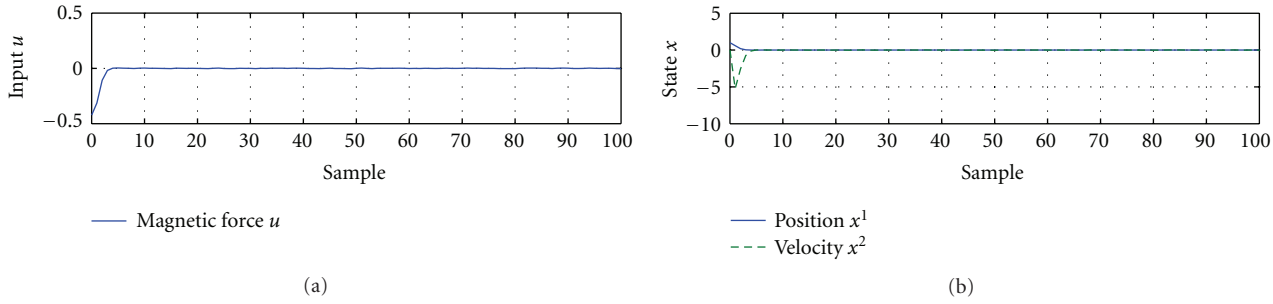


Figure 3: Closed-loop response of the solenoid system with $\gamma^2 = 1$.

The model in terms of disturbance and perturbation variables is

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 1 & 0 & 0.1 & 0 \\ 0 & 1 & 0 & 0.1 \\ -0.1\dfrac{K}{m_1} & 0.1\dfrac{K}{m_1} & 1 & 0 \\ 0.1\dfrac{K}{m_2} & -0.1\dfrac{K}{m_2} & 1 & 0 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 0 \\ \dfrac{0.1}{m_1} \\ 0 \end{bmatrix} u_k \\ &\quad + B_w w_k + B_p p_k, \\ q_k &= C_q x_k + D_{qu} u_k + D_{qw} w_k, \\ p_k &= \Delta q_k, \\ y_k &= \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} x_k, \end{aligned} \tag{57}$$

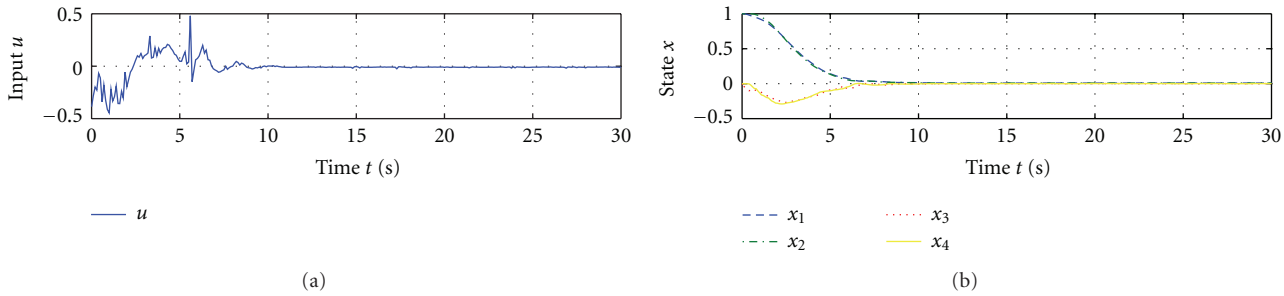
where

$$B_w = \begin{bmatrix} 0 \\ 0.01 \\ 0 \\ 0 \end{bmatrix}, \qquad B_p = \begin{bmatrix} 0 \\ 0 \\ -0.1 \\ 0.1 \end{bmatrix}, \tag{58}$$

$$C_q = \begin{bmatrix} 0.475 & -0.475 & 0 & 0 \end{bmatrix},$$

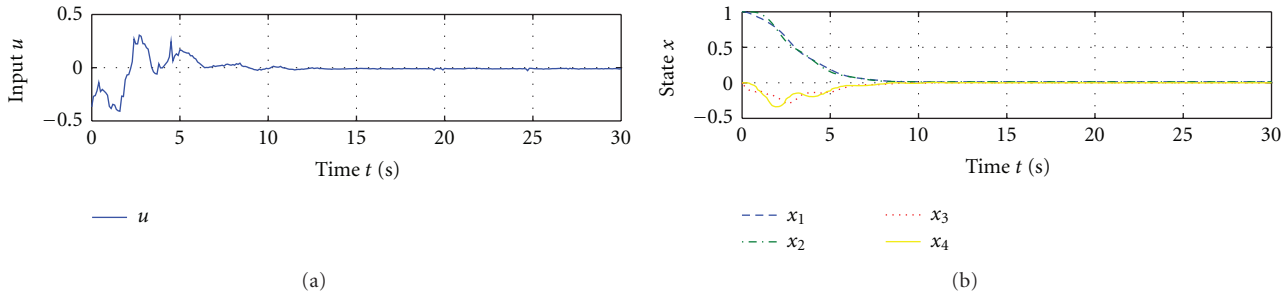
$$D_{qw} = 0, \qquad D_{qu=0},$$

where x_1 and x_2 are the positions of body 1 and 2, and x_3 and x_4 are their velocities, respectively. m_1 and m_2



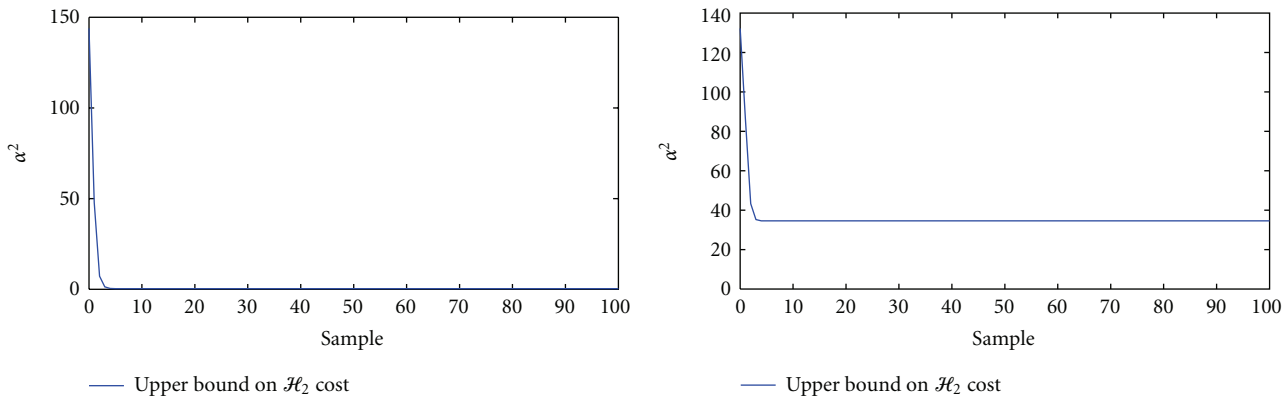
(a) (b)

Figure 4: Closed-loop response of the coupled spring-mass system with $\gamma^2 = 0.45$.



(a) (b)

Figure 5: Closed-loop response of the coupled spring-mass system with $\gamma^2 = 4$.



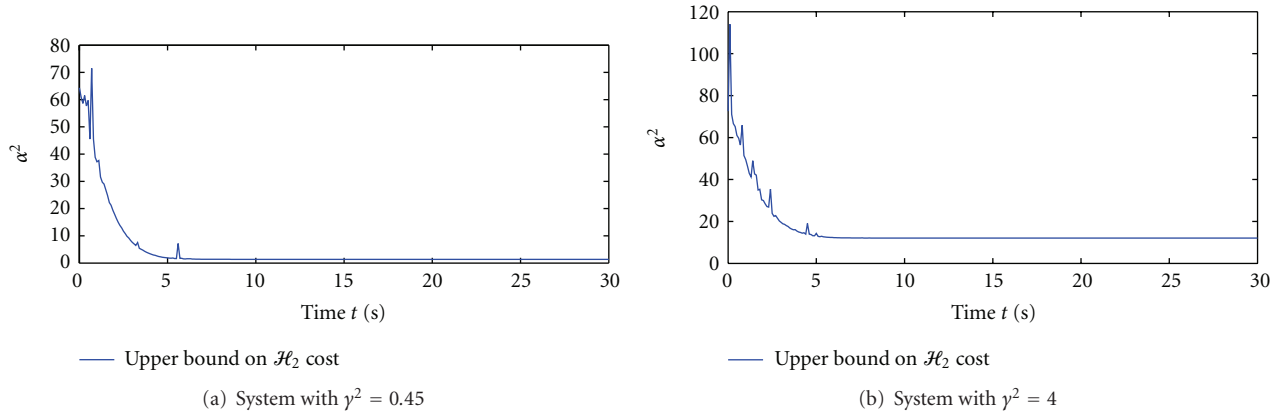
(a) System with $\gamma^2 = 0.01$ (b) System with $\gamma^2 = 1$

Figure 6: Upper bound on $\mathcal{H}_2$ cost function for solenoid system.

are the masses of the two bodies and K is the spring constant. The initial state is given as $x_0 = [1 \ 1 \ 0 \ 0]^T$. The cost function is specified using $C_Z = \text{diag}(1, 1, 1, 1)$, and $D_{zu} = 1$. We consider the system with $m_1 = m_2 = 1$ and $K \in [0.5, 10]$.

A persistent disturbance of the form $w_i = 0.1$ for all sample time was considered. Here we set $\gamma^2 = 0.45$ and $\gamma^2 = 4$. Figures 4 and 5 compare the closed-loop response for the high and low disturbance rejection levels, respectively, for randomly generated Δ's. The value of γ^2 for high disturbance rejection was the lowest value for which a feasible solution exists. An input constraint of $|u_k| \leq 1$ is imposed, which is satisfied. The computation time for 300 samples was about 47 s, making 0.16 s per sample.

4.3. Discussion. Note that the performance and response of the systems based on the high disturbance rejection level were better than those obtained using low disturbance rejection level, since the states and control are regulated to smaller steady state values. Constraints on the input were satisfied in both cases; however, the constraints were more conservative with respect to the control signal for the low disturbance rejection level. For example in the solenoid system, the control signal for high disturbance rejection level was 0.4826 and that for low disturbance rejection level was 0.4144. The issue of conservativeness in the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ setting has been considered in [18]. For the systems considered, the upper bound on the $\mathcal{H}_2$ cost function α^2 is depicted in Figures 6 and 7 for the high and low disturbance rejection



(a) System with $\gamma^2 = 0.45$

(b) System with $\gamma^2 = 4$

FIGURE 7: Upper bound on $\mathcal{H}_2$ cost function for coupled spring-mass system.

levels. It can been seen that the performance coefficient obtained for the high and low disturbance rejection levels is small. However, the offset level on the low disturbance level is higher than that for the high disturbance level. This is due to the higher value of γ^2 in $x_0^T P x_0 + \gamma^2 \overline{w}^2$. In Example 2, we have considered uncertainty in the model by using variable spring constant K.

5. Conclusion

In this paper, we proposed a robust model predictive control design technique using mixed $\mathcal{H}_2/\mathcal{H}_\infty$ for time invariant discrete-time linear systems subject to constraints on the inputs and states. This method takes account of disturbances naturally by imposing the $\mathcal{H}_\infty$-norm constraint in (14) and thus extends the work in [9] by the introduction of structured, norm-bounded uncertainty. The uncertain system was represented by LFTs. The development is based on full state feedback assumption and the on-line optimization involves the solution of an LMI-based linear objective minimization (convex optimization). Hence, the resulting state-feedback control law minimizes an upper bound on the robust objective function. The new approach reduces to that of [9] when there are no perturbations present in the system and to [7] when there are no disturbances. Thus, we have been able to show that it is possible to handle uncertainty and disturbance in the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ model predictive control design approach. The two examples illustrate the application of the proposed method.

Acknowledgments

This was parts supported by Commonwealth Scholarship Commission in the United Kingdom (CSCUK) under Grant NGCS-2004-258. With regard to this paper, particular thanks go to Dr. Imad M. Jaimoukha and Dr. Eric Kerrigan.

References

[1] E. F. Camacho and C. Bordons, *Model Predictive Control*, Springer, 2nd edition, 2004.

[2] J. Richalet, A. Rault, J. L. Testud, and J. Papon, "Model predictive heuristic control. Applications to industrial processes," *Automatica*, vol. 14, no. 5, pp. 413–428, 1978.

[3] P. E. Orukpe, X. Zheng, I. M. Jaimoukha, A. C. Zolotas, and R. M. Goodall, "Model predictive control based on mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control approach for active vibration control of railway vehicles," *Vehicle System Dynamics*, vol. 46, no. 1, pp. 151–160, 2008.

[4] H. S. Witsenhausen, "A minimax control problem for sampled linear systems," *IEEE Transactions on Automatic Control*, vol. 13, no. 1, pp. 5–21, 1968.

[5] R. S. Smith, "Robust model predictive control of constrained linear systems," in *Proceedings of the 2004 American Control Conference (AAC '04)*, pp. 245–250, July 2004.

[6] P. E. Orukpe and I. M. Jaimoukha, "A semidefinite relaxation for the quadratic minimax problem in $\mathcal{H}_\infty$ model predictive control," in *Proceedings of IEEE Conference on Decision and Control*, New Orleans, La, USA, 2007.

[7] M. V. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.

[8] R. S. Smith, "Model predictive control of uncertain constrained linear systems: Lmi-based," Tech. Rep. CUED/F-INFENG/TR.462, University of Cambridge, Cambridge, UK, 2006.

[9] P. E. Orukpe, I. M. Jaimoukha, and H. M. H. El-Zobaidi, "Model predictive control based on mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control approach," in *Proceedings of the American Control Conference (ACC '07)*, pp. 6147–6150, July 2007.

[10] P. E. Orukpe and I. M. Jaimoukha, "Robust model predictive control based on mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control approach," in *Proceedings of the European Control Conference*, pp. 2223–2228, Budapest, Hungary, 2009.

[11] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in Systems and Control Theory*, SIAM Studies in Applied Mathematics, 1994.

[12] A. Packard and J. Doyle, "The complex structured singular value," *Automatica*, vol. 29, no. 1, pp. 71–109, 1993.

[13] K. Sun and A. Packard, "Robust $\mathcal{H}_2$ and $\mathcal{H}_\infty$ filters for uncertain LFT systems," in *Proceedings of the 41st IEEE Conference on Decision and Control*, pp. 2612–2618, Las Vegas, Nev, USA, December 2002.

[14] C. K. Li and R. Mathias, "Extremal characterizations of the Schur complement and resulting inequalities," *SIAM Review*, vol. 42, no. 2, pp. 233–246, 2000.

[15] P. E. Orukpe, *Model predictive control for linear time invariant systems using linear matrix inequality techniques*, Ph.D. thesis, Imperial College, London, UK, 2009.

[16] H. Huang, L. Dewei, and X. Yugeng, "Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ robust model predictive control based on mixed $\mathcal{H}_2/\mathcal{H}_\infty$ with norm-bounded uncertainty," in *Proceedings of the 30th Chinese Control Conference*, pp. 3327–3331, Yantai, China, 2011.

[17] D. Jia, B. H. Krogh, and O. Stursberg, "LMI approach to robust model predictive control," *Journal of Optimization Theory and Applications*, vol. 127, no. 2, pp. 347–365, 2005.

[18] P. E. Orukpe, "Towards a less conservative model predictive control based on mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control approach," *International Journal of Control*, vol. 84, no. 5, pp. 998–1007, 2011.

Hindawi Publishing Corporation
Journal of Control Science and Engineering
Volume 2012, Article ID 159072, 18 pages
doi:10.1155/2012/159072

Research Article

Experimental Application of Predictive Controllers

C. H. F. Silva,[1] H. M. Henrique,[2] and L. C. Oliveira-Lopes[2]

[1] *Cemig Geração e Transmissão SA, Avenida Barbacena 1200, 16° Andar Ala B1 (TE/AE), 30190-131 Belo Horizonte, MG, Brazil*
[2] *Faculdade de Engenharia Química, Universidade Federal de Uberlândia, Avenida João Naves de Ávila, 2121, Bloco 1K do Campus Santa Mônica 38408-100 Uberlândia, MG, Brazil*

Correspondence should be addressed to L. C. Oliveira-Lopes, lcol@ufu.br

Received 31 May 2011; Revised 22 October 2011; Accepted 25 October 2011

Academic Editor: Baocang Ding

Model predictive control (MPC) has been used successfully in industry. The basic characteristic of these algorithms is the formulation of an optimization problem in order to compute the sequence of control moves that minimize a performance function on the time horizon with the best information available at each instant, taking into account operation and plant model constraints. The classical algorithms Infinite Horizon Model Predictive Control (IHMPC) and Model Predictive Control with Reference System (RSMPC) were used for the experimental application in the multivariable control of the pilot plant (level and pH). The simulations and experimental results indicate the applicability and limitation of the control technique.

1. Introduction

The process control is related to the application of the automatic control principles to industrial processes. The globalization effect upon industries brought a perception of the importance associated to the product quality over organization profits. Because of that, the process control has been more and more demanded and explored, in order not only to assure that the process is according to acceptable performance levels but also to address legal requirements in terms of safety and quality of the products [1]. In this context, predictive controllers are able to deal with system requirements in a proper way and simple to be implemented. One of the biggest concerns on control theory is related to the stability of closed loop of systems. It is natural that only stable closed-loop response is considered for real implementation. A representative algorithm of this controller class is Infinite Horizon Model Predictive Control (IHMPC) [2].

During the closed-loop project, there are some theoretical tools that can be incorporated into the controllers and aggregate desirable characteristics. This is the case of predictive controllers that incorporate a reference path in their formulation (RSMPC) [3]. In this case, the value of control action is computed from a QDMC (Quadratic Dynamic Matrix Control) problem that has a first-order reference system incorporated directly into the formulation. Consequently, there is an effort for eliminating undesirable situations of too high speed and for balancing the system response.

Most industrial chemical processes have a nonlinear characteristic, although linear controllers are used in many of these systems. The great advantage of this approximation is having an analytical solution to the control problem and also the low computational complexity. This kind of approach is very common in the industry practice.

The pH system is used as a benchmark for applications in process control, mainly because of its strong nonlinear behavior. For the experimental application case addressed in this work, it is a control of multiple inputs and multiple outputs (MIMO), in which inputs are acid and base flows and outputs are reactor level and stream pH output. This system presents not only a nonlinear feature, but also an interaction between inputs and outputs linked to the system directionality. In this kind of system, classical controllers do not work to keep the stability and required performance. In addition, MPC solves the problem of open loop optimal control in real time, considering the current state of the system to be controlled as much as a policy feedback offline. As can be seen in the experimental data, the multivariable system could not be controlled with optimal classical controllers.

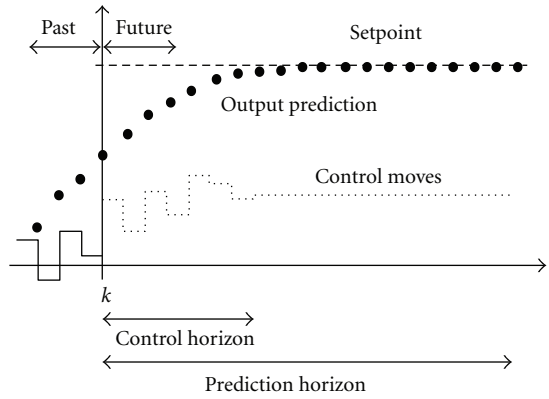


Figure 1: MPC and the receding horizon strategy.

That is why it was chosen to implement IHMPC and RSMPC predictive controllers in real time [4].

Section 2 presents the control algorithms used in the implementation described in this work. In Section 3 there is a description of the experimental system. Section 4 presents the simulations with the controllers and experimental results of the closed loop. Section 5 shows the main conclusions of this paper.

2. Model Predictive Control

The model predictive control (MPC) theory was originated in the end of 1970s decade and developed considerably ever since [5–10]. MPC became an important control strategy of industrial applications, mainly due to great success of its implementations in the petrochemical industry. The reason for MPC being so popular might be due to its ability to deal with quite difficult control challenges, although other characteristics such as its ability to control constrained multivariable plants, have been claimed as one of the most important features [6]. The ability to handle input, output, and internal state constraints can be assigned as the most significant contribution to the many successful industrial applications [11–13]. Other benefits to consider are the possibility of embedding safety limits, the possibility of obtaining advantages in using it in highly nonlinear plants as well as in time-varying plants, the consideration of process noise in the formulation, the high level of control performance, reduced maintenance requirements, and improved flexibility and agility [14].

The various predictive control algorithms differ from each other depending on the way the predictive model is used to represent the process, the noise description, and on the cost function to be minimized. There are many control predictive applications well succeeded not only in chemical industry but also in other areas [15]. The use of state-space models, in spite of other formulations, was responsible for a substantial maturing of the predictive control theory during the 1990s decade [16]. The state-space formulation not only allows the application of linear system theorems already known, but also it makes easier to generalize for more complex cases. In this situation, the MPC controller can be understood as a compensator based on a state observer and its stability, performance, and robustness are determined by the observer poles, established directly by parameter adjustment, and by regulator poles, determined by performance horizons and weights [17]. Figure 1 shows the receding horizon strategy, which is the center of the MPC theory.

Figure 2, based on Richalet [18] and Richards and How [19], shows interfacial areas of the MPC control developments.

Considering its application to MIMO processes, MPC deals directly with coordination and balancing interactions between inputs and outputs and also with associated constraints [20]. In spite of being really effective in suitable situations, MPC still has limitations as, for instance, operation difficulties, high maintenance cost (as it demands specialized work) and flexibility loss, which can result in a controller weakness. These limitations are not connected only to the algorithm itself, but also to the necessity of a plant model, for which is necessary maintenance. In practical terms, the limits of MPC applicability and performance should not be connected to the algorithm deficiencies, but to the issues linked to the modeling difficulties, sensor adequacy, and insufficient robustness in the presence of faults [15, 21].

2.1. IHMPC. The IHMPC formulation was introduced by Muske and Rawlings [2]. For simplicity the algorithm will be presented only for stable systems. For unstable systems see the reference section [2]. The discrete dynamical system used is shown in (1) in which $\mathbf{y}$ is the vector output, $\mathbf{u}$ is the vector input, and $\mathbf{x}$ is the states vector:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{u}_k, \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k. \end{aligned} \tag{1}$$

The receding horizon regulator is based on the minimization of the infinite horizon open-loop quadratic objective function at time k (2). $\mathbf{Q}$ is a symmetric positive semidefinite penalty matrix on the outputs. $\mathbf{R}$ is a symmetric positive semidefinite penalty matrix on the inputs. $\mathbf{S}$ is a symmetric

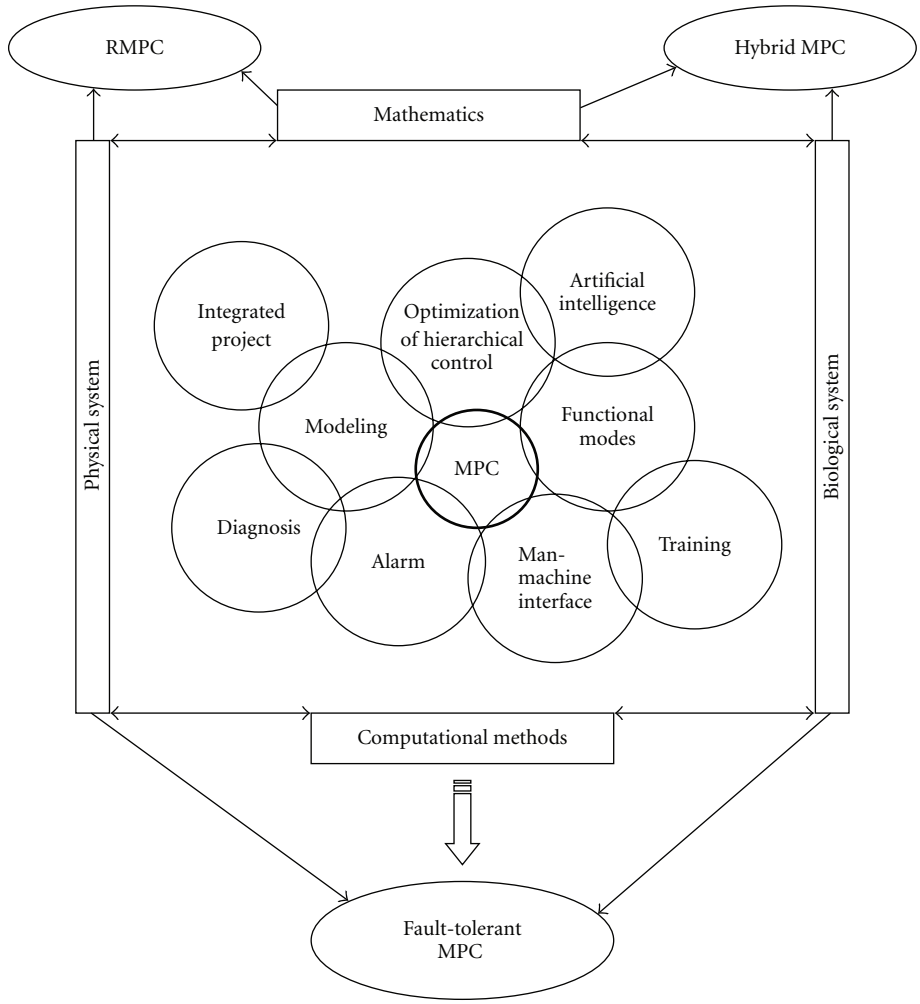


Figure 2: MPC and related areas.

Figure 3: Experimental system.

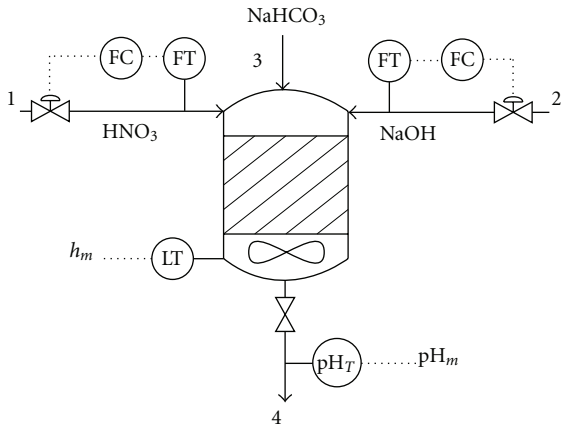


Figure 4: Diagram representation of the experimental system.

positive semidefinite penalty matrix on the rate of the input change with $\Delta\mathbf{u} = \mathbf{u}_{k+i} - \mathbf{u}_k$. The vector $\mathbf{u}^N$ contains the N future open-loop control moves (3). The infinite horizon open-loop objective function can be expressed as a finite horizon open-loop objective. For stable system, $\overline{\mathbf{Q}}$ is defined as the infinite sum (4). This infinite sum can be determined from the solution of the discrete Lyapunov equation (5). Using simple algebraic manipulation, the quadratic objective

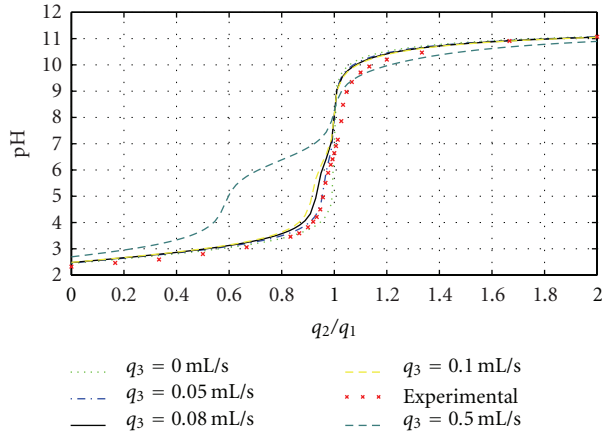


Figure 5: Titration Curves of the neutralization system.

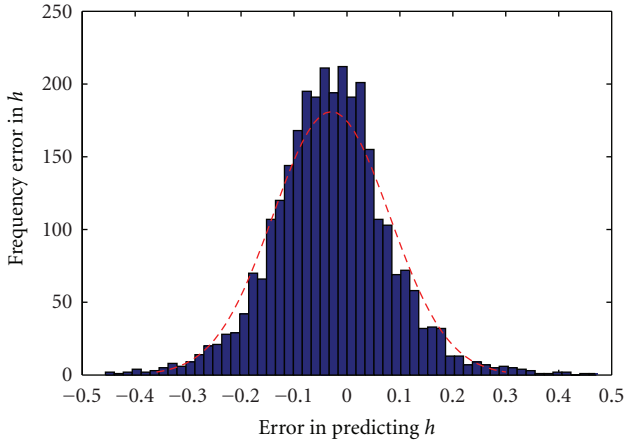


Figure 6: Error in predicting the level of the reactor.

is show in (6). The matrices **H**, **G**, and **F** are showed in (7)–(9):

$$J_k = \min_{\mathbf{u}^N} \sum_{i=0}^{\infty} \mathbf{y}_{k+i}^T \mathbf{Q} \mathbf{y}_{k+i} + \mathbf{u}_{k+i}^T \mathbf{R} \mathbf{u}_{k+i} + \Delta \mathbf{u}_{k+i}^T \mathbf{S} \Delta \mathbf{u}_{k+i}, \quad (2)$$

$$\mathbf{u}^N = \begin{bmatrix} \mathbf{u}_k \\ \mathbf{u}_{k+1} \\ \vdots \\ \mathbf{u}_{k+N-1} \end{bmatrix}, \quad (3)$$

$$\overline{\mathbf{Q}} = \sum_{i=0}^{\infty} \mathbf{A}^{T^i} \mathbf{C}^T \mathbf{Q} \mathbf{C} \mathbf{A}^i, \quad (4)$$

$$\overline{\mathbf{Q}} = \mathbf{C}^T \mathbf{Q} \mathbf{C} + \mathbf{A}^T \overline{\mathbf{Q}} \mathbf{A}, \quad (5)$$

$$\min_{\mathbf{u}^N} \Phi_k = \min_{\mathbf{u}^N} \mathbf{u}^{N^T} \mathbf{H} \mathbf{u}^N + 2\mathbf{u}^{N^T} (\mathbf{G}\mathbf{x}_k - \mathbf{F}\mathbf{u}_{k-1}), \quad (6)$$

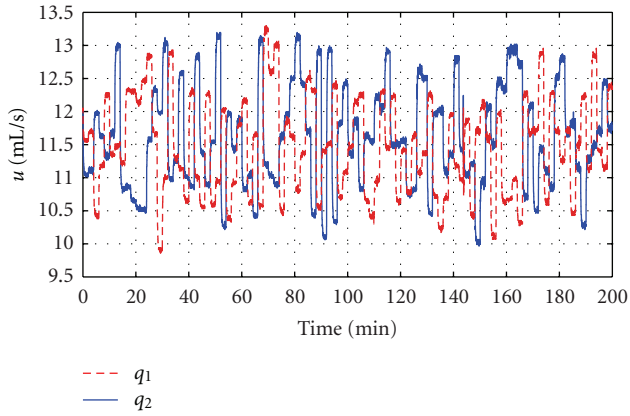


Figure 7: Random inputs for parameter estimation.

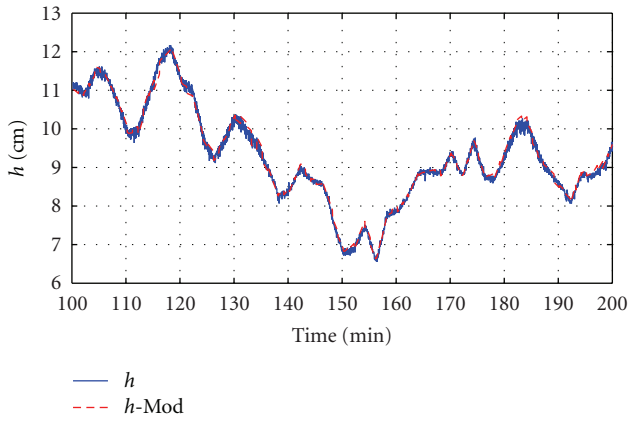


Figure 8: Open-loop run behavior (Mod indicates that it is a result of model simulation).

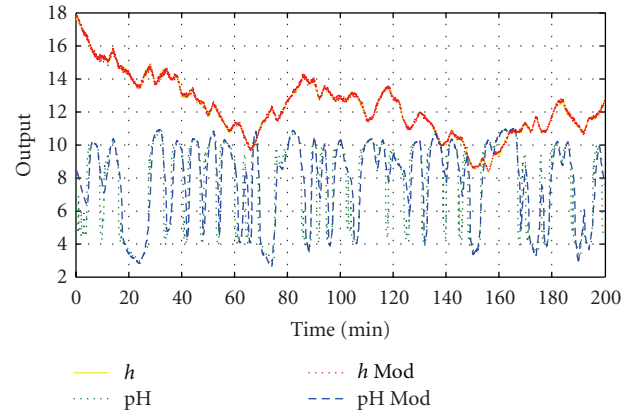


Figure 9: Real data x simulated data (Mod).

$$\mathbf{H} = \begin{bmatrix} \mathbf{B}^T\overline{\mathbf{Q}}\mathbf{B} + \mathbf{R} + 2\mathbf{S} & \mathbf{B}^T\mathbf{A}^T\overline{\mathbf{Q}}\mathbf{B} - \mathbf{S} & \cdots & \mathbf{B}^T\mathbf{A}^{T^{N-1}}\overline{\mathbf{Q}}\mathbf{B} \\ \mathbf{B}^T\overline{\mathbf{Q}}\mathbf{A}\mathbf{B} - \mathbf{S} & \mathbf{B}^T\overline{\mathbf{Q}}\mathbf{B} + \mathbf{R} + 2\mathbf{S} & \cdots & \mathbf{B}^T\mathbf{A}^{T^{N-2}}\overline{\mathbf{Q}}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{B}^T\overline{\mathbf{Q}}\mathbf{A}^{N-1}\mathbf{B} & \mathbf{B}^T\overline{\mathbf{Q}}\mathbf{A}^{N-2}\mathbf{B} & \cdots & \mathbf{B}^T\overline{\mathbf{Q}}\mathbf{B} + \mathbf{R} + 2\mathbf{S} \end{bmatrix}, \quad (7)$$

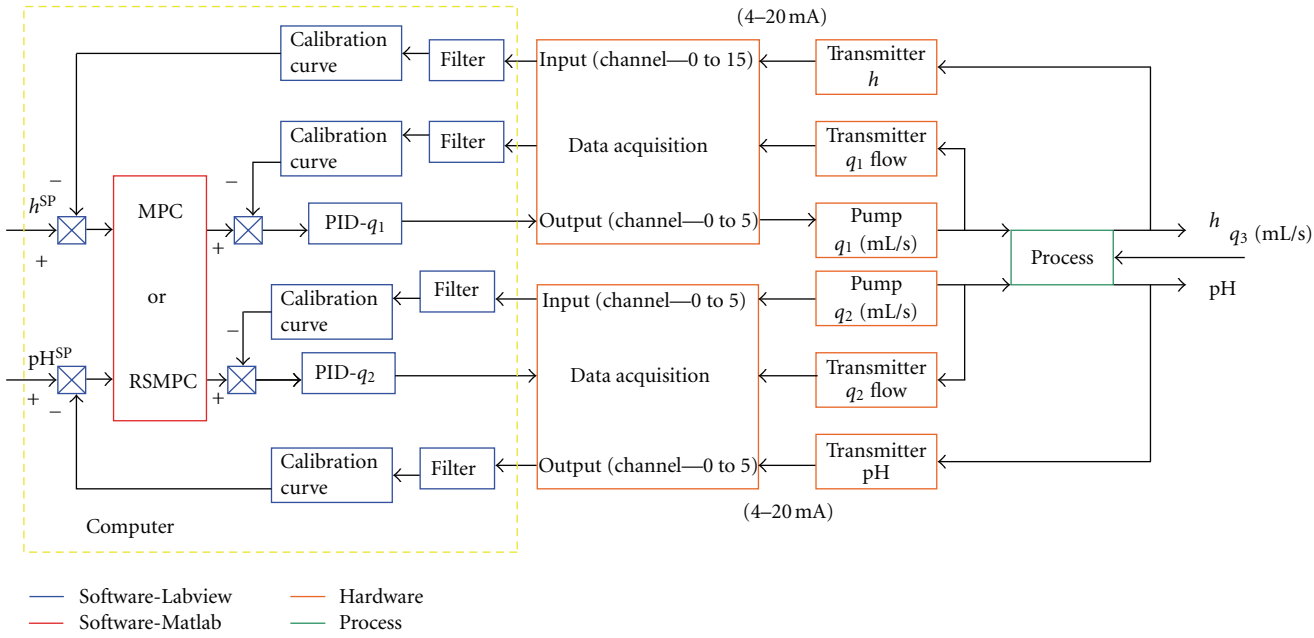


Figure 10: Structure of operation—experimental plant.

$$\mathbf{G} = \begin{bmatrix} \mathbf{B}^T\overline{\mathbf{Q}}\mathbf{A} \\ \mathbf{B}^T\overline{\mathbf{Q}}\mathbf{A}^2 \\ \vdots \\ \mathbf{B}^T\overline{\mathbf{Q}}\mathbf{A}^N \end{bmatrix}, \tag{8}$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{S} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}. \tag{9}$$

2.2. RSMPC. The RSMPC strategy is basically a transformation of an optimization problem with constraints into a quadratic programming problem [3]. The RSMPC development is presented next. Consider a general system described by (10). By linearizing (10) with the Taylor's series expansion around the point immediately before the current operational point, one obtains (11). The matrices are defined in (12)–(16). Equation (11) is integrated from t to $t + \Delta t$ and by assuming $\mathbf{u}_k$ constant between samples, one can write (17):

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \qquad \mathbf{y} = \mathbf{h}(\mathbf{x}), \tag{10}$$

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}_{k-1}\mathbf{x} + \mathbf{B}_{k-1}\mathbf{u} + \mathbf{f}_{k-1}, \qquad \mathbf{y} = \mathbf{C}_{k-1}\mathbf{x} + \mathbf{h}_{k-1}, \tag{11}$$

$$\mathbf{A}_{k-1} = \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)\Big|_{\mathbf{x}=\mathbf{x}_{k-1},\, \mathbf{u}=\mathbf{u}_{k-1}}, \tag{12}$$

$$\mathbf{B}_{k-1} = \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right)\Big|_{\mathbf{x}=\mathbf{x}_{k-1},\, \mathbf{u}=\mathbf{u}_{k-1}}, \tag{13}$$

$$\mathbf{C}_{k-1} = \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right)\Big|_{\mathbf{x}=\mathbf{x}_{k-1}}, \tag{14}$$

$$\mathbf{f}_{k-1} = \mathbf{f}(\mathbf{x}, \mathbf{u})|_{\mathbf{x}=\mathbf{x}_{k-1},\, \mathbf{u}=\mathbf{u}_{k-1}}, \tag{15}$$

$$\mathbf{h}_{k-1} = \mathbf{h}|_{\mathbf{x}=\mathbf{x}_{k-1}}, \tag{16}$$

$$\begin{aligned} \hat{\mathbf{x}}_{k+1} &= \mathbf{\Phi}\mathbf{x}_k + \mathbf{\Psi}\mathbf{B}\mathbf{u}_k + \mathbf{\Psi}\mathbf{f}_{k-1}, \\ \hat{\mathbf{y}}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{h}_{k-1}, \end{aligned} \tag{17}$$

where

$$\mathbf{\Phi} = e^{\mathbf{A}\Delta t}, \quad \mathbf{\Psi} = \mathbf{A}^{-1}\left(e^{\mathbf{A}\Delta t} - \mathbf{I}\right)\mathbf{B}, \quad \mathbf{\Omega} = \mathbf{A}^{-1}\left(e^{\mathbf{A}\Delta t} - \mathbf{I}\right). \tag{18}$$

Since A is not singular, then (17) can be written for each prediction instant from $k = 1$ to $k = P$, where P is the prediction horizon and N is the control horizon, with $P \geq N$ and $\Delta\mathbf{u}_{k+j} = \mathbf{0}$ to $N \leq j \leq P$. The resulting set of equation is showed below, where

$$\hat{\mathbf{y}} = \mathbf{\Gamma}\Delta\mathbf{u} + \boldsymbol{\gamma}, \tag{19}$$

$$\hat{\mathbf{y}} = \left[\hat{\mathbf{y}}_{k+1} \;\; \hat{\mathbf{y}}_{k+2} \;\; \cdots \;\; \hat{\mathbf{y}}_{k+N} \;\; \cdots \;\; \hat{\mathbf{y}}_{k+P}\right], \tag{20}$$

$$\Delta\mathbf{u} = \left[\Delta\mathbf{u}_{k+1} \;\; \Delta\mathbf{u}_{k+2} \;\; \cdots \;\; \Delta\mathbf{u}_{k+N}\right]. \tag{21}$$

The controller is designed in order to transform the closed-loop behavior in a fist-order system behavior according to (24). To overcome the problem of infeasible solutions of the MPC optimization problem, a slack variable ($\boldsymbol{\lambda}$) is introduced in order to allow the system to deviate from reference system and to satisfy hard constraints [3] resulting in (25). This equation can be written for each prediction instant (26):

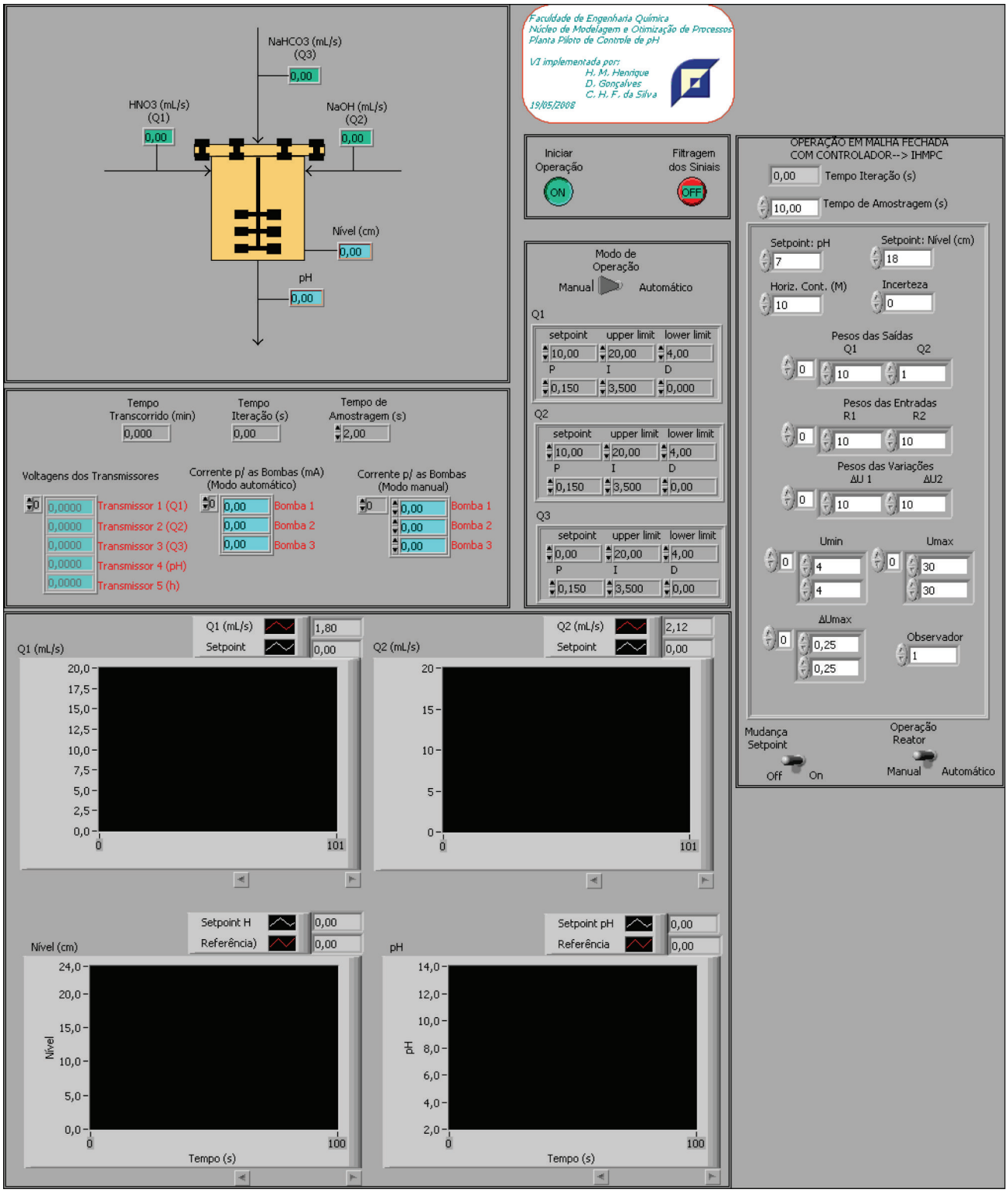


Figure 11: Frontal panel—IHMPC.

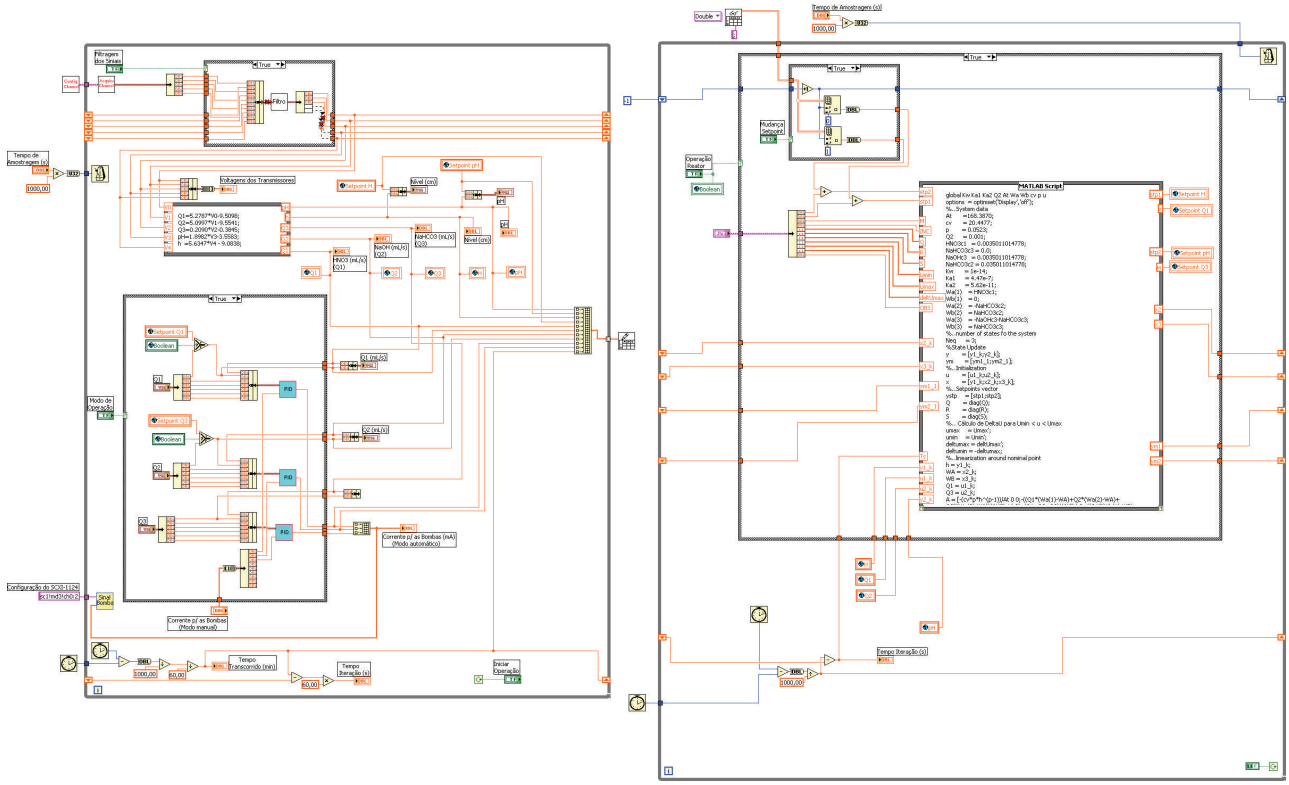


Figure 12: Block diagram—IHMPC.

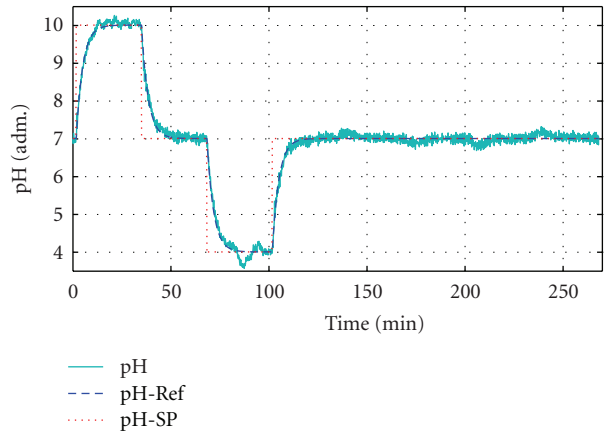


Figure 13: Closed-loop simulation: RSMPC—pH (setpoint: SP and reference system: Ref).

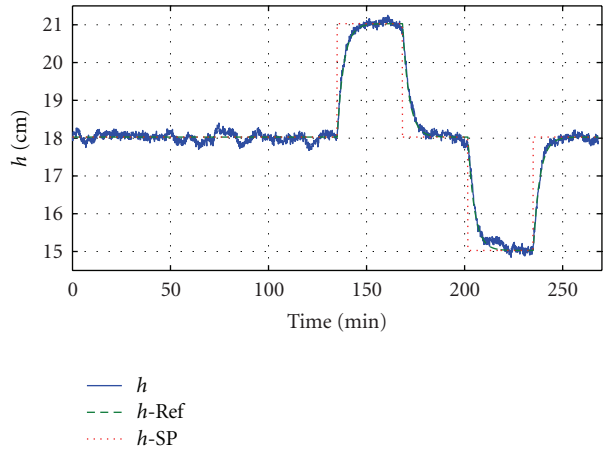


Figure 14: Closed-loop simulation: RSMPC—h (setpoint: SP and reference system: Ref).

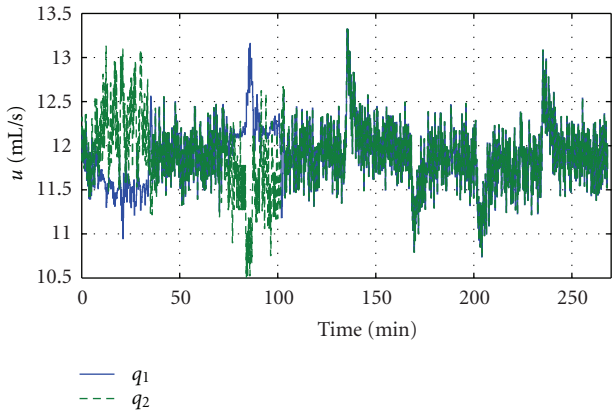


Figure 15: Closed-loop simulation: RSMPC—control moves.

$$\boldsymbol{\Gamma} = \begin{bmatrix} \mathbf{C}\boldsymbol{\Psi} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \mathbf{C}(\boldsymbol{\Phi}+\mathbf{I})\boldsymbol{\Psi} & \mathbf{C}\boldsymbol{\Psi} & \cdots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \mathbf{C}\left(\sum_{i=1}^{N}\boldsymbol{\Phi}^{i-1}\right)\boldsymbol{\Psi} & \mathbf{C}\left(\sum_{i=1}^{N-1}\boldsymbol{\Phi}^{i-1}\right)\boldsymbol{\Psi} & \cdots & \mathbf{C}\boldsymbol{\Psi} \\ \vdots & \vdots & & \vdots \\ \mathbf{C}\left(\sum_{i=1}^{P}\boldsymbol{\Phi}^{i-1}\right)\boldsymbol{\Psi} & \mathbf{C}\left(\sum_{i=1}^{P-1}\boldsymbol{\Phi}^{i-1}\right)\boldsymbol{\Psi} & \cdots & \mathbf{C}\left(\sum_{i=1}^{P-N}\boldsymbol{\Phi}^{i-1}\right)\boldsymbol{\Psi} \end{bmatrix}, \tag{22}$$

$$\boldsymbol{\gamma} = \begin{bmatrix} \mathbf{y}_k + \mathbf{C}\boldsymbol{\Phi}\Delta\mathbf{x}_k \\ \mathbf{y}_k + \mathbf{C}\left(\sum_{i=1}^{2}\boldsymbol{\Phi}^{i}\right)\Delta\mathbf{x}_k \\ \vdots \\ \mathbf{y}_k + \mathbf{C}\left(\sum_{i=1}^{N}\boldsymbol{\Phi}^{i}\right)\Delta\mathbf{x}_k \\ \vdots \\ \mathbf{y}_k + \mathbf{C}\left(\sum_{i=1}^{P}\boldsymbol{\Phi}^{i}\right)\Delta\mathbf{x}_k \end{bmatrix}, \tag{23}$$

$$\frac{d\mathbf{y}}{dt} = \mathbf{K}\left(\mathbf{y}^{\mathrm{SP}} - \mathbf{y}\right), \tag{24}$$

$$\mathbf{C}\mathbf{f}_{k-1} + \mathbf{C}\mathbf{A}_{k-1}\mathbf{x}_k + \mathbf{C}\mathbf{B}_{k-1}\mathbf{u}_k + \boldsymbol{\lambda}_k = \mathbf{K}\left(\mathbf{y}^{\mathrm{SP}} - \mathbf{y}\right), \tag{25}$$

$$\mathbf{D}\boldsymbol{v} = \mathbf{b}, \tag{26}$$

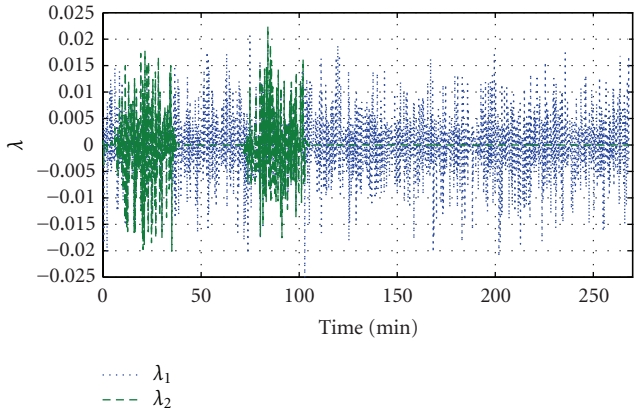


Figure 16: Closed-loop simulation: RSMPC—slack variables.

where

$$\boldsymbol{v} = \left[\Delta \mathbf{u}_k \ \cdots \ \Delta \mathbf{u}_{k+N} \ \boldsymbol{\lambda}_k \ \cdots \ \boldsymbol{\lambda}_{k+P}\right], \tag{27}$$

$$\mathbf{b} = \begin{bmatrix} \mathbf{K}\mathbf{y}_k^{\mathrm{SP}} - (\mathbf{CA} + \mathbf{KC})\mathbf{x}_k - \mathbf{C}\mathbf{f}_{k-1} \\ \mathbf{K}\mathbf{y}_{k+1}^{\mathrm{SP}} - (\mathbf{CA} + \mathbf{KC})\boldsymbol{\Phi}\mathbf{x}_k \\ \mathbf{K}\mathbf{y}_{k+2}^{\mathrm{SP}} - (\mathbf{CA} + \mathbf{KC})\boldsymbol{\Phi}^2\mathbf{x}_k \\ \vdots \\ \mathbf{K}\mathbf{y}_{k+P}^{\mathrm{SP}} - (\mathbf{CA} + \mathbf{KC})\boldsymbol{\Phi}^P\mathbf{x}_k \end{bmatrix}, \tag{28}$$

$$\mathbf{D} = \begin{bmatrix} \mathbf{CB} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ (\mathbf{CA} + \mathbf{KC})\boldsymbol{\Psi} & \mathbf{CB} & \cdots & \mathbf{0} & -\mathbf{I} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ (\mathbf{CA} + \mathbf{KC})\boldsymbol{\Psi}\mathbf{C} & (\mathbf{CA} + \mathbf{KC})\boldsymbol{\Psi} & \cdots & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ (\mathbf{CA} + \mathbf{KC})\boldsymbol{\Phi}^{P-1}\boldsymbol{\Psi} & (\mathbf{CA} + \mathbf{KC})\boldsymbol{\Phi}^{P-2}\boldsymbol{\Psi} & \cdots & (\mathbf{CA} + \mathbf{KC})\boldsymbol{\Phi}^{P-M-1}\boldsymbol{\Psi} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} \end{bmatrix}. \tag{29}$$

The cost function can be written as (30). This equation can also be reorganized as a quadratic programming problem (31):

$$\min_{\Delta \mathbf{u}(k),\ldots,\Delta \mathbf{u}(k+N_m)\boldsymbol{\lambda}(k),\ldots,\boldsymbol{\lambda}(k+N_p)} J = \frac{1}{2}\left(\Delta \mathbf{u}^T \mathbf{R} \Delta \mathbf{u} + \boldsymbol{\lambda}^T \mathbf{S}\, \boldsymbol{\lambda}\right), \tag{30}$$

$$\begin{aligned} \min_{v(k),\ldots,v(k+N_m+N_p+2)} \quad & J = \frac{1}{2}\boldsymbol{v}^T \boldsymbol{\varepsilon} \boldsymbol{v} \\ \text{subject to:} \quad & \mathbf{D}\boldsymbol{v} = \mathbf{b} \\ & \mathbf{u}_{\min k+j} \le \mathbf{u}_{k+j} \le \mathbf{u}_{\max k+j} \\ & -|\Delta \mathbf{u}_{\min}|_{k+j} \le \Delta \mathbf{u}_{k+j} \le |\Delta \mathbf{u}_{\max}|_{k+j} \\ & j = 0, \ldots, P \end{aligned} \tag{31}$$

in which

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \mathbf{R} & [\mathbf{0}] \\ [\mathbf{0}] & \mathbf{S} \end{bmatrix}. \tag{32}$$

The controller cannot eliminate offset unless (33) is handled [3]:

$$\mathbf{C}\mathbf{f}_{k-1} \approx \frac{\mathbf{y}_{k-}\mathbf{y}_{k-1}}{\Delta t}. \tag{33}$$

3. The Experimental System

The experimental system built was a neutralization process that occurs in a shaken reactor tank. Figure 3 presents a photo of the experimental system.

Figure 4 shows a diagram indicating how the system works. The numbers indicate acid flow (1), base flow (2), buffer flow (3), and output flow (4). This number is used in the indexes of the modeling of the experimental system. The flows of acid (q_1) and base (q_2) are the manipulated variables ($\mathbf{u}$); the flow of buffer solution (q_3) is the unmeasured or measured disturbance ($\mathbf{d}$), depending on how it is treated in the control formulation. For the case of this study, the buffer solution flow was used only to make simulation tests. The level height of the reactor (h) and the output flow pH (q_4) are the controlled variables or the outputs ($\mathbf{y}$).

The system was specially chosen for this study because of its strong nonlinear dynamic behavior and low operational cost, considering the consumption of inputs as electrical energy and raw material [3, 22]. The system becomes interesting under the control perspective, as it is a multivariable and nonlinear process in which the static gain suffers strong variations inside the flow operation band. Figure 5 presents the system titration curve. It can be observed that the process pH changes from 4 to 9 in the [0.87; 1.22] interval of the ratio q_2/q_1. This change in the process gain is just what makes the neutralization problem difficult to solve under the control perspective, as small modifications on the relation q_2/q_1 can cause big or small pH variations, depending on where the operating point is in the operational region. It has to be emphasized that, in an analytical process of titration, only one drop is enough to change the pH from 4 to 9. Thus, it can be inferred that the sensitivity of the dynamic process associated to this offers a challenge problem for control in real time. It can be noticed that, once a buffer solution is added, the nonlinear behavior of the system smoothes up, so that the controller action gets

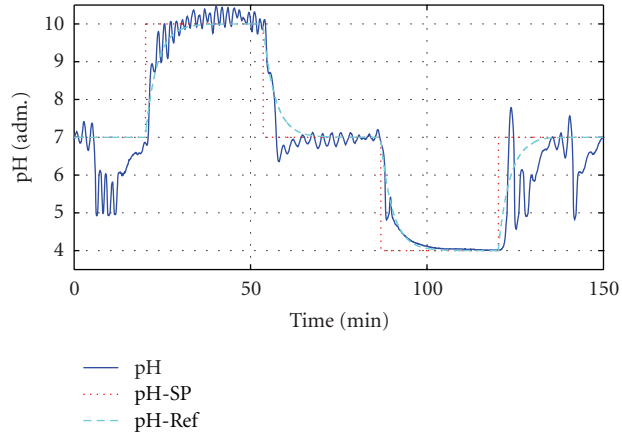


Figure 17: Experimental response RSMPC—pH (setpoint: SP and reference system: Ref).

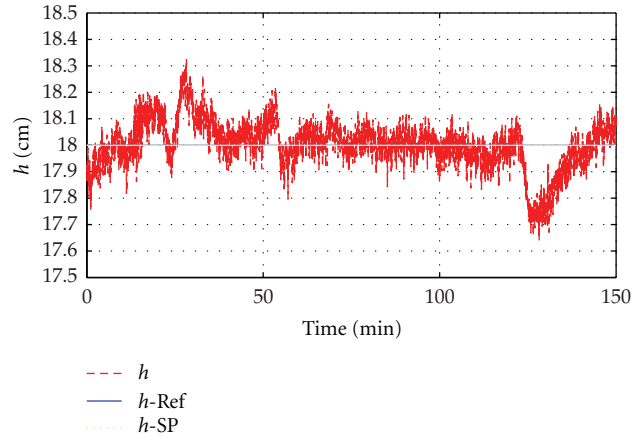


Figure 18: Experimental response: RSMPC—h (setpoint: SP and reference system: Ref).

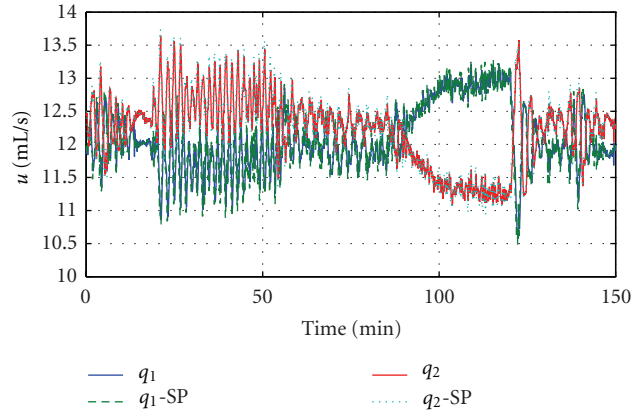


Figure 19: Experimental response: inputs.

easier, according to what the simulation curves with this solution indicate. If the system was linear, the relation given by pH $\times q_2/q_1$ plots would be a line whose approximation can be realized by the simulated curve of larger amount of buffer solution. Montandon [3] presented results for the closed-loop system using pH control of the acid flow with classical PID controller. The conclusion was that this implementation was not able to reject disturbances in a satisfactory manner. During the preparation of this work it has been studied certain PID testing for the MIMO case,

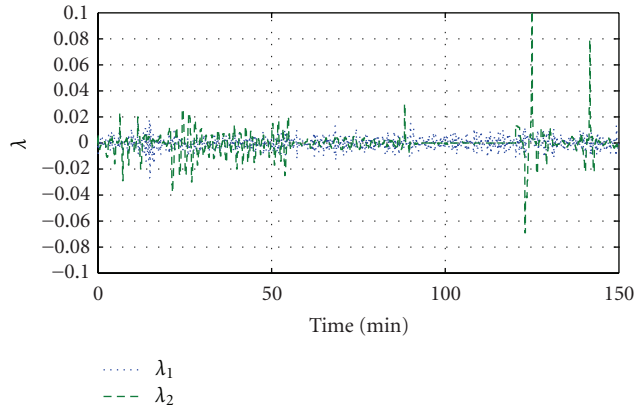


Figure 20: Experimental response: RSMPC—slack variables.

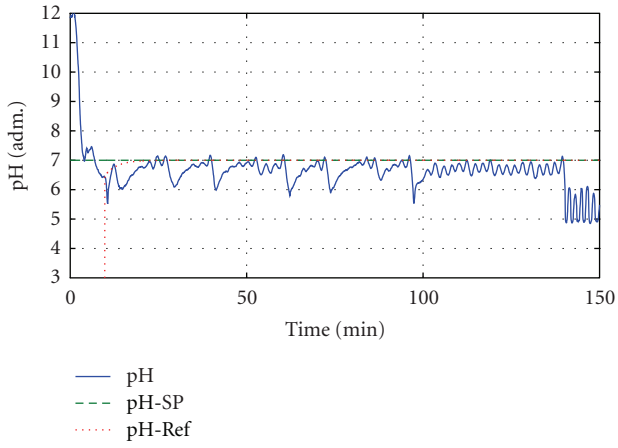


Figure 21: Experimental response RSMPC—pH (setpoint: SP and reference system: Ref).

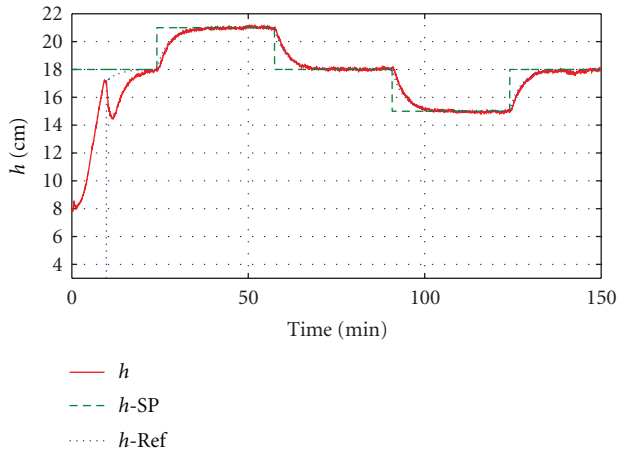


Figure 22: Experimental response: RSMPC—h (setpoint: SP and reference system: Ref).

but the answers proved to be mostly unstable and hard to tune.

3.1. Phenomenological Model. Hall [23] developed the physical model of this process, which is based on the hypotheses of the perfect mixture, constant density, and total solubility of the present ions. The chemical reactions involved in the acid-base neutralization (HNO_3–NaOH) are shown in (34)–(38). In order to have a complete model it will be considered the presence of a buffer ($NaHCO_3$). However, for the experimental results, this solution was not used:

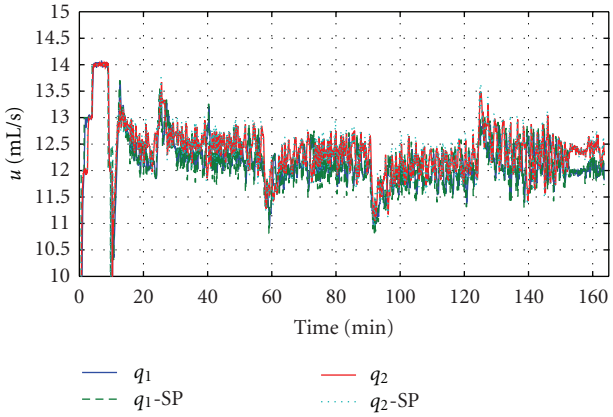


Figure 23: Experimental response: inputs.

$$HNO_3 \longrightarrow H^+ + NO_3^-, \tag{34}$$

$$NaOH \longrightarrow Na^+ + OH^-, \tag{35}$$

$$H_2O \overset{kw}{\longleftrightarrow} H^+ + OH^-, \tag{36}$$

$$NaHCO_3 + H_2O \longrightarrow NaOH + H_2CO_3, \tag{37}$$

$$H_2CO_3 \overset{ka1}{\longleftrightarrow} H^+ + HCO3^-,$$

$$HCO_3^- \overset{ka2}{\longleftrightarrow} H^+ + CO3^{-2}, \tag{38}$$

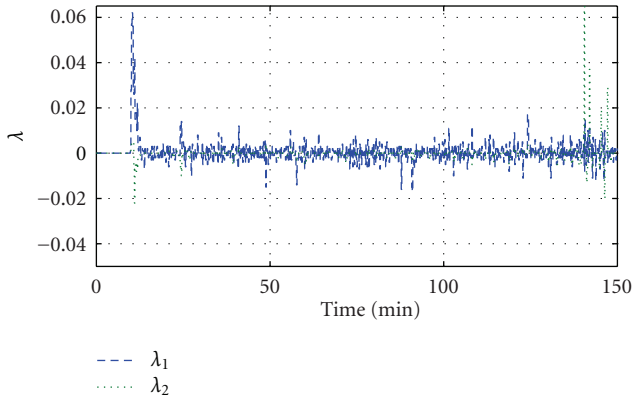


Figure 24: Experimental response: RSMPC—slack variables.

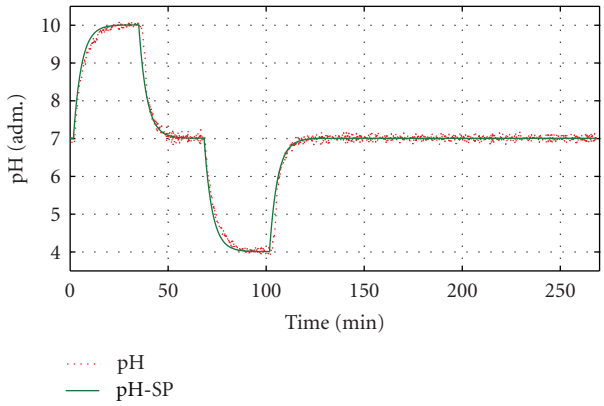


Figure 25: Closed-loop simulation: IHMPC-pH.

where

$$\text{ka1} = \frac{[\text{HCO}_3^-][\text{H}^+]}{[\text{H}_2\text{CO}_3]}, \tag{39}$$

$$\text{ka2} = \frac{[\text{CO}_3^-][\text{H}^+]}{[\text{HCO}_3^-]}. \tag{40}$$

The amounts of W_a and W_b are called as invariants, because their concentrations are not affected along the reaction. The reactions are fast enough to allow the system to be considered as in equilibrium. Then, the equilibrium reactions can be used to determine the concentration of the hydrogen (H) ions through the concentration of the reaction invariants. Equation (41) gives the equilibrium concentration. According to Gustafsson and Waller [24], there are two reaction invariants defined for the ith stream and presented in (42) and (43), which, when are combined,

Table 1: System nominal parameters.

Variables	Symbol	Nominal values
Level	H	25 cm
Area	Ar	168.38 cm^2
Volume	Vr	4209.67 cm^3
Acid flow	q_1	11.9130 mL/s
Base flow	q_2	11.8235 mL/s
Buffer solution flow	q_3	0.01 mL/s
pH	pH	7.0
Acid conc. in q_1	$[\text{HNO}_3]$	3.510×10^{-03} M
Base conc. in q_2	[NaOH]	3.510×10^{-03} M

result in an implicit relation of H, W_a, and W_b, as (44) shows:

$$kw = [\text{H}^+][\text{OH}^-], \tag{41}$$

$$W_{ai} = [\text{H}^+]_i - [\text{OH}^-]_i - [\text{HCO}_3^-]_i - 2\left[\text{CO}_3^{-2}\right]_i, \tag{42}$$

$$W_{bi} = [\text{H}_2\text{CO}_3]_i + [\text{HCO}_3^-]_i - 2\left[\text{CO}_3^{-2}\right]_i, \tag{43}$$

$$W_a = \text{H} - \frac{\text{kw}}{\text{H}} - W_b \frac{(\text{ka1/H}) + 2(\text{ka1ka2})/\text{H}^2}{1 + (\text{ka1/H}) + \left(\text{ka1ka2/H}^2\right)}. \tag{44}$$

Making the mass balance in the reactor, together with the invariant equations and considering that the density is constant by hypothesis, will result in a differential equation system (45)–(47):

$$\frac{dh}{dt} = \frac{1}{\text{Ar}}(q_1 + q_2 + q_3 - q_4), \tag{45}$$

$$\frac{dW_a}{dt} = \frac{1}{\text{Vr}}(q_1(W_{a1} - W_a) + q_2(W_{a2} - W_a) + q_3(W_{a3} - W_a)), \tag{46}$$

$$\frac{dW_b}{dt} = \frac{1}{\text{Vr}}(q_1(W_{b1} - W_b) + q_2(W_{b2} - W_b), + q_3(W_{b3} - W_b)), \tag{47}$$

where Ar is the reactor area, Vr is the reaction volume; h is the solution height in the reactor; W_a is the acid reaction invariant; W_b is the base reaction invariant.

The flow output of the system is driven by gravity. The output flow (q_4) is done by a globe-type valve (48). This equation shows the relation between output flow and the reactor height and the parameters to be estimated (c_v and $p7$). The valve parameters were estimated by using the model and experimental data from open loop. After this process of defining the parameters of the valve it was locked so as to avoid modifying these parameters. Nominal operation conditions are presented in Table 1:

$$q_4 = c_v h^{p7}. \tag{48}$$

For estimation and validation of the model, the plant operation must be done through the inputs that excite all the dynamic modes and in a large frequency band, so that the

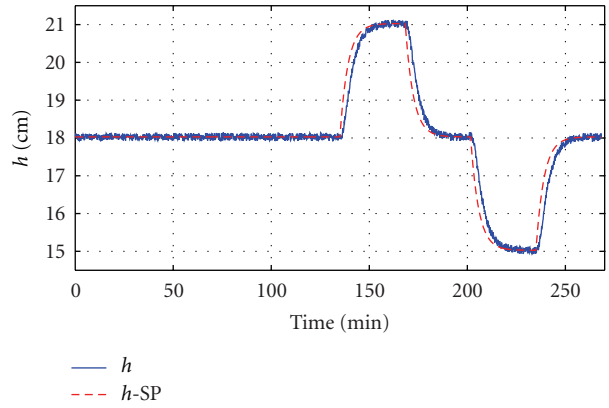


Figure 26: Closed-loop simulation: IHMPC-h.

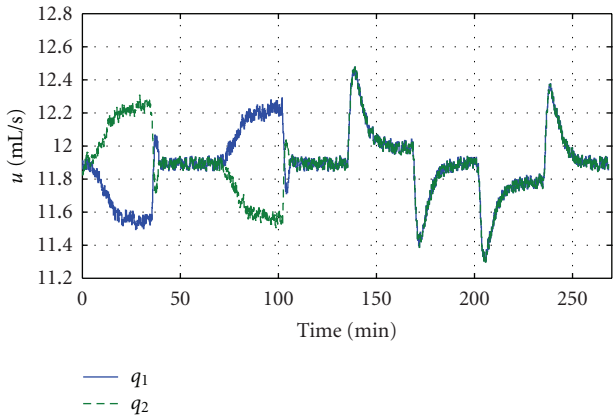


Figure 27: Closed-loop simulation: IHMPC-inputs.

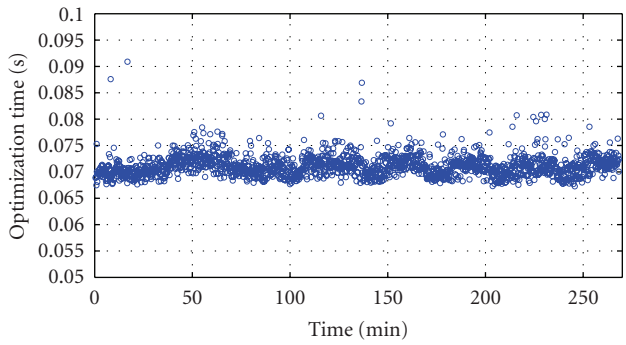


Figure 28: Closed-loop simulation: IHMPC-optimization time.

outputs have enough information for that procedure. These are the most usual excitation types: ramp, step, and impulse. However, in order to gather information in a broader range of frequency, random inputs made of sequences of steps with variable amplitude and duration are applied, as it is indicated by Montandon [3]. The sampling time for the process was 10 s. The acid and base flows vary as $q_1 \in [9.856 \;\; 13.310]$, $q_2 \in [9.977 \;\; 13.215]$, with a step probability equal to 0,8, in which the duration of each step was 120 s. The flows in which the variations occurred were 12 mL/s each, in order to keep the flow sum inside the limit given by $q_1 + q_2 \in [21.5 \;\; 24.5]$. The values presented were defined in such a way that, during the operation, the reactor did not get empty nor presents overflow. A set of experimental data composed by 6000

Table 2: Controller tuning.

Controller	Parameter
IHMPC	$N = 10$
	$\mathbf{R} = 10\mathbf{I} \rightarrow 100\mathbf{I}$
	$\mathbf{Q} = \mathbf{I}$
	$\mathbf{S} = 10\mathbf{I}$
	$\Delta\mathbf{u}_{\max} = 0.25$ mL/s
RSMPC	$\mathbf{K} = 0.005\mathbf{I}$
	$\mathbf{R} = \mathbf{I} \rightarrow \mathbf{R} = 100\mathbf{I}$
	$\mathbf{S} = 10000\mathbf{I}$
	$N = 2$
	$P = 10$
	$\Delta\mathbf{u}_{\max} = 0.5$ mL/s
General parameters	$\Delta t = 10$ s
	$\mathbf{u}_{\max} = 38$ mL/s
	$\mathbf{u}_{\min} = 4$ mL/s

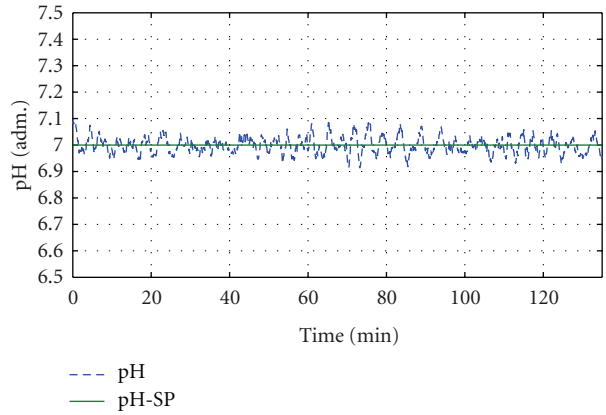


Figure 29: Experimental response: IHMPC-pH.

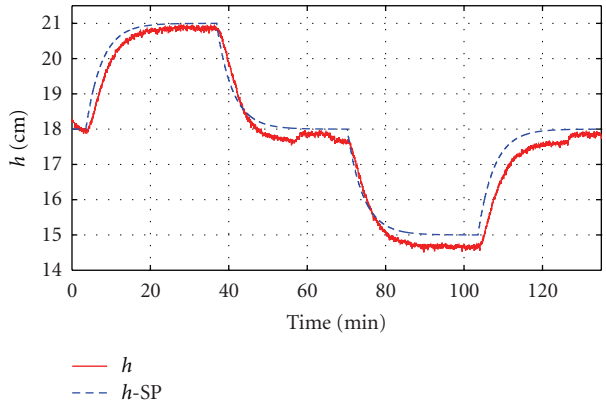


Figure 30: Experimental response: IHMPC-h.

acquisition points was used. The estimation procedure used half of the data and the validation employed the remaining data, in order to assure that the model would have the ability for predicting unknown data. The estimation is a nonlinear solution of the approximation curve that uses the least square method with a first trial given by a random value and a step equal to 15. The estimated values that resulted were $c_v = 20.4477$ and $p7 = 0.0523$. Figure 6 shows the error in predicting h. Figure 7 presents the set of input random data. Figure 8 presents the results of the simulation and the experimental ones for the closed-loop run, in which one can notice that the model is very representative of the process, with difficulties to predict around pH 7; this phenomenon was already expected, taking into account the titration curve of the system. Figure 9 shows the validation of the model by the prediction of the output that comes very close to the experimental data.

3.2. State-Space Model. The state-space approach resulting from the algebraic manipulation is given by (49)–(55). The terms $dWadz$ and $dWbdz$ are defined in (56) and (57). Equation (58) presents dydz relation:

$$\mathbf{x} = \begin{bmatrix} h & W_a & W_b \end{bmatrix}, \tag{49}$$

$$\mathbf{u} = \begin{bmatrix} q_1 & q_2 \end{bmatrix}, \tag{50}$$

$$\mathbf{y} = \begin{bmatrix} h & \text{pH} \end{bmatrix}, \tag{51}$$

$$\mathbf{d} = [q_3], \tag{52}$$

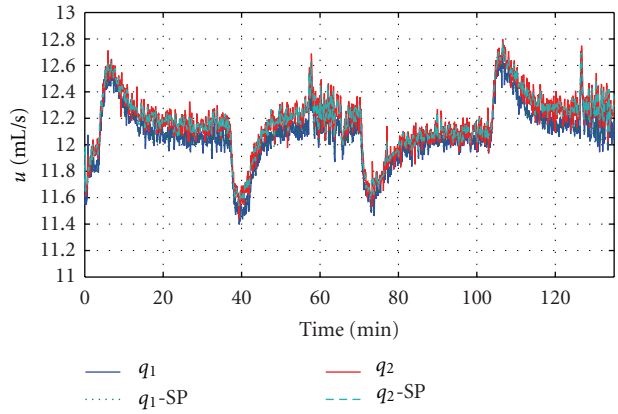


Figure 31: Experimental response: IHMPC-inputs.

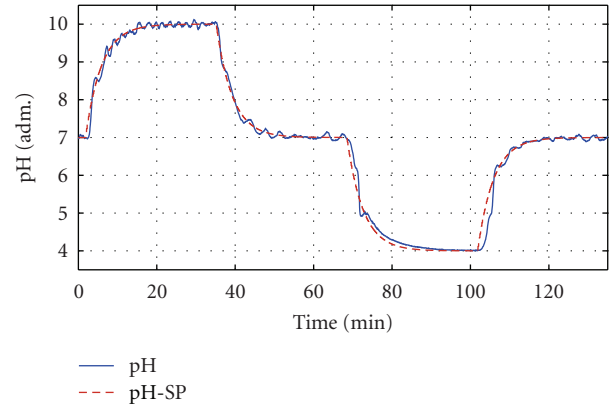


Figure 32: Experimental response: IHMPC-pH.

$$\mathbf{A} = \begin{bmatrix} -c_v p 7 h^{p7-1} & 0 & 0 \\ -\dfrac{q_1(W_{a1}-W_a)+q_2(W_{a2}-W_a)+q_3(W_{a3}-W_a)}{\mathrm{Arh}^2} & \dfrac{-q_1-q_2-q_3}{\mathrm{Arh}} & 0 \\ -\dfrac{q_1(W_{b1}-W_b)+q_2(W_{b2}-W_b)+q_3(W_{b3}-W_b)}{\mathrm{Arh}^2} & 0 & \dfrac{-q_1-q_2-q_3}{\mathrm{Arh}} \end{bmatrix}, \tag{53}$$

$$\mathbf{B} = \begin{bmatrix} \dfrac{1}{\mathrm{Ar}} & \dfrac{1}{\mathrm{Ar}} \\ \dfrac{(W_{a1}-W_a)}{\mathrm{Arh}} & \dfrac{(W_{a2}-W_a)}{\mathrm{Arh}} \\ \dfrac{(W_{b1}-W_b)}{\mathrm{Arh}} & \dfrac{(W_{b2}-W_b)}{\mathrm{Arh}} \end{bmatrix}, \qquad \mathbf{G} = \begin{bmatrix} \dfrac{1}{\mathrm{Ar}} \\ \dfrac{(W_{a3}-W_a)}{\mathrm{Arh}} \\ \dfrac{(W_{b3}-W_b)}{\mathrm{Arh}} \end{bmatrix}, \tag{54}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \dfrac{dydz}{dWadz} & \dfrac{dydz}{dWdz} \end{bmatrix}, \tag{55}$$

$$dWadz = 1 + \frac{\mathrm{kw}}{\mathrm{H}^2} - \frac{W_b\left(-\left(\mathrm{ka1}/\mathrm{H}^2\right)-\left(4\mathrm{ka1}\cdot\mathrm{ka2}/\mathrm{H}^3\right)\right)}{1+(\mathrm{ka1}/\mathrm{H})+\left(\mathrm{ka1}\cdot\mathrm{ka2}/\mathrm{H}^2\right)} + \frac{W_b\left((\mathrm{ka1}/\mathrm{H})+\left(2\cdot\mathrm{ka1.ka2}/\mathrm{H}^2\right)\right)\left(-\left(\mathrm{ka1}/\mathrm{H}^2\right)+\left(2\cdot\mathrm{ka1}\cdot\mathrm{ka2}/\mathrm{H}^3\right)\right)}{\left(1+(\mathrm{ka1}/\mathrm{H})+\left(\mathrm{ka1}\cdot\mathrm{ka2}/\mathrm{H}^2\right)\right)^2}, \tag{56}$$

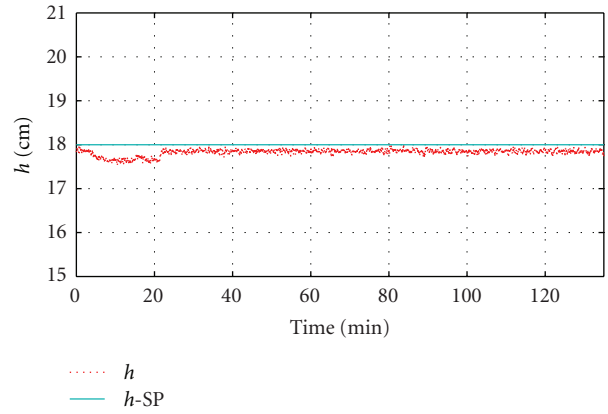


Figure 33: Experimental response: IHMPC-h.

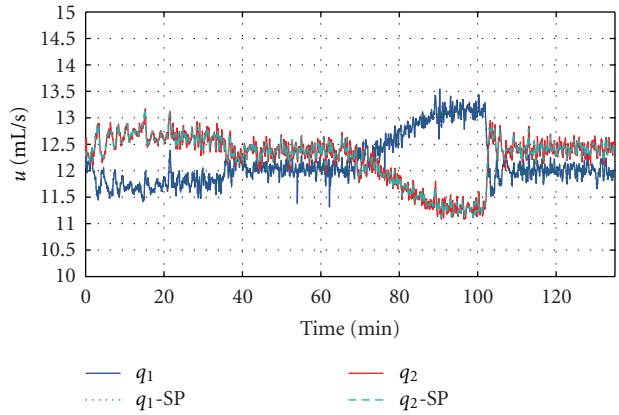


Figure 34: Experimental response: IHMPC-inputs.

$$dWdz = \frac{\left(1+\left(\mathrm{kw/H^2}\right)\right)\left(1+(\mathrm{ka1/H})+\left(\mathrm{ka1}\cdot\mathrm{ka2/H^2}\right)\right)}{(\mathrm{ka1/H})+\left(2\cdot\mathrm{ka1}\cdot\mathrm{ka2/H^2}\right)} - \frac{(\mathrm{H}-(\mathrm{kw/H})-W_a)\left(1+(\mathrm{ka1/H})+\left(\mathrm{ka1}\cdot\mathrm{ka2/H^2}\right)\right)\left(-\left(\mathrm{ka1/H^2}\right)-\left(4\cdot\mathrm{ka1}\cdot\mathrm{ka2/H^3}\right)\right)}{\left((\mathrm{ka1/H})+\left(2\cdot\mathrm{ka1}\cdot\mathrm{ka2/H^2}\right)\right)^2} + \frac{(\mathrm{H}-(\mathrm{kw/H})-W_a)\left(-\left(\mathrm{ka1/H^2}\right)-\left(2\cdot\mathrm{ka1}\cdot\mathrm{ka2/H^3}\right)\right)}{(\mathrm{ka1/H})+\left(2\cdot\mathrm{ka1}\cdot\mathrm{ka2/H^2}\right)}, \tag{57}$$

$$dydz = -\log_{10}\frac{e}{\mathrm{H}}. \tag{58}$$

4. Results and Discussions

The real-time implementation was done using the successive linearization around the operating point. In this case, the linear form results (11).

A controller designed around a stationary state would not be able to control the system, because, for some regions, the stationary state would be very far from the desired point and so the plant model mismatch. Even with the successive linearization, the system control would not be successful, mainly in regions with high gain. There are several ways to deal with this problem. One of them is to expand the states, creating a new state matrix, and then the original structure of the optimization problem can be kept (59). The term related to the output linearization does originate problems for the controller, because it only maps state output and does not interfere in the systems dynamics as the term $\mathbf{f}_{k-1}$ does. Once this modification is done, it is possible to work on the controller previously defined. This option was chosen for real-time operation. Another form can be found in Reis [25]:

$$\frac{d\mathbf{x}}{dt} = \begin{bmatrix} \mathbf{A}_{k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{f}_{k-1} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} + \mathbf{B}_{k-1}\mathbf{u}. \tag{59}$$

For the real application in the experimental system, which has nonlinear characteristics, it is necessary to perform successive linearization of the process model in order to apply linear algorithms. The dynamic evolution of the process results in a time-varying system (LTV). The equation of the IHMPC and RSMPC in an LTV form can be easily extrapolated from the formulation presented herein, and for simplicity its description is omitted.

In the experimental system, there are the states h, W_a, and W_b. For this case, the last two states are not measured, as it was shown in the modeling section of the system, so you need to estimate them ($\hat{\mathbf{x}}$). To this end, we used a Kalman filter [26], (60). For the IHMPC controller, we also used an open-loop observer (61), informing the model that the error ($\mathbf{e}$) enters into the plant and model in order to minimize the interference of state estimation in the system behavior [27]:

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{A}\hat{\mathbf{x}}_{k|k-1} + \mathbf{B}\mathbf{u}_k + \mathbf{L}\left(\mathbf{y}_k - \mathbf{C}\hat{\mathbf{x}}_{k|k-1}\right), \tag{60}$$

$$\mathbf{e} = \mathbf{y}_{\text{Plant}} - \mathbf{y}_{\text{Model}}. \tag{61}$$

RSMPC and IHMPC were implemented in real-time operation of the experimental plant, using the interface through LabVIEW and the controller computation, using the routines implemented in Matlab. In the literature there are several techniques of controller tuning. In this work Henson and Seborg [11] and Montadon's [3] indications were adopted. Besides, a field refinement was done during the initial run in closed loop with each controller, making small adjustments in parameters, so that the system would deliver a better experimental performance in closed loop. Such results were omitted in this paper. Table 2 shows the parameters of each controller resulting from the simulation and used in the experimental run. The controllers were at first tuned with identity matrices and a control horizon equal to 10; besides, a tuning was done until achieving a satisfactory response. Figure 10 presents the structure of the experimental plant, and Figures 11 and 12 show the LabVIEW implementation.

Equation (62) shows that the calibration curves of the instruments. V_0, V_1, V_2, V_3, and V_4 are channel voltages. The regression coefficients are equal to 1 [4, 28]:

$$\begin{aligned} q_1 &= 5.2788V_0 - 9.4776, \\ q_2 &= 5.0997V_1 - 9.5541, \\ q_3 &= 0.2090V_2 - 0.3845, \\ \text{pH} &= 1.8982V_3 - 3.5583, \\ h &= 5.6347V_4 - 9.0838. \end{aligned} \tag{62}$$

Setpoint variations were done to evaluate the ability of the controllers to lead with transitions. Starting the system with a pH equal to 7 and a reactor height equal to 18 cm, variations were done according to the vector in (63), considering that each step took 2000 seconds. For all controller simulations a random noise of average equal to zero was added with about 10% of the outputs (h and pH):

$$\begin{bmatrix} h \\ \text{pH} \\ d \end{bmatrix} = \begin{bmatrix} 18 & 18 & 18 & 18 & 18 & 21 & 18 & 15 & 18 \\ 7 & 10 & 7 & 4 & 7 & 7 & 7 & 7 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{63}$$

Figures 13, 14, 15, and 16 present the closed-loop simulation that uses RSMPC. The performance shown is quite reasonable.

The experimental run was done in two steps to preserve and manipulate the data separately. In the first part was made a variation in the pH setpoint and kept the level constant. In the second part it was done otherwise. Figures 17, 18, 19, and 20 show the experimental run of the pH setpoint changes, and Figures 21, 22, 23, and 24 show the experimental run of level setpoint changes. The system was controlled in all runs. During the experiment, there was a need to increase the weight matrix of the control action (as indicated in Table 2); as the output response started to oscillate, it was not able to reach the desired setpoint. However, the overall response can be considered satisfactory.

The results for the IHMPC simulation are presented in Figures 25, 26, 27, and 28. The results are suitable to apply on line the control to the real process.

Like RSMPC, experimental runs with IHMPC were done in two steps. Figures 29, 30, and 31 show the experimental run of the pH setpoint changes, and Figures 32, 33, and 34 show the experimental run of level setpoint changes. This controller showed a performance superior to the RSMPC controller performance and with satisfactory and adequate response and performance.

5. Conclusions

Through this research it was able to better understand the use of predictive controllers in a real-time application, paving the way for research in the area. The experimental application led to an approximation of reality and industrial practice, experiencing some of the common problems and issues in the implementation of controllers. It was possible to verify the need for a mastery of techniques and concepts of process control, system modeling, parameter identification, scheduling, optimization, and, in addition, common sense engineering to solve the experimental problems.

The control of such class of nonlinear processes is a very challenging area with many possibilities for development and that undoubtedly has importance and influence on the performance of process and in consequence results in their organizations.

This work carried out the experimental application of two predictive controllers: IHMPC and RSMPC, both in simulation environment and in the experimental plant dealing with the level and pH control.

The experimental plant was modeled and has the required parameters identified through the run in open loop. The simulation results were satisfactory and indicated that the model is representative of the real process and suitable for the control purposes that were aimed. The simulation

answers of the system outputs subjected to the controllers were adequate and satisfactory. The theoretical and experimental responses of the IHMPC runs were satisfactory.

The controller adjustment for real-time operations was suitable and feasible. The interference of matters related to leaking, process noises, and noises from electrical source and other problems associated to real-time application brought additional difficulties that demanded process knowledge of the process.

The experimental application of robust controllers associated or not to control systems tolerant to failures and the use of online identification employing neural networks will be presented elsewhere.

References

[1] G. Stephanopoulos, *Chemical Process Control: An Introduction to Theory and Practice*, Prentice Hall, Upper Saddle River, NJ, USA, 1984.

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

[3] A. G. Montandon, *Controle preditivo em tempo real com trajetória de referência baseado em modelo neural para reator de neutralização*, M.S. thesis, Federal University of Uberlândia, 2005.

[4] C. H. F. Silva, *Uma contribuição ao estudo de controladores robustos*, Ph.D. thesis, Universidade Federal de Uberlândia , 2009.

[5] E. F. Camacho and C. Bordons, *Model Predictive Control*, Springer, Barcelona, Spain, 1999.

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

[7] M. Nikolaou, "Model predictive controllers: a critical synthesis of theory and industrial needs," *Advances in Chemical Engineering*, vol. 26, pp. 131–204, 2001.

[8] M. Morari, "Model predictive control: multivariable control technique of choice in the 1990?" Tech. Rep. CIT/CDS 93-024, California Institute Of Technology, 1993.

[9] D. R. Saffer II and F. J. Doyle III, "Analysis of linear programming in model predictive control," *Computers and Chemical Engineering*, vol. 28, no. 12, pp. 2749–2763, 2004.

[10] J. B. Rawlings, "Tutorial: model predictive control technology," in *Proceedings of the American Control Conference (ACC'99)*, pp. 662–676, June 1999.

[11] M. A. Henson and D. E. Seborg, *Nonlinear Process Control*, Prentice Hall, Upper Saddle River, NJ, USA, 1997.

[12] S. de Oliveira and M. Morari, "Robust model predictive control for nonlinear systems with constraints," in *Advanced Control Of Chemical Processes*, pp. 295–300, Kyoto, Japan, 1994.

[13] J. H. Lee, "Recent advances in model predictive control and other related areas," *AIChE Symposium Series*, pp. 201–216, 1997.

[14] J. W. Eaton and J. B. Rawlings, "Model-predictive control of chemical processes," *Chemical Engineering Science*, vol. 47, no. 4, pp. 705–720, 1992.

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

[16] J. Löfberg, *Minimax approaches to robust model predictive control*, Ph.D. thesis, Linköping, Sweden, 2003.

[17] N. L. Ricker, "Model predictive control with state estimation," *Industrial and Engineering Chemistry Research*, vol. 29, no. 3, pp. 374–382, 1990.

[18] J. Richalet, "Industrial applications of model based predictive control," *Automatica*, vol. 29, no. 5, pp. 1251–1274, 1993.

[19] A. Richards and J. How, "Robust model predictive control with imperfect information," in *Proceedings of the American Control Conference, (ACC'05)*, pp. 268–273, June 2005.

[20] Q. L. Zhang and Y. G. Xi, "An efficient model predictive controller with pole placement," *Information Sciences*, vol. 178, no. 3, pp. 920–930, 2008.

[21] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Computers and Chemical Engineering*, vol. 23, no. 4-5, pp. 667–682, 1999.

[22] H. M. Henrique, *Uma contribuição ao estudo de redes neuronais aplicadas ao controle de processos*, Ph.D. thesis, COPPE/UFRJ, 1999.

[23] R. C. Hall, *Development of a multivariable pH experiment*, M.S. thesis, University of California, Santa Barbara, Calif, USA, 1987.

[24] T. K. Gustafsson and K. V. Waller, "Dynamic modeling and reaction invariant control of pH," *Chemical Engineering Science*, vol. 38, no. 3, pp. 389–398, 1983.

[25] L. L. G. Reis, *Controle tolerante com reconfiguração estrutural acoplado a sistema de diagnóstico de falhas*, M.S. thesis, Federal University of Uberlândia, 2008.

[26] T. Kailath, *Linear Systems*, Prentice Hall, Englewood Cliffs, NJ, USA, 1980.

[27] J. M. Maciejowski, *Predictive Control With Constraints*, Pearson Education Limited, 2002.

[28] D. Gonçalves, *Controle preditivo em tempo real de uma planta piloto*, M. Sc qualification report, Federal University of Uberlândia, 2007.

Hindawi Publishing Corporation
Journal of Control Science and Engineering
Volume 2012, Article ID 485784, 7 pages
doi:10.1155/2012/485784

Research Article

Robust Model Predictive Control Using Linear Matrix Inequalities for the Treatment of Asymmetric Output Constraints

Mariana Santos Matos Cavalca,[1] Roberto Kawakami Harrop Galvão,[2] and Takashi Yoneyama[2]

[1] *Departamento de Engenharia Elétrica, Centro de Ciências Tecnológicas, Universidade do Estado de Santa Catarina, Praça Rua Paulo Malschitzki, Zona Industrial Norte, 89.219-710 Joinville, SC, Brazil*
[2] *Divisão de Engenharia Eletrônica, Instituto Tecnológico de Aeronáutica, Praça Marechal Eduardo Gomes, 50 Vila das Acácias, 12.228-900 São José dos Campos, SP, Brazil*

Correspondence should be addressed to Mariana Santos Matos Cavalca, mariana.smcavalca@gmail.com

Received 30 June 2011; Revised 22 September 2011; Accepted 6 October 2011

Academic Editor: Marcin T. Cychowski

One of the main advantages of predictive control approaches is the capability of dealing explicitly with constraints on the manipulated and output variables. However, if the predictive control formulation does not consider model uncertainties, then the constraint satisfaction may be compromised. A solution for this inconvenience is to use robust model predictive control (RMPC) strategies based on linear matrix inequalities (LMIs). However, LMI-based RMPC formulations typically consider only symmetric constraints. This paper proposes a method based on pseudoreferences to treat asymmetric output constraints in integrating SISO systems. Such technique guarantees robust constraint satisfaction and convergence of the state to the desired equilibrium point. A case study using numerical simulation indicates that satisfactory results can be achieved.

1. Introduction

Model-based predictive control (MPC) is a strategy in which a sequence of control actions is obtained by minimizing a cost function considering the predictions of a process model within a certain prediction horizon. At each sample time, only the first value of this sequence is applied to the plant, and the optimization is repeated in order to use feedback information [1, 2]. One of the main advantages of MPC is the possibility to consider explicitly the physical and operational constraints of the system during the design of the control loop [1–3]. However, if there are mismatches between the nominal model and the actual behavior of the process, then the performance of the control loop can be degraded and the optimization problem may even become unfeasible. Thus, the study of new strategies for design of robust MPC (RMPC) with guaranteed stability and constraint satisfaction properties even in the presence of uncertainties is an area with great potential for research [4–7].

In this context, Kothare et al. [7] proposed an RMPC strategy with infinite horizon employing linear matrix inequalities (LMIs) for dealing with model uncertainty and symmetric constraints on the manipulated and output variables. This approach was later extended to encompass multimodel representations [8, 9], setpoint management [10], integrator resetting [11], and offline solutions [12, 13].

Within this scope, Cavalca et al. [4] proposed a heuristic procedure that allows the inclusion of asymmetric constraints on the plant output, but without stability or constraint satisfaction guarantees. The present paper presents a formal strategy to handle asymmetric output constraints in the control of integrating single input, single output (SISO) systems, for the case where the output is linear in states. It is shown that robust constraint satisfaction is achieved, as well as convergence of the state trajectory to the desired equilibrium point. The effectiveness of the proposed method is illustrated by means of numerical simulations.

The remainder of this paper is organized as follows. Section 2 reviews the design of RMPC based on LMI. Section 3 formalizes the proposed technique for the inclusion of asymmetric constraints. Section 4 presents a case study consisting of a discrete time model of a double integrator. The results are evaluated through numerical simulations as presented in Section 5. Concluding remarks are presented in Section 6.

Throughout the text, I represents an identity matrix, the notation $(\cdot \mid k)$ is used in predictions with respect to time k, and the $*$ superscript indicates an optimal solution. For brevity of notation, only the upper triangular part of symmetric matrices is explicitly presented.

2. LMI-Based RMPC

Consider a linear time-invariant system described by an uncertain model of the following form:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ y(k) &= Cx(k), \end{aligned} \tag{1}$$

where $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^p$, and $y(k) \in \mathbb{R}^q$ are the states, the manipulated and the output variables, respectively, for each time k, and A, B, and C are constant matrices of appropriate dimensions. The uncertainty is represented in polytopic form; that is, matrices A and B are unknown to the designer, but they are assumed to belong to a convex polytope Ω with L known vertices (A_i, B_i), $i = 1, 2, \ldots, L$, so that [2, 7]:

$$(A, B) = \sum_{i=1}^{L} \lambda_i (A_i, B_i) \tag{2}$$

for some unknown set of coefficients $\lambda_1, \lambda_2, \ldots, \lambda_L$ that satisfy

$$\sum_{i=1}^{L} \lambda_i = 1 \qquad \lambda_i \geq 0, \ i = 1, \ldots, L. \tag{3}$$

At each time k, the sequence of future controls $U_\infty = \{u(k \mid k), u(k+1 \mid k), u(k+2 \mid k), \ldots\}$ is obtained as solution of the following min-max optimization problem:

$$\min_{U_\infty} \max_{(A,B)\in\Omega} J_\infty(A, B, U_\infty), \tag{4}$$

where

$$J_\infty(A, B, U_\infty) = \sum_{j=0}^{\infty} \left[\| x(k+j \mid k) \|^2_{W_x} + \| u(k+j \mid k) \|^2_{W_u} \right] \tag{5}$$

in which $W_x > 0$ and $W_u > 0$ are symmetric weighting matrices. By assumption, all states are available for feedback so that $x(k \mid k) = x(k)$.

This min-max problem can be replaced with the following convex optimization problem with variables $\gamma \in \mathbb{R}$, $Q \in \mathbb{R}^{n\times n}$, $\Sigma \in \mathbb{R}^{p\times n}$, and LMI constraints [2, 7]:

$$\min_{\gamma, Q, \Sigma} \quad \gamma \tag{6}$$

$$\text{subject to} \quad \begin{bmatrix} Q & x(k) \\ \cdot & 1 \end{bmatrix} > 0, \tag{7}$$

$$\begin{bmatrix} Q & 0 & 0 & A_i Q + B_i \Sigma \\ \cdot & \gamma I & 0 & W_x^{1/2} Q \\ \cdot & \cdot & \gamma I & W_u^{1/2} \Sigma \\ \cdot & \cdot & \cdot & Q \end{bmatrix} > 0, \qquad i = 1, 2, \ldots, L. \tag{8}$$

If the problem (6)–(8) has a solution γ_k^*, Q_k^*, Σ_k^*, then the optimal control sequence is given by

$$u(k+j \mid k) = K_k^* x(k+j \mid k), \tag{9}$$

where

$$K_k^* = \Sigma_k^* \left(Q_k^* \right)^{-1}. \tag{10}$$

Symmetric constraints on the manipulated variables of the form $|u_l(k+j \mid k)| < \overline{u}_l$, $l = 1, 2, \ldots, p$, $j \geq 0$ and on the output variables of the form $|y_m(k+j+1 \mid k)| < \overline{y}_m$, $m = 1, 2, \ldots, q$, $j \geq 0$ can be imposed by including additional LMIs [7]

$$\begin{bmatrix} X & \Sigma \\ \cdot & Q \end{bmatrix} > 0 \tag{11}$$

with

$$\begin{aligned} X_{ll} &< \overline{u_l}^2, \qquad l = 1, 2, \ldots, p, \\ \begin{bmatrix} Q & [A_i Q + B_i \Sigma]^T C_m^T \\ \cdot & \overline{y}_m^2 \end{bmatrix} &> 0, \qquad i = 1, 2, \ldots, L \end{aligned} \tag{12}$$

for $m = 1, 2, \ldots, q$, where C_m denotes the mth row of C.

Let $\mathbb{P}(x(k))$ denote the optimization problem (6) with constraints (7), (8), (11), and (12). Suppose that the control law uses the concept of receding horizon; that is, $u(k) = K_k^* x(k)$ with K_k^* recalculated at each sampling time k. The following lemma is concerned with convergence of the state trajectory to the origin under the constraints imposed on the operation of the plant.

Lemma 1. *If $\mathbb{P}(x(k_0))$ is feasible at some time $k = k_0$, then $\|x(k)\| \to 0$ as $k \to \infty$ with satisfaction of the input and output constraints.*

The proof of Lemma 1 follows directly from the recursive feasibility and asymptotic stability properties demonstrated in [7].

Remark 2. As shown in Appendix A of Kothare et al. [7], $\mathbb{P}(x(k))$ is equivalent to minimizing a function $V(x(k \mid k)) = x(k \mid k)^T P_k x(k \mid k)$ with $P_k = \gamma_k^* (Q_k^*)^{-1} > 0$. Such $V(x(k \mid k))$ function is found to be an upper bound for the cost function $J_\infty(A, B, U_\infty)$ in (5) and can be used as a candidate Lyapunov function in the proof of asymptotic stability.

Remark 3. For a regulation problem around a point different from the origin, a change of variables can be used, so that the new origin corresponds to the desired equilibrium point [7]. In the present work, the new value for the reference signal will be termed a pseudoreference. It is assumed that the process is integrating, and, therefore, the control value in steady state (u_{ss}) is zero for any value of the pseudoreference. Otherwise, the determination of u_{ss} would not be trivial since the system matrices A and B are subjected to model uncertainties.

Remark 4. The RMPC problem formulation presented in this section considers that the matrices A and B are unknown but dot not vary with time. This is a particular case of the general framework introduced in [7], which was concerned with time-varying matrices $A(k)$ and $B(k)$.

3. Treatment of Asymmetric Constraints

The present work is concerned with regulation problems around the origin involving a SISO system with output variable $y(k) = Cx(k)$. This is a particular case of the problem described in Section 2, with $p = q = 1$. Therefore, the indexes l and m in (12) can be omitted. Moreover, the system is assumed to be integrating, so that $u_{ss} = 0$ regardless of the pseudoreference, as discussed in Remark 3. It is also considered that the manipulated variable $u(k)$ is subjected to a symmetric constraint $\overline{u}$ as in Section 2. Suppose that the constraints on y are of the following form:

$$y_{\min} < y(k) < y_{\max} \tag{13}$$

with $y_{\min} < 0$ and $y_{\max} > 0$. If $|y_{\min}| = y_{\max}$, then the symmetric constraint formulation presented in Section 2 can be applied directly by making $\overline{y} = y_{\max}$. If $|y_{\min}| \neq y_{\max}$, a different approach is required. One alternative is to adopt a more conservative constraint, that is,

$$\overline{y} = \begin{cases} y_{\max}, & \text{if } y_{\max} < |y_{\min}| \\ -y_{\min}, & \text{if } y_{\max} > |y_{\min}|. \end{cases} \tag{14}$$

However, this procedure may not be convenient in the following cases:

(i) $y_{\max} < |y_{\min}|$ with $y_{\min} \leq y(k) < -y_{\max}$,

(ii) $y_{\max} > |y_{\min}|$ with $-y_{\min} < y(k) \leq y_{\max}$.

In these cases, the initial value of the output is admissible under the original asymmetric constraints, but not under the more conservative constraint (14). Figure 1(a) provides an illustration for case (i). As can be seen, the imposition of the more conservative constraint $-y_{\max} < y(k) < y_{\max}$ makes the output variable $y(k)$ be located outside of the range of admissible values.

Cavalca et al. [4] proposed a heuristic solution, based on a time-varying pseudoreference $r(k) = \min\{(y_{\max} + y(k))/2,\ 0\}$ as shown in Figure 1(b). The problem of asymmetric output constraints is then redefined in terms of new symmetric constraints (a/a) around $r(k)$. However, it should be noted that this technique does not lead to guaranteed stability and constraint satisfaction.

Unlike the approach described above, which involves a pseudoreference $r(k)$ that may change at each sampling time k, the solution proposed in the present paper employs a sequence of pseudoreferences r_i ($i = 1, 2, \ldots, i_{\max}$) which are defined at $k = 0$ on the basis of the initial output value $y(0)$. As illustrated in Figure 2, symmetric constraints $(a/a, b/b, c/c, \ldots)$ are established around each pseudoreference. It will be shown that the use of such pseudoreferences, together with a convenient commutation rule, provides robust constraint satisfaction and ensures that the state trajectory converges to the origin. For sake of brevity of presentation, only case (i) will be treated. Case (ii) can be recast into case (i) by defining $\check{y}(k) = -Cx(k)$ and replacing constraints $y_{\min} \leq y(k) \leq y_{\max}$ with $-y_{\max} \leq \check{y}(k) \leq -y_{\min}$.

Given an initial state $x(0)$ such that $y(0) = Cx(0)$ falls within the scope of case (i), the following algorithm defines the pseudoreferences r_i, as well as a sequence of associated matrices $\tilde{Q}_i^*$. These matrices will be subsequently employed in the control law to establish a rule of commutation from one pseudoreference to the next.

It is assumed that the set X_s of possible equilibrium values x_s for the state of the plant is known from the physics of the process to be controlled.

Algorithm for Determination of the Pseudoreferences and Associated Ellipsoids (PR Algorithm).

Step 1. Define the pseudoreferences r_i as follows:

Step 1.1. Let $r_0 = [y_{\max} + y(0)]/2$

Step 1.2. Let $i = 1$

Step 1.3. While $|r_{i-1}| > y_{\max}$ do

$$r_i = (y_{\max} + r_{i-1})/2$$

$$i = i + 1$$

End While

Step 1.4. Let $i_{\max} = i$

Step 1.5. Let $r_{i_{\max}} = 0$.

Step 2. For each r_i determine $x_{s,i}$ such that:

$$Cx_{s,i} = r_i, \tag{15}$$

$$x_{s,i} \in X_s. \tag{16}$$

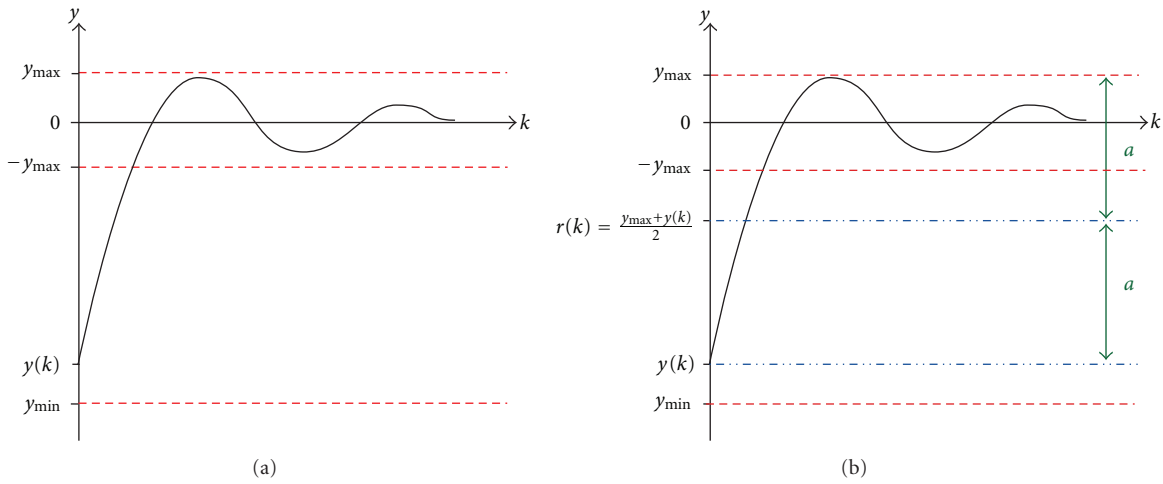


Figure 1: (a) Illustration of the more conservative constraint (14) in case (i). (b) Heuristic solution proposed by Cavalca et al. [4].

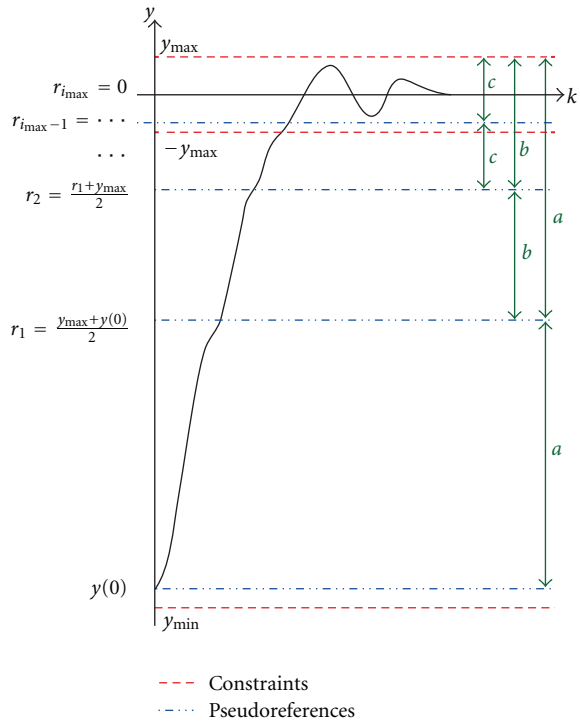


Figure 2: Illustration of the proposed pseudoreference scheme.

Step 3. Let

$$\xi_0 = x(0) - x_{s,0}$$

$$\xi_i = x_{s,i-1} - x_{s,i}, \quad 1 \le i \le i_{max}$$

$$\overline{y}_i = y_{max} - r_i, \quad 0 \le i \le i_{max}.$$

Step 4. Solve the problem $\mathbb{P}(\xi_i)$ with the constraints $\overline{y}_i$, $\overline{u}$ for each i ($i = 0, 1, \ldots, i_{max}$) and denote the resulting solution by $(\widetilde{y}_i^*, \widetilde{Q}_i^*, \widetilde{\Sigma}_i^*)$.

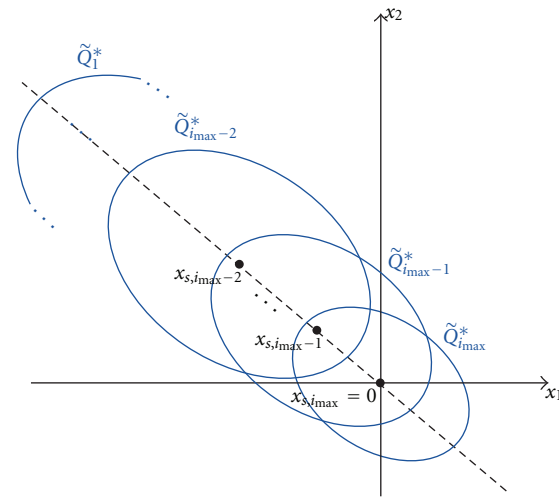


Figure 3: Result of the PR algorithm.

The matrices $\widetilde{Q}_i^*$ obtained in Step 4 define ellipsoids of the form $(x - x_{s,i})^T(\widetilde{Q}_i^*)^{-1}(x - x_{s,i}) < 1$, as illustrated in Figure 3 for the case of a second-order system. It is noteworthy that, by construction (in view of LMI (7) with $x(k) = \xi_i$), the ellipsoid i contains the center of the ellipsoid $(i - 1)$.

The PR algorithm is said to be feasible if (15) and (16) in Step 2 and the optimization problem $\mathbb{P}(\xi_i)$ in Step 4 are feasible for every $i = 1, 2, \ldots, i_{max}$. In this case, the resulting $x_{s,i}$, $\overline{y}_i$, $\widetilde{Q}_i^*$, $i = 0, 1, \ldots, i_{max}$ are used in the control algorithm proposed below.

Algorithm for Control Using the Pseudoreferences (CPR Algorithm).

Initialization. Let $k = 0$ and $i = 0$.

Step 1. Read the state $x(k)$.

Step 2. If $i < i_{\max}$ then

Let $\tilde{x}_{i+1}(k) = x(k) - x_{i+1,s}$

If $\tilde{x}_{i+1}^T(k)(\tilde{Q}_{i+1}^*)^{-1}\tilde{x}_{i+1}(k) < 1$ (condition of transition) then

Let $i = i + 1$

End If

End If.

Step 3. Let $\tilde{x}_i(k) = x(k) - x_{s,i}$ and solve the problem $\mathbb{P}(\tilde{x}_i(k))$ with the constraints $\overline{y_i}$ and $\overline{u}$ in order to obtain $(\gamma_k^*, Q_k^*, \Sigma_k^*)$.

Step 4. Calculate $K_k^* = \Sigma_k^*(Q_k^*)^{-1}$ and $u(k) = K_k^*\tilde{x}_i(k)$.

Step 5. Apply $u(k)$ to the plant.

Step 6. Let $k = k+1$, wait a sample time and return to Step 1.

The main result of this work is stated in the following theorem, which is concerned with the satisfaction of constraints and convergence of the state trajectory to the origin under the control law given by the CPR algorithm.

Theorem 5. *If the PR algorithm is feasible and the CPR algorithm is applied to control the plant, then* $\|x(k)\| \to 0$ *as* $k \to \infty$, *with satisfaction of the input and output constraints.*

Proof. For $k = 0$, the state $x(k)$ lies in the ellipsoid associated with $\tilde{Q}_0^*$, which was obtained by solving problem $\mathbb{P}(x(0) - x_{s,0})$ in the PR algorithm. Therefore, problem $\mathbb{P}(\tilde{x}_0(0))$ is feasible by hypothesis. Lemma 1 then guarantees that $\|\tilde{x}_0(k)\| \to 0$ (i.e., $\|x(k) - x_{s,0}\| \to 0$) as $k \to \infty$, with satisfaction of the input and output constraints, under the control law stated in Steps 3 and 4 of the CPR algorithm with $i = 0$. Since the ellipsoid associated with $\tilde{Q}_1^*$ contains $x_{s,0}$, by construction, it can be concluded that the condition of transition stated in Step 2 of the CPR algorithm with $i = 1$ will be satisfied in finite time. Let k_1 be the first time when this condition is satisfied, that is,

$$\tilde{x}_1^T(k_1)\left(\tilde{Q}_1^*\right)^{-1}\tilde{x}_1(k_1) < 1. \tag{17}$$

This condition ensures that the optimization problem $\mathbb{P}(\tilde{x}_1(k_1))$ is feasible, since, by construction, the solution $(\tilde{\gamma}_1^*, \tilde{Q}_1^*, \tilde{\Sigma}_1^*)$ of $\mathbb{P}(\xi_1)$ satisfies the constraints of $\mathbb{P}(\tilde{x}_1(k_1))$. In fact, condition (17) ensures that LMI (7) is satisfied with $x(k)$ and Q replaced with $\tilde{x}_1(k_1)$ and $\tilde{Q}_1^*$, respectively. Moreover, the remaining LMIs (8),(11)-(12) are satisfied by $\tilde{Q}_1^*$ and $\tilde{\Sigma}_1^*$ by hypothesis. Therefore, after switching from $i = 0$ to $i = 1$, Lemma 1 ensures that $\|\tilde{x}_1(k)\| \to 0$ (i.e., $\|x(k) - x_{s,1}\| \to 0$) as $k \to \infty$ with satisfaction of the input and output constraints. As a result, the condition of transition with $i = 1$ will be satisfied in finite time. A similar reasoning can be applied to show that the condition of transition will be satisfied for all $i = 0, 1, \ldots, i_{\max} - 1$ in finite time. Finally, when $i = i_{\max}$, the state $x(k)$ will be inside the last ellipsoid, centered at $x_{s,i_{\max}} = [0\ 0\ \cdots\ 0]^T$, and then Lemma 1 will ensure that $\|x(k)\| \to 0$ as $k \to \infty$, again with satisfaction of the constraints. □

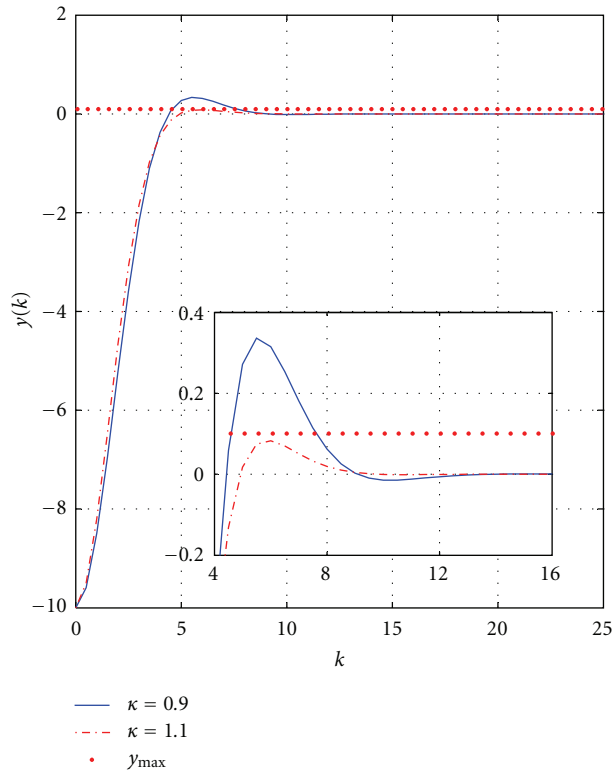


FIGURE 4: Simulation results using relaxed output constraints.

4. Numerical Example

A discrete state-space model of a double integrator will be employed to illustrate the proposed strategy. The matrices of the model are given by

$$A = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}\kappa, \qquad C = \begin{bmatrix} 1 & 0.2 \end{bmatrix}, \tag{18}$$

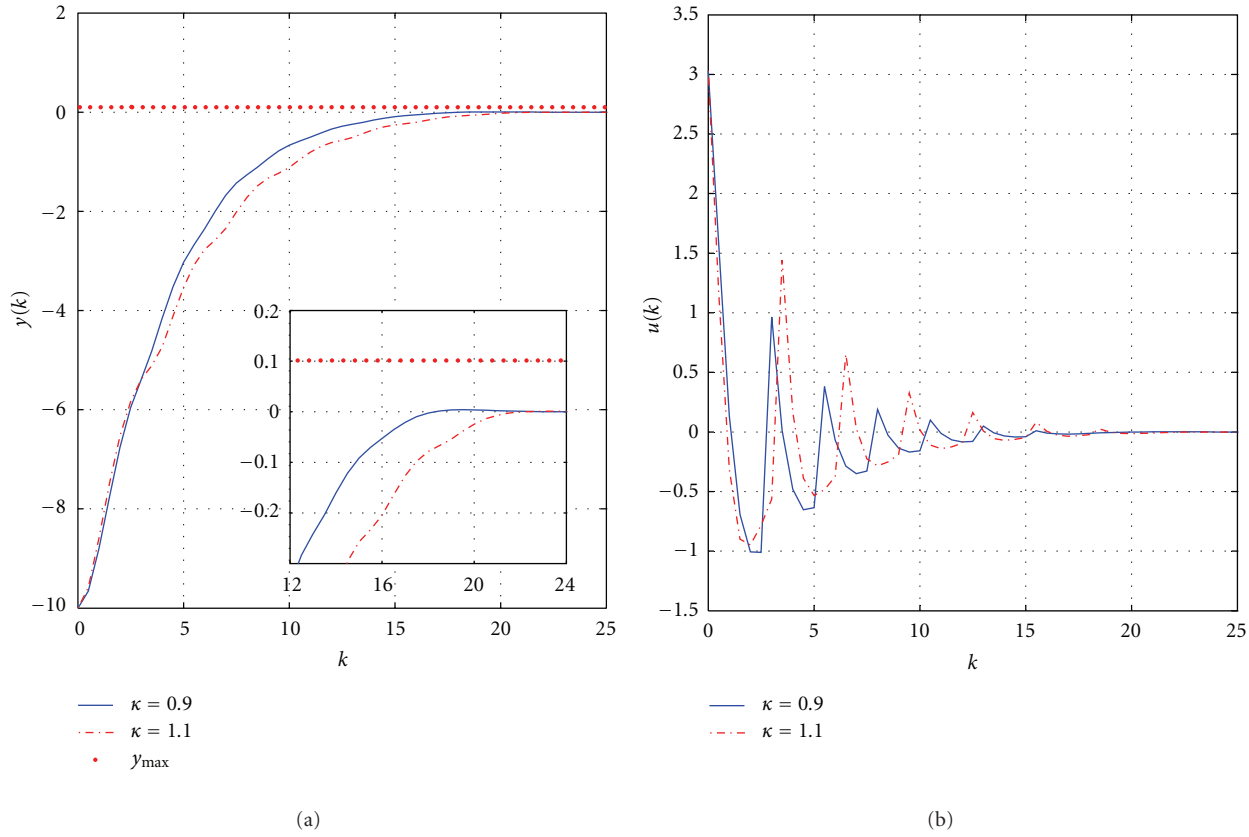
where T is the sampling time, and κ is an uncertain gain parameter.

The initial condition is set to $x(0) = [-10\ 0^T]$, the sampling time is $T = 0.5$ s, and the constraints are defined as $\overline{u} = u_{\max} = u_{\min} = 5$, $y_{\min} = -10$ and $y_{\max} = 0.1$. The uncertain parameter κ is assumed to be in the range $0.9 - 1.1$. The weight matrices of the controller are defined as

$$W_x = \begin{bmatrix} 100 & 0 \\ 0 & 0.01 \end{bmatrix}, \qquad W_u = 100. \tag{19}$$

In this case, the set X_s of possible equilibrium points is given by

$$X_s = \{(x_1, x_2) \mid x_2 = 0\}. \tag{20}$$



(a) (b)

Figure 5: Simulation results using the proposed strategy: (a) output and (b) control signals.

Table 1: Pseudoreferences and associated output constraints.

i	r_i	$\overline{y}_i$
0	−4.95	5.05
1	−2.425	2.525
2	−1.1625	1.2625
3	−0.53125	0.63125
4	−0.21563	0.31562
5	−0.057813	0.15781
6	0	0.1

In fact, the state variables x_1 and x_2 can be regarded as position and velocity, respectively, and thus the equilibrium can only be achieved if the velocity x_2 is zero.

The pseudoreferences r_i and associated symmetric output constraints $\overline{y}_i$, which define the controllers $i = 0, \ldots, 6$, are presented in Table 1. The simulations were performed in the Matlab environment.

5. Results and Discussions

As discussed in Section 3, a possible approach to address asymmetric constraints consists of adopting the conservative bounds defined in (14). In the present example, such procedure amounts to setting $\overline{y} = y_{\max} = 0.1$. However, the resulting optimization problem $\mathbb{P}(x(0))$ becomes infeasible. In fact, given the control constraint $-5 < u(k) < 5$, it is not possible to steer the output from $y(0) = -10$ to the range $[-0.1, 0.1]$ in a single sampling period.

On the other hand, if the constraints are relaxed by setting $\overline{y} = -y_{\min} = 10$, there is no guarantee that the resulting output trajectory will remain within the original $(y_{\min}, y_{\max})$ bounds. In fact, the inset in Figure 4 shows that such a relaxation of the output constraints does lead to a violation of the original upper bound for $\kappa = 0.9$.

These findings motivate the adoption of the proposed strategy for handling the asymmetric output constraints. Figures 5(a) and 5(b) present the simulation results obtained by using the CPR algorithm. As can be seen, both the output and control constraints were properly enforced, even by using the extreme values of κ in the simulation.

The commutation between the successive pseudoreferences is illustrated in Figure 6. This graph indicates that the commutation from one pseudoreference to the next occurs in finite time, as expected. The final pseudoreference ($i = 6$) corresponds to the origin, which is the desired equilibrium point for the regulation problem.

6. Conclusion

This paper presented a strategy for handling asymmetric output constraints within the scope of an LMI-based RMPC

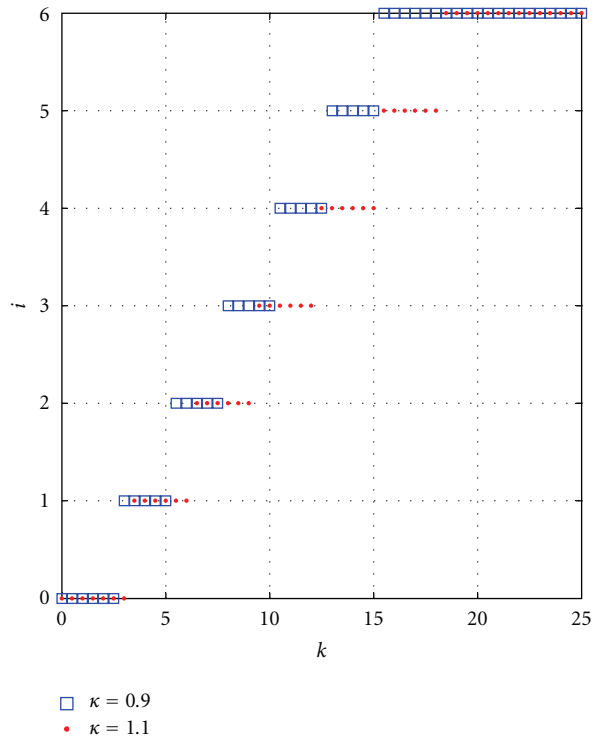


Figure 6: Commutation between pseudoreferences.

scheme. For this purpose, a procedure for defining a sequence of pseudoreferences was devised, along with a rule for commutation from one pseudoreference to the next. The proposed approach guarantees constraint satisfaction and convergence of the state trajectory to the origin, provided that the algorithm for determination of the pseudoreferences is feasible. The results of a numerical simulation study indicated that the proposed procedure may be a suitable alternative to the use of either more conservative constraints (which may lead to infeasibility issues) or more relaxed constraints (which do not guarantee satisfaction of the original restrictions). Future research could be concerned with the extension of the proposed approach to multiple input-multiple output (MIMO) systems. In this case, it may be necessary to define different pseudoreferences for each constrained output under consideration.

Acknowledgments

The authors gratefully acknowledge the support of FAPESP (scholarship 2008/54708-6 and grant 2006/58850-6), CAPES (Pró-Engenharias), and CNPq (research fellowships).

References

[1] E. F. Camacho and C. Bordons, *Model Predictive Control*, Spring, London, UK, 1999.

[2] J. M. Maciejowski, *Predictive Control with Constraints*, Prentice-Hall, Harlow, UK, 2002.

[3] J. A. Rossiter, *Model-based predictive control—a practical approach*, CRC Press, New York, NY, USA, 2003.

[4] M. S. M. Cavalca, R. K. H. Galvão, and T. Yoneyama, "Robust model predictive control for a magnetic levitation system employing linear matrix inequalities," in *ABCM Symposium Series in Mechatronics*, vol. 4, Rio de Janeiro, Brazil, 2010.

[5] A. H. González, J. L. Marchetti, and D. Odloak, "Robust model predictive control with zone control," *IET Control Theory and Applications*, vol. 3, no. 1, pp. 121–135, 2009.

[6] M. A. Rodrigues and D. Odloak, "Output feedback MPC with guaranteed robust stability," *Journal of Process Control*, vol. 10, no. 6, pp. 557–572, 2000.

[7] M. V. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.

[8] L. Özkan and M. V. Kothare, "Stability analysis of a multimodel predictive control algorithm with application to control of chemical reactors," *Journal of Process Control*, vol. 16, no. 2, pp. 81–90, 2006.

[9] Z. Wan and M. V. Kothare, "A framework for design of scheduled output feedback model predictive control," *Journal of Process Control*, vol. 18, no. 3-4, pp. 391–398, 2008.

[10] V. T. Minh and F. B. M. Hashim, "Robust model predictive control schemes for tracking setpoints," *Journal of Control Science and Engineering*, vol. 2010, Article ID 6494361, 2010.

[11] M. S. M. Cavalca, R. K. H. Galvão, and T. Yoneyama, "Integrator resetting with guaranteed feasibility for an LMI-based robust model predictive control approach," in *Proceedings of the 18th Mediterranean Conference on Control and Automation*, pp. 634–639, Marrakech, Morocco, 2010.

[12] Z. Wan and M. V. Kothare, "An efficient off-line formulation of robust model predictive control using linear matrix inequalities," *Automatica*, vol. 39, no. 5, pp. 837–846, 2003.

[13] Z. Wan and M. V. Kothare, "Robust output feedback model predictive control using off-line linear matrix inequalities," *Journal of Process Control*, vol. 12, no. 7, pp. 763–774, 2002.

Hindawi Publishing Corporation
Journal of Control Science and Engineering
Volume 2012, Article ID 729748, 13 pages
doi:10.1155/2012/729748

Research Article

Efficient Model Predictive Algorithms for Tracking of Periodic Signals

Yun-Chung Chu and Michael Z. Q. Chen

Department of Mechanical Engineering, The University of Hong Kong, Hong Kong

Correspondence should be addressed to Michael Z. Q. Chen, mzqchen@hku.hk

Received 30 June 2011; Accepted 31 August 2011

Academic Editor: Baocang Ding

This paper studies the design of efficient model predictive controllers for fast-sampling linear time-invariant systems subject to input constraints to track a set of periodic references. The problem is decomposed into a steady-state subproblem that determines the optimal asymptotic operating point and a transient subproblem that drives the given plant to this operating point. While the transient subproblem is a small-sized quadratic program, the steady-state subproblem can easily involve hundreds of variables and constraints. The decomposition allows these two subproblems of very different computational complexities to be solved in parallel with different sampling rates. Moreover, a receding horizon approach is adopted for the steady-state subproblem to spread the optimization over time in an efficient manner, making its solution possible for fast-sampling systems. Besides the conventional formulation based on the control inputs as variables, a parameterization using a dynamic policy on the inputs is introduced, which further reduces the online computational requirements. Both proposed algorithms possess nice convergence properties, which are also verified with computer simulations.

1. Introduction

One of the most attractive features of model predictive control (MPC) is its ability to handle constraints [1]. Many other control techniques are conservative in handling constraints, or even try to avoid activating them, thus, sacrificing the best performance that is achievable. MPC, on the contrary, tends to make the closed-loop system operate near its limits and hence produces far better performance. This property of MPC gives it the strength in practice, leading to a wide acceptance by the industry.

A very good example of system operating near its limits is a plant being driven by periodic signals to track periodic references. Under this situation, some of the system constraints will be activated repeatedly, and the optimal operating control signal is far from trivial. Just clipping the control signal to fit into the system constraints produces inferior performance typically. And the loss being considered here is not just a transient loss due to sudden disturbances, but indeed a steady-state loss due to a suboptimal operating point. Therefore, the loss is on long term and severe.

On the other hand, the successful real-life applications of MPC are mostly on systems with slower dynamics such as industrial and chemical processes [2]. The reason is simply that MPC requires a constrained optimization to be carried out online in a receding horizon fashion [3, 4]. Therefore, to apply MPC to fast-sampling systems, the computational power needed will be substantial. In any case, because of its great success in slow-sampling systems, the trend to extend MPC to fast-sampling systems is inevitable, and many recent researches have been carried out to develop efficient methods to implement MPC in such cases. While some of these works focus on unloading the computational burdens [5–9], others emphasize on code optimization [10–12] and new algorithmic paradigms [13–17].

If MPC is applied to the tracking of periodic signals in a receding horizon fashion, the horizon length will be related to the period length, and a long period will imply an online optimization problem of many variables and constraints. For a fast-sampling system, it is essentially an attempt to solve a very big computational problem within a very small time frame. In this paper, we shall analyze the structure of this

problem and then propose two efficient algorithms for the task. They aim to make the application of MPC to a fast-sampling system possible by a slight sacrifice on the transient performance, but the optimal or near-optimal steady-state performance of periodic tracking will be maintained.

In Section 2, the mathematical formation of the problem will be presented. The two algorithms, one based on the concept of receding horizon quadratic programming and the other based on the idea of dynamic MPC policy, will be presented in Sections 3 and 4, respectively. A comment on the actual implementation will be given in Section 5, followed by some computer simulations in Section 6 to illustrate several aspects of the proposed algorithms. Finally, Section 7 concludes the paper.

To avoid cumbersome notations like $u(k \mid k), u(k+1 \mid k), \ldots, u(k+N_u-1 \mid k)$, the MPC algorithms in this paper will only be presented as if the current time is $k = 0$, and we shall write $u(0), u(1), \ldots, u(N_u - 1)$ instead. The reader is asked to bear in mind that the algorithm is actually to be implemented in a receding horizon fashion.

2. Problem Formulation

Consider a linear time-invariant plant subject to a periodic disturbance:

$$x^+ = Ax + B_1 w + B_2 u, \tag{1}$$

$$y = Cx + D_1 w + D_2 u, \tag{2}$$

$$x(0) = x_0, \tag{3}$$

where the superscript + denotes the time-shift operator, that is,

$$x^+(k) = x(k+1), \tag{4}$$

and the disturbance w is measurable and periodic with period N_p. The control objective is to construct a control signal u such that the plant output y will track a specific periodic reference r of the same period N_p asymptotically with satisfactory transient performance. The control input u is also required to satisfy some linear inequality constraints (e.g., to be within certain bounds). The reference r is not necessarily fixed but may be different for different disturbance w. (For that reason, it may be more appropriate to call w an exogenous signal rather than a disturbance).

The algorithms developed in this paper are motivated by the following situations:

(1) the period N_p is very long compared with the typical transient behaviours of the closed-loop system;

(2) the linear inequality constraints on u are persistently active, that is, for any given $\widetilde{k}$, there exists a $k > \widetilde{k}$ such that $u(k)$ will meet at least one of the associated linear *equality* constraints;

(3) there is not sufficient computational power to solve the associated quadratic program completely within one sampling interval unless both the control horizon and the prediction horizon are much shorter than N_p.

As a matter of fact, without the above considerations and restrictions, the problem is not very challenging and can be tackled with standard MPC approaches for linear systems.

The underlying idea of the approach proposed in this paper is that since the transient behaviour of the closed-loop system is expected to be much shorter than the period N_p, we should decompose the original control problem into two: one we call the steady-state subproblem and the other we call the transient subproblem. Hence, the transient subproblem can be solved with a control horizon and a prediction horizon much shorter than N_p. While the steady-state subproblem is still very computationally intensive and cannot be solved within one sampling interval, it is not really urgent compared with the transient subproblem, and its computation can be spread over several sampling intervals. Indeed, the two subproblems need not be synchronized even though the transient subproblem depends on the solution to the steady-state subproblem due to the coupled input constraints. The former will utilize the latter whenever the latter is updated and made available to the former. It is only that the transient control will try to make the plant output y track a *suboptimal* reference if the optimal steady-state control is not available in time.

Now let us present the detailed mathematical formulation of our proposed method. Naturally, since both w and r are periodic with period N_p, the solution u should also be periodic of the same period asymptotically, that is, there should exist a periodic signal $u_s(k)$ such that

$$\lim_{k\to\infty} (u(k) - u_s(k)) = 0. \tag{5}$$

Let x_s and y_s be the corresponding asymptotic solutions of x and y, and they obviously satisfy the dynamics inherited from (1) and (2):

$$\begin{aligned} x_s^+ &= Ax_s + B_1 w + B_2 u_s, \\ y_s &= Cx_s + D_1 w + D_2 u_s, \\ x_s(0) &= x_s\left(N_p\right). \end{aligned} \tag{6}$$

Ideally, we want $y_s = r$ but this might not be achievable when u_s is required to satisfy the specific linear inequality constraints. Therefore, following the spirit of MPC, we shall find u_s, such that

$$J_s = \sum_k e_s(k)^{\mathrm{T}} Q e_s(k) \tag{7}$$

is minimized for some positive definite matrix Q, where $e_s(k)$ is the asymptotic tracking error defined by

$$e_s = y_s - r, \tag{8}$$

and the summation in (7) is over one period of the signals. This is what we call the steady-state subproblem. In Sections 3 and 4 below, we shall present two different approaches to this steady-state subproblem, with an emphasis on their computational efficiencies.

Once the steady-state signals are known, the transient signals defined by

$$u_t = u - u_s, \qquad x_t = x - x_s, \qquad y_t = y - y_s, \tag{9}$$

satisfy the dynamics

$$\begin{aligned} x_t^+ &= Ax_t + B_2 u_t, \\ y_t &= Cx_t + D_2 u_t, \\ x_t(0) &= x_0 - x_s(0), \end{aligned} \tag{10}$$

derived from (1)–(3), subject to the original linear inequality constraints being applied to $u_t(k) + u_s(k)$. Since the control horizon and the prediction horizon for this transient subproblem are allowed to be much shorter than N_p, it can be tackled with existing MPC algorithms.

It is important to note that in this steady-state/transient decomposition, the steady-state control u_s is actually a *feedforward* control signal determined from w and r, whereas the transient control u_t is a *feedback* control signal depending on x. As an unstable plant can only be stabilized by feedback, but the main interest of the current paper is the computational complexity of the steady-state subproblem, we shall not discuss in depth the stabilization issue, which has been studied quite extensively in the MPC literature. Typically, stabilizability of a constrained system using MPC would be cast as the feasibility of an associated optimization problem. For the numerical example in Section 6, we shall conveniently pick a plant where A is already stable, and hence the following quadratic cost may be adopted for the transient subproblem:

$$J_t = \sum_{k=0}^{N_u-1} \left(y_t(k)^{\mathrm{T}} Q y_t(k) + u_t(k)^{\mathrm{T}} R u_t(k) \right) + x_t(N_u)^{\mathrm{T}} P_T x_t(N_u), \tag{11}$$

where N_u is the control horizon with $N_u \ll N_p$, Q and R are chosen positive definite matrices, and P_T is the (weighted) observability gramian obtained from the Lyapunov equation

$$A^{\mathrm{T}} P_T A - P_T + C^{\mathrm{T}} Q C = 0. \tag{12}$$

The minimization of J_t is simply a standard quadratic program over $u_t(0), u_t(1), \ldots, u_t(N_u - 1)$ for a given $x_t(0)$. The situation will be more complicated when A is not stable, but one well-known approach is to force the unstable modes to zero at the end of the finite horizon [18].

Remark 1. Essentially, the choice of the cost function (11) with P_T from (12) for a stable A means that the projected control action after the finite horizon is set to zero, that is, $u_t(k) = 0$ for $k \geq N_u$ since the "tail" of the quadratic cost is then given by

$$\begin{aligned} \sum_{k=N_u}^{\infty} y_t(k)^{\mathrm{T}} Q y_t(k) &= \sum_{k=N_u}^{\infty} x_t(k)^{\mathrm{T}} C^{\mathrm{T}} Q C x_t(k) \\ &= x_t(N_u)^{\mathrm{T}} P_T x_t(N_u). \end{aligned} \tag{13}$$

This terminal policy is valid because the steady-state subproblem has already required that u_s satisfies the linear inequality constraints imposed on u. Hence, J_t is obviously a Lyapunov function which will be further decreased by the receding horizon implementation when the future control $u_t(N_u)$ turns into an optimization variable from zero.

Remark 2. We have deliberately omitted the R-matrix in the steady-state cost J_s in (7). The reason is simply that we want to recover the standard linear solution (for perfect asymptotic tracking) as long as u_s does not hit any constraint.

3. Steady-State Subproblem: A Receding Horizon Quadratic Programming Approach

When the periodic disturbance w is perfectly known, the steady-state subproblem is also a conceptually simple (but computationally high-dimensional) quadratic program. One way to know w is simply to monitor and record it over one full period. This, however, does not work well if w is subject to sudden changes. For example, the plant to be considered in our simulations in Section 6 is a power quality device called Unified Power Quality Conditioner [19], where w consists of the supply voltage and the load current of the power system, and both may change abruptly if there are supply voltage sags/swells and variations in the load demands. Indeed, the main motivation of the receding horizon approach in MPC is that things never turn out as expected and the control signal should adapt in an autonomous manner. If the suddenly changed disturbance w can be known precisely only after one full period of observation, the transient response of the steady-state subproblem (not to be confused with the transient subproblem described in Section 2) will be unsatisfactory.

One way to overcome this is to introduce an exogenous model for the signals w and r, as adopted in [19]. Specifically, we construct a state-space model:

$$v^+ = A_v v, \tag{14}$$

$$w = C_w v, \tag{15}$$

$$r = C_r v, \tag{16}$$

and assume that both w and r are generated from this model. Since w and r are periodic with period N_p, we have

$$A_v^{N_p} = I. \tag{17}$$

One simple (but not the only) way to construct A_v, as demonstrated similarly in [19] in the continuous-time case, is to make A_v a block-diagonal matrix with each block taking the form:

$$\begin{bmatrix} \cos(n\omega T_s) & -\sin(n\omega T_s) \\ \sin(n\omega T_s) & \cos(n\omega T_s) \end{bmatrix}, \tag{18}$$

where n is an integer, T_s is the sampling time and $\omega T_s \times N_p = 2\pi$. Then the matrices C_w and C_r are just to sum up their respective odd components of v. This essentially performs a Fourier decomposition of the signals w and r, and hence their approximations by $C_w v$ and $C_r v$ will be arbitrarily good when more and more harmonics are included in the model.

Based on the exogenous model (14)–(17), an observer can be easily constructed to generate (an estimate of) v from the measurements of w and r. From $v(0)$, the model (14)–(17) can then generate predictions of $w(0), w(1), \ldots, w(N_p - 1)$ and $r(0), r(1), \ldots, r(N_p - 1)$, and these can be used to find $u_s(0), u_s(1), \ldots, u_s(N_p - 1)$ by the quadratic program. The use of the exogenous model (14)–(17) typically allows the changed w and r to be identified much sooner than the end of one full period.

The quadratic program for the steady-state subproblem can be written as follows:

$$\min_{\mathbf{u}_s(0)} J_s,$$

$$J_s := \begin{bmatrix} \mathbf{u}_s(0) \\ v(0) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{H} & \mathbf{F} \\ \mathbf{F}^{\mathrm{T}} & \mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{u}_s(0) \\ v(0) \end{bmatrix},$$

$$\mathbf{u}_s(0) := \begin{bmatrix} u_s(0) \\ u_s(1) \\ \vdots \\ u_s\left(N_p - 1\right) \end{bmatrix}, \tag{19}$$

$$\text{subject to} \quad \mathbf{a}_j^{\mathrm{T}} \mathbf{u}_s(0) \le \mathbf{b}_j, \qquad j = 1, 2, \ldots, N_m, \tag{20}$$

where N_m is the total number of linear inequality constraints. Note that since we assume only input but not state constraints for our original problem, (20) does not depend on $v(0)$ and, hence, the feasibility of any $\mathbf{u}_s(0)$ remains the same even if there is an abrupt change in $v(0)$ (i.e., if $v(0)$ is different from the predicted value from (14) and $v(-1)$). Furthermore, the active set of constraints remains the same.

Definition 3. The active set $\mathcal{A}(\mathbf{u}_s(0))$ of any feasible $\mathbf{u}_s(0)$ satisfying (20) is the subset of $\{1, 2, \ldots, N_m\}$ such that $j \in \mathcal{A}(\mathbf{u}_s(0))$ if and only if $\mathbf{a}_j^{\mathrm{T}} \mathbf{u}_s(0) = \mathbf{b}_j$.

Next, we present a one-step active set algorithm to solve the quadratic program (19)-(20) partially.

Algorithm 4. Given an initial feasible $\mathbf{u}_s(0)$ and a working set $\mathcal{W}_0 \subset \mathcal{A}(\mathbf{u}_s(0))$. Let the set of working constraints

$$\mathbf{a}_j^{\mathrm{T}} \mathbf{u}_s(0) \le \mathbf{b}_j, \quad j \in \mathcal{W}_0, \tag{21}$$

be represented by

$$\mathbf{A}_0 \mathbf{u}_s(0) \le \mathbf{b}_0, \tag{22}$$

where the inequality sign applies componentwise, that is, each row of $\mathbf{A}_0$, $\mathbf{b}_0$ represents a working constraint in (21).

(1) Compute the gradient

$$\mathbf{g}_0 = \mathbf{H}\mathbf{u}_s(0) + \mathbf{F}v(0), \tag{23}$$

and the null space of $\mathbf{A}_0$, denoted $\mathbf{Z}_0$ by

$$\mathbf{A}_0 \mathbf{Z}_0 = 0. \tag{24}$$

If $\mathbf{Z}_0^{\mathrm{T}} \mathbf{g}_0 \approx 0$, go to step (5).

(2) Compute a search direction $\mathbf{w}_0 = \mathbf{Z}_0 \hat{\mathbf{w}}_0$, where

$$\left(\mathbf{Z}_0^{\mathrm{T}} \mathbf{H} \mathbf{Z}_0\right) \hat{\mathbf{w}}_0 + \mathbf{Z}_0^{\mathrm{T}} \mathbf{g}_0 = 0. \tag{25}$$

(3) Let

$$\alpha_0 = \min_{\substack{j \in \mathcal{A}(\mathbf{u}_s(0)) \setminus \mathcal{W}_0 \\ \text{s.t. } \mathbf{a}_j^{\mathrm{T}} \mathbf{w}_0 > 0.}} \frac{\mathbf{b}_j - \mathbf{a}_j^{\mathrm{T}} \mathbf{u}_s(0)}{\mathbf{a}_j^{\mathrm{T}} \mathbf{w}_0}. \tag{26}$$

(4) If $\alpha_0 \ge 1$, go to step (5). Otherwise, update $\mathbf{u}_s(0)$ to $\mathbf{u}_s^*(0)$ by

$$\mathbf{u}_s^*(0) = \mathbf{u}_s(0) + \alpha_0 \mathbf{w}_0, \tag{27}$$

and add a (blocking) constraint to $\mathcal{W}_0$ to form a new working set $\mathcal{W}_0^*$ according to the method described in Remark 7 below. Quit.

(5) Update $\mathbf{u}_s(0)$ to $\mathbf{u}_s^*(0)$ by

$$\mathbf{u}_s^*(0) = \mathbf{u}_s(0) + \mathbf{w}_0. \tag{28}$$

Compute the Lagrange multiplier λ_0 from

$$\mathbf{A}_0^{\mathrm{T}} \lambda_0 + \mathbf{g}_0 = 0 \tag{29}$$

to see whether any component of λ_0 is negative. If yes, remove one of the constraints corresponding to a negative component of λ_0 from $\mathcal{W}_0$ to form a new working set $\mathcal{W}_0^*$ according to the method described in Remark 7 below. Quit.

Algorithm 4 can be interpreted as follows. It solves the equality-constrained quadratic program:

$$\min_{\mathbf{u}_s'(0)} J_s',$$

$$J_s' := \begin{bmatrix} \mathbf{u}_s'(0) \\ v(0) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{H} & \mathbf{F} \\ \mathbf{F}^{\mathrm{T}} & \mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{u}_s'(0) \\ v(0) \end{bmatrix}, \tag{30}$$

$$\text{subject to} \quad \mathbf{a}_j^{\mathrm{T}} \mathbf{u}_s'(0) = \mathbf{b}_j, \quad j \in \mathcal{W}_0,$$

and then searches along the direction

$$\mathbf{w}_0 = \mathbf{u}_s'(0) - \mathbf{u}_s(0), \tag{31}$$

until it is either blocked by a constraint not in $\mathcal{W}_0$ (step (4)) or the optimal $\mathbf{u}_s'(0)$ is reached (step (5)). This is indeed a single step of the standard active set method (the null-space approach) for quadratic programming [20, 21] except for the modifications that will be detailed in Remark 7 below. In other words, if we apply Algorithm 4 repeatedly to the new $\mathbf{u}_s^*(0)$, $\mathcal{W}^*(0)$, it will converge to the solution of the original inequality-constrained quadratic program (19)-(20) within a finite number of steps, and the cost function J_s is strictly decreasing. However, here we only apply a single step of the active set method due to the limited computational power available within one sampling interval. Furthermore, we do not even assume that the single step of computation

can be completed within T_s. Let N_aT_s be the time required or allowed to carry out Algorithm 4. To complete the original quadratic program (19)-(20) in a receding horizon fashion, we need to forward $\mathbf{u}_s^*(0)$, $\mathcal{W}^*(0)$ to $\mathbf{u}_s(N_a)$, $\mathcal{W}(N_a)$ by rotating the components of $\mathbf{u}_s^*(0)$ by an amount of N_a since it is supposed to be periodic:

$$\begin{bmatrix} u_s(N_a) \\ u_s(N_a+1) \\ \vdots \\ u_s\left(N_p-1\right) \\ u_s\left(N_p\right) \\ u_s\left(N_p+1\right) \\ \vdots \\ u_s\left(N_a+N_p-1\right) \end{bmatrix} = \begin{bmatrix} u_s^*(N_a) \\ u_s^*(N_a+1) \\ \vdots \\ u_s^*\left(N_p-1\right) \\ u_s^*(0) \\ u_s^*(1) \\ \vdots \\ u_s^*(N_a-1) \end{bmatrix}. \tag{32}$$

Obviously, Algorithm 4 will then continue to solve an equivalent quadratic program as long as v strictly follows the exogenous dynamics (14). Hence we have the following convergence result.

Proposition 5. *Algorithm 4 together with the rotation of the components of* $\mathbf{u}_s^*(0)$ *in* (32) *will solve the quadratic program* (19)-(20) *in finite time as long as* $v(k)$ *satisfies the exogenous dynamics* (14).

Proof. From the argument above it is easy to see that as long as $v(k)$ follows the dynamics (14), the algorithm is consistently solving essentially the same quadratic program. So it remains to check that the convergence proof of the standard active set algorithm remains valid despite the modifications we shall detail in Remark 7, which is indeed the case. □

Of course, the most interesting feature of the receding horizon approach is that the solution will adapt autonomously to the new quadratic program if there is an abrupt change in v. Since constraint (20) is independent of v, an abrupt change in v will not destroy the feasibility of the previously determined $\mathbf{u}_s$ and the working set $\mathcal{W}$ determined previously also remains a valid subset of the active set. Hence, the receding horizon active set method will continue to work even though the objective function (19) has changed. However, if it is necessary to include not only control but also state constraints into the original problem formulation, we shall then require a quadratic programming algorithm (other than the active set method in its simplest form) that does not assume the initial guess to be feasible.

Remark 6. There could be two possible ways to arrange the steps in Algorithm 4. One is to update the working set $\mathcal{W}$ followed by $\mathbf{u}_s$, and the other is to update $\mathbf{u}_s$ followed by the working set $\mathcal{W}$. In the receding horizon framework, it might seem at first glance that the first choice is the right one, since we shall then avoid basing the optimization of $\mathbf{u}_s$ on an "outdated" working set if v happens to have changed. However, it turns out that the first choice is actually undesirable. One well-known "weakness" of the active set method is that it is not so easy to remove a constraint once it enters the working set $\mathcal{W}$. This becomes even more a concern in the receding horizon framework. If v has changed, and so has the objective function (19), the original stationary $\mathbf{u}_s^*$ obtained in step (5) may no longer be stationary, and it will require at least an additional iteration to identify the new stationary point before we can decide whether any constraint can be dropped from the working set or not. This will seriously slow down the transient response of the steady-state subproblem. Indeed, once v has changed, many of the constraints in the previous working set are no longer sensible, and it will be wiser to drop them hastily rather than being too cautious only to find much later that the constraints should still be dropped eventually.

Remark 7. One key element in the active set method of the quadratic program is to add or drop a constraint to or from the working set $\mathcal{W}$. The constraint being added belongs to the blocking set $\mathcal{B}$, defined as those constraints corresponding to the minimum α_0 in (26). Physically, they are the constraints that will be first encountered when we try to move $\mathbf{u}_s(0)$ to $\mathbf{u}_s'(0)$ in (31). The constraint being dropped belongs to the set $\mathcal{L}$, defined as those constraints corresponding to a negative component of the Lagrange multiplier in (29). The active set method will converge in finite time no matter which constraint in $\mathcal{B}$ will be added or which constraint in $\mathcal{L}$ will be dropped. One standard and popular choice in the conventional active set method is that the one in $\mathcal{L}$ corresponding to the most negative component of λ will be dropped, whereas the choice from $\mathcal{B}$ will be arbitrary. This is a very natural choice when there is no other consideration.

However, in the receding horizon framework, one other (and in fact important) consideration emerges, which is the execution time of the control input(s) associated with a constraint. Specifically, if Algorithm 4 takes time N_aT_s to carry out, then $\mathcal{W}_0^*$ updated in the current iteration will be used to optimize $\mathbf{u}_s(N_a)$ in the next iteration,

$$\mathbf{u}_s(N_a) := \begin{bmatrix} u_s(N_a) \\ u_s(N_a+1) \\ \vdots \\ u_s\left(N_a+N_p-1\right) \end{bmatrix}, \tag{33}$$

but the outcome of that optimization is ready only at $k = 2N_a$, based on which the transient control u_t is computed. Suppose that the transient subproblem takes one sampling interval to solve, then the transient subproblem at $k = 2N_a$ will update $u(2N_a+1) = u_s(2N_a+1) + u_t(2N_a+1)$ (see Section 5 below for a more detailed discussion). Hence, the "time priority" of u_s is $2N_a+1, 2N_a+2, \ldots, N_p-1, 0, 1, \ldots, 2N_a$ and from this argument, we choose to drop the constraint in $\mathcal{L}$ that is associated with the first u_s in this sequence or to add the constraint in $\mathcal{B}$ that is associated with the last u_s in this sequence (of course the most negative Lagrange multiplier can still be used as the second criterion

if two constraints in $\mathcal{L}$ happen to have the same urgency). The proposal here aims to assign most freedom to the most urgent control input in the optimization, which makes sense in the receding horizon framework since the less urgent inputs may be reoptimized later.

Remark 8. Basically, the approach proposed in this section is to spread the original quadratic program over many intervals, so that each interval only carries out one iteration of the algorithm, and also to ensure that the quadratic program being solved is consistent when the prediction of the exogenous model is valid, but will migrate to a new quadratic program when there is a change in $v(k)$. It is worth mentioning that the original standard MPC is a static controller by nature, since the true solution of a complete quadratic program is independent of the MPC's decisions in the past (past decisions can help to speed up the computations but will not affect the solution), but by spreading the algorithm over several intervals, it is turned into a dynamic controller with internal state $\mathbf{u}_s(k)$, $\mathcal{W}(k)$.

4. Steady-State Subproblem: A Dynamic Policy Approach

The approach proposed in Section 3 optimizes $\mathbf{u}_s$. Consequently, the number of (scalar) variables being optimized is proportional to N_p. To further cut down the computations required, this section proposes another approach based on the idea of a dynamic policy, inspired by the works of [13, 22, 23]. This approach optimizes a smaller number of variables typically, and the number is independent from N_p, although the number of linear inequality constraints remains the same. In return, the optimization result is expected to be slightly inferior to that of Section 3 due to the reduction of variables (degree of freedom). However, it should be noted that the number of optimized variables in this second approach is adjustable based upon the designer's wish.

The central idea of the dynamic policy approach [13, 22, 23] is that instead of optimizing the control input directly, we generate the control input by a dynamic system of which the initial system state is optimized. This is similar to what we have done to w and r in Section 3. Specifically, we assume u_s is also generated from a state-space model:

$$\hat{v}^+ = A_{\hat{v}}\hat{v}, \tag{34}$$

$$u_s = C_{\hat{v}}\hat{v}, \tag{35}$$

where

$$A_{\hat{v}}^{N_p} = I. \tag{36}$$

This state-space model is designed *a priori* but the initial state $\hat{v}(0)$ will be optimized online. Obviously, the quadratic program (19)-(20) becomes

$$\min_{\hat{v}(0)} \hat{J}_s,$$

$$\hat{J}_s := \begin{bmatrix} \hat{v}(0) \\ v(0) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \hat{\mathbf{H}} & \hat{\mathbf{F}} \\ \hat{\mathbf{F}}^{\mathrm{T}} & \mathbf{G} \end{bmatrix} \begin{bmatrix} \hat{v}(0) \\ v(0) \end{bmatrix}, \tag{37}$$

$$\text{subject to} \quad \hat{\mathbf{a}}_j^{\mathrm{T}}\hat{v}(0) \le \mathbf{b}_j, \quad j = 1, 2, \ldots, N_m, \tag{38}$$

where

$$\hat{\mathbf{H}} = \hat{\mathcal{O}}^{\mathrm{T}}\mathbf{H}\hat{\mathcal{O}},$$

$$\hat{\mathbf{F}} = \mathbf{F}\hat{\mathcal{O}}, \tag{39}$$

$$\hat{\mathbf{a}}_j = \mathbf{a}_j\hat{\mathcal{O}}, \quad j = 1, 2, \ldots, N_m,$$

and $\hat{\mathcal{O}}$ is the observability matrix

$$\hat{\mathcal{O}} := \begin{bmatrix} C_{\hat{v}} \\ C_{\hat{v}}A_{\hat{v}} \\ \vdots \\ C_{\hat{v}}A_{\hat{v}}^{N_p-1} \end{bmatrix}. \tag{40}$$

The number of variables in this new quadratic program is the dimension of $\hat{v}(0)$, denoted by $n_{\hat{v}}$. If $A_{\hat{v}}$ is constructed from the method of Fourier decomposition described in Section 3, Shannon's sampling theorem implies that a sufficiently large but finite $n_{\hat{v}}$ will guarantee a full reconstruction of the original optimization variable $\mathbf{u}_s(0)$. On the other hand, a smaller $n_{\hat{v}}$ restricts the search to a lower dimensional subspace of $\mathbf{u}_s(0)$ and hence the optimization is easier but suboptimal.

One natural choice of the dynamics $A_{\hat{v}}$ is to make $A_{\hat{v}} = A_v$ in the exogenous model (14)–(17). Of course, it should be noted that constrained control is generally a nonlinear problem, and therefore the number of harmonics to be included in u_s may exceed that of w and r in order to achieve the true optimal performance. However, we could have overdesigned the exogenous model (14)–(17) to include more harmonics in A_v than necessary for w and r, thus making the choice $A_{\hat{v}} = A_v$ here not so conservative. The simulation results in Section 6 will demonstrate both cases.

It remains to choose the matrix $C_{\hat{v}}$ in (35). The one we suggest here is based on the linear servomechanism theory [24–26], which solves the linear version of our problem when there is no input constraint. Essentially, when there is no constraint, perfect asymptotic tracking (i.e., $y_s = r$ or $e_s = 0$) can be achieved by solving the regulator equation:

$$\begin{aligned} XA_v &= AX + B_1C_w + B_2U, \\ C_r &= CX + D_1C_w + D_2U, \end{aligned} \tag{41}$$

for the matrices X, U, and then let

$$u_s = Uv, \tag{42}$$

which also implies

$$x_s = Xv. \tag{43}$$

Therefore, to recover the optimal (or perfect) solution in the linear case when u_s does not hit any constraint, the state-space model of u_s may be chosen as

$$\begin{aligned} \tilde{v}^+ &= A_v\tilde{v}, \\ u_s &= U\tilde{v}. \end{aligned} \tag{44}$$

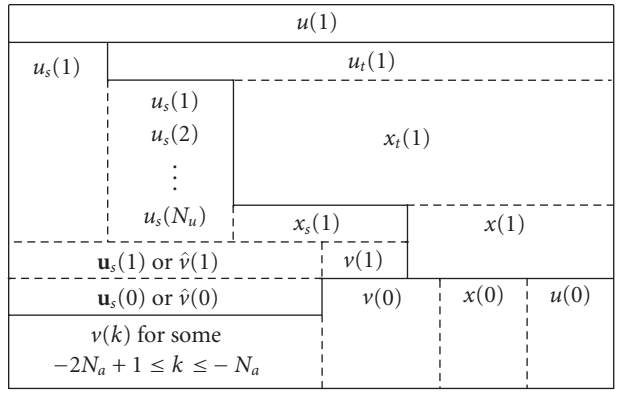


Figure 1: Derivations of unknown variables from known variables.

However, this state-space model is not guaranteed to be observable. When it is not, the resulting $\hat{\mathbf{H}}$ in (37) will become semidefinite instead of strictly positive definite. To overcome this, we suggest to perform an orthogonal state transformation

$$\begin{bmatrix} * \\ \hat{v} \end{bmatrix} = T\tilde{\hat{v}}, \tag{45}$$

to bring (44) to the Kalman decomposition form

$$\begin{aligned} \begin{bmatrix} * \\ \hat{v} \end{bmatrix}^+ &= \begin{bmatrix} * & * \\ 0 & A_{\hat{v}} \end{bmatrix} \begin{bmatrix} * \\ \hat{v} \end{bmatrix}, \\ u_s &= \begin{bmatrix} 0 & C_{\hat{v}} \end{bmatrix} \begin{bmatrix} * \\ \hat{v} \end{bmatrix}, \end{aligned} \tag{46}$$

and hence obtain a reduced-order model to become (34)-(35). It is easy to verify that $A_{\hat{v}}^{N_p} = I$ since

$$\begin{bmatrix} * & * \\ 0 & A_{\hat{v}} \end{bmatrix} \tag{47}$$

is upper block-triangular and

$$\begin{aligned} \begin{bmatrix} * & * \\ 0 & A_{\hat{v}} \end{bmatrix}^{N_p} &= \left(TA_v T^{-1}\right)^{N_p} \\ &= TA_v^{N_p} T^{-1} \\ &= I. \end{aligned} \tag{48}$$

Certainly, the discussion above only provides a suggestion of how to choose the state-space model for u_s, which we shall also adopt for our simulations in Section 6, but, in general, the designer should feel free to employ any valid state-space model to suit his problem.

Remark 9. Having reparameterized the quadratic program in terms of $\hat{v}(0)$ rather than $\mathbf{u}_s(0)$, we can apply a similar version of Algorithm 4 to (37)-(38). In other words, it is not necessary to solve the quadratic program completely within one sampling interval. Instead of rotating the components of $\mathbf{u}_s^*(0)$ to obtain $\mathbf{u}_s(N_a)$, we obtain $\hat{v}(N_a)$ by $A_{\hat{v}}^{N_a}\hat{v}^*(0)$.

5. Implementation Issues and Impacts on Transient Performance

Before we present the simulation results, let us comment on the impact of computational delay on the transient subproblem in Section 2. First of all, we assume that the transient quadratic program can be solved completely within one sampling interval. Therefore, despite the way we presented the cost function J_t in (11), in actual implementation we shall optimize $u_t(1), u_t(2), \ldots, u_t(N_u)$ based on $x_t(1)$ at time $k = 0$, instead of $u_t(0), u_t(1), \ldots, u_t(N_u - 1)$ based on $x_t(0)$. The unknown $x_t(1)$ can be projected from the known variables and system dynamics. After the optimization is carried out to obtain $u_t(1), u_t(2), \ldots, u_t(N_u)$, the control input to be executed at $k = 1$ will be $u(1) = u_s(1) + u_t(1)$. Bear in mind that all these calculations can only be based on the best knowledge of the signals at $k = 0$.

Figure 1 summarizes how the unknown variables can be computed from the known variables. The variables in each layer are derived from those variables directly below them, but in actual implementation it is sometimes possible to derive explicit formulas to compute the upper layer from the lower layer bypassing the intermediate layer, thus not requiring those intermediate calculations online. The variable on top is the control action $u(1)$, computed from the variables of the steady-state subproblem on the left, and those of the transient subproblem on the right, separated by a solid line. The variables in the bottom layer, $v(0)$ is provided by the observer described in Section 3, $x(0)$ is a measurement of the current plant state, and $u(0)$ is the control input calculated by the algorithm at $k = -1$. Note that to compute $u_t(1)$, the values of $u_s(1), u_s(2), \ldots, u_s(N_u)$ are needed to form the constraints for the transient quadratic program since the original linear inequality constraints apply to $u(k) = u_s(k) + u_t(k)$. On the other hand, $x_s(1)$ can be written as a linear function of $v(1)$ and $\mathbf{u}_s(1)$ (or $\hat{v}(1)$) explicitly. Finally, the steady-state subproblem requires a computational time of $N_a T_s$, implying that the solution $\mathbf{u}_s(0)$

provided by the steady-state subproblem at $k = 0$ is based on a measurement of $v(k)$ at some k between $-2N_a+1$ and $-N_a$. So in the worst case, $u(1)$ is based on some information as old as $v(-2N_a + 1)$, which corresponds to a worst-case delay of $2N_aT_s$. For instance, if $N_a = 1$, the control $u(k)$ is computed from a measurement of $x(k-1)$, $v(k-1)$, and $v(k-2)$.

Remark 10. Although we said in Section 2 that the transient subproblem was not the main interest of this paper, it is an ideal vehicle to demonstrate the power of MPC since the "useful freedom" of $u_t(k)$ may have been totally consumed by $u_s(t)$ when the latter hits a constraint. For example, the simulations to be discussed in Section 6 have the constraint

$$|u(k)| = |u_s(k) + u_t(k)| \le 1. \tag{49}$$

If $u_s(k)$ already saturates at ± 1, one side of $u_t(k)$ is lost but that could be the only side to make the reduction of J_t possible, thus forcing $u_t(k)$ to zero. So the input constraint does not only restrict the magnitude, but also the time instant to apply the transient control u_t. Such problem is extremely difficult for conventional control techniques.

6. Simulation Results

In this section we use an example to demonstrate the performance of our algorithms by computer simulations. The plant is borrowed from [19] and represents a power quality device called Unified Power Quality Conditioner (UPQC), which has the following continuous-time state-space model:

$$\begin{aligned}
\dot{x} &= Ax + B_1 w + B_2 u,\\
y &= Cx + D_1 w + D_2 u,\\
A &= \begin{bmatrix}
\frac{-R_l}{L_l} & 0 & 0 & \frac{-1}{L_l} & \frac{-1}{L_l}\\
0 & \frac{-R_{se}}{L_{se}} & 0 & \frac{-1}{L_{se}} & 0\\
0 & 0 & \frac{-R_{sh}}{L_{sh}} & 0 & \frac{-1}{L_{sh}}\\
\frac{1}{C_{se}} & \frac{1}{C_{se}} & 0 & 0 & 0\\
\frac{1}{C_{sh}} & 0 & \frac{1}{C_{sh}} & 0 & 0
\end{bmatrix},\\
B_1 &= \begin{bmatrix}
\frac{1}{L_l} & 0\\
0 & 0\\
0 & 0\\
0 & 0\\
0 & \frac{-1}{C_{sh}}
\end{bmatrix}, \qquad
B_2 = \begin{bmatrix}
0 & 0\\
\frac{V_{dc}}{2L_{se}} & 0\\
0 & \frac{V_{dc}}{2L_{sh}}\\
0 & 0\\
0 & 0
\end{bmatrix},\\
C &= \begin{bmatrix}
0 & 0 & 0 & 0 & 1\\
1 & 0 & 0 & 0 & 0
\end{bmatrix},\\
D_1 &= \begin{bmatrix}
0 & 0\\
0 & 0
\end{bmatrix}, \qquad
D_2 = \begin{bmatrix}
0 & 0\\
0 & 0
\end{bmatrix}.
\end{aligned} \tag{50}$$

The exogenous input w is composed of the supply voltage and the load current, which are periodic at 50 Hz but may consist of higher-order harmonics. The plant output y is composed of the load voltage and the supply current, which will be made to track designated pure sine waves of 50 Hz. The control input u is composed of two switching signals across the voltage source inverters (VSIs), both of which are required to satisfy the bounds $-1 \le u \le 1$. The general control objective is to maintain y to the desired waveforms despite possible fluctuations in w like supply voltage sags/swells or load demand changes. To apply the MPC algorithms proposed, we obtain a discrete-time version of the above state-space model by applying a sampling interval of $T_s = 0.2$ ms (i.e., 100 samples per period). Small-sized quadratic programs (such as our transient subproblem) can possibly be solved within such a short time thanks to the state-of-the-art code optimization [12], which reports sampling rates in the range of kHz, but to solve a big quadratic program like our steady-state subproblem we shall resort to the technique of Algorithm 4. Note that in our formulation, the transient subproblem and the steady-state subproblem can be solved in parallel. Although the optimization of u_t depends on u_s, the transient control $u_t(k+1)$ is computed from $\mathbf{u}_s(k)$ which is made available by the steady-state subproblem in the previous step. So it is independent of the current steady-state subproblem being solved.

As typical in a power system, we assume only odd harmonics in the supply voltage and the load current. Hence we can reduce the computations in the steady-state subproblem by the following easy modifications from the standard algorithms presented in Sections 3 and 4; N_p may be chosen to represent *half* of the period instead of the whole period, satisfying

$$x_s(0) = -x_s\left(N_p\right), \tag{51}$$

instead of (6), and

$$\begin{aligned}
A_v^{N_p} &= -I,\\
A_{\hat{v}}^{N_p} &= -I,
\end{aligned} \tag{52}$$

instead of (17) and (36), with $\omega T_s \times N_p = \pi$ instead of 2π. The rotation operation in (32) should also become

$$\begin{bmatrix}
u_s(N_a)\\
u_s(N_a+1)\\
\vdots\\
u_s\left(N_p-1\right)\\
u_s\left(N_p\right)\\
u_s\left(N_p+1\right)\\
\vdots\\
u_s\left(N_a+N_p-1\right)
\end{bmatrix} = \begin{bmatrix}
u_s^*(N_a)\\
u_s^*(N_a+1)\\
\vdots\\
u_s^*\left(N_p-1\right)\\
-u_s^*(0)\\
-u_s^*(1)\\
\vdots\\
-u_s^*(N_a-1)
\end{bmatrix}. \tag{53}$$

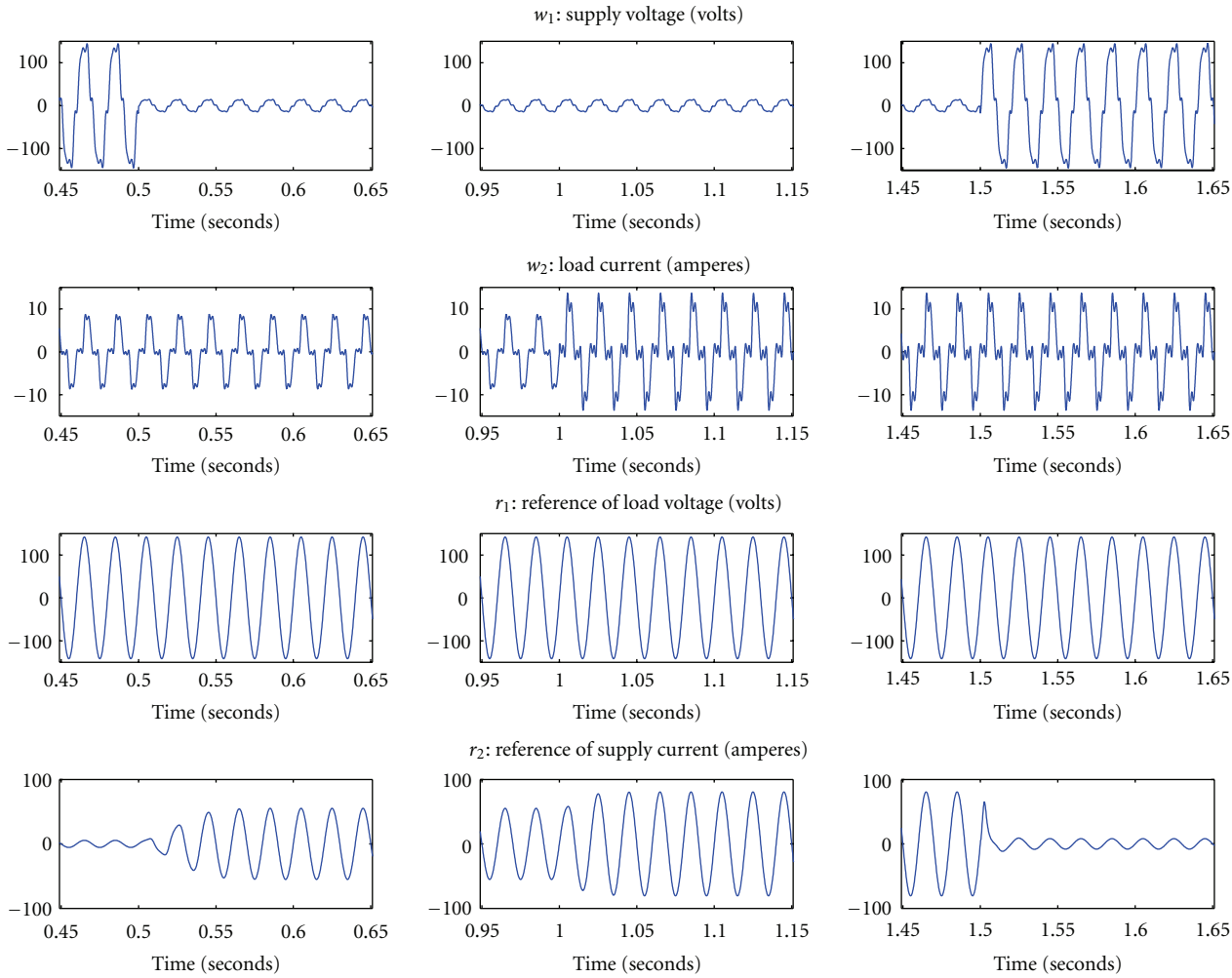


Figure 2: Simulation scenario. Voltage sag at $t = 0.5$ s; load demand changed at $t = 1.0$ s; sag cleared at $t = 1.5$ s.

Table 1: Values of the components of the UPQC.

Component	V_{dc}	L_{se}	L_{sh}	C_{se}	C_{sh}
Value	320 V	5.0 mH	1.2 mH	10 μF	20 μF

Table 2: Line impedance and VSI impedances of the UPQC.

Component	R_l	L_l	R_{se}	R_{sh}
Value	0.01 Ω	1.0 mH	0.01 Ω	0.01 Ω

This cuts down half of the scalar variables as well as constraints in the quadratic program.

The model parameters of the UPQC used in our simulations are summarized in Table 1 for the circuit components and Table 2 for the line and VSI impedances. They are the same as those values in [19], except for V_{dc} which we have changed from 400 V to 320 V so as to produce a saturated control u more easily. Note that V_{dc} is the DC-link voltage, which determines how big a fluctuation in the supply voltage or load current the UPQC can handle. In other words, saturation occurs when the UPQC is trying to deal with an unexpected voltage sag/swell or load demand that is beyond its designed capability.

The simulation scenario is summarized in Figure 2. Both the supply voltage and the load current consist of odd harmonics up to the 9th order. Despite the harmonics, it is desirable to regulate the load voltage to a *fixed* pure sine wave, whereas the supply current should also be purely sinusoidal, but its magnitude and phase are selected to maintain a power factor of unity and to match the supply active power to the active power demanded by the load, which means the reference of this supply current is w-dependent. The waveforms of both w and r are shown in Figure 2. The simulation scenario is designed such that the steady-state control u_s is not saturated at the beginning. At $t = 0.5$ s, a voltage sag occurs which reduces the supply voltage to 10% of its original value. The UPQC is expected to keep the load voltage unchanged but (gradually) increase the supply current so as to retain the original active power. This will drive u_s into a *slightly* saturated situation. At $t = 1.0$ s, the load demand increases, causing the reference of the supply current to increase again, and u_s will become *deeply* saturated. At $t = 1.5$ s, the voltage sag is cleared and the

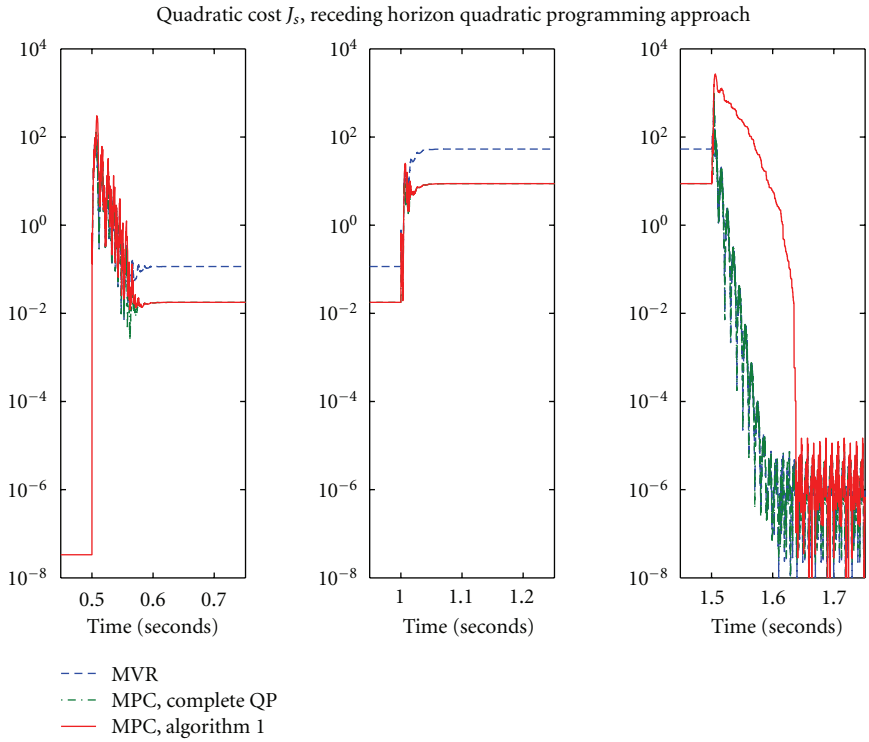


FIGURE 3: Quadratic cost J_s of the steady-state subproblem, the receding horizon quadratic programming approach.

supply voltage returns to its starting value, but the (new) load demand remains. Although the load demand is still higher than its initial value, the u_s required will just be within the bounds of ± 1, thus leaving the saturation region to return to the linear region. So, in short, u_s is expected to experience "*linear → slightly saturated → deeply saturated → linear but nearly saturated*" in this simulation scenario.

To evaluate the performance of our algorithms, we compare them to two other cases. In the first case, instead of Algorithm 4, the complete quadratic program is solved in each iteration of the steady-state subproblem every N_a-sampling intervals. We call this case the complete QP, and it serves to indicate how much transient performance has been sacrificed (in theory) by spreading the quadratic program over a number of iterations. In the second case, the constraints are totally ignored, such that the optimal $\mathbf{u}_s(0)$ in the steady-state subproblem is supposed to be

$$\mathbf{u}_s(0) = \begin{bmatrix} U \\ UA_v \\ \vdots \\ UA_v^{N_p-1} \end{bmatrix} v(0), \tag{54}$$

where U is the solution to the regulator equation (41). The transient subproblem can also be solved by

$$u_t = Fx_t, \tag{55}$$

where F is the optimal state-feedback gain minimizing the transient quadratic cost

$$\sum_{k=0}^{\infty} \left(y_t(k)^{\mathrm{T}} Q y_t(k) + u_t(k)^{\mathrm{T}} R u_t(k) \right). \tag{56}$$

However, the combined input u is still clipped at ± 1. We label this control law the multivariable regulator (MVR) following the linear servomechanism theory. This case serves to indicate how bad the quadratic cost J_s can be if a linear control law is used without taking the constraints into consideration.

Note that in both cases, the computational delays discussed in Section 5 will be in force, where $u(k+1)$ instead of $u(k)$ will be optimized and the steady-state control $\mathbf{u}_s$ is only updated every N_a-sampling intervals. In reality, of course the MVR should involve negligible computational delay, whereas the complete QP should need a longer time to solve than Algorithm 4, but we are merely using their associated quadratic costs here to analyze the behaviours of our algorithms.

Figure 3 plots the steady-state cost J_s of our first approach in Section 3 based on receding horizon quadratic programming together with the costs in the other two cases. N_a is assumed to be 3 in this simulation. The transient subproblem has a control horizon of $N_u = 5$, corresponding to 10 scalar variables and 20 scalar inequality constraints. On the other hand, the steady-state subproblem has $N_p = 50$ corresponding to 100 variables and 200 constraints. As shown in Figure 3, all J_s are zero prior to $t = 0.5$ s and should also settle down to zero after $t = 1.5$ s. It is observed

Table 3: Summary of J_s values for various cases studied.

	Steady-state cost J_s		N_a	Number of	
	at $t < 1.0$ s	at $t < 1.5$ s		Variables	Constraints
MVR	0.1156	53.276			
MPC, receding horizon quadratic programming	0.0178	8.7527	3	100 ($N_p \times 2$)	200 ($N_p \times 2 \times 2$)
MPC, dynamic policy (up to 9th harmonics)	0.0225	9.0273	1	$n_{\hat{v}} = 20$ ($5 \times 2 \times 2$)	200 ($N_p \times 2 \times 2$)
MPC, dynamic policy (up to 29th harmonics)	0.0182	8.7640	2	$n_{\hat{v}} = 60$ ($15 \times 2 \times 2$)	200 ($N_p \times 2 \times 2$)
Transient subproblem				10 ($N_u \times 2$)	20 ($N_u \times 2 \times 2$)

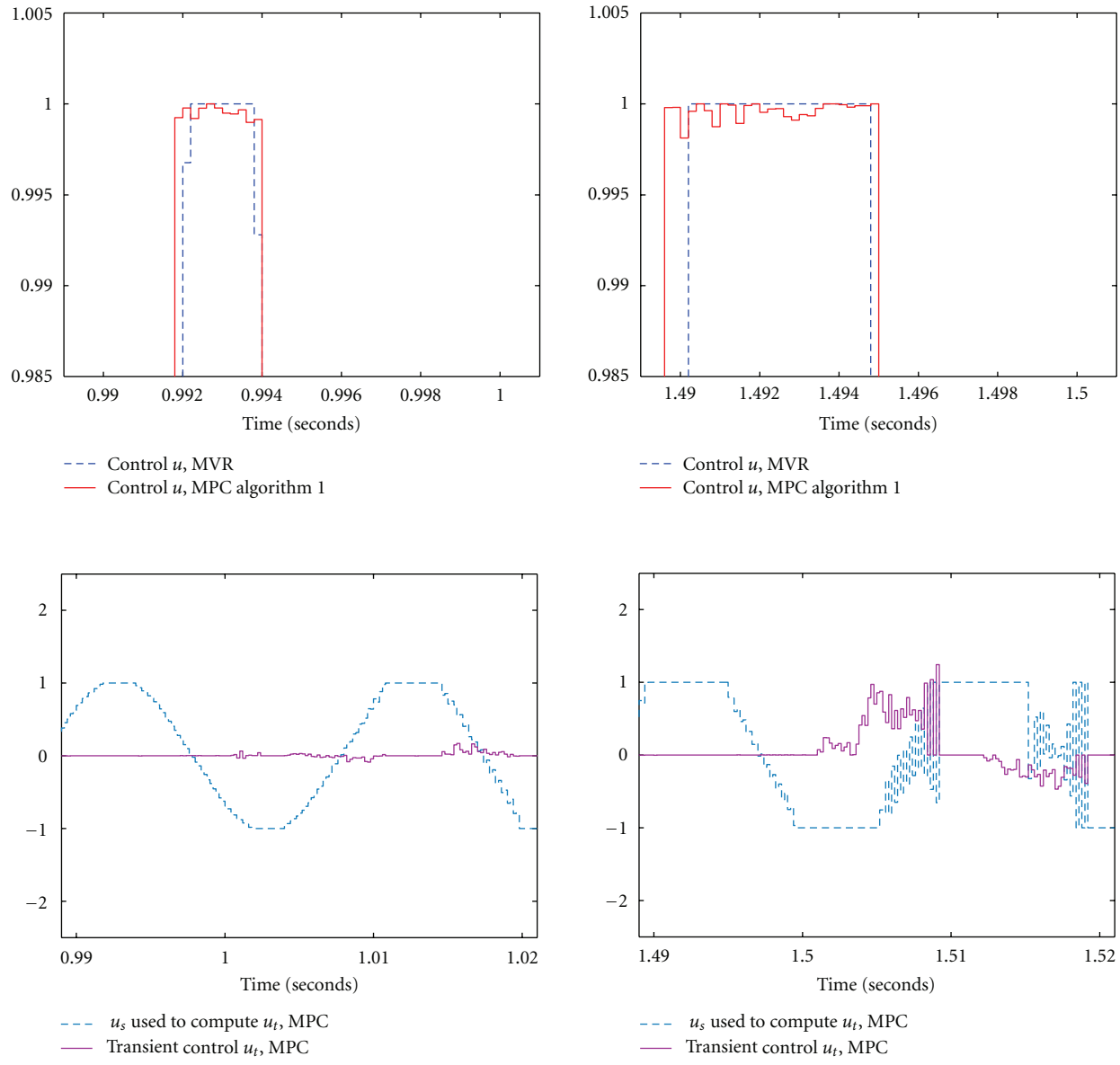


Figure 4: The first component of controls u and u_t before and after transitions at $t = 1.0$ s and $t = 1.5$ s.

that the transient response of our J_s is pretty close to that of the complete QP during the transitions from "linear" to "slightly saturated" and from "slightly saturated" to "deeply saturated," but is poorer when it tries to return from "deeply saturated" to "linear." This can probably be attributed to the weakness of the active set method in removing a constraint from the working set as discussed in Remark 6. Figure 3 also indicates that the MVR settles down to a much higher J_s value when a saturation occurs, due to its ignorance of the control constraints. The exact values of J_s just prior to $t = 1.0$ s and $t = 1.5$ s are summarized in Table 3, which are about 6-7 times the J_s values of our algorithm.

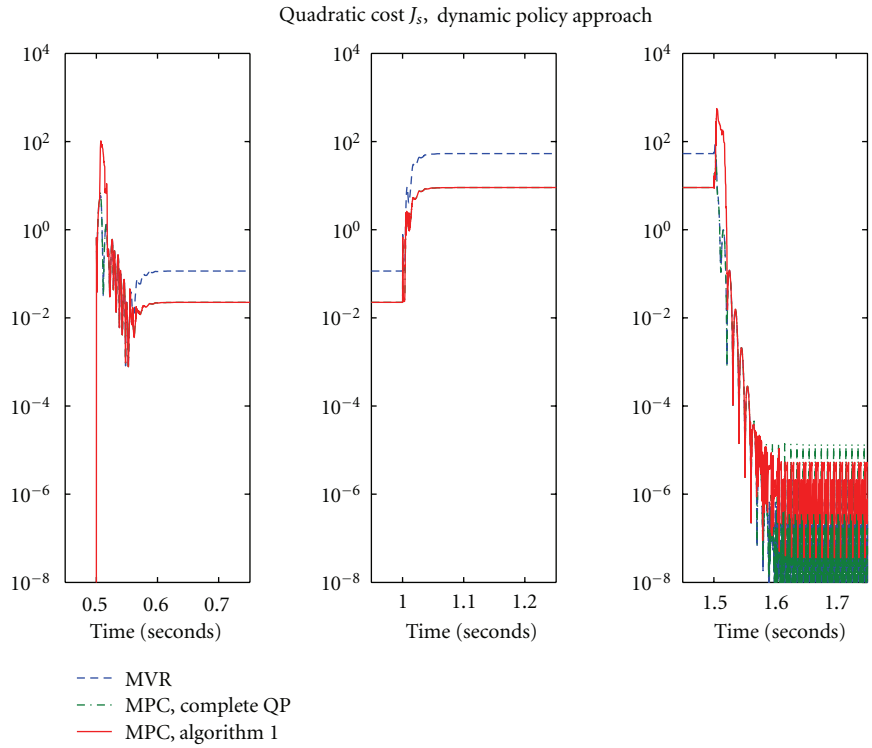


FIGURE 5: Quadratic cost J_s of the steady-state subproblem, the dynamic policy approach.

Figure 4 zooms into the first component of the control input u. The top two plots draw attentions to the steady-state control u_s (u is essentially u_s just prior to $t = 1.0$ s and $t = 1.5$ s). It is observed that the differences in u_s between MVR and MPC are very subtle. Compared with the MVR, the MPC u_s merely reaches and leaves its limit (+1 here) at very slightly different time instants and also produces some "optimized ripples" of less than 0.25% around that limit instead of a "flat" value as adopted in the clipped linear control law, but by doing these little things the MPC manages to bring J_s down by almost one order of magnitude. This demonstrates how nontrivial the optimal u_s can be. We can also see from the plots that only one constraint is active in the "slightly saturated" situation whereas multiple constraints are active in the "deeply saturated" saturation. On the other hand, the bottom two plots in Figure 4 illustrates our discussion in Remark 10. The plots clearly show that during certain moments of the transient stages ($t > 1.0$ s and $t > 1.5$ s), the transient control u_t is "disabled" due to the saturation of the steady-state control u_s. Note that we are labeling the dashed blue curve as "u_s used to compute u_t" since it is slightly different from the actual u_s. For instance, $u_t(k+1)$ is computed from the knowledge of u_s at time k, which is not exactly the same as $u_s(k+1)$. Obviously, u_t is not just disabled whenever u_s saturates. It happens only when the desired direction of u_t violates the active constraint.

Next, let us look at the performance of our second MPC approach in Section 4 based on dynamic policy. Odd harmonics up to the 9th order are included in $A_{\hat{v}}$ resulting in a total of $n_{\hat{v}} = 20$ variables. See Table 3. Since the number of variables is much lower than that of the first approach, we assume $N_a = 1$ here, that is, *one iteration* of Algorithm 4 (equivalent version) is carried out in each sampling interval, whereas the transient subproblem is solved *completely* within each sampling interval. The transient performance of J_s is plotted in Figure 5. Note that the MVR curve exhibits a slightly different transient from Figure 3 since their N_a values are different. The dynamic policy approach clearly shows a faster transient response than the receding horizon quadratic programming approach, not only because of a smaller N_a but also a smaller-sized quadratic program overall. However, the drawback is a slightly suboptimal J_s, as indicated in Table 3.

As mentioned in Section 4, it is possible to over-design A_v, and hence $A_{\hat{v}}$, so that the optimal J_s in this second MPC method will approach the first MPC method. For example, although we only have odd harmonics up to the 9th order in w, we may include odd harmonics up to the 29th order in A_v and $A_{\hat{v}}$. The results are also recorded in Table 3, and we see that this J_s value is very close to the optimal one in the first method.

7. Conclusions

To apply MPC to fast-sampling systems with input constraints for the tracking of periodic references, efficient algorithms to reduce online computational burdens are necessary. We have decomposed the tracking problem into a computationally complex steady-state subproblem and a computationally simple transient subproblem, and then proposed two approaches to solve the former. The first approach,

based on the concept of receding horizon quadratic programming, spreads the optimization over several sampling intervals, thus reducing the computational burdens at the price of a slower transient response. The second approach, based on the idea of a dynamic policy on the control input, further reduces the online computations at the price of a slightly suboptimal asymptotic performance. Despite the limitations, these approaches make the application of MPC to fast-sampling systems possible. Their transient behaviours and steady-state optimality have been analyzed via computer simulations, which have also demonstrated that the steady-state subproblem and the transient subproblem can be solved in parallel with different sampling rates. When the methods proposed in this paper are combined with modern code optimizations, the applicability of MPC to the servomechanism of fast-sampling constrained systems will be greatly enhanced.

Acknowledgment

This work is supported by HKU CRCG 201008159001.

References

[1] J. M. Maciejowski, *Predictive Control with Constraints*, Prentice-Hall, New Jersey, NY, USA, 2002.

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

[3] P. Whittle, *Optimization Over Time*, Wiley, New York, NY, USA, 1982.

[4] W. H. Kwon and S. Han, *Receding Horizon Control*, Springer, New York, NY, USA, 2005.

[5] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[6] A. Bemporad, F. Borrelli, and M. Morari, "Min-max control of constrained uncertain discrete-time linear systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 9, pp. 1600–1606, 2003.

[7] A. Bemporad and C. Filippi, "Suboptimal explicit receding horizon control via approximate multiparametric quadratic programming," *Journal of Optimization Theory and Applications*, vol. 117, no. 1, pp. 9–38, 2003.

[8] Z. Wan and M. V. Kothare, "Robust output feedback model predictive control using off-line linear matrix inequalities," *Journal of Process Control*, vol. 12, no. 7, pp. 763–774, 2002.

[9] Z. Wan and M. V. Kothare, "An efficient off-line formulation of robust model predictive control using linear matrix inequalities," *Automatica*, vol. 39, no. 5, pp. 837–846, 2003.

[10] M. Åkerblad and A. Hansson, "Efficient solution of second order cone program for model predictive control," *International Journal of Control*, vol. 77, no. 1, pp. 55–77, 2004.

[11] C. V. Rao, S. J. Wright, and J. B. Rawlings, "Application of interior-point methods to model predictive control," *Journal of Optimization Theory and Applications*, vol. 99, no. 3, pp. 723–757, 1998.

[12] J. Mattingley, Y. Wang, and S. Boyd, "Receding horizon control: automatic generation of high-speed solvers," *IEEE Control Systems Magazine*, vol. 31, no. 3, pp. 52–65, 2011.

[13] M. Cannon and B. Kouvaritakis, "Optimizing prediction dynamics for robust MPC," *IEEE Transactions on Automatic Control*, vol. 50, no. 11, pp. 1892–1897, 2005.

[14] A. Richards, K.-V. Ling, and J. Maciejowski, "Robust multiplexed model predictive control," in *Proceedings of the European Control Conference*, Kos, Greece, July 2007.

[15] K. V. Ling, W. K. Ho, Y. Feng, and B. Wu, "Integral-square-error performance of multiplexed model predictive control," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 2, pp. 196–203, 2011.

[16] D. Li and Y. Xi, "Aggregation based robust model predictive controller for systems with bounded disturbance," in *Proceedings of the 7th World Congress on Intelligent Control and Automation, (WCICA '08)*, pp. 3374–3379, Chongqing, China, June 2008.

[17] D. Li and Y. Xi, "Aggregation based closed-loop MPC with guaranteed performance," in *Proceedings of the 48th IEEE Conference on Decision and Control, (CDC/CCC '09)*, pp. 7400–7405, Shanghai, China, December 2009.

[18] J. B. Rawlings and K. R. Muske, "The stability of constrained receding horizon control," *IEEE Transactions on Automatic Control*, vol. 38, no. 10, pp. 1512–1516, 1993.

[19] K. H. Kwan, Y. C. Chu, and P. L. So, "Model-based H∞ control of a unified power quality conditioner," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 7, pp. 2493–2504, 2009.

[20] S. G. Nash and A. Sofer, *Linear and Nonlinear Programming*, McGraw-Hill, New York, NY, USA, 1996.

[21] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, NY, USA, 2nd edition, 2006.

[22] A. Gautam, Y.-C. Chu, and Y. C. Soh, "Robust MPC with optimized controller dynamics for linear polytopic systems with bounded additive disturbances," in *Proceedings of the 7th Asian Control Conference, (ASCC '09)*, pp. 1322–1327, Hong Kong, China, August 2009.

[23] A. Gautam, Y.-C. Chu, and Y. C. Soh, "Optimized dynamic policy for receding horizon control of linear time-varying systems with bounded disturbances," *IEEE Transactions on Automatic Control*. In press.

[24] E. J. Davison, "The robust control of a servomechanism problem for linear time-invariant multivariable systems," *IEEE Transactions on Automatic Control*, vol. 21, no. 1, pp. 25–34, 1976.

[25] B. A. Francis and W. M. Wonham, "The internal model principle of control theory," *Automatica*, vol. 12, no. 5, pp. 457–465, 1976.

[26] B. A. Francis, "The linear multivariable regulator problem," *SIAM Journal on Control Optimization*, vol. 15, no. 3, pp. 486–505, 1977.

Hindawi Publishing Corporation
Journal of Control Science and Engineering
Volume 2012, Article ID 630645, 6 pages
doi:10.1155/2012/630645

Research Article

Pole Placement-Based NMPC of Hammerstein Systems and Its Application to Grade Transition Control of Polypropylene

He De-Feng and Yu Li

College of Information Engineering, Zhejiang University of Technology, Zhejiang, Hangzhou 310023, China

Correspondence should be addressed to He De-Feng, hdfzj@zjut.edu.cn

Received 1 July 2011; Accepted 11 September 2011

Academic Editor: Baocang Ding

This paper presents a new nonlinear model predictive control (MPC) algorithm for Hammerstein systems subject to constraints on the state, input, and intermediate variable. Taking no account of constraints, a desired linear controller of the intermediate variable is obtained by applying pole placement to the linear subsystem. Then, actual control actions are determined in consideration of constraints by online solving a finite horizon optimal control problem, where only the first control is calculated and others are approximated to reduce the computational demand. Moreover, the asymptotic stability can be guaranteed in certain condition. Finally, the simulation example of the grade transition control of industrial polypropylene plants is used to demonstrate the effectiveness of the results proposed here.

1. Introduction

Hammerstein systems consist of the cascade connection of a static (memoryless) nonlinear function followed from a linear dynamic system. Under certain assumptions (such as fading memory), the Hammerstein model could approximately represent nonlinear dynamics of real systems and has been successfully applied to many kinds of industrial processes such as pH neutralization [1], distillation [2], and polymerization transition [3]. In recent years, constrained control of Hammerstein systems has become one of the most needed and yet very difficult tasks for the process industry.

Model predictive control (MPC) is an effective control algorithm for handling constrained control problems, and various MPC algorithms have been proposed for control of Hammerstein systems with constraints [4–11]. Making use of the geometry structure of Hammerstein model, [4–9] developed two-step MPC schemes for Hammerstein systems with input constraints, where the intermediate variable was firstly obtained by linear MPC and then the actual control was calculated by solving nonlinear algebraic equations, desaturation, and so forth. Although two-step MPC algorithms possess more light computational burden than that where the nonlinearities were incorporated into optimal control problems directly [10, 11], solving nonlinear algebraic equations will inevitably have error and the restricted control is generally different from the desired one for constrained Hammerstein systems [6]. Therefore, the performance and stability properties may be deteriorated in the presence of constraints. Moreover, to the best of authors' knowledge, the conversional two-step MPC schemes usually have limited ability to dealing with input constraints. So, [12] presented a new two-step MPC scheme for Hammerstein systems with overall constraints, where the actual controller was determined by solving a finite horizon optimization problem that was described as tracking quadratic, optimal trajectories of the intermediate variable. However, the online computational demand of optimization problem of the scheme is still an open question.

In this paper, we consider the problem of the grade transition control of industrial polypropylene plants and propose an efficient NMPC scheme for Hammerstein systems with overall constraints based on the pole placement method [13]. We firstly obtained desired trajectories of Hammerstein systems by using pole placement to the linear subsystem and then calculated actual control actions by solution of finite

horizon optimal control problems. In order to reduce the online computational burden of the optimization problem, only the first control action is calculated and the remaining is approximated by solving nonlinear algebraic equations over predictive horizon. Thus, the performance and stability properties can be guaranteed in the face of overall constraints, with moderate increment of the computational demand. Finally, an example of the grade transition control of propylene plants is used to demonstrate the effectiveness of the proposed algorithm.

2. Pole Placement-Based NMPC Algorithm

Consider the discrete-time Hammerstein systems described by

$$x(t+1) = Ax(t) + Bv(t), \quad v(t) = g(u(t), t), \ t = 0, 1, \ldots, \tag{1}$$

where $x \in R^n$, $v \in R^m$, and $u \in R^r$ are the state, intermediate variable and control input, respectively. Vector field g represents the nonlinear relationship between the input and intermediate variable, satisfying $g(0, t) = 0$. It is assumed that (A, B) is stabilizable and the state is available for feedback control. Without loss of generality, we also assume that the origin is the equilibrium of system (1).

Given constraints of system (1) as follows:

$$\begin{aligned} x(t) &\in M_x = \left\{x \in R^n : x^{\mathrm{LB}} \le x \le x^{\mathrm{UB}}\right\}, \quad t = 0, 1, \ldots, \\ v(t) &\in M_v = \left\{v \in R^m : v^{\mathrm{LB}} \le v \le v^{\mathrm{UB}}\right\}, \quad t = 0, 1, \ldots, \\ u(t) &\in M_u = \left\{u \in R^r : u^{\mathrm{LB}} \le u \le u^{\mathrm{UB}}\right\}, \quad t = 0, 1, \ldots, \end{aligned} \tag{2}$$

where sets $M_x \subseteq R^n$, $M_v \subseteq R^m$, and $M_u \subseteq R^r$ are constraints on the state, intermediate variable, and input, respectively. The goal of this paper is to design an efficient NMPC algorithm for the Hammerstein system (1) with the constraint (2).

2.1. Design of NMPC Algorithm. Consider n desired poles, denoted by $\{\lambda_1, \ldots, \lambda_n\}$, of the linear subsystem of (1), which represent some desired performance of system (1). Associated with this set of desired poles is a linear controller of the intermediate variable

$$v(t) = -Kx(t) \tag{3}$$

which is generated via Lyapunov's method without any constraints [13]. Since the controller (3) is unable to be implemented by real control action, here it is labeled as $v(t)^d = -Kx(t)^d$ with associated closed-loop states $x(t)^d$.

Define a finite horizon objective function $J(t)$ as follows:

$$\begin{aligned} J(t) = \sum_{i=0}^{Tp-1} \Big\{ &\left[x(t+i+1 \mid t) - x(t+i+1)^d\right]^{\mathrm{T}} \\ &\times Q \left[x(t+i+1 \mid t) - x(t+i+1)^d\right] \\ &+ u(t+i \mid t)^{\mathrm{T}} R u(t+i \mid t) \Big\}, \end{aligned} \tag{4}$$

where the vector $\bullet(t+i \mid t)$ is i-step ahead prediction from time instant t, integer $0 < T_p < \infty$ is the predictive horizon, and matrices $0 \le Q$ and $0 < R$ are the weighting matrices of the state and input, respectively. Then a new two-step MPC algorithm of Hammerstein systems with overall constraints is presented below.

Algorithm 1 (modified two-step MPC, MTMPC). This algorithm comprises the following steps:

Step 1. Given n desired poles $\{\lambda_1, \ldots, \lambda_n\}$ and off-line compute the gain matrix K in (3) such that the eigenvalues of $A - BK$ are those specified in vector $\{\lambda_1, \ldots, \lambda_n\}$.

Step 2. Set T_p, Q, R in the objective function in (4).

Step 3. With $v(\cdot)^d = -Kx(\cdot)^d$ and the current state $x(t)$, solve on-line the nonlinear algebra equations (5) without constraints to obtain T_p desired control actions $u(t+i)^d$

$$\begin{aligned} &v(t+i)^d - g\left(u(t+i)^d, t+i\right) = 0, \\ &x(t+i+1)^d = (A - BK)x(t+i)^d, \\ &x(t)^d = x(t), \quad i = 0, 1, \ldots, T_p - 1. \end{aligned} \tag{5}$$

Step 4. With the current state $x(t)$, determine on-line the actual control action $u(t)$ as follows:

$$\begin{aligned} &u(t) = \arg\min_{u(t|t)} J(t), \\ &\text{s.t.} \quad x(t+i+1 \mid t) = Ax(t+i \mid t) + Bg(u(t+i \mid t), t+i), \\ &x(t+i+1)^d = (A - BK)x(t+i)^d, \\ &u(t+i \mid t) = u(t+i)^d, \quad i = 1, \ldots, T_p - 1, \\ &x(t+1 \mid t) \in M_x, \quad g(u(t \mid t), t) \in M_v, \quad u(t \mid t) \in M_u, \\ &x(t \mid t) = x(t)^d = x(t), \end{aligned} \tag{6}$$

where $J(t)$ is defined by (4).

Step 5. Implement $u(t)$, set $t = t + 1$, and go back Step 3.

Remark 2. For the algorithm here, only the first control action (i.e., $u(t \mid t)$) is optimized subject to constraints at each time. So, the online optimization problem has only r decision variables versus $T_p r$ decision variables for general

nonlinear MPC, which results in a significant reduction of the online computational demand since the computational demand, in general, grows exponentially with the number of decision variables [14]. Thus, the predictive horizon T_p should be chosen reasonably large to ensure adequate performance. On the other hand, since only the first control is implemented in MPC scheme, it is expected that only enforcing constraints for the first prediction should give excellent results. In addition, the closed-loop performance of MPC systems clearly depends on how n desired poles $\{\lambda_1,\ldots,\lambda_n\}$ are arranged. The further work will focus on how to determine $\{\lambda_1,\ldots,\lambda_n\}$ optimally.

It should be pointed out that the feasibility of optimization problem (6) may not be guaranteed theoretically with respect to arbitrary constraints in (2). However, it is usually ensured in practice via adjusting the intermediate variable constraint. Next, we show that the closed-loop MPC system (1) is asymptotically stable, provided that the feasibility of (6) is hold.

2.2. Analysis of Stability. In terms of Algorithm MTMPC, the MPC law of constrained Hammerstein system (1), (2) is defined as

$$u(t)^{\mathrm{MPC}} = u(t \mid t), \quad t = 0, 1, \ldots \tag{7}$$

with the closed-loop system

$$\begin{aligned} x(t+1) &= Ax(t) + Bg\left(u(t)^{\mathrm{MPC}}, t\right) \\ &= (A - BK)x(t) + B\left[Kx(t) + g\left(u(t)^{\mathrm{MPC}}, t\right)\right]. \end{aligned} \tag{8}$$

Definition 3. Set $M \subseteq R^n$ is called an attractive region of system (8) if for all $x(t) \in M$, the system trajectories $x(s; x(t)) \to 0$ when $s \to +\infty$ and satisfies

$$x(t) \in M \Longrightarrow x(t+1) \in M, \quad \forall u(t)^{\mathrm{MPC}} \in M_{uv},\ t = 0, 1, \ldots, \tag{9}$$

where subset $M_{uv} = \{u \in R^r : g(u, \cdot) \in M_v\} \subseteq M_u$.

Let a set of points $\{\lambda_1,\ldots,\lambda_n\}$ be the desired poles of the linear subsystem $x(t+1) = (A - BK)x(t)$. Then from inverse Lyapunov's theorem [13], there is systematic matrices $P > 0$ and $Q > 0$ such that

$$(A - BK)^{\mathrm{T}} P (A - BK) - P < -Q. \tag{10}$$

In following, we present the result on our proposed algorithm.

Theorem 4. *Consider the system* (1), (2), *there exists a nonempty region M such that the closed-loop system* (8) *is asymptotically stable if the inequality* (11) *is satisfied*

$$V(x(t+1)) \le V\left(x(t+1)^d\right), \quad \forall x(t) \in M,\ t = 0, 1, \ldots, \tag{11}$$

where function $V(x) = x^{\mathrm{T}}Px$ *and* $x(t+1)^d = (A - BK)x(t)^d$ *with* $x(t)^d = x(t)$. *Moreover, the set M is an attractive region of system* (8).

Proof. Consider the system (1), (2), define a level set M of function $V(x)$ as

$$M = \{x \in R^n : V(x) \le c,\ c > 0\} \subseteq M_x \tag{12}$$

such that $u(t)^{\mathrm{MPC}} \in M_{uv}$, for all $x(t) \in M$, $t = 0, 1, \ldots$. Note that the set M is always nonempty since the constraint in (2) includes the origin as the internal point.

Let function $V(x)$ be a Lyapunov function candidate of system (8). Then, for any $x(t) \in M$, we have

$$\begin{aligned} &V(x(t+1)) - V(x(t)) \\ &\quad = x(t+1)^{\mathrm{T}} P x(t+1) - x(t)^{\mathrm{T}} P x(t) \\ &\quad = x(t)^{\mathrm{T}} (A - BK)^{\mathrm{T}} P (A - BK) x(t)^{\mathrm{T}} \\ &\qquad - x(t)^{\mathrm{T}} P x(t) + 2x(t)^{\mathrm{T}} (A - BK)^{\mathrm{T}} PB\Delta + \Delta^{\mathrm{T}} B^{\mathrm{T}} PB\Delta \\ &\quad = x(t)^{\mathrm{T}} \Big[(A - BK)^{\mathrm{T}} P (A - BK) - P\Big] x(t) \\ &\qquad + 2x(t)^{\mathrm{T}} (A - BK)^{\mathrm{T}} PB\Delta + \Delta^{\mathrm{T}} B^{\mathrm{T}} PB\Delta, \end{aligned} \tag{13}$$

where $\Delta = g(u(t)^{\mathrm{MPC}}, t) + Kx(t) = g(u(t)^{\mathrm{MPC}}, t) - v(t)^d$. Substituting (10) into (13) yields

$$\begin{aligned} &V(x(t+1)) - V(x(t)) \\ &\quad = -x(t)^{\mathrm{T}} Q x(t) + 2[(A - BK)x(t)]^{\mathrm{T}} PB\Delta + \Delta^{\mathrm{T}} B^{\mathrm{T}} PB\Delta \\ &\quad = -x(t)^{\mathrm{T}} Q x(t) + 2x(t)^{\mathrm{T}} A^{\mathrm{T}} PB\Delta \\ &\qquad + 2(g - \Delta)^{\mathrm{T}} B^{\mathrm{T}} PB\Delta + \Delta^{\mathrm{T}} B^{\mathrm{T}} PB\Delta \\ &\quad = -x(t)^{\mathrm{T}} Q x(t) + 2x(t+1)^{\mathrm{T}} PB\Delta - \Delta^{\mathrm{T}} B^{\mathrm{T}} PB\Delta \\ &\quad = -x(t)^{\mathrm{T}} Q x(t) - x(t+1)^{d\mathrm{T}} P x(t+1)^d \\ &\qquad + x(t+1)^{\mathrm{T}} P x(t+1) \\ &\quad = -x(t)^{\mathrm{T}} Q x(t) - V\left(x(t+1)^d\right) + V(x(t+1)). \end{aligned} \tag{14}$$

From the inequality (11), it is straight forward to obtain that

$$V(x(t+1)) - V(x(t)) \le -x(t)^{\mathrm{T}} Q x(t), \quad \forall x(t) \in M. \tag{15}$$

Hence, function $V(x)$ is a Lyapunov function of system (8), and the system is asymptotically stable in region M. Furthermore, from the definition of attractive regions and set M in (12), we derive that M is an attractive region of system (8).

This establishes the theorem. □

Remark 5. From the proof of the above theorem, we know that the stability property of closed-loop system (8) is guaranteed by the linear controller of the intermediate variable, provided that the feasibility of optimization problem (6) and condition (11) holds. This suggests that the stability and optimality can be separated to some extent. Therefore, the design parameter of the objective function in (4) can be tuned freely to achieve more satisfactory performance, regardless the stability of the closed-loop system. In addition, it is pointed out that the attractive region M obtained here is merely a theoretical concept and the computation of M is not an easy task.

Remark 6. Instead of conventional two-step MPC algorithms, where the actual control was determined by solving nonlinear algebraic equations based on desired controllers of the intermediate variable resulted from linear MPC, actual control actions here are derived by solving a finite horizon optimal control problem (i.e., nonlinear MPC), which is defined as tracking the desired state trajectories driven by pole placement state feedback controller. Thus, overall constraints of Hammerstein systems are taken into account when designing the predictive controller, and then performance and stability properties are guaranteed. Meanwhile, the ability to handling constraints is enhanced in the algorithm proposed here. Note that the linear controller of intermediate variables in (3) can be any one that stabilizes the linear subsystems of Hammerstein models with certain desired performance.

3. Simulation Example

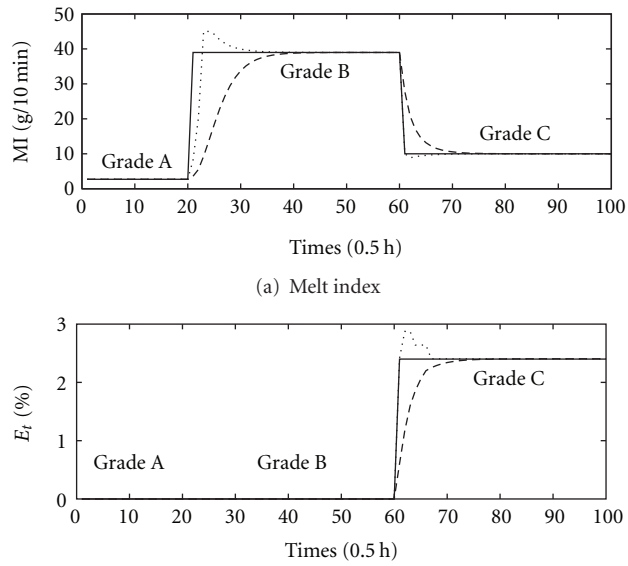
Taking an example of polypropylene (PP) grade transition control, we illustrate the effectiveness of the proposed algorithm.

In propylene polymerization industry, melt index (MI, g/10 min) is usually used to identify grade indices of PP homopolymer and both of MI and concentration of ethylene (E_t, %) in PP denote grade indices of total phase PP [12, 15–18]. This implies that the nature of grade transition process is to operate the response of the variables MI and E_t in polymer. Then according to the propylene polymerization mechanism, we have the following mathematical model which formulates the dynamics of the grade transition in polypropylene process [12]:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-\frac{1}{\tau} & 0\\ 0 & -\frac{1}{\tau}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}\frac{1}{\tau} & 0\\ 0 & \frac{1}{\tau}\end{bmatrix}\begin{bmatrix}g_1(u,t)\\ g_2(u,t)\end{bmatrix},$$

$$\begin{bmatrix}g_1(u,t)\\ g_2(u,t)\end{bmatrix} = \begin{bmatrix}k_1 + \dfrac{k_2}{u_1} + k_3\log(k_4 + k_5u_2 + k_6u_3)\\ \dfrac{k_7u_2 + 1}{k_8/u_3 + k_9u_3 + k_{10}}\end{bmatrix}, \quad (16)$$

$$y(t) = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}x(t), \quad t \geq 0,$$



(a) Melt index

(b) Concentration of ethylene in polypropylene

Figure 1: MI and E_t output trajectories of grade transition processes. (Solid line is the set-point, dotted line the instantaneous, and dashed line the cumulative).

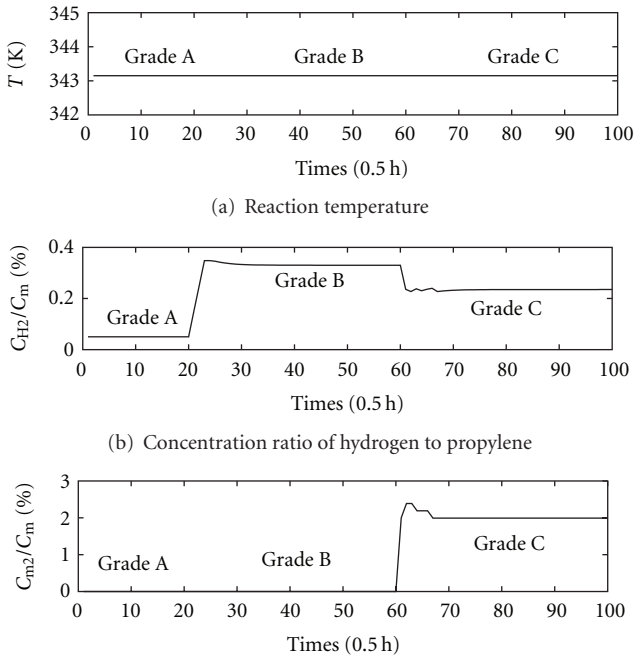
where states x_1 and x_2 are the cumulative MI (MI_c, g/10 min) and E_t (E_{tc}, %) of PP, respectively; intermediate variables g_1 and g_2 denote the instantaneous MI (MI_i, g/10 min) and E_t (E_{ti}, %) of PP, respectively; input, u_1, u_2, u_3 denote the reaction temperature T(K), concentration ratio of hydrogen to propylene C_{H2}/C_m (%), and concentration ratio of ethylene to propylene C_{m2}/C_m (%), respectively, and parameter τ is the average polymer residence time ($\tau = 2$ h). The parameters of model k_i ($i = 1, \ldots, 10$) can be identified on-line by industrial data [15].

The grade transition sequence considered here is A → B → C, where grades A and B are PP homopolymer and C is total phase PP. It should be pointed out that the feed flow rate of ethylene is always zero (i.e., $[C_{m2}/C_m] = 0$) in production of PP homopolymer grades A and B. The grade transition process is defined by the set-point of states and input/output constraints as shown in Table 1, where superscripts c, i, and Δ denote accumulation, instantaneous value, and increment constraints, respectively. Each grade transition is optimized over 10 h horizon with a sampling interval of 0.5 h.

In the simulation running, the closed-loop desired poles of system (16) are placed at $\{\lambda_1, \lambda_2\} = \{0.72, 0.60\}$, and the corresponding state-feedback matrix K in (3) is computed by $K = [0.12, 0; 0, 0.6]$. Moreover, the parameters in the objective function (4) are chosen as penalty on the states, $Q = qI_2$, with $q = 1$, penalty on the inputs, $R = I_3$ and a horizon length of $T_p = 20$. Assume that grade transition A → B takes place at the tenth hour and B → C at the thirtieth hour. Then some curves of the grade transition process being controlled by NMPC law are shown in Figures 1 and 2, where the solid line in Figure 1 is the set-point, the dotted line the instantaneous, and the dashed line the cumulative.

Table 1: Grade indices and transition constraints.

Grade	MI (g/10 min)	E_t (%)	T (K)	C_{H2} (%)	C_{m2}/C_m (%)
A	2.7	—	343.15	0.050	—
B	39.0	—	343.15	0.330	—
C	10.0	2.4	343.15	0.236	1.99
A → B	$[2.4, 42.0]^c$	—	$[341.15, 345.15]$	$[0.02, 0.35]$	—
	$[2.0, 45.0]^i$	—	—	$[-0.1, 0.10]^\Delta$	—
B → C	$[8.0, 42.0]^c$	$[0.0, 2.5]^c$	$[341.15, 345.15]$	$[0.20, 0.35]$	$[0.00, 2.50]$
	$[6.0, 45.0]^i$	$[0.0, 2.6]^i$	—	$[-0.1, 0.10]^\Delta$	$[-0.1, 0.10]^\Delta$



(a) Reaction temperature

(b) Concentration ratio of hydrogen to propylene

(c) Concentration ratio of ethylene to propylene

Figure 2: Input profiles of grade transition processes.

In order to calculate the quantity of off-specification polymer in grade transition process, we employ the criterion where on-specification polymer is defined as the polymer with MI_c and E_{tc} within a ±5% deviation of desired specification in Table 1. All other polymers beyond the criterion are categorized as off-specification production. Similarly, the transition time is defined as the time interval when the cumulative properties of polymer are out of the specification range ±5% of the previous grade and lasting until the cumulative properties enter and stay within the specification range ±5% of the new target grade. Therefore, the time of grade A → B transition is 8.5 hours and the time of grade B → C transition is 6.5 hours. Compare to the time of 10 hours in actual grade transition operation, this decreases the quantity of off-specification polymer in grade transition process and then increases the productive time of on-specification polymer. Also, Figure 2(a) suggests that there is no change in reaction temperature during the grade transition processes, which is desired in actual operation. Finally, it can be seen that the control laws, intermediate variables, and the states in the grade transition operation do not violate the constraints in Table 1.

4. Conclusion

Together with the pole placement method, the paper presented an efficient MPC algorithm for control of Hammerstein systems subject to constraints on states, inputs, and intermediate variables. Applying pole placement to the linear subsystem of Hammerstein models yielded linear controllers of the intermediate variable and then the actual control action was determined by online solving an optimization problem. The size of the online optimization problem depended only on the number of inputs, not on the predictive horizon, which greatly reduced the computational demand of the algorithm. With condition of feasibility of the optimization problem, the asymptotical stability of the closed-loop system with constraints was guaranteed and the results on simulation example of grade transition control of polypropylene plants showed the effectiveness of the results obtained here.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant no. 60904040, Specialized Research Fund for Doctoral Program of Higher Education of China under Grant no. 20093317120002 and Natural Science Foundation of Zhejiang Province under Grant no. Y1100911. Finally, the authors are grateful to the reviews' constructive comments. This paper includes revised and extended text of the first author's presentation at the 29th CCC, July 27, 2010, Beijiang, China.

References

[1] Z. Hasiewicz, "Hammerstein system identification by the Haar multiresolution approximation," *International Journal of Adaptive Control and Signal Processing*, vol. 49, pp. 691–717, 1999.

[2] N. Bhandari and D. Rollins, "Continuous-time Hammerstein nonlinear modeling applied to distillation," *AIChE Journal*, vol. 50, no. 2, pp. 530–533, 2004.

[3] D. F. He, L. Yu, and M. S. Xue, "On operation guidance for grade transition in propylene polymerization plants," in *Proceedings of the 19th Chinese Process Control Conference*, pp. 339–342, Beijiang, China, 2008, (Chinese).

[4] Q. M. Zhu, K. Warwick, and J. L. Douce, "Adaptive general predictive controller for nonlinear systems," *IEE Proceedings D*, vol. 138, no. 1, pp. 33–40, 1991.

[5] K. P. Fruzzetti, A. Palazoğlu, and K. A. McDonald, "Nonlinear model predictive control using Hammerstein models," *Journal of Process Control*, vol. 7, no. 1, pp. 31–41, 1997.

[6] B. C. Ding, Y. G. Xi, and S. Y. Li, "Stability analysis on predictive control of discrete-time systems with input nonlinearity," *Acta Automatica Sinica*, vol. 29, no. 6, pp. 827–834, 2003.

[7] B. C. Ding, Y. G. Xi, and S. Y. Li, "On the stability of output feedback predictive control for systems with input nonlinearity," *Asian Journal of Control*, vol. 6, no. 3, pp. 388–397, 2004.

[8] B. Ding and Y. Xi, "A two-step predictive control design for input saturated Hammerstein systems," *International Journal of Robust and Nonlinear Control*, vol. 16, no. 7, pp. 353–367, 2006.

[9] H. T. Zhang, H. X. Li, and G. Chen, "Dual-mode predictive control algorithm for constrained Hammerstein systems," *International Journal of Control*, vol. 81, no. 10, pp. 1609–1625, 2008.

[10] H. H. J. Bloemen and T. J. J. van den Boom, "Model-based predictive control for Hammerstein systems," in *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 5, pp. 4963–4968, Sydney, Australia, 2000.

[11] H. H. J. Bloemen, T. J. J. van den Boom, and H. B. Verbruggen, "Model-based predictive control for Hammerstein-Wiener systems," *International Journal of Control*, vol. 74, pp. 482–495, 2005.

[12] D. F. He and L. Yu, "Nonlinear predictive control of constrained Hammerstein systems and its research on simulation of polypropylene grade transition," *Acta Automatica Sinica*, vol. 35, no. 12, pp. 1558–1563, 2009 (Chinese).

[13] C. T. Chen, *Linear system Theory and Design*, Oxford University Press, New York, NY, USA, 3rd edition, 1999.

[14] A. Zheng, "A computationally efficient nonlinear MPC algorithm," in *Proceedings of the American Control Conference*, pp. 1623–1627, Albuquerque, NM, USA, June 1997.

[15] J. R. Richards and J. P. Congalidis, "Measurement and control of polymerization reactors," *Computers and Chemical Engineering*, vol. 30, no. 10–12, pp. 1447–1463, 2006.

[16] H. S. Yi, J. H. Kim, C. Han, J. Lee, and S. S. Na, "Plantwide optimal grade transition for an industrial high-density polyethylene plant," *Industrial and Engineering Chemistry Research*, vol. 42, no. 1, pp. 91–98, 2003.

[17] Y. Wang, H. Seki, S. Ohyama, K. Akamatsu, M. Ogawa, and M. Ohshima, "Optimal grade transition control for polymerization reactors," *Computers and Chemical Engineering*, vol. 24, no. 2–7, pp. 1555–1561, 2000.

[18] T. Kreft and W. F. Reed, "Predictive control of average composition and molecular weight distributions in semibatch free radical copolymerization reactions," *Macromolecules*, vol. 42, no. 15, pp. 5558–5565, 2009.