LNCIS
Lecture Notes in Control and Information Sciences
384

Lalo Magni
Davide Martino Raimondo
Frank Allgöwer (Eds.)

Nonlinear Model Predictive Control

Towards New Challenging Applications

Springer

Lecture Notes
in Control and Information Sciences 384

Editors: M. Thoma, F. Allgöwer, M. Morari

Lalo Magni, Davide Martino Raimondo,
Frank Allgöwer (Eds.)

Nonlinear Model Predictive Control

Towards New Challenging Applications

Springer

Editors

Prof. Lalo Magni

Dipartimento di Informatica e Sistemistica
Universita' di Pavia
via Ferrata 1, 27100 Pavia
Italy
E-mail: Lalo.Magni@unipv.it

Prof. Dr.-Ing. Frank Allgöwer

Institute for Systems Theory and
Automatic Control
University of Stuttgart
Pfaffenwaldring 9
70550 Stuttgart
Germany
E-mail: allgower@ist.uni-stuttgart.de

Dr. Davide M. Raimondo

ETH Zurich
Automatic Control Laboratory
ETL I 24.2, Physikstrasse 3
CH-8092 Zürich
Switzerland
E-mail: davide.raimondo@control.ee.ethz.ch

ISBN 978-3-642-01093-4 e-ISBN 978-3-642-01094-1

DOI 10.1007/978-3-642-01094-1

Lecture Notes in Control and Information Sciences ISSN 0170-8643

Library of Congress Control Number: Applied for

Typeset & Cover Design: Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed in acid-free paper

5 4 3 2 1 0

springer.com

Preface

Model Predictive Control (MPC) is an area in rapid development with respect to both theoretical and application aspects. The former petrochemical applications of MPC were ‘easy’, in the sense that they involved only a small number of rather similar problems, most of which required only control near steady-state conditions. Further control performance specifications were not very challenging. The improving of technology and control theory enabled the application of MPC in new problems often requiring Nonlinear MPC because of the large transients involved, as it has been already seen even in the chemical process industry for the control of product grade changes. There is now a great interest in introducing MPC in many process and non-process applications such as paper-making, control of many kinds of vehicles, including marine, air, space, road and off-road. Some interesting biomedical applications are also very promising. Finally, the interest in the control of complex systems and networks is significantly increasing.

The new applications frequently involve tight performance specifications, model changes or adaptations because of changing operating points, and, perhaps more significantly, safety-criticality. MPC formulations which offer guarantees of stability and robustness feasibility are expected to be of great importance for the deployment of MPC in these applications. The significant effort in developing efficient solutions of the optimisation problem both using an explicit and a numerical approach is of paramount importance for a wider diffusion of NMPC.

In order to summarize these recent developments, and to consider these new challenges, on September 5-9, 2008, we organized an international workshop entitled “International Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control" (NMPC08) which was held in Pavia, Italy. In the spirit of the previous successful workshops held in Ascona, Switzerland, in 1998, and in Freudenstadt-Lauterbad, Germany in 2005, internationally recognized researchers from all over the world, working in the area of nonlinear model predictive control, were joined together. The number of participants has sensibly increased with respect to the previous editions and 21 countries from 4 continents were represented. The aim of this workshop was to lead to an open and critical exchange of ideas and to lay the foundation for new research directions and future international collaborations, facilitating the practical and theoretical advancement of NMPC technologies.

This volume contains a selection of papers presented at the workshop that cover the following topics: stability and robustness, control of complex systems, state estimation, tracking, control of stochastic systems, algorithms for explicit solution, algorithms for numerical solutions and applications. The high quality of the papers has been guaranteed by a double careful pear-review process.

We would like to thank all authors for their interesting contributions. Likewise, we are grateful to all of the involved reviewers for their invaluable comments.

The workshop and the present volume have been supported by University of Pavia, Risk and Security Study Center of the Institute for Advanced Study (IUSS) and Magneti Marelli.

Lalo Magni
Davide Martino Raimondo
Frank Allgöwer

Contents

Stochastic Systems

State Estimation

Tracking

Algorithms for Explicit Solution

Algorithms for Numerical Solution

Applications

Input-to-State Stability: A Unifying Framework for Robust Model Predictive Control

D. Limon, T. Alamo, D.M. Raimondo, D. Muñoz de la Peña,
J.M. Bravo, A. Ferramosca, and E.F. Camacho

Abstract. This paper deals with the robustness of Model Predictive Controllers for constrained uncertain nonlinear systems. The uncertainty is assumed to be modeled by a state and input dependent signal and a disturbance signal. The framework used for the analysis of the robust stability of the systems controlled by MPC is the well-known Input-to-State Stability. It is shown how this notion is suitable in spite of the presence of constraints on the system and of the possible discontinuity of the control law.

For the case of nominal MPC controllers, existing results on robust stability are extended to the ISS property, and some novel results are presented. Afterwards, stability property of robust MPC is analyzed. The proposed robust predictive controller uses a semi-feedback formulation and the notion of sequence of reachable sets (or tubes) for the robust constraint satisfaction. Under mild assumptions, input-to-state stability of the predictive controller based on nominal predicted cost is proved. Finally, stability of min-max predictive controllers is analyzed and sufficient conditions for the closed-loop system exhibits input-to-state practical stability property are stated. It is also shown how using a modified stage cost can lead to the ISS property. It is remarkable that ISS of predictive controllers is preserved in case of suboptimal solution of the minimization problem.

Keywords: Robust Nonlinear Model Predictive Control, Input-to-State Stability, Robust Constraint Satisfaction.

D. Limon, T. Alamo, D. Muñoz de la Peña, A. Ferramosca, and E.F. Camacho
Departamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla (Spain)
e-mail: {limon,alamo,davidmps,ferramosca,eduardo}@cartuja.us.es

D.M. Raimondo
Automatic Control Laboratory, ETH Zurich (Switzerland)
e-mail: davide.raimondo@control.ee.ethz.ch

J.M. Bravo
Departamento de Ingeniería Electrónica, Sistemas Informáticos y Automática,
Universidad de Huelva (Spain)
e-mail: caro@uhu.es

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 1–26.
springerlink.com

1 Introduction

Model predictive control (MPC) is one of the few control techniques capable to cope with constrained system providing an optimal control for a certain performance index. This control technique has been widely used in the process industry and studied in academia [51, 6, 34]. The theoretical development in issues such as stability, constraint satisfaction and robustness has recently matured. The main features of this problem are the following [42]: (i) stability must be ensured considering that the control law is a nonlinear (and maybe discontinuous [45]) function of the state ; (ii) recursive feasibility must be ensured to guarantee that the control law is well-posed, (iii) constraints must be robustly fulfilled along the evolution of the closed-loop system and (iv) performance and domain of attraction of the closed-loop system should be optimized.

For the nominal control problem, the Lyapunov theory combined with the invariant set theory provide a suitable theoretical framework to deal with the stability problem [42]. When model mismatches and/or disturbances exist, some different stability frameworks, such as robust stability, ultimately bounded evolution or asymptotic gain property, are used [8, 59, 7, 14, 20, 54, 56]. The problem to assure recursive feasibility and constraint satisfaction in presence of uncertainties is more involved, especially for the case of nonlinear prediction models [4, 21, 41, 54, 39].

This paper presents the notion of input-to-state stability (ISS) [60, 18, 19] as a suitable framework for the analysis of the stabilizing properties of model predictive controllers in presence of uncertainties. The use of ISS analysis in the context of nonlinear MPC is not new (see for instance [28, 30, 37, 53, 25, 26]), but this paper aims to show that it can be used as a general framework of robust stability analysis of constrained nonlinear discontinuous systems. Based on this notion, existing robust MPC techniques are studied: firstly, inherent robustness of the nominal MPC is analyzed and novel sufficient conditions for local ISS are presented, extending existing results [59, 29, 14, 46]. It is demonstrated that uniform continuity of the closed-loop model function or of the cost function are sufficient conditions to ensure robustness of the nominal MPC.

Then robust predictive controllers are studied. These controllers must ensure robust constraint satisfaction and recursive feasibility in spite of the uncertainties as well as a suit closed-loop performance. In order to enhance these properties, a semi-feedback parametrization and a tube (i.e. sequence of reachable sets) based approach [7, 28, 32, 5, 44, 55, 43] has been considered. Under suitable novel assumptions on the tube robust constraint satisfaction is proved.

In the case of cost functions based on nominal predictions [7, 47, 28, 32, 5, 44, 55, 43], ISS is proved subject to continuity of some ingredients. In the case of a worst-case cost function (i.e. min-max predictive controllers) [42, 41, 35, 10, 30, 37, 26, 49] the stability property is analyzed following [53] and shows how, under some standard assumptions, min-max predictive controllers are input to state practically stabilizing due to the worst-case based nature of the controller. Moreover, it

is shown, how ISS can be achieved by means of a dual mode approach or by using an $\mathscr{H}_\infty$ formulation.

A remarkable property derived from this analysis is that the ISS property of the closed-loop system is preserved in the case that the solution of the optimization problem does not provide the optimal solution, but the best suboptimal one.

Notation and basic definitions

Let $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{Z}$ and $\mathbb{Z}_{\geq 0}$ denote the real, the non-negative real, the integer and the non-negative integer numbers, respectively. Given two integers $a,b \in \mathbb{Z}_{\geq 0}$, $\mathbb{Z}_{[a,b]} \triangleq \{j \in \mathbb{Z}_{\geq 0} : a \leq j \leq b\}$. Given two vectors $x_1 \in \mathbb{R}^a$ and $x_2 \in \mathbb{R}^b$, $(x_1,x_2) \triangleq [x_1', x_2']' \in \mathbb{R}^{a+b}$. A norm of a vector $x \in \mathbb{R}^a$ is denoted by $|x|$. Given a signal $w \in \mathbb{R}^a$, the signal sequence is denoted by $\mathbf{w} \triangleq \{w(0), w(1), \cdots\}$ where the cardinality of the sequence is inferred from the context. $\mathbf{0}$ denotes a suitable signal sequence taking a null value. If a sequence depends on a parameter, as $\mathbf{w}(x)$, $w(j,x)$ denotes its j-th element. The sequence $\mathbf{w}_{[\tau]}$ denotes the truncation of sequence $\mathbf{w}$, i.e. $w_{[\tau]}(j) = w(j)$ if $0 \leq j \leq \tau$ and $w_{[\tau]}(j) = 0$ if $j > \tau$. For a given sequence, we denote $\|\mathbf{w}\| \triangleq \sup_{k\geq 0}\{|w(k)|\}$. The set of sequences $\mathbf{w}$, whose elements $w(j)$ belong to a set $W \subseteq \mathbb{R}^a$ is denoted by $\mathscr{M}_W$. For a compact set A, $A^{sup} \triangleq \sup_{a\in A}\{|a|\}$.

Consider a function $f(x,y) : A \times B \to \mathbb{R}^c$ with $A \subseteq \mathbb{R}^a$ and $B \subseteq \mathbb{R}^b$, then f is said to be uniformly continuous in x for all $x \in A$ and $y \in B$ if for all $\varepsilon > 0$, there exists a real number $\delta(\varepsilon) > 0$ such that $|f(x_1,y) - f(x_2,y)| \leq \varepsilon$ for all $x_1, x_2 \in A$ with $|x_1 - x_2| \leq \delta(\varepsilon)$ and for all $y \in B$. For a given set $\hat{A} \subseteq A$ and $y \in B$, the range of the function w.r.t. x is $f(\hat{A},y) \triangleq \{f(x,y) : x \in \hat{A}\} \subseteq \mathbb{R}^c$.

A function $\gamma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is of class $\mathscr{K}$ (or a "$\mathscr{K}$-function") if it is continuous, strictly increasing and $\gamma(0) = 0$. A function $\gamma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is of class $\mathscr{K}_\infty$ if it is a $\mathscr{K}$-function and $\gamma(s) \to +\infty$ as $s \to +\infty$. A function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{Z}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is of class $\mathscr{KL}$ if, for each fixed $t \geq 0$, $\beta(\cdot,t)$ is of class $\mathscr{K}$, for each fixed $s \geq 0$, $\beta(s,\cdot)$ is decreasing and $\beta(s,t) \to 0$ as $t \to \infty$. Consider a couple of $\mathscr{K}$-functions σ_1 and σ_2, then $\sigma_1 \circ \sigma_2(s) \triangleq \sigma_1(\sigma_2(s))$, besides $\sigma_1^j(s)$ denotes the j-th composition of σ_1, i.e. $\sigma_1^{j+1}(s) = \sigma_1 \circ \sigma_1^j(s)$ with $\sigma_1^1(s) \triangleq \sigma_1(s)$. A function $V : \mathbb{R}^a \to \mathbb{R}_{\geq 0}$ is called positive definite if $V(0) = 0$ and there exists a $\mathscr{K}$-function α such that $V(x) \geq \alpha(|x|)$.

2 Problem Statement

Consider that the plant to be controlled is modeled by a discrete-time invariant nonlinear difference equation as follows

$$x(k+1) = f(x(k),u(k),d(k),w(k)),\ k \geq 0 \tag{1}$$

where $x(k) \in \mathbb{R}^n$ is the system state, $u(k) \in \mathbb{R}^m$ is the current controlled variable, $d(k) \in \mathbb{R}^q$ is a signal which models external disturbances and $w(k) \in \mathbb{R}^p$ is a signal which models mismatches between the real plant and the model. The solution of

system (1) at sampling time k for the initial state $x(0)$, a sequence of control inputs $\mathbf{u}$, disturbances $\mathbf{w}$ and $\mathbf{d}$ is denoted as $\phi(k,x(0),\mathbf{u},\mathbf{d},\mathbf{w})$, where $\phi(0,x(0),\mathbf{u},\mathbf{d},\mathbf{w}) = x(0)$.

This system is supposed to fulfil the following standing conditions.

Assumption 1

1. *System (1) has an equilibrium point at the origin, that is* $f(0,0,0,0) = 0$.
2. *The control and state of the plant must fulfill the following constraints on the state and the input:*
$$(x(k),u(k)) \in Z \tag{2}$$
where $Z \subseteq \mathbb{R}^{n+m}$ *is closed and contains the origin in its interior.*
3. *The uncertainty signal w is modeled as follows*
$$w(k) = w_\eta(k)\ \eta(x(k),u(k)) \tag{3}$$
for all $k \geq 0$*, where* η *is a known function* $\eta : Z \subseteq \mathbb{R}^{n+m} \to \mathbb{R}_{\geq 0}$ *such that it is continuous and* $\eta(0,0) = 0$. $w_\eta \in \mathbb{R}^p$ *is an exogenous signal contained in a known compact set* $W_\eta \subset \mathbb{R}^p$. $W \subseteq \mathbb{R}^p$ *denotes the set where the signal w is confined, i.e.* $W \triangleq \{w \in \mathbb{R}^p : w = w_\eta \eta(x,u), w_\eta \in W_\eta, (x,u) \in Z\}$
4. *The disturbance signal d is such that* $d(k) \in D$ *for all* $k \geq 0$*, where* $D \subset \mathbb{R}^q$ *is a known compact set containing the origin.*
5. *The state of the plant* $x(k)$ *can be measured at each sample time.*

Remark 1. The distinction between a disturbance signal d and a state and input dependent uncertainty signal w is aimed to enhance the controller performance by taking advantage of the structure of the uncertainties [30]. However, if merely bounded disturbances are used to model uncertainties, then all the presented results can be applied by merely taking $w(k) = 0$ for all $k \geq 0$.

The nominal model of the plant (1) denotes the system considering zero-disturbance and it is given by

$$\tilde{x}(k+1) = \tilde{f}(\tilde{x}(k),u(k)),\ k \geq 0 \tag{4}$$

where $\tilde{f}(x,u) \triangleq f(x,u,0,0)$. The solution to this equation for a given initial state $x(0)$ is denoted as $\tilde{\phi}(k,x(0),\mathbf{u}) \triangleq \phi(k,x(0),\mathbf{u},\mathbf{0},\mathbf{0})$.

This paper is devoted to the stability analysis of the constrained uncertain system (1) (not necessarily continuous) controlled by a model predictive control law $u(k) = \kappa_N(x(k))$ (not necessarily continuous). This requires that there exists a stability region X_N where, if the initial state is inside this region, i.e. $x(0) \in X_N$, the evolution of the uncertain system fulfils the constraints (that is $(x(k), \kappa_N(x(k))) \in Z$ for all $k \geq 0$ for any possible evolution of the disturbance signals $d(k)$ and $w(k)$) and the robust stability property of the system is ensured.

In the following section of the paper, we show that the notion of input-to-state stability (ISS) is a suitable framework for the robust stability analysis of systems controlled by predictive controllers. Furthermore, this stability notion unifies in a

single framework other commonly-used robust stability notions, allowing existing results on this topic to be reviewed in a more general approach.

3 Input-to-State Stability

Consider that system (1) is controlled by a certain control law $u(k) = \kappa(x(k))$, then the closed loop system can be expressed as follows:

$$x(k+1) = f_\kappa(x(k), d(k), w(k)),\ k \geq 0 \tag{5}$$

where $f_\kappa(x,d,w) \triangleq f(x,\kappa(x),d,w)$. Consider also that assumption 1 holds for the controlled system. Define the set $X^\kappa \triangleq \{x \in \mathbb{R}^n : (x,\kappa(x)) \in Z\}$ and define the function $\eta_\kappa(x) \triangleq \eta(x,\kappa(x))$. The solution of this equation at sampling time k, for the initial state $x(0)$ and the sequences $\mathbf{d}$ and $\mathbf{w}$ is denoted as $\phi_\kappa(k,x(0),\mathbf{d},\mathbf{w})$. The nominal model function is denoted as $\tilde{f}_\kappa(x) \triangleq \tilde{f}(x,\kappa(x))$ and its solution is denoted as $\tilde{\phi}_\kappa(k,x(0)) \triangleq \phi_\kappa(k,x(0),\mathbf{0},\mathbf{0})$.

In this section, the input-to-state stability property is recalled showing its benefits for the analysis of robustness of a controlled system. Afterwards, a more involved notion, useful for the robustness analysis of predictive controllers, is presented: the regional input-to-state practical stability.

3.1 A Gentle Motivation for the ISS Notion

For the sake of clarity, in this section, state dependent uncertainties w are not considered (or considered zero) and, with a slight abuse of notation, this argument will be dropped in the previously defined functions. Besides, constraints are not taken into account.

A primary requirement of the controlled system is that, in absence of uncertainties, i.e. $d(k) = 0$ for all $k \in \mathbb{Z}_{\geq 0}$, the controlled system $\tilde{x}(k+1) = f_\kappa(\tilde{x}(k),0) \triangleq \tilde{f}_\kappa(\tilde{x}(k))$ is asymptotically stable. This property is defined as follows:

Definition 1. *The system $\tilde{x}(k+1) = \tilde{f}_\kappa(\tilde{x}(k))$ is (globally) asymptotically stable (0-AS) if there exists a $\mathcal{KL}$-function β such that $|\tilde{\phi}_\kappa(j,x(0))| \leq \beta(|x(0)|,j)$.*

This property is usually demonstrated by means of the existence of a (not necessarily continuous) Lyapunov function, which is defined as follows [61, §5.9].

Definition 2. *A function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is a Lyapunov function of system $\tilde{x}(k+1) = \tilde{f}_\kappa(\tilde{x}(k))$ if there exist three $\mathcal{K}_\infty$-functions, α_1, α_2 and α_3, such that $\alpha_1(|x|) \leq V(x) \leq \alpha_2(|x|)$ and $V(\tilde{f}_\kappa(x)) - V(x) \leq -\alpha_3(|x|)$.*

On the other hand, the effect of the uncertainty makes the system evolution differs from what expected. Then, it would be desirable that this effect is bounded and depends on the size of the uncertainty. This robustness condition has been expressed in the literature using the following notions [18]:

AG: System (5) has an asymptotic gain (AG) if there exists a $\mathcal{K}$-function γ_a such that for each $x(0)$ and $\mathbf{d} \in \mathscr{M}_D$, the state of the system satisfies the following property:

$$\limsup_{j\to\infty} |\phi_\kappa(j,x(0),\mathbf{d})| \leq \gamma_a\left(\limsup_{j\to\infty} |d(j)|\right)$$

This notion is closely related to the ultimately bounded property of a system [22]: the trajectories of the system converge asymptotically to a set which bound depends uniformly on the ultimate bound of the uncertainty.

SM: System (5) has a stability margin (SM) if there exists a $\mathcal{K}_\infty$-function σ such that for all $d(k) = \delta(k)\sigma(|x(k)|)$ with $|\delta(k)| \leq 1$, system (5) is asymptotically stable, i.e. there exist a $\mathcal{KL}$-function β such that for all $x(0)$

$$|\phi_\kappa(j,x(0),\mathbf{d})| \leq \beta(|x(0)|,j)$$

The function σ is called a stability margin. This definition states the existence of a (sufficiently small) state-dependent signal for which asymptotic stability of the uncertain system is maintained. This property is also called robust stability.

The notion of ISS is shown in the following definition [18]:

Definition 3. *System (5) is ISS if there exist a $\mathcal{KL}$-function β and a $\mathcal{K}$-function γ such that for all initial state $x(0)$ and sequence of disturbances $\mathbf{d} \in \mathscr{M}_D$,*

$$|\phi_\kappa(j,x(0),\mathbf{d})| \leq \beta(|x(0)|,j) + \gamma(\|\mathbf{d}_{[j-1]}\|) \tag{6}$$

The definition of input-to-state stability of a system comprises both effects (nominal stability and uniformly bounded effect of the uncertainties) in a single condition. In effect, notice that the ISS condition implies asymptotic stability of the undisturbed system (0-AS) (just taking $\mathbf{d} = \mathbf{0}$) and that the effect of the disturbance on the evolution of the states is bounded. Furthermore, if the disturbance signal fades, then the disturbed system asymptotically converges to the origin. Then, it is sensible to think that there exists a relation between ISS and the previous robust stability definitions [60]. This assertion is stated in the following theorem.

Theorem 1. *Consider system (5), then the following statements are equivalent*

1. *It is input-to-state stable (ISS).*
2. *The nominal system is asymptotically stable and the disturbed system has an asymptotic gain (0-AS+AG).*
3. *There exists a stability margin for the disturbed system (SM).*

The equivalence between ISS and SM has been proved in [18]. The fact that ISS is equivalent to 0-AS+AG has been proved in [11] as the counterpart for continuous-time systems [60]. If the model is discontinuous, it can be proved that the equivalence properties are valid [33].

Remark 2. Stability margin property can be used to demonstrate that a system is ISS even in the case that the uncertainty signal $d(k)$ is not decaying with the norm of the

state. The existence of a SM does not imply that the real signal will be bounded by the SM along the time.

Therefore, the ISS notion generalizes existing classic notions on stability of disturbed system allowing the study of the effect of state dependent, persistent or fading disturbances in a single framework. Moreover, as it will be presented in the following section, there exists Lyapunov-like conditions for the analysis of this property.

It is also interesting to study if a nominally asymptotically stable system (0-AS) has a certain degree of robustness. The answer to this question is negative in general, since examples can be found where the 0-AS systems presents zero robustness [20, 14]. Then, additional requirements on the system are necessary. The following theorem gives some sufficient conditions on the nominal system to be ISS, which extends the results of [14, 20] to ISS and generalizes [29]:

Theorem 2. *Assume that system (5) is such that the function $f_\kappa(x,d)$ is uniformly continuous in d for all $x \in \mathbb{R}^n$ and $d \in D$. Assume that system (5) is nominally asymptotically stable (0-AS), then this system is ISS if one of the following conditions holds:*

1. *There exists a Lyapunov function $V(x)$ for the nominal system which is uniformly continuous in $\mathbb{R}^n$.*
2. *Function $\tilde{f}_\kappa(x)$ is uniformly continuous in $x \in \mathbb{R}^n$.*

The proof of this theorem can be found in the appendix.

In some cases, robustness can only be ensured in a neighborhood of the origin and/or for small enough uncertainties. This problem can also be analyzed within the ISS framework by means of the local ISS notion.

Definition 4. *System (5) is said to be locally ISS if there exist constants c_1 and c_2, a $\mathcal{KL}$-function β and a $\mathcal{K}$-function γ such that*

$$|\phi_\kappa(j,x(0),\mathbf{d})| \leq \beta(|x(0)|,j) + \gamma(\|\mathbf{d}_{[j-1]}\|)$$

for all initial state $|x(0)| \leq c_1$ and disturbances $|d(j)| \leq c_2$.

Under milder conditions, the local ISS property can be ensured.

Corollary 1 (Local ISS). *The uniform continuity conditions of the latter theorem can be relaxed to achieve local ISS. In effect, assume that the uniform continuity condition of f_κ, $\tilde{f}_\kappa$ and V is replaced by merely continuity at a neighborhood of $x = 0$ and $d = 0$. Then in virtue of the Heine-Cantor theorem, there exist c_1 and c_2 such that for all x and d such that $|x| \leq c_1$ and $|d| \leq c_2$, these functions are uniformly continuous and the theorem can be applied yielding local ISS.*

3.2 Regional Input-to-State Practical Stability (ISpS)

In this section, a more general definition of the input-to-state stability is recalled: the regional input-to-state practical stability. The term regional refers to the fact that stability property holds in a certain region, which is compulsory for the analysis

of constrained systems. The term practical means that the input-to-state stability of a neighborhood of the origin could be only ensured. Furthermore, based on [53], extending the results of [30, 37, 26], Lyapunov-type sufficient conditions for ISpS are presented.

In the stability analysis of constrained systems, the invariance notion plays an important role. In the following definition, robust invariance for system (5) is presented.

Definition 5 (Robust positively invariant (RPI) set). *Consider that hypothesis 1 holds for system (5) . A set $\Gamma \subseteq \mathbb{R}^n$ is a robust positively invariant (RPI) set for system (5) if $f_\kappa(x,d,w) \in \Gamma$, for all $x \in \Gamma$, all $w_\eta \in W_\eta$ and all $d \in D$. Furthermore, if $\Gamma \subseteq X^\kappa$, then Γ is called admissible RPI set.*

For the robust stability analysis of systems controlled by predictive controllers it is appropriate to use a quite general notion of ISS: regional input-to-state practical stability (ISpS) [60, 18, 37] which is defined as follows:

Definition 6 (Regional *ISpS* in Γ). *Suppose that assumption 1 is satisfied for system (5). Given a set $\Gamma \subseteq \mathbb{R}^n$, including the origin as an interior point, system (5) is said to be input-to-state practical stable (ISpS) in Γ with respect to d if Γ is a robust positively invariant set for (5) and if there exist a $\mathscr{KL}$-function β, a $\mathscr{K}$-function γ_2 and a constant $c \geq 0$ such that*

$$|\phi_\kappa(j, x(0), \mathbf{d}, \mathbf{w})| \leq \beta(|x(0)|, j) + \gamma_2(||\mathbf{d}_{[j-1]}||) + c \tag{7}$$

for all $x(0) \in \Gamma$, $\mathbf{w}_\eta \in \mathscr{M}_{W_\eta}$, $\mathbf{d} \in \mathscr{M}_D$ and $k \geq 0$.

Remark 3. In the case that $c = 0$ in (7) , the system (5) is said to be input-to-state stable (ISS) in Γ with respect to d.

Remark 4. The constant c describes the fact that, in the case of zero disturbances, the controlled system (5) may not evolve to the origin, but to a compact neighborhood of the origin. Thus the ISpS property can also be defined as ISS with respect to a compact nominal invariant set [60, 18].

Regional ISpS with respect to d will be now associated to the existence of a suitable Lyapunov-like function (not necessarily continuous), which is defined below [30, 37].

Definition 7. *(ISpS-Lyapunov function in Γ) Suppose that assumption 1 is satisfied for system (5). Consider that Γ is a RPI set containing the origin in its interior. A function $V : \mathbb{R}^n \to \mathbb{R}_{\geq}0$ is called an ISpS-Lyapunov function in Γ for system (5) with respect to d, if there exists a compact set $\Omega \subseteq \Gamma$ (including the origin as an interior point), suitable $\mathscr{K}_\infty$-functions α_1, α_2, α_3, a $\mathscr{K}$-function λ_2 and a couple of constants $c_1, c_2 \geq 0$ such that:*

$$V(x) \geq \alpha_1(|x|), \qquad \forall x \in \Gamma \tag{8}$$

$$V(x) \leq \alpha_2(|x|) + c_1, \quad \forall x \in \Omega \tag{9}$$

and for all $x \in \Gamma$, $w_\eta \in W_\eta$, and $d \in D$,

$$V(f_\kappa(x,d,w)) - V(x) \leq -\alpha_3(|x|) + \lambda_2(|d|) + c_2 \tag{10}$$

Remark 5. A function $V\colon \mathbb{R}^n \to \mathbb{R}_{\geq}0$ is called an ISS-Lyapunov function in Γ if it is an ISpS-Lyapunov function in Γ with $c_1 = c_2 = 0$.

A sufficient condition, that extends the ISS results of [37] and [30] by means of lemma 4 (see the appendix), is stated in the following theorem.

Theorem 3. *Consider system (5) which fulfils assumption 1. If this system admits an ISpS-Lyapunov function in Γ w.r.t. d, then it is ISpS in Γ w.r.t. d.*

The proof can be derived from [53] taking into account lemma 4.

Remark 6. Notice also that *ISpS* w.r.t. d implicitly states that the state-dependent uncertainty w is bounded by a stability margin. Under this assumption, an ISpS-Lyapunov function w.r.t. w and d is also an ISpS-Lyapunov function w.r.t. d with a suitable redefinition of the supply functions [53].

4 Input-to-State Stability of Nominal MPC

In this section, robust stability of a nominal model predictive controller for system (1) fulfilling assumption 1 is analyzed. This predictive controller is called nominal because it has been designed using the nominal model of the system (4), that is, neglecting the disturbances. We focus on the standard model predictive control formulation presented in [42], which control law is derived from the solution of the following mathematical programming problem $P_N(x)$ parameterized in the current state x.

$$\min_{\mathbf{u}} \; V_N(x,\mathbf{u}) \triangleq \sum_{j=0}^{N-1} L(\tilde{x}(j),u(j)) + V_f(\tilde{x}(N)) \tag{11}$$

$$s.t. \;\; \tilde{x}(j) = \tilde{\phi}(j,x,\mathbf{u}),\; j \in \mathbb{Z}_{[0,N]} \tag{12}$$

$$(\tilde{x}(j),u(j)) \in Z,\; j \in \mathbb{Z}_{[0,N-1]} \tag{13}$$

$$\tilde{x}(N) \in X_f \tag{14}$$

where $L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ is the stage cost function, $V_f : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is the terminal cost function and $X_f \subseteq \mathbb{R}^n$ is the terminal region. It is assumed that $P_N(x)$ is feasible in a non-empty region denoted X_N. For each $x \in X_N$, the optimal decision variable of $P_N(x)$ is denoted $\mathbf{u}^*(x)$ and the optimal cost is $V_N^*(x)$. The MPC control law derives from the application of the solution in a receding horizon manner $\kappa_N(x) \triangleq u^*(0,x)$ and it is defined for all $x \in X_N$.

The stage cost L, the terminal cost V_f and the terminal set X_f of the MPC controller are considered to fulfil the following conditions.

Assumption 2. *The stage cost function $L(x,u)$ is such that $L(0,0) = 0$ and there exists a $\mathcal{K}_\infty$ function $\alpha(\cdot)$ such that $L(x,u) \geq \alpha(|x|)$ for all $(x,u) \in Z$. X_f is an*

admissible control invariant set for system (4), i.e. for all $x \in X_f$ there exists $u \in \mathbb{R}^m$ such that $(x,u) \in Z$ and $\tilde{f}(x,u) \in X_f$. V_f is a control Lyapunov function (CLF) for system (4) such that for all $x \in X_f$ there exist two $\mathscr{K}_\infty$-functions α_{V_f} and β_{V_f} satisfying $\alpha_{V_f}(|x|) \leq V_f(x) \leq \beta_{V_f}(|x|)$ and

$$\min_u \{V_f(\tilde{f}(x,u)) - V_f(x) + L(x,u) : (x,u) \in Z, \tilde{f}(x,u) \in X_f\} \leq 0, \quad \forall x \in X_f$$

It is well known that this assumption suffices to prove that the optimal cost $V_N^*(x)$ is a Lyapunov function of the closed-loop nominal system and the feasible region X_N is an admissible positively invariant set. Then the MPC control law asymptotically stabilizes system (4) with a domain of attraction X_N [42]. This property holds for any prediction horizon N, but a larger N may yield to a better closed-loop performance and a larger domain of attraction, but at expense of a larger computational burden of the calculation of the optimal solution.

The obtained control law stabilizes the nominal system but, is the stability maintained when it is applied to the uncertain system (1)? This important question has been recently studied. In [12, 8, 40, 17] robustness is analyzed for nonlinear systems based on the optimality of the control problem for unconstrained systems. In [59, 14, 29, 9, 56] more general results are obtained using continuity conditions on the solution of the optimization problem. In the following theorem, some of these results are generalized and extended to the ISS notion.

Theorem 4. *Consider a system given by (1) fulfilling assumption 1. Let $\kappa_N(x)$ be the predictive controller derived from the solution of $P_N(x)$ satisfying assumption 2 and let X_N be its feasibility region. Let the model function of system, $f(x,\kappa_N(x),d,w)$, be uniformly continuous in d and w for all $x \in X_N$, $d \in D$ and $w \in W$. If one of the following conditions holds:*

1. *Function $\tilde{f}(x,\kappa_N(x))$ is uniformly continuous in x for all $x \in X_N$.*
2. *The optimal cost $V_N^*(x)$ is uniformly continuous in X_N.*

then system (1) controlled by the nominal model predictive controller $u(k) = \kappa_N(x(k))$ fulfils the ISS property in a robust invariant set $\Omega_r \subseteq X_N$ for a sufficiently small bound of the uncertainties.

The proof of this theorem can be found in the appendix.

Between the two conditions for ISS stated in the latter theorem, uniform continuity of the optimal cost function results more interesting from a practical point of view since this can be ensured under certain conditions on the MPC problem. Some of these conditions are shown in the following proposition.

Proposition 1. *Assume that hypotheses of theorem 4 hold.*

C1: If the plant has only constraints on the inputs (i.e. $Z \triangleq \mathbb{R}^n \times U$ where $U \subseteq \mathbb{R}^m$), the terminal region is $X_f \triangleq \mathbb{R}^n$ and $V_N(x,\mathbf{u})$ is uniformly continuous in x for all $x \in \mathbb{R}^n$ and $\mathbf{u} \in \mathscr{M}_U$, then the optimal cost $V_N^*(x)$ is uniformly continuous in $\mathbb{R}^n$.

Besides, if $f(x,u,d,w)$ is uniformly continuous in x, $L(x,u)$ is uniformly continuous in x and $V_f(x)$ is uniformly continuous for all $x \in \mathbb{R}^n$, $u \in U$, $d \in D$ and $w \in W$, then uniform continuity condition of $V_N(x,\mathbf{u})$ holds.

C2: If the plant has constraints on the state and on the inputs (i.e. $Z \triangleq X \times U$ with $X \subseteq \mathbb{R}^n$ and $U \subseteq \mathbb{R}^m$), $V_f(x)$ is given by $V_f(x) \triangleq \lambda \mathscr{V}(x)$ where $\mathscr{V}(x)$ is a uniformly continuous CLF in X_f such that

$$\min_u \{\mathscr{V}(\tilde{f}(x,u)) - \mathscr{V}(x) + L(x,u) : (x,u) \in Z, \tilde{f}(x,u) \in X_f\} \leq 0$$

for all $x \in X_f$ with $\lambda \geq 1$ a given weighting factor, and $V_N(x,\mathbf{u})$ is uniformly continuous in x for all $x \in X_N$ and $\mathbf{u} \in \mathscr{M}_U$ then there exists a region $\Omega_r \triangleq \{x \in \mathbb{R}^n : V_N^*(x) \leq r\} \subseteq X_N$ in which the optimal cost is uniformly continuous. Besides larger λ yields to larger region Ω_r.

C3: If the nominal model is linear, the cost function $V_N(x,\mathbf{u})$ is linear and the set of constraints Z is a convex closed polyhedron, then optimal cost $V_N^*(x)$ is uniformly continuous in X_N.

C4: If the nominal model is linear, the cost function $V_N(x,\mathbf{u})$ is continuous and the set of constraints Z is a compact convex polyhedron, then optimal cost $V_N^*(x)$ is uniformly continuous in X_N.

The proof can be found in the appendix.

As studied in [14], MPC controllers may exhibit zero-robustness and hence may not be ISS. From the previously presented results, it can be seen that ISS can be ensured for some special classes of models and ingredients of the MPC. Thus, linear systems with bounded inputs controlled with a predictive control law are ISS if the cost function is continuous. For the case of nonlinear systems uniformly continuous in the uncertainty signals, uniform continuity of the cost function is assumed to derive ISS property. In the uncommon case that the set of constraints of the optimization problem does not depend on the state, ISS is proved. In the case that there exist some constraints depending on the state, such as constraints on the predicted states of the system or terminal constraints, these may cause discontinuity [45] and zero-robustness [14, Section 5]. Fortunately, in this case the closed-loop system will be ISS by merely weighting the terminal cost function. This is a simple and practical method to ensure robustness of the nominal MPC for continuous functions.

Uniform continuity of some functions involved in theorem 4 and property 1 play an important role in the demonstration of the ISS property. Taking into account the Heine-Cantor theorem, the conditions on the functions can be relaxed obtaining simpler results, as stated in the following corollary.

Corollary 2. *Under the hypotheses of theorem 4 and proposition 1, if the uniform continuity condition is replaced by continuity of the corresponding functions at a neighborhood of $x = 0$ and $d = 0$, then there exist c_1 and c_2 such that for all x and d such that $|x| \leq c_1$ and $|d| \leq c_2$, these functions are uniformly continuous and theorem 4 and proposition 1 can be applied yielding to local ISS.*

Remark 7 (Inherent robustness of suboptimal nominal MPC). Notice that in the previously presented results on local robustness of the nominal MPC it is implicitly

assumed that the optimal solution of $P_N(x)$ is achieved. However, it is well-known that this requirement is difficult to address when the system is non-linear. In [58], a stabilizing control algorithm based on suboptimal solutions to the optimization problem is presented. Nominal stability of this suboptimal practical procedure has been proved [58, 31].

The question that arises from this fact is if the local ISS property of the MPC based on optimal solutions still holds in case of suboptimality. Observe that suboptimal nominal predictive controller makes sense in presence of uncertainties since, in absence of uncertainties in the system, the feasible solution computed from the last optimal solution suffices for stability. Extending the results from [27, Thm 5.16], it can be proved that continuity of the optimal cost suffices to prove local ISS of the suboptimal controller, but a larger degree of suboptimality implies a lower stability margin.

5 Input-to-State Stability of Robust MPC

Earlier approaches of robust MPC formulations derive the control law from the solution of an optimization problem based on open-loop predictions of the uncertain system evolution. This open-loop scheme results to be very conservative from both a performance and domain of attraction points of view (see [42, Section 4]). In order to reduce this conservativeness, a closed-loop (or feedback) formulation of the MPC has been proposed [57]. In this case, control policies instead of control actions are taken as decision variables, yielding to an infinite dimensional optimization problem that is in general very difficult to solve and for which there exist few efficient algorithms in the literature in the case of linear systems [48, 13]. A practical formulation between these two approaches is the so-called semi-feedback formulation, where a family of parameterized control laws is used [23, 10]. Thus the decision variables are the sequence of the parameters of the control laws, and hence the optimization problem is a finite-dimensional mathematical programming problem.

Consider that the control actions are derived from a given family of controllers parameterized by $v \in \mathbb{R}^s$, $u(k) = \pi(x(k), v(k))$. Thus, system (1) is transformed in

$$x(k+1) = f_\pi(x(k), v(k), d(k), w(k)),\ k \geq 0 \tag{15}$$

where $f_\pi(x, v, d, w) \triangleq f(x, \pi(x, v), d, w)$ and v plays the role of the input of the modified system. The family of control laws is typically chosen as an affine function of the state. Notice that the open-loop formulation is included in the proposed semi-feedback approach.

Consider assumption 1 holds for system (1). Then it is convenient to pose constraint (2) in terms of v as follows

$$(x(k), v(k)) \in Z_\pi \tag{16}$$

where Z_π is such that $(x, \pi(x, v)) \in Z$ for all $(x, v) \in Z_\pi$. State and input dependent uncertainty signal w can also be written as follows

$$w(k) = w_\eta(k)\,\eta_\pi(x(k), v(k)) \tag{17}$$

where $\eta_\pi(x,v) \triangleq \eta(x,\pi(x,v))$. It will be useful to define the model function in terms of w_η as follows

$$f_{\pi\eta}(x,v,d,w_\eta) \triangleq f_\pi(x,v,d,w_\eta\,\eta_\pi(x,v)) \tag{18}$$

The nominal model of system (15) is denoted by $\tilde{f}_\pi(x,v) \triangleq f_\pi(x,v,0,0)$. The solution to the difference equation (15) at k sampling times, starting from x and for inputs $\mathbf{v}$, $\mathbf{d}$ and $\mathbf{w}$, is denoted by $\phi_\pi(k,x,\mathbf{v},\mathbf{d},\mathbf{w})$ and the solution to the nominal model $\tilde{\phi}_\pi(k,x,\mathbf{v}) \triangleq \phi_\pi(k,x,\mathbf{v},\mathbf{0},\mathbf{0})$.

In the following sections, robust stability of robust predictive controllers based on the nominal cost function and based on the worst-case cost function are analyzed by means of input to state stability framework. In both cases, robust constraint satisfaction will be guaranteed by means of the calculation of a sequence of reachable sets, commonly known as tube.

5.1 Tube Based Methods for Robust Constraint Satisfaction

The notion of tube (of trajectories), or sequence of reachable sets, was firstly introduced in [4] as a sequence of sets such that each set can be reached from the previous one. Recently, this idea has emerged again as a tool for robust constraint satisfaction [7, 28, 32, 5, 15] or for design robust predictive controllers for linear [24, 44] and for nonlinear systems [54, 55, 43]. In this section, this notion is presented as a practically attractive method for solving the robust constraint satisfaction.

Definition 8. *A sequence of sets $\{X_0, X_1, \cdots, X_N\}$, with $X_i \subset \mathbb{R}^n$, is called a tube (or a sequence of reachable sets) for system (15) and a given sequence of control inputs* $\mathbf{v}$*, if $f_{\pi\eta}(X_i, v(i), D, W_\eta) \subseteq X_{i+1}$, for all $i \in \mathbb{Z}_{[0,N-1]}$.*

A tube can be calculated by means of a suitable procedure to estimate the range of a function.

Definition 9. *Let $\mathbb{C}^b$ be a class of compact sets contained in $\mathbb{R}^b$ and let $F : \mathbb{R}^a \to \mathbb{R}^b$. Then a procedure is called a (guaranteed) range estimator of the function F, denoted by $\diamond F$, if for every compact set $X \subset \mathbb{R}^a$, $\diamond F(X)$ returns a set in $\mathbb{C}^b$ such that $F(x) \in \diamond F(X)$ for all $x \in X$.*

This procedure is assumed to be computationally tractable and uses a specialized algorithm or property to calculate a compact set of a certain class (as for instance, balls, boxes, intervals, zonotopes, polytopes) which is an outer bound of the exact range [32, 5, 1, 54]. Using this range estimator procedure of the function model $f_\pi(\cdot,\cdot,\cdot,\cdot)$, a tube can be computed by means of the following recursion:

$$X_{i+1} = \diamond f_{\pi\eta}(X_i, v(i), D, W_\eta) \tag{19}$$

for $i \in \mathbb{Z}_{[0,N-1]}$ and a given X_0. Notice that the procedure computes the range for the four arguments. This method has been used to compute a tube by several authors. In [47, 28] Lipschitz continuity of the model function is used in the estimation range procedure providing a ball $B_r \triangleq \{x \in \mathbb{R}^n : |x| \leq r\}$ as an estimation set (i.e. $\mathbb{C}^n$ is the set of balls in $\mathbb{R}^n$). In [32], the tube is calculated by using a range estimation procedure based on the interval extension of a function, returning interval sets as estimation. In [5] the procedure used is based on zonotope inclusion and an extension of the mean value theorem, and returns a zonotope (i.e. an affine mapping of a hypercube) as estimation region. In [1] a procedure based on DC-programming is used resulting in paralleltopes. In the case of linear systems, a tube centered in a nominal trajectory and a robust invariant set as cross section is used in [44] and in [55] a tube is computed for a class of nonlinear systems by means of a suitable transformation and using linear-case methods. Notice the latter procedures return convex compact sets, typically polytopes. For the computation of the tubes, the procedure is assumed to fulfil the following hypotheses.

Assumption 3. *The procedure* $\diamond f_{\pi\eta}(X, v, D, W_\eta)$ *is such that*

1. *For every* $A, B \subset \mathbb{R}^n$ *such that* $A \subseteq B$, *we have that* $\diamond f_\pi(A, v, D, W_\eta) \subseteq \diamond f_{\pi\eta}(B, v, D, W_\eta)$ *for every* v, D *and* W_η.
2. *Let* A *be a robust invariant set for system* (1) *controlled by* $u = \pi(x, v_f)$, *for all* $d(k) \in D$ *and* $w_\eta(k) \in W_\eta$. *Then* $\diamond f_{\pi\eta}(A, v_f, D, W_\eta) \subseteq A$.

These conditions are slightly restrictive but provide useful properties to the tube when used to design a predictive controller. These properties can be addressed (or relaxed) by a suitable procedure.

In the following sections, the tube-based method will be used in the predictive control formulations. The idea consists in replacing the constraints on the sequence of predicted trajectory with the constraints on the sequence of predicted reachable sets derived from the current state [32, 5]. In [44], the authors propose a tube with an invariant set as constant section and centered in a nominal predicted trajectory which initial state is considered as decision variable. Based on this ingredients and the superposition principle, a nice robust controller is derived. This formulation will not be used in this paper.

5.2 Predictive Controllers Based on Nominal Predictions

The robust nominal MPC is a natural extension of the nominal MPC to the case of uncertain systems [47, 28, 15] where the cost function is calculated for nominal predictions while the decision variables must be such that the constraints are fulfilled for any possible realization of the uncertainties.

The proposed predictive controller based on tubes is derived from the following optimization problem $P_N^{nt}(x)$:

$$\min_{\mathbf{v}} \; V_N(x,\mathbf{v}) \triangleq \sum_{j=0}^{N-1} L_\pi(\tilde{x}(j),v(j)) + V_f(\tilde{x}(N)) \tag{20}$$

$$s.t. \;\; \tilde{x}(j) = \tilde{\phi}_\pi(j,x,\mathbf{v}) \tag{21}$$

$$X_0 = \{x\} \tag{22}$$

$$X_{j+1} = \diamond f_{\pi\eta}(X_j, v(j), D, W_\eta),\; j \in \mathbb{Z}_{[0,N-1]} \tag{23}$$

$$X_j \times v(j) \subseteq Z_\pi,\; j \in \mathbb{Z}_{[0,N-1]} \tag{24}$$

$$X_N \subseteq X_f \tag{25}$$

where $L_\pi(x,v) \triangleq L(x,\pi(x,v))$. The feasibility region of this optimization problem is denoted by X_N^{nt} and $\mathbf{v}^*(x)$ denotes the optimal solution. The predictive control law is given by $u(k) = \kappa_N^{nt}(x(k)) \triangleq \pi(x(k),v^*(0;x(k)))$. The ingredients of the optimization problem and the system function must fulfil the following assumption.

Assumption 4. *The stage cost function $L(x,u)$ is such that $L(0,0) = 0$ and there exists a $\mathscr{K}_\infty$ function $\alpha(\cdot)$ such that $L(x,u) \geq \alpha(|x|)$ for all $(x,u) \in Z$. X_f is an admissible robust invariant set for system (15) for a suitable parameter v_f, i.e. for all $x \in X_f$, $(x,v_f) \in Z_\pi$ and $f_{\pi\eta}(x,v_f,d,w_\eta) \in X_f$ for all $d \in D$, $w_\eta \in W_\eta$. V_f is a Lyapunov function for the nominal system such that for all $x \in X_f$ there exist α_{V_f} and β_{V_f}, both $\mathscr{K}_\infty$-functions, satisfying $\alpha_{V_f}(|x|) \leq V_f(x) \leq \beta_{V_f}(|x|)$ and*

$$V_f(\tilde{f}_\pi(x,v_f)) - V_f(x) \leq -L_\pi(x,v_f)$$

Function $f_\pi(x,v,d,w)$ is uniformly continuous in x, d and w, $V_f(x)$ is uniformly continuous and $L_\pi(x,v)$ is uniformly continuous in x for all $(x,v) \in Z_\pi$, $d \in D$, $w \in W$.

In the following theorem, stability of this controller is stated.

Theorem 5. *Assume that the feasible set X_N^{nt} is not empty and consider that assumptions 3 and 4 hold. Then for all $x(0) \in X_N^{nt}$, system (1) controlled by $u = \kappa_N^{nt}(x)$ robustly fulfils constraint (2) and it is ISS in X_N^{nt}.*

The proof of this theorem can be found in the appendix

Remark 8. From the proof of theorem 5 it can be derived that at any sample $k > 0$, a feasible solution ensuring the ISS property can be constructed from the solution of the last sample. Then, optimality of the solution of $P_N^{nt}(x)$ is not required, but merely an enhanced solution (if possible) to the constructed feasible solution.

5.3 Min-Max Model Predictive Controllers

Robust predictive controllers based on nominal predictions have demonstrated to robustly stabilize the uncertain system. From a closed-loop performance index point

of view, there may exist uncertainty scenarios where the cost function based on nominal predictions is not a suitable measure of the closed-loop performance. This leads us to the well-known min-max formulation of the robust predictive controller [57, 42, 41, 10, 37, 30] and the $\mathscr{H}_\infty$ control [36, 16, 35, 38]. In this section, the ISS property of these controllers is studied. As in the previous section, a semi-feedback formulation of the controller is considered due to practical reasons, but the presented results can be extended to the feedback case. A tube-based formulation for the robust constraint satisfaction is also used leading to a novel min-max formulation. Notice that if it is assumed that the range estimation procedure returns the exact set, then the presented formulation reduces to the standard one [42, 53]. The min-max optimization problem $P_N^r(x)$ is as follows:

$$\begin{aligned}
\min_{\mathbf{v}} \max_{\mathbf{d}\in\mathscr{M}_D,\mathbf{w}_\eta\in\mathscr{M}_{W_\eta}} \quad & \sum_{j=0}^{N-1} L_\pi(x(j),v(j),d(j),w(j)) + V_f(x(N)) \\
s.t. \quad & x(j) = \phi_\pi(j,x,\mathbf{v},\mathbf{d},\mathbf{w}), \\
& w(j) = w_\eta(j)\,\eta_\pi(x(j),v(j)) \\
& X_0 = \{x\}, \\
& X_{j+1} = \diamond f_{\pi\eta}(X_j,v(j),D,W_\eta),\ j\in\mathbb{Z}_{[0,N-1]} \\
& X_j \times v(j) \subseteq Z_\pi,\ j\in\mathbb{Z}_{[0,N-1]} \\
& X_N \subseteq X_f.
\end{aligned}$$

where $L_\pi(x,v,d,w) \triangleq L(x,\pi(x,v),d,w)$. The feasibility region of this optimization problem is denoted by X_N^r and the optimal cost of this optimization problem is denoted by $V_N^r(x)$. The control law is given by $u(k) = \kappa_N^r(x(k)) \triangleq \pi(x(k),v^*(0;x(k)))$. The parameters of the controller are supposed to fulfil the following conditions:

Assumption 5. *The stage cost function $L(x,u,d,w)$ is such that $L(0,0,0,0)=0$ and there exists a couple of $\mathscr{K}_\infty$ functions α_{Lx}, α_{Ld} such that $L(x,u,d,w) \geq \alpha_{Lx}(|x|) - \alpha_{Ld}(|d|)$ for all $(x,u)\in Z$, $w\in W$ and $d\in D$. X_f is an admissible robust invariant set for system (15) for a suitable parameter v_f, i.e. for all $x\in X_f$, $(x,v_f)\in Z_\pi$ and $f_\pi(x,v_f,d,w)\in X_f$ for all $w_\eta\in W_\eta$ and $d\in D$. V_f is an ISS-Lyapunov function w.r.t. signal d in X_f such that for all $x\in X_f$, there exist α_{V_f} and β_{V_f}, both $\mathscr{K}_\infty$-functions and a $\mathscr{K}$-function ρ, satisfying $\alpha_{V_f}(|x|) \leq V_f(x) \leq \beta_{V_f}(|x|)$ and*

$$V_f(f_\pi(x,v_f,d,w)) - V_f(x) \leq -L_\pi(x,v_f,d,w) + \rho(|d|), \forall w_\eta\in W_\eta, \forall d\in D$$

Notice that this assumption implies that the system controlled by the terminal control law $u(k)=\pi(x(k),v_f)$ is ISS w.r.t the disturbance d in X_f and hence w is bounded by a stability margin for all $x(k)\in X_f$. Input-to-state stability of the min-max controller is proved in the following theorem which is a generalization of [30, 37].

Theorem 6. *[53] Assume that the feasibility region X_N^r is a non-empty compact set and assumptions 3 and 5 hold, then system (1) controlled by $u(k) = \kappa_N^r(x(k))$ is ISpS with respect to d in the robust invariant region X_N^r.*

The proof of this theorem is derived demonstrating that the optimal cost function of $P_N^r(x)$ is an ISpS-Lyapunov function in X_N^r. Moreover, this satisfies the following inequality

$$V_N^r(f(x,\kappa_N^r(x),d,w)) - V_N^r(x) \leq -\alpha_{Lx}(|x|) + \alpha_{Ld}(|d|) + \rho(D^{sup})$$

This highlights two interesting properties of min-max controllers [30, 26]: if the uncertainty $w(k)$ is locally bounded by a stability margin for the system controlled by the terminal control law, then the min-max controller inherits the stability margin and extends it to its domain of attraction X_N^r (see remark 6). The second interesting property stems from the term $\rho(D^{sup})$. This term is the responsible for the practical stability of the closed-loop system and it is derived from the worst-case nature of the min-max optimization problem. Thus, despite the signal $d(k)$ fades, the state of the closed loop system may not converge to the origin.

Input-to-state stability of the min-max controllers can be achieved by two methods: the most simple one is based on the application of the controller in a dual-mode manner, switching to the terminal control law once $x(k) \in X_f$ [26]. The second method is based on the $\mathscr{H}_\infty$ strategy. This is derived from the optimization problem $P_N^r(x)$ taking a suitable choice of the stage cost [35], as stated in the following theorem.

Theorem 7. *[53] Let $L_d(d)$ be a definite positive function and let the stage cost function $L(x,u,d,w) \triangleq L_x(x,u,w) - L_d(d)$ be such that assumption 5 holds for $\rho \equiv 0$. Assume that the feasibility region X_N^r is non-empty then, under assumptions 3 and 5, the closed-loop system $x(k+1) = f(x(k),\kappa_N^r(x(k)),d(k),w(k))$ is ISS with respect to d in the robust invariant region X_N^r.*

The stability proof of min-max predictive controllers can be obtained by means of dynamic programming [41, 30] or by means of monotonicity of the cost function [35, 37, 2].

Remark 9. Notice that no assumption on the continuity of the model function, stage cost function or terminal cost function are considered in the stability conditions of the min-max controller. This means that min-max is suitable for robust stability of discontinuous systems and/or discontinuous cost functions.

Remark 10. From standard arguments, it can be proved that suboptimal solutions in the minimization of $P_N^r(x)$ can be tolerated retaining stability, but sub-optimality of the maximization stage requires further analysis. This problem is rather interesting since the maximization stage may be computationally more demanding than the minimization one. In [2] it is demonstrated for linear systems how maximization can be relaxed yielding to input-to-state practical stability. This problem has been extended to nonlinear systems in [52].

6 Conclusions

In this paper, existing results on robust model predictive control are reviewed demonstrating how an unifying framework for robust stability analysis can be used: input-to-state stability. Sufficient conditions for local ISS of nominal MPC are stated. In the case of robust MPC, a semi-feedback tube-based formulation is proposed and sufficient conditions for ISS are given in the case of nominal predictions. This framework has also been used for the min-max MPC and its stability has also been studied. Moreover, the robust stability results are valid in the suboptimal case.

However, there are many interesting and open topics on stability and robustness analysis of MPC, as the following: stability in presence of discontinuous model functions, robust constraint satisfaction and enhanced estimation of the tubes, suboptimal approaches of predictive controllers, mainly in the min-max approach, decentralized and distributed formulations for large-scale systems applications or tracking of changing operating points and trajectories.

A Proof of the Theorems

A.1 Technical Lemmas

The following lemmas play an important role in the theoretical development of the results presented in this paper. Their proofs are omitted by lack of space.

Lemma 1. *Let f be a function $f(x,y) : \mathbb{R}^a \times \mathbb{R}^b \to \mathbb{R}^c$. Then f is an uniformly continuous function in x for all $x \in A$ and $y \in B$ iff there exists a $\mathcal{K}_\infty$-function σ such that*

$$|f(x_1,y) - f(x_2,y)| \leq \sigma(|x_1 - x_2|), \quad \forall x_1, x_2 \in A, \forall y \in B$$

Lemma 2. *Consider a system defined by the difference equation $x(k+1) = f(x(k),u(k))$ with $(x(k),u(k)) \in Z$. Denote $\phi(j,x,\mathbf{u})$ the solution to this equation. Assume that f is assumed to be uniformly continuous in x for all $(x,u) \in Z$ and σ_x is a suitable function such that $|f(x,u) - f(y,u)| \leq \sigma_x(|x-y|)$. Then*

$$|\phi(j,x,\mathbf{u}) - \phi(j,y,\mathbf{u})| \leq \sigma_x^j(|x-y|).$$

Lemma 3. *Let $\Gamma \subseteq \mathbb{R}^n$ be a set with the origin in its interior and let $V(x) : \Gamma \to \mathbb{R}_{\geq 0}$ be a positive definite function continuous at a neighborhood of the origin, then there exists a $\mathcal{K}_\infty$-function α such that $V(x) \leq \alpha(|x|)$ for all $x \in \Gamma$.*

Lemma 4. *Consider a couple of sets $\Gamma, \Omega \subseteq \mathbb{R}^n$ both of them with the origin in their interior and such that $\Omega \subseteq \Gamma$. Let $V(x) : \Gamma \subseteq \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ be a function such that: (i) $V(x) < \infty$ for all $x \in \Gamma$, (ii) there exists a $\mathcal{K}$-function α_1 such that $V(x) \geq \alpha_1(|x|)$ for all $x \in \Gamma$ and (iii) there exists a $\mathcal{K}$-function α_2 and a positive constant $c \geq 0$ such that $V(x) \leq \alpha_2(|x|) + c, \quad \forall x \in \Omega$. Then there exists a $\mathcal{K}_\infty$-function β such that $V(x) \leq \beta(|x|) + c, \forall x \in \Gamma$.*

A.2 Proof of Theorem 2

1. Let $V(x)$ be the uniform continuous Lyapunov function according to the definition 2. In virtue of lemma 1, there exists a $\mathscr{K}$-function σ_V such that $|V(y) - V(x)| \leq \sigma_V(|y-x|)$. Moreover, from the uniform continuity of f_κ w.r.t. d, there exists a $\mathscr{K}$-function σ_d such that $|f_\kappa(x,d_1) - f_\kappa(x,d_2)| \leq \sigma_d(|d_2 - d_1|)$. From these facts, it is inferred that

$$\begin{aligned} V(f_\kappa(x,d)) - V(x) &= V(f_\kappa(x,d)) - V(f_\kappa(x,0)) + V(f_\kappa(x,0)) - V(x) \\ &\leq |V(f_\kappa(x,d)) - V(f_\kappa(x,0))| - \alpha_3(|x|) \\ &\leq \sigma_V(|f_\kappa(x,d) - f_\kappa(x,0)|) - \alpha_3(|x|) \\ &\leq \sigma_V \circ \sigma_d(|d|) - \alpha_3(|x|) \end{aligned}$$

   Then $V(x)$ is a ISS-Lyapunov function and in virtue of [18] the system is ISS.
2. The asymptotic stability of the nominal system implies that there exists a $\mathscr{KL}$-function β such that $|\phi_\kappa(j,x(0),\mathbf{0})| \leq \beta(|x(0)|,j)$.

   From this property, it will be proved that there exists a uniform continuous Lyapunov function for the nominal system, following a similar procedure to [50]. For every $\mathscr{KL}$-function β, there exists a couple of $\mathscr{K}_\infty$ functions α_1 and α_2 and a constant $a > 0$ such that $\beta(s,t) \leq \alpha_1^{-1}(\alpha_2(s)e^{-at})$ [50].

   Define the function

$$V(x) \triangleq \sup_{j\geq 0} \left(\alpha_1\left(\phi_\kappa(j,x,\mathbf{0})\right) e^{aj}\right)$$

   Since $f_\kappa(x,0)$ is uniformly continuous, then $\phi_\kappa(j,x,\mathbf{0})$ is uniformly continuous in x, and hence $V(x)$ inherits this property.

   See that $\alpha_1\left(\phi_\kappa(j,x,\mathbf{0})\right) e^{aj}$ is upper bounded by $\alpha_2(|x|)$, and hence the function is well-defined. Besides, this function satisfies $\alpha_1(|x|) \leq V(x) \leq \alpha_2(|x|)$. Then, it suffices to proof the decreasing property:

$$\begin{aligned} V(f_\kappa(x,0)) &= \sup_{j\geq 0} \left(\alpha_1\left(|\phi_\kappa(j,f_\kappa(x,0),\mathbf{0})|\right) e^{aj}\right) \\ &= \sup_{j\geq 1} \left(\alpha_1\left(|\phi_\kappa(j,x,\mathbf{0})|\right) e^{a(j-1)}\right) \\ &\leq \sup_{j\geq 0} \left(\alpha_1\left(|\phi_\kappa(j,x,\mathbf{0})|\right) e^{aj}\right) e^{-a} \\ &= V(x)e^{-a} = V(x) - (1-e^{-a})V(x) \\ &\leq V(x) - (1-e^{-a})\alpha_1(|x|) \end{aligned}$$

   Hence there exists a uniformly continuous Lyapunov function $V(x)$ for the nominal system and then in virtue of the first statement of theorem 2 the disturbed system is ISS.

A.3 Proof of theorem 4

First, the ISS condition will be proved. Notice that $f(x,\kappa_N(x),d,w) \in \Omega_r \subseteq X_N$ for all $x \in \Omega_r$ and hence the control law is defined along the time.

1. Since the closed-loop model function $f(x,\kappa_N(x),d,w)$ is uniformly continuous in all its arguments, in virtue of theorem 2, the ISS property of the closed-loop system is derived.
2. ISS property directly stems from theorem 2.

Once the ISS property is derived, the existence of a compact robust invariant set Ω_r is proved for a sufficiently small bound of the uncertainties. From the first part of the proof, it can be said that in all the cases there exists a uniformly continuous Lyapunov function $\tilde{V}(x)$ in X_N. Define Ω_r as $\Omega_r \triangleq \{x : \tilde{V}(x) \leq r\}$ such that $\Omega_r \subseteq X_N$. Then there exist a couple of $\mathcal{K}$-functions γ_d, γ_w

$$\tilde{V}(f(x,\kappa_N(x),d,w)) \leq \tilde{V}(x) - L(x,\kappa_N(x)) + \gamma_d(|d|) + \gamma_w(|w|)$$

for all $x \in X_N$. From standard Lyapunov theory arguments (see [18] for instance) there exists a $\mathcal{K}_\infty$ function α_1 such that $\tilde{V}(x) \geq \alpha_1(|x|)$. Furthermore, there exists a $\mathcal{K}_\infty$ function ρ such that $\theta(s) = s - \rho(s)$ is a $\mathcal{K}$-function and ensures that $\tilde{V}(f(x,\kappa_N(x),d,w)) \leq \rho \circ \tilde{V}(x) + \gamma_d(|d|) + \gamma_w(|w|)$ [18, Lemma 3.5]. Taking a couple of constants c_1 and c_2 such that $\gamma_d(c_1) + \gamma_w(c_2) \leq \theta(r)$, then we have that for all $|d| \leq c_1$ and $|w| \leq c_2$,

$$\begin{aligned}\tilde{V}(f(x,\kappa_N(x),d,w)) &\leq \rho(r) + \gamma_d(|d|) + \gamma_w(|w|)\\ &\leq \rho(r) + \theta(r) = r\end{aligned}$$

Then, robust invariance of Ω_r is derived.

A.4 Proof of Proposition 1

1. Consider a given sequence of control actions $\mathbf{u}$ and $x, z \in \mathbb{R}^n$, then in virtue of the uniform continuity of the cost function $V_N(x,\mathbf{u})$ there exists a $\mathcal{K}_\infty$-function θ such that $|V_N(x,\mathbf{u}) - V_N(z,\mathbf{u})| \leq \theta(|x-z|)$.

   Let $\mathbf{u}^*(x)$ be the optimal solution of $P_N(x)$. Since the constraints of $P_N(x)$ do not depend on the state x, $\mathbf{u}^*(x)$ is feasible for any $x \in \Omega_r$. Assume (without loss of generality) that $V_N^*(z) \geq V_N^*(x)$, then

   $$|V_N^*(z) - V_N^*(x)| \leq |V_N(z,\mathbf{u}^*(x)) - V_N(x,\mathbf{u}^*(x))| \leq \theta(|x-z|)$$

   Therefore, the optimal cost is uniformly continuous in $\mathbb{R}^n$.

   In order to prove the second statement, denote $\tilde{x}(i) = \tilde{\phi}(i,\tilde{x},\mathbf{u})$ and $\tilde{z}(i) = \tilde{\phi}(i,\tilde{z},\mathbf{u})$. Uniform continuity of the model function implies the existence of a $\mathcal{K}_\infty$-function σ_x such that $|f(x,u,d,w) - f(z,u,d,w)| \leq \sigma_x(|x-z|)$ for all x, z in $\mathbb{R}^n$. Then from lemma 2 it is derived that $|\tilde{x}(i) - \tilde{z}(i)| \leq \sigma_x^i(|\tilde{x}(0) - \tilde{z}(0)|)$.

Analogously, there also exists a couple of $\mathscr{K}_\infty$-functions σ_L, σ_{V_f} such that $|L(x,u)-L(z,u)| \leq \sigma_L(|x-z|)$ and $|V_f(x)-V_f(z)| \leq \sigma_{V_f}(|x-z|)$ for all $x,z \in \mathbb{R}^n$ and $u \in U$. Considering these results we have that

$$\begin{aligned} |V_N(x,\mathbf{u})-V_N(z,\mathbf{u})| &\leq \sum_{j=0}^{N-1} |L(\tilde{x}(j),u(j))-L(\tilde{z}(j),u(j))| \\ &\quad +|V_f(\tilde{x}(N))-V_f(\tilde{z}(N))| \\ &\leq \sum_{j=0}^{N-1} \sigma_L \circ \sigma_x^j(|x-z|)+\sigma_{V_f}\circ\sigma_x^N(|x-z|) \end{aligned}$$

Hence the cost function is uniformly continuous.

2. Consider the compact set $\Omega_r \triangleq \{x \in \mathbb{R}^n : V_N^*(x) \leq r\}$ contained in the interior of $X_N \subseteq X$. Then this set is an admissible invariant set for the nominal system controlled by the MPC. Take a $x \in \Omega_r$ and denote $x^*(j)$ the optimal trajectory of the solution of $P_N(x)$, then from the Bellman's optimality principle, $V_N^*(x) \geq V_{N-j}^*(x^*(j))$ for all $j \in \mathbb{Z}_{[0,N]}$ Given that for all $z \in X_N$, $V_N^*(z) \leq V_{N-j}^*(z)$, we have that $V_N^*(x^*(j)) \leq V_N^*(x)$. Then for all $x \in \Omega_r$, the predicted optimal trajectory remains in Ω_r and hence the constraint on the states is not active. Therefore, this can be removed from $P_N(x)$.

   On the other hand, taking into account the results reported in [31], for any λ such that $V_f(x) = \lambda \mathscr{V}(x)$, there exists a value of $r(\lambda)$ such that for all $x \in \Omega_r$ the terminal constraint is not active throughout the state evolution. Besides, larger λ, larger the region Ω_r. Then, the terminal constraint could also be removed from the optimization problem yielding to a set of constraints of $P_N(x)$ which does not depend on the state x. From the arguments of the latter case, the optimal cost is uniformly continuous in Ω_r.
3. In this case, the optimal cost function is a piece-wise affine continuous function [3]. Then $V_N^*(x)$ is Lipschitz and hence uniformly continuous in its domain.
4. This proof can be derived from [14, Proposition 12].

A.5 Proof of Theorem 5

First, recursive feasibility will be proved. Thus, consider a state $x_k \in X_N^{nt}$ and denote $\mathbf{v}^*(x)$ the optimal solution of $P_N^{nt}(x)$ and denote $\{X_j^*(x)\}$ the sequence of reachable sets for the optimal solution.

Let x^+ be the actual state at next sampling time, i.e. $x^+ = f(x, \kappa_N^{nt}(x), d, w)$. Define the following sequence of control inputs to be applied at this state $\bar{\mathbf{v}}(x^+) \triangleq \{v^*(1,x), \cdots, v^*(N-1,x), v_f\}$ and define the sequence of sets

$$\bar{X}_{j+1}(x^+) \triangleq \diamond f_{\pi\eta}(\bar{X}_j(x^+), \bar{v}(j,x^+), D, W_\eta)$$

with $\bar{X}_0(x^+) = \{x^+\}$. Since $x^+ \in X_1^*(x)$, then it can be proved from the property 1 of assumption 3 $\bar{X}_j(x^+) \subseteq X_{j+1}^*(x)$. Therefore

$$\bar{X}_j(x^+) \times \bar{v}(j,x^+) \subseteq X^*_{j+1}(x) \times v^*(j+1,x) \subseteq Z_\pi, \quad j \in \mathbb{Z}_{[0,N-2]}$$

Moreover, since $\bar{X}_{N-1}(x^+) \subseteq X^*_N(x) \subseteq X_f$ we have that

$$\bar{X}_{N-1}(x^+) \times \bar{v}(N-1,x^+) \subseteq X_f \times v_f \subseteq Z_\pi$$

Finally, from assumption 3, we have that

$$\bar{X}_N(x^+) \triangleq \diamond f_{\pi\eta}(\bar{X}_{N-1}(x^+), v_f, D, W_\eta) \subseteq \diamond f_{\pi\eta}(X_f, v_f, D, W_\eta) \subseteq X_f$$

Therefore, the sequence $\bar{\mathbf{v}}(x^+)$ is a feasible solution of $P^{nt}_N(x^+)$ and hence recursive feasibility and robust invariance of X^{nt}_N are proved.

To demonstrate the ISS property, let define the following sequences $\tilde{z}(i) \triangleq \tilde{\phi}_\pi(i,x^+,\bar{\mathbf{v}}(x^+))$ and $\tilde{x}^*(i) \triangleq \tilde{\phi}_\pi(i,x,\mathbf{v}^*(x))$ for $i \in \mathbb{Z}_{[0,N-1]}$. From the assumption 4, there exist some $\mathscr{K}_\infty$-functions σ_x, σ_d, σ_w, σ_L, σ_{V_f}, such that

$$\begin{aligned} |f_\pi(x_1,v,d_1,w_1) - f_\pi(x_2,v,d_2,w_2)| &\leq \sigma_x(|x_1-x_2|) + \sigma_d(|d_1-d_2|) \\ &\quad + \sigma_w(|w_1-w_2|) \\ |L_\pi(x,v) - L_\pi(z,v)| &\leq \sigma_L(|x-z|) \\ |V_f(x) - V_f(z)| &\leq \sigma_{V_f}(|x-z|) \end{aligned}$$

for all (x,u) in Z_π $d \in D$ and $w \in W$.

Taking into account this fact and considering that $\bar{v}(i,x^+) \triangleq v^*(i+1,x)$ for all $i \in \mathbb{Z}_{[0,N-1]}$, it can be inferred that $|\tilde{z}(i) - \tilde{x}^*(i+1)| \leq \sigma^i_x(|x^+ - \tilde{x}^*(1)|), \quad i \in \mathbb{Z}_{[0,N-1]}$. Denote $\bar{V}_N(x^+) \triangleq V_N(x^+,\bar{\mathbf{v}}(x^+))$, then, we can state

$$\begin{aligned} \bar{V}_N(x^+) - V^*_N(x) = & -L(x,\kappa^{nt}_N(x)) \\ & + \sum_{j=0}^{N-2} [L_\pi(\tilde{z}(j),\bar{v}(j,x^+)) - L_\pi(\tilde{x}^*(j+1),v^*(j+1,x))] \\ & + L_\pi(\tilde{z}(N-1),v_f) + V_f(\tilde{z}(N)) - V_f(\tilde{z}(N-1)) \\ & + V_f(\tilde{z}(N-1)) - V_f(\tilde{x}^*(N)) \end{aligned}$$

Taking into account that $\bar{v}(i,x^+) \triangleq v^*(i+1,x)$, the following bound can be obtained

$$L_\pi(\tilde{z}(j),\bar{v}(j,x^+)) - L_\pi(\tilde{x}^*(j+1),v^*(j+1,x)) \leq \sigma_L \circ \sigma^j_x(|x^+ - x^*(1)|)$$

for all $j \in \mathbb{Z}_{[0,N-2]}$ and similarly,

$$V_f(\tilde{z}(N-1)) - V_f(\tilde{x}^*(N)) \leq \sigma_{V_f} \circ \sigma^N_x(|x^+ - x^*(1)|)$$

Considering that $\tilde{z}(N-1) \in \bar{X}_{N-1}(x^+)$, and $\bar{X}_{N-1}(x^+) \subseteq X^*_N(x) \subseteq X_f$, we have that $\tilde{z}(N-1) \in X_f$. From assumption 4, we have that

$$L_\pi(\tilde{z}(N-1),v_f) + V_f(\tilde{z}(N)) - V_f(\tilde{z}(N-1)) \leq 0$$

This yields to $\bar{V}_N(x^+) - V_N^*(x) \leq -L(x, \kappa_N^{nt}(x)) + \gamma(|x^+ - x^*(1)|)$, where $\gamma(s) \triangleq \sum_{j=0}^{N-2} \sigma_L \circ \sigma_x^j(s) + \sigma_{V_f} \circ \sigma_x^N(s)$. On the other hand, since $x^+ \triangleq f(x, \kappa_N^{nt}(x), d, w)$ and $x^*(1) \triangleq f(x, \kappa_N^{nt}(x), 0, 0)$, we have that

$$|x^+ - \tilde{x}^*(1)| = |f(x, \kappa_N^{nt}(x), d, w) - f(x, \kappa_N^{nt}(x), 0, 0)| \leq \sigma_d(|d|) + \sigma_w(|w|)$$

Then there exists a couple of $\mathscr{K}$-functions θ_d and θ_w such that

$$\bar{V}_N(x^+) - V_N^*(x) \leq -\alpha(|x|) + \theta_d(|d|) + \theta_w(|w|)$$

Since $\bar{\mathbf{v}}(x^+)$ is a feasible solution of $P_N^{nt}(x^+)$ we have that $V_N^*(x^+) \leq \bar{V}_N(x^+)$, and then $V_N^*(x^+) - V_N^*(x) \leq -\alpha(|x|) + \theta_d(|d|) + \theta_w(|w|)$. Taking into account that $V_N^*(x)$ is a definite positive function continuous in a neighborhood of the origin, in virtue of lemma 3 there exists a couple of $\mathscr{K}$-functions α_1 and α_2 such that $\alpha_1(|x|) \leq V_N^*(x) \leq \alpha_2(|x|)$ for all $x \in X_N^{nt}$. Hence the closed-loop system is ISS.

Acknowledgements. This work has been partially supported by the MCINN-Spain (contract DPI2007-66718-C04-01). We would like to thank Dr. Bruno Picasso and the reviewers for their interesting and useful comments.

References

1. Alamo, T., Bravo, J.M., Redondo, M.J., Camacho, E.F.: A set-membership state estimaton algorithm based on DC programming. Automatica 44, 216–224 (2008)
2. Alamo, T., Muñoz de la Peña, D., Limon, D., Camacho, E.F.: Constrained min-max predictive control: modifications of the objective function leading to polynomial complexity. IEEE Transactions on Automatic Control 50(5), 710–714 (2005)
3. Bemporad, A., Borelli, F., Morari, M.: Model predictive control based on linear programming - the explicit solution. IEEE Transactions on Automatic Control 47(12), 1974–1985 (2002)
4. Bertsekas, D.M., Rhodes, I.B.: On the minimax reachability of target sets and target tubes. Automatica 7, 233–247 (1971)
5. Bravo, J.M., Alamo, T., Camacho, E.F.: Robust MPC of constrained discrete time nonlinear systems based on approximated reachable sets. Automatica 42, 1745–1751 (2006)
6. Camacho, E.F., Bordons, C.: Model Predictive Control. Springer, Heidelberg (2004)
7. Chisci, L., Rossiter, J.A., Zappa, G.: Systems with persistent disturbances: predictive control with restricted constraints. Automatica 37, 1019–1028 (2001)
8. De Nicolao, G., Magni, L., Scattolini, R.: On the robustness of receding-horizon control with terminal constraints. IEEE Transactions on Automatic Control 41, 451–453 (1996)
9. De Nicolao, G., Magni, L., Scattolini, R.: Stability and robustness of nonlinear receding-horizon control. In: NMPC Workshop - Assessment and Future Directions, Ascona, Switzerland, June 3-5, pp. 77–90 (1998)
10. Fontes, F.A.C.C., Magni, L.: Min-max model predictive control of nonlinear systems using discontinuous feedbacks. IEEE Transactions on Automatic Control 48(10), 1750–1755 (2003)

11. Gao, K., Lin, Y.: On equivalent notions of input-to-state stability for nonlinear discrete time systems. In: IASTED International Conference on Control and Applications, pp. 81–86 (2000)
12. Geromel, J.C., Da Cruz, J.J.: On the robustness of optimal regulators for nonlinear discrete-time systems. IEEE Transactions on Automatic Control AC-32, 703–710 (1987)
13. Goulart, P.J., Kerrigan, E.C., Maciejowski, J.M.: Optimization over state feedback policies for robust control with constraints. Automatica 42(4), 523–533 (2006)
14. Grimm, G., Messina, M.J., Tuna, S.E., Teel, A.R.: Examples when nonlinear model predictive control is nonrobust. Automatica 40(10), 1729–1738 (2004)
15. Grimm, G., Messina, M.J., Tuna, S.E., Teel, A.R.: Nominally Robust Model Predictive Control With State Constraints. IEEE Transactions on Automatic Control 52(10), 1856–1870 (2007)
16. Gyurkovics, E.: Receding horizon $\mathscr{H}_\infty$ control for nonlinear discrete-time systems. IEE Proceedings - Control Theory and Applications 149, 540–546 (2002)
17. Imsland, L., Findeisen, R., Bullinger, E., Allgöwer, F., Foss, B.A.: A note on stability, robustness and performance of output feedback nonlinear model predictive control. Journal of Process Control 13, 633–644 (2003)
18. Jiang, Z.-P., Wang, Y.: Input-to-state stability for discrete-time nonlinear systems. Automatica 37, 857–869 (2001)
19. Jiang, Z.-P., Wang, Y.: A converse Lyapunov theorem for discrete-time systems with disturbances. System & Control Letters 45, 49–58 (2002)
20. Kellett, C.M., Teel, A.R.: Smooth lyapunov functions and robustness of stability for difference inclusions. Systems & Control Letters 52, 395–405 (2004)
21. Kerrigan, E.C., Maciejowski, J.M.: Robust feasibility in model predictive control: Necessary and sufficient conditions. In: Proceedings of the IEEE Conference on Decision and Control (2001)
22. Khalil, H.K.: Nonlinear systems. Prentice-Hall, Englewood Cliffs (1996)
23. Kothare, M.V., Balakrishnan, M., Morari, M.: Robust constrained model predictive control using linear matrix inequalities. Automatica 32(10), 1361–1379 (1996)
24. Langson, W., Chryssochoos, I., Rakovic, S.V., Mayne, D.Q.: Robust model predictive control using tubes. Automatica 40, 125–133 (2004)
25. Lazar, M., Heemels, W.P.M.H., Teel, A.R.: Subtleties in robust stability of discrete-time piecewise affine systems. In: Proceedings of the IEEE Conference on Decision and Control (2007)
26. Lazar, M., Muñoz de la Peña, D., Heemels, W.P.M.H., Alamo, T.: On input-to-state stability of min-max nonlinear model predictive control. Systems & Control Letters 57, 39–48 (2008)
27. Limon, D.: Predictive control of constrained nonlinear systems: stability and robustness. PhD thesis, Universidad de Sevilla (2002) (in spanish)
28. Limon, D., Alamo, T., Camacho, E.F.: Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties. In: 41st IEEE CDC, pp. 4619–4624 (2002)
29. Limon, D., Alamo, T., Camacho, E.F.: Stability analysis of systems with bounded additive uncertainties based on invariant sets: Stability and feasibility of MPC. In: ACC 2002, pp. 364–369 (2002)
30. Limon, D., Alamo, T., Salas, F., Camacho, E.F.: Input to state stability of min-max MPC controllers for nonlinear systems with bounded uncertainties. Automatica 42, 797–803 (2006)

31. Limon, D., Alamo, T., Salas, F., Camacho, E.F.: On the stability of constrained MPC without terminal constraint. IEEE Transactions on Automatic Control 51, 832–836 (2006)
32. Limon, D., Bravo, J.M., Alamo, T., Camacho, E.F.: Robust MPC of constrained nonlinear systems based on interval arithmetic. IEE Proceedings-Control Theory and Applications 152, 325 (2005)
33. Limon, D., Raimondo, D.M., Picasso, B.: On the input-to-state stability of discontinuous discrete-time systems. Internal Report. Dpt. Ing. Sistemas y Automática. U. de Sevilla (2009)
34. Maciejowski, J.M.: Predictive Control: With Constraints. Prentice-Hall, Englewood Cliffs (2002)
35. Magni, L., de Nicolao, G., Scattolini, R., Allgöwer, F.: Robust model predictive control for nonlinear discrete-time systems. Int. J. Robust and Nonlinear Control 13(3-4), 229–246 (2003)
36. Magni, L., Nijmeijer, H., van der Schaft, A.J.: A receding-horizon approach to the nonlinear H_∞ control problem. Automatica 37, 429–435 (2001)
37. Magni, L., Raimondo, D.M., Scattolini, R.: Regional input-to-state stability for nonlinear model predictive control. IEEE Transactions on Automatic Control 51(9), 1548–1553 (2006)
38. Magni, L., Scattolini, R.: Control design for nonlinear systems: trading robustness and performance with the model predictive control approach. IEE Proceedings - Control Theory and Applications 152, 333–339 (2005)
39. Magni, L., Scattolini, R.: Robustness and robust design of MPC for nonlinear discrete-time systems. In: Findeisen, R., et al. (eds.) Assessment and Future Directions. LNCIS, vol. 358, pp. 239–254. Springer, Heidelberg (2007)
40. Magni, L., Sepulchre, R.: Stability margins of nonlinear receding horizon control via inverse optimality. Systems & Control Letters 32, 241–245 (1997)
41. Mayne, D.Q.: Control of constrained dynamic systems. European Journal of Control 7, 87–99 (2001)
42. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: Stability and optimality. Automatica 36, 789–814 (2000)
43. Mayne, D.Q., Kerrigan, E.C.: Tube-based robust nonlinear model predictive control. In: Proceedings of the IFAC Symp. on Nonlinear Control Systems (2007)
44. Mayne, D.Q., Seron, M.M., Rakovic, S.V.: Robust model predictive control of constrained linear systems with bounded disturbances. Automatica 41, 219–224 (2005)
45. Meadows, E.S., Henson, M.A., Eaton, J.W., Rawlings, J.B.: Receding horizon control and discontinuous state feedback stabilization. Int. J. Control 62, 1217–1229 (1995)
46. Messina, M.J., Tuna, S.E., Teel, A.R.: Discrete-time certainty equivalence output feedback: allowing discontinuous control laws including those from model predictive control. Automatica 41, 617–628 (2005)
47. Michalska, H., Mayne, D.Q.: Robust receding horizon control of constrained nonlinear systems. IEEE Transactions on Automatic Control 38, 1623–1633 (1993)
48. Muñoz de la Peña, D., Alamo, T., Bemporad, A., Camacho, E.F.: A decomposition algorithm for feedback min-max model predictive control. IEEE Transactions on Automatic Control 51, 688–1692 (2006)
49. Muñoz de la Peña, D., Alamo, T., Bemporad, A., Camacho, E.F.: Explicit solution of min-max mpc with additive uncertainties and quadratic criterion. Systems & Control Letters 55, 266–274 (2006)

50. Nesic, D., Teel, A.R., Kokotovic, P.V.: Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete-time approximations. Systems & Control Letters 38, 259–270 (1999)
51. Qin, S.J., Badgwell, T.A.: A survey of industrial model predictive control technology. Control Engineering Practice 11(7), 733–764 (2003)
52. Raimondo, D.M., Alamo, T., Limon, D., Camacho, E.F.: Towards the practical implementation of min-max nonlinear Model Predictive Control. In: 46th IEEE CDC, pp. 1257–1262 (2007)
53. Raimondo, D.M., Limon, D., Lazar, M., Magni, L., Camacho, E.F.: Min-max model predictive control of nonlinear systems: A unifying overview on stability. Europen Journal of Control 1 (2009)
54. Rakovic, S.V.: Robust Control of Constrained Discrete Time Systems: Characterization and Implementation. PhD thesis, Imperial College of Science, Technology and Medicine. Univ. of London (2005)
55. Rakovic, S.V., Teel, A.R., Mayne, D.Q., Astolfi, A.: Simple robust control invariant tubes for some classes of nonlinear discrete time systems. In: Proceedings of the IEEE Conference on Decision and Control (2006)
56. Santos, L.O., Biegler, L.T., Castro, J.A.A.M.: A tool to analyze robust stability for constrained nonlinear mpc. Journal of Process Control 18, 383–390 (2008)
57. Scokaert, P.O.M., Mayne, D.Q.: Min-max feedback model predictive control for constrained linear systems. IEEE Transactions on Automatic Control 43, 1136–1142 (1998)
58. Scokaert, P.O.M., Mayne, D.Q., Rawlings, J.B.: Suboptimal model predictive control (feasibility implies stability). IEEE Transactions on Automatic Control AC-44, 648–654 (1999)
59. Scokaert, P.O.M., Rawlings, J.B., Meadows, E.S.: Discrete-time stability with perturbations: Application to model predictive control. Automatica 33, 463–470 (1997)
60. Sontag, E.D., Wang, Y.: New characterizations of input-to state-stability. IEEE Transactions on Automatic Control 41, 1283–1294 (1996)
61. Vidyasagar, M.: Nonlinear System Theory, 2nd edn. Prentice-Hall, Englewood Cliffs (1993)

Self-optimizing Robust Nonlinear Model Predictive Control

M. Lazar*, W.P.M.H. Heemels, and A. Jokic

Abstract. This paper presents a novel method for designing robust MPC schemes that are self-optimizing in terms of disturbance attenuation. The method employs convex control Lyapunov functions and disturbance bounds to optimize robustness of the closed-loop system on-line, at each sampling instant - a unique feature in MPC. Moreover, the proposed MPC algorithm is computationally efficient for nonlinear systems that are affine in the control input and it allows for a decentralized implementation.

Keywords: nonlinear systems, robust model predictive control (MPC), input-to-state stability (ISS), decentralized control.

1 Introduction

Robustness of nonlinear model predictive controllers has been one of the most relevant and challenging problems within MPC, see, e.g., [1, 2, 3, 4, 5]. From a conceptual point of view, three main categories of robust nonlinear MPC schemes can be identified, each with its pros and cons: inherently robust, tightened constraints and min-max MPC schemes, respectively. In all these approaches, the input-to-state stability property [6] has been employed as a theoretical tool for characterizing robustness, or robust stability[1].

M. Lazar and A. Jokic
Eindhoven University of Technology, Dept. of Electrical Eng.,
e-mail: m.lazar@tue.nl

W.P.M.H. Heemels
Eindhoven University of Technology Dept. of Mechanical Eng.
P.O. Box 513, 5600 MB, Eindhoven, The Netherlands

* Corresponding author.

[1] Other characterizations of robustness used in MPC, such as ultimate boundedness or stability of a robustly positively invariant set, can be recovered as a particular case of ISS or shown to be related.

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 27–40.
springerlink.com

The goal of the existing design methods for synthesizing control laws that achieve ISS [7, 8, 9] is to a priori guarantee a *predetermined* closed-loop ISS gain. Consequently, the ISS property, with a predetermined, constant ISS gain, is in this way enforced for all state space trajectories of the closed-loop system and at all time instances. As the existing approaches, which are also employed in the design of MPC schemes that achieve ISS, can lead to overly conservative solutions along particular trajectories, it is of high interest to develop a control (MPC) design method with the explicit goal of adapting the closed-loop ISS gain depending of the evolution of the state trajectory.

In this article we present a novel method for synthesizing robust MPC schemes with this feature. The method employs convex control Lyapunov functions (CLFs) and disturbance bounds to embed the standard ISS conditions of [8] using a finite number of inequalities. This leads to a finite dimensional optimization problem that has to be solved on-line, in a receding horizon fashion. The proposed inequalities govern the evolution of the closed-loop state trajectory through the sublevel sets of the CLF. The unique feature of the proposed robust MPC scheme is to allow for the *simultaneous* on-line (i) computation of a control action that achieves ISS and (ii) minimization of the closed-loop ISS gain depending of an actual state trajectory. As a result, the developed nonlinear MPC scheme is self-optimizing in terms of disturbance attenuation. From the computational point of view, following a particular design recipe, the self-optimizing robust MPC algorithm can be implemented as *a single linear program* for discrete-time nonlinear systems that are affine in the control variable and the disturbance input. Furthermore, we demonstrate that the freedom to optimize the closed-loop ISS gain on-line makes self-optimizing robust MPC suitable for decentralized control of networks of nonlinear systems.

2 Preliminary Definitions and Results

2.1 Basic Notions and Definitions

Let $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{Z}$ and $\mathbb{Z}_+$ denote the field of real numbers, the set of non-negative reals, the set of integer numbers and the set of non-negative integers, respectively. We use the notation $\mathbb{Z}_{\geq c_1}$ and $\mathbb{Z}_{(c_1,c_2]}$ to denote the sets $\{k \in \mathbb{Z}_+ \mid k \geq c_1\}$ and $\{k \in \mathbb{Z}_+ \mid c_1 < k \leq c_2\}$, respectively, for some $c_1, c_2 \in \mathbb{Z}_+$. For a set $\mathscr{S} \subseteq \mathbb{R}^n$, we denote by $\text{int}(\mathscr{S})$ the interior of $\mathscr{S}$. For two arbitrary sets $\mathscr{S} \subseteq \mathbb{R}^n$ and $\mathscr{P} \subseteq \mathbb{R}^n$, let $\mathscr{S} \sim \mathscr{P} := \{x \in \mathbb{R}^n \mid x + \mathscr{P} \subseteq \mathscr{S}\}$ denote their Pontryagin difference. A polyhedron (or a polyhedral set) in $\mathbb{R}^n$ is a set obtained as the intersection of a finite number of open and/or closed half-spaces. The Hölder p-norm of a vector $x \in \mathbb{R}^n$ is defined as $\|x\|_p := (|[x]_1|^p + \ldots + |[x]_n|^p)^{\frac{1}{p}}$ for $p \in \mathbb{Z}_{[1,\infty)}$ and $\|x\|_\infty := \max_{i=1,\ldots,n} |[x]_i|$, where $[x]_i$, $i = 1, \ldots, n$, is the i-th component of x and $|\cdot|$ is the absolute value. For a matrix $M \in \mathbb{R}^{m \times n}$, let $\|M\|_p := \sup_{x \neq 0} \frac{\|Mx\|_p}{\|x\|_p}$ denote its corresponding induced matrix norm. Then $\|M\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^{n} |[M]_{ij}|$, where $[M]_{ij}$ is the ij-th entry of M. Let $\mathbf{z} := \{z(l)\}_{l \in \mathbb{Z}_+}$ with $z(l) \in \mathbb{R}^o$ for all $l \in \mathbb{Z}_+$ denote an arbitrary sequence. Define $\|\mathbf{z}\| := \sup\{\|z(l)\| \mid l \in \mathbb{Z}_+\}$, where $\|\cdot\|$ denotes an arbitrary p-norm, and

$\mathbf{z}_{[k]} := \{z(l)\}_{l\in\mathbb{Z}_{[0,k]}}$. A function $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ belongs to class $\mathcal{K}$ if it is continuous, strictly increasing and $\varphi(0) = 0$. A function $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ belongs to class $\mathcal{K}_\infty$ if $\varphi \in \mathcal{K}$ and $\lim_{s\to\infty} \varphi(s) = \infty$. A function $\beta : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ belongs to class $\mathcal{KL}$ if for each fixed $k \in \mathbb{R}_+$, $\beta(\cdot,k) \in \mathcal{K}$ and for each fixed $s \in \mathbb{R}_+$, $\beta(s,\cdot)$ is decreasing and $\lim_{k\to\infty} \beta(s,k) = 0$.

2.2 ISS Definitions and Results

Consider the discrete-time nonlinear system

$$x(k+1) \in \Phi(x(k), w(k)), \quad k \in \mathbb{Z}_+, \tag{1}$$

where $x(k) \in \mathbb{R}^n$ is the state and $w(k) \in \mathbb{R}^l$ is an unknown disturbance input at the discrete-time instant k. The mapping $\Phi : \mathbb{R}^n \times \mathbb{R}^l \hookrightarrow \mathbb{R}^n$ is an arbitrary nonlinear set-valued function. We assume that $\Phi(0,0) = \{0\}$. Let $\mathbb{W}$ be a subset of $\mathbb{R}^l$.

Definition 1. *We call a set $\mathcal{P} \subseteq \mathbb{R}^n$* robustly positively invariant (RPI) *for system (1) with respect to $\mathbb{W}$ if for all $x \in \mathcal{P}$ it holds that $\Phi(x,w) \subseteq \mathcal{P}$ for all $w \in \mathbb{W}$.*

Definition 2. *Let $\mathbb{X}$ with $0 \in \mathrm{int}(\mathbb{X})$ and $\mathbb{W}$ be subsets of $\mathbb{R}^n$ and $\mathbb{R}^l$, respectively. We call system (1)* ISS($\mathbb{X}$, $\mathbb{W}$) *if there exist a $\mathcal{KL}$-function $\beta(\cdot,\cdot)$ and a $\mathcal{K}$-function $\gamma(\cdot)$ such that, for each $x(0) \in \mathbb{X}$ and all $\mathbf{w} = \{w(l)\}_{l\in\mathbb{Z}_+}$ with $w(l) \in \mathbb{W}$ for all $l \in \mathbb{Z}_+$, it holds that all corresponding state trajectories of (1) satisfy $\|x(k)\| \le \beta(\|x(0)\|,k) + \gamma(\|\mathbf{w}_{[k-1]}\|)$, $\forall k \in \mathbb{Z}_{\ge 1}$. We call the function $\gamma(\cdot)$ an ISS gain of system (1).*

Theorem 1. *Let $\mathbb{W}$ be a subset of $\mathbb{R}^l$ and let $\mathbb{X} \subseteq \mathbb{R}^n$ be a RPI set for (1) with respect to $\mathbb{W}$, with $0 \in \mathrm{int}(\mathbb{X})$. Furthermore, let $\alpha_1(s) := as^\delta$, $\alpha_2(s) := bs^\delta$, $\alpha_3(s) := cs^\delta$ for some $a,b,c,\delta \in \mathbb{R}_{>0}$, $\sigma \in \mathcal{K}$ and let $V : \mathbb{R}^n \to \mathbb{R}_+$ be a function such that:*

$$\alpha_1(\|x\|) \le V(x) \le \alpha_2(\|x\|), \tag{2a}$$

$$V(x^+) - V(x) \le -\alpha_3(\|x\|) + \sigma(\|w\|) \tag{2b}$$

for all $x \in \mathbb{X}$, $w \in \mathbb{W}$ and all $x^+ \in \Phi(x,w)$. Then the system (1) is ISS($\mathbb{X}$,$\mathbb{W}$) with

$$\beta(s,k) := \alpha_1^{-1}(2\rho^k\alpha_2(s)),\ \gamma(s) := \alpha_1^{-1}\left(\frac{2\sigma(s)}{1-\rho}\right),\ \rho := 1 - \frac{c}{b} \in [0,1). \tag{3}$$

If inequality (2b) holds for $w = 0$, then the 0-input system $x(k+1) \in \Phi(x(k),0)$, $k \in \mathbb{Z}_+$, is asymptotically stable in $\mathbb{X}$.

The proof of Theorem 1 is similar in nature to the proof given in [8, 10, 11] by replacing the difference *equation* with the difference *inclusion* as in (1).

2.3 Inherent ISS through Continuous and Convex Control Lyapunov Functions

Consider the discrete-time constrained nonlinear system

$$x(k+1) = \phi(x(k),u(k),w(k)) := f(x(k),u(k)) + g(x(k))w(k),\ k \in \mathbb{Z}_+, \quad (4)$$

where $x(k) \in \mathbb{X} \subseteq \mathbb{R}^n$ is the state, $u(k) \in \mathbb{U} \subseteq \mathbb{R}^m$ is the control action and $w(k) \in \mathbb{W} \subset \mathbb{R}^l$ is an unknown disturbance input at the discrete-time instant k. $\phi : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^l \to \mathbb{R}^n$, $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times l}$ are arbitrary nonlinear functions with $\phi(0,0,0) = 0$ and $f(0,0) = 0$. Note that we allow that $g(0) \neq 0$. We assume that $0 \in \text{int}(\mathbb{X})$, $0 \in \text{int}(\mathbb{U})$ and $\mathbb{W}$ is bounded. We also assume that $\phi(\cdot,\cdot,\cdot)$ is bounded in $\mathbb{X}$. Next, let $\alpha_1, \alpha_2, \alpha_3 \in \mathscr{K}_\infty$ and let $\sigma \in \mathscr{K}$.

Definition 3. *A function $V : \mathbb{R}^n \to \mathbb{R}_+$ that satisfies (2a) for all $x \in \mathbb{X}$ is called a control Lyapunov function (CLF) for system $x(k+1) = \phi(x(k),u(k),0)$, $k \in \mathbb{Z}_+$, if for all $x \in \mathbb{X}$, $\exists u \in \mathbb{U}$ such that $V(\phi(x,u,0)) - V(x) \leq -\alpha_3(\|x\|)$.*

Problem 1. Let a CLF $V(\cdot)$ be given. At time $k \in \mathbb{Z}_+$ measure the state $x(k)$ and calculate a control action $u(k)$ that satisfies:

$$u(k) \in \mathbb{U},\ \phi(x(k),u(k),0) \in \mathbb{X}, \quad (5a)$$
$$V(\phi(x(k),u(k),0)) - V(x(k)) + \alpha_3(\|x(k)\|) \leq 0. \quad (5b)$$

Let $\pi_0(x(k)) := \{u(k) \in \mathbb{R}^m \mid (5) \text{ holds}\}$. Let $x(k+1) \in \phi_0(x(k),\pi_0(x(k))) := \{f(x(k),u) \mid u \in \pi_0(x(k))\}$ denote the difference inclusion corresponding to the 0-input system (4) in "closed-loop" with the set of feasible solutions obtained by solving Problem 1 at each instant $k \in \mathbb{Z}_+$.

Theorem 2. *Let $\alpha_1, \alpha_2, \alpha_3 \in \mathscr{K}_\infty$ of the form specified in Theorem 1 and a corresponding CLF $V(\cdot)$ be given. Suppose that Problem 1 is feasible for all states x in $\mathbb{X}$. Then:* (i) *The difference inclusion*

$$x(k+1) \in \phi_0(x(k),\pi_0(x(k))), \quad k \in \mathbb{Z}_+, \quad (6)$$

is asymptotically stable in $\mathbb{X}$; (ii) *Consider a perturbed version of (6), i.e.*

$$\tilde{x}(k+1) \in \phi_0(\tilde{x}(k),\pi_0(\tilde{x}(k))) + g(\tilde{x}(k))w(k), \quad k \in \mathbb{Z}_+ \quad (7)$$

and let $\widetilde{\mathbb{X}} \subseteq \mathbb{X}$ be a RPI set for (7) with respect to $\mathbb{W}$. If $\mathbb{X}$ is compact, the CLF $V(\cdot)$ is convex and continuous[2] on $\mathbb{X}$ and $\exists M \in \mathbb{R}_{>0}$ such that $\|g(x)\| \leq M$ for all $x \in \mathbb{X}$, then system (7) is ISS($\widetilde{\mathbb{X}}$,$\mathbb{W}$).

Proof: (i) Let $x(k) \in \mathbb{X}$ for some $k \in \mathbb{Z}_+$. Then, feasibility of Problem 1 ensures that $x(k+1) \in \phi_0(x(k),\pi_0(x(k))) \subseteq \mathbb{X}$ due to constraint (5a). Hence, Problem 1 remains feasible and thus, $\mathbb{X}$ is a PI set for system (6). The result then follows directly from Theorem 1. (ii) By convexity and continuity of $V(\cdot)$ and compactness of $\mathbb{X}$, $V(\cdot)$ is Lipschitz continuous on $\mathbb{X}$ [12]. Hence, letting $\mathscr{L} \in \mathbb{R}_{>0}$ denote a Lipschitz constant of $V(\cdot)$ in $\mathbb{X}$, one obtains $|V(\phi(x,u,w)) - V(\phi(x,u,0))| = |V(f(x,u) + g(x)w) - V(f(x,u))| \leq \mathscr{L}M\|w\|$ for all $x \in \mathbb{X}$ and all w. From this property, together with inequality (5b) we have that inequality (2b) holds with $\sigma(s) := \mathscr{L}Ms \in \mathscr{K}$. Since

[2] Continuity of $V(\cdot)$ alone is sufficient, but it requires a somewhat more complex proof.

$\widetilde{\mathbb{X}}$ is an RPI set for (7) by the hypothesis, ISS($\widetilde{\mathbb{X}}$,$\mathbb{W}$) of the difference inclusion (7) follows from Theorem 1. □

3 Problem Definition

Theorem 2 establishes that all feasible solutions of Problem 1 are stabilizing feedback laws which, under additional assumptions even achieve ISS. However, this inherent ISS property of a feedback law calculated by solving Problem 1 relies on a fixed, possibly large gain of $\sigma(\cdot)$, which depends on $V(\cdot)$. This gain is explicitly related to the ISS gain of the closed-loop system via (3). To optimize disturbance attenuation for the closed-loop system, at each time instant $k \in \mathbb{Z}_+$ and for a given $x(k) \in \mathbb{X}$, it would be desirable to *simultaneously* compute a control action $u(k) \in \mathbb{U}$ that satisfies:

$$\text{(i)} \quad V(\phi(x(k),u(k),w(k))) - V(x(k)) + \alpha_3(\|x\|) - \sigma(\|w(k)\|) \leq 0,\ \forall w(k) \in \mathbb{W} \tag{8}$$

and some function $\sigma(s) := \eta(k)s^{\delta}$ and (ii) minimize $\eta(k)$ ($\eta(k), \delta \in \mathbb{R}_{>0}, \forall k \in \mathbb{Z}_+$).

Remark 1. It is not possible to directly include (8) in Problem 1, as it leads to an infinite dimensional optimization problem. If $\mathbb{W}$ is a compact polyhedron, a possibility to resolve this issue would be to evaluate the inequality (8) only for $w(k)$ taking values in the set of vertices of $\mathbb{W}$. However, this does not guarantee that (8) holds for all $w(k) \in \mathbb{W}$ due to the fact that the left-hand term in (8) is not necessarily a convex function of $w(k)$, i.e. it contains the difference of two, possibly convex, functions of $w(k)$. This makes the considered problem challenging and interesting. □

4 Main Results

In what follows we present a solution to the problem stated in Section 3. More specifically, we demonstrate that by considering continuous and convex CLFs and compact polyhedral sets $\mathbb{X}, \mathbb{U}, \mathbb{W}$ (that contain the origin in their interior) a solution to inequality (8) can be obtained via a finite set of inequalities that only depend on the vertices of $\mathbb{W}$. The standing assumption throughout the remainder of the article is that the considered system, i.e. (4), is affine in the disturbance input w.

4.1 Optimized ISS through Continuous and Convex CLFs

Let w^e, $e = 1,...,E$, be the vertices of $\mathbb{W}$. Next, consider a finite set of simplices $S_1,\ldots,S_M$ with each simplex S_i equal to the convex hull of a subset of the vertices of $\mathbb{W}$ and the origin, and such that $\cup_{i=1}^{M} S_i = \mathbb{W}$. More precisely, $S_i = \text{Co}\{0, w^{e_{i,1}},\ldots,w^{e_{i,l}}\}$ and $\{w^{e_{i,1}},\ldots,w^{e_{i,l}}\} \subset \{w^1,\ldots,w^E\}$ (i.e. $\{e_{i,1},\ldots,e_{i,l}\} \subset \{1,\ldots,E\}$) with $w^{e_{i,1}},\ldots,w^{e_{i,l}}$ linearly independent. For each simplex S_i we define

the matrix $W_i := [w^{e_{i,1}} \ \dots \ w^{e_{i,l}}] \in \mathbb{R}^{l\times l}$, which is invertible. Let $\lambda_e(k)$, $k \in \mathbb{Z}_+$, be optimization variables associated with each vertex w^e. Let $\alpha_3 \in \mathscr{K}_\infty$, suppose that $x(k)$ at time $k \in \mathbb{Z}_+$ is given and consider the following set of inequalities depending on $u(k)$ and $\lambda_1(k), \dots, \lambda_E(k)$:

$$V(\phi(x(k),u(k),0)) - V(x(k)) + \alpha_3(\|x(k)\|) \le 0, \tag{9a}$$

$$V(\phi(x(k),u(k),w^e)) - V(x(k)) + \alpha_3(\|x(k)\|) - \lambda_e(k) \le 0, \ \forall e = \overline{1,E}. \tag{9b}$$

Theorem 3. *Let $V(\cdot)$ be a convex CLF. If for $\alpha_3 \in \mathscr{K}_\infty$ and $x(k)$ at time $k \in \mathbb{Z}_+$ there exist $u(k)$ and $\lambda_e(k)$, $e = 1, \dots, E$, such that (9a) and (9b) hold, then (8) holds for the same $u(k)$, with $\sigma(s) := \eta(k)s$ and*

$$\eta(k) := \max_{i=1,\dots,M} \|\bar{\lambda}_i(k) W_i^{-1}\|, \tag{10}$$

where $\bar{\lambda}_i(k) := [\lambda_{e_{i,1}}(k) \ \dots \ \lambda_{e_{i,l}}(k)] \in \mathbb{R}^{1\times l}$.

Proof: Let $\alpha_3 \in \mathscr{K}_\infty$ and $x(k)$ be given and suppose (9b) holds for some $\lambda_e(k)$, $e = 1, \dots, E$. Let $w \in \mathbb{W} = \bigcup_{i=1}^M S_i$. Hence, there exists an i such that $w \in S_i = \mathrm{Co}\{0, w^{e_{i,1}}, \dots, w^{e_{i,l}}\}$, which means that there exist non-negative numbers $\mu_0, \mu_1, \dots, \mu_l$ with $\sum_{j=0}^l \mu_j = 1$ such that $w = \sum_{j=1}^l \mu_j w^{e_{i,j}} + \mu_0 0 = \sum_{j=1}^l \mu_j w^{e_{i,j}}$. In matrix notation we have that $w = W_i[\mu_1 \ \dots \ \mu_l]^\top$ and thus $[\mu_1 \ \dots \ \mu_l]^\top = W_i^{-1} w$. Multiplying each inequality in (9b) corresponding to the index $e_{i,j}$ and the inequality (9a) with $\mu_j \ge 0$, $j = 0, 1, \dots, l$, summing up and using $\sum_{j=0}^l \mu_j = 1$ yield:

$$\mu_0 V(\phi(x(k),u(k),0)) + \sum_{j=1}^l \mu_j V(\phi(x(k),u(k),w^{e_{i,j}})) - V(x(k)) + \alpha_3(\|x(k)\|) - \sum_{j=1}^l \mu_j \lambda_{e_{i,j}}(k) \le 0.$$

Furthermore, using $\phi(x(k),u(k),w^{e_{i,j}}) = f(x(k),u(k)) + g(x(k))w^{e_{i,j}}$, convexity of $V(\cdot)$ and $\sum_{j=0}^l \mu_j = 1$ yields

$$V(\phi(x(k),u(k),\sum_{j=1}^l \mu_j w^{e_{i,j}})) - V(x(k)) + \alpha_3(\|x(k)\|) - \sum_{j=1}^l \mu_j \lambda_{e_{i,j}}(k) \le 0,$$

or equivalently

$$V(\phi(x(k),u(k),w)) - V(x(k)) + \alpha_3(\|x(k)\|) - \bar{\lambda}_i(k)[\mu_1 \ \dots \ \mu_l]^\top \le 0.$$

Using that $[\mu_1 \ \dots \ \mu_l]^\top = W_i^{-1} w$ we obtain (8) with $w(k) = w$ for $\sigma(s) = \eta(k)s$ and $\eta(k) \ge 0$ as in (10). □

4.2 Self-optimizing Robust Nonlinear MPC

For any $x \in \mathbb{X}$ let $\mathbb{W}_x := \{g(x)w \mid w \in \mathbb{W}\} \subset \mathbb{R}^n$ (note that $0 \in \mathbb{W}_x$) and assume that $\mathbb{X} \sim \mathbb{W}_x \neq \emptyset$. Let $\bar{\lambda} := [\lambda_1, \ldots, \lambda_E]^\top$ and let $J(\bar{\lambda}) : \mathbb{R}^E \to \mathbb{R}_+$ be a function that satisfies $\alpha_4(\|\bar{\lambda}\|) \leq J(\bar{\lambda}) \leq \alpha_5(\|\bar{\lambda}\|)$ for some $\alpha_4, \alpha_5 \in \mathscr{K}_\infty$; for example, $J(\bar{\lambda}) := \max_{i=1,\ldots,M} \|\bar{\lambda}_i W_i^{-1}\|$.

Problem 2. Let $\alpha_3 \in \mathscr{K}_\infty$, $J(\cdot)$ and a CLF $V(\cdot)$ be given. At time $k \in \mathbb{Z}_+$ measure the state $x(k)$ and minimize the cost $J(\lambda_1(k), \ldots, \lambda_E(k))$ over $u(k), \lambda_1(k), \ldots, \lambda_E(k)$, subject to the constraints

$$u(k) \in \mathbb{U},\ \lambda_e(k) \geq 0,\ f(x(k), u(k)) \in \mathbb{X} \sim \mathbb{W}_{x(k)}, \tag{11a}$$

$$V(\phi(x(k), u(k), 0)) - V(x(k)) + \alpha_3(\|x(k)\|) \leq 0, \tag{11b}$$

$$V(\phi(x(k), u(k), w^e)) - V(x(k)) + \alpha_3(\|x(k)\|) - \lambda_e(k) \leq 0,\ \forall e = \overline{1, E}. \tag{11c}$$

□

Let $\pi(x(k)) := \{u(k) \in \mathbb{R}^m \mid (11) \text{ holds}\}$ and let

$$x(k+1) \in \phi_{\text{cl}}(x(k), \pi(x(k)), w(k)) := \{\phi(x(k), u, w(k)) \mid u \in \pi(x(k))\}$$

denote the difference inclusion corresponding to system (4) in "closed-loop" with the set of feasible solutions obtained by solving Problem 2 at each $k \in \mathbb{Z}_+$.

Theorem 4. *Let $\alpha_1, \alpha_2, \alpha_3 \in \mathscr{K}_\infty$ of the form specified in Theorem 7, a continuous and convex CLF $V(\cdot)$ and a cost $J(\cdot)$ be given. Suppose that Problem 2 is feasible for all states x in $\mathbb{X}$. Then the difference inclusion*

$$x(k+1) \in \phi_{cl}(x(k), \pi(x(k)), w(k)), \quad k \in \mathbb{Z}_+ \tag{12}$$

is ISS($\mathbb{X}$, $\mathbb{W}$).

Proof: Let $x(k) \in \mathbb{X}$ for some $k \in \mathbb{Z}_+$. Then, feasibility of Problem 2 ensures that $x(k+1) \in \phi_{\text{cl}}(x(k), \pi(x(k)), w(k)) \subseteq \mathbb{X}$ for all $w(k) \in \mathbb{W}$, due to $g(x(k))w(k) \in \mathbb{W}_{x(k)}$ and constraint (11a). Hence, Problem 2 remains feasible and thus, $\mathbb{X}$ is a RPI set with respect to $\mathbb{W}$ for system (12). From Theorem 3 we also have that $V(\cdot)$ satisfies (2b) with $\sigma(s) := \eta(k)s$ and $\eta(k)$ as in (10). Let

$$\lambda^* := \sup_{x \in \mathbb{X}, u \in \mathbb{U}, e=1,\ldots,E} \{V(\phi(x, u, w^e)) - V(x) + \alpha_3(\|x\|)\}.$$

Due to continuity of $V(\cdot)$, compactness of $\mathbb{X}$, $\mathbb{U}$ and boundedness of $\phi(\cdot,\cdot,\cdot)$, λ^* exists and is finite (the sup above is a max if $\phi(\cdot,\cdot,\cdot)$ is continuous in x and u). Hence, inequality (11c) is always satisfied for $\lambda_e(k) = \lambda^*$ for all $e = 1, \ldots, E$, $k \in \mathbb{Z}_+$, and for all $x \in \mathbb{X}$, $u \in \mathbb{U}$. This in turn, via (10) ensures the existence of a $\eta^* \in \mathbb{R}_{>0}$ such that $\eta(k) \leq \eta^*$ for all $k \in \mathbb{Z}_+$. Hence, we proved that inequality (8) holds for all $x \in \mathbb{X}$ and all $w \in \mathbb{W}$. Then, since $\mathbb{X}$ is RPI, ISS($\mathbb{X}$,$\mathbb{W}$) follows directly from Theorem 1. □

Remark 2. An alternative proof to Theorem 4 can be obtained by simply applying the reasoning used in the proof of Theorem 2. Hence, inherent ISS can be established directly from constraint (11b). Also, notice that in the proof of Theorem 4 we used a worst case evaluation of $\lambda_e(k)$ to prove ISS. However, it is important to observe that compared to Problem 1, nothing is lost in terms of feasibility, while Problem 2, *although it inherently guarantees a constant ISS gain, it provides freedom to optimize the ISS gain of the closed-loop system, by minimizing the variables* $\lambda_1(k),\ldots,\lambda_E(k)$ *via the cost* $J(\cdot)$. As such, in reality the gain $\eta(k)$ of the function $\sigma(\cdot)$ can be much smaller for $k \geq k_0$, for some $k_0 \in \mathbb{Z}_+$, depending on the state trajectory $x(k)$. □

In Theorem 4 we assumed for simplicity that Problem 2 is feasible for all $x \in \mathbb{X}$; in other words, *feasibility implies ISS*. Whenever Problem 2 can be solved explicitly (see the implementation paragraph below), it is possible to calculate the maximal RPI set for the closed-loop dynamics that is contained within the explicit set of feasible solutions. Alternatively, we establish next an easily verifiable sufficient condition under which any sublevel set of $V(\cdot)$ contained in $\mathbb{X}$ is a RPI subset of the set of feasible solutions of Problem 2.

Lemma 1. *Given a CLF* $V(\cdot)$ *that satisfies the hypothesis of Theorem 4, let* $\mathscr{V}_\Delta := \{x \in \mathbb{R}^n \mid V(x) \leq \Delta\}$. *Then, for any* $\Delta \in \mathbb{R}_{>0}$ *such that* $\mathscr{V}_\Delta \subseteq \mathbb{X}$, *if* $\lambda^* \leq (1-\rho)\Delta$, *with* ρ *as defined in (3), Problem 2 is feasible for all* $x \in \mathscr{V}_\Delta$ *and remains feasible for all resulting closed-loop trajectories that start in* $\mathscr{V}_\Delta$.

Proof: From the proof of Theorem 4 we know that inequalities (11c) are feasible for all $x(k) \in \mathbb{X}$, $u(k) \in \mathbb{U}$ and $e = \overline{1,E}$ by taking $\lambda(k) = \lambda^*$ for all $k \in \mathbb{Z}_+$. Thus, for any $x(k) \in \mathscr{V}_\Delta \subseteq \mathbb{X}$, $\Delta \in \mathbb{R}_{\geq 0}$, we have that:

$$\begin{aligned} V(\phi(x(k),u(k),w(k))) &\leq V(x(k)) - \alpha_3(\|x(k)\|) + \lambda^* \leq \rho V(x(k)) + \lambda^* \\ &\leq \rho\Delta + \lambda^* \leq \rho\Delta + (1-\rho)\Delta = \Delta, \end{aligned}$$

which yields $\phi(x(k),u(k),w(k)) \in \mathscr{V}_\Delta \subseteq \mathbb{X}$. This in turn ensures feasibility of (11a), while (11b) is feasible by definition of the CLF $V(\cdot)$, which concludes the proof. □

Remark 3. The result of Theorem 4 holds for all inputs $u(k)$ for which Problem 2 is feasible. To select on-line one particular control input from the set $\pi(x(k))$ and to improve closed-loop performance (in terms of settling time) it is useful to also penalize the state and the input. Let $F : \mathbb{R}^n \to \mathbb{R}_+$ and $L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_+$ with $F(0) = L(0,0) = 0$ be arbitrary nonlinear functions. For $N \in \mathbb{Z}_{\geq 1}$ let $\bar{\mathbf{u}}(k) := (\bar{u}(k),\bar{u}(k+1),\ldots,\bar{u}(k+N-1)) \in \mathbb{U}^N$ and $J_{\mathrm{RHC}}(x(k),\bar{\mathbf{u}}(k)) := F(\bar{x}(k+N)) + \sum_{i=0}^{N-1} L(\bar{x}(k+i),\bar{u}(k+i))$, where $\bar{x}(k+i+1) := f(\bar{x}(k+i),\bar{u}(k+i))$ for $i = \overline{0,N-1}$ and $\bar{x}(k) := x(k)$. Then one can add this cost to Problem 2, i.e. at time $k \in \mathbb{Z}_+$ measure the state $x(k)$ and minimize $J_{\mathrm{RHC}}(x(k),\bar{\mathbf{u}}(k)) + J(\lambda_1(k),\ldots,\lambda_E(k))$ over $\bar{\mathbf{u}}(k),\lambda_1(k),\ldots,\lambda_E(k)$, subject to constraints (11) and $\bar{x}(k+i) \in \mathbb{X}$, $i = \overline{2,N}$. Observe that the optimum needs not to be attained at each sampling instant to achieve ISS, which is appealing for practical reasons but also in the case of a possibly discontinuous value function. □

Remark 4. Besides enhancing robustness, the constraints (11b)-(11c) also ensure that Problem 2 recovers performance (in terms of settling time) when the state of the closed-loop system approaches the origin. Loosely speaking, when $x(k) \approx 0$, solving Problem 2 will produce a control action $u(k) \approx 0$ (because of constraint (11b) and the fact that the cost $J_{\mathrm{RHC}}(\cdot) + J(\cdot)$ is minimized). This yields $V(\phi(0,0,w^e)) - \lambda_e(k) \leq 0$, $e = \overline{1,E}$, due to constraint (11c). Thus, solving Problem 2 with the above cost will not optimize each variable $\lambda_e(k)$ below the corresponding value $V(\phi(0,0,w^e))$, $e = \overline{1,E}$, when the state reaches the equilibrium. This property is desirable, since it is known from min-max MPC [11] that considering a worst case disturbance scenario leads to poor performance when the real disturbance is small or vanishes. □

4.3 Decentralized Formulation

In this paragraph we give a brief outline of how the proposed self-optimizing MPC algorithm can be implemented in a decentralized fashion. We consider a connected directed graph $\mathscr{G} = (\mathscr{S}, \mathscr{E})$ with a finite number of vertices $\mathscr{S}$ and a set of directed edges $\mathscr{E} \subseteq \{(i,j) \in \mathscr{S} \times \mathscr{S} \mid i \neq j\}$. A dynamical system is assigned to each vertex $i \in \mathscr{S}$, with the dynamics governed by the following equation:

$$x_i(k+1) = \phi_i(x_i(k), u_i(k), v_i(x_{\mathscr{N}_i}(k)), w_i(k)), \;\; k \in \mathbb{Z}_+. \tag{13}$$

In (13), $x_i \in \mathbb{X}_i \subset \mathbb{R}^{n_i}$, $u_i \in \mathbb{U}_i \subset \mathbb{R}^{m_i}$ are the state and the control input of the i-th system, and $w_i \in \mathbb{W}_i \subset \mathbb{R}^{l_i}$ is an exogenous disturbance input that directly affects only the i-th system. With each directed edge $(j,i) \in \mathscr{E}$ we associate a function $v_{ij} : \mathbb{R}^{n_j} \to \mathbb{R}^{n_i}$, which defines the interconnection signal $v_{ij}(x_j(k)), k \in \mathbb{Z}_+$, between system j and system i, i.e. $v_{ij}(\cdot)$ characterizes how the states of system j influence the dynamics of system i. The set $\mathscr{N}_i := \{j \mid (j,i) \in \mathscr{E}\}$ denotes the set of direct neighbors (observe that $j \in \mathscr{N}_i \not\Rightarrow i \in \mathscr{N}_j$) of the system i. For simplicity of notation we use $x_{\mathscr{N}_i}(k)$ and $v_i(x_{\mathscr{N}_i}(k))$ to denote $\{x_j(k)\}_{j \in \mathscr{N}_i}$ and $\{v_{ij}(x_j(k))\}_{j \in \mathscr{N}_i}$, respectively. Both $\phi_i(\cdot,\cdot,\cdot,\cdot)$ and $v_{ij}(\cdot)$ are arbitrary nonlinear, possibly discontinuous functions that satisfy $\phi_i(0,0,0,0) = 0$, $v_{ij}(0) = 0$ for all $(i,j) \in \mathscr{S} \times \mathscr{N}_i$. For all $i \in \mathscr{S}$ we assume that $\mathbb{X}_i$, $\mathbb{U}_i$ and $\mathbb{W}_i$ are compact sets that contain the origin in their interior.

Assumption 1. *The value of all interconnection signals $v_{ij}(x_j(k))$ is known at all discrete-time instants $k \in \mathbb{Z}_+$ for any system $i \in \mathscr{S}$.*

From a technical point of view, Assumption 1 is satisfied, e.g., if all interconnection signals $v_{ij}(x_j(k))$ are directly measurable at all $k \in \mathbb{Z}_+$ or, if all directly neighboring systems $j \in \mathscr{N}_i$ are able to communicate their local measured state $x_j(k)$ to system $i \in \mathscr{S}$. Consider next the following decentralized version of Problem 2, where the notation and definitions employed so far are carried over *mutatis mutandis*.

Problem 3. For system $i \in \mathscr{S}$ let $\alpha_3^i \in \mathscr{K}_\infty$, $J_i(\cdot)$ and a CLF $V_i(\cdot)$ be given. At time $k \in \mathbb{Z}_+$ measure the local state $x_i(k)$ and the interconnection signals $v_i(x_{\mathscr{N}_i}(k))$ and

minimize the cost $J_i(\lambda_1^i(k),\ldots,\lambda_{E_i}^i(k))$ over $u_i(k),\lambda_1^i(k),\ldots,\lambda_{E_i}^i(k)$, subject to the constraints

$$u_i(k)\in\mathbb{U},\ \lambda_e^i(k)\geq 0,\ \phi_i(x_i(k),u_i(k),v_i(x_{\mathcal{N}_i}(k)),0)\in\mathbb{X}_i\sim\mathbb{W}_{x_i(k)}, \tag{14a}$$

$$V_i(\phi_i(x_i(k),u_i(k),v_i(x_{\mathcal{N}_i}(k)),0))-V_i(x_i(k))+\alpha_3^i(\|x_i(k)\|)\leq 0, \tag{14b}$$

$$V_i(\phi_i(x_i(k),u_i(k),v_i(x_{\mathcal{N}_i}(k)),w_i^e))-V_i(x_i(k))+\alpha_3^i(\|x_i(k)\|)-\lambda_e^i(k)\leq 0,$$
$$\forall e=\overline{1,E_i}. \tag{14c}$$

□

Let $\pi_i(x_i(k),v_i(x_{\mathcal{N}_i}(k)))$:= $\{u_i(k)\in\mathbb{R}^{m_i}\mid (14)\text{ holds}\}$ and let

$$\begin{aligned}x_i(k+1)\in&\phi_i^{\mathrm{cl}}(x_i(k),\pi_i(x_i(k),v_i(x_{\mathcal{N}_i}(k)),v_i(x_{\mathcal{N}_i}(k)),w_i(k))\\ &:=\{\phi_i(x_i(k),u,v_i(x_{\mathcal{N}_i}(k)),w_i(k))\mid u\in\pi_i(x_i(k),v_i(x_{\mathcal{N}_i}(k)))\}\end{aligned}$$

denote the difference inclusion corresponding to system (13) in "closed-loop" with the set of feasible solutions obtained by solving Problem 3 at each $k\in\mathbb{Z}_+$.

Theorem 5. *Let, $\alpha_1^i,\alpha_2^i,\alpha_3^i\in\mathcal{K}_\infty$ of the form specified in Theorem 1, continuous and convex CLFs $V_i(\cdot)$ and costs $J_i(\cdot)$ be given for all systems indexed by $i\in\mathscr{S}$. Suppose Assumption 1 holds and Problem 3 is feasible for each system $i\in\mathscr{S}$ and for all states x_i in $\mathbb{X}_i$ and all corresponding $v_i(x_{\mathcal{N}_i})$. Then the interconnected dynamically coupled nonlinear system described by the collection of difference inclusions*

$$x_i(k+1)\in\phi_i^{cl}(x_i(k),\pi_i(x_i(k),v_i(x_{\mathcal{N}_i}(k)),v_i(x_{\mathcal{N}_i}(k)),w_i(k)),\ i\in\mathscr{S},\ k\in\mathbb{Z}_+ \tag{15}$$

is ISS($\mathbb{X}_1\times\ldots\times\mathbb{X}_S$, $\mathbb{W}_1\times\ldots\times\mathbb{W}_S$).

The proof is omitted due to space limitations. Its central argument is that each continuous and convex CLF $V_i(x_i)$ is in fact Lipschitz continuous on $\mathbb{X}_i$ [12], which makes $\sum_{i\in\mathscr{S}}V_i(x_i)=:V(\{x_i\}_{i\in\mathscr{S}})$ a Lipschitz continuous CLF for the global interconnected system. The result then follows similarly to the proof of Theorem 2-(ii). Theorem 5 guarantees a constant ISS gain for the global closed-loop system, while the ISS gain of each closed-loop system $i\in\mathscr{S}$ can still be optimized on-line.

Remark 5. Problem 3 defines a set of *decoupled* optimization problems, implying that the computation of control actions can be performed in completely decentralized fashion, i.e. with no communication among controllers (if each $v_{ij}(\cdot)$ is measurable at all $k\in\mathbb{Z}_+$). Inequality (14b) can be further significantly relaxed by replacing the zero on the righthand side with an optimization variable $\tau_i(k)$ and adding the *coupling constraint* $\sum_{i\in\mathscr{S}}\tau_i(k)\leq 0$ for all $k\in\mathbb{Z}_+$. Using the dual decomposition method, see e.g. [13], it is then possible to devise a *distributed* control scheme, which yields an optimized ISS-gain of the global interconnected system in the sense that $\sum_{i\in\mathscr{S}}J_i(\cdot)$ is minimized. Further relaxations can be obtained by asking that the sum of $\tau_i(k)$ is non-positive over a finite horizon, rather than at each time step. □

4.4 Implementation Issues

In this section we briefly discuss the ingredients, which make it possible to implement Problem 2 (or its corresponding decentralized version Problem 3) as a single linear or quadratic program. Firstly, we consider nonlinear systems of the form (4) that are affine in control. Then it makes sense that there exist functions $f_1 : \mathbb{R}^n \to \mathbb{R}^n$ with $f_1(0) = 0$ and $f_2 : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ such that:

$$x(k+1) = \phi(x(k), u(k), w(k)) := f_1(x(k)) + f_2(x(k))u(k) + g(x(k))w(k). \quad (16)$$

Secondly, we restrict our attention to CLFs defined using the ∞-norm, i.e. $V(x) := \|Px\|_\infty$, where $P \in \mathbb{R}^{p\times n}$ is a matrix (to be determined) with full-column rank. We refer to [14] for techniques to compute CLFs based on norms.

Then, the first step is to show that the ISS inequalities (11b)-(11c) can be specified, without introducing conservatism, via a finite number of linear inequalities. Since by definition $\|x\|_\infty = \max_{i\in\mathbb{Z}_{[1,n]}} |[x]_i|$, for a constraint $\|x\|_\infty \le c$ with $c > 0$ to be satisfied, it is *necessary and sufficient* to require that $\pm[x]_i \le c$ for all $i \in \mathbb{Z}_{[1,n]}$. Therefore, as $x(k)$ in (11) is the measured state, which is known at every $k \in \mathbb{Z}_+$, for (11b)-(11c) to be satisfied it is necessary and sufficient to require that:

$$\begin{aligned}
&\pm[P(f_1(x(k)) + f_2(x(k))u(k))]_i - V(x(k)) + \alpha_3(\|x(k)\|) \le 0\\
&\pm[P(f_1(x(k)) + f_2(x(k))u(k) + g(x(k))w^e)]_i - V(x(k)) + \alpha_3(\|x(k)\|) - \lambda_e(k) \le 0,\\
&\forall i \in \mathbb{Z}_{[1,p]},\ e = \overline{1,E},
\end{aligned}$$

which yields $2p(E+1)$ linear inequalities in the variables $u(k), \lambda_1(k), \ldots, \lambda_E(k)$. If the sets $\mathbb{X}$, $\mathbb{U}$ and $\mathbb{W}_{x(k)}$ are polyhedra, which is a reasonable assumption, then clearly the inequalities in (11a) are also linear in $u(k), \lambda_1(k), \ldots, \lambda_E(k)$. Thus, a solution to Problem 2, *including minimization of the cost* $J_{RHC}(\cdot) + J(\cdot)$ *for any* $N \in \mathbb{Z}_{\ge 1}$, can be obtained by solving a nonlinear optimization problem subject to linear constraints.

Following some straightforward manipulations [10], the optimization problem to be solved on-line can be further simplified as follows. If the model is (i) piecewise affine or (ii) affine and the cost functions $J_{\mathrm{RHC}}(\cdot)$ and $J(\cdot)$ are defined using quadratic forms or infinity norms, then a solution to Problem 2 (with the cost $J_{\mathrm{RHC}}(\cdot) + J(\cdot)$) can be obtained by solving (i) a single mixed integer quadratic or linear program (MIQP - MILP), or (ii) a single QP - LP, respectively, for any $N \in \mathbb{Z}_{\ge 1}$. Alternatively, for $N = 1$ and quadratic or ∞-norm based costs, Problem 2 can be formulated as a single QP or LP for any discrete-time nonlinear model that is affine in the control variable and the disturbance input.

5 Illustrative Example

Consider the nonlinear system (13) with $\mathscr{S} = \{1,2\}$, $\mathscr{N}_1 = \{2\}$, $\mathscr{N}_2 = \{1\}$, $\mathbb{X}_1 = \mathbb{X}_2 = \{\xi \in \mathbb{R}^2 \mid \|\xi\|_\infty \le 5\}$, $\mathbb{U}_1 = \mathbb{U}_2 = \{\xi \in \mathbb{R} \mid |\xi| \le 2\}$ and $\mathbb{W}_1 = \mathbb{W}_2 = \{\xi \in \mathbb{R}^2 \mid \|\xi\|_1 \le 0.2\}$. The dynamics are given by:

$$\phi_1(x_1,u_1,v_1(x_{\mathcal{N}_1}),w_1) := \begin{bmatrix}1 & 0.7\\ 0 & 1\end{bmatrix} x_1 + \begin{bmatrix}\sin([x_1]_2)\\ 0\end{bmatrix} + \begin{bmatrix}0.245\\ 0.7\end{bmatrix} u_1 + \begin{bmatrix}0\\ ([x_2]_1)^2\end{bmatrix} + w_1, \quad (17a)$$

$$\phi_2(x_2,u_2,v_2(x_{\mathcal{N}_2}),w_2) := \begin{bmatrix}1 & 0.5\\ 0 & 1\end{bmatrix} x_2 + \begin{bmatrix}\sin([x_2]_2)\\ 0\end{bmatrix} + \begin{bmatrix}0.125\\ 0.5\end{bmatrix} u_2 + \begin{bmatrix}0\\ [x_1]_1\end{bmatrix} + w_2. \quad (17b)$$

The technique of [14] was used to compute the weights $P_1, P_2 \in \mathbb{R}^{2\times 2}$ of the CLFs $V_1(x) = \|P_1 x\|_\infty$ and $V_2(x) = \|P_2 x\|_\infty$ for $\alpha_3^1(s) = \alpha_3^2(s) := 0.01s$ and the linearizations of (17a), (17b), respectively, around the origin, in closed-loop with $u_1(k) := K_1 x_1(k)$, $u_2(k) := K_2 x_2(k)$, $K_1, K_2 \in \mathbb{R}^{2\times 1}$, yielding

$$P_1 = \begin{bmatrix}1.3204 & 0.6294\\ 0.5629 & 2.0811\end{bmatrix}, \; K_1 = \begin{bmatrix}-0.2071 & -1.2731\end{bmatrix},$$

$$P_2 = \begin{bmatrix}1.1356 & 0.5658\\ 0.7675 & 2.1356\end{bmatrix}, \; K_2 = \begin{bmatrix}-0.3077 & -1.4701\end{bmatrix}.$$

Note that the control laws $u_1(k) = K_1 x(k)$ and $u_2(k) = K_2 x_2(k)$ are only employed off-line, to calculate the weight matrices P_1, P_2 and they are never used for controlling the system. To optimize robustness, 4 optimization variables $\lambda_1^i(k), \ldots, \lambda_4^i(k)$ were introduced for each system, each one assigned to a vertex of the set $\mathbb{W}_i$, $i = 1, 2$, respectively. The following cost functions were employed in the optimization problem, as specified in Remark 3: $J^i_{\text{RHC}}(x_i(k), u_i(k)) := \|Q_1^i \phi_i(x_i, u_i, v_i(x_{\mathcal{N}_i}), 0)\|_\infty + \|Q^i x_i(k)\|_\infty + \|R^i u_i(k)\|_\infty$, $J_i(\lambda_1^i(k), \ldots, \lambda_4^i(k)) := \Gamma^i \sum_{j=1}^4 |\lambda_j^i(k)|$, where $i = 1, 2$, $Q_1^1 = Q_1^2 = 4I_2$, $Q^1 = Q^2 = 0.1I_2$, $R^1 = R^2 = 0.4$, $\Gamma^1 = 1$ and $\Gamma^2 = 0.1$. For each system, the resulting linear program has 7 optimization variables and 42 constraints. During the simulations, *the worst case computational time required by the CPU* (Pentium 4, 3.2GHz, 1GB RAM) *over* 400 *runs was* 5 *milliseconds*, which shows the potential for controlling networks of fast nonlinear systems. In the simulation scenario we tested the closed-loop system response for $x_1(0) = [3, -1]^\top$, $x_2(0) = [1, -2]^\top$ and for the following disturbance scenarios: $w_1(k) = w_2(k) = [0, 0]^\top$ for $k \in \mathbb{Z}_{[0,40]}$ (nominal stabilization), $w_i(k)$ takes random values in $\mathbb{W}_i$, $i = 1, 2$, for $k \in \mathbb{Z}_{[41,80]}$ (robustness to random inputs), $w_1(k) = w_2(k) = [0, 0.1]^\top$ for $k \in \mathbb{Z}_{[81,120]}$ (robustness to constant inputs) and $w_1(k) = w_2(k) = [0, 0]^\top$ for $k \in \mathbb{Z}_{[121,160]}$ (to show that asymptotic stability is recovered for zero inputs).

In Figure 1 the time history of the states, control input and the optimization variables $\lambda_1^1(k)$ and $\lambda_1^2(k)$, assigned to $w_1^1 = w_2^1 = [0, 0.2]^\top$, are depicted for each system. In the state trajectories plots, the dashed horizontal lines give an approximation of the bounded region in which the system's states remain despite disturbances, i.e. approximately within the interval $[-0.2, 0.2]$. In the input trajectory plots the dashed line shows the input constraints. In all plots, the dashed vertical lines delimit the time intervals during which one of the four disturbance scenarios is active. One can observe that the feedback to disturbances is provided actively, resulting in good robust performance, while state and input constraints are satisfied at all times, despite the strong nonlinear coupling present. In the λ_1 plot, one can see that whenever the disturbance is acting on the system, or when the state is far from the origin (in the first disturbance scenario), these variables act to optimize the decrease of each $V_i(\cdot)$ and to counteract the influence of the interconnecting signal.

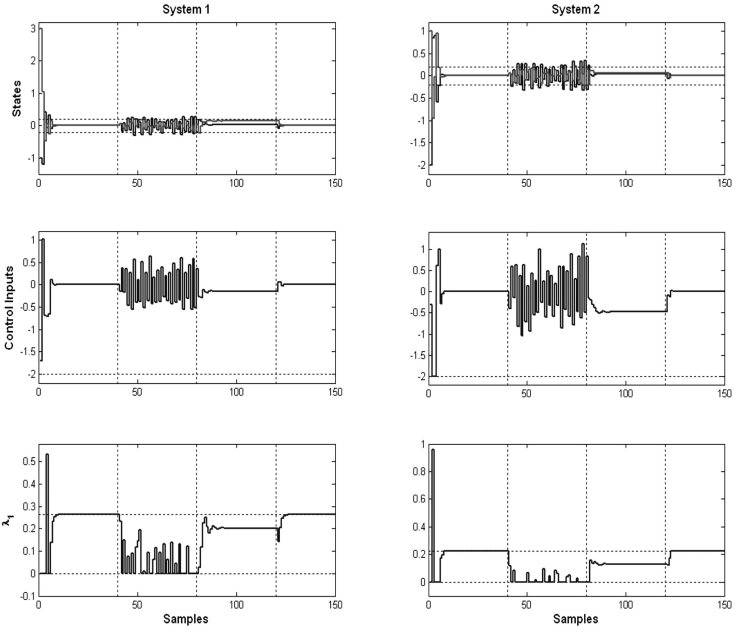


Fig. 1 States, inputs and first optimization variable histories for each system

Whenever the equilibrium is reached, the optimization variables satisfy the constraint $V_i(\phi_i(0,0,w_i^e)) \leq \lambda_e^i(k)$, $e = 1,\ldots,4$, as explained in Remark 4. In Figure 1, the λ_1 plot, the values $V_1(\phi_1(0,0,w_1^1)) = 0.2641$ and $V_2(\phi_2(0,0,w_2^1)) = 0.2271$ are depicted with dashed horizontal lines.

6 Conclusions

In this article we studied the design of robust MPC schemes with focus on adapting the closed-loop ISS gain on-line, in a receding horizon fashion. Exploiting convex CLFs and disturbance bounds, we were able to construct a finite dimensional optimization problem that allows for the simultaneous on-line (i) computation of a control action that achieves ISS, and (ii) minimization of the ISS gain of the resulting closed-loop system depending on the actual state trajectory. As a consequence, the proposed robust nonlinear MPC algorithm is self-optimizing in terms of disturbance attenuation. Solutions for establishing recursive feasibility and for decentralized implementation have also been briefly presented. Furthermore, we indicated a design recipe that can be used to implement the developed self-optimizing MPC scheme as a single linear program, for nonlinear systems that are affine in the control variable and the disturbance input. This brings the application to (networks of) fast nonlinear systems within reach.

References

1. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: Stability and optimality. Automatica 36(6), 789–814 (2000)
2. Lazar, M., Heemels, W.P.M.H., Bemporad, A., Weiland, S.: Discrete-time non-smooth NMPC: Stability and robustness. In: Assessment and Future Directions of NMPC. LNCIS, vol. 358, pp. 93–103. Springer, Heidelberg (2007)
3. Magni, L., Scattolini, R.: Robustness and robust design of MPC for nonlinear discrete-time systems. In: Assessment and Future Directions of NMPC. LNCIS, vol. 358, pp. 239–254. Springer, Heidelberg (2007)
4. Mayne, D.Q., Kerrigan, E.C.: Tube-based robust nonlinear model predictive control. In: 7th IFAC Symposium on Nonlinear Control Systems, Pretoria, pp. 110–115 (2007)
5. Raković, S.V.: Set theoretic methods in model predictive control. In: Proc. of the 3rd Int. Workshop on Assessment and Future Directions of NMPC, Pavia, Italy (2008)
6. Sontag, E.D.: Smooth stabilization implies coprime factorization. IEEE Transactions on Automatic Control AC-34, 435–443 (1989)
7. Sontag, E.D.: Stability and stabilization: Discontinuities and the effect of disturbances. In: Clarke, F.H., Stern, R.J. (eds.) Nonlinear Analysis, Differential Equations, and Control, pp. 551–598. Kluwer, Dordrecht (1999)
8. Jiang, Z.-P., Wang, Y.: Input-to-state stability for discrete-time nonlinear systems. Automatica 37, 857–869 (2001)
9. Kokotović, P., Arcak, M.: Constructive nonlinear control: A historical perspective. Automatica 37(5), 637–662 (2001)
10. Lazar, M.: Model predictive control of hybrid systems: Stability and robustness, PhD Thesis, Eindhoven University of Technology, The Netherlands (2006)
11. Lazar, M., Muñoz de la Peña, D., Heemels, W.P.M.H., Alamo, T.: On input-to-state stability of min-max MPC. Systems & Control Letters 57, 39–48 (2008)
12. Wayne, S.U.: Mathematical Dept., Coffee Room. Every convex function is locally Lipschitz, The American Mathematical Monthly 79(10), 1121–1124 (1972)
13. Bertsekas, D.P.: Nonlinear programming, 2nd edn. Athena Scientific (1999)
14. Lazar, M., Heemels, W.P.M.H., Weiland, S., Bemporad, A.: Stabilizing model predictive control of hybrid systems. IEEE Transactions on Automatic Control 51(11), 1813–1818 (2006)

Set Theoretic Methods in Model Predictive Control

Saša V. Raković

Abstract. The main objective of this paper is to highlight the role of the set theoretic analysis in the model predictive control synthesis. In particular, the set theoretic analysis is invoked to: (i) indicate the fragility of the model predictive control synthesis with respect to variations of the terminal constraint set and the terminal cost function and (ii) discuss a simple, tube based, robust model predictive control synthesis method for a class of nonlinear systems.

Keywords: Control Synthesis, Set Invariance, Tube Model Predictive Control.

1 Introduction

The contemporary research has recognized the need for an adequate mathematical framework permitting the meaningful robust control synthesis for constrained control systems. An appropriate framework to address the corresponding robust control synthesis problems is based on the utilization of the set theoretic analysis, see, for instance, a partial list of pioneering contributions [1–4] and comprehensive monographs [5, 6] for a detailed overview. A set of alternative but complementary control synthesis methods utilizing game–theoretic approaches is also studied [7, 8].

The robust model predictive control synthesis problem is one of the most important and classical problems in model predictive control [9, 10]. The power of the set theoretic analysis has been already utilized in the tube model predictive control synthesis [11–15] and the characterization of the minimal invariant sets [16, 17]. The main objective of this paper is to indicate a further role of the set theoretic analysis in the model predictive control synthesis [9, 12, 18].

Saša V. Raković
Honorary Research Associate at the Centre for Process Systems Engineering,
Imperial College London, SW7 2AZ, UK
and
Scientific Associate at the Institute for Automation Engineering
of Otto–von–Guericke–Universität Magdeburg, Germany
e-mail: svr@sasavrakovic.com

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 41–54.
springerlink.com

Outline of the paper

Section 2 introduces systems under considerations and highlights the role of information available for the control synthesis. Section 3 collects some basic notions, definitions and results relevant for the control synthesis under constraints and uncertainty. Section 4 recalls basic results of the receding horizon control synthesis and utilizes the set theoretic analysis to indicate fragility of the receding horizon control. Section 5 proposes a simple, tube based, robust model predictive control synthesis for a particular class of non–linear systems. Section 6 provides concluding remarks.

Nomenclature and Basic Definitions

The sets of non–negative, positive integers and non–negative real numbers are denoted, respectively, by $\mathbb{N}$, $\mathbb{N}_+$ and R_+, i.e. $\mathbb{N} := \{0,1,2,\ldots\}$, $\mathbb{N}_+ := \{1,2,\ldots\}$ and $R_+ := \{x \in R \ : \ x \geq 0\}$. For $q_1, q_2 \in \mathbb{N}$ such that $q_1 < q_2$ we denote $\mathbb{N}_{[q_1:q_2]} := \{q_1, q_1+1, \ldots, q_2-1, q_2\}$. For two sets $X \subset R^n$ and $Y \subset R^n$, the Minkowski set addition is defined by $X \oplus Y := \{x+y \ : \ x \in X, \ y \in Y\}$ and the Minkowski set subtraction is $X \ominus Y := \{z \in R^n \ : \ z \oplus Y \subseteq X\}$. For a set $X \subset R^n$ and a vector $x \in R^n$ we write $x \oplus X$ instead of $\{x\} \oplus X$. A set $X \subset R^n$ is a C set if it is compact, convex, and contains the origin. A set $X \subset R^n$ is a proper C set if it is a C set and the origin is in its non–empty interior. A set $X \subseteq R^n$ is a symmetric set (with respect to the origin in R^n) if $X = -X$. We denote by $|x|_L$ norm of the vector x induced by a symmetric, proper C set L. For sets $X \subset R^n$ and $Y \subset R^n$, the Hausdorff semi–distance and the Hausdorff distance of X and Y are, respectively, given by:

$$h_L(X,Y) := \inf_{\alpha}\{\alpha \ : \ X \subseteq Y \oplus \alpha L, \ \alpha \geq 0\} \text{ and}$$
$$H_L(X,Y) := \max\{h_L(X,Y), h_L(Y,X)\},$$

where L is a given, symmetric, proper C set in R^n. Given a function $f(\cdot)$, $f^k(x)$, $k \in \mathbb{N}$ stands for its k-th iterate at the point x, i.e $f^k(x) = f(f^{k-1}(x)) = f(f(f^{k-2}(x))) = \ldots$. If $f(\cdot)$ is a set-valued function from, say, X into U, namely, its values are subsets of U, then its graph is the set $\{(x,y) : x \in X, \ y \in f(x)\}$.

2 System Description and Role of Information

In the deterministic case, the variables inducing the dynamics are the state $z \in R^n$ and the control $v \in R^m$. The underlying dynamics in the deterministic case is discrete–time and time–invariant and is generated by a mapping $\bar{f}(\cdot,\cdot) \ : \ R^n \times R^m \to R^n$. When the current state and control are, respectively, z and v, then:

$$z^+ = \bar{f}(z,v) \tag{1}$$

is the successor state. The system variables, i.e. the state z and the control v are subject to hard constraints:

$$z \in \mathbb{Z} \text{ and } v \in \mathbb{V}, \tag{2}$$

where sets $\mathbb{Z}$ and $\mathbb{V}$ are, respectively, subsets of R^n and R^m. Likewise, in the basic uncertain model, the variables inducing the dynamics are the state $x \in R^n$, the control $u \in R^m$ and the disturbance $w \in R^p$. The considered dynamics is discrete–time and time–invariant and is generated by a mapping $f(\cdot,\cdot,\cdot) : R^n \times R^m \times R^p \to R^n$. As in the basic deterministic model, when the current state, control and disturbance are, respectively, x, u and w, then:

$$x^+ = f(x,u,w) \tag{3}$$

is the successor state. The system variables, i.e. the state x, the control u and the disturbance w are subject to hard constraints:

$$x \in \mathbb{X},\ u \in \mathbb{U} \text{ and } w \in \mathbb{W}, \tag{4}$$

where $\mathbb{X}$, $\mathbb{U}$ and $\mathbb{W}$ are, respectively, subsets of R^n, R^m and R^p. In this paper we invoke the following technical assumption:

Assumption 1. *(i) The function $\bar{f}(\cdot,\cdot) : R^n \times R^m \to R^n$ is continuous and sets $\mathbb{Z}$ and $\mathbb{V}$ are compact. (ii) The function $f(\cdot,\cdot,\cdot) : R^n \times R^m \times R^p \to R^n$ is continuous and sets $\mathbb{X}$, $\mathbb{U}$ and $\mathbb{W}$ are compact.*

An additional ingredient playing a crucial role in the uncertain case is the one of the information available for the control synthesis.

Interpretation 1 (Inf–Sup Type Information). *At any time k when the decision concerning the control input u_k is taken, the state x_k is known while the disturbance w_k is not known and can take arbitrary value $w_k \in \mathbb{W}$.*

Under Interpretation 1, at any time instance k, the feedback rules $u_k = u_k(x_k)$ are allowed.

Interpretation 2 (Sup–Inf Type Information). *At any time k when the decision concerning the control input u_k is taken, both the state x_k and the disturbance $w_k \in \mathbb{W}$ are known while future disturbances w_{k+i}, $i \in \mathbb{N}_+$ are not known and can take arbitrary values $w_{k+i} \in \mathbb{W}$, $i \in \mathbb{N}_+$.*

Clearly, under Interpretation 2 the feedback rules $u_k = u_k(x_k, w_k)$ are also, in addition to the feedback rules $u_k = u_k(x_k)$, allowed at any time instance k.

3 Constrained Controllability

An important role of the set theoretic analysis in the control synthesis is the characterization of controllability sets under constraints and uncertainty. A very simple, natural and basic problem, in the control synthesis in the deterministic case, is: *Given a target set $\mathbb{T} \subseteq \mathbb{Z}$, characterize the set of all states $z \in \mathbb{Z}$, say $\mathbb{S}$, and all control laws $v(\cdot) : \mathbb{S} \to \mathbb{V}$ such that for all $z \in \mathbb{S}$ and a control law $v(\cdot)$ it holds that $\bar{f}(z, v(z)) \in \mathbb{T}$.* Obviously, similar questions can be posed, in a transparent way, for both variants, i.e. inf–sup and sup–inf variants, of control synthesis in the uncertain case. We indicate the mathematical formalism providing answers to these

basic questions and obtained by a direct utilization of the set theoretic analysis. We consider in the deterministic case, for a given, non–empty, set $Z \subseteq \mathbb{Z}$, the preimage mapping $\bar{\mathscr{B}}(\cdot)$ and the set–valued control map $\mathscr{V}(\cdot)$ given by:

$$\begin{aligned} &\bar{\mathscr{B}}(Z) := \{z \,:\, \exists v \in \mathbb{V} \text{ such that } \bar{f}(z,v) \in Z\} \cap \mathbb{Z} \text{ and} \\ &\forall z \in \bar{\mathscr{B}}(Z),\ \mathscr{V}(z) := \{v \in \mathbb{V} \,:\, \bar{f}(z,v) \in Z\}. \end{aligned} \tag{5}$$

Similarly, for a given, non–empty, set $X \subseteq \mathbb{X}$, under Interpretation 1, the inf–sup preimage mapping $\mathscr{B}_{\text{inf}-\text{sup}}(\cdot)$ and the inf–sup set–valued control map $\mathscr{U}_{\text{inf}-\text{sup}}(\cdot)$ are given by:

$$\begin{aligned} &\mathscr{B}_{\text{inf}-\text{sup}}(X) := \{x \,:\, \exists u \in \mathbb{U} \text{ such that } \forall w \in \mathbb{W},\ f(x,u,w) \in X\} \cap \mathbb{X} \text{ and} \\ &\forall x \in \mathscr{B}_{\text{inf}-\text{sup}}(X),\ \mathscr{U}_{\text{inf}-\text{sup}}(x) := \{u \in \mathbb{U} \,:\, \forall w \in \mathbb{W},\ f(x,u,w) \in X\}. \end{aligned} \tag{6}$$

Likewise, under Interpretation 2, the sup–inf preimage mapping $\mathscr{B}_{\text{sup}-\text{inf}}(\cdot)$ and the sup–inf set–valued control map $\mathscr{U}_{\text{sup}-\text{inf}}(\cdot,\cdot)$ are given by:

$$\begin{aligned} &\mathscr{B}_{\text{sup}-\text{inf}}(X) := \{x \,:\, \forall w \in \mathbb{W},\ \exists u \in \mathbb{U} \text{ such that } f(x,u,w) \in X\} \cap \mathbb{X} \text{ and} \\ &\forall (x,w) \in \mathscr{B}_{\text{sup}-\text{inf}}(X) \times \mathbb{W},\ \mathscr{U}_{\text{sup}-\text{inf}}(x,w) := \{u \in \mathbb{U} \,:\, f(x,u,w) \in X\}. \end{aligned} \tag{7}$$

Evidently, given a non–empty set $Z \subseteq \mathbb{Z}$, the set of states that are one step controllable to Z is the set $\bar{\mathscr{B}}(Z)$ and any control law $v(\cdot) \,:\, \bar{\mathscr{B}}(Z) \to \mathbb{V}$ ensuring that the successor state $\bar{f}(z,v(z))$ is in the set Z is a selection of the set–valued control map $\mathscr{V}(\cdot)$. Similarly, the set of states that are one step inf–sup controllable to X is the set $\mathscr{B}_{\text{inf}-\text{sup}}(X)$ and any control law $u(\cdot) \,:\, \bar{\mathscr{B}}(X) \to \mathbb{U}$ ensuring that all possible successor states $f(x,u(x),w),\ w \in \mathbb{W}$ are in the set X is a selection of the inf–sup set–valued control map $\mathscr{U}_{\text{inf}-\text{sup}}(\cdot)$. Likewise, the set of states that are one step sup–inf controllable to X is the set $\mathbb{S}_{\text{sup}-\text{inf}} = \mathscr{B}_{\text{sup}-\text{inf}}(X)$ and any control law $u(\cdot,\cdot) \,:\, \bar{\mathscr{B}}(X) \times \mathbb{W} \to \mathbb{U}$ ensuring that any successor state $f(x,u(x,w),w)$ is in the set X is a selection of the sup–inf set–valued control map $\mathscr{U}_{\text{sup}-\text{inf}}(\cdot,\cdot)$. If Assumption 1 (i) holds and a target set Z is compact, the set $\bar{\mathscr{B}}(Z)$ and the graph of the set–valued control map $\mathscr{V}(\cdot)$ are compact when non–empty. Likewise, if Assumption 1 (ii) holds and a target set X is compact then the set $\mathscr{B}_{\text{inf}-\text{sup}}(X)$ and the graph of the inf–sup set–valued control map $\mathscr{U}_{\text{inf}-\text{sup}}(\cdot)$ are compact when non–empty and, similarly, the set $\mathscr{B}_{\text{sup}-\text{inf}}(X)$ and the graph of the sup–inf set–valued control map $\mathscr{U}_{\text{sup}-\text{inf}}(\cdot,\cdot)$ are compact when non–empty. The semi–group property of preimage mappings permits the characterization of the N–step, the N–step inf–sup and the N–step sup–inf controllable sets and corresponding set–valued control maps by dynamic programming procedures indicated next. Let N be an arbitrary integer and let $\mathbb{T}$ be a given target set. The N–step controllable sets and corresponding set–valued control maps are given in the deterministic case, for $j \in \mathbb{N}_{[1:N]}$, by:

$$\begin{aligned} &\mathscr{Z}_j := \bar{\mathscr{B}}(\mathscr{Z}_{j-1}) \text{ and } \forall z \in \mathscr{Z}_j, \\ &\mathscr{V}_j(z) := \{v \in \mathbb{V} \,:\, \bar{f}(z,v) \in \mathscr{Z}_{j-1}\}, \end{aligned} \tag{8}$$

with the boundary condition $\mathscr{Z}_0 := \mathbb{T} \subseteq \mathbb{Z}$. Similarly, the N–step inf–sup controllable sets and corresponding inf–sup set–valued control maps are given, for $j \in \mathbb{N}_{[1:N]}$, by:

$$\begin{aligned} \mathscr{X}_{\text{inf}-\text{sup}_j} &:= \mathscr{B}_{\text{inf}-\text{sup}}(\mathscr{X}_{\text{inf}-\text{sup}_{j-1}}) \text{ and } \forall x \in \mathscr{X}_{\text{inf}-\text{sup}_j}, \\ \mathscr{U}_{\text{inf}-\text{sup}_j}(x) &:= \{u \in \mathbb{U} \,:\, \forall w \in \mathbb{W},\ f(x,u,w) \in \mathscr{X}_{\text{inf}-\text{sup}_{j-1}}\}, \end{aligned} \tag{9}$$

with the boundary condition $\mathscr{X}_{\text{inf}-\text{sup}_0} := \mathbb{T} \subseteq \mathbb{X}$. Likewise, the N–step sup–inf controllable sets and corresponding sup–inf set–valued control maps are given, for $j \in \mathbb{N}_{[1:N]}$, by:

$$\begin{aligned} \mathscr{X}_{\text{sup}-\text{inf}_j} &:= \mathscr{B}_{\text{sup}-\text{inf}}(\mathscr{X}_{\text{sup}-\text{inf}_{j-1}}) \text{ and } \forall (x,w) \in \mathscr{X}_{\text{sup}-\text{inf}_j} \times \mathbb{W}, \\ \mathscr{U}_{\text{sup}-\text{inf}_j}(x,w) &:= \{u \in \mathbb{U} \,:\, f(x,u,w) \in \mathscr{X}_{\text{sup}-\text{inf}_{j-1}}\}, \end{aligned} \tag{10}$$

with the boundary condition $\mathscr{X}_{\text{sup}-\text{inf}_0} := \mathbb{T} \subseteq \mathbb{X}$. If Assumption 1 holds and a target set $\mathbb{T}$ is compact then: (i) the k–step controllable set $\mathscr{Z}_k$ satisfies $\mathscr{Z}_k = \bar{\mathscr{B}}^k(\mathbb{T})$ and $\mathscr{Z}_k$ and the graph of the set–valued control map $\mathscr{V}_k(\cdot)$ are compact when non–empty; (ii) the k–step inf–sup controllable set $\mathscr{X}_{\text{inf}-\text{sup}_k}$ satisfies $\mathscr{X}_{\text{inf}-\text{sup}_k} = \mathscr{B}^k_{\text{inf}-\text{sup}}(\mathbb{T})$ and $\mathscr{X}_{\text{inf}-\text{sup}_k}$ and the graph of the inf–sup set–valued control map $\mathscr{U}_{\text{inf}-\text{sup}_k}(\cdot)$ are compact when non–empty; and (iii) the k–step sup–inf controllable set $\mathscr{X}_{\text{sup}-\text{inf}_k}$ satisfies $\mathscr{X}_{\text{sup}-\text{inf}_k} = \mathscr{B}^k_{\text{sup}-\text{inf}}(\mathbb{T})$ and $\mathscr{X}_{\text{sup}-\text{inf}_k}$ and the graph of the sup–inf set–valued control map $\mathscr{U}_{\text{sup}-\text{inf}_k}(\cdot,\cdot)$ are compact when non–empty. It is well known that the non–emptiness of the N–step, the N–step inf–sup and the N–step sup–inf controllable sets represents, respectively, necessary and sufficient conditions for solvability of the N–step, the N–step inf–sup and the N–step sup–inf controllability to a target set control problems [2, 4–6]. Likewise, the afore mentioned non–emptiness plays a crucial role in the solvability of the finite horizon (of length N) optimal and robust (inf–sup and sup–inf) optimal control problems in the presence of terminal set constraints (as is the case in the receding horizon control [9]).

A further role of preimage mappings is also evident in set invariance [5, 6]:

Definition 1. *A set Z is a control invariant set for the system $z^+ = \bar{f}(z,v)$ and constraints set $(\mathbb{Z},\mathbb{V})$ if and only if $Z \subseteq \bar{\mathscr{B}}(Z)$.*

A set X is an inf–sup control invariant set for the system $x^+ = f(x,u,w)$ and constraints set $(\mathbb{X},\mathbb{U},\mathbb{W})$ if and only if $X \subseteq \mathscr{B}_{\text{inf}-\text{sup}}(X)$.

A set X is a sup–inf control invariant set for the system $x^+ = f(x,u,w)$ and constraints set $(\mathbb{X},\mathbb{U},\mathbb{W})$ if and only if $X \subseteq \mathscr{B}_{\text{sup}-\text{inf}}(X)$.

A more subtle issue is related to properties of fixed points of preimage mappings. The fixed point set equations of preimage mappings take the form:

$$X = \bar{\mathscr{B}}(X),\ X = \mathscr{B}_{\text{inf}-\text{sup}}(X) \text{ and } X = \mathscr{B}_{\text{sup}-\text{inf}}(X), \tag{11}$$

where the unknown, in any of the three cases, is the set X. It is known [5, 6] that, under Assumption 1, special fixed points of preimage mappings are the maximal

control invariant set $\bar{\Omega}_\infty$, the maximal inf–sup control invariant set $\Omega_\infty^{\inf-\sup}$ and the maximal sup–inf control invariant set $\Omega_\infty^{\sup-\inf}$ given, respectively, by:

$$\bar{\Omega}_\infty = \bigcap_{k=0}^{\infty} \bar{\mathscr{B}}^k(\mathbb{Z}),\ \Omega_\infty^{\inf-\sup} = \bigcap_{k=0}^{\infty} \mathscr{B}^k_{\inf-\sup}(\mathbb{X}),\ \text{and}\ \Omega_\infty^{\sup-\inf} = \bigcap_{k=0}^{\infty} \mathscr{B}^k_{\sup-\inf}(\mathbb{X}).$$

Under Assumption 1, sets $\bar{\Omega}_\infty$, $\Omega_\infty^{\inf-\sup}$ and $\Omega_\infty^{\sup-\inf}$ are compact, when non–empty, and are, in fact, unique maximal (with respect to set inclusion) fixed points of corresponding preimage mappings $\bar{\mathscr{B}}(\cdot)$, $\mathscr{B}_{\inf-\sup}(\cdot)$ and $\mathscr{B}_{\sup-\inf}(\cdot)$.

4 Fragility of Receding Horizon Control

We indicate fragility of the receding horizon control by utilizing the set theoretic analysis and exploiting the fact that fixed points of preimage mappings are, in general, non–unique. Given an integer $N \in \mathbb{N}$ let $\mathbf{v}_N := \{v_0, v_1, \ldots, v_{N-1}\}$ denote the control sequence of length N, let also $\phi(i,z,\mathbf{v}_N)$ denote the solution of (1) at time $i \in \mathbb{N}_{[0:N]}$ when the initial state at time 0 is z and the control sequence is $\mathbf{v}_N$. The cost function $V_N(\cdot,\cdot) : \mathbb{Z} \times \mathbb{V}^N \to R_+$ is specified by:

$$V_N(z,\mathbf{v}_N) := \sum_{i=0}^{N-1} \ell(\phi(i;z,\mathbf{v}_N), v_i) + V_f(\phi(N;z,\mathbf{v}_N)), \tag{12}$$

where functions $\ell(\cdot,\cdot) : \mathbb{Z} \times \mathbb{V} \to R_+$ and $V_f(\cdot) : \mathbb{Z}_f \to R_+$ are the path and terminal cost and $\mathbb{Z}_f \subseteq \mathbb{Z}$ is the terminal constraint set. Let also

$$\mathbb{V}_N(z) := \{\mathbf{v}_N \in \mathbb{V}^N : \forall i \in \mathbb{N}_{[0,N-1]},\ (\phi(i;z,\mathbf{v}_N), v_i) \in \mathbb{Z} \times \mathbb{V} \text{ and } \phi(N;z,\mathbf{v}_N) \in \mathbb{Z}_f\}, \tag{13}$$

denote the set of admissible control sequences at an initial condition $z \in \mathbb{Z}$. We invoke usual assumptions employed in the model predictive control [9]:

Assumption 2. *(i) The function $\bar{f}(\cdot,\cdot)$ satisfies $0 = \bar{f}(0,0)$. (ii) The terminal constraint set $\mathbb{Z}_f \subseteq \mathbb{Z}$ is a compact set and $(0,0) \in \mathbb{Z}_f \times \mathbb{V}$. (iii) The path and terminal cost functions $\ell(\cdot,\cdot) : \mathbb{Z} \times \mathbb{V} \to R_+$ and $V_f(\cdot) : \mathbb{Z}_f \to R_+$ are continuous, $\ell(0,0) = 0$ and $V_f(0) = 0$ and there exist positive scalars c_1, c_2, c_3 and c_4 such that for all $(z,v) \in \mathbb{Z} \times \mathbb{V}$ it holds that $c_1|z|^2 \le \ell(z,v) \le c_2$ and for all $x \in \mathbb{Z}_f$ it holds that $c_3|z|^2 \le V_f(z) \le c_4|z|^2$ (iv) There exists a control law $\kappa_f(\cdot) : \mathbb{Z}_f \to \mathbb{V}$ such that for all $z \in \mathbb{Z}_f$ it holds that $\bar{f}(z,\kappa_f(z)) \in \mathbb{Z}_f$ and $V_f(\bar{f}(z,\kappa_f(z))) + \ell(z,\kappa_f(z)) \le V_f(z)$.*

Given an integer $N \in \mathbb{N}$, we consider the optimal control problem $\mathbb{P}_N(z)$:

$$\begin{aligned} \mathbb{P}_N(z): \qquad & V_N^*(z) := \min_{\mathbf{v}_N} \{V_N(z,\mathbf{v}_N) : \mathbf{v}_N \in \mathbb{V}_N(z)\} \\ & \mathbf{v}_N^*(z) := \arg\min_{\mathbf{v}_N} \{V_N(z,\mathbf{v}_N) : \mathbf{v}_N \in \mathbb{V}_N(z)\}. \end{aligned} \tag{14}$$

In the model predictive control, the optimal control problem $\mathbb{P}_N(z)$ is solved on–line at the state z, encountered in the process, and the optimizing control sequence $\mathbf{v}_N^*(z) = \{v_0^*(z), v_1^*(z), \ldots, v_{N-1}^*(z)\}$ is utilized to obtain the model predictive control law by applying its first term $v_0^*(z)$ (or its selection when $\mathbf{v}_N^*(z)$ is set–valued) to the system (1). The domain of the value function $V_N^*(\cdot)$ and the optimizing control sequence $\mathbf{v}_N^*(\cdot)$ is given by:

$$\mathscr{Z}_{\mathrm{MPC}N} := \{z \,:\, \mathbb{V}_N(z) \neq \emptyset\}. \tag{15}$$

At the conceptual level, in the deterministic case, the model predictive control law is an on–line implementation of the receding horizon control law; The explicit form of the receding horizon control law and the value function can be obtained by solving the optimal control problem $\mathbb{P}_N(z)$ utilizing parametric mathematical programming techniques [19] or by employing parametric mathematical programming in conjunction with the standard dynamic programming procedure given, for each $j \in \mathbb{N}_{[1:N]}$, by:

$$\begin{aligned}
\mathscr{Z}_j &:= \bar{\mathscr{B}}(\mathscr{Z}_{j-1}), \quad \forall z \in \mathscr{Z}_j,\ \mathscr{V}_j(z) := \{v \in \mathbb{V} \,:\, \bar{f}(z,v) \in \mathscr{Z}_{j-1}\},\\
\forall z \in \mathscr{Z}_j,\ V_j^*(z) &:= \min_v \{\ell(z,v) + V_{j-1}^*(\bar{f}(z,v)) \,:\, v \in \mathscr{V}_j(z)\},\\
\forall z \in \mathscr{Z}_j,\ \kappa_j^*(z) &:= \arg\min_v \{\ell(z,v) + V_{j-1}^*(\bar{f}(z,v)) \,:\, v \in \mathscr{V}_j(z)\},\ \text{with}\\
\mathscr{Z}_0 &:= \mathbb{Z}_f \text{ and } \forall z \in \mathscr{Z}_0,\ V_0^*(z) := V_f(z).
\end{aligned} \tag{16}$$

Obviously, qualitative system theoretic properties of the model predictive control and the receding horizon control are equivalent. Under Assumptions 1 and 2, the origin is an exponentially stable attractor for the controlled system, possibly set–valued system when $\kappa_N^*(\cdot)$ is set–valued,:

$$\forall z \in \mathscr{Z}_N,\ z^+ \in \bar{F}(z) := \{\bar{f}(z,\tilde{v}) \,:\, \tilde{v} \in \kappa_N^*(z)\}, \tag{17}$$

with the basin of attraction being equal to a compact set $\mathscr{Z}_N = \bar{\mathscr{B}}^N(\mathbb{Z}_f)$. The corresponding stability property is, in fact, a strong property, that is it holds for all state trajectories $\{z_k\}_{k=0}^{\infty}$ satisfying (17). The first role of set theoretic analysis is the characterization of the domain $\mathscr{Z}_N$ of the value function $V_N^*(\cdot)$ and the receding horizon control law $\kappa_N^*(\cdot)$; The corresponding domain is the set $\bar{\mathscr{B}}^N(\mathbb{Z}_f)$ which is, under Assumptions 1 and 2, compact for any fixed integer $N \in \mathbb{N}$ and, additionally, an invariant set (i.e. $\forall z \in \mathscr{Z}_N,\ \bar{F}(z) \subseteq \mathscr{Z}_N$). Clearly, it is, then, of interest to understand how is the domain of functions $V_N^*(\cdot)$ and $\kappa_N^*(\cdot)$ affected by the variation of the terminal constraint set $\mathbb{Z}_f$ (and possibly variation of the terminal cost function $V_f(\cdot)$). A crucial point in understanding this important issue is closely related to properties of the preimage mapping $\bar{\mathscr{B}}(\cdot)$ and, in fact, non–uniqueness and attractivity properties of its fixed points. We now deliver a simple example illustrating the fragility of the receding horizon control.

Example 1. Consider the two dimensional system generated by a linear map:

$$\bar{f}(z,v) = \frac{1}{2}Iz + Iv, \text{ so that } z^+ = \frac{1}{2}Iz + Iv,$$

with constraints on system variables $z \in R^2$ and $v \in R^2$:

$$\mathbb{Z} = [-4,4] \times [-2^l, 2^l] \text{ and } \mathbb{V} = [-1,1] \times \{0\} \text{ with } l \in \mathbb{N}.$$

The maximal control invariant set is, clearly, the set $\mathbb{Z}$ as:

$$\mathbb{Z} = \bar{\mathscr{B}}(\mathbb{Z}) \text{ and, consequently, } \mathbb{Z} = \bar{\mathscr{B}}^N(\mathbb{Z}) \text{ for any integer } N \in \mathbb{N}.$$

However, the set $Z := [-4,4] \times \{0\}$ is also a fixed point of the mapping $\bar{\mathscr{B}}(\cdot)$:

$$Z = \bar{\mathscr{B}}(Z) \text{ and, consequently, } Z = \bar{\mathscr{B}}^N(Z) \text{ for any integer } N \in \mathbb{N}.$$

A moment of reflection reveals that for any compact set Y such that $0 \in Y \subseteq Z$:

$$[-2,2] \times \{0\} \subseteq \bar{\mathscr{B}}(Y) \subseteq [-4,4] \times \{0\} \text{ and } \bar{\mathscr{B}}(Y) \subseteq \bar{\mathscr{B}}^{k+1}(Y) = [-4,4] \times \{0\},$$

for any integer $k \in \mathbb{N}_+$. A further examination shows that for any compact set X such that $\{0\} \times [-2\varepsilon, 2\varepsilon] = X \subseteq \mathbb{Z}$ (with $\varepsilon > 0$ arbitrarily small) we have:

$$\begin{aligned}
&X \subseteq [-2,2] \times [-4\varepsilon, 4\varepsilon] \cap \mathbb{Z} = \bar{\mathscr{B}}(X) \subseteq [-4,4] \times [-4\varepsilon, 4\varepsilon] \cap \mathbb{Z}, \\
&\forall k \in \mathbb{N}_+,\ \bar{\mathscr{B}}^k(X) \subseteq \bar{\mathscr{B}}^{k+1}(X) = [-4,4] \times [-2^{k+2}\varepsilon, 2^{k+2}\varepsilon] \cap \mathbb{Z}, \text{ and} \\
&\text{for all } k \in \mathbb{N} \text{ such that } k \geq 2 \text{ and } 2^{k+1}\varepsilon \geq 2^l,\ \bar{\mathscr{B}}^k(X) = \mathbb{Z}.
\end{aligned}$$

A variant of the example relevant to the receding horizon control follows. Pick an integer $N \in \mathbb{N}$ and consider the receding horizon control synthesis problem with the following ingredients. The path cost function is:

$$\ell(z,v) = z'Qz + v'Rv \text{ with } Q = \frac{14}{16}I \text{ and } R = I,$$

The terminal cost function is the unconstrained infinite horizon value function:

$$V_f(z) = z'Iz.$$

The corresponding unconstrained infinite horizon optimal control law is:

$$\kappa_f(z) = -\frac{1}{4}Iz$$

and the terminal constraint set $\mathbb{Z}_f$ is the maximal positively invariant set for the system $z^+ = \frac{1}{4}Iz$ subject to constraints $z \in \mathbb{Z}$ and $\frac{1}{4}Iz \in \mathbb{V}$ which turns out to be the set $Z = [-4,4] \times \{0\}$. All assumptions (our Assumptions 1 and 2) commonly employed in the model predictive control literature [9] are satisfied. Unfortunately,

the set $\mathbb{Z}_f = Z = [-4,4] \times \{0\}$ is a fixed point of the preimage mapping $\bar{\mathscr{B}}(\cdot)$ so that $\mathbb{Z}_f = \bar{\mathscr{B}}^N(\mathbb{Z}_f)$ for any integer N. In turn, regardless of the choice of the horizon length $N \in \mathbb{N}$, the receding horizon control law $\kappa_N^*(\cdot)$ and the corresponding value function $V_N^*(\cdot)$ are defined only over the set $\mathscr{Z}_N = \bar{\mathscr{B}}^N(\mathbb{Z}_f) = \mathbb{Z}_f = Z$, which is a compact, zero measure, subset of the maximal control invariant set $\mathbb{Z}$. The exact constrained infinite horizon control value function and control law are given by:

$$\forall z \in \mathbb{Z},\ V_\infty^*(z) = z' \begin{pmatrix} 1 & 0 \\ 0 & \frac{14}{12} \end{pmatrix} z, \text{ and } \kappa_\infty^*(z) = \begin{pmatrix} -\frac{1}{4} & 0 \\ 0 & 0 \end{pmatrix} z.$$

In fact, in our example, the following, fixed–point type, relations hold true:

$$\begin{aligned} &\mathbb{Z} = \bar{\mathscr{B}}(\mathbb{Z}),\ \forall z \in \mathbb{Z},\ \mathscr{V}_\infty(z) = \{v \in \mathbb{V} \ :\ \bar{f}(z,v) \in \mathbb{Z}\} \neq \emptyset, \\ &\forall z \in \mathbb{Z},\ V_\infty^*(z) = \min_v \{\ell(z,v) + V_\infty^*(\bar{f}(z,v)) \ :\ v \in \mathscr{V}_\infty(z)\}, \\ &\forall z \in \mathbb{Z},\ \kappa_\infty^*(z) = \arg\min_v \{\ell(z,v) + V_\infty^*(\bar{f}(z,v)) \ :\ v \in \mathscr{V}_\infty(z)\}. \end{aligned}$$

Choosing the terminal constraint set $\mathbb{Z}_f = [-4,4] \times [-2\varepsilon, 2\varepsilon] \subseteq \mathbb{Z}$, with $\varepsilon > 0$, and the terminal cost function $V_f(\cdot) = V_\infty^*(\cdot)$ leads (for a sufficiently large horizon length N, for example for $N \in \mathbb{N}$ such that $2^{N+1}\varepsilon \geq 2^l$) to the receding horizon control law $\kappa_N^*(\cdot)$ and the value function $V_N^*(\cdot)$ identically equal over the whole set $\mathbb{Z}$ to the infinite horizon control law $\kappa_\infty^*(\cdot)$ and the infinite horizon value function $V_\infty^*(\cdot)$. Hence an $\varepsilon > 0$ variation of the ingredients (the set $\mathbb{Z}_f$ and the function $V_f(\cdot)$) for the receding horizon control synthesis results in a discontinuous change of domains of the corresponding receding horizon control law $\kappa_N^*(\cdot)$ and the corresponding value function $V_N^*(\cdot)$ (notice that the Hausdorff distance between sets $Z = [-4,4] \times \{0\}$ and $\mathbb{Z} = [-4,4] \times [-2^l, 2^l]$ can be made as large as we please by setting l large enough).

Conclusion 1. *Standard assumptions employed in the model predictive control [9] (summarized by Assumptions 1 and 2) are not, in the general case, sufficient assumptions to ensure convergence (as $N \to \infty$) of the receding horizon control law $\kappa_N^*(\cdot)$, the value function $V_N^*(\cdot)$ and the corresponding domain $\mathscr{Z}_N$ to the infinite horizon control law $\kappa_\infty^*(\cdot)$, the value function $V_\infty^*(\cdot)$ and the corresponding domain denoted Z_∞. Furthermore, an $\varepsilon > 0$ variation of the terminal constraint set $\mathbb{Z}_f$ and the terminal cost function $V_f(\cdot)$, such that the perturbed data satisfy usual assumptions can result, in general case, in the discontinuous change of the domain $\mathscr{Z}_N$ of the receding horizon control law $\kappa_N^*(\cdot)$ and the corresponding value function $V_N^*(\cdot)$; Hence the receding horizon control synthesis is fragile, even in the linear–polytopic case, with respect to feasible perturbations of the terminal constraint set $\mathbb{Z}_f$ and the terminal cost function $V_f(\cdot)$.*

5 Simple Tube Model Predictive Control

The potential structure of the underlying mapping $f(\cdot,\cdot,\cdot)$ generating the dynamics is beneficial for the simplified inf–sup tube model predictive control synthesis, as

illustrated in [11–14] for special classes of discrete time systems (including linear, piecewise affine and some classes of nonlinear systems). Ideas employed in [11–14] are, now, demonstrated by considering a class of non–linear systems (that has interesting structure and has not been treated in [11–15]) for which:

$$f(x,u,w) = g(x) + Bu + w \text{ so that } x^+ = g(x) + Bu + w. \tag{18}$$

With the uncertain system (18) we associate a nominal system generated by:

$$\bar{f}(z,v) = g(z) + Bv \text{ so that } z^+ = g(z) + Bv, \tag{19}$$

and work in this subsection under the following simplifying assumption:

Assumption 3. *There exists a function* $\theta(\cdot,\cdot) : R^n \times R^n \to R^m$ *such that:*
(i) for all x *and* z, $|g(x) - g(z) + B\theta(x,z)|_L \leq \lambda |x-z|_L$ for some $\lambda \in [0,1)$*; (ii) for all* x *and* z *such that* $|x-z|_L \leq \gamma$*, where* $\gamma := (1-\lambda)^{-1}\mu$ *and* $\mu := h_L(\mathbb{W},\{0\})$*, it holds that* $|\theta(x,z)|_M \leq \eta$*; (iii) for all* $x \in \gamma L$ *and* $y \in \gamma L$, $|g(x) + B\theta(x,0) - g(y) - B\theta(y,0)|_L \leq \lambda^* |x-y|_L$ for some $\lambda^* \in [0,\lambda] \subset [0,1)$.

Since $|g(x)+B(v+\theta(x,z))+w-g(z)-Bv|_L \leq \lambda|x-z|_L+|w|_L$ by Assumption 3 (i), and, since $\lambda(1-\lambda)^{-1}h_L(\mathbb{W},\{0\})+h_L(\mathbb{W},\{0\}) = (1-\lambda)^{-1}h_L(\mathbb{W},\{0\})$, the following simple but useful fact is affirmative:

Lemma 1. *Suppose Assumptions 3 (i) and 3 (ii) hold and consider a set* $X := z \oplus \gamma L$ *where* $z \in R^n$, $\gamma := (1-\lambda)^{-1}\mu$ *and* $\mu := h_L(\mathbb{W},\{0\})$*. Then for all* $x \in X$ *and all* $v \in R^m$ *it holds that* $\theta(x,z) \in \eta M$ *and* $g(x) + B(v+\theta(x,z)) \oplus \mathbb{W} \subseteq z^+ \oplus \gamma L$ *with* $z^+ = g(z) + Bv$.

Lemma 1 motivates the use of the parameterized inf–sup tube–control policy pair. The parameterized tube $\mathbf{X}_{\inf-\sup_N}$ is the sequence of sets $\{X_{\inf-\sup_k}\}_{k=0}^N$ where:

$$\forall k \in \mathbb{N}_{[0:N]},\ X_{\inf-\sup_k} := z_k \oplus \gamma L. \tag{20}$$

The corresponding parameterized policy $\Pi_{\inf-\sup_N}$ is the sequence of control laws $\{\pi_{\inf-\sup_k}(\cdot,\cdot)\}_{k=0}^{N-1}$ where:

$$\forall k \in \mathbb{N}_{[0:N-1]},\ \forall y \in X_{\inf-\sup_k},\ \pi_{\inf-\sup_k}(y,z_k) := v_k + \theta(y,z_k). \tag{21}$$

To exploit fully Lemma 1 we need an additional and mild assumption:

Assumption 4. *Sets* $\mathbb{Z} := \mathbb{X} \ominus \gamma L$ *and* $\mathbb{V} := \mathbb{U} \ominus \eta M$ *are non–empty and such that* $(0,0) \in \mathbb{Z} \times \mathbb{V}$.

A direct argument exploiting mathematical induction and Lemma 1 yields:

Proposition 1. *Suppose Assumptions 3 (i), 3 (ii) and Assumption 4 hold. Assume also that sequences* $\{z_k\}_{k=0}^N$ *and* $\{v_k\}_{k=0}^{N-1}$ *are such that* $z_0 \in \mathbb{Z}$ *and, for all* $k \in \mathbb{N}_{[0:N-1]}$, $z_{k+1} = g(z_k) + Bv_k \in \mathbb{Z}$ *and* $v_k \in \mathbb{V}$*. Consider the parameterized tube–control policy pair* $(\mathbf{X}_{\inf-\sup_N}, \Pi_{\inf-\sup_N})$ *given by (20) and (21). Then* $X_{\inf-\sup_0} = z_0 \oplus \gamma L \subseteq \mathbb{Z} \oplus \gamma L \subseteq \mathbb{X}$ *and for all* $k \in \mathbb{N}_{[0:N-1]}$ *it holds that:*

$$\begin{aligned}
&\forall y \in X_{\inf-\sup_k},\ \pi_{\inf-\sup_k}(y,z_k) = v_k + \theta(y,z_k) \in \mathbb{V} \oplus \eta M \subseteq \mathbb{U},\\
&\qquad\qquad X_{\inf-\sup_{k+1}} = z_{k+1} \oplus \gamma L \subseteq \mathbb{Z} \oplus \gamma L \subseteq \mathbb{X},\ \text{and}\ ,\\
&\forall y \in X_{\inf-\sup_k},\ g(y) + B\pi_{\inf-\sup_k}(y,z_k) \oplus \mathbb{W} \subseteq z_{k+1} \oplus \gamma L = X_{\inf-\sup_{k+1}}.
\end{aligned}$$

We now provide a more general result under an additional assumption and utilize it in conjunction with Proposition 1 for the tube model predictive control.

Assumption 5. *There exists a compact set $\mathscr{Z}_N \subseteq \mathbb{Z}$ with $0 \in \mathscr{Z}_N$ and functions $\kappa_N^*(\cdot) : \mathscr{Z}_N \to \mathbb{V}$ with $\kappa_N^*(0) = 0$ and $V_N^*(\cdot) : \mathscr{Z}_N \to R_+$ with $V_N^*(0) = 0$ such that: (i) For all $z \in \mathscr{Z}_N$ it holds that $z^+ = g(z) + B\kappa_N^*(z) \in \mathscr{Z}_N$; (ii) The origin is exponentially stable for the controlled system $z^+ = g(z) + B\kappa_N^*(z)$ with the basin of attraction $\mathscr{Z}_N$, i.e. all sequences $\{z_k\}_{k=0}^{\infty}$ with arbitrary $z_0 \in \mathscr{Z}_N$ and $z_{k+1} = g(z_k) + B\kappa_N^*(z_k)$ satisfy $|z_k|_L \leq \alpha^k \beta |z_0|_L$ for some $\alpha \in [0,1)$ and $\beta \in [0,\infty)$; (iii) The function $V_N^*(\cdot)$ is lower semi–continuous over the set $\mathscr{Z}_N$, continuous at the origin and it induces the property assumed above in (ii).*

For all $x \in \mathscr{Z}_N \oplus \gamma L$ let $\mathscr{Z}(x) := \{z \in \mathscr{Z}_N\ :\ (x - z) \in \gamma L\}$ and define:

$$\begin{aligned}
&\forall x \in \mathscr{Z}_N \oplus \gamma L,\ V_N^0(x) := \min_z \{V_N^*(z)\ :\ z \in \mathscr{Z}(x)\},\ \text{and}\\
&\forall x \in \mathscr{Z}_N \oplus \gamma L,\ z^0(x) := \arg\min_z \{V_N^*(z)\ :\ z \in \mathscr{Z}(x)\}.
\end{aligned} \tag{22}$$

We consider the feedback control law and the corresponding induced controlled uncertain system given by:

$$\begin{aligned}
&\forall x \in \mathscr{Z}_N \oplus \gamma L,\ \kappa_N^0(x) := \kappa_N^*(z^0(x)) + \theta(x, z^0(x))\ \text{and}\\
&\forall x \in \mathscr{Z}_N \oplus \gamma L,\ x^+ \in F(x) := \{g(x) + B\kappa_N^0(x) + w\ :\ w \in \mathbb{W}\},
\end{aligned} \tag{23}$$

A straight–forward utilization of Lemma 1 and construction above yields:

Theorem 1. *Suppose Assumptions 1, 3 (i), 3 (ii), 4 and 5 hold. Then: (i) for all $x \in \mathscr{Z}_N \oplus \gamma L$ it holds that $\mathscr{Z}(x) \neq \emptyset$ and for any $x \in \mathscr{Z}_N \oplus \gamma L$ there exists at least one $z \in \mathscr{Z}(x)$ such that $V_N^*(z) = V_N^0(x)$; (ii) for all $x \in z \oplus \gamma L$ with arbitrary $z \in \mathscr{Z}_N$ it holds that $V_N^0(x) = V_N^*(z^0(x)) \leq V_N^*(z)$; (iii) for all $x \in \gamma L$ it holds that $V_N^0(x) = 0$, $z^0(x) = 0$, $\kappa_N^0(x) = \theta(x,0)$ and $g(x) + B\kappa_N^0(x) \oplus \mathbb{W} \subseteq \gamma L$; (iv) For all state sequences $\{x_k\}_{k=0}^{\infty}$ with arbitrary $x_0 \in \mathscr{Z}_N \oplus \gamma L$ and generated by (23) it holds that, for all k,:*

$$\begin{aligned}
&x_k \in z^0(x_k) \oplus \gamma L \subseteq \mathscr{Z}_N \oplus \gamma L \subseteq \mathbb{X},\\
&\kappa_N^0(x_k) = \kappa_N^*(z^0(x_k)) + \theta(x_k, z^0(x_k)) \in \mathbb{V} \oplus \eta M \subseteq \mathbb{U},\\
&V_N^0(x_{k+1}) = V_N^*(z^0(x_{k+1})) \leq V_N^*(g(z^0(x_k)) + B\kappa_N^*(z^0(x_k))),\\
&h_L(z^0(x_k) \oplus \gamma L, \gamma L) \leq \alpha^k \beta |z^0(x_0)|_L\ \text{ and } h_L(\{x_k\}, \gamma L) \leq \alpha^k \beta |z^0(x_0)|_L,
\end{aligned}$$

for some scalars $\alpha \in [0,1)$ and $\beta \in [0,\infty)$.

Under Assumptions 1 and 3 a direct application of results in [16, Section 4] yields that the mapping $\tilde{F}(X) := \{g(x) + B\theta(x,0) + w\ :\ x \in X, w \in \mathbb{W}\}$ is a contraction

on the space of compact subsets of γL (note that Assumption 3 implies $\tilde{F}(\gamma L) \subseteq \gamma L$) and it admits the unique fixed point, namely there exists a compact subset $\mathbb{O}$ of γL such that $\mathbb{O} = \tilde{F}(\mathbb{O})$ and iterates $\tilde{F}^{k+1}(\gamma L) \subseteq \tilde{F}^{k}(\gamma L)$ converge, with respect to the Hausdorff distance, exponentially fast to the set $\mathbb{O}$ as $k \to \infty$. Hence, in addition to assertions of Theorem 1, we have:

Corollary 1. *Suppose Assumptions 1, 3, 4 and 5 hold. Then there exists a compact subset $\mathbb{O}$ of γL such that $\{g(x) + B\kappa_N^0(x) + w \ : \ x \in \mathbb{O}, w \in \mathbb{W}\} = \mathbb{O}$ where $\kappa_N^0(\cdot)$ is given by (23). Furthermore, for all state sequences $\{x_k\}_{k=0}^{\infty}$ with arbitrary $x_0 \in \mathscr{Z}_N \oplus \gamma L$ and generated by (23) it holds that, for all k, $h_L(\{x_k\}, \mathbb{O}) \leq \tilde{\alpha}^k \tilde{\beta} |z^0(x_0)|_L$ for some scalars $\tilde{\alpha} \in [0,1)$ and $\tilde{\beta} \in [0, \infty)$.*

It should be clear that the set $\mathscr{Z}_N$ and functions $\kappa_N^*(\cdot) \ : \ \mathscr{Z}_N \to \mathbb{V}$ and $V_N^*(\cdot) \ : \ \mathscr{Z}_N \to R_+$ appearing in Assumption 5 and utilized in (22) and (23), Theorem 1 and Corollary 1 can be obtained implicitly, under Assumptions 1, 4 and 2 by the standard model predictive control synthesis considered in Section 4 (namely, functions $\kappa_N^*(\cdot)$ and $V_N^*(\cdot)$ can be computed implicitly by solving $\mathbb{P}_N(z)$, specified in (14), on–line and the set $\mathscr{Z}_N$ is given, implicitly, by (15) or alternatively by $\mathscr{Z}_N = \mathscr{B}^N(\mathbb{Z}_f)$ where $\mathbb{Z}_f \subseteq \mathbb{Z}$ is the corresponding terminal constraint set utilized in (14)). Utilizing Proposition 1 and the implicit representation of the set $\mathscr{Z}_N$, given in (15), and functions $\kappa_N^*(\cdot)$ and $V_N^*(\cdot)$, obtained from (14), we provide a formulation of an optimal control problem that when solved on–line provides the implementation of the parameterized tube receding horizon control law (23). Given an integer $N \in \mathbb{N}$, the corresponding parameterized tube optimal control problem $\mathbb{P}_{\text{tube}N}(x)$ is:

$$\begin{aligned} \mathbb{P}_{\text{tube}N}(x): \quad V_N^0(x) &:= \min_{(z,\mathbf{v}_N)} \{V_N(z,\mathbf{v}_N) \ : \ \mathbf{v}_N \in \mathbb{V}_N(z), (x-z) \in \gamma L\} \\ (z,\mathbf{v}_N)^0(x) &:= \arg\min_{(z,\mathbf{v}_N)} \{V_N(z,\mathbf{v}_N) \ : \ \mathbf{v}_N \in \mathbb{V}_N(z), (x-z) \in \gamma L\}. \end{aligned}$$

Note that the tube model predictive control problem $\mathbb{P}_{\text{tube}N}(x)$ is marginally more complex than the conventional model predictive control problem $\mathbb{P}_N(z)$, specified in (14), as it includes z as an additional decision variable and has an additional constraint $(x-z) \in \gamma L$ which, by construction, can be satisfied for all $x \in \mathscr{Z}_N \oplus \gamma L$. The control applied to the system $x^+ = g(x) + Bu + w$, $w \in \mathbb{W}$ at the state $x \in \mathscr{Z}_N \oplus \gamma L$, encountered in the process, is given by:

$$\kappa_N^0(x) = v_0^0(x) + \theta(x, z^0(x));$$

It is, in fact, the on–line implicit implementation of the feedback utilized in (23) and, hence, it ensures, under Assumptions 1, 3, 4 and 2, that properties established in Theorem 1 and Corollary 1 hold for the controlled uncertain system given by $\forall x \in \mathscr{Z}_N \oplus \gamma L$, $x^+ \in \{g(x) + B(v_0^0(x) + \theta(x, z^0(x))) + w \ : \ w \in \mathbb{W}\}$.

Remark 1. All results established above are applicable to the sup–inf case with direct modifications. One of many possible and simple synthesis methods for the sup–inf case would require only changes in Assumption 3 (i.e. the use of function $\theta(\cdot,\cdot,\cdot)$ rather than $\theta(\cdot,\cdot)$ and direct modifications of remaining parts of

Assumption 3) and the utilization of the parameterized sup–inf tube–control policy pair $(\mathbf{X}_{\sup-\inf_N}, \mathit{\Pi}_{\sup-\inf_N})$ where, as in (20) and (21), for any k, we employ $X_{\sup-\inf_k} := z_k \oplus \gamma L$ and for any $(y, \tilde{w}) \in X_{\sup-\inf_k} \times \mathbb{W}$, we consider parameterized control laws $\pi_{\sup-\inf_k}(y, \tilde{w}, z_k) := v_k + \theta(y, \tilde{w}, z_k)$.

6 Concluding Remarks

We highlighted the role of the set theoretic analysis in the model predictive control synthesis and suggested that it provides qualitative insights that are beneficial for the receding horizon control synthesis. We indicated the fragility of the model predictive control and proposed a simple tube model predictive control synthesis method for a particular class of non–linear systems.

Acknowledgements. The author is grateful to E. Crück, S. Olaru and H. Benlaoukli for research interactions leading to an ongoing, collaborative, research project on the fragility of the receding horizon control.

References

1. Witsenhausen, H.S.: A minimax control problem for sampled linear systems. IEEE Transactions on Automatic Control 13, 5–21 (1968)
2. Bertsekas, D.P., Rhodes, I.B.: On the Minimax Reachability of Target Sets and Target Tubes. Automatica 7, 233–247 (1971)
3. Glover, J.D., Schweppe, F.C.: Control of Linear Dynamic Systems with Set Constrained Disturbances. IEEE Transactions on Automatic Control 16(5), 411–423 (1971)
4. Kurzhanski, A.B.: Control and Estimation Under Uncertainty. Nauka, Moscow (1977) (in Russian)
5. Aubin, J.P.: Viability Theory. Birkhäuser, Basel (1991)
6. Blanchini, F., Miani, S.: Set–Theoretic Methods in Control. Birkhäuser, Basel (2008)
7. Krasovski, N.N., Subbotin, A.I.: Game–theoretical Control Problems. Springer, New York (1988)
8. Başar, T., Olsder, J.G.: Dynamic noncooperative game theory. Academic Press, New York (1995)
9. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained Model Predictive Control: Stability and Optimality. Automatica 36(6), 789–814 (2000)
10. Magni, L., De Nicolao, G., Scattolini, R., Allgöwer, F.: Robust Model predictive Control of Nonlinear Discrete–Time Systems. International Journal of Robust and Nonlinear Control 13(3-4), 229–246 (2003)
11. Raković, S.V., Mayne, D.Q.: Robust model predictive control of constrained piecewise affine discrete time systems. In: Proceedings of the 6th IFAC Symposium – NOLCOS 2004, Stuttgart, Germany (September 2004)
12. Raković, S.V., Mayne, D.Q.: Set Robust Control Invariance for Linear Discrete Time Systems. In: Proceedings of the 44th IEEE CDC and ECC conference CDCECC 2005, Sevilla, Spain (2005)
13. Mayne, D.Q., Seron, M.M., Raković, S.V.: Robust model predictive control of constrained linear systems with bounded disturbances. Automatica 41(2), 219–224 (2005)

14. Raković, S.V., Teel, A.R., Mayne, D.Q., Astolfi, A.: Simple Robust Control Invariant Tubes for Some Classes of Nonlinear Discrete Time Systems. In: Proceedings of the 45th IEEE CDC Conference, CDC 2006, San Diego, USA (December 2006)
15. Mayne, D.Q., Kerrigan, E.C.: Tube–Based Robust Nonlinear Model Predictive Control. In: Proceedings of the 7th IFAC Symposium – NOLCOS 2007, Pretoria, South Africa (August 2007)
16. Artstein, Z., Raković, S.V.: Feedback and Invariance Under Uncertainty via Set–Iterates. Automatica 44(2), 520–525 (2008)
17. Raković, S.V.: Minkowski Algebra and Banach Contraction Principle in Set Invariance for Linear Discrete Time Systems. In: Proceedings of 46th IEEE CDC Conference, CDC 2007, New Orleans, USA (2007)
18. Barić, M., Raković, S.V., Besselmann, T., Morari, M.: Max–Min Optimal Control of Constrained Discrete–Time Systems. In: Proceedings of 17th IFAC World Congress, IFAC 2008, Seoul, Korea (2008)
19. Bank, B., Guddat, J., Klatte, D., Kummer, B., Tammer, K.: Non–linear Parametric Optimization. Birkhäuser, Basel (1983)

Adaptive Robust MPC: A Minimally-Conservative Approach

Darryl DeHaan, Martin Guay, and Veronica Adetola

Abstract. Although there is great motivation for adaptive approaches to nonlinear model prediction control, few results to date can guarantee feasible adaptive stabilization in the presence of state or input constraints. By adapting a set-valued measure of the parametric uncertainty within the framework of robust nonlinear-MPC, the results of this paper establish such constrained adaptive stability. Furthermore, it is shown that the ability to account for future adaptation has multiple benefits, including both the ability to guarantee an optimal notion of excitation in the system without requiring dither injection, as well as the ability to incorporate substantially less conservative designs of the terminal penalty.

Keywords: Model Predictive Control, Adaptive Control, Robust Control, Nonlinear Systems.

1 Introduction

Model predictive control (MPC) has rapidly emerged as the control method of choice for systems involving constraints on the system states and inputs. Despite its enormous industrial success, in particular within the process industries, the issue of prediction error due to model uncertainty remains a serious concern for many applications.

From a theoretical perspective, the most well established framework for addressing model uncertainty in MPC is that of a min-max robust MPC formulation (see [10]). Although this approach suffers a significant computational burden, it offers the least conservative control with a guarantee of stability in the presence

Darryl DeHaan
Praxair Inc. 175 East Park Dr Tonawanda, NY, USA 14221

Martin Guay and Veronica Adetola
Dept. Chemical Engineering, Queen's University, Kingston Ontario Canada K7L 3N6
e-mail: martin.guay@chee.queensu.ca

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 55–67.
springerlink.com

of significant disturbance. Of particular relevance is "feedback-MPC" (see [12], amongst others), in which the optimization solves for the optimal *feedback policy* rather than an open-loop input trajectory. By accounting for the effects of future feedback decisions on disturbance attenuation, conservativeness is significantly reduced. However, a significant downside to robust control approaches in general lies in that treating static model uncertainties as an exogenous disturbance is inevitably conservative.

Classically, a second framework for addressing model uncertainty is that of adaptive control. Applicable for systems with unknown constant, or slowly time-varying, parameters, a nominal parameter estimate is generally adapted online until it converges to the true system model. Unfortunately, incorporating adaptive control into an MPC framework has, to date, met with only limited success. For example, the approach in [9] imposes an excitation condition on the prediction trajectories, showing that under ideal circumstances, the parameter estimator will converge in some finite time T_θ, after which the controller becomes stabilizing. However, while $t < T_\theta$ the prediction error may result in poor performance, including violation of state constraints. The approach in [1] uses a global ISS-CLF to limit the impact of such prediction errors, with the obvious downside being the dearth of design approaches to construct such a CLF, in particular in the presence of constraints.

This work takes a very different perspective on the role of adaptation in MPC. Rather than adapting a nominal estimate $\hat{\theta} \in \mathbb{R}^p$ of the uncertain parameter θ, instead the uncertainty set $\hat{\Theta} \subseteq \mathbb{R}^p$ (within which θ is known to lie) is gradually contracted. This notion of adapting the uncertainty set has similarities to hierarchical approaches to adaptive control of nonlinearly-parameterized systems, such as explored in [3]. It is also related to the unfalsified control approach [11], a data-driven direct adaptive control approach based on the falsification of sets of controller parameters. By imbedding this within a robust MPC framework, the controller combines the strong guarantees of stability and feasibility of robust MPC with the ability of an adaptive mechanism to improve performance over time. Extending the concept of feedback-MPC, maximum performance is achieved by accounting for future adaptation via optimizing over classes of *adaptive* feedbacks. An alternate perspective favoring tractability over performance is found in [5].

This paper is organized as follows. Section 2 gives the statement of the problem, while Section 3 develops the proposed adaptive MPC framework. Some practical considerations are discussed in Section 4, with a discussion of robustness in Section 5 followed by brief conclusions in Section 6.

1.1 Notation and Mathematical Preliminaries

For any quantity $s \in \mathbb{R}^s$, the notation $s_{[a,b]}$ denotes its continuous time trajectory $s(\tau)$, $\tau \in [a,b]$, and $\dot{s}(\tau)$ denotes its forward directional derivative. For any set $\mathbb{S} \subseteq \mathbb{R}^s$, denote

1) its interior $\text{int}\{\mathbb{S}\}$, and closure $\text{cl}\{\mathbb{S}\}$
2) its boundary $\partial\mathbb{S} \triangleq \text{cl}\{\mathbb{S}\} \setminus \text{int}\{\mathbb{S}\}$

3) its orthogonal distance function $d_{\mathbb{S}}(s) \triangleq \inf_{s' \in \mathbb{S}} \|s - s'\|$
4) a closed ε-neighbourhood $B(\mathbb{S}, \varepsilon) \triangleq \{s \in \mathbb{R}^s \,|\, d_{\mathbb{S}}(s) \leq \varepsilon\}$
5) an interior approximation $\overleftarrow{B}(\mathbb{S}, \varepsilon) \triangleq \{s \in \mathbb{S} \,|\, d_{\partial\mathbb{S}}(s) \geq \varepsilon\}$
6) a (finite, closed, open) cover of $\mathbb{S}$ as any (finite) collection $\{\mathbb{S}^i\}$ of (open, closed) sets $\mathbb{S}^i$ whose union contains $\mathbb{S}$.
7) the maximal closed subcover $\overline{\text{cov}}\{\mathbb{S}\}$ as the infinite collection $\{\mathbb{S}^i\}$ containing *all possible* closed subsets $\mathbb{S}^i \subseteq \mathbb{S}$; i.e. $\overline{\text{cov}}\{\mathbb{S}\}$ is a maximal "set of subsets".

A function $S : \mathbb{S} \to (-\infty, \infty]$ is *lower semi-continuous* ($\mathscr{LS}$-continuous) [4] at s if it satisfies

$$\liminf_{s' \to s} S(s') \geq S(s) \tag{1}$$

A continuous function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is *class* $\mathscr{K}_\infty$ if i) $\alpha(0) = 0$, ii) α strictly increases, and iii) $\lim_{s \uparrow \infty} \alpha(s) = \infty$.

2 Problem Description

The problem of interest in this work is to achieve robust regulation, by means of state-feedback, of the system state to some compact target set $\Sigma_x^o \in \mathbb{R}^n$. The state and input trajectories are required to obey point-wise constraints $(x, u) \in \mathbb{X} \times \mathbb{U}$, and optimality is measured with respect to accumulation of an instantaneous cost $L(x, u) \geq 0$.

It is assumed that the system dynamics are not fully known, with uncertainty stemming from both unmodelled static nonlinearities as well as additional exogenous inputs. As such, the dynamics are assume to be of the general form

$$\dot{x} = f(x, u, \theta, d(t)) \tag{2}$$

where f is a locally Lipschitz function of state $x \in \mathbb{R}^n$, control input $u \in \mathbb{R}^m$, disturbance input $d \in \mathbb{R}^d$, and constant parameters $\theta \in \mathbb{R}^p$. The entries of θ may represent physically meaningful model parameters (whose values are not exactly known *a-priori*), or alternatively they may represent parameters associated with a (finite) set of universal basis functions. The disturbance $d(t)$ represents the combined effects of actual exogenous inputs, neglected system states, or static nonlinearities lying outside the span of θ (such as truncation error resulting from a finite basis).

Assumption 1. *$\theta \in \Theta^o$, a known compact subset of $\mathbb{R}^p$.*

Assumption 2. *$d(\cdot) \in \mathscr{D}_\infty$, the set of all right-continuous $\mathscr{L}^\infty$-bounded functions $d : \mathbb{R} \to \mathscr{D}$; i.e. composed of continuous subarcs $d_{[a,b)}$, and satisfying $d(\tau) \in \mathscr{D}$, with $\mathscr{D} \subset \mathbb{R}^d$ a compact vectorspace, $\forall \tau \in \mathbb{R}$.*

Unlike much of the robust or adaptive MPC literature, we will not necessarily assume exact knowledge of the system equilibrium manifold, or its stabilizing equilibrium control map. Instead, we make the following (weaker) set of

assumptions, which essentially imply that for any (unknown) θ, there must exist a subset of Σ_x^o that can be rendered robustly invariant by adaptive feedback of the form $u = k_\Sigma(x, \hat{\theta}) \in \Sigma_u^o$, using any $\hat{\theta}$ in a sufficiently small neighborhood of θ^*.

Assumption 3. *Letting $\Sigma_u^o \subseteq \mathbb{U}$ be a chosen compact set, assume $L : \mathbb{X} \times \mathbb{U} \to \mathbb{R}_{\geq 0}$ is continuous, $L(\Sigma_x^o, \Sigma_u^o) \equiv 0$, and $L(x,u) \geq \alpha_L\big(d_{\Sigma_x^o \times \Sigma_u^o}(x,u)\big)$, $\alpha_L \in \mathscr{K}_\infty$.*

Definition 1. *For each $\Theta \subseteq \Theta^o$, let $\Sigma_x(\Theta) \subseteq \Sigma_x^o$ denote the maximal (strongly) control-invariant subset for the differential inclusion $\dot{x} \in f(x,u,\Theta,\mathscr{D})$, using only controls $u \in \Sigma_u^o$.*

Assumption 4. *$\exists N_\Sigma \leq \infty$, and a finite minimal cover of Θ^o (not necessarily unique), denoted $\{\Theta\}^\Sigma$, such that*

i. the collection $\{int\{\Theta\}\}^\Sigma$ is an open cover for $int\{\Theta^o\}$.
ii. $\Theta \in \{\Theta\}^\Sigma \implies \Sigma_x(\Theta) \neq \emptyset$.
iii. $\{\Theta\}^\Sigma$ contains N_Σ elements, the smallest number for which any such collection satisfies i. and ii.

3 Adaptive Robust Design Framework

3.1 Adaptation of Parametric Uncertainty Sets

Unlike standard approaches to adaptive control, this work does not involve explicitly generating a parameter estimator $\hat{\theta}$ for the unknown θ. Instead, the focus is on adapting the parametric uncertainty set Θ^o to gradually eliminate regions within which it is guaranteed that θ does *not* lie. To this end, we define the infimal uncertainty set

$$\mathscr{Z}(\Theta, x_{[a,b]}, u_{[a,b]}) \triangleq \{\, \theta \in \Theta \mid \dot{x}(\tau) \in f(x(\tau), u(\tau), \theta, \mathscr{D}),\ \forall \tau \in [a,b] \,\} \tag{3}$$

Since the computation of (3) online is impractical, we assume that the set $\mathscr{Z}$ is approximated online using any estimator Ψ which satisfies the following

Criterion 1. *$\Psi(\cdot,\cdot,\cdot)$ is designed such that for $a \leq b \leq c$*

C1.1 $\mathscr{Z} \subseteq \Psi$
C1.2 $\Psi(\Theta,\cdot,\cdot) \subseteq \Theta$, and closed.
C1.3 $\Psi(\Theta_1, x_{[a,b]}, u_{[a,b]}) \subseteq \Psi(\Theta_2, x_{[a,b]}, u_{[a,b]})$, $\Theta_1 \subseteq \Theta_2$
C1.4 $\Psi(\Theta, x_{[a,b]}, u_{[a,b]}) \supseteq \Psi(\Theta, x_{[a,c]}, u_{[a,c]})$
C1.5 $\Psi(\Theta, x_{[a,c]}, u_{[a,c]}) \equiv \Psi(\Psi(\Theta, x_{[a,b]}, u_{[a,b]}), x_{[b,c]}, u_{[b,c]})$

The set Ψ is an approximation of $\mathscr{Z}$ in two senses: first of all, both Θ^o and Ψ can be restricted to a particular class of finitely-parameterized sets (for example, convex polytopes). Secondly, contrary to the definition of (3), Ψ can essentially be computed by *removing* values from Θ^o as they are determined to violate the differential inclusion. As such, the search for infeasible values can be terminated at any time without violating C1.

The closed loop dynamics of (2) then take the form

$$\dot{x} \in F(x, \kappa_{mpc}(x, \Theta(t)), \theta, d(t)), \quad x(t_0) = x_0 \tag{4a}$$

$$\Theta(t) \in \Psi(\Theta^o, x_{[t_0,t]}, u_{[t_0,t]}) \tag{4b}$$

where $\kappa_{MPC}(x,\Theta)$ represents the MPC feedback policy, detailed in Section 3.2. The (set-valued) controller state Θ could be generated differentially by an update law $\dot{\Theta}$ designed to gradually contract the set (satisfying C1). However, the given statement of (4b) is more general, as it allows for $\Theta(t)$ to evolve discontinuously.

3.2 Feedback-MPC Framework

In the context of min-max robust MPC, it is well known (see [10]) that feedback-MPC provides significantly less conservative performance than standard open-loop MPC implementations, due to its ability to account for the effects of future feedback decisions on disturbance attenuation. In the following, the same principle is applied to also incorporate the effects of future parameter adaptation.

The receding horizon control law in (4) is defined as $\kappa_{mpc}(x,\Theta) \triangleq \kappa^*(0,x,\Theta)$, where $\kappa^* : [0,T] \times \mathbb{R}^n \times \overline{\text{cov}}\{\Theta^o\} \to \mathbb{R}^m$ is a minimizer for the following optimal control problem:

$$J(x,\Theta) \triangleq \min_{\kappa(\cdot,\cdot,\cdot)} \max_{\substack{\theta \in \Theta(0) \\ d(\cdot) \in \mathscr{D}_\infty}} \int_0^T L(x^p, u^p) d\tau + W(x_f^p, \hat{\Theta}_f) \tag{5a}$$

$$s.\forall \tau \in [0,T]$$

$$\frac{d}{d\tau} x^p = f(x^p, u^p, \theta, d), \; x^p(0) = x \tag{5b}$$

$$\hat{\Theta}(\tau) = \Psi_p(\Theta(t), x^p_{[0,\tau]}, u^p_{[0,\tau]}) \tag{5c}$$

$$x^p(\tau) \in \mathbb{X} \tag{5d}$$

$$u^p(\tau) \triangleq \kappa(\tau, x^p(\tau), \hat{\Theta}(\tau)) \in \mathbb{U} \tag{5e}$$

$$x_f^p \triangleq x^p(T) \in \mathbb{X}_f(\hat{\Theta}_f) \tag{5f}$$

$$\hat{\Theta}_f \triangleq \Psi_f(\Theta(t), x^p_{[0,T]}, u^p_{[0,T]}). \tag{5g}$$

In the remainder, we will drop the explicit constraints (5d)-(5f) by extending the domain of L and W as follows:

$$L(x,u) = \begin{cases} L(x,u) < \infty & (x,u) \in \mathbb{X} \times \mathbb{U} \\ +\infty & \text{otherwise} \end{cases} \tag{6a}$$

$$W(x,\Theta) = \begin{cases} W(x,\Theta) < \infty & x \in \mathbb{X}_f(\Theta) \\ +\infty & \text{otherwise} \end{cases} \tag{6b}$$

The parameter identifiers Ψ_p and Ψ_f are similar (possibly identical) to Ψ in (4b), and must satisfy C1 in addition to:

Criterion 2. *For identical arguments,* $\mathscr{Z} \subseteq \Psi \subseteq \Psi_f \subseteq \Psi_p$.

There are two important characteristics which distinguish (5) from a standard (non-adaptive) feedback-MPC approach. First of all, future evolution of $\hat{\Theta}$ in (5c) is fed back into (5b),(5e). The benefits of this feedback are analogous to those of adding state-feedback into the MPC calculation; the resulting cone of potential trajectories $x^p(\cdot)$ is narrowed, resulting in less conservative worst-case predictions. Similarly, parameterizing W and $\mathbb{X}_f$ as functions of $\hat{\Theta}_f$ in (5g) also acts to reduce the conservativism of the cost. More importantly, accounting for $\hat{\Theta}_f$ in W and $\mathbb{X}_f$ may allow for a stabilizing terminal penalty to be constructed when otherwise no such function exists, as will become apparent in later sections.

3.3 Generalized Terminal Conditions

To guide the selection of $W(x_f, \hat{\Theta}_f)$ and $\mathbb{X}_f(\hat{\Theta}_f)$ in (5), it is important to outline (sufficient) conditions under which (4)-(5) can guarantee stabilization to the target Σ_x^o. A set of such conditions for robust MPC is outlined in [10], from which the results of this section are extended.

For reasons that will become clear in Section 4.3, it will be useful to present these conditions in a more general context in which $W(\cdot,\Theta)$ is allowed to be $\mathscr{LS}$-continuous with respect to x, as may occur if W is generated by a switching mechanism. This adds little additional complexity, since (5) is already discontinuous due to constraints.

Criterion 3. *The set-valued terminal constraint function* $\mathbb{X}_f : \overline{cov}\{\Theta^o\} \to \overline{cov}\{\mathbb{X}\}$ *and terminal penalty function* $W : \mathbb{R}^n \times \overline{cov}\{\Theta^o\} \to [0,+\infty]$ *are such that for each* $\Theta \in \overline{cov}\{\Theta^o\}$*, there exists* $k_f(\cdot,\Theta) : \mathbb{X}_f \to \mathbb{U}$ *satisfying*

C3.1 $\mathbb{X}_f(\Theta) \neq \emptyset \Rightarrow \Sigma_x^o \cap \mathbb{X}_f(\Theta) \neq \emptyset$ *and* $\mathbb{X}_f(\Theta) \subseteq \mathbb{X}$ *closed*
C3.2 $W(\cdot,\Theta)$ *is* $\mathscr{LS}$*-continuous w.r.t.* $x \in \mathbb{R}^n$
C3.3 $k_f(x,\Theta) \in \mathbb{U}$, $\forall x \in \mathbb{X}_f(\Theta)$.
C3.4 $\mathbb{X}_f(\Theta)$ *is strongly positive invariant with respect to the differential inclusion* $\dot{x} \in f(x,k_f(x,\Theta),\Theta,\mathscr{D})$.
C3.5 $\forall x \in \mathbb{X}_f(\Theta)$*, and denoting* $\mathscr{F} \triangleq f(x,k_f(x,\Theta),\Theta,\mathscr{D})$,

$$\max_{f \in \mathscr{F}} \left(L(x,k_f(x,\Theta)) + \liminf_{\substack{v \to f \\ \delta \downarrow 0}} \left(\frac{W(x+\delta v,\Theta) - W(x,\Theta)}{\delta} \right) \right) \leq 0$$

Criterion 4. *For any* $\Theta_1, \Theta_2 \in \overline{cov}\{\Theta^o\}$ *s.t.* $\Theta_1 \subseteq \Theta_2$,

C4.1 $\mathbb{X}_f(\Theta_2) \subseteq \mathbb{X}_f(\Theta_1)$
C4.2 $W(x,\Theta_1) \leq W(x,\Theta_2), \quad \forall x \in \mathbb{X}_f(\Theta_2)$

Designing W and $\mathbb{X}_f$ as functions of Θ satisfying Criteria 3 and 4 may appear prohibitively complex; however, the task is greatly simplified by noting that neither criterion requires any notion of continuity of W or $\mathbb{X}_f$ with respect to Θ. A constructive design approach exploiting this fact is presented in Section 4.3.

3.4 Closed-Loop Stability

Theorem 1 (Main result). *Given system (2), target Σ_x^o, and penalty L satisfying Assumptions 1-4, assume the functions Ψ, Ψ_p, Ψ_f, W and $\mathbb{X}_f$ are designed to satisfy Criteria 1-4. Furthermore, let $\mathscr{X}_0 \triangleq \mathscr{X}_0(\Theta^o) \subseteq \mathbb{X}$ denote the set of initial states, with uncertainty $\Theta(t_0) = \Theta^o$, for which (5) has a solution. Then under (4), Σ_x^o is feasibly asymptotically stabilized from any $x_0 \in \mathscr{X}_0$.*

Proof

This proof will follow a direct method of establishing stability by directly proving strict decrease of $J(x(t),\Theta(t))$, $\forall x \notin \Sigma_x^o$. Stability analysis involving $\mathscr{L}\mathscr{S}$-continuous Lyapunov functions [4, Thm5.5] typically involves the proximal subgradient $\partial_p J$ (a generalization of ∇J), an ambiguous quantity in the context of (4b). Instead, this proof will exploit an alternative framework [4, Prop5.3] involving subderivates (generalized Dini-derivatives). Together, the following two conditions can be shown sufficient to ensure decrease of J, where $\mathscr{F} \triangleq f(x,\kappa_{MPC}(x,\Theta(t)),\Theta(t),\mathscr{D})$

i. $\max\limits_{f\in\mathscr{F}} \overrightarrow{D}J(x,\Theta) \triangleq \max\limits_{f\in\mathscr{F}} \liminf\limits_{\substack{v\to f\\ \delta\downarrow 0}} \frac{J(x+\delta v,\Theta(t+\delta))-J(x,\Theta(t))}{\delta} < 0$

ii. $\min\limits_{f\in\mathscr{F}} \overleftarrow{D}J(x,\Theta) \triangleq \min\limits_{f\in\mathscr{F}} \limsup\limits_{\substack{v\to f\\ \delta\downarrow 0}} \frac{J(x-\delta v,\Theta(t-\delta))-J(x,\Theta(t))}{\delta} > 0$

i.e. J is decreasing on both open future and past neighborhoods of t, for all $t \in \mathbb{R}$, where $\overrightarrow{D}J$, $\overleftarrow{D}J \in [-\infty,+\infty]$.

To prove condition i., let x^p, L^p, W^p correspond to any worst-case minimizing solution of $J(x(t),\Theta(t))$, defined on $\tau \in [0,T]$. Additional notations: $\hat{\Theta}_\tau^p \equiv \hat{\Theta}_f(\tau)$ (corresponding to x^p solution); also, $W_\tau^p(\mathbb{S})$ is interpreted $W(x^p(\tau),\mathbb{S})$; i.e. the set $\mathbb{S}$ need not equal $\hat{\Theta}_f(\tau)$. The max lim inf, dropped for clarity, is everywhere implied.

$$
\begin{aligned}
\max_{f\in\mathscr{F}} \overrightarrow{D}J(x,\Theta) &= \tfrac{1}{\delta}\Big[J(x+\delta v,\Theta(t+\delta)) - \textstyle\int_0^T L^p d\tau - W_T^p(\hat{\Theta}_T^p)\Big] \\
&\le \tfrac{1}{\delta}\Big[J(x+\delta v,\Theta(t+\delta)) - \textstyle\int_0^\delta L^p d\tau - \int_\delta^T L^p d\tau - W_T^p(\hat{\Theta}_T^p)\Big] \\
&\quad + \frac{1}{\delta}\Big(W_T^p(\hat{\Theta}_T^p) - W_{T_\delta}^p(\hat{\Theta}_T^p) - \delta L(x_T^p, k_f(x_T^p,\hat{\Theta}_{T_\delta}^p))\Big) \\
&\quad \textit{(by C3.5; append } k_f \textit{ segment on } [T,T_\delta] \textit{ onto } x^p\textit{)} \\
&\le \tfrac{1}{\delta}\Big[J(x+\delta v,\Theta(t+\delta)) - \textstyle\int_\delta^{T_\delta} L^p d\tau - W_{T_\delta}^p(\hat{\Theta}_T^p)\Big] - L^p|_\delta \\
&\le \tfrac{1}{\delta}\Big[J(x^p(\delta),\hat{\Theta}^p(\delta)) - \textstyle\int_\delta^{T_\delta} L^p d\tau - W_{T_\delta}^p(\hat{\Theta}_T^p)\Big] - L^p|_\delta \\
&\le -L(x,\kappa_{MPC}(x,\Theta))
\end{aligned}
$$

The final inequalities are achieved by recognizing:

- the $\int L^p d\tau + W^p$ term is a suboptimal cost on the interval $[\delta, T_\delta]$, starting from the point $(x^p(\delta), \hat{\Theta}_p(\delta))$
- by C2, $\Theta(t+\delta) \triangleq \Psi(\Theta(t), x_{[0,\delta]}, u_{[0,\delta]}) \subseteq \Psi_p(\Theta(t), x_{[0,\delta]}, u_{[0,\delta]})$, along any locus connecting x and $x + \delta v$.
- the $\liminf_v$ applies over all sequences $\{v_k\} \to f$, of which the sequence $\{v(\delta_k) = \frac{x^p(\delta_k) - x}{\delta}\}$ is a member.
- there exists an arbitrary perturbation of the sequence $\{v(\delta_k)\}$ satisfying $\Psi_p(\Theta(t), x_{[0,\delta]}) = \hat{\Theta}^p(\delta)$.
- the $\liminf_v$ includes the limiting cost $J(x^p(\delta), \hat{\Theta}^p(\delta))$ of any such perturbation of $\{v(\delta_k)\}$. This cost is optimal on $[\delta, T_\delta]$, originating from $(x^p(\delta), \hat{\Theta}_p(\delta))$, and hence the bracketed expression is non-positive.

By analogous arguments, condition ii. follows from:

$$\begin{aligned}
&\min_{f \in \mathscr{F}} \overleftarrow{D} J(x, \Theta) \\
&= \max_{f \in \mathscr{F}} \limsup_{\substack{v \to f \\ \delta \downarrow 0}} \frac{1}{\delta} \left[\int_{-\delta}^{T-\delta} L^p d\tau + W^p_{T-\delta}(\hat{\Theta}^p_{T-\delta}) - J(x, \Theta) \right] \\
&\geq \tfrac{1}{\delta} \left[\int_0^T L^p d\tau + W^p_T(\hat{\Theta}^p_{T-\delta}) - J(x, \Theta) \right] + L^p|_{-\delta} \\
&\geq L(x, \kappa_{MPC}(x, \Theta))
\end{aligned}$$

Given the above, and Assumption 3, it follows that $J(t)$ is strictly decreasing on $x \notin \Sigma^o_x$, and non-increasing on $x \in \Sigma^o_x$, from which it follows that $\lim_{t \to \infty}(x, \Theta)$ must converge to an invariant subset of $\Sigma^0_x \times \overline{\text{cov}}\{\Theta^o\}$. Assumption 1 guarantees that such an invariant subset exists, since it implies $\exists \varepsilon^* > 0$ such that $\Sigma_x(B(\theta^*, \varepsilon^*)) \neq \emptyset$, with θ^* the unknown parameter in (2). Continued solvability of (5) as $(x(t), \Theta(t))$ evolve follows by: 1) $x(\tau) \notin \mathscr{X}_0(\Theta(\tau)) \Rightarrow J(\tau) = +\infty$, and 2) if $x(t) \in \mathscr{X}_0(\Theta(t))$ and $x(t') \notin \mathscr{X}_0(\Theta(t'))$, then $(t' - t) \downarrow 0$ contradicts either condition i. at time t, or ii. at time t'. ■

4 Computation and Performance Issues

4.1 Excitation of the Closed-Loop Trajectories

Contrary to much of the adaptive control literature, including adaptive-MPC approaches such as [9], the result of Theorem 1 does not depend on any auxiliary excitation signal, or require any assumptions regarding the persistency or quality of excitation in the closed-loop behaviour.

Instead, the effects of injecting excitation into the input signal are predicted by (5c) and (5g), and thus automatically accounted for in the optimization. In the case

where $\Psi_p \equiv \Psi_f \equiv \Psi$, then the controller generated by (5) will automatically inject the exact type and amount of excitation to optimize the worst-case cost $J(x,\Theta)$; i.e. the closed-loop behaviour (4) could be considered "optimally-exciting". Unlike most *a-priori* excitation signal design methods, excess actuation is not wasted in trying to identify parameters which have little impact on the closed-loop performance (as measured by J).

As Ψ_p and Ψ_f deviate from Ψ, the convergence result of Theorem 1 remains valid. However, in the limit as Ψ_p, Ψ_f approach identity maps, the optimization ceases to account for any benefits of future identification, which means there ceases to be any incentive for the optimal solution to contain additional excitation. The resulting identification will converge more slowly, and result in more conservative control behaviour with a smaller domain of attraction $\mathscr{X}_0$.

4.2 Extension to Open-Loop MPC

While the feedback-MPC framework in Section 3.2 is more general, the above results can also be extended to a more standard open-loop MPC framework. In fact, open-loop MPC is just a special case of (5) in which the minimization is restricted to "feedbacks" of the form $\kappa(\tau,x,\Theta) \triangleq \kappa^{ol}(\tau,\Theta)$, or even $\kappa^{ol}(\tau) \equiv u(\tau)$ (in which case (5c) is omitted, but (5g) still retained). The stability result of Theorem 1 can then be shown to hold as long as the horizon length $T \in [0, T_{max}]$ is added as a minimization variable (without which Assumption 4.5 no longer ensures decrease of the cost). However, in most cases the resulting domain $\mathscr{X}_0^{ol}$ will be significantly smaller than the original $\mathscr{X}_0$, and the overall control much more conservative, albiet easier to compute.

4.3 A Practical Design Approach for W and $\mathbb{X}_f$

Proposition 1. *Let $\{(W^i,\mathbb{X}_f^i)\}$ denote a finitely-indexed collection of terminal function candidates, with indices $i \in \mathscr{I}$, where each pair $(W^i,\mathbb{X}_f^i)$ satisfies Criteria 3 and 4. Then*

$$W(x,\Theta) \triangleq \min_{i\in\mathscr{I}} \{W^i(x,\Theta)\}, \quad \mathbb{X}_f(\Theta) \triangleq \bigcup_{i\in\mathscr{I}} \{\mathbb{X}_f^i(\Theta)\} \tag{7}$$

satisfy Criteria 3 and 4.

Proof

The fact that C4 holds is a direct property of the union and min operations for closed $\mathbb{X}_f^i$, and the fact that the Θ-dependence of individual $(W^i,\mathbb{X}_f^i)$ satisfy C4. For the purposes of C3, the Θ argument is a constant, and is omitted from notation. Properties C3.1 and C3.2 follow directly by (7), the closure of $\mathbb{X}_f^i$, and (1). Define

$$\mathscr{I}_f(x) = \{i \in \mathscr{I} \,|\, x \in \mathbb{X}_f^i \text{ and } W(x) = W^i(x)\}$$

Denoting $\mathscr{F}^i \triangleq f(x,k_f^i(x),\Theta,\mathscr{D})$, the following inequality holds for every $i \in \mathscr{I}_f(x)$:

$$\max_{f^i\in\mathscr{F}^i} \liminf_{\substack{v\to f^i \\ \delta\downarrow 0}} \frac{W(x+\delta v)-W(x)}{\delta} \leq \max_{f^i\in\mathscr{F}^i} \liminf_{\substack{v\to f^i \\ \delta\downarrow 0}} \frac{W^i(x+\delta v)-W(x)}{\delta} \leq -L(x,k_f^i(x))$$

It then follows that $u = k_f(x) \triangleq k_f^{i(x)}(x)$ satisfies C3.5 for any arbitrary selection rule $i(x) \in \mathscr{I}_f(x)$ (from which C3.3 is obvious). Condition C3.4 follows from continuity of the $x(\cdot)$ flows, and observing that by (6), C3.5 would be violated at any point of departure from $\mathbb{X}_f$. ■

Using Proposition 1, it is clear that $W(\cdot,\cdot)$, $\mathbb{X}_f(\cdot)$ can be constructed from a collection of pairs of the form

$$\left(W^i(x,\Theta),\mathbb{X}_f^i(\Theta)\right) = \begin{cases} \left(W^i(x),\mathbb{X}_f^i\right) & \Theta\subseteq\Theta^i \\ (+\infty,\emptyset) & \text{otherwise} \end{cases}$$

Constructively, this could be achieved as follows:

1. **Generate a finite collection $\{\Theta^i\}$ of sets covering Θ^o**
   - The elements of the collection can, and should, be overlapping, nested, and ranging in size.
   - Categorize $\{\Theta^i\}$ in a hierarchical (i.e. "tree") structure such that
     i. level 1 (i.e. the top) consists of Θ^o. (Assuming $\Theta^o \in \{\Theta^i\}$ is w.l.o.g., since $W(\cdot,\Theta^o) \equiv +\infty$ and $\mathbb{X}_f(\Theta^o) = \emptyset$ satisfy Criteria 3 and 4)
     ii. every set in the l'th vertical level is nested inside one or more "parents" on level $l-1$
     iii. at every level, the "horizontal peers" cover Θ^o.
2. **For every set $\Theta^j \in \{\Theta^i\}$, calculate a robust CLF $W^j(\cdot) \equiv W^j(\cdot,\Theta^j)$, and approximate its domain of attraction $\mathbb{X}_f^j \equiv \mathbb{X}_f^j(\Theta^j)$.**
   - Generally, $W^j(\cdot,\Theta^j)$ is selected first, after which $\mathbb{X}_f(\Theta^j)$ is approximated as either a maximal level set of $W^j(\cdot,\Theta^j)$, or otherwise (e.g. via polytopes).
   - Since the elements of $\{\Theta^i\}$ need not be unique, one could actually define multiple $(W^i,\mathbb{X}_f^i)$ pairs associated with the same Θ^j.
   - While certainly not an easy task, this is a standard robust-control calculation. As such, there is a wealth of tools in the robust control and viability literatures (see, for example [2]) to tackle this problem.
3. **Calculate $W(\cdot,\Theta)$ and $\mathbb{X}_f(\Theta)$ online:**
   i. Given Θ, identify indices of all sets which are active: $\mathscr{I}^* = \mathscr{I}^*(\Theta) \triangleq \left\{ j \,|\, \Theta\subseteq\Theta^j \right\}$. Using the hierarchy, test only immediate children of active parents.

ii. Given x, search over the active indices to identify $\mathscr{I}_f^* = \mathscr{I}_f^*(x, \mathscr{I}^*) \triangleq \{j \in \mathscr{I}^* \,|\, x \in \mathbb{X}_f^j\}$. Define $W(x,\Theta) \triangleq \min_{j\in\mathscr{I}^*} W^j(x)$ by testing indices in $\mathscr{I}_f^*$, setting $W(x,\Theta) = +\infty$ if $\mathscr{I}_f^* = \emptyset$.

5 Robustness Issues

One could technically argue that if the disturbance model $\mathscr{D}$ in (2) encompassed *all* possible sources of model uncertainty, then the issue of robustness is completely addressed by the min-max formulation of (5). In practice this is unrealistic, since it is desirable to explicitly consider only significant disturbances (or exclude $\mathscr{D}$ entirely, if Θ sufficiently encompasses dominant uncertainties). The lack of nominal robustness to model error in constrained nonlinear MPC is a well documented problem, as discussed in [7]. In particular, [8, 6] establish nominal robustness (for "perfect-model", discrete-time MPC) in part by implementing the constraint $x \in \mathbb{X}$ as a succession of strictly nested sets. We present here a modification to this approach, as relevant to the current adaptive framework. Due to space restrictions, proof of results in this section are omitted.

In the following, for any $\gamma, \varepsilon \geq 0$ we denote

$$\mathscr{Z}^{\varepsilon,\gamma}(\Theta, x_{[a,b]}, u_{[a,b]}) \triangleq \{\,\theta \in \Theta \,|\, B(\dot{x}, \varepsilon + \gamma\tau_a) \cap f(B(x,\gamma\tau_a), u, \theta, \mathscr{D}) \neq \emptyset,\ \forall \tau\}$$

where $\tau_a \triangleq \tau - a$. We furthermore denote $\mathscr{Z}^{\varepsilon} \equiv \mathscr{Z}^{\varepsilon,0}$, with analogous notations for Ψ, Ψ_f, Ψ_p.

Claim. For any $a < b < c$, $\gamma \geq 0$, and $\varepsilon \geq \varepsilon' \geq 0$, let $x'_{[a,c]}$ be an arbitrary, continuous perturbation of $x_{[a,b]}$ satisfying

i. $\|x'(\tau) - x(\tau)\| \leq \begin{cases} \gamma(\tau - a) & \tau \in [a,b] \\ \gamma(b-a) & \tau \in [b,c] \end{cases}$

ii. $\|\dot{x}'(\tau) - \dot{x}(\tau)\| \leq \begin{cases} \varepsilon - \varepsilon' + \gamma(\tau - a) & \tau \in [a,b] \\ \gamma(b-a) & \tau \in [b,c] \end{cases}$

Then, $\mathscr{Z}^{\varepsilon,\gamma}$ satisfies

$$\mathscr{Z}^{\varepsilon,\gamma}\Big(\mathscr{Z}^{\varepsilon'}(\Theta, x'_{[a,b]}, u_{[a,b]}), x'_{[b,c]}, u_{[b,c]}\Big) \subseteq \mathscr{Z}^{\varepsilon,\gamma}(\Theta, x_{[a,c]}, u_{[a,c]}) \tag{8}$$

Proposition 2. *Assume that the following modifications are made to the design in Section 3:*

i. $W(x,\Theta)$ and $\mathbb{X}_f(\Theta)$ are constructed as per Prop. 1, but with C3.2 strengthened to require the individual $W^i(x,\Theta)$ to be continuous w.r.t $x \in \mathbb{X}_f^i(\Theta)$.

ii. For some $\varepsilon_x > 0$, (6) and (7) are redefined as:

$$\tilde{L}(\tau, x, u) = \begin{cases} L(x,u) & (x,u) \in \overleftarrow{B}(\mathbb{X}, \varepsilon_x \frac{\tau}{T}) \times \mathbb{U} \\ +\infty & \textit{otherwise} \end{cases}$$

$$\tilde{W}^i(x,\Theta) = \begin{cases} W^i(x) & x \in \overleftarrow{B}(\mathbb{X}_f^i(\Theta), \varepsilon_x) \\ +\infty & \textit{otherwise} \end{cases}$$

iii. For some $\varepsilon_f > 0$, C3.4 holds for every inner approximation $\overleftarrow{B}(\mathbb{X}_f^i(\Theta), \varepsilon_x')$, $\varepsilon_x' \in [0, \varepsilon_x]$, where positive invariance is with respect to $\dot{x} \in B(f(x, k_f^i(x,\Theta), \Theta, \mathscr{D}), \varepsilon_f)$

iv. Identifiers of the form $\Psi^{\varepsilon,\gamma}$, $\Psi_f^{\varepsilon,\gamma}$, $\Psi_p^{\varepsilon,\gamma}$, each satisfying C1.1-4, C2, and the statement of Claim 5, are implemented as follows

- *For some $\varepsilon' > 0$, $\Psi^{\varepsilon'}$ replaces Ψ in (4b)*
- *For some $\gamma > 0$ and $\varepsilon > \varepsilon'$, $\Psi_p^{\varepsilon,\gamma}$ and $\Psi_f^{\varepsilon,\gamma}$ replace Ψ_p in (5c) and Ψ_f in (5g), respectively.*

Then for any compact subset $\bar{\mathscr{X}_0} \subseteq \mathscr{X}_0(\Theta^o)$, $\exists c^* = c^*(\gamma, \varepsilon_x, \varepsilon_f, \varepsilon, \varepsilon', \bar{\mathscr{X}_0}) > 0$ such that, for all $x_0 \in \bar{\mathscr{X}_0}$ and disturbances $\|d_2\| \le c^*$, the target Σ_x^o and actual dynamics

$$\dot{x} = f(x, \kappa_{mpc}(x, \Theta(t)), \theta, d(t)) + d_2(t),\ x(t_0) = x_0 \tag{9a}$$

$$\Theta(t) = \Psi^{\varepsilon'}(\Theta^o, x_{[t_0,t]}, u_{[t_0,t]}) \tag{9b}$$

are input-to-state stable (ISS); i.e. there exists $\alpha_d \in \mathscr{K}$ such that $x(t)$ will asymptotically converge to $B(\Sigma_x^o, \alpha_d(c^*))$.

6 Conclusions

In this paper, we have demonstrated a new methodology for adaptive MPC, in which the adverse effects of parameter identification error are explicitly minimized using methods from robust MPC. As a result, it is possible to feasibly address both state and input constraints within the adaptive framework. Another key advantage of this approach is that the effects of future parameter estimation can be incorporated into optimization problem, raising the potential to significantly reduce the conservativeness of the solutions, especially with respect to design of the terminal penalty.

References

1. Adetola, V., Guay, M.: Adaptive receding horizon control of nonlinear systems. In: Proc. IFAC Symposium on Nonlinear Control Systems, Stuttgart, Germany, pp. 1055–1060 (2004)
2. Aubin, J.P.: Viability Theory. In: Systems & Control: Foundations & Applications. Birkhäuser, Boston (1991)
3. Cao, C., Annaswamy, A.: Parameter convergence in systems with a general nonlinear parameterization using a hierarchical algorithm. In: Proc. American Control Conference, pp. 376–381 (2002)
4. Clarke, F.H., Ledyaev, Y.S., Stern, R.J., Wolenski, P.R.: Nonsmooth Analysis and Control Theory. Grad. Texts in Math, vol. 178. Springer, New York (1998)

5. DeHaan, D., Adetola, V., Guay, M.: Adaptive robust MPC: An eye towards computational simplicity. In: Proc. IEEE Conf. on Decision and Control (submitted, 2006), http://chee.queensu.ca/~dehaan/publications.html
6. Grimm, G., Messina, M., Tuna, S., Teel, A.: Nominally robust model predictive control with state constraints. In: Proc. IEEE Conf. on Decision and Control, pp. 1413–1418 (2003)
7. Grimm, G., Messina, M., Tuna, S., Teel, A.: Examples when model predictive control is non-robust. Automatica 40(10), 1729–1738 (2004)
8. Marruedo, D., Alamo, T., Camacho, E.: Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties. In: Proc. IEEE Conf. on Decision and Control, pp. 4619–4624 (2002)
9. Mayne, D.Q., Michalska, H.: Adaptive receding horizon control for constrained nonlinear systems. In: Proc. IEEE Conf. on Decision and Control, pp. 1286–1291 (1993)
10. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: Stability and optimality. Automatica 36, 789–814 (2000)
11. Safonov, M., Tsao, T.: The unfalsified control concept and learning. IEEE Trans. Automat. Contr. 42(6), 843–847 (1997)
12. Scokaert, P.O.M., Mayne, D.Q.: Min-max feedback model predictive control for constrained linear systems. IEEE Trans. Automat. Contr. 43(8), 1136–1142 (1998)

Enlarging the Terminal Region of NMPC with Parameter-Dependent Terminal Control Law

Shuyou Yu, Hong Chen, Christoph Böhm, and Frank Allgöwer

Abstract. Nominal stability of a quasi-infinite horizon nonlinear model predictive control (QIH-NMPC) scheme is obtained by an appropriate choice of the terminal region and the terminal penalty term. This paper presents a new method to enlarge the terminal region, and therefore the domain of attraction of the QIH-NMPC scheme. The proposed method applies a parameter-dependent terminal controller. The problem of maximizing the terminal region is formulated as a convex optimization problem based on linear matrix inequalities. Compared to existing methods using a linear time-invariant terminal controller, the presented approach may enlarge the terminal region significantly. This is confirmed via simulations of an example system.

Keywords: Nonlinear Model predictive control; Terminal invariant sets; Linear differential inclusion; Linear matrix inequality.

1 Introduction

Nonlinear model predictive control (NMPC) is a control technique capable of dealing with multivariable constrained control problems. One of the main stability results for NMPC is the quasi-infinite horizon approach [1, 2]. A remainding issue for QIH-NMPC is how to enlarge the terminal region since the size of the terminal region affects directly the size of the domain of attraction for the nonlinear optimization problem. Many efforts have been made to determine the terminal penalty term and the associated terminal controller such that the terminal region is enlarged.

Shuyou Yu, Christoph Böhm, and Frank Allgöwer
Institute for Systems Theory and Automatic Control, University of Stuttgart, Germany
e-mail: {shuyou,cboehm,allgower}@ist.uni-stuttgart.de

Shuyou Yu and Hong Chen
Department of Control Science and Engineering, Jilin University, China
e-mail: chenh@jlu.edu.cn

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 69–78.
springerlink.com

For nonlinear systems, using either a local polytopic linear differential inclusions (LDI) representation [3] or a local norm-bounded LDI representation [4], the terminal region is obtained by solving an linear matrix inequality (LMI) optimization problem. In [5], a local LDI representation is used as well, and a polytopic terminal region and an associated terminal penalty are computed. Using support vector machine learning [6], freedom in the choice of the terminal region and terminal penalty needed for asymptotic stability is exploited in [6].

Here, we generalize the scheme in [7]. A more general polytopic LDI description is used to capture the nonlinear dynamics and the condition of twice continuous differentiability of the nonlinear system is relaxed to continuous differentiability. The approach results in a parameter-dependent terminal control law. Compared with the use of time-invariant linear state feedback laws, the proposed approach provides more freedom in the choice of the terminal region and terminal cost needed for asymptotic stability. Thus a larger terminal region is obtained.

The remainder of this paper is organized as follows. Section 2 briefly introduces the QIH-NMPC scheme. The condition to calculate terminal region of QIH-NMPC based on linear differential inclusions and the optimization algorithm to maximize the terminal region are proposed in Section 3 and 4. The efficacy of the algorithm is illustrated by a numerical example in Section 5.

2 Preliminaries

Consider the smooth nonlinear control system

$$\dot{x}(t) = f(x(t), u(t)), \quad x(t_0) = x_0, \quad t \geq t_0 \tag{1a}$$

$$z(t) = g(x(t), u(t)), \tag{1b}$$

subject to

$$z(t) \in Z \subset \mathbb{R}^p, \quad \forall t \geq t_0, \tag{2}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ are the state and input vector, and $z(t)$ is the output vector. Denote X and U as the projection of the output vector space Z to the state vector space and the input vector space, respectively.

Fundamental assumptions of (1) are as follows:

A0) The nonlinear fuctions f and g are continuously differentiable, and satisfy $f(0,0) = 0$ and $g(0,0) = 0$. The equilibrium is a hyperbolic fixed point.

A1) System (1) has a unique solution for any initial condition $x_0 \in X$ and any piecewise right-continuous input function $u(\cdot) : [0, T_p] \longrightarrow U$;

A2) $U \subset \mathbb{R}^m$ and $X \subseteq \mathbb{R}^n$ are compact and the point $(0,0)$ is contained in the interior of $X \times U$.

For the actual state $x(t)$, the optimization problem in the QIH-NMPC is formulated as follows [2, 8]:

$$\min_{\bar{u}(\cdot)} J\big(x(t), \bar{u}(\cdot)\big) \tag{3}$$

subject to

$$\dot{\bar{x}} = f(\bar{x},\bar{u}), \quad \bar{x}(t;x(t)) = x(t), \tag{4a}$$
$$\bar{z}(\tau) \in Z, \quad \tau \in [t, t+T_p], \tag{4b}$$
$$\bar{x}(t+T_p;\bar{x}(t)) \in \Omega(\alpha), \tag{4c}$$

where $J(x(t),\bar{u}(\tau,x(t))) = V(\bar{x}(t+T_p);x(t)) + \int_t^{t+T_p} F\big(\bar{x}(\tau;x(t)),\bar{u}(\tau)\big)d\tau$, T_p is the prediction horizon, $\bar{x}(\cdot;x(t))$ denotes the state trajectory starting from the current state $x(t)$ under the control $\bar{u}(t)$. The pair $(\bar{x},\bar{u})$ denotes the optimal solution to the open-loop optimal control problem (3). $F(\cdot,\cdot)$ is the stage cost satisfying the following condition:

A3) $F(x,u) : \mathbb{R}^n \times U \to \mathbb{R}$ is continuous and satisfies $F(0,0) = 0$ and $F(x,u) > 0$, $\forall (x,u) \in \mathbb{R}^n \times U \setminus \{0,0\}$.

In (4), the set $\Omega(\alpha)$ is a neighborhood of the origin and defined as a level set of a positive definite function $V(\cdot)$ as follows

$$\Omega(\alpha) := \{x \in R^n \mid V(x) \le \alpha\}. \tag{5}$$

Moreover, $\Omega(\alpha)$ and $V(x)$ are said to be the terminal region and the terminal penalty respectively, if there exists a continuous local controller $u = \kappa(x)$ such that the following conditions are satisfied:

B0) $\Omega(\alpha) \subseteq X$,
B1) $g(x,\kappa(x)) \in Z$, for all $x \in \Omega(\alpha)$,
B2) $V(x)$ satisfies inequality

$$\frac{\partial V(x)}{\partial x} f(x,\kappa(x)) + F(x,\kappa(x)) \le 0, \qquad \forall x \in \Omega(\alpha). \tag{6}$$

Clearly, $\Omega(\alpha)$ has the following additional properties [8]:

- The point $0 \in \mathbb{R}^n$ is contained in the interior of $\Omega(\alpha)$ due to the positive definiteness of $V(x)$ and $\alpha > 0$,
- $\Omega(\alpha)$ is closed and connected due to the continuity of V in x.
- Since (6) holds, $\Omega(\alpha)$ is invariant for the nonlinear system (1) controlled by local control $u = \kappa(x)$.

The following stability results can be established:

Lemma 1. *[8] Suppose that*

(a) assumptions A0)-A3) are satisfied,
(b) for the system (1), there exist a locally asymptotically stabilizing controller $u = \kappa(x)$, a continuously differentiable, positive definite function $V(x)$ that satisfies (6) for $\forall x \in \Omega(\alpha)$ and a terminal region $\Omega(\alpha)$ defined by (5),
(c) the open-loop optimal control problem described by (3) is feasible at time $t = 0$.

Then, the closed-loop system is nominally asymptotically stable with the region of attraction D being the set of all states for which the open-loop optimal control problem has a feasible solution.

3 Enlarging the Terminal Region of Quasi-infinite Horizon NMPC

In this section we derive a sufficient condition for the calculation of the terminal region and a linear parameter-dependent terminal control law based on a polytopic differential inclusion description of the nonlinear system (1). The constraints under consideration are

$$-\hat{z}_k \le z_k(t) \le \hat{z}_k, \quad k = 1,2,\ldots,p, \quad t \ge t_0, \tag{7}$$

where $z_k(\cdot)$ is the kth element of the outputs, and $\hat{z}_k$ is positive scalar.

We choose the stage cost $F(x,u) = x^T Qx + u^T Ru$ with $0 \le Q \in R^{n\times n}$ and $0 \le R \in^{m\times m}$. Suppose that the Jacobian linearization of the system (1) at the origin is stabilizable. Then a quadratic Lyapunov function and a local region round the equilibrium defined by the level set of the Lyapunov function exist [9] which serve as terminal penalty and terminal region, respectively. Therefore, we choose the terminal region $\Omega(\alpha,P) := \{x \in R^n | x^T Px \le \alpha\}$ which represents an ellipsoid.

3.1 Polytopic Linear Differential Inclusions

Suppose that for each x,u and t there is a matrix $G(x,u,t) \in \Pi$ such that

$$\begin{bmatrix} f(x,u) \\ g(x,u) \end{bmatrix} = G(x,u,t) \begin{bmatrix} x \\ u \end{bmatrix} \tag{8}$$

where $\Pi \subseteq R^{(n+p)\times(n+p)}$. If we can prove that every trajectory of the LDI defined by Π has some property, then we have proved that every trajectory of the nonlinear system (1) has this property. Conditions that guarantee the existence of such a G are $f(0,0) = 0$, $g(0,0) = 0$, and $\begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial u} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial u} \end{bmatrix} \in \Pi$ for all x,u,t [10].

The set Π is called a polytopic linear differential inclusion (PLDI) if Π is described by a list of its vertices [10]

$$\Omega = \mathrm{Co}\left\{ \begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix}, \begin{bmatrix} A_2 & B_2 \\ C_2 & D_2 \end{bmatrix}, \ldots, \begin{bmatrix} A_N & B_N \\ C_N & D_N \end{bmatrix} \right\}, \tag{9}$$

where $\begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}$, $i = 1,2,\ldots,N$ are vertex matrices of the set Π, and N is the number of vertex matrices. Then the nonlinear system (1) can be represented in the form of a linear parameter-varying dynamic system [11]

$$\dot{x}(t) = \sum_{i=1}^{N} \beta_i(\lambda)(A_i x(t) + B_i u(t)), \tag{10a}$$

$$z(t) = \sum_{i=1}^{N} \beta_i(\lambda)(C_i x(t) + D_i u(t)) \tag{10b}$$

where $\lambda \in \mathbb{R}^{n_\lambda}$ is the time-variant parameter vector, and $\beta_i(\lambda)$ are non-negative scalar continuous weighting functions satisfying $\beta_i(\lambda) > 0$ and $\sum_{i=1}^{N} \beta_i(\lambda) = 1$. In the following we denote $\beta(\lambda) = \left[\beta_1(\lambda)\ \beta_2(\lambda)\ \ldots\ \beta_N(\lambda)\right]^T$. Suppose that $K_j \in \mathbb{R}^{m\times n}$ is a time-invariant feedback gain of the ith vertex system, the control law for the whole PLDI system can be inferred as a weighted average of controllers designed for all vertices, i.e.,

$$\kappa(\lambda) = \sum_{j=1}^{N} \beta_j(\lambda) K_j. \tag{11}$$

Substituting (11) into (10), we obtain the closed-loop system

$$\dot{x}(t) = A_{cl}\big(\beta(\lambda)\big)x(t), \tag{12a}$$
$$z(t) = C_{cl}\big(\beta(\lambda)\big)x(t), \tag{12b}$$

with $A_{cl}\big(\beta(\lambda)\big) = \sum_{i=1}^{N}\sum_{j=1}^{N} \beta_i(\lambda)\beta_j(\lambda)(A_i + B_i K_j)$, $C_{cl}\big(\beta(\lambda)\big) = \sum_{i=1}^{N}\sum_{j=1}^{N} \beta_i(\lambda)\beta_j(\lambda)$ $(C_i + D_i K_j)$.

3.2 Terminal Region of NMPC Based on PLDI

Based on the PLDI of nonlinear system (1), the inequality condition (6) can be formulated as a linear matrix inequality (LMI) problem. This is attractive since computationally efficient methods to solve such problems are available [10, 11].

Theorem 1. *For system (12), suppose that there exist a matrix $X > 0$ and matrices Y_j such that*

$$\sum_{i=1}^{N}\sum_{j=1}^{N} \beta_i(\lambda)\beta_j(\lambda) \begin{bmatrix} A_iX + B_iY_j + (A_iX + B_iY_j)^T & X & Y_i^T \\ X & -Q^{-1} & 0 \\ Y_j & 0 & -R^{-1} \end{bmatrix} \le 0, \tag{13}$$

Then, with $\kappa(\lambda) = \sum_{j=1}^{N} \beta_j(\lambda) K_j$ as in (11) and $V(x) := x^T P x$, where $P = X^{-1}$ and $K_j = Y_j X^{-1}$, the inequality (6) is satisfied.

Proof: By substituting $P = X^{-1}$ and $Y_j = K_j X$ in (13) and performing a congruence transformation with the matrix $\{X^{-1}, I, I\}$, we obtain

$$\begin{bmatrix} A_{cl}(\lambda)^T P + P A_{cl}(\lambda) & X & \kappa(\lambda)^T \\ X & -Q^{-1} & 0 \\ \kappa(\lambda) & 0 & -R^{-1} \end{bmatrix} \le 0,$$

It follows from the Schur complement that the inequalities (13) are equivalent to

$$A_{cl}(\lambda(t))^T P + P A_{cl}(\lambda(t)) + Q + \kappa(\lambda(t))^T R \kappa(\lambda(t)) \le 0. \tag{14}$$

We choose $V(\xi) = \xi^T P \xi$ as a Lyapunov function candidate. The time derivative of $V(x)$ along the trajectory of (12) is given as follows:

$$\begin{aligned}\frac{dV(x(t))}{dt} &= \dot{x}(t)^T P x(t) + x(t)^T P \dot{x}(t) \\ &= x(t)^T \Big\{ \sum_{i=1}^{N} \sum_{j=1}^{N} h_i(\lambda) h_j(\lambda) \Big((A_i + B_i K_j)^T P + P(A_i + B_i K_j) \Big) \Big\} x(t) \\ &= x(t)^T \Big\{ A_{cl}(\lambda)^T P + P A_{cl}(\lambda) \Big\} x(t) \end{aligned} \tag{15}$$

Using (14), we have $\frac{dV(x(t))}{dt} \leq -x(t)^T Q x(t) - x(t)^T \kappa(\lambda)^T R \kappa(\lambda) x(t)$. Thus the inequality (6) holds, and $\kappa(\lambda)$ is the associated terminal control law. □

Now we derive conditions such that the constraints (7) are satisfied by the controller $\kappa(\lambda)$ for any $x \in \Omega(P, \alpha)$.

Theorem 2. *If X and $Y_j, j = 1, 2, \ldots, N$ satisfy (13) and furthermore the following matrix inequalities*

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \beta_i(\lambda) \beta_j(\lambda) \begin{bmatrix} \frac{1}{\alpha} \hat{z}_k^2 & e_k^T (C_i X + D_i Y_j) \\ * & X \end{bmatrix} \geq 0, k = 1, 2, \ldots, p, \tag{16}$$

hold, where e_k is kth element of the basis vector in the constraint vector space, then for any $x(t) \in \Omega(P, \alpha)$, the parameter-dependent feedback law (11) controls the system (12) satisfying the the constraint (7).

Proof: Using (12b), satisfaction of the constraints (7) requires

$$x(t)^T (C_{cl}(\beta(\lambda)))^T e_k e_k^T C_{cl}(\beta(\lambda)) x(t) \leq \hat{z}_k^2, \tag{17}$$

due to $x(t) \in \Omega(P, \alpha)$, which holds if

$$\frac{x(t)^T (C_{cl}(\beta(\lambda)))^T e_k e_k^T C_{cl}(\beta(\lambda)) x(t)}{\hat{z}_k^2} \leq \frac{x(t)^T P x(t)}{\alpha}, \tag{18}$$

For any $x(t) \neq 0$ (note that $x(t) = 0$ leads to $z(t) = 0$ and satisfaction of (7)), inequality (18) holds if

$$\frac{P}{\alpha} - \frac{(C_{cl}(\beta(\lambda)))^T e_k e_k^T C_{cl}(\beta(\lambda))}{\hat{z}_k^2} \geq 0. \tag{19}$$

Applying the Schur Complement, the matrix inequality (19) is equivalent to

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \beta_i(\lambda) \beta_j(\lambda) \begin{bmatrix} P & * \\ e_k^T (C_i + D_i K_j) & \frac{1}{\alpha} \hat{z}_k^2 \end{bmatrix} \geq 0, \quad k = 1, 2, \cdots, p. \tag{20}$$

Performing a congruence transformation with $\mathrm{diag}(I, X)$ on both sides of (20), we obtain the required inequality (16). □

Following the above discussions, we now state the main results of this paper:

Theorem 3. *Suppose that the PLDI model of the nonlinear system (1) is given by (12). If there exist a positive definite matrix $X \in \mathbb{R}^{n\times n}$, matrices $Y_j \in \mathbb{R}^{m\times n}$, $j = 1,2,\cdots,N$, and a scalar $\alpha > 0$, independent of the unknown parameter vector $\beta(\lambda)$ such that (13) and (16), then the ellipsoid $\Omega(\alpha,P)$ with $P = X^{-1}$ and $V(x) = x^T Px$ serve as a terminal region and a terminal penalty for NMPC, respectively. The associated terminal controller is $\kappa(\lambda) = \sum_{j=1}^{N} \beta_j(\lambda) K_j x(t)$ with $K_j = Y_j X^{-1}$.*

Proof: The inequalities (13) and (16) guarantee that the nonlinear system (1) satisfy inequality (6) and constraints (2), respectively, i.e. B1) and B2).

The positive definite matrix $X \in \mathbb{R}^{n\times n}$, the matrices $Y_j \in \mathbb{R}^{m\times n}$, and the scalar $\alpha > 0$ are independent of the unknown parameter vector $\beta(\lambda)$. Thus $\Omega(\alpha,P)$ is the terminal region and $V(x)$ is the terminal penalty of the nonlinear system, respectively. □

4 Optimization of the Terminal Region

In order to reduce the functional inequalities (13) and (16) to finitely many LMIs, we utilize the following lemma:

Lemma 2. *[12] If there exist matrices $\Gamma_{ii} = \Gamma_{ii}^T$, $\Gamma_{ij} = \Gamma_{ji}^T$, $(i \neq j, i, j = 1,2,\cdots,r)$ such that the matrix $\Lambda_{ij} (1 \le i, j \le r)$*

$$\Lambda_{ii} \le \Gamma_{ii}, \quad i = 1,2,\cdots,r, \tag{21a}$$

$$\Lambda_{ij} + \Lambda_{ji} \le \Gamma_{ij} + \Gamma_{ij}^T, \quad j < i, \tag{21b}$$

$$[\Gamma_{ij}]_{r\times r} \le 0, \tag{21c}$$

then the parameter matrix inequalities

$$\sum_{i=1}^{r} \sum_{j=1}^{r} \delta_i(\cdot)\delta_j(\cdot)\Lambda_{ij} \le 0, \tag{22}$$

are feasible, where $\delta_i(\cdot) \ge 0$, $\sum_{i=1}^{r} \delta_i(\cdot) = 1, \forall t$, and $[\Gamma_{ij}]_{r\times r} = \begin{pmatrix} \Gamma_{11} & \cdots & \Gamma_{1r} \\ \vdots & \ddots & \vdots \\ \Gamma_{r1} & \cdots & \Gamma_{rr} \end{pmatrix}$.

Let $\Omega(\alpha,P)$ denote the ellipsoid centered at the origin defined by P and α. The volume of Ω is proportional to $\det(\alpha X)$, $X = P^{-1}$ [10]. The geometric mean of the eigenvalues [13], leading to minimization of $\det(\alpha X)^{\frac{1}{n}}$, where n is dimension of X, can be used for solving the determinant maximization problem. Define

$$X_0 = \alpha X, \quad Y_{j0} = \alpha Y_j. \tag{23}$$

The inequality constraints (13), (16) can be rewritten as

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \beta_i(\lambda)\beta_j(\lambda)\mathscr{L}_{ij} \le 0, \tag{24}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\beta_i(\lambda)\beta_j(\lambda)\mathscr{F}_{i,j}\geq 0,\quad k=1,2,\ldots,m, \tag{25}$$

where $\mathscr{L}_{ij}=\begin{pmatrix}\Xi & X_0 & Y_{j0}^T\\ * & -\alpha Q^{-1} & 0\\ * & * & -\alpha R^{-1}\end{pmatrix}$, $\mathscr{F}_{i,j}=\begin{bmatrix}\bar{z}_k^2 & e_k^T(C_iX_0+D_iY_{j0})\\ * & X_0\end{bmatrix}$, $\Xi=X_0A_i^T+Y_{j0}^TB_i^T+A_iX_0+B_iY_{j0}$.

It follows from Lemma 2 that if there exist matrices T_{ij} $(i,j=1,\cdots,N)$ such that

$$\mathscr{L}_{ii}\leq\mathscr{T}_{ii},\quad i=1,2,\cdots,N, \tag{26a}$$
$$\mathscr{L}_{ij}+\mathscr{L}_{ji}\leq\mathscr{T}_{ij}+\mathscr{T}_{ij}^T,\quad j<i, \tag{26b}$$
$$[\mathscr{T}_{ij}]_{N\times N}\leq 0, \tag{26c}$$

then the inequality (6) is satisfied. Furthermore, if there exist matrices M_{ij} $(i,j=1,2,\cdots,N)$ such that

$$\mathscr{F}_{ii}\geq\mathscr{M}_{ii},\quad i=1,2,\cdots,N, \tag{27a}$$
$$\mathscr{F}_{ij}+\mathscr{F}_{ji}\geq\mathscr{M}_{ij}+\mathscr{M}_{ij}^T,\quad j<i, \tag{27b}$$
$$[\mathscr{M}_{ij}]_{N\times N}\geq 0, \tag{27c}$$

then the output constraints (7) are satisfied.

Hence, the maximization problem of the ellipsoid Ω can be reformulated as

$$\max_{\alpha,\,X_0,\,Y_{j0}}(\det X_0)^{\frac{1}{n}},\ \text{s.t. } \alpha>0,\ X_0>0,\ (26)\text{ and }(27). \tag{28}$$

Solving the convex optimization problem (28), the optimal solutions $X_0, Y_{j0}, (j=1,\cdots,N)$ and α are determined. The matrices X and Y_j can be found from (23). Then the optimal terminal penalty matrix P, the terminal region Ω, and the terminal feedback law can be determined by Theorem 3. Sometimes solving the optimization problem (28) gives a very large terminal penalty such that the effect of the integral term in the performance index (3) almost disappears. A very strong penalty on the terminal states may have a negative influence on achieving the desired performance which is specified by the finite horizon cost [2]. The trade off between a large terminal region and good control performance can be made by limiting the norm of the matrix P [3]. Since $P=X^{-1}=\alpha(X_0)^{-1}$, it can be achieved by imposing the requirement that α has to be less than or equal to a given constant.

5 A Numerical Example

In this section, the proposed method is applied to a continuous stirred tank reactor(CSTR) [14]. Assuming constant liquid volume, the CSTR for an exothermic, irreversible reaction, A→B, is described by the following dynamic model based on a component balance for reactant A and an energy balance:

$$\dot{C}_A = \frac{q}{V}(C_{Af} - C_A) - k_0 \exp(-\frac{E}{RT})C_A,$$
$$\dot{T} = \frac{q}{V}(T_f - T) - \frac{\triangle H}{\rho C_p} k_0 \exp(-\frac{E}{RT})C_A + \frac{UA}{V\rho C_p}(T_c - T).$$

here C_A is the concentration of the reactor, T is the reactor temperature, and T_c is the temperature of the coolant steam. The parameters are $q = 100$ l/min, $V = 100$ l, $C_{Af} = 1$ mol/l, $T_f = 350$ K, $\rho = 10^3$ g/l, $C_p = 0.239$ J/(g K), $k_0 = 7.2 \times 10^10$ min^{-1}, $E/R = 8750$ K, $\triangle H = -5 \times 10^4$ J/mol, $UA = 5 \times 10^4$ J/(min K). Under these conditions the steady state is $C_A^{eq} = 0.5$ mol/l, $T_c^{eq} = 300$ K, and $T^{eq} = 350$ K, which is an unstable equilibrium. The temperature of the coolant steam is constrained to $250\,\text{K} \leq T_c \leq 350\,\text{K}$. The concentration of the reactor has to satisfy $0 \leq C_A \leq 1$ mol/l, and the temperature of the reactor is constrained to $300\,\text{K} \leq T \leq 400\,\text{K}$. The objective is to regulate the concentration C_A and the reactor steam temperature T around the steady state via NMPC by using the temperature of the coolant as an input, while the constraints have to be hold. The dynamics of the CSTR can be expressed by the parameter-dependent weighted linear model of the nonlinear system (10) with $A_1 = \begin{bmatrix} -23.7583 & 0 \\ 4761.2 & -739/239 \end{bmatrix}$, $A_2 = \begin{bmatrix} -1.0155 & 0 \\ 3.2433 & -739/239 \end{bmatrix}$, $B_1 = B_2 = \begin{bmatrix} 0 & 500/239 \end{bmatrix}^T$. The weighting matrices of the stage cost are $Q = \begin{bmatrix} \frac{1}{0.5} & \frac{1}{350} \end{bmatrix}$ and $R = \frac{1}{300}$, respectively.

The volume of the terminal region of the proposed method is compared to previous results which were based on a Lipschitz approach [2]. In order to preserve a dominating effect of the integral part in the cost function, we impose the constraint $\alpha \leq 5$ on the optimization problem (28). The terminal region given by [2] is represented by the dashed ellipsoid, and the terminal region yielded by the PLDI

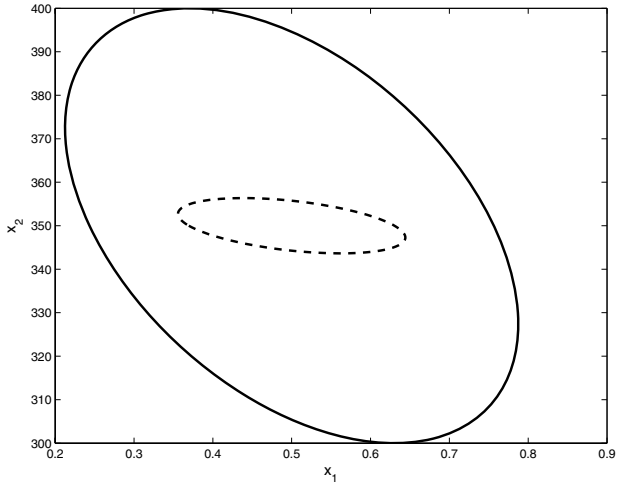


Fig. 1 Comparison of the terminal region

approach with parameter-dependent terminal control law is shown by the solid ellipsoid in Figure 1. The associated terminal penalty is $P = \begin{bmatrix} 15.2100 & 0.0395 \\ 0.0395 & 0.0005 \end{bmatrix}$.

6 Conclusions

In this paper we propose a method to expand the terminal region which replaces the time invariant linear state feedback control law by a parameter-dependent terminal control law. The new algorithm provides an extra degree of freedom to enlarge the terminal set. The problem of maximizing the terminal region is formulated as an LMI based optimization problem. It is shown that, compared to the algorithms with static linear terminal control law, a parameter-dependent terminal control results in a larger terminal region, which is confirmed by a numerical example.

References

1. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: Stability and optimality. Automatica 36(6), 789–814 (2000)
2. Chen, H., Allgöwer, F.: A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. Automatica 34(10), 1205–1217 (1998)
3. Chen, W.H., O'Relly, J., Ballance, D.J.: On the terminal region of model predictive control for non-linear systems with input/state constraints. Int. J. Control Signal Process 17, 195–207 (2003)
4. Yu, S.-Y., Chen, H., Zhang, P., Li, X.-J.: An LMI optimization approach for enlarging the terminal region of NMPC. Acta Automatica Sinca 34, 798–804 (2008)
5. Cannon, M., Deshmukh, V., Kouvaritakis, B.: Nonlinear model predictive control with polytopic invariant sets. Automatica 39, 1487–1494 (2003)
6. Ong, C., Sui, D., Gilbert, E.: Enlarging the terminal region of nonlinear model predictive control using the supporting vector machine method. Automatica 42, 1011–1016 (2006)
7. Yu, S.-Y., Chen, H. Li, X.-J.: Enlarging the terminal region of NMPC based on T-S fuzzy model. Int. J. Control, Automation, and Systems (accepted, 2009)
8. Chen, H.: Stability and Robustness Considerations in Nonlinear Model Predictive Control. usseldorf: Fortschr.-Ber. VDI Reihe 8 Nr. 674, VDI Verlag (1997)
9. Khalil, H.: Nonlinear Systems (third version). Prentice-Hall, New York (2002)
10. Boyd, S., El Ghaoui, L., Feron, E., Balakishnan, V.: Linear Matrix Inequalities in System and Control Theory. SIAM, Philadelphia (1994)
11. Scherer, C.W., Weiland, S.: Linear Matrix Inequalities in Control, DISC Lecture note, Dutch Institute of Systems and Control (2000)
12. Gao, X.: Control for T-S fuzzy systems based on LMI optimization. Dissertation, Jilin University (2006) (in Chinese)
13. Nesterov, Y., Nemirovsky, A.: Interior point polynomial methods in convex programming. SIAM Publications, Philadelphia (1994)
14. Magni, L., Nicolao, G.D., Scattolini, R.: A stabilizing model-based predictive control algorithm for nonlinear systems. Automatica 37, 1351–1362 (2001)

Model Predictive Control with Control Lyapunov Function Support

Keunmo Kang and Robert R. Bitmead

Abstract. A new idea to construct stabilizing model predictive control is studied for a constrained system based on the adaptation of an existing stabilizing controller with a Control Lyapunov Function. We focus on systems which are difficult to stabilize via classical model predictive control because the initial state can be so large that the origin is not reachable in a limited time horizon. We handle this by using a varying terminal state equality constraint, which eventually converges to the neighborhood of the origin. Open-loop, pre-computed terminal state trajectories and closed-loop variants are developed and compared to Artstein's controllers. In the closed-loop case, it is shown that the model predictive approach leads to improved degree of stability over the original stabilizing control law.

Keywords: stabilization, control lyapunov function, terminal constraint.

1 Introduction

We study the stabilization of a constrained discrete-time system, whose behavior is limited by input and state constraints. In particular, we focus on a Model Predictive Control (MPC) scheme for the situation where the system is not effectively stabilized by typical MPC with the zero-terminal-state [7, 10] or terminal set around the origin [1, 4] because the initial state might be too far from the origin. Here we consider a different strategy that consists of two stages:

– solving a finite sequence of finite-horizon open-loop optimal control problems with a varying terminal state equality constraint: the terminal target state changes at each time step and moves closer to the origin, then

Keunmo Kang
United Technologies Research Center, 411 Silver Ln., East Hartford, CT 06109, USA
e-mail: kangk@utrc.utc.com

Robert R. Bitmead
Department of Mechanical and Aerospace Engineering, University of California San Diego, 9500 Gilman Dr., La Jolla, CA 92093, USA
e-mail: rbitmead@ucsd.edu

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 79–87.
springerlink.com

– solving a sequence of finite-horizon open-loop optimal control problems with the origin as the target terminal state to achieve closed-loop stability.

The first stage does not appear in typical MPC contexts and it functions to manipulate the system into a situation where typical zero-terminal-state MPC such as the second stage can be applied. We will show that, if the system has a known stabilizing controller with a Control Lyapunov Function (CLF), then the above idea can be feasibly realized, achieving an improved convergence rate over the original controller.

The authors of [2, 11] discussed the idea of including a CLF in the MPC formulation and pointed out its possible performance improvement due to complementary aspects of a CLF and MPC. In this paper, by example, we demonstrate that our scheme may also bring improvement to control performance and give flexibility to tune system behavior. Comparisons are drawn with [11].

2 System Description and Problem

Consider the following discrete time-invariant system

$$x^+ = f(x,u), \tag{1}$$

where the x^+ is the successor state and the state and input are constrained to

$$x \in \mathbf{X} \subset \mathbb{R}^n,\ u \in \mathbf{U} \subset \mathbb{R}^m.$$

We assume that perfect full-state measurement is available and $0 = f(0,0)$. Consider a constrained finite-horizon optimal control problem with the current state $x \in \mathbf{X}$, the horizon N, and the terminal target state $\bar{x} \in \mathbf{X}$,

$$\begin{aligned} \mathscr{P}(x,\bar{x},N) : \arg \min_{\{u(0,x),\ldots,u(N-1,x)\}} & \sum_{i=0}^{N-1} h(x(i,x),u(i,x)), \\ \textit{subject to}\ \ & x(i+1,x) = f(x(i,x),u(i,x)), \\ & x(i,x) \in \mathbf{X}, \\ & u(i,x) \in \mathbf{U}, \\ & x(N,x) = \bar{x}. \end{aligned} \tag{2}$$

Here $x(i,x)$ $[x(0,x) = x]$ and $u(i,x)$ represent the $i-$step-ahead state and control, which are computed for the given current state x. We assume the following.

Assumption A1. $\mathbf{X}$ is closed and $0 \in \mathbf{X}$,
Assumption A2. $\mathbf{U}$ is closed and $0 \in \mathbf{U}$,
Assumption A3. $f : \mathbf{X} \times \mathbf{U} \longmapsto \mathbb{R}^n$ is continuous and $f(0,0) = 0$,
Assumption A4. $h : \mathbf{X} \times \mathbf{U} \longmapsto \mathbb{R}$ is nonnegative definite, upper-semi-continuous, and satisfies $h(0,0) = 0$.

Closure of $\mathbf{X}$ and $\mathbf{U}$, continuity of f and h, and positive definiteness of h are the conditions for $\mathscr{P}(x,\bar{x},N)$ to have an optimal solution provided there exists an admissible sequence pair $\{x(i,x),u(i,x)\}_{i=0}^{N-1}$ [6]. If Assumptions A1∼4 hold, $\mathscr{P}(x,\bar{x},N)$ has an optimal open-loop control solution sequence, $\{u^*(i,x)\}_{i=0}^{N-1}$ and a corresponding predicted state sequence $\{(x^*(i,x)\}_{i=0}^{N}$ with $x(0,x)=x$ and $x^*(N,x)=\bar{x}$. In classical MPC, at each time step, $\mathscr{P}(x,\bar{x},N)$ with $\bar{x}=0$ is solved and only the first element $u^*(0,x)$ of the control solution is applied to the system. By doing this, the cost of $\mathscr{P}(x,0,N)$ decreases over time and the system is asymptotically stabilized [7], provided that the origin is feasibly reachable from the initial state in N steps. Quite naturally the following question arises: what can we do for initial states that cannot be steered to the origin in N steps? We want to develop an MPC scheme to deal with this situation by using the same structure as (2), in which case $\bar{x}$ might be nonzero and varying every time $\mathscr{P}(x,\bar{x},N)$ is solved for a new x. Therefore the main focus is on how we construct the sequence of terminal target states $\{\bar{x}\}$ that preserve feasibility of $\mathscr{P}(x,\bar{x},N)$ over time and finally achieve asymptotic stability.

3 Model Predictive Control Algorithm

Definition 1. *A scalar function $V(x)$ is said to be a Control Lyapunov Function (CLF) for the system (1) in the domain $\mathbf{X}^V$, containing the origin, if $\mathbf{X}^V \subseteq \mathbf{X}$ and there exists a function $V(x) : \mathbf{X}^V \mapsto \mathbb{R}$ such that*

- *V is continuous,*
- *V is positive definite and $V(0) = 0$,*
- *for each $x \in \mathbf{X}^V$, there exists a feasible control $u \in \mathbf{U}$ such that $x^+ = f(x,u) \in \mathbf{X}^V$ and $V(x^+) \leq V(x) - \beta(||x||)$, where $||x|| \triangleq \sqrt{x^T x}$ and β is positive definite, continuous, and satisfies $\beta(0) = 0$.*

Definition 2. *The subset of $\mathbf{X}$, $\mathscr{X}(0,N_1)$, is the set of states, from which the origin is reachable in N_1 steps.*

Now the MPC algorithm is given for a given initial state $x_0 \in \mathbf{X}^V \subseteq \mathbf{X}$, where $\mathbf{X}^V$ is the known domain.

Offline Preparation

`choose` control horizons for the first stage (N) and the second stage ($N+N_1$),
`choose` any feasible $\bar{x}=\bar{x}_0 \in \mathbf{X}^V \subseteq \mathbf{X}$ based on the initial state $x=x_0 \in \mathbf{X}^V \subseteq \mathbf{X}$,
`compute` $\mathscr{X}(0,N_1) \in \mathbf{X}$.

Control Lyapunov Function Based Model Predictive Control Law

[First Stage] `while` $\bar{x} \notin \mathscr{X}(0,N_1)$
`solve` $\mathscr{P}(x,\bar{x},N)$,
`choose` $\bar{x}^+$ such that $V(\bar{x}^+) \leq V(\bar{x}) - \beta(||\bar{x}||)$ with a positive definite function β and it is feasibly reachable from $x^*(1,x)$,
`apply` $u^*(0,x)$ to the system: $x^+ = x^*(1,x) = f(x,u^*(0,x))$,
`update` $x \leftarrow x^+$ and $\bar{x} \leftarrow \bar{x}^+$,
`end`

[Second Stage] `repeat`
`solve` $\mathscr{P}(x,0,N_1+N)$,
`apply` $u^*(0,x)$ to the system,
`update` $x \leftarrow x^+$,
`end`

Finite time termination of the first stage and asysmptotic stability by the second stage are described by the following theorem.

Theorem 1. *Suppose that the following hold:*

- *Assumptions A1∼4.*
- *a CLF, $V(x)$, with domain $\mathbf{X}^V \subseteq \mathbf{X}$ is given.*
- *initial problem $\mathscr{P}(x_0,\bar{x}_0,N)$ with $x{=}x_0{\in}\mathbf{X}^V$ and $\bar{x}{=}\bar{x}_0{\in}\mathbf{X}^V$ is feasible.*
- *the origin is an interior point of $\mathscr{X}(0,N_1)$.*

Then the CLF Based MPC Law remains feasible and drives the terminal target state $\bar{x}$ in finite time to $\mathscr{X}(0,N_1)$, the set of states from which the origin is feasibly reachable in N_1 steps, and the system is asysmptotically stabilized.

Proof: First, we show that, if the initial problem $\mathscr{P}(x_0,\bar{x}_0,N)$ is feasible, then feasibility of $\mathscr{P}(x,\bar{x},N)$ throughout the first stage is guaranteed. Since $V(x)$ is a CLF and $\bar{x}_0 \in \mathbf{X}^V$, there exists $\bar{x}^+$ such that

$$V(\bar{x}^+) \leq V(\bar{x}_0) - \beta(||\bar{x}_0||). \tag{3}$$

Then, for the next time step, $x^+ = x^*(1,x_0)$ can be steered to $\bar{x}^+$ in (3) by the following feasible state trajectory:

$$\{x^+ = x^*(1,x_0), x^*(2,x_0), \ldots, \bar{x}_0 = x^*(N,x_0), \bar{x}^+\}. \tag{4}$$

Hence, provided that the initial $\mathscr{P}(x_0,\bar{x}_0,N)$ is feasible, feasibility is guaranteed for all time. Now it remains to show that successive update of $\bar{x}$ in the first stage results in $\bar{x} \in \mathscr{X}(0,N_1)$ in finite time. By a standard Lyapunov stability argument, (3) implies that $\bar{x}_j^+ \to 0$ as $j \to \infty$, where $\bar{x}_j^+$ is the terminal target state after j time-updates from the initial terminal target state $\bar{x}_0$. That is, given sufficiently small $\gamma > 0$ such that $\{x \big| ||x|| \leq \gamma\} \subseteq \mathscr{X}(0,N_1)$, there exists an finite integer M_1 such that $||\bar{x}_j^+|| \leq \gamma$ from all $j \geq M_1$. This leads to the finite termination of the first stage. Once the first stage terminates, asymptotic stability of the system is achieved by the second stage as classically established in [7] by the zero terminal state constraint. □

Remark 1. If the set $D \subseteq \mathbf{X}$ is any control invariant set for the system, such as $\mathbf{X}^V$, using the tools of [8], one might construct a larger set of states from which D is feasibly reachable in N steps. Hence, one may expand the domain of attraction of $\mathbf{X}^V$. Such a construction might be used to expand the permissible set of initial states.

Remark 2. There is no requirement that $x(i,x) \in \mathbf{X}^V$, $\forall i = 1,2,\ldots,N-1$. Only the current and terminal states need to be in $\mathbf{X}^V$.

Remark 3. The requirement for the additional N_1 horizon steps in the second stage is that its corresponding $\mathscr{X}(0,N_1)$ has the origin as an interior point. This requires that no proper subspace of the state space be outside the N_1-step reachable set, which observation in the linear case links the minimal choice of N_1 to the rank of the reachability matrix of the system for the various horizons. The horizon value N, because of the CLF property of V, may be chosen as any positive integer. Its selection represents a balance between the computational resources and the benefits inherited by MPC through having longer horizon, which we will next show includes the capacity to improve control performance.

4 Choice of $\bar{x}^+$

If a CLF $V(x)$ is given in the domain $\mathbf{X}^V$, then we may associate with it control laws determined by two methods to choose $\bar{x}^+$. As stated in the algorithm, the requirements for $\bar{x}^+$ are that $V(\bar{x}^+) \leq V(\bar{x}) - \beta(||\bar{x}||)$ and $\bar{x}^+$ is feasibly reachable from $x^*(1,x)$.

4.1 Open-Loop Approach

The first approach is to use the current terminal target state $\bar{x}$ to determine the next time's terminal target state $\bar{x}^+$ via:

$$\bar{x}^+ = \arg\min_x \{V(x) \big| x \in f(\bar{x},\mathbf{U}) \cap \mathbf{X}^V\}, \tag{5}$$

where $f(\bar{x},\mathbf{U}) \triangleq \{x | x = f(\bar{x},\bar{u}), \forall \bar{u} \in \mathbf{U}\}$, the one-step reachable set from $\bar{x}$. The stabilizing property of the CLF Based MPC is summarized by the following.

Corollary 1. *Suppose that the system (1) has a CLF $V(x)$ in the domain $\mathbf{X}^V \subseteq \mathbf{X}$ and Assumptions A1∼4 are satisfied. Then, by choosing $\bar{x}^+$ according to (5), the CLF Based MPC Law remains feasible and stabilizes the system, provided that the initial $\mathscr{P}(x_0,\bar{x}_0,N)$ with x_0, $\bar{x}_0 \in \mathbf{X}^V$ is feasible and $\mathscr{X}(0,N_1)$ contains the origin as an interior point.*

The choice of $\bar{x}^+$ in (5) is a discrete-time variant of the control law of [9] in continuous time. For a given continuous system $\dot{x} = f_c(x,u)$ with a given Lyapunov function $V_c(x)$, the closed-loop control law $u \in \mathbf{U}_c$ was found in a way that achieves the minimum gradient of V_c. Also note that we use Artstein's idea to generate the terminal target state trajectories, not a closed-loop control solution. That is, the terminal target state trajectory given by (5) may be computed offline ahead of time since $\bar{x}^+$ in (5) is not a function of the closed-loop state x and so the terminal target state sequence is not determined by feedback. This motivates the following closed-loop approach to choosing $\bar{x}^+$.

4.2 Closed-Loop Approach

Definition 3. *For the state $x^+ = x^*(1,x)$ from $\mathscr{P}(x,\bar{x},N)$, $\mathbb{X}_N(x^*(1,x))$ is the set of states that are feasibly reachable from $x^*(1,x)$ in N steps.*

Remark 3. The set, $\mathbb{X}_N(x^*(1,x))$, relies on information available from the solution of $\mathscr{P}(x,\bar{x},N)$ and so is denoted as dependent on $x^*(1,x)$.

The second approach to choose $\bar{x}^+$ is

$$\bar{x}^+ = \arg\min_{\check{x}} \{V(\check{x}) \big| \check{x} \in \mathbb{X}_N(x^*(1,x)) \cap \mathbf{X}^V\}. \tag{6}$$

Corollary 2. *Suppose that the system (1) has a CLF $V(x)$ in the domain $\mathbf{X}^V \subseteq \mathbf{X}$ and Assumptions A1∼4 are satisfied. Then, by choosing $\bar{x}^+$ according to (6), the CLF Based MPC Law remains feasible and stabilizes the system, provided that the initial $\mathscr{P}(x_0,\bar{x}_0,N)$ with $x_0,\ \bar{x}_0 \in \mathbf{X}^V$ is feasible and $\mathscr{X}(0,N_1)$ contains the origin as an interior point.*

The terminal target state update (6) may cause a significant computational burden for systems with high dimension and short sampling time because the $N-$step reachable set, $\mathbb{X}_N(x^*(1,x))$, has to be computed at each time, while (5) requires only a one-step reachable set. This difficulty can be resolved if one formulates the constraint set $\mathbb{X}_N(x^*(1,x)) \cap \mathbf{X}^V$ into a function $g(\cdot)$ such that

$$\mathbb{X}_N(x^*(1,x)) \cap \mathbf{X}^V \triangleq \{z | \exists u_i \in \mathbf{U},\ i = 1,2,\ldots,N,\ g(z,u_1,\ldots,u_N,x^*(1,x)) \le 0\}.$$

Then computing $\mathbb{X}_N(x^*(1,x))$ at each time can be avoided by calculating

$$\begin{aligned}(\bar{x}^+,\bar{u}_1^+,\ldots,\bar{u}_N^+) = \arg \min_{z,u_1,\ldots,u_N} \{V(z) \big| g(z,u_1,\ldots,u_N,x^*(1,x)) \le 0,\\ u_i \in \mathbf{U},\ i = 1,2,\ldots,N\},\end{aligned} \tag{7}$$

where the controls $u_1,\ldots,u_N$ function as slack variables and are discarded. The approaches used in (6) and (7) are tractable at least for linear systems with polytopic constraints using the tools of [5] and [3]. If (6) (or equivalently (7)) is feasibly realized, compared to (5), it can be beneficial in the following sense.

Corollary 3. *Suppose that the system has a CLF in $\mathbf{X}^V \subseteq \mathbf{X}$ and Assumptions A1∼4 are satisfied. Provided the initial $\mathscr{P}(x_0,\bar{x}_0,N)$ with $x_0,\bar{x}_0 \in \mathbf{X}^V$ is feasible and $\mathscr{X}(0,N_1)$ contains the origin as an interior point, the CLF Based MPC Law with closed-loop selection rule (6) achieves at least the same or faster termination of the first stage than than that with open-loop selection rule (5).*

Proof. By the definition of $\mathbb{X}_N(x^*(1,x))$, we have $f(\bar{x},\mathbf{U}) \subseteq \mathbb{X}_N(x^*(1,x))$. Therefore (6) finds $\bar{x}^+$ that achieves the greatest reduction in $V(\bar{x}^+)$ over at least the same or larger set than $f(\bar{x},\mathbf{U}) \cap \mathbf{X}^V$ in (5). This completes the proof. □

The CLF Based MPC Law using (6) deserves comparison with the control laws proposed in [11]. In [11], a CLF is used to develop a state constraint. That is, for the given state $x^+ = x^*(1,x)$, they constrain the open-loop control to achieve

$$V(x^+(i+1,x^+)) \leq V(x^+(i,x^+)) - \beta(||x^+(i,x^+)||), \tag{8}$$

for the complete horizon $i = 0,1,\ldots,N-1$ or just for the next step $i = 0$. In other words, their open-loop control problem only requires the predicted states $x^+(i+1,x^+)$ to satisfy (8). In our case, for the given $x^*(1,x) = x^+$, we choose the terminal target state $\bar{x}^+$ to cause the greatest reduction in $V(\cdot)$ and find an open-loop control solution for $\mathscr{P}(x^+,\bar{x}^+,N)$. Thus, our method may lead to faster convergence of the state to the neighborhood of the origin than the MPC law of [11]. The above aspect to use a CLF only for choosing $\bar{x}^+$ also may improve control performance over that associated with the CLF alone being used for feedback control: this will be discussed further in the next section with an example.

5 Example

We consider a single scalar integrator plant

$$x^+ = x + u,$$

since its behavior is quite simple and hence makes it easy to appreciate the main idea and the performance enhancement of our scheme. We assume 0.2 second sampling time. The objective of control is to steer the initial state x_0 to the origin using the CLF Based MPC Law. We use the terminal state update schemes (5) and (6). [Denote them by *State Independent Update* and *State Dependent Update* respectively.] We use the open-loop constrained control problem (2) with the cost function $h(x(i,x),u(i,x)) = x(i,x)^2 + 100u(i,x)^2$. We assume the input constraint $\mathbf{U} = [-5,5]$ and no state constraints. The control horizons $\bar{N} = N = 3$ and $\bar{N} = N + N_1 = 6$ ($N_1 = 3$) are chosen for the first and second stages respectively. The set $\mathscr{X}(0,3)$ is $[-15,15]$. The initial state is given by $x_0 = 50$ and the initial target terminal target state is chosen to be $\bar{x} = 42$. The system has a nominal (low gain but) stabilizing feedback control $u = -0.1x$, which does not violate the control constraint. The associated CLF is $V(x) = x^2$. As shown in Figure 1, the state trajectory forced by the nominal feedback law converges to the origin quite slowly. Even if one wants to use the classical MPC with the zero terminal state equality constraint, a corresponding open-loop finite horizon control problem at each time step requires at least the horizon size $N = 10$ to achieve the zero terminal state equality constraint for the given initial state. When the *State Independent Update* (5) is used for selecting $\bar{x}^+$ from the initial $\bar{x}_0$, $u = -5$ minimizes $V(\bar{x}+u)$ and results in the terminal target state trajectory $\{42,37,32,\ldots,17\}$ until $\bar{x}$ enters $\mathscr{X}(0,3)$. If the *State Dependent Update* (6) is used, its corresponding transient closed-loop state approaches the (radius 15) neighborhood of the origin faster than that with the *State Independent Update* as shown in

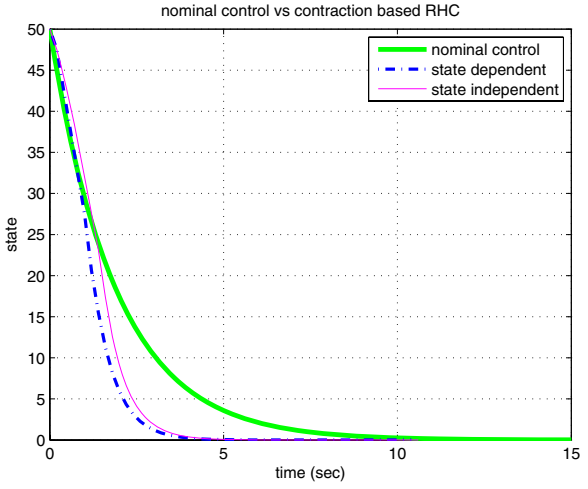


Fig. 1 State trajectories for the nominal control and the CLF Based MPC with two terminal target state selection approaches: *State Independent Update* (5) [thin line] and *State Dependent Update* (6) [dashed-dot line]

Corollary 3. If there were no penalty on the control [i.e. $h(x(i,x),u(i,x)) = x(i,x)^2$], for both cases of using *State Independent Update* and *State Dependent Update*, the corresponding control solution would have allowed the full control capacity to steer the state, which results in a straight-line-like trajectory and faster convergence to the origin than the nominal control. These results suggest that the CLF Based MPC Law allows greater capacity to shape the transient closed-loop state trajectory than when the control design is solely based on the known CLF via Artstein's formula.

6 Conclusion and Future Work

The existence of a CLF permits the construction of MPC for the case where the initial states are not feasibly steered to the origin in reasonable time by classical MPC. Our viewpoint provides value when an existing stabilizing but low performance controller is available, such as the low-gain linear feedback on the integrator plant. Then this controller's CLF may be used as the basis for MPC to augment the closed-loop system behavior. Further investigation is necessary for the system with disturbance and the computability of the proposed algorithms.

Acknowledgements. This work supported by AFOSR under Grant FA9550-05-1-0401. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of AFOSR.

References

1. Michalska, H., Mayne, D.Q.: Robust Receding Horizon Control of Constrained Nonlinear Systems. IEEE Transactions on Automatic Control 38, 1623–1633 (1993)
2. Primbs, J.A., Nevistić, V., Doyle, J.C.: Nonlinear Optimal Control: A Control Lyapunov Function and Receding Horizon Perspective. Asian Journal of Control 1, 14–24 (1999)
3. Kang, K., Bitmead, R.R.: Online Reference Computation for Feasible Model Predictive Control. In: Proceedings of the 46th IEEE Conference on Decision and Control (2006)
4. Chisci, L., Lombardi, A., Mosca, E.: Dual Receding Horizon Control of Constrained Discrete Time Systems. European Journal of Control 2, 278–285 (1996)
5. Kvasnica, M., Grieder, P., Baotić, M.: Multi-Parametric Toolbox, MPT (2004), http://control.ee.ethz.ch/~mpt/
6. Keerthi, S.S., Gilbert, E.G.: An Existence Theorem for Discrete-Time Infinite-Horizon Optimal Control Problems. IEEE Transactions on Automatic Control 30, 907–909 (1985)
7. Keerthi, S.S., Gilbert, E.G.: Optimal Infinite-Horizon Feedback Laws for a General Class of Constrained Discrete-Time Systems: Stability and Moving-Horizon Approximations. Journal of Optimization Theory and Applications 57, 265–293 (1988)
8. Raković, S.V., Kerrigan, E.C., Mayne, D.Q., Lygeros, J.: Reachability Analysis of Discrete-Time Systems with Disturbances. IEEE Transactions on Automatic Control 51, 546–561 (2006)
9. Artstein, Z.: Stabilization with Relaxed Controls. Nonlinear Analysis, Theory, Methods, and Applications 7, 1163–1173 (1983)
10. Kwon, W., Pearson, A.E.: On Feedback Stabilization of Time-Varying Discrete Time Linear Systems. IEEE Transactions on Automatic Control 23, 479–481 (1978)
11. Kwon, W., Han, S.: Receding Horizon Control. Springer, Heidelberg (2005)

Further Results on "Robust MPC Using Linear Matrix Inequalities"

M. Lazar*, W.P.M.H. Heemels, D. Muñoz de la Peña, and T. Alamo

Abstract. This paper presents a novel method for designing the terminal cost and the auxiliary control law (ACL) for robust MPC of uncertain linear systems, such that ISS is a priori guaranteed for the closed-loop system. The method is based on the solution of a set of LMIs. An explicit relation is established between the proposed method and $\mathscr{H}_\infty$ control design. This relation shows that the LMI-based optimal solution of the $\mathscr{H}_\infty$ synthesis problem solves the terminal cost and ACL problem in inf-sup MPC, for a particular choice of the stage cost. This result, which was somehow missing in the MPC literature, is of general interest as it connects well known linear control problems to robust MPC design.

Keywords: robust model predictive control (MPC), linear matrix inequalities (LMIs), $\mathscr{H}_\infty$ control, input-to-state stability (ISS).

1 Introduction

Perhaps the most utilized method for designing stabilizing and robustly stabilizing model predictive controllers (MPC) is the terminal cost and constraint set approach [1]. This technique, which applies to both nominally stabilizing and inf-sup robust MPC schemes, relies on the off-line computation of a suitable terminal cost along with an auxiliary control law (ACL). For nominally stabilizing MPC with quadratic costs, the terminal cost can be calculated for linear dynamics by solving a discrete-time Riccati equation, with the optimal linear quadratic regulator (LQR) as the ACL [2]. In [3] it was shown that an alternative solution to the same problem,

M. Lazar and W.P.M.H. Heemels
Eindhoven Univ. of Technology, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands
e-mail: m.lazar@tue.nl

D. Muñoz de la Peña and T. Alamo
Dept. de Ingeniería de Sistemas y Automática, Univ. of Seville, 41092 Seville, Spain

* Corresponding author.

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 89–98.
springerlink.com

which also works for parametric uncertainties, can be obtained by solving a set of LMIs. The design of inf-sup MPC schemes that are robust to additive disturbances was treated in [4], where it was proven that the terminal cost can be obtained as a solution of a discrete-time $\mathscr{H}_\infty$ Riccati equation, for an ACL that solves the corresponding $\mathscr{H}_\infty$ control problem.

In this article we present an LMI-based solution for obtaining a terminal cost and an ACL, such that inf-sup MPC schemes [7, 8] achieve input-to-state stability (ISS) [5] for linear systems affected by both parametric and additive disturbances. The proposed LMIs generalize the conditions in [3] to allow for additive uncertainties as well. Moreover, we establish an explicit relation between the developed solution and the LMI-based[1] optimal solution of the discrete-time $\mathscr{H}_\infty$ synthesis problem corresponding to a specific performance output, related to the MPC cost. This result, which was somehow missing in the MPC literature, adds to the results of [4] and to the well known connection between design of nominally stabilizing MPC schemes and the optimal solution of the LQR problem. Such results are of general interest as they connect well known linear control problems to MPC design.

2 Preliminary Definitions and Results

2.1 Basic Notions and Definitions

Let $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{Z}$ and $\mathbb{Z}_+$ denote the field of real numbers, the set of non-negative reals, the set of integer numbers and the set of non-negative integers, respectively. We use the notation $\mathbb{Z}_{\geq c_1}$ and $\mathbb{Z}_{(c_1,c_2]}$ to denote the sets $\{k \in \mathbb{Z}_+ \mid k \geq c_1\}$ and $\{k \in \mathbb{Z}_+ \mid c_1 < k \leq c_2\}$, respectively, for some $c_1, c_2 \in \mathbb{Z}_+$. For $i \in \mathbb{Z}_+$, let $i = \overline{1,N}$ denote $i = 1,\ldots,N$. For a set $\mathscr{S} \subseteq \mathbb{R}^n$, we denote by $\text{int}(\mathscr{S})$ the interior of $\mathscr{S}$. A polyhedron (or a polyhedral set) in $\mathbb{R}^n$ is a set obtained as the intersection of a finite number of open and/or closed half-spaces. The Hölder p-norm of a vector $x \in \mathbb{R}^n$ is defined as $\|x\|_p := (|[x]_1|^p + \ldots + |[x]_n|^p)^{\frac{1}{p}}$ for $p \in \mathbb{Z}_{[1,\infty)}$ and $\|x\|_\infty := \max_{i=1,\ldots,n} |[x]_i|$, where $[x]_i$, $i = 1,\ldots,n$, is the i-th component of x and $|\cdot|$ is the absolute value. For a positive definite and symmetric matrix M, denoted by $M \succ 0$, $M^{\frac{1}{2}}$ denotes its Cholesky factor, which satisfies $(M^{\frac{1}{2}})^\top M^{\frac{1}{2}} = M^{\frac{1}{2}}(M^{\frac{1}{2}})^\top = M$ and, $\lambda_{\min}(M)$ and $\lambda_{\max}(M)$ denote the smallest and the largest eigenvalue of M, respectively. We will use 0 and I to denote a matrix with all elements zero and the identity matrix, respectively, of appropriate dimensions. Let $\mathbf{z} := \{z(l)\}_{l \in \mathbb{Z}_+}$ with $z(l) \in \mathbb{R}^o$ for all $l \in \mathbb{Z}_+$ denote an arbitrary sequence. Define $\|\mathbf{z}\| := \sup\{\|z(l)\| \mid l \in \mathbb{Z}_+\}$, where $\|\cdot\|$ denotes an arbitrary p-norm, and $\mathbf{z}_{[k]} := \{z(l)\}_{l \in \mathbb{Z}_{[0,k]}}$. A function $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ belongs to class $\mathscr{K}$ if it is continuous, strictly increasing and $\varphi(0) = 0$. A function $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ belongs to class $\mathscr{K}_\infty$ if $\varphi \in \mathscr{K}$ and $\lim_{s\to\infty} \varphi(s) = \infty$. A function $\beta : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ belongs to class $\mathscr{KL}$ if for each fixed $k \in \mathbb{R}_+$, $\beta(\cdot,k) \in \mathscr{K}$ and for each fixed $s \in \mathbb{R}_+$, $\beta(s,\cdot)$ is decreasing and $\lim_{k\to\infty} \beta(s,k) = 0$.

[1] A similar connection is established in [4], with the difference that the Riccati-based solution to the optimal $\mathscr{H}_\infty$ synthesis problem is exploited, rather than the LMI-based solution; also, parametric uncertainties are not considered.

2.2 Input-to-State Stability

Consider the discrete-time nonlinear system

$$x(k+1) = \Phi(x(k), w(k), v(k)), \quad k \in \mathbb{Z}_+, \tag{1}$$

where $x(k) \in \mathbb{R}^n$ is the state and $w(k) \in \mathbb{R}^{d_w}$, $v(k) \in \mathbb{R}^{d_v}$ are unknown disturbance inputs at the discrete-time instant k. The mapping $\Phi : \mathbb{R}^n \times \mathbb{R}^{d_w} \times \mathbb{R}^{d_v} \to \mathbb{R}^n$ is an arbitrary nonlinear function. We assume that $\Phi(0,w,0) = 0$ for all w. Let $\mathbb{W}$ and $\mathbb{V}$ be subsets of $\mathbb{R}^{d_w}$ and $\mathbb{R}^{d_v}$, respectively.

Definition 1. *We call a set $\mathscr{P} \subseteq \mathbb{R}^n$* robustly positively invariant (RPI) *for system (1) with respect to $(\mathbb{W},\mathbb{V})$ if for all $x \in \mathscr{P}$ it holds that $\Phi(x,w,v) \in \mathscr{P}$ for all $(w,v) \in \mathbb{W} \times \mathbb{V}$.*

Definition 2. *Let $\mathbb{X}$ with $0 \in \text{int}(\mathbb{X})$ be a subset of $\mathbb{R}^n$. We call system (1)* ISS($\mathbb{X}$, $\mathbb{W}$, $\mathbb{V}$) *if there exist a $\mathscr{KL}$-function $\beta(\cdot,\cdot)$ and a $\mathscr{K}$-function $\gamma(\cdot)$ such that, for each $x(0) \in \mathbb{X}$, all $\mathbf{w} = \{w(l)\}_{l\in\mathbb{Z}_+}$ with $w(l) \in \mathbb{W}$, $\forall l \in \mathbb{Z}_+$ and all $\mathbf{v} = \{v(l)\}_{l\in\mathbb{Z}_+}$ with $v(l) \in \mathbb{V}$, $\forall l \in \mathbb{Z}_+$ it holds that the corresponding state trajectory of (1) satisfies $\|x(k)\| \le \beta(\|x(0)\|,k) + \gamma(\|\mathbf{v}_{[k-1]}\|)$, $\forall k \in \mathbb{Z}_{\ge 1}$. We call the function $\gamma(\cdot)$ an ISS gain of system (1).*

2.3 Input-to-State Stability Conditions for Inf-sup Robust MPC

Consider the discrete-time constrained nonlinear system

$$x(k+1) = \phi(x(k), u(k), w(k), v(k)), \; k \in \mathbb{Z}_+, \tag{2}$$

where $x(k) \in \mathbb{X} \subseteq \mathbb{R}^n$ is the state, $u(k) \in \mathbb{U} \subseteq \mathbb{R}^m$ is the control action and $w(k) \in \mathbb{W} \subset \mathbb{R}^{d_w}$, $v(k) \in \mathbb{V} \subset \mathbb{R}^{d_v}$ are unknown disturbance inputs at the discrete-time instant k. $\phi : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{d_w} \times \mathbb{R}^{d_v} \to \mathbb{R}^n$ is an arbitrary nonlinear function with $\phi(0,0,w,0) = 0$ for all $w \in \mathbb{W}$. We assume that $0 \in \text{int}(\mathbb{X})$, $0 \in \text{int}(\mathbb{U})$ and $\mathbb{W}$, $\mathbb{V}$ are bounded. Next, let $F : \mathbb{R}^n \to \mathbb{R}_+$ and $L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_+$ with $F(0) = L(0,0) = 0$ be arbitrary nonlinear functions. For $N \in \mathbb{Z}_{\ge 1}$ let $\bar{\mathbf{u}}_{[N-1]}(k) := (\bar{u}(k), \bar{u}(k+1), \ldots, \bar{u}(k+N-1)) \in \mathbb{U}^N = \mathbb{U} \times \ldots \times \mathbb{U}$ denote a sequence of future inputs and, similarly, let $\bar{\mathbf{w}}_{[N-1]}(k) \in \mathbb{W}^N$, $\bar{\mathbf{v}}_{[N-1]}(k) \in \mathbb{V}^N$ denote some sequences of future disturbances. Consider the MPC cost

$$J(x(k), \bar{\mathbf{u}}_{[N-1]}(k), \bar{\mathbf{w}}_{[N-1]}(k), \bar{\mathbf{v}}_{[N-1]}(k)) := F(\bar{x}(k+N)) + \sum_{i=0}^{N-1} L(\bar{x}(k+i), \bar{u}(k+i)),$$

where $\bar{x}(k+i+1) := \phi(\bar{x}(k+i), \bar{u}(k+i), \bar{w}(k+i), \bar{v}(k+i))$ for $i = \overline{0, N-1}$ and $\bar{x}(k) := x(k)$. Let $\mathbb{X}_T \subseteq \mathbb{X}$ with $0 \in \text{int}(\mathbb{X}_T)$ denote a target set and define the following set of feasible input sequences:

$$\mathscr{U}_N(x(k)) := \{\mathbf{u}_{[N-1]}(k) \in \mathbb{U}^N \mid \bar{x}(k+i) \in \mathbb{X},\ i = \overline{1,N-1},\ \bar{x}(k+N) \in \mathbb{X}_T,$$
$$\bar{x}(k) := x(k), \forall \bar{\mathbf{w}}_{[N-1]}(k) \in \mathbb{W}^N, \forall \bar{\mathbf{v}}_{[N-1]}(k) \in \mathbb{V}^N\}.$$

Problem 1. Let $\mathbb{X}_T \subseteq \mathbb{X}$ and $N \in \mathbb{Z}_{\geq 1}$ be given. At time $k \in \mathbb{Z}_+$ let $x(k) \in \mathbb{X}$ be given and infimize

$$\sup_{\bar{\mathbf{w}}_{[N-1]}(k) \in \mathbb{W}^N, \bar{\mathbf{v}}_{[N-1]}(k) \in \mathbb{V}^N} J(x(k), \bar{\mathbf{u}}_{[N-1]}(k), \bar{\mathbf{w}}_{[N-1]}(k), \bar{\mathbf{v}}_{[N-1]}(k))$$

over all input sequences $\bar{\mathbf{u}}_{[N-1]}(k) \in \mathscr{U}_N(x(k))$. □

Assuming the infimum in Problem 1 exists and can be attained, the MPC control law is obtained as $u_{\mathrm{MPC}}(x(k)) := \bar{u}^*(k)$, where $*$ denotes the optimum[2].

Next, we summarize recently developed a priori sufficient conditions for guaranteeing robust stability of system (2) in closed-loop with $u(k) = u_{\mathrm{MPC}}(x(k))$, $k \in \mathbb{Z}_+$. Let $h : \mathbb{R}^n \to \mathbb{R}^m$ denote an auxiliary control law (ACL) with $h(0) = 0$ and let $\mathbb{X}_{\mathbb{U}} := \{x \in \mathbb{X} \mid h(x) \in \mathbb{U}\}$.

Assumption 1. *There exist functions $\alpha_1, \alpha_2, \alpha_3 \in \mathscr{K}_\infty$ and $\sigma \in \mathscr{K}$ such that:*
(i) $\mathbb{X}_T \subseteq \mathbb{X}_{\mathbb{U}}$;
(ii) $\mathbb{X}_T$ is a RPI set for system (2) in closed-loop with $u(k) = h(x(k))$, $k \in \mathbb{Z}_+$;
(iii) $L(x,u) \geq \alpha_1(\|x\|)$ for all $x \in \mathbb{X}$ and all $u \in \mathbb{U}$;
(iv) $\alpha_2(\|x\|) \leq F(x) \leq \alpha_3(\|x\|)$ for all $x \in \mathbb{X}_T$;
(v) $F(\phi(x,h(x),w,v)) - F(x) \leq -L(x,h(x)) + \sigma(\|v\|)$, $\forall x \in \mathbb{X}_T$, $\forall w \in \mathbb{W}$, $\forall v \in \mathbb{V}$.

In [7, 8] it was shown that Assumption 1 is sufficient for guaranteeing ISS of the MPC closed-loop system corresponding to Problem 1. Notice that although in Problem 1 we have presented the "open-loop" formulation of inf-sup MPC for simplicity of exposition, Assumption 1 is also sufficient for guaranteeing ISS for "feedback" inf-sup variants of Problem 1, see [7, 8] for the details.

Remark 1. The sufficient ISS conditions of Assumption 1 are an extension for robust MPC of the well known *terminal cost and constraint set* stabilization conditions for nominal MPC, see A1-A4 in [1]. While the stabilization conditions for MPC [1] require that the terminal cost is a local *Lyapunov function* for the system in closed-loop with an ACL, Assumption 1 requires in a similar manner that the terminal cost is a local *ISS Lyapunov function* [5] for the system in closed-loop with an ACL. □

3 Problem Formulation

For a given stage cost $L(\cdot,\cdot)$, to employ Assumption 1 for setting-up robust MPC schemes with an a priori ISS guarantee (or to compute state feedback controllers that achieve local ISS), one needs systematic methods for computing a terminal cost $F(\cdot)$, a terminal set $\mathbb{X}_T$ and an ACL $h(\cdot)$ that satisfy Assumption 1.

[2] If the infimum does not exist, one has to resort to ISS results for sub-optimal solutions, see, e.g., [6].

Once $F(\cdot)$ and $h(\cdot)$ are known, several methods are available for calculating the maximal RPI set contained in $\mathbb{X}_{\mathbb{U}}$ for certain relevant subclasses of system (2), in closed-loop with $u(k) = h(x(k))$, $k \in \mathbb{Z}_+$, see, for example, [9, 10] and the references therein. As a consequence, therefore, we focus on solving the following problem.

Problem 2. Calculate $F(\cdot)$ and $h(\cdot)$ such that Assumption 1-(v) holds. □

This problem comes down to computing an input-to-state stabilizing state-feedback given by $h(\cdot)$ along with an ISS Lyapunov function (i.e. $F(\cdot)$) for system (2) in closed-loop with the ACL. This is a non-trivial problem, which depends on the type of MPC cost, system class and on the type of candidate ISS Lyapunov function $F(\cdot)$. Furthermore, it would be desirable that the MPC cost function is continuous and convex.

3.1 Existing Solutions

Several solutions have been presented for the considered problem for particular subclasses of system (2). Most methods consider quadratic cost functions, $F(x) := x^\top Px$, $P \succ 0$, $L(x,u) = x^\top Qx + u^\top Ru$, $Q,R \succ 0$, and linear state feedback ACLs given by $h(x) := Kx$.

(i) *The nominal linear case*: $\phi(x,u,0,0) := Ax+Bu$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$. In [2] it was proven that the solutions of the unconstrained infinite horizon linear quadratic regulation problem with weights Q,R satisfy Assumption 1-(v), i.e.

$$K = -(R+B^\top PB)^{-1}B^\top PA$$

and

$$P = (A+BK)^\top P(A+BK) + K^\top RK + Q. \tag{3}$$

Numerically, this method amounts to solving the discrete-time Riccati equation (3).

(ii) *The linear case with parametric disturbances*: $\phi(x,u,w,0) := A(w)x+B(w)u$, $A(w) \in \mathbb{R}^{n\times n}$, $B(w) \in \mathbb{R}^{n\times m}$ are affine functions of $w \in \mathbb{W}$ with $\mathbb{W}$ a compact polyhedron. In [3] it was proven that $P = Z^{-1}$ and $K = YZ^{-1}$ satisfy Assumption 1-(v), where $Z \in \mathbb{R}^{n\times n}$ and $Y \in \mathbb{R}^{m\times n}$ are solutions of the linear matrix inequality

$$\begin{pmatrix} Z & (A(w_i)Z+B(w_i)Y)^\top & (R^{\frac{1}{2}}Y)^\top & (Q^{\frac{1}{2}}Z)^\top \\ (A(w_i)Z+B(w_i)Y) & Z & 0 & 0 \\ R^{\frac{1}{2}}Y & 0 & I & 0 \\ Q^{\frac{1}{2}}Z & 0 & 0 & I \end{pmatrix} \succ 0, \forall i = \overline{1,E},$$

with $w_1,\ldots,w_E$ the vertices of the polytope $\mathbb{W}$. Numerically, this method amounts to solving a semidefinite programming problem. This solution trivially applies also to the case (i) and, moreover, it was extended to piecewise affine discrete-time hybrid systems in [11].

(iii) *The nonlinear case with additive disturbances*: $\phi(x,u,0,v) = f(x)+g_1(x)u+g_2(x)v$ with suitably defined functions $f(\cdot)$, $g_1(\cdot)$ and $g_2(\cdot)$. A nonlinear ACL given

by $h(x)$ was constructed in [4] using linearization of the system, so that Assumption 1-(v) holds for all states in a sufficiently small sublevel set of $V(x) = x^\top Px$, $P \succ 0$. Numerically this method amounts to solving a discrete-time $\mathscr{H}_\infty$ Riccati equation.

For the linear case with additive disturbances (i.e. $f(x) = A$, $g_1(x) = B$ and $g_1(x) = B_1$), it is worth to point out that an LMI-based design method to obtain the terminal cost, *for a given ACL*, was presented in [12].

4 Main Results

In this section we derive a novel LMI-based solution to the problem of finding a suitable terminal cost and ACL that applies to linear systems affected by both parametric and additive disturbances, i.e.

$$x(k+1) = \phi(x(k),u(k),w(k),v(k)) := A(w(k))x(k) + B(w(k))u(k) + B_1(w(k))v(k), \quad (4)$$

where $A(w) \in \mathbb{R}^{n\times n}$, $B(w) \in \mathbb{R}^{n\times m}$, $B_1(w) \in \mathbb{R}^{n\times d_v}$ are affine functions of w. We will also consider quadratic cost functions, $F(x) := x^\top Px$, $P \succ 0$, $L(x,u) = x^\top Qx + u^\top Ru$, $Q, R \succ 0$, and linear state feedback ACLs given by $h(x) := Kx$.

4.1 LMI-Based-Solution

Consider the linear matrix inequalities,

$$\begin{pmatrix} Z & 0 & (A(w_i)Z + B(w_i)Y)^\top & (R^{\frac{1}{2}}Y)^\top & (Q^{\frac{1}{2}}Z)^\top \\ 0 & \tau I & B_1(w_i)^T & 0 & 0 \\ (A(w_i)Z + B(w_i)Y) & B_1(w_i) & Z & 0 & 0 \\ R^{\frac{1}{2}}Y & 0 & 0 & I & 0 \\ Q^{\frac{1}{2}}Z & 0 & 0 & 0 & I \end{pmatrix} \succ 0,$$
$$\forall i = \overline{1,E}, \quad (5)$$

where $w_1, \ldots, w_E$ are the vertices of the polytope $\mathbb{W}$, $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ are known positive definite and symmetric matrices, and $Z \in \mathbb{R}^{n\times n}$, $Y \in \mathbb{R}^{m\times n}$ and $\tau \in \mathbb{R}_{>0}$ are the unknowns.

Theorem 1. *Suppose that the LMIs (5) are feasible and let Z, Y and τ be a solution with $Z \succ 0$, $\tau \in \mathbb{R}_{>0}$. Then, the terminal cost $F(x) = x^\top Px$, the stage cost $L(x,u) = x^\top Qx + u^\top Ru$ and the ACL $h(x) = Kx$ with $P := Z^{-1}$ and $K := YZ^{-1}$ satisfy Assumption 1-(v) with $\sigma(\|v\|) := \tau\|v\|_2^2 = \tau v^\top v$.*

Proof: For brevity let $\Delta(w_i)$ denote the matrix in the left-hand side of (5). Using $\mathbb{W} = \mathrm{Co}\{w_1, \ldots, w_E\}$ (where $\mathrm{Co}\{\cdot\}$ denotes the convex hull) and the fact that $A(w), B(w)$ and $B_1(w)$ are affine functions of w, it is trivial to observe that if (5) holds for all vertices $w_1, \ldots, w_E$ of $\mathbb{W}$, then $\Delta(w) \succ 0$ holds for all $w \in \mathbb{W}$.

Applying the Schur complement to $\Delta(w) \succ 0$ (pivoting after $\mathrm{diag}(Z, I, I)$) and letting $M(w) := A(w)Z + B(w)Y$ yields the equivalent matrix inequalities:

$$\begin{pmatrix} Z - M(w)^\top Z^{-1} M(w) - Z^\top Q Z - Y^\top R Y & -M(w)^\top Z^{-1} B_1(w) \\ -B_1(w)^\top Z^{-1} M(w) & \tau I - B_1(w)^\top Z^{-1} B_1(w) \end{pmatrix} \succ 0$$

and $Z \succ 0$. Letting $A_{\mathrm{cl}}(w) := A(w) + B(w)K$, substituting $Z = P^{-1}$ and $Y = KP^{-1}$, and performing a congruence transformation on the above matrix inequality with $\mathrm{diag}(P, I)$ yields the equivalent matrix inequalities:

$$\begin{pmatrix} P - A_{\mathrm{cl}}(w)^\top P A_{\mathrm{cl}}(w) - Q - K^\top R K & -A_{\mathrm{cl}}(w)^\top P B_1(w) \\ -B_1(w)^\top P A_{\mathrm{cl}}(w)) & \tau I - B_1(w)^\top P B_1(w) \end{pmatrix} \succ 0$$

and $P \succ 0$. Pre multiplying with $\left(\begin{smallmatrix} x \\ v \end{smallmatrix}\right)^\top$ and post multiplying with $\left(\begin{smallmatrix} x \\ v \end{smallmatrix}\right)$ the above matrix inequality yields the equivalent inequality:

$$(A_{\mathrm{cl}}(w)x + B_1(w)v)^\top P (A_{\mathrm{cl}}(w)x + B_1(w)v) - x^\top P x \le -x^\top (Q + K^\top R K)x + \tau v^\top v,$$

for all $x \in \mathbb{R}^n$ and all $v \in \mathbb{R}^{d_v}$. Hence, Assumption 1-(v) holds with $\sigma(\|v\|) = \tau \|v\|_2^2$. □

Remark 2. In [8] the authors established an explicit relation between the gain $\tau \in \mathbb{R}_{>0}$ of the function $\sigma(\cdot)$ and the ISS gain of the corresponding closed-loop MPC system. Thus, since τ enters (5) linearly, one can minimize over τ subject to the LMIs (5), leading to a smaller ISS gain from v to x. □

4.2 Relation to LMI-Based $\mathscr{H}_\infty$ Control Design

In this section we formalize the relation between the considered robust MPC design problem and $\mathscr{H}_\infty$ design for linear systems. But first, we briefly recall the $\mathscr{H}_\infty$ design procedure for the discrete-time linear system (4). For simplicity, we remove the parametric disturbance w and consider only additive disturbances $v \in \mathbb{V}$. However, the results derived below that relate to the optimal $\mathscr{H}_\infty$ gain also hold if parametric disturbances are considered, in the sense of an optimal $\mathscr{H}_\infty$ gain for linear parameter varying systems.

Consider the system corresponding to (4) without parametric uncertainties, i.e.

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + B_1 v(k), \\ z(k) &= Cx(k) + Du(k) + D_1 v(k), \end{aligned} \tag{6}$$

where we added the performance output $z \in \mathbb{R}^{d_z}$. Using the results of [13], [14] it can be demonstrated that system (6) in closed-loop with $u(k) = h(x(k)) = Kx(k)$, $k \in \mathbb{Z}_+$, has an $\mathscr{H}_\infty$ gain less than $\sqrt{\gamma}$ if and only if there exists a symmetric matrix P such that:

$$\begin{pmatrix} P & 0 & (A+BK)^\top P & (C+DK)^\top \\ 0 & \gamma I & B_1^\top P & D_1^\top \\ P(A+BK) & PB_1 & P & 0 \\ C+DK & D_1 & 0 & I \end{pmatrix} \succ 0. \tag{7}$$

Letting $Z = P^{-1}$, $Y = KP^{-1}$ and performing a congruence transformation using $\mathrm{diag}(Z, I, Z, I)$ one obtains the equivalent LMI:

$$\begin{pmatrix} Z & 0 & (AZ+BY)^\top & (CZ+DY)^\top \\ 0 & \gamma I & B_1^\top & D_1^\top \\ AZ+BY & B_1 & Z & 0 \\ CZ+DY & D_1 & 0 & I \end{pmatrix} \succ 0. \tag{8}$$

Indeed, from the above inequalities, where $V(x) := x^\top Px$, one obtains the dissipation inequality:

$$V(x(k+1)) - V(x(k)) \leq -\|z(k)\|_2^2 + \gamma\|v\|_2^2. \tag{9}$$

Hence, we can infer that $\sum_{i=0}^{\infty}\|z(i)\|_2^2 \leq \gamma\sum_{i=0}^{\infty}\|v(i)\|_2^2$ and conclude that the $\mathscr{H}_\infty$ norm of the system is not greater than $\sqrt{\gamma}$. Minimizing γ subject to the above LMI yields the optimal $\mathscr{H}_\infty$ gain as the square root of the optimum.

Remark 3. In [13], [14] an equivalent formulation of the matrix inequality (7) is used, i.e. with γI in the south east corner of (7)-(8) instead of I, which leads to the adapted dissipation inequality $V(x(k+1)) - V(x(k)) \leq -\gamma^{-1}\|z(k)\|_2^2 + \gamma\|v\|_2^2$. Then, by minimizing over γ subject to the LMIs (8), one obtains the optimal $\mathscr{H}_\infty$ gain directly as the optimal solution, without having to take the square root. However, regardless of which LMI set-up is employed, the resulting optimal $\mathscr{H}_\infty$ gain and corresponding controller (defined by the gain K) are the same, with a difference in the storage function $V(x) = x^\top Px$ with a factor γ. □

Theorem 2. *Suppose that the LMIs (5) without parametric uncertainties and (8) with* $C = \begin{pmatrix} Q^{\frac{1}{2}} \\ 0 \end{pmatrix}$, $D = \begin{pmatrix} 0 \\ R^{\frac{1}{2}} \end{pmatrix}$ *and* $D_1 = 0$ *are feasible for system (6). Then the following statements are equivalent:*

1. *Z, Y and* τ *are a solution of (5);*
2. *Z, Y and* γ *are a solution of (8) with* $C = \begin{pmatrix} Q^{\frac{1}{2}} \\ 0 \end{pmatrix}$, $D = \begin{pmatrix} 0 \\ R^{\frac{1}{2}} \end{pmatrix}$ *and* $D_1 = 0$*;*
3. *System (6) in closed-loop with* $u(k) = Kx(k)$ *and* $K = YZ^{-1}$ *satisfies the dissipation inequality (9) with storage function* $V(x) = x^\top Px$ *and* $P = Z^{-1}$*, and it has an* $\mathscr{H}_\infty$ *norm less than* $\sqrt{\gamma} = \sqrt{\tau}$*;*
4. *Assumption 1-(v) holds for* $F(x) = x^\top Px$, $L(x,u) = x^\top Qx + u^\top Ru$ *and* $h(x) = Kx$*, with* $P = Z^{-1}$, $K = YZ^{-1}$ *and* $\sigma(\|v\|) = \tau\|v\|_2^2 = \gamma\|v\|_2^2$.

The proof of Theorem 2 is trivially obtained by replacing C, D and D_1 in (8) and (9), respectively, and using Theorem 1 and the results of [13], [14].

Theorem 2 establishes that the LMI-based solution for solving Problem 2 proposed in this paper guarantees an $\mathscr{H}_\infty$ gain equal to the square root of the gain $\tau = \gamma$ of the $\sigma(\cdot)$ function for the system in closed-loop with the ACL. It also shows that the optimal $\mathscr{H}_\infty$ control law obtained by minimizing $\gamma = \tau$ subject to (8) (for a particular performance output related to the MPC cost) solves the terminal cost and ACL problem in inf-sup robust MPC. These results establish an intimate connection between $\mathscr{H}_\infty$ design and inf-sup MPC, in a similar way as LQR design is connected to nominally stabilizing MPC. This connection is instrumental in improving the

closed-loop ISS gain of inf-sup MPC closed-loop systems as follows: an optimal gain $\tau = \gamma$ of the $\sigma(\cdot)$ function results in a smaller gain of the function $\gamma(\cdot)$ of Definition 2 for the MPC closed-loop system, as demonstrated in [8].

5 Conclusions

In this article we proposed a novel LMI-based solution to the terminal cost and auxiliary control law problem in inf-sup robust MPC. The developed conditions apply to a more general class of systems than previously considered, i.e. linear systems affected by both parametric and additive disturbances. Since LMIs can be solved efficiently, the proposed method is computationally attractive. Furthermore, we have established an intimate connection between the proposed LMIs and the optimal $\mathscr{H}_\infty$ control law. This result, which was somehow missing in the MPC literature, adds to the well-known connection between design of nominally stabilizing MPC schemes and the optimal solution of the LQR problem. Such results are of general interest as they connect well known linear control problems to MPC design.

References

1. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: Stability and optimality. Automatica 36(6), 789–814 (2000)
2. Scokaert, P.O.M., Rawlings, J.B.: Constrained linear quadratic regulation. IEEE Transactions on Automatic Control 43(8), 1163–1169 (1998)
3. Kothare, M.V., Balakrishnan, V., Morari, M.: Robust constrained model predictive control using linear matrix inequalities. Automatica 32(10), 1361–1379 (1996)
4. Magni, L., De Nicolao, G., Scattolini, R., Allgöwer, F.: Robust MPC for nonlinear discrete-time systems. International Journal of Robust and Nonlinear Control 13, 229–246 (2003)
5. Jiang, Z.-P., Wang, Y.: Input-to-state stability for discrete-time nonlinear systems. Automatica 37, 857–869 (2001)
6. Lazar, M., Heemels, W.P.M.H.: Predictive control of hybrid systems: Input-to-state stability results for sub-optimal solutions. Automatica 45(1), 180–185 (2009)
7. Magni, L., Raimondo, D.M., Scattolini, R.: Regional input-to-state stability for nonlinear model predictive control. IEEE Transactions on Automatic Control 51(9), 1548–1553 (1998)
8. Lazar, M., Muñoz de la Peña, D., Heemels, W.P.M.H., Alamo, T.: On input-to-state stability of min-max MPC. Systems & Control Letters 57, 39–48 (2008)
9. Kolmanovsky, I., Gilbert, E.G.: Theory and computation of disturbance invariant sets for discrete-time linear systems. Mathematical Problems in Engineering 4, 317–367 (1998)
10. Alessio, A., Lazar, M., Heemels, W.P.M.H., Bemporad, A.: Squaring the circle: An algorithm for generating polyhedral invariant sets from ellipsoidal ones. Automatica 43(12), 2096–2103 (2007)
11. Lazar, M., Heemels, W.P.M.H., Weiland, S., Bemporad, A.: Stabilizing model predictive control of hybrid systems. IEEE Transactions on Automatic Control 51(11), 1813–1818 (2006)

12. Alamo, T., Muñoz de la Peña, D., Limon, D., Camacho, E.F.: Constrained minmax predictive control: Modifications of the objective function leading to polynomial complexity. IEEE Transactions on Automatic Control 50(5), 710–714 (2005)
13. Kaminer, I., Khargonekar, P.P., Rotea, M.A.: Mixed $\mathscr{H}_2/\mathscr{H}_\infty$ control for discrete time systems via convex optimization. Automatica 29, 57–70 (1993)
14. Chen, H., Scherer, C.W.: Moving horizon $\mathscr{H}_\infty$ with performance adaptation for constrained linear systems. Automatica 42, 1033–1040 (2006)

LMI-Based Model Predictive Control for Linear Discrete-Time Periodic Systems

Christoph Böhm, Tobias Raff, Marcus Reble, and Frank Allgöwer

Abstract. This paper presents a new model predictive control (MPC) scheme for linear constrained discrete-time periodic systems. In each period of the system, a new periodic state feedback control law is computed via a convex optimization problem that minimizes an upper bound on an infinite horizon cost function subject to state and input constraints. The performance of the proposed model predictive controller, that stabilizes the discrete-time periodic system if it is initially feasible, is illustrated via an example.

Keywords: Model predictive control, periodic system, linear matrix inequality.

1 Introduction

MPC based on linear discrete-time systems is widely used in industrial applications and much research has been done on its system theoretical properties, e.g. stability [16] and robustness [2]. However, only few MPC schemes have been developed for linear discrete-time periodic systems, see e.g. [7, 11, 13]. Periodic systems are of great importance for engineering applications. Examples of processes that can be modeled through a linear periodic system are sampled-data systems, satellites [15], rotors of helicopters [1], or chemical processes. Several methods to guarantee stability of linear periodic systems have been developed in the past [3, 5, 8, 9, 10]. A survey of the analysis and control of periodic systems is given in [4].

In this paper a new MPC scheme for linear discrete-time periodic systems based on results in [8] and [12] is proposed. In each period of the system (not at each time instant as usually in MPC), a new periodic state feedback control law is computed that minimizes an upper bound on an infinite horizon cost function subject to input and state constraints. This design problem is formulated as a convex optimization problem involving linear matrix inequalities (LMIs). It is shown that the

Christoph Böhm, Tobias Raff, Marcus Reble, and Frank Allgöwer
Institute for Systems Theory and Automatic Control,
University of Stuttgart, Germany
e-mail: {cboehm,raff,reble,allgower}@ist.uni-stuttgart.de

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 99–108.
springerlink.com

proposed model predictive controller, if initially feasible, stabilizes linear constrained discrete-time periodic systems. Note that the proposed MPC scheme can incorporate input and state constraints compared to existing MPC schemes for linear periodic systems [7, 11, 13] and improves the performance due to the online optimization compared to existing state feedback control laws [8, 9, 10]. Satisfaction of input and state constraints is obtained using invariant ellipsoids. Although not considered in this work, it could be of interest to apply a polyhedral set based MPC approach similar to [14] in order to obtain a less conservative constraint handling.

The remainder of the paper is organized as follows: In Section 2, the class of linear discrete-time systems is introduced and the considered control problem is formulated. Section 3 provides the main result of the paper, namely, a new MPC scheme for linear constrained discrete-time periodic systems. The applicability of the proposed MPC scheme is illustrated via an example in Section 4. Section 5 concludes the paper with a summary.

2 Problem Setup

Consider the linear N-periodic system of the form

$$x_{k+1} = A_k x_k + B_k u_k \tag{1}$$

with initial condition $x_0 = \bar{x}_0$. In (1) $x_k \in \mathbb{R}^n$ is the system state, $u_k \in \mathbb{R}^m$ the control input, $k \geq 0$ the time variable, and $A_{k+N} = A_k \in \mathbb{R}^{n\times n}$, $B_{k+N} = B_k \in \mathbb{R}^{n\times m}$ are linear periodic matrices with the time period N.

The control problem considered in this paper is to stabilize the origin of system (1) via a model predictive controller such that the state and input constraints defined by the polyhedral set

$$\mathscr{C} = \left\{ \begin{bmatrix} x_k \\ u_k \end{bmatrix} \in \mathbb{R}^{n+m} : c_j x_k + d_j u_k \leq 1, j = 1, \ldots, p \right\} \tag{2}$$

are satisfied at every time instant k, where $c_i \in \mathbb{R}^{1\times n}$ and $d_i \in \mathbb{R}^{1\times m}$. Since MPC takes constraints explicitly into account it is a suitable method to achieve the given control task. In MPC the future states of the system are predicted based on the measurement x_k at the time instant k. Therefore, a prediction model is used, which in the case of discrete-time systems can be written as

$$x_{k+i+1|k} = A_{k+i} x_{k+i|k} + B_{k+i} u_{k+i|k},\ x_{k|k} = x_k. \tag{3}$$

In the prediction model the time variable k is fixed whereas $i \geq 0$ is the free prediction time variable. Thus, the state $x_{k+i|k}$ denotes the predicted state at time instant $k+i$ (under the input $u_{k+i|k}$) that is calculated based on the state measurement x_k of system (1) at time instant k, i.e. $x_{k|k} = x_k$.

The basic idea of MPC is to solve online an optimal control problem at each time instant k based on the measured system state x_k. The underlying optimal control problem is to minimize the infinite horizon quadratic cost function

$$J_{\infty|k} = \sum_{i=0}^{\infty} x_{k+i|k}^T Q x_{k+i|k} + u_{k+i|k}^T R u_{k+i|k} \tag{4}$$

with the positive definite and symmetric weighting matrices $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$. In the following a linear state feedback law

$$u_{k+i|k} = K_{k+i|k} x_{k+i|k} \tag{5}$$

is considered, where $K_{k+i|k} \in \mathbb{R}^{m\times n}$, $i = 0,\ldots,\infty$, is a time-variant feedback matrix. Substituting (5) in (4), the cost function can be written as

$$J_{\infty|k} = \sum_{i=0}^{\infty} x_{k+i|k}^T (Q + K_{k+i|k}^T R K_{k+i|k}) x_{k+i|k}. \tag{6}$$

Minimizing this cost function at the time instant k would require the calculation of an infinite number of feedback matrices. This is practically impossible. The key idea of the approach presented in this paper is to calculate a control law (5) defined by an N-periodic feedback matrix, i.e. $K_{k+i+N|k} = K_{k+i|k}$. The underlying controller design problem is formulated as a convex optimization problem which minimizes an upper bound on the considered cost function (6). In the next section an MPC approach is derived where the periodic feedback matrix is recalculated after one time period N, i.e. the feedback law (5) is updated at each time instant $\kappa_r = rN$, $r = 0,\ldots,\infty$, based on the measurement of the system state x_{κ_r}. The obtained control law is then applied to system (1) for one time period N, i.e. for $k \in [\kappa_r, \kappa_r + N - 1]$. At the next recalculation instant $\kappa_{r+1} = (r+1)N$ a new feedback matrix $K_{\kappa_{r+1}+i|\kappa_{r+1}}$ is computed based on the new measured system state $x_{\kappa_{r+1}}$. Hence, the proposed model predictive controller requires the recalculation only once in each period while in classical MPC approaches the control input is computed at every time instant. In the next section the model predictive controller is introduced in more detail.

3 MPC For Linear Periodic Systems

The MPC approach proposed in this section is based on the minimization of an upper bound on cost function (6) at time instants $\kappa_r = rN$, $r = 0,\ldots,\infty$. To simplify notation, in the remainder of this paper the index of the time instant κ_r is skipped. Thus, κ represents an arbitrary multiple of the time period N. The following lemma, often referred to as periodic Lyapunov lemma, is useful in the stability analysis and feedback design of N-periodic systems, see e.g. [3, 5, 8].

Lemma 1. *It is shown in [3] and [5] that system (1) with $u_k = 0$ is asymptotically stable if and only if there exits an N-periodic positive definite matrix $P_k \in \mathbb{R}^{n\times n}$, i.e. $P_{k+N} = P_k$, such that $A_k^T P_{k+1} A_k - P_k < 0$ holds for all $k \geq 0$. Furthermore, this inequality holds if the following LMIs with $W_k = P_k^{-1}$ hold:*

$$\begin{bmatrix} -W_k & W_k A_k^T \\ A_k W_k & -W_{k+1} \end{bmatrix} < 0, \quad k = 0,1,\ldots,N-1. \tag{7}$$

Now, an upper bound on the cost function (6) is derived via conditions for the calculation of a control law (5) that stabilizes the prediction model (3).

Theorem 1. *Consider the N-periodic prediction model defined in (3) without constraints (2). Assume that there exist matrices* $0 < \Lambda_{\kappa+i|\kappa} \in \mathbb{R}^{n\times n}$, $\Gamma_{\kappa+i|\kappa} \in \mathbb{R}^{m\times n}$ *and a positive constant* α_κ *such that the LMIs*

$$\begin{bmatrix} \Lambda_{\kappa+i|\kappa} & \Delta^T_{\kappa+i|\kappa} & \Lambda_{\kappa+i|\kappa}Q^{\frac{1}{2}} & \Gamma^T_{\kappa+i|\kappa}R^{\frac{1}{2}} \\ \Delta_{\kappa+i|\kappa} & \Lambda_{\kappa+i+1|\kappa} & 0 & 0 \\ Q^{\frac{1}{2}}\Lambda_{\kappa+i|\kappa} & 0 & \alpha_\kappa I & 0 \\ R^{\frac{1}{2}}\Gamma_{\kappa+i|\kappa} & 0 & & \alpha_\kappa I \end{bmatrix} \geq 0, \tag{8}$$
$$\Lambda_{\kappa+N|\kappa} = \Lambda_{\kappa|\kappa},\ i = 0,\dots,N-1$$

with $\Delta_{\kappa+i|\kappa} = A_{\kappa+i}\Lambda_{\kappa+i|\kappa} + B_{\kappa+i}\Gamma_{\kappa+i|\kappa}$ *are satisfied at the time instant* κ*. Then the control law (5) with the N-periodic matrix* $K_{\kappa+i+N|\kappa} = K_{\kappa+i|\kappa} = \Gamma_{\kappa+i|\kappa}\Lambda^{-1}_{\kappa+i|\kappa}$*, where* $\Lambda^{-1}_{\kappa+i|\kappa}\alpha_\kappa = P_{\kappa+i|\kappa} = P_{\kappa+i+N|\kappa}$*, asymptotically stabilizes the origin of the prediction model (3). Furthermore,* $x^T_\kappa P_{\kappa|\kappa}x_\kappa$ *upper bounds the cost function (6) at time instant* κ*, where* x_κ *is the state of system (1) at* κ*.*

Proof. Substituting $\Delta_{\kappa+i|\kappa}$, $\Gamma_{\kappa+i|\kappa}$, $\Lambda_{\kappa+i|\kappa}$ and α_k in the LMIs (8) as defined in the theorem by $A_{\kappa+i}$, $B_{\kappa+i}$, $P^{-1}_{\kappa+i|\kappa}$ and $K_{\kappa+i|\kappa}$ and applying the Schur complement one obtains

$$P^{-1}_{\kappa+i|\kappa} - P^{-1}_{\kappa+i|\kappa}\Big(Q + K^T_{\kappa+i|\kappa}RK_{\kappa+i|\kappa} + \tilde{A}^T_{\kappa+i|\kappa}P_{\kappa+i+1|\kappa}\tilde{A}_{\kappa+i|\kappa}\Big)P^{-1}_{\kappa+i|\kappa} \geq 0, \tag{9}$$

where $\tilde{A}_{\kappa+i|\kappa} = A_{\kappa+i} + B_{\kappa+i}K_{\kappa+i|\kappa}$ and $i = 0,\dots,N-1$. Using the periodicity of system (3) and of the matrices $K_{\kappa+i|\kappa}$ and $P_{\kappa+i|\kappa}$, one can show after some straightforward manipulations that if inequality (9) is satisfied the inequality

$$\begin{aligned}(A_{\kappa+i}+B_{\kappa+i}K_{\kappa+i|\kappa})^T P_{\kappa+i+1|\kappa}(A_{\kappa+i}+B_{\kappa+i}K_{\kappa+i|\kappa}) \\ -P_{\kappa+i|\kappa} + Q + K^T_{\kappa+i|\kappa}RK_{\kappa+i|\kappa} \leq 0,\end{aligned} \tag{10}$$

holds for $i = 0,\dots,\infty$. Since the matrices Q and R are positive definite, one can conclude from Lemma 1 that the considered prediction model (3) is asymptotically stabilized by the N-periodic feedback matrix $K_{\kappa+i|\kappa}$. This implies that $\lim_{i\to\infty}\|x_{\kappa+i|\kappa}\| = 0$. Furthermore, it follows from (10) that

$$x^T_{\kappa+i|\kappa}\Big(\tilde{A}^T_{\kappa+i}P_{\kappa+i+1|\kappa}\tilde{A}_{\kappa+i} - P_{\kappa+i|\kappa} + Q + K_{\kappa+i|\kappa}RK_{\kappa+i|\kappa}\Big)x_{\kappa+i|\kappa} \leq 0 \tag{11}$$

is satisfied for all $i \geq 0$. Using (3) one obtains that

$$\begin{aligned}x^T_{\kappa+i|\kappa}P_{\kappa+i|\kappa}x_{\kappa+i|\kappa} - x^T_{\kappa+i+1|\kappa}P_{\kappa+i+1|\kappa}x_{\kappa+i+1|\kappa} \\ \geq x^T_{\kappa+i|\kappa}(Q + K^T_{\kappa+i|\kappa}RK_{\kappa+i|\kappa})x_{\kappa+i|\kappa}\end{aligned} \tag{12}$$

holds for all $i \geq 0$. The sum of this inequality from $i = 0$ to $i \to \infty$ is

$$x_{\kappa|\kappa}^T P_{\kappa|\kappa} x_{\kappa|\kappa} \geq \sum_{i=0}^{\infty} x_{\kappa+i|\kappa}^T (Q + K_{\kappa+i|\kappa}^T R K_{\kappa+i|\kappa}) x_{\kappa+i|\kappa} = J_{\infty|\kappa}. \tag{13}$$

Hence, $x_{\kappa|\kappa}^T P_{\kappa|\kappa} x_{\kappa|\kappa} = x_{\kappa}^T P_{\kappa|\kappa} x_{\kappa}$ is an upper bound on the considered cost function (6) at time instant κ. ■

In Theorem 1 LMI conditions to calculate an upper bound on the cost function (6) for linear periodic systems have been introduced. The obtained results will be used to derive an MPC controller that minimizes this upper bound at each time instant κ. Besides the minimization of an upper bound, the model predictive controller has to satisfy input and state constraints as introduced in (2). Since we consider the time-variant control law (5), the constraint set (2) clearly can be written in a time-variant form only dependent on the state x:

$$\mathscr{C}_{\kappa+i|\kappa} = \left\{ x \in \mathbb{R}^n : (c_j + d_j K_{\kappa+i|\kappa}) x \leq 1 \right\}, \tag{14}$$

where $j = 1, \ldots, p$ and $i = 0, \ldots, \infty$. To guarantee satisfaction of these constraints the following lemma is useful, see e.g. [6].

Lemma 2. *The ellipsoid* $\mathscr{E}_{\kappa+i|\kappa} = \{x \in \mathbb{R}^n : x^T P_{\kappa+i|\kappa} x \leq \alpha_\kappa\}$ *is contained in the constraint set* $\mathscr{C}_{\kappa+i|\kappa}$ *at the time instant* $\kappa + i$ *if and only if*

$$(c_j + d_j K_{\kappa+i|\kappa}) \alpha_\kappa P_{\kappa+i|\kappa}^{-1} (c_j + d_j K_{\kappa+i|\kappa})^T \leq 1, \ \forall j = 1, \ldots, p. \tag{15}$$

Hence, it follows from Lemma 2 that if the predicted state $x_{\kappa+i|\kappa}$ at the time instant $\kappa + i$ is contained in the ellipsoid $\mathscr{E}_{\kappa+i|\kappa}$, then the input and state constraints are satisfied at this time instant. Using the results of Theorem 1 and Lemma 2 the next theorem proposes a model predictive controller that asymptotically stabilizes system (1) without violating the constraints (2), and which minimizes an upper bound on the cost function (6) at each time instant κ.

Theorem 2. *Consider the N-periodic system (1), the constraints (2) and the cost function (6). The model predictive controller with the convex optimization problem*

$$\min_{\alpha_\kappa, \Lambda_{\kappa+i|\kappa}, \Gamma_{\kappa+i|\kappa}} \alpha_\kappa, \tag{16}$$

subject to

$$\begin{bmatrix} 1 & x_\kappa^T \\ x_\kappa & \Lambda_{\kappa|\kappa} \end{bmatrix} \geq 0, \tag{17}$$

$$\begin{bmatrix} \Lambda_{\kappa+i|\kappa} & \Delta_{\kappa+i|\kappa}^T & \Lambda_{\kappa+i|\kappa} Q^{\frac{1}{2}} & \Gamma_{\kappa+i|\kappa}^T R^{\frac{1}{2}} \\ \Delta_{\kappa+i|\kappa} & \Lambda_{\kappa+i+1|\kappa} & 0 & 0 \\ Q^{\frac{1}{2}} \Lambda_{\kappa+i|\kappa} & 0 & \alpha_\kappa I & 0 \\ R^{\frac{1}{2}} \Gamma_{\kappa+i|\kappa} & 0 & 0 & \alpha_\kappa I \end{bmatrix} \geq 0, \tag{18}$$

$$\begin{bmatrix} 1 & c_j \Lambda_{\kappa+i|\kappa} + d_j \Gamma_{\kappa+i|\kappa} \\ (c_j \Lambda_{\kappa+i|\kappa} + d_j \Gamma_{\kappa+i|\kappa})^T & \Lambda_{\kappa+i|\kappa} \end{bmatrix} \geq 0, \tag{19}$$

$$\Lambda_{\kappa+N|\kappa} = \Lambda_{\kappa|\kappa}, \ i = 0, \ldots, N-1, \ j = 1, \ldots, p,$$

that is solved at each time instant κ, where $\Lambda_{\kappa+i|\kappa} > 0$, $\Delta_{\kappa+i|\kappa} = A_{\kappa+i}\Lambda_{\kappa+i|\kappa} + B_{\kappa+i}\Gamma_{\kappa+i|\kappa}$, $P_{\kappa+i|\kappa} = \Lambda^{-1}_{\kappa+i|\kappa}\alpha_\kappa$, $K_{\kappa+i|\kappa} = \Gamma_{\kappa+i|\kappa}\Lambda^{-1}_{\kappa+i|\kappa}$, and x_k is the measured state of system (1), has the following properties: (a) The upper bound $x^T_\kappa P_{\kappa|\kappa} x_\kappa$ on the cost function (6) is minimized at each time instant κ. (b) The predicted states $x_{\kappa+i|\kappa}$ and inputs $u_{\kappa+i|\kappa}$ satisfy state and input constraints defined in (2) for all $i \geq 0$. (c) If the optimization problem (16)-(19) of the proposed model predictive controller is feasible at time instant $\kappa = 0$, then it is feasible at all future time instants κ. (d) The control signal u_k for $k \in [\kappa, \kappa+N-1]$ of the model predictive controller defined by

$$u_{\kappa+i} = K_{\kappa+i|\kappa} x_{\kappa+i},\ i = 0,\ldots,N-1, \tag{20}$$

guarantees attractivity of the origin of system (1), i.e. $\|x_k\| \to 0$ as $k \to \infty$, without violating the constraints (2).

Proof. The proof is divided into four parts in order to show separately that the properties (a)-(d) hold.

Part (a): The LMIs (18) correspond to the LMIs (8) in Theorem 1. Therefore, it is known that $x^T_\kappa P_{\kappa|\kappa} x_\kappa$ is an upper bound on cost function (6) at the time instant κ. The LMI (17) is equivalent to the inequality $x^T_\kappa P_{\kappa|\kappa} x_\kappa \leq \alpha_\kappa$. Thus, minimizing α_κ implies minimizing this upper bound on the cost function.

Part (b): The LMIs (18) satisfy the conditions of Theorem 1. Therefore, the N-periodic feedback matrix $K_{\kappa+i|\kappa}$ stabilizes the origin of the prediction model (3). Furthermore, from inequality (12) in the proof of Theorem 1 it follows with $Q > 0$ and $R > 0$ that

$$x^T_{\kappa+i+1|\kappa} P_{\kappa+i+1|\kappa} x_{\kappa+i+1|\kappa} - x^T_{\kappa+i|\kappa} P_{\kappa+i|\kappa} x_{\kappa+i|\kappa} < 0 \tag{21}$$

holds for all $i \geq 0$. Combining this with (17) one obtains

$$x^T_{\kappa+i+1|\kappa} P_{\kappa+i+1|\kappa} x_{\kappa+i+1|\kappa} < x^T_{\kappa+i|\kappa} P_{\kappa+i|\kappa} x_{\kappa+i|\kappa} \leq \alpha_\kappa,\ i \geq 0. \tag{22}$$

Therefore, the state $x_{\kappa+i|\kappa}$ lies in the ellipsoid

$$\mathscr{E}_{\kappa+i|\kappa} = \left\{x \in \mathbb{R}^n : x^T P_{\kappa+i|\kappa} x \leq \alpha_\kappa\right\}. \tag{23}$$

Using the results of Lemma 2, the ellipsoids $\mathscr{E}_{k+i|k}$, $i = 1,\ldots,\infty$, lie in the corresponding constraint sets (14) if and only if (15) holds for all $j = 1,\ldots,p$ and all $i \geq 0$. Since $P_{\kappa+i|\kappa}$ and $K_{\kappa+i|\kappa}$ are N-periodic matrices it is obvious that (15) holds for all $i \geq 0$ if it holds for all $i = 0,\ldots,N-1$. With some straightforward manipulations it can be shown that (19) is equivalent to (15). Thus, it has been shown that the LMIs (19) imply that the state $x_{\kappa+i|\kappa}$ lies in the ellipsoid $\mathscr{E}_{\kappa+i|\kappa}$ which lies in the constraint set $\mathscr{C}_{\kappa+i|\kappa}$ at each time instant $\kappa+i$. Therefore, the predicted sequences $x_{\kappa|\kappa}, x_{\kappa+1|\kappa}, \ldots, x_{\infty|\kappa}$ and $u_{\kappa|\kappa}, u_{\kappa+1|\kappa}, \ldots, u_{\infty|\kappa}$ calculated at time instant κ satisfy state and input constraints at each predicted time instant $\kappa+i$.

Part (c): Suppose that the optimization problem (16)-(19) of Theorem 2 is feasible at time instant κ. It follows from inequality (22) that

$$\alpha_\kappa \geq x_{\kappa|\kappa}^T P_{\kappa|\kappa} x_{\kappa|\kappa} > x_{\kappa+1|\kappa}^T P_{\kappa+1|\kappa} x_{\kappa+1|\kappa} > \cdots$$
$$\cdots > x_{\kappa+N|\kappa}^T P_{\kappa+N|\kappa} x_{\kappa+N|\kappa} \tag{24}$$

holds. Therefore, with $x_{\kappa+N|\kappa+N} = x_{\kappa+N|\kappa}$ and $P_{\kappa+N|\kappa} = P_{\kappa|\kappa}$ the inequality

$$x_{\kappa+N|\kappa+N}^T P_{\kappa|\kappa} x_{\kappa+N|\kappa+N} < x_{\kappa|\kappa}^T P_{\kappa|\kappa} x_{\kappa|\kappa} \leq \alpha_\kappa \tag{25}$$

is satisfied. Thus, the matrix $\Lambda_{\kappa|\kappa} = P_{\kappa|\kappa}^{-1}\alpha_\kappa$ satisfies the LMI (17) at time instant $\kappa+N$. Since only (17) depends on the system state $x_{\kappa+N|\kappa+N}$, this implies that the matrices $\Lambda_{\kappa+i|\kappa}$ and $\Gamma_{\kappa+i|\kappa}$ and the constant α_κ are a feasible solution to the optimization problem (16)-(19) at time instant $\kappa+N$. By induction it follows that feasibility at $\kappa = 0$ implies feasibility at all future time instants.

Part (d): The LMIs (17)-(19) are solved at the time instant κ. The control signal applied to system (1) for $k \in [\kappa, \kappa+N-1]$ is defined by equation (20). The initial condition of the prediction model (3) equals the state of system (1), i.e. $x_{\kappa|\kappa} = x_\kappa$. It follows from (20) that the states and inputs of system (1) correspond exactly to those of the prediction model (3) in the considered time interval, i.e. $x_{\kappa+i} = x_{\kappa+i|\kappa}$, $i = 0\ldots,N$, and $u_{\kappa+i} = u_{\kappa+i|\kappa}$, $i = 0,\ldots,N-1$. Thus, from part (b) of the proof it follows that the inputs and states of system (1) satisfy the constraints at each time instant k.

What remains is to show that the control law (20) stabilizes the origin of system (1). From inequality (22) follows with the N-periodic matrix $P_{\kappa+N|\kappa} = P_{\kappa|\kappa}$ and with $x_{\kappa+i|\kappa} = x_{\kappa+i}$ for $i = 0,\ldots,N$, that

$$x_{\kappa+N}^T P_{\kappa|\kappa} x_{\kappa+N} < x_\kappa^T P_{\kappa|\kappa} x_\kappa \tag{26}$$

holds. From part (c) of the proof it is known that the solution at the recalculation instant κ, $P_{\kappa+i|\kappa}$ and $K_{\kappa+i|\kappa}$, also solve the LMIs (17)-(19) at the following recalculation instant $\kappa+N$. However, in general this solution is non-optimal. Therefore, one obtains that

$$x_{\kappa+N}^T P_{\kappa+N|\kappa+N} x_{\kappa+N} \leq x_{\kappa+N}^T P_{\kappa|\kappa} x_{\kappa+N} < x_\kappa^T P_{\kappa|\kappa} x_\kappa \tag{27}$$

holds for all κ, see [12] for a more detailed explanation. This implies that the states of system (1) at the recalculation instants $\kappa \in [0, N, 2N, \ldots, \infty)$ converge to zero, i.e. $\|x_\kappa\| \to 0$ as $\kappa \to \infty$. It still has to be shown that also the states between the recalculation instants converge to zero. From inequality (22) follows

$$x_{\kappa+i}^T P_{\kappa+i|\kappa} x_{\kappa+i} < x_\kappa^T P_{\kappa|\kappa} x_\kappa,\ i = 0,\ldots,N-1. \tag{28}$$

At each time instant κ a solution for the LMIs (17)-(19) with positive definite matrices $P_{\kappa+i|\kappa}$, $i = 0\ldots,N-1$, exists. This implies that $x_{\kappa+i}$ is bounded for $i = 0,\ldots,N-1$. Furthermore, since $\|x_\kappa\| \to 0$ as $\kappa \to \infty$ it follows from $P_{k+i|k} > 0$ that $\|x_{\kappa+i}\| \to 0, i = 0,\ldots,N-1$, as $\kappa \to \infty$. Summarizing, the MPC control law (20) guarantees attractivity of the origin of the considered periodic system (1), i.e. $\|x_k\| \to 0$ as $k \to \infty$. ■

As shown above, the repeated solution of the LMIs in Theorem 2 and the application of the obtained feedback matrices stabilizes the origin of the considered periodic system (1) without violating input and state constraints. The recalculation of the N-periodic feedback matrix $K_{\kappa+i|\kappa}$ at each time instant κ reduces the conservatism of the solution obtained at the previous time instant $\kappa - N$. An example system demonstrates these results in the following section.

4 Simulation Results

To illustrate the results of the previous section a periodic example system of the form (1) is considered. The system is of third order and has two inputs u_k^1 and u_k^2, i.e. $n = 3$ and $m = 2$. Its time period is $N = 3$. The example system is described by the matrices

$$A_0 = \begin{bmatrix} 0.4 & 0.3 & 1.0 \\ 0.6 & 0.6 & 0.0 \\ 0.1 & 0.9 & 0.1 \end{bmatrix}, \; A_1 = \begin{bmatrix} 0.4 & 0.7 & 0.8 \\ 0.1 & 1.0 & 0.1 \\ 0.3 & 0.4 & 0.8 \end{bmatrix}, \; A_2 = \begin{bmatrix} 0.3 & 0.7 & 0.3 \\ 0.8 & 0.5 & 0.3 \\ 0.6 & 0.4 & 0.7 \end{bmatrix},$$

$$B_0 = \begin{bmatrix} 0.3 & 0.3 \\ 0.5 & 1.0 \\ 0.4 & 1.0 \end{bmatrix}, \; B_1 = \begin{bmatrix} 0.2 & 0.6 \\ 0.1 & 0.1 \\ 0.3 & 0.9 \end{bmatrix}, \; B_2 = \begin{bmatrix} 0.9 & 0.0 \\ 0.5 & 0.2 \\ 0.3 & 0.8 \end{bmatrix}.$$

The initial condition for the system states is given by $x_0 = [10\ 10\ 0]^T$. For simplicity, no state constraints are considered. However, the controller has to be designed such that the inputs satisfy $-3 \leq u_k^1 \leq 3$ and $-4.5 \leq u_k^2 \leq 4.5$ for all $k \geq 0$.

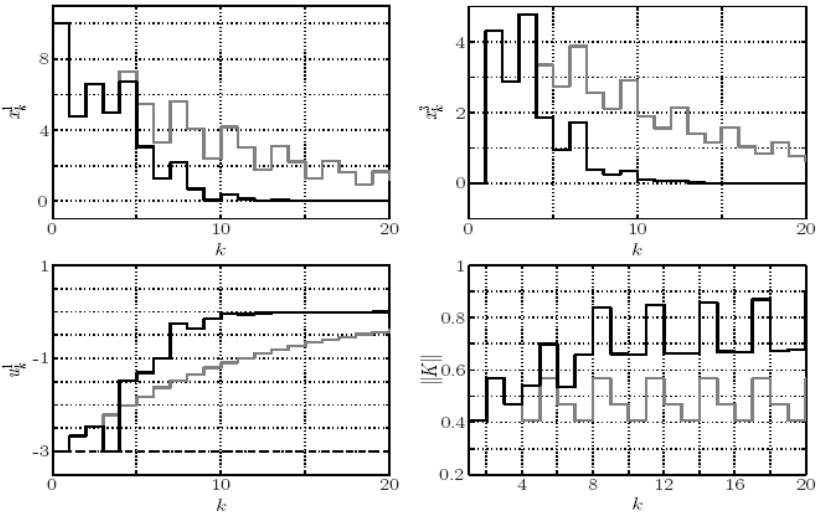


Fig. 1 Black lines: Proposed model predictive controller. Gray lines: Controller with "constant" feedback

Thus, the number of constraints is $p = 4$. Since only input constraints are considered, the state constraint vectors are $c_j = [0\ 0\ 0]\ \forall j = 1,\ldots,4$. The input constraint vectors are $d_1 = -d_2 = [\frac{1}{3}\ 0]$ and $d_3 = -d_4 = [0\ \frac{2}{9}]$. As design parameters for the predictive controller the weighting matrices $Q = \mathrm{diag}(0.001, 0.001, 0.001)$ and $R = \mathrm{diag}(0.005, 0.005)$ are chosen. The black lines in Figure 1 show the results obtained by the proposed MPC controller. To illustrate the effectiveness of the MPC approach, the results are compared with those obtained when the N-periodic feedback matrix calculated at the first time instant $k = \kappa = 0$, i.e. the sequence $K_{0|0}, K_{1|0}, K_{2|0}, K_{0|0}, K_{1|0}, \ldots$, is applied to the system without a recalculation of the feedback matrix at the recalculation instants κ. Exemplarily, Figure 1 shows the trajectories x_k^1, x_k^2, u_k^1 and the norm of the N-periodic feedback matrix. Clearly, the example system is steered to the origin much faster by the MPC controller. The reason for this is that the norm of the N-periodic feedback matrix increases after each recalculation instant leading to exploitation of the whole available input energy. The update of the feedback matrix makes the controller more aggressive and therefore, the performance of the MPC controller is significantly better than that of the controller with "constant" feedback matrix.

Summarizing, the MPC controller can use the available input sequences more efficiently since the periodic feedback matrix is updated at each time instant κ.

5 Conclusions

In this paper a model predictive controller for constrained linear discrete-time periodic systems has been proposed. The control law is obtained by the repeated solution of a convex optimization problem. It has been shown that the MPC controller stabilizes the origin of the considered system without violating the constraints. The effectiveness of the proposed approach has been demonstrated via simulations of an example system. Further work is necessary to investigate its applicability on practical control problems such as e.g. satellite control.

References

1. Arcara, P., Bittanti, S., Lovera, M.: Periodic control of helicopter rotors for attenuation of vibrations in forward flight. IEEE Transactions on Control Systems Technology 8(6), 883–894 (2000)
2. Badgwell, T.A.: Robust model predictive control of stable linear systems. International Journal of Control 68(4), 797–818 (1997)
3. Bittanti, S., Bolzern, P., Colaneri, P.: Stability analysis of linear periodic systems via the Lyapunov equation. In: Proceedings of the 9th IFAC World Congress, Budapest, Hungary, pp. 169–173 (1984)
4. Bittanti, S., Colaneri, P.: Periodic Control. In: Encyclopedia of Electrical and Electronics Engineering, pp. 59–74 (1999)
5. Bolzern, P., Colaneri, P.: The periodic Lyapunov equation. SIAM Journal on Matrix Analysis and Applications 9(4), 499–512 (1988)

6. Chen, W., Ballance, D.J.: On attraction domain of model predictive control of nonlinear systems with input/state constraints. Technical report, CSC-99009, University of Glasgow (1999)
7. De Nicolao, G.: Cyclomonotonicity, Riccati equations and periodic receding horizon control. Automatica 30(9), 1375–1388 (1994)
8. De Souza, C., Trofino, A.: An LMI approach to stabilization of linear discrete-time periodic systems. International Journal of Control 73(8), 696–703 (2000)
9. Farges, C., Peaucelle, D., Arzelier, D.: Output feedback stabilization of linear periodic systems. Technical report, LAAS-CNRS (2005)
10. Farges, C., Peaucelle, D., Arzelier, D., Daafouz, J.: Robust $\mathscr{H}_2$ performance analysis and synthesis of linear polytopic discrete-time periodic systems via LMIs. Systems & Control Letters 56(2), 159–166 (2007)
11. Kim, K.B., Lee, J.W., Kwon, W.H.: Intervalwise receding horizon $\mathscr{H}_\infty$ tracking control for discrete linear periodic systems. IEEE Transactions on Automatic Control 45(4), 747–752 (2000)
12. Kothare, M.V., Balakrishnan, V., Morari, M.: Robust constrained model predictive control using linear matrix inequalities. Automatica 32(10), 1361–1379 (1996)
13. Kwon, W.H., Byun, D.G.: Receding horizon tracking control as a predictive control and its stability properties. International Journal of Control 50(5), 1807–1824 (1989)
14. Pluymers, B.: Robust Model Based Predictive Control – An Invariant Set Approach. PhD thesis, Katholieke Universiteit Leuven (2006)
15. Psiaki, M.L.: Magnetic torquer attitude control via asymptotic periodic linear quadratic regulation. Journal of Guidance, Control, and Dynamics 24(2), 386–394 (2001)
16. Rawlings, J.B., Muske, K.R.: The stability of constrained receding horizon control. IEEE Transactions on Automatic Control 38(10), 1512–1516 (1993)

Receding Horizon Control for Linear Periodic Time-Varying Systems Subject to Input Constraints

Benjamin Kern, Christoph Böhm, Rolf Findeisen, and Frank Allgöwer

Abstract. In this paper, a receding horizon control scheme able to stabilize linear periodic time-varying systems, in the sense of asymptotic convergence, is proposed. The presented approach guarantees that input constraints are always satisfied if the optimization problem is feasible at the initial time.

Unlike the usual approaches for linear systems, a finite prediction horizon is used. Stability is ensured by choosing a time-varying terminal cost, that approximates an infinite horizon cost and is related to the solution of a Matrix Riccati differential equation. Sufficient conditions on the system for the design of its corresponding time-varying terminal region are derived, such that it is also possible to incorporate input constraints. This region is based on the time-varying terminal cost and can be calculated off-line.

Keywords: linear periodic time-varying systems, predictive control, input constraints, time-varying terminal region, periodic Riccati equation.

1 Introduction

Linear periodic time-varying systems (LPTV systems) are often encountered in practice. Typical examples are mechanical applications, in which periodic motions are present, or systems where different parameters oscillate periodically. Other examples are the rotor motion of wind-turbines [7] or the attitude control of small satellites [6], which are usually described in this way.

Benjamin Kern and Rolf Findeisen
Institute for Automation Engineering, Otto-von-Guericke University Magdeburg, Germany
e-mail: {benjamin.kern,rolf.findeisen}@ovgu.de

Christoph Böhm and Frank Allgöwer
Institute for Systems Theory and Automatic Control, University of Stuttgart, Germany
e-mail: {cboehm,allgower}@ist.uni-stuttgart.de

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 109–117.
springerlink.com

Due to the nature of these systems, more advanced control methods are necessary to achieve good performance. For instance, there are H_∞ approaches [8] or feedback techniques based on the solution of a Matrix Riccati equation [5], which can deal with these kind of systems adequately. However, the implementation of these controllers in practical applications is generally difficult, since it is hard to consider some physical limitations, in terms of input or state constraints. One way to resolve these issues is to use receding horizon control schemes, since they are generally capable of handling constraints. Design methods for nonlinear systems can be found in [1], [3].

The main contribution of this paper is the development of a receding horizon control scheme, which guarantees stability for continuous LPTV systems. It differs from the method proposed in [9], since it takes directly into account some special properties of LPTV systems. To deal with input constraints, sufficient conditions for the design of a time-varying terminal region are derived. Due to the structure of the considered systems, it is possible to calculate the terminal region and the cost off-line.

The remainder of this paper is structured as follows. In Section 2, the system class and some necessary assumptions are provided. In Section 3, the proposed approach and the main results are introduced. Finally, in Section 4, simulation results for a benchmark system are provided.

2 Problem Statement

Consider the following linear periodic time-varying system

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), \quad x(t_0) = x_0 \tag{1}$$

with $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $A(t) \in \mathbb{R}^{n\times n}, B(t) \in \mathbb{R}^{n\times m}$ for all $t \geq t_0$. The system matrices $A(t)$ and $B(t)$ are periodic with a period time T, i.e. there is a $T > 0, T \in \mathbb{R}$, such that $A(t) = A(t+T)$ and $B(t) = B(t+T)$ for all $t \geq t_0$.
For the remainder of the paper the following assumptions are made:

A1. The system (1) is uniformly controllable in t.
A2. The solution of the Matrix-Riccati-Equation

$$\dot{P}(t) = -A(t)^T P(t) - P(t)A(t) + P(t)B(t)R^{-1}B(t)^T P(t) - Q \tag{2}$$

with $Q = Q^T > 0$ and $R = R^T > 0$ is known.

Corollary 1. *Suppose assumption A1 and A2 hold. Then the solution of (2) is periodic and positive definite, i.e.*

$$P(t) = P(t+T) \text{ and } P^T(t) = P(t) > 0, \forall t \geq t_0$$

and $u_R(s) = -R^{-1}B(s)^T P(s)x$ *stabilizes the system (1) uniformly in* t.

Proof. See e.g. [5]. ■

As shown later, the stabilizing feedback law $u_R(s)$ and $P(t)$ can be used to calculate the terminal region and cost in a receding horizon control framework.

2.1 Receding Horizon Control without Input Constraints

To stabilize system (1), the quasi-infinite-horizon setup proposed in [4] is adopted to LPTV systems. In a first step, the problem is relaxed by avoiding input constraints. Assuming that $P(t)$ is known, the following optimal control problem is used:

$$J^*(x(t),t) := \min_{\bar{u}(\cdot)} J(x(\cdot),\bar{u}(\cdot)) \tag{3}$$

$$\text{s. t.: } \dot{\bar{x}}(s) = A(s)\bar{x}(s) + B(s)\bar{u}(s), \quad x(t) = x_t,$$

where $J(x(\cdot),\bar{u}(\cdot))$ is given by

$$J(x(\cdot),\bar{u}(\cdot)) = \int_t^{t+T_p} \bar{x}^T N \bar{x} + \bar{u}^T M \bar{u}\, ds + \bar{x}(t+T_p)^T P(t+T_p)\bar{x}(t+T_p),$$

where s denotes the time of the predicted state.

The sampled-data receding horizon scheme can be used to control the system, i.e at each sampling time $t_{i+1} = t_i + \delta$, the measured state x_t is used as an initial value to solve the optimization problem (3). The obtained solution $\bar{u}(\cdot)$, is then applied to the plant for a sampling time $\delta \leq T_p$, until the next measurement of the plant is available. By following this approach, it can be proved that the following theorem holds.

Theorem 1. *Consider the optimal control problem (3). If*

$$N - Q + P(t)B(t)R^{-1}(M-R)R^{-1}B(t)^T P(t) < 0, \quad t \in \mathbb{R} \tag{4}$$

with $N = N^T > 0, M = M^T > 0$, then, the sampled-data receding horizon control stabilizes the closed loop, in the sense of asymptotic convergence.

Proof. The proof follows the lines of [10] and [2]. It is shown, that there exists admissible input policies $\hat{u}_{t_i}(\cdot)$ at each sampling time $t_{i+1} = t_i + \delta$, that are able to make the optimal value of (3) strictly decreasing for all $t \geq t_0$.

Since there are no constraints, the optimization problems are feasible for each sampling time t_i. Consider, therefore, the following input policy

$$\hat{u}_{t_i+\delta}(s) = \begin{cases} \bar{u}_{t_i}(s), & s \in [t_i+\delta, t_0+T_p] \\ u_R(s), & s \in [t_0+T_p, t_0+T_p+\delta]. \end{cases} \tag{5}$$

Here, $\bar{u}_{t_i}(\cdot)$ denotes the optimal solution at sampling time t_i, which results in an optimal state trajectory $\bar{x}_{t_i}(\cdot)$. The optimal value $J^*(x(t_i),t_i)$ at time t_i is then given by

$$J^*(x(t_i),t_i) = \int_{t_i}^{t_i+T_p} \bar{x}_{t_i}^T N \bar{x}_{t_i} + \bar{u}_{t_i}^T M \bar{u}_{t_i}\, ds + \bar{x}_{t_i}(t_i+T_p)^T P(t_i+T_p)\bar{x}_{t_i}(t_i+T_p).$$

The not necessarily optimal input policy $\hat{u}_{t_{i+1}}(\cdot)$ can be used to approximate the optimal value $J^*(x(t_i+\delta),t_i+\delta)$ at sampling time t_{i+1}, i.e.

$$\begin{aligned} J^*(x(t_{i+1}),t_{i+1}) \leq & \int_{t_i+\delta}^{t_i+\delta+T_p} \hat{x}_{t_i}^T N \hat{x}_{t_i} + \hat{u}_{t_i}^T M \hat{u}_{t_i}\, ds \\ & + \hat{x}_{t_i}(t_{i+1}+T_p)^T P(t_{i+1}+T_p)\hat{x}_{t_i}(t_{i+1}+T_p). \end{aligned} \tag{6}$$

By using the fact, that for arbitrary $t \geq t_0$

$$\begin{aligned} x(t+T_p)^T P(t+T_p)x(t+T_p) &= \int_{t+T_p}^{\infty} x^T Q x + u_R^T R u_R\, ds \\ &= \int_{t+T_p}^{\infty} x^T \left(Q + P(s)B(s)R^{-1}B(s)^T P(s)\right) x\, ds, \end{aligned}$$

it can be derived that

$$\begin{aligned} J^*(x(t_{i+1}),t_{i+1}) - J^*(x(t_i),t_i) \leq & -\int_{t_i}^{t_i+T_p} \bar{x}_{t_i}^T N \bar{x}_{t_i} + \bar{u}_{t_i}^T M \bar{u}_{t_i}\, ds \\ & + \int_{t_i+T_p}^{t_i+T_p+\delta} x^T (N - Q + P(s)B(s)R^{-1}(M-R)R^{-1}B(s)^T P(s)) x\, ds. \end{aligned}$$

With (4), it follows that

$$J^*(x(t_{i+1}),t_{i+1}) - J^*(x(t_i),t_i) \leq -\int_{t_i}^{t_i+\delta} \bar{x}^T N \bar{x}\, ds. \tag{7}$$

As expected, the optimal value at sampling time t_{i+1} is strictly smaller than the value at t_i. By introducing the "*MPC-Value-Function*" $V^\delta(t,x)$ presented in [10], it can then be shown that

$$V^\delta(t,\bar{x}(t)) = J^*(x(s),t_i) \leq J^*(x(s),t_0) - \int_{t_0}^{t} \bar{x}^T N \bar{x}\, ds.$$

Since $\hat{u}_{t_i+\delta}(\cdot)$ is admissible at every sampling time t_i, $J^*(x(s),t_i)$ can be approximated at every sampling time t_i with (6), i.e. $J^*(x(s),t_i)$ and consequently the "*MPC-Value-Function*" $V^\delta(t,x)$ are finite for every t. Therefore, from (7), it can be easily concluded that $\int_0^\infty \bar{x}^T N \bar{x}\, ds$ exists and is finite, since the "*MPC-Value-Function*" is finite at all times t. Eventually, it follows from a variant of Barlabat's Lemma, that $x(t)$ converges to the origin. ■

2.2 Receding Horizon Control with Input Constraints

So far, no input constraints have been considered in the problem setup. If input constraints are present, the input policy (5) is still admissible at every sampling time whenever a suitable terminal region is chosen. In this case, the results from Theorem 1 are still applicable. Consider, in fact, the modified optimization problem where $J(x(\cdot),\bar{u}(\cdot))$ is given by

$$\min_{\bar{u}} J(x(\cdot),\bar{u}(\cdot)) = \int_t^{t+T_p} \bar{x}^T N\bar{x} + \bar{u}^T M\bar{u}\, ds + \bar{x}(t+T_p)^T P(t+T_p)\bar{x}(t+T_p) \qquad (8)$$

subject to:

$$\begin{aligned} &\dot{\bar{x}}(s) = A(s)\bar{x}(s) + B(s)\bar{u}(s), x(t) = x_t \\ &\bar{u}(s) \in \mathscr{U}\,, \text{ with compact set } \mathscr{U} \\ &\bar{x}(t_i+T_p) \in \Omega_{t_i+T_p}. \end{aligned}$$

The following theorem holds:

Theorem 2. *Consider the optimal control problem (8). If Ω_t is chosen such that*

$$\Omega_t = \left\{x \in \mathbb{R}^n | x(t)^T P(t) x(t) \le \alpha\right\}, \text{ and}$$

(i) the upper bound α is obtained from

$$\begin{aligned} \alpha = &\min_{t_0 \le \tau \le t_0+T} \max_{x\in\mathbb{R}} \beta, \\ \text{subject to: } &x(\tau)^T P(\tau) x(\tau) \le \beta, \\ &u_R := -R^{-1}B(\tau)^T P(\tau) x \in \mathscr{U}, \end{aligned}$$

(ii) for each state $x(t) \in \Omega_t \Rightarrow u_R(\tau) \subset \mathscr{U} \quad \forall\, \tau \ge t$,
(iii) $N - Q + P(t)B(t)R^{-1}(M-R)R^{-1}B(t)^T P(t) < 0, \quad t \in \mathbb{R}$,

with $N = N^T > 0, M = M^T > 0$, then, the sampled-data receding horizon control stabilizes the closed loop, in the sense of asymptotic convergence.

Proof. The idea is to show that the same input policy from Theorem 1 is admissible at every sampling time $t_{i+1} = t_i + \delta$, provided that Ω_t is chosen adequality. Thus, the same argumentation can be used to show that the optimal value $J^*(x(t_i),t_i)$ is decreasing, leading to the same results of Theorem 1. Consider a modified terminal region at $\overline{\Omega}_\zeta$ at the time $\zeta := t_0 + T_p$, given by $\overline{\Omega}_\zeta = \left\{x \in \mathbb{R}^n | x(\zeta)^T P(\zeta) x(\zeta) \le \bar{\alpha}\right\}$, where $\bar{\alpha}$ is chosen, such that the condition (ii) is fulfilled and

$$\begin{aligned} \bar{\alpha} = &\max_{x\in\mathbb{R}} \beta \\ \text{subject to: } &x(\zeta)^T P(\zeta) x(\zeta) \le \beta \\ &\Pi_\zeta := -R^{-1}B(\zeta)^T P(\zeta) x \in U. \end{aligned}$$

Assume now that the modified optimization problem is feasible at t_0, i.e. there is $\bar{u}_{t_0}(\cdot)$ that transfers every initial state x_0 into the terminal region $\overline{\Omega}_\zeta$. Next it is

shown, how to choose a region $\overline{\Omega}_{\zeta+\delta}$, for the optimization problem at time $t_1 = t_0 + \delta$, such that $\hat{u}_{t_0+\delta}(\cdot)$ becomes admissible. Using condition (ii), it is possible to approximate the terminal cost at every time $\tau \geq t_0$, with

$$\begin{aligned} x(\tau)^T P(\tau) x(\tau) &= \int_{\tau}^{\infty} x^T Q x + u_R^T R u_R \, ds \\ &= \int_{\tau}^{\tau+\varepsilon} x^T Q x + u_R^T R u_R \, ds + \int_{\tau+\varepsilon}^{\infty} x^T Q x + u_R^T R u_R \, ds \\ &= \int_{\tau}^{\tau+\varepsilon} x^T Q x + u_R^T R u_R ds + x(\tau+\varepsilon)^T P(\tau+\varepsilon) x(\tau+\varepsilon). \end{aligned} \tag{9}$$

This implies that there is always a relation between the value of the terminal cost at the times τ and $\tau+\varepsilon$, when $u_R(\cdot)$ is applied to the system.

Now suppose, that $u_R(\tau)$ is applied to the system for all $\tau \in [\zeta, \zeta+\varepsilon]$ with $\varepsilon < \infty$, then according to (9) and the definition of $\overline{\Omega}_{\zeta}$,

$$x(\zeta+\varepsilon)^T P(\zeta+\varepsilon) x(\zeta+\varepsilon) + \int_{\zeta}^{\zeta+\varepsilon} x^T Q x + u_R^T R u_R ds \leq \bar{\alpha}, \quad \forall x(\zeta) \in \overline{\Omega}_{\zeta}.$$

But, this is equal to

$$\begin{aligned} &x(\zeta+\varepsilon)^T P(\zeta+\varepsilon) x(\zeta+\varepsilon) \leq \bar{\alpha} - \int_{\zeta}^{\zeta+\varepsilon} x^T (Q + P(s) B(s) R^{-1} B(s)^T P(s)) x \, ds, \\ &\forall x(\zeta) \in \overline{\Omega}_{\zeta}. \end{aligned}$$

Since $Q > 0$ and $R > 0$, there is also a non-negative upper bound $\beta(\varepsilon)$ for the second term on the right hand, i.e. $\exists \beta(\varepsilon) \geq 0$ for all $\varepsilon \geq 0$ such that

$$0 \leq \int_{\zeta}^{\zeta+\varepsilon} x^T (Q + P(s) B(s) R^{-1} B(s)^T P(s)) x \, ds \leq \beta(\varepsilon)$$

and thus for all $\varepsilon \geq 0$

$$x(\zeta+\varepsilon)^T P(\zeta+\varepsilon) x(\zeta+\varepsilon) \leq \bar{\alpha} - \beta(\varepsilon), \quad \forall x(\zeta) \in \overline{\Omega}_{\zeta}. \tag{10}$$

Now consider the following two time-dependent regions, for $\varepsilon \in (\zeta, \zeta+\delta]$

$$\begin{aligned} E(\varepsilon) &= \left\{ x \in \mathbb{R}^n | x(\zeta+\varepsilon)^T P(\zeta+\varepsilon) x(\zeta+\varepsilon) \leq \bar{\alpha} \right\} \\ E_{\beta}(\varepsilon) &= \left\{ x \in \mathbb{R}^n | x(\zeta+\varepsilon)^T P(\zeta+\varepsilon) x(\zeta+\varepsilon) \leq \bar{\alpha} - \beta(\varepsilon) \right\}. \end{aligned}$$

Suppose the optimization problem at sampling time t_1 has the terminal region $\Omega_{t_1} := E(\delta)$. Obviously, $E_{\beta}(\varepsilon)$ is a subset of $E(\varepsilon)$ for every $\varepsilon \in (\zeta, \zeta+\delta]$, provided $u_R(\cdot)$ is applied to the system (see Figure 1). By definition, $\hat{u}_{t_0+\delta}(\tau) = u_R(\tau)$ for all $\tau \in [\zeta, \zeta+\delta]$. Thus, the input $\hat{u}_{t_0+\delta}(\cdot)$ transfers each state into the region $E(\delta)$. However, $E(\delta)$ does not necessarily lie in the subset $\Pi_{\zeta+\delta}$. By modifying $\overline{\Omega}_{\zeta}$, where a new $\bar{\alpha}$ satisfies condition (ii) and is calculated by

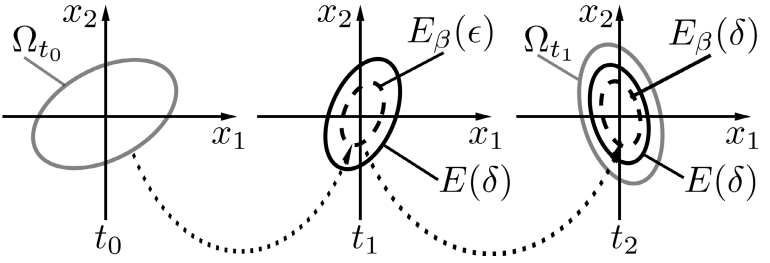


Fig. 1 Regions $E(\varepsilon)$, $E_\beta(\varepsilon)$ and Ω_ζ

$$\bar{\alpha} = \min_{\zeta \le \tau \le \zeta+\delta} \max_{x \in \mathbb{R}} \beta$$

$$\text{subject to: } x(\tau)^T P(\tau) x(\tau) \le \beta$$

$$\Pi_\tau := -R^{-1} B(\tau)^T P(\tau) x \in U$$

it can be concluded that $E(\delta)$ is a subset of $\Pi_{\zeta+\delta}$. Thus $\hat{u}_{t_0+\delta}(\cdot)$ becomes an admissible solution of the optimization problem at time t_1. By repeating the same procedure it can be concluded, that $\hat{u}_{t_i+\delta}(\cdot)$ is admissible at every sampling time. Furthermore, since $P(t)$, $B(t)$ and are time-periodic, the parameter β from the min-max problem will be periodic as well. Thus it is sufficient to calculate α from the min-max problem for the first time period. Finally, since $\hat{u}_{t_i+\delta}(\cdot)$ is admissible at every samlping time, the results from Theorem 1 still valid. ■

Note that condition (ii) from Theorem 2 is difficult to check. However, this condition is necessary since it avoids problems with the time-changing constraint set Π_τ. It is possible to show that this condition is not needed anymore, if the min-max problem is solved with a constraint set Π, which is the smallest static set that is contained in Π_τ.

3 Simulation Results

A LPTV system of the form (1) is considered. The matrices $A(t)$ and $B(t)$ are given by:

$$A(t) = \begin{bmatrix} 0 & 1 \\ 1 & 2 + 0.4 \cdot \sin(t) \end{bmatrix}, \quad B(t) = \begin{bmatrix} 1 + 0.3 \cdot \sin(t) \\ 1 + 0.3 \cdot \sin(t) \end{bmatrix}. \tag{11}$$

We want to show that the scheme stabilizes the system, while satisfying the input constraint $|u(t)| \le 1$. In addition, the variation of the time-varying terminal region is illustrated, showing that it is changing noticably at different sampling times.

The solution of the Matrix-Riccati equation is approximated by backwards integration for a sufficiently large time. The actual computation of the min-max problem is approximated by discretizing one time period in 63 different time instants. Then, at each time instant t_i, for the corresponding period, the maximum $\beta(t_i)$ was computed. The actual α is approximated by minimizing $\beta(t_i)$, i.e. $\alpha = \min \beta(t_i)$. As

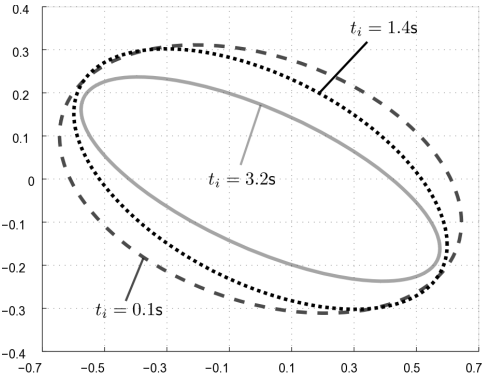


Fig. 2 Terminal region Ω_{t_i} at different sampling times

mentioned, condition (ii) from Theorem 2 is generally difficult to check. The computation is therefore simplified by approximating the terminal region with a suitable rectangle at every sampling time in a specific period. After that, only initial states at the corners and points on the vertices of each rectangle are tested off-line in a heurictic manner.

In Figure 2 the terminal region Ω_t at some specific sampling times is depicted. It can be seen, that the region is in fact changing noticeably. Figure 3 illustrates, that the receding horizon setup stabilizes this system and additionally satisfies the input constraint.

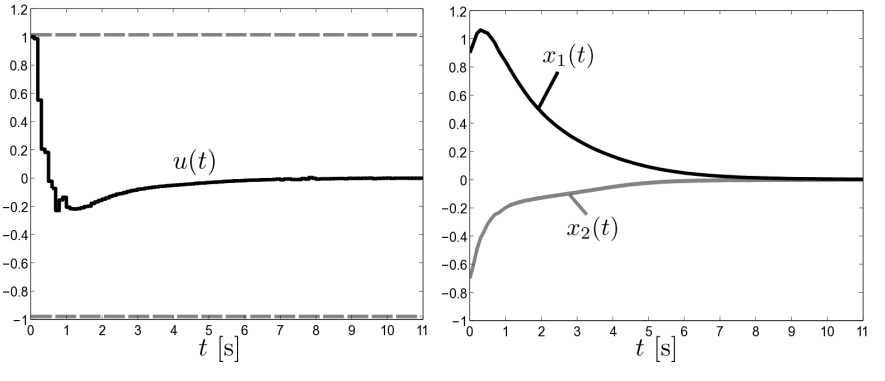


Fig. 3 Simulation Results with initial condition $x_0 = [0.9, -0.7]$

4 Conclusion

A receding horizon control setup for LPTV system has been presented. The scheme makes use of the fact that the solution of a Matrix-Riccati equation for uniformly controllable systems is periodic. Because of this periodic nature, it is possible to choose a suitable representation for the terminal region, such that input constraints could be explicitly considered. Although it is generally difficult to check the

conditions for the suitable choice of the terminal region, it is possible to enlarge it by making it time-varying.

Acknowledgements. This work was supported by the German Research Foundation (DFG) under AL-316/5-1. The first author would also like to acknowledge the support of the International Max Planck Research School for Analysis, Design, and Optimization in Chemical and Biochemical Process Engineering, Magdeburg.

References

1. Mayne, D.Q., Michalska, H.: Receding horizon control of nonlinear systems. IEEE Trans. Automat. Contr. 35(7), 814–824 (1990)
2. Findeisen, R.: Nonlinear Model Predictive Control: A Sampled-Data Feedback Approach. PhD thesis, University of Stuttgart (2004)
3. Chen, H., Allgöwer, F.: A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. Automatica 34(10), 1205–1217 (1998)
4. Findeisen, R., Allgöwer, F.: The quasi-infinite horizon approach to nonlinear model predictive control. In: Zinober, A., Owens, D. (eds.) Nonlinear and Adaptive Control. LNCIS, pp. 89–105. Springer, Berlin (2002)
5. Bittanti, S., Colaneri, P., Guardabassi, G.: Periodic solutions of periodic riccati equations. IEEE Trans. Automat. Contr. 29(7), 665–667 (1984)
6. Lovera, M., De Marchi, E., Bittanti, S.: Periodic attitude control techniques for small satellites with magnetic actuators. IEEE Trans. Contr. Sys. Tech. 10(1), 90–95 (2002)
7. Dugundji, J., Wendell, J.H.: Some analysis methods for rotating systems with periodic coefficients. AIAA Journal 21(6), 890–897 (1983)
8. Bamieh, B.A., Pearson, J.B.: A general framework for linear periodic systems with applications to H^∞ sampled-data control. IEEE Trans. Automat. Contr. 37(4), 418–435 (1992)
9. De Nicolao, G., Magni, L., Scattolini, R.: Stabilizing receding-horizon control of nonlinear time-varying systems. IEEE Trans. Automat. Contr. 43(7), 1030–1036 (1998)
10. Fontes, F.A.: A general framework to design stabilizing nonlinear model predictive controllers. Sys. Contr. Lett. 42(2), 127–143 (2001)

Optimizing Process Economic Performance Using Model Predictive Control

James B. Rawlings and Rishi Amrit

Keywords: optimal control, constrained control, process economics, unreachable setpoints.

1 Introduction

The current paradigm in essentially all industrial advanced process control systems is to decompose a plant's economic optimization into two levels. The first level performs a steady-state optimization. This level is usually referred to as real-time optimization (RTO). The RTO determines the economically optimal plant operating conditions (setpoints) and sends these setpoints to the second level, the advanced control system, which performs a dynamic optimization. Many advanced process control systems use some form of model predictive control or MPC for this layer. The MPC uses a dynamic model and regulates the plant dynamic behavior to meet the setpoints determined by the RTO.

This paper considers aspects of the question of how to use the dynamic MPC layer to optimize directly process economics. We start with the problem of a setpoint that becomes unreachable due to the system constraints. A popular method to handle this problem is to transform the unreachable setpoint into a reachable steady-state target using a separate steady-state optimization. This paper explores the alternative approach in which the unreachable setpoint is retained in the controller's stage cost and objective function. The use of this objective function induces an interesting fast/slow asymmetry in the system's tracking response that depends on the system initial condition, speeding up approaches to the unreachable setpoint, but slowing down departures from the unreachable setpoint.

James B. Rawlings and Rishi Amrit
Department of Chemical and Biological Engineering, University of Wisconsin, Madison, WI
e-mail: rawlings@engr.wisc.edu, amrit@wisc.edu

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 119–138.
springerlink.com

Analysis of the closed-loop properties of this approach for linear dynamic models is summarized next. This problem formulation leads to consideration of optimal control problems with unbounded cost. The first studies of this class of problems arose in the economics literature in the 1920s in which the problem of interest was to determine optimal savings rates to maximize capital accumulation. Much of the economics literature focused on establishing existence results for optimal control with infinite horizon and unbounded cost, and the famous "turnpike" theorems that characterize the optimal trajectory. We provide a brief summarizing overview of these optimal control results.

Next we consider the case of replacing the setpoint objective function with an objective measuring economic performance. Many such objective functions are possible. We focus attention on the strictly convex stage cost and analyze the nominal closed-loop stability of this class of controller. We compare performance of this new controller to the standard RTO/MPC approach to this same problem.

The paper concludes by briefly presenting open issues and promising areas of future research. Topics of special interest and high potential impact include: nonlinear dynamic models, distributed implementation, and robustness to disturbances and model errors.

2 Overview of the Process Control Literature

Morari et al. [31] state that the objective in the synthesis of a control structure is to *translate the economic objectives into process control objectives*. As mentioned before, in most industrial advanced control systems, the goal of optimizing dynamic plant economic performance is addressed by a control structure that splits the problem into two levels [26]. More generally, the overall plant hierarchical planning and operations structure is summarized in numerous books, for example Findeisen et al. [12], Marlin [25], Luyben et al. [24]. Planning focuses on economic forecasts and provides production goals. It answers questions like what feed-stocks to purchase, which products to make and how much of each product to make. Scheduling addresses the timing of actions and events necessary to execute the chosen plan, with the key consideration being feasibility. The planning and scheduling unit also provides parameters of the cost functions (e.g. prices of products, raw materials, energy costs) and constraints (e.g. availability of raw materials). The RTO is concerned with implementing business decisions in real time based on a fundamental steady-state model of the plant. It is based on a profit function of the plant and it seeks additional profit based on real-time analysis using a calibrated nonlinear model of the process. The data are first analyzed for stationarity of the process and, if a stationary situation is confirmed, reconciled using material and energy balances to compensate for systematic measurement errors. The reconciled plant data are used to compute a new set of model parameters (including unmeasured external inputs) such that the plant model represents the plant as accurately as possible at the current (stationary) operating point. Then new values for critical state variables of the plant are computed that optimize an economic cost function while meeting the constraints

imposed by the equipment, the product specifications, and safety and environmental regulations as well as the economic constraints imposed by the plant management system. These values are filtered by a supervisory system that usually includes the plant operators (e.g. checked for plausibility, mapped to ramp changes and clipped to avoid large changes [30]) and forwarded to the process control layer as setpoints. When viewed from the dynamic layer, these setpoints are often inconsistent and unreachable because of the discrepancies between the models used for steady-state optimization and dynamic regulation. Rao and Rawlings [36] discuss methods for resolving these inconsistencies and finding reachable steady-state targets that are as close as possible to the unreachable setpoints provided by the RTO.

Engell [11] reviews the developments in the field of feedback control for optimal plant operations, in which the various disadvantages of the two layer strategy are pointed out. The two main disadvantages of the current two layer approach are:

- Models in the optimization layer and in the control layer are not fully consistent [2, 41]. It is pointed out that, in particular, their steady-state gains may be different.
- The two layers have different time scales. The delay in optimization is inevitable because of the steady-state assumption [7].

Because of the disadvantages of long sampling times, several authors have proposed reducing the sampling time in the RTO layer [41]. In an attempt to narrow the gap between the sampling rates of the nonlinear steady-state optimization performed in the RTO layer and the linear MPC layer, the so called LP-MPC and QP-MPC two-stage MPC structures have been suggested [32, 48, 33, 5]. Jing and Joseph [18] perform a detailed analysis of this approach and analyze its properties. The task of the upper MPC layer is to compute the setpoints both for the controlled variables and for the manipulated inputs for the lower MPC layer by solving a constrained linear or quadratic optimization problem, using information from the RTO layer and from the MPC layer. The optimization is performed with the same sampling period as the lower-level MPC controller.

Forbes and Marlin [13], Zhang and Forbes [51] introduce a performance measure for RTO systems to compare the actual profit with theoretic profit. Three losses were considered as a part of the cost function: the loss in the transient period before the system reaches a steady state, the loss due to model errors, and the loss due to propagation of stochastic measurement errors. The issue of model fidelity is discussed by Yip and Marlin [47]. Yip and Marlin [46] proposed the inclusion of effect of setpoint changes on the accuracy of the parameter estimates into the RTO optimization. Duvall and Riggs [10] evaluate the performance of RTO for Tennessee Eastman Challenge Problem and point out "RTO profit should be compared to optimal, knowledgeable operator control of the process to determine the true benefits of RTO. Plant operators, through daily control of the process, understand how process setpoint selection affects the production rate and/or operating costs."

Backx et al. [2] describe the need for dynamic operations in the process industries in an increasingly market-driven economy where plant operations are embedded in flexible supply chains striving for just-in-time production in order to maintain

competitiveness. Minimizing operation cost while maintaining the desired product quality in such an environment is considerably harder than in an environment with infrequent changes and disturbances, and this minimization cannot be achieved by relying solely on experienced operators and plant managers using their accumulated knowledge about the performance of the plant. Profitable agile operations call for a new look at the integration of process control with process operations.

Kadam et al. [20] point out that the RTO techniques are limited with respect to the achievable flexibility and economic benefit, especially when considering intentionally dynamic processes such as continuous processes with grade transitions and batch processes. They also describe dynamics as the core of plant operation, motivating economically profitable dynamic operation of processes. Helbig et al. [16] introduce the concept of a dynamic real time optimization (D-RTO) strategy. Instead of doing a steady state economic optimization to compute setpoints, a dynamic optimization over a fixed horizon is done to compute a reference trajectory. To avoid dynamic re-optimization, the regulator tracks the reference trajectory using a simpler linear model (or PID controller) with the standard tracking cost function, hence enabling the regulator to act at a faster sampling rate. When using a simplified linear model for tracking a dynamic reference trajectory, an inconsistency remains between the model used in the two layers. Often an additional disturbance model would be required in the linear dynamic model to resolve this inconsistency. These disturbance states would have to be estimated from the output measurements. Kadam and Marquardt [19] review the D-RTO strategy and improvements to it, and discuss the practical considerations behind splitting the dynamic real-time optimization into two parts. A trigger strategy is also introduced, in which D-RTO reoptimization is only invoked if predicted benefits are significant, otherwise linear updates to the reference trajectory are provided using parametric sensitivity techniques.

Skogestad [43] describes one approach to implement optimal plant operation by a conventional feedback control structure, termed "self-optimizing" control. In this approach, the feedback control structure is chosen so that maintaining some function of the measured variables constant automatically maintains the process near an economically optimal steady state in the presence of disturbances. The problem is posed from the plantwide perspective, since the economics are determined by overall plant behavior. Aske et al. [1] also point out the lack of capability in steady-state RTO, in the cases when there are frequent changes in active constraints of large economic importance. The important special case is addressed in which prices and market conditions are such that economic optimal operation of the plant is the same as maximizing plant throughput. A coordinator model predictive control strategy is proposed in which a coordinator controller regulates local capacities in all the units.

Sakizlis et al. [39] describe an approach of integrating optimal process design with process control. They discuss integration of process design, process control and process operability together, and hence deal with the economics of the process. The incorporation of advanced optimizing controllers in simultaneous process and control design is the goal of the optimization strategy. It deals with an offline control approach where an explicit optimizing control law is derived for the process. The approach is said to yield a better economic performance.

Zanin et al. [50, 49], Rotava and Zanin [38] report the formulation, solution and industrial implementation of a combined MPC/optimizing control scheme for a fluidized bed catalytic cracker. The economic criterion is the amount of liquefied petroleum gas produced. The optimization problem that is solved in each controller sampling period is formulated in a mixed manner: range control MPC with a fixed linear plant model (imposing soft constraints on the controlled variables by a quadratic penalty term that only becomes active when the constraints are violated) plus a quadratic control move penalty plus an economic objective that depends on the values of the manipulated inputs at the end of the control horizon. Extremum seeking control is another approach in which the controller drives the system states to steady values that optimize a chosen performance objective. Krstić and Wang [21] address closed-loop stability of the general extremum seeking approach when the performance objective is directly measured online. Guay and Zhang [14] address the case in which the performance objective is not measurable and available for feedback. This approach has been evaluated for temperature control in chemical reactors subject to state constraints [15, 8].

Huesman et al. [17] point out that doing economic optimization in the dynamic sense leaves some degrees of freedom of the system unused. With the help of examples, it is shown that economic optimization problems can result in multiple solutions suggesting unused degrees of freedom. It is proposed to utilize these additional degrees of freedom for further optimization based on non economic objectives to get a unique solution.

3 Turnpike Theorems

We find it useful in the sequel to address problems in which the (positive) stage cost cannot be brought to zero by any admissible control sequence regardless of the length of time horizon considered. This problem can arise in several ways. One way is to choose squared distance from setpoint as the stage cost, and consider an unreachable setpoint. The controller objective function that sums the stage cost over a horizon then becomes unbounded as the horizon goes to infinity for any admissible control sequence.

Optimal control problems with unbounded cost are not new to control theory. The first studies of this class of problems arose in the economics literature in the 1920s [35] in which the problem of interest was to determine optimal savings rates to maximize capital accumulation. Since this problem has no natural final time, it was considered on the infinite horizon. A flurry of activity in the late 1950s and 1960s led to generalizations regarding future uncertainty, scarce resources, expanding populations, multiple products and technologies, and many other economic considerations. Much of this work focused on establishing existence results for optimal control with infinite horizon and unbounded cost, and the famous "turnpike" theorems [9] that characterize the optimal trajectory. McKenzie [28, 29] provides a comprehensive and readable overview of this research. In addition to their high degree of

technical relevance, turnpike theorems have a fascinating history. Consider the following nested quotation taken from McKenzie

> A turnpike theorem was first proposed, at least in a way that came to wide attention, by Dorfman, Samuelson, and Solow in their famous Chapter 12 of *Linear Programming and Economic Analysis* [9], entitled "Efficient Programs of Capital Accumulation." . . . I would like to quote the critical passage:
>
> > It is, in a sense, the single most effective way for the system to grow, so that if we are planning long-run growth, no matter where we start, and where we desire to end up, it will pay in the intermediate stages to get into a growth phase of this kind. It is exactly like a turnpike paralleled by a network of minor roads. There is a fastest route between any two points; and if the origin and destination are close together and far from the turnpike, the best route may not touch the turnpike. But if the origin and destination are far enough apart, it will always pay to get on the turnpike and cover distance at the best rate of travel, even if this means adding a little mileage at either end.
> > —Dorfman, Samuelson, and Solow [9, p.331]
>
> It is due to this reference, I believe, that theorems on asymptotic properties of efficient, or optimal, paths of capital accumulation came to be known as "turnpike theorems."
> —McKenzie [29]

It seems only fitting that *economists* developed some of the early optimal control ideas we now intend to further develop for optimizing *economic* performance of chemical plants.

This class of problems was transferred to and further generalized in the control literature. For infinite horizon optimal control of continuous time systems, Brock and Haurie [4] established the existence of overtaking optimal trajectories. Convergence of these trajectories to an optimal steady state is also demonstrated. Leizarowitz [23] extended the results of [4] to infinite horizon control of discrete time systems. Reduction of the unbounded cost, infinite horizon optimal control problem to an equivalent optimization problem with finite costs is established. Carlson et al. [6] provide a comprehensive overview of these infinite horizon results.

Turnpike example

The optimal steady state plays the role of the turnpike in the control problems we are addressing. Consider a linear system with the stage cost measuring distance from setpoint on both input and output

$$\begin{aligned} x^+ &= Ax + Bu \\ L(x,u) &= (1/2)\left(|Cx - y_{\mathrm{sp}}|_Q^2 + |u - u_{\mathrm{sp}}|_R^2\right) \quad Q > 0, R > 0 \end{aligned}$$

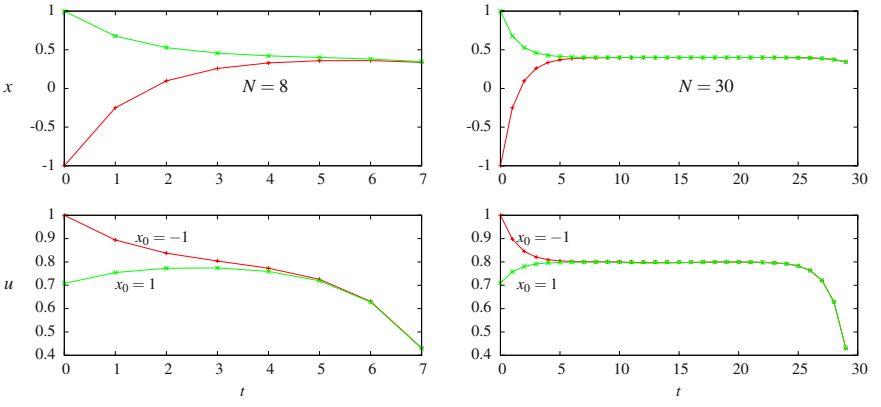


Fig. 1 An inconsistent setpoint pair $(x_{sp}, u_{sp}) = (2, 0)$ and the turnpike optimal control. For large horizon N, the optimal control spends most of the time near the optimal steady state, $(x^*, u^*) = (0.4, 0.8)$

where x^+ is the state at next time step and $|.|$ denotes the vector norm. The simplest way to generate a turnpike result is to supply *inconsistent* setpoints for the input and output, i.e. $y_{sp} \neq Gu_{sp}$ with G the steady-state gain. With these setpoints, the stage cost cannot be brought to zero *and maintained there*, and an infinite horizon problem has unbounded cost. Consider the SISO system

$$A = 1/2 \qquad B = 1/4 \qquad C = 1 \qquad G = 1/2 \qquad Q = 1 \qquad R = 1$$
$$u_{sp} = 0 \qquad y_{sp} = 2$$

For an unconstrained problem without integrators, the optimal steady state is given by

$$u^* = (G'QG + R)^{-1}(G'Qy_{sp} + Ru_{sp}) \qquad x^* = (I - A)^{-1}Bu^*$$
$$u^* = 0.8 \qquad x^* = 0.4$$

If we solve the open-loop dynamic optimal control problem from $x_0 = \pm 1$ for two different horizon lengths, we obtain the results in Figure 1. We see in Figure 1 that the optimal solution given an unreachable setpoint and a long horizon is to (i) move from the initial state near to the turnpike (optimal steady state), (ii) remain in the vicinity of this steady state for most of the control horizon, and then (iii) make a small transient move away from the turnpike to obtain a further small benefit to the cost function. Roughly speaking, the reason such trajectories are optimal is that for sufficiently long horizons, one cannot beat the cost obtained by hanging out at the optimal steady state.

4 Linear Models

We now consider the general case of MPC with a strictly convex stage cost and linear model. We do not assume the stage cost can be brought to zero nor that the infinite horizon cost is bounded for some admissible input sequence. The system model is assumed linear and time invariant

$$x^+ = Ax + Bu \tag{1}$$

$x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$. The cost function $L(x,u)$ is strictly convex and nonnegative and vanishes only at the setpoint, $L(x_{\mathrm{sp}}, u_{\mathrm{sp}}) = 0$. In this paper we employ the popular choice of a quadratic measure of distance from the setpoint

$$L(x,u) = (1/2)\left(|x - x_{\mathrm{sp}}|_Q^2 + |u - u_{\mathrm{sp}}|_R^2 + |u(j+1) - u(j)|_S^2\right) \quad Q > 0, R, S \geq 0$$

In the above definition, atleast one of $R, S > 0$. This system can be put in the standard LQR form by augmenting the state as $\tilde{x}(k) = [x(k) \quad u(k-1)]$ [36]. We denote the MPC controller using this stage cost as sp-MPC. We assume the input constraints define a nonempty polytope in $\mathbb{R}^m$

$$\mathbb{U} = \{u | Hu \leq h\}$$

Definition 1 (Optimal steady state). *The optimal steady state, denoted* (x^*, u^*)*, is the solution to the following optimization problem*

$$\min_{x,u} L(x,u) \qquad \textit{subject to } x^+ = Ax + Bu \quad u \in \mathbb{U}$$

For comparison purposes, we also consider standard MPC in which the optimal steady state is chosen as the center of the cost function

$$L_{\mathrm{targ}}(x,u) = (1/2)\left(|x - x^*|_Q^2 + |u - u^*|_R^2\right) \quad Q > 0, R > 0$$

We denote the MPC controller using this stage cost as targ-MPC.

4.1 Terminal Constraint MPC

We consider first the case in which the terminal constraint $x(N) = x^*$ is added to the controller. Define the controller cost function

$$V(x, \{u(i)\}_{i=0}^{N-1}) = \sum_{k=0}^{N-1} L(x(k), u(k)) \qquad x^+ = Ax + Bu, \quad x(0) = x_0$$

For notational simplicity, we define $\mathbf{u} = \{u(i)\}_{i=0}^{N-1}$. Hence, we express the MPC control problem as

$$\min_{\mathbf{u}} V(x, \mathbf{u}) \qquad \text{subject to: } x(N) = x^* \quad \mathbf{u} \in \mathbb{U} \tag{2}$$

We denote the optimal input sequence by

$$\mathbf{u}^0(x) = \{u^0(0,x), u^0(1,x), \ldots, u^0(N-1,x)\}$$

The MPC feedback law is the first move of this optimal sequence, which we denote as $u^0(x) = u^0(0,x)$, and we denote the optimal cost by $V^0(x)$. The closed-loop system is given by

$$x^+ = Ax + Bu^0(x)$$

with $u(k) = u^0(x(k))$ so

$$x^+ = f(x) \qquad f(\cdot) = A(\cdot) + Bu^0(\cdot)$$

Definition 2 (Steerable set). *The steerable set $\mathbb{X}_N$ is the set of states steerable to x^* in N steps*

$$\mathbb{X}_N = \{x | x^* = A^N x + A^{N-1}Bu(0) + \cdots + Bu(N-1), \quad u(j) \in \mathbb{U}, j = 0, \ldots, N-1\}$$

The usual comparison of the optimal cost at two subsequent stages gives the following inequality [37]

$$V^0(x^+) \leq V^0(x) - L(x, u^0(x)) + L(x^*, u^*) \tag{3}$$

In standard MPC with a reachable steady-state target, the origin can be shifted to (x^*, u^*) and the term $L(x^*, u^*)$ does not appear. That leads immediately to a cost decrease from x to x^+ and the optimal cost is a Lyapunov function for the closed-loop system. With the unreachable setpoint, the situation is completely different. The term $-L(x(k), u^0(x(k))) + L(x^*, u^*)$ changes sign with k on typical closed-loop trajectories. The cost decrease is lost and $V^0(x)$ is not a Lyapunov function for the closed-loop system. Losing the decreasing cost property does not mean asymptotic stability is lost, however. The following stability result has recently been established for this case [37].

Theorem 1 (Asymptotic Stability of Terminal Constraint MPC). *The optimal steady state (x^*, u^*) is the asymptotically stable solution of the closed-loop system under terminal constraint MPC. Its region of attraction is the steerable set.*

Unreachable setpoint example

Consider the single input-single output system

$$g(s) = \frac{-0.2623}{60s^2 + 59.2s + 1} \tag{4}$$

sampled with $T = 10$ sec. The input u is constrained as $|u| \leq 1$. The desired output setpoint is $y_{\text{sp}} = 0.25$ which corresponds to a steady-state input value of -0.953.

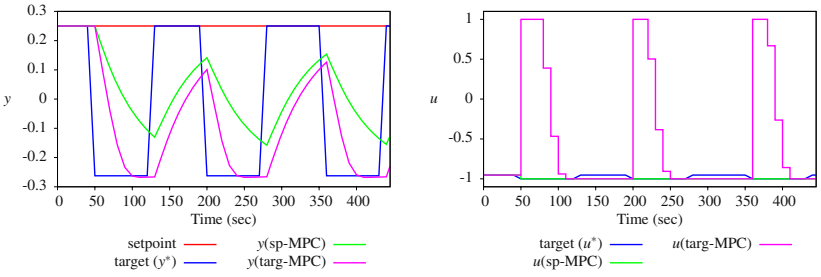


Fig. 2 Closed-loop performance of sp-MPC and targ-MPC

The regulator parameters are $Q_y = 10, R = 0, S = 1, Q = C'Q_yC + 0.01I_2$. A horizon length of $N = 80$ is used. Between times $50-130, 200-270$ and $360-430$, a state disturbance $d_x = [17.1, 1.77]'$ causes the input to saturate at its lower limit. The output setpoint is unreachable under the influence of this state disturbance (d_x). The closed-loop performance of sp-MPC and targ-MPC under the described disturbance scenario are shown in Figure 2. The closed-loop performance of the two control formulations are compared in Table 1. The following three closed-loop control performance measures are use to assess controller performance

$$V_u(k) = \frac{1}{kT}(1/2)\sum_{j=0}^{k-1}\left[\left|u(j)-u_{\mathrm{sp}}\right|_R^2 + \left|u(j+1)-u(j)\right|_S^2\right]$$

$$V_y(k) = \frac{1}{kT}(1/2)\sum_{j=0}^{k-1}\left|x(j)-x_{\mathrm{sp}}\right|_Q^2$$

$$V(k) = V_u(k) + V_y(k)$$

Comments

In the targ-MPC approach, the controller tries to reject the state disturbance and minimize the deviation from the new steady-state target. This requires a large, undesirable control action that forces the input to move between the upper and lower limits of operation. The sp-MPC approach, on the other hand, attempts to minimize the deviation from setpoint and subsequently the input just rides the lower limit input constraint. The benefit here is that the sp-MPC controller slows down the departure from setpoint, but speeds up the approach to setpoint. The traditional targ-MPC can be tuned to be fast or slow through relative choice of tuning parameters Q and R, but it is fast or slow from all initial conditions, some of which lead to an approach setpoint, but others of which lead to a departure from setpoint.

The greater cost of control action in targ-MPC is shown by the cost index V_u in Table 1. The cost of control action in targ-MPC exceeds that of sp-MPC by nearly 100%. The control in targ-MPC causes the output of the system to move away from the (unreachable) setpoint faster than the corresponding output of sp-MPC.

Table 1 Comparison of controller performance

Performance Measure	targ-MPC	sp-MPC	Δ(index)%
V_u	0.016	2.2×10^{-6}	99.98
V_y	3.65	1.71	53
V	3.67	1.71	54

Since the control objective is to be close to the setpoint, this undesirable behavior is eliminated by sp-MPC.

4.2 Terminal Penalty MPC

It is well known in the MPC literature that terminal constraint MPC with a short horizon is not competitive with terminal penalty MPC in terms of the size of the set of admissible initial states, and the undesirable difference between open-loop prediction and closed-loop behavior [27]. Here we briefly outline how the previous result can be used to prove asymptotic stability of terminal penalty MPC. First define the *rotated cost* L_r [23]

$$L_r(x,u) = L(x,u) - L(x^*,u^*)$$

and compute the infinite horizon rotated cost-to-go under control law $u_c(x) = K(x-x^*)+u^*$ with K chosen so that $\overline{A} = A+BK$ is asymptotically stable. A simple calculation gives

$$\begin{aligned} L_r^\infty(x) &= \sum_{k=0}^{\infty} L_r(x(k),u(k)) \qquad x^+ = Ax+Bu, \quad x(0)=x_0 \\ &= (1/2)(x-x^*)'\Pi(x-x^*) - \pi'(x-x^*) \end{aligned}$$

in which Π satisfies the usual Lyapunov equation and π is given by

$$\Pi = \overline{A}'\Pi\overline{A} + Q + K'RK \qquad \pi = (I-\overline{A}')^{-1}\left(Q(x_{\mathrm{sp}}-x^*) + K'R(u_{\mathrm{sp}}-u^*)\right)$$

Note that the rotated cost-to-go satisfies $L_r^\infty(x) = L_r^\infty(Ax+Bu_c(x)) + L_r(x,u_c(x))$. Next define the terminal penalty MPC cost function as

$$V_r(x,\mathbf{u}) = \sum_{k=0}^{N-1} L_r(x(k),u(k)) + L_r^\infty(x(N)) \qquad x^+ = Ax+Bu, \quad x(0)=x$$

and controller

$$\min_{\mathbf{u}} V_r(x,\mathbf{u}) \qquad \text{subject to: } \mathbf{u}\in\mathbb{U} \tag{5}$$

If the system is unstable and feasible K does not exist to make $(A+BK)$ stable given the active constraints at u^*, constraints to zero the unstable modes at stage N are added to Equation 5 [34]. The set of admissible initial states is chosen to ensure positive invariance and feasibility of control law u_c. These plus compactness of $\mathbb{U}$ ensure system evolution in a compact set. The user has some flexibility in choosing K, such as the simple choice $K=0$. A procedure to choose feasible K that more closely approximates the infinite horizon solution is given in Rao and Rawlings [36].

Using the rotated cost function, one can then establish the following inequality for terminal penalty MPC

$$V_r^0(x^+) \le V_r^0(x) - L(x,u^0(x)) + L(x^*,u^*)$$

which plays the same role as Equation 3 in the terminal constraint case, and asymptotic stability of terminal penalty MPC can also be established [37].

4.3 Economic Cost Function

We now look at an example in which the economic profit function is a linear function of state and input

$$L_{\text{eco}}(x,u) = \alpha' x + \beta' u$$

Notice this stage cost also is not bounded for an unconstrained system. We consider a linear process with two states and wish to see how optimizing the profit function in the MPC control problem is different than the regular the tracking implementation.

$$x^+ = \begin{bmatrix} 0.857 & 0.884 \\ -0.0147 & -0.0151 \end{bmatrix} x + \begin{bmatrix} 8.57 \\ 0.884 \end{bmatrix} u$$

The following values of α and β are chosen for the economic cost.

$$\alpha = -\begin{bmatrix} 3 & 2 \end{bmatrix}' \quad \beta = -2$$

The input is constrained between -1 and 1, and so, the economics being linear, the feasible optimum lies on the boundary of the constraint at $u^*=1$, which corresponds to $x^*=(60,0)$. For the tracking objective, the standard target cost function is chosen

$$L_{\text{targ}} = (1/2)\left(|x-x^*|_Q^2 + |u-u^*|_R^2\right) \qquad Q = 2I_2 \qquad R = 2$$

Figure 3 shows the state space system evolution when the system was initialized at the state (80,10). It is observed that the descent directions of tracking contours and the economic contours oppose each other. Hence the dynamic regulator in the standard two-layer approach does not address the economic incentive. The targ-MPC controller moves quickly to the target in a direction opposite to the best economics, while the eco-MPC controller maximizes the profit and only slowly moves toward

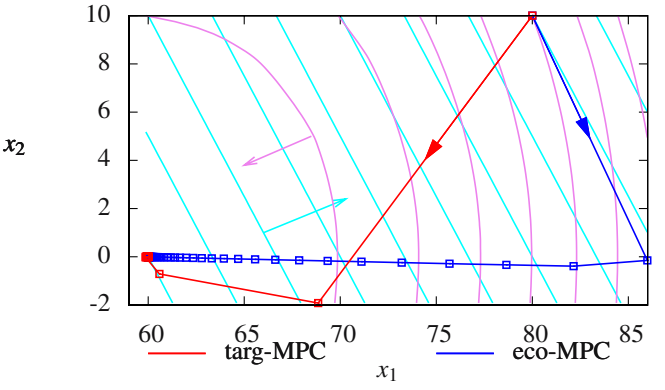


Fig. 3 The direction of increasing profit is opposite to the direction of decreasing setpoint error. The eco-MPC controller therefore takes a slow path to the steady-state setpoint while the standard targ-MPC controller takes a fast path

the steady target. The eco-MPC achieves a profit of 8034 units, which is a 6.3% improvement over the standard targ-MPC's profit of 7557.

5 Nonlinear Models

To fully support the goal of optimizing dynamic economic performance of a chemical process, we will need to find suitable ways to extend these results to *nonlinear* models. Linear models are remarkably useful in the existing two-level chemical plant control paradigm. The nonlinear model is reserved for a steady-state economic evaluation. The linear model is often perfectly adequate as the forecaster in the dynamic regulation problem because the controller's goal is to stay near this economically optimal steady state in the face of disturbances. If the controller is able to maintain the system near the steady state, the linear model is reasonably accurate. If we wish to optimize the dynamic economic performance, however, the nonlinear

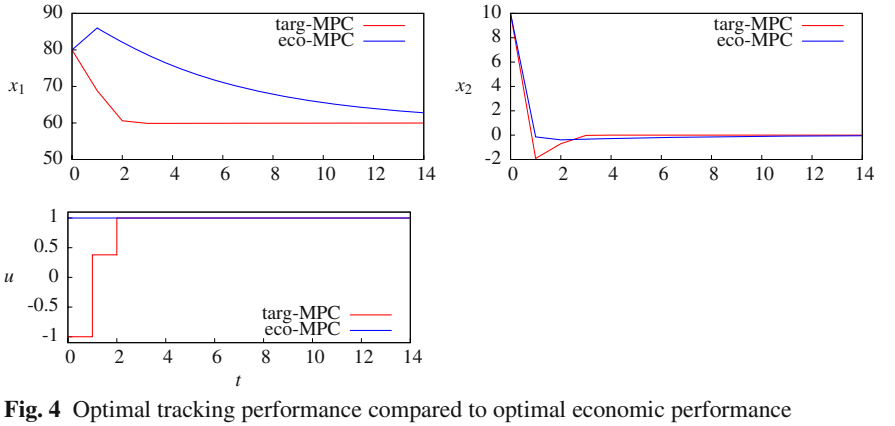


Fig. 4 Optimal tracking performance compared to optimal economic performance

model is required. This aspect of the problem presents some research and implementation challenges and may slow the introduction of these ideas into routine chemical plant practice. On the other hand, this class of problems motivates the development of improved methods for *nonlinear* MPC, which is after all the main goal of this conference. We next examine an illustrative nonlinear example.

5.1 Maximizing Production Rate in a CSTR

In many studies, it has been established that the performance of many continuous chemical processes can be improved by forced periodic operation [22, 42, 44, 45]. Bailey [3] provides a comprehensive review of periodic operation of chemical reactors.

Consider a single second-order, irreversible chemical reaction in an isothermal CSTR [42]

$$\mathrm{A} \longrightarrow \mathrm{B} \qquad r = kc_A^n$$

in which k is the rate constant and n is the reaction order. The material balance for component A is

$$\begin{aligned} \frac{dc_A}{dt} &= \frac{1}{\tau}(c_{Af} - c_A) - kc_A^n \\ \frac{dx}{dt} &= \frac{1}{\tau}(u - x) - kx^n \qquad \tau = 10, \quad k = 1.2, \quad n = 2 \end{aligned} \tag{6}$$

in which $c_A = x$ is the molar A concentration, $c_{Af} = u$ is the feed A concentration, and $\tau = 10$ is the reactor residence time. Consider the simple case in which the process economics for the reactor are completely determined by the average production rate of B. The reactor processes a mean feedrate of component A. The available manipulated variable is the instantaneous feed concentration. The constraints are that the feed rate is nonnegative, and the mean feed rate must be equal to the amount of A to be processed

$$u(t) \geq 0 \qquad \frac{1}{T}\int_0^T u(t)dt = 1 \tag{7}$$

in which T is the time interval considered. We wish to maximize the average production rate or minimize the negative production rate

$$V(x(0), u(t)) = -\frac{1}{T}\int_0^T kx^n(t)dt \qquad \text{subject to Equation 6} \tag{8}$$

The optimal control problem is then

$$\min_{u(t)} V(x(0, u(t)) \qquad \text{subject to Equation 6–7}$$

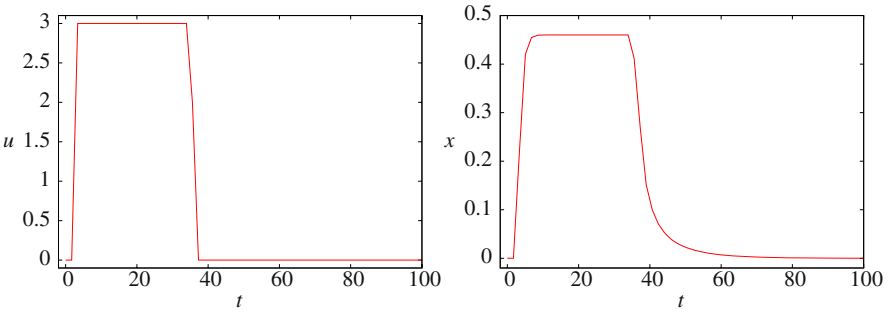


Fig. 5 Optimal periodic input and state. The achieved production rate is $V^* = -0.0835$, an 11% improvement over steady operation

The optimal steady operation is readily determined. In fact, the average flowrate constraint admits only a single steady feed rate, $u^* = 1$, which determines the optimal steady-state reactor A concentration and production rate

$$u^* = 1 \qquad x^* = 0.25 \qquad V^* = -0.075$$

For the second-order reaction, we can easily beat this production rate with a non-steady A feed policy. Consider the following extreme policy

$$u(t) = T\delta(t) \qquad 0 \le t \le T$$

which satisfies the mean feed rate constraint, and let $x(0) = 0$ be the reactor initial condition at $t = 0^-$. The impulsive feed policy gives a jump in x at $t = 0$ so $x(0^+) = T/\tau$. Solving the reactor material balance for this initial condition and $u = 0$ over the remaining time interval gives

$$x(t) = \frac{T/\tau}{(1 + 12T/\tau)e^{0.1t} - 12T/\tau} \qquad 0 \le t \le T$$

We see from this solution that by choosing T large, $x(T)$ is indeed close to zero, and we are approaching a periodic solution with a large period. Substituting $x(t)$ into Equation 8 and performing the integral in the limit of large T gives

$$V^* = \lim_{T\to\infty} -\frac{1}{T}\int_0^T kx^2(t)dt = -0.1$$

The sequence of impulses has increased the average production rate by 33% compared to steady operation. Of course, we cannot implement this extreme policy, but we can understand why the production rate is higher. The impulse increases the reactor A concentration sharply. For *second-order* kinetics, that increase pays off in the production of B, and we obtain a large instantaneous production rate which leads to a large average production rate.

For an implementable policy we can add upper bounding constraints on u and constrain the period

$$0 \le u(t) \le 3 \qquad \frac{1}{T}\int_0^T u(t)dt = 1 \qquad 0 \le T \le 100$$

Solving the optimal control problem subject to this constraint and periodic boundary conditions on $x(t)$ gives the results in Figure 5. With the new constraints, switching the input between the bounds (bang bang control), yields a time average production rate of 0.0835, which is an 11% improvement over the steady-state value of 0.075. Again, the character of the optimal solution is similar to the extreme policy: increase the reactor A concentration to the highest achievable level by maximizing the feed concentration for as long as possible while meeting the mean constraint.

Finally it is instructive to compare these results to the case of a first-order reaction, $n = 1$. For $n = 1$, the optimal steady state is

$$u^* = 1 \qquad x^* = \frac{1}{1+k\tau}u^* = 0.0769 \qquad V^* = -\frac{k}{1+k\tau}u^* = -0.09238$$

This operation is optimal on the infinite horizon because the model is linear and the cost is convex. Imagine we inject the extreme policy that was so effective in raising the production rate for the second-order kinetics. We obtain

$$x(t) = \frac{T}{\tau}e^{-(k+1/\tau)t} \qquad V = -\frac{k}{1+k\tau}$$

and we see that we obtain the *same* production rate as the steady operation. If we add a small regularization term so the stage cost is strictly convex, the steady operation is the unique optimal policy. The objective of maximizing production rate is a reasonable goal, and second-order kinetics are not an unusual nonlinearity. We see that some care must be taken in defining the control problem because simply changing the order of the reaction from $n = 1$ to $n = 2$ significantly changes the character of the optimal policy.

Carlson et al. [6, pp.51–52] provide one set of sufficient conditions for nonlinear $f(x,u)$ and $L(x,u)$ so that the optimal open-loop trajectory satisfies the turnpike property and converges to the optimal steady state. More research on this issue is certainly warranted.

6 Conclusions and Future Work

The broad goal of this paper was to explore how to take advantage of the powerful online optimization capabilities provided by model predictive control to address other more advanced process objectives beyond the traditional tracking and disturbance rejection objective. These more advanced objectives include methods for handling unreachable setpoints, and optimizing economic performance. In the case of linear models with convex objectives, it is well known that the optimal control

problem has the turnpike property. Moreover, under feedback control using MPC, it has recently been established that the optimal steady state is asymptotically stable for a reasonably broad class of control horizon and performance parameter values. The region of attraction is an appropriately defined steerable set.

Consideration of nonlinear models is an active area of research and invites continued research attention. The turnpike property was shown not to hold for simple and relevant chemical process examples with reasonable process objectives. We require better methods for analyzing and understanding the following question: for which models and objectives is "steady operation" asymptotically "optimal operation." When optimal operation is nonsteady we have to make choices. How large is the economic benefit for nonsteady operation? Do we wish to change the process objective so that steady operation is optimal? If so, how do we best make that choice? What are our available options for objectives? Or do we wish to take advantage of nonsteady operation? In that case, what classes of nonsteady operation are deemed acceptable: periodic, quasi-periodic, others? All of these issues are on the table for future research.

Obviously, the online solution of these nonlinear models requires reliable and efficient numerical optimization methods. Even fundamental research and exploration cannot be accomplished efficiently without these numerical tools. Hopefully the community of researchers will continue their efforts in the development and dissemination of high-quality optimization methods. Despite the many recent advances in this area, researchers still invest large amounts of their creative time setting up an optimal control problem, debugging software, and analyzing the causes of optimizer failures. Because the excellent closed-loop properties of the controller do not depend critically on strict optimality, however, the more modest computational requirements of suboptimal MPC increase the likelihood that these methods can be applied on even challenging industrial applications [40].

Summarizing the state of affairs: nonlinear MPC enables dynamic process optimization. As a research community, we have not yet fully explored how much process improvement this capability provides. Opportunities abound to address exciting new research challenges that may produce significant and long term industrial impact.

Acknowledgements. The authors would like to thank David Q. Mayne and David Angeli for fruitful discussion of these ideas. The first author would like to thank Wolfgang Marquardt for hosting his visit at RWTH Aachen where many of these ideas were developed. Don Bartusiak and ExxonMobil are acknowledged for financial support of this visit. The authors acknowledge financial support from the industrial members of the Texas-Wisconsin-California Control Consortium (TWCCC), and the NSF through grant #CTS–0825306.

References

[1] Aske, E.M.B., Strand, S., Skogestad, S.: Coordinator MPC for maximizing plant throughput. Comput. Chem. Eng. 32, 195–204 (2008)

[2] Backx, T., Bosgra, O., Marquardt, W.: Integration of model predictive control and optimization of processes. In: Advanced Control of Chemical Processes (June 2000)

[3] Bailey, J.E.: Periodic Operation of Chemical Reactors: A Review. Chem. Eng. Commun. 1(3), 111–124 (1974)
[4] Brock, W.A., Haurie, A.: On existence of overtaking optimal trajectories over an infinite time horizon. Math. Oper. Res. 1, 337–346 (1976)
[5] Brosilow, C., Zhao, G.Q.: A linear programming approach to constrained multivariable process control. Contr. Dyn. Syst. 27, 141 (1988)
[6] Carlson, D.A., Haurie, A.B., Leizarowitz, A.: Infinite Horizon Optimal Control, 2nd edn. Springer, Heidelberg (1991)
[7] Cutler, C.R., Perry, R.T.: Real time optimization with multivariable control is required to maximize profits. Comput. Chem. Eng. 7, 663–667 (1983)
[8] DeHaan, D., Guay, M.: Extremum seeking control of nonlinear systems with parametric uncertainties and state constraints. In: Proceedings of the 2004 American Control Conference, pp. 596–601 (July 2004)
[9] Dorfman, R., Samuelson, P., Solow, R.: Linear Programming and Economic Analysis. McGraw-Hill, New York (1958)
[10] Duvall, P.M., Riggs, J.B.: On-line optimization of the Tennessee Eastman challenge problem. J. Proc. Cont. 10, 19–33 (2000)
[11] Engell, S.: Feedback control for optimal process operation. J. Proc. Cont. 17, 203–219 (2007)
[12] Findeisen, W., Bailey, F., Bryds, M., Malinowski, K., Tatjewski, P., Wozniak, A.: Control and Coordination in Hierarchical Systems. John Wiley & Sons, New York (1980)
[13] Forbes, J.F., Marlin, T.E.: Design cost: a systematic approach to technology selection for model-based real-time optimization systems. Comput. Chem. Eng. 20, 717–734 (1996)
[14] Guay, M., Zhang, T.: Adaptive extremum seeking control of nonlinear dynamic systems with parametric uncertainty. Automatica 39, 1283–1293 (2003)
[15] Guay, M., Dochain, D., Perrier, M.: Adaptive extremum seeking control of nonisothermal continuous stirred tank reactors with temperature constraints. In: Proceedings of the 42nd IEEE Conference on Decision and Control, Maui, Hawaii (December 2003)
[16] Helbig, A., Abel, O., Marquardt, W.: Structural Concepts for Optimization Based Control of Transient Processes. Nonlinear Model Predictive Control (2000)
[17] Huesman, A.E.M., Bosgra, O.H., Van den Hof, P.M.J.: Degrees of Freedom Analysis of Economic Dynamic Optimal Plantwide Operation. In: Preprints of 8th IFAC International Symposium on Dynamics and Control of Process Systems (DYCOPS), vol. 1, pp. 165–170 (2007)
[18] Jing, C.M., Joseph, B.: Performance and stability analysis of LP-MPC and QP-MPC cascade control systems. AIChE J. 45, 1521–1534 (1999)
[19] Kadam, J., Marquardt, W.: Integration of Economical Optimization and Control for Intentionally Transient Process Operation. LNCIS, vol. 358, pp. 419–434 (2007)
[20] Kadam, J.V., Marquardt, W., Schlegel, M., Backx, T., Bosgra, O.H., Brouwer, P.J., Dünnebier, G., van Hessem, D., Tiagounov, A., de Wolf, S.: Towards integrated dynamic real-time optimization and control of industrial processes. In: Proceedings Foundations of Computer Aided Process Operations (FOCAPO 2003), pp. 593–596 (2003)
[21] Krstić, M., Wang, H.-H.: Stability of extremum seeking feedback for general nonlinear dynamic systems. Automatica 36, 595–601 (2000)
[22] Lee, C., Bailey, J.: Modification of Consecutive-Competitive Reaction Selectivity by Periodic Operation. Ind. Eng. Chem. Proc. Des. Dev. 19(1), 160–166 (1980)
[23] Leizarowitz, A.: Infinite horizon autonomous systems with unbounded cost. Appl. Math. Opt. 13, 19–43 (1985)

[24] Luyben, W.L., Tyreus, B., Luyben, M.L.: Plantwide Process Control. McGraw-Hill, New York (1999)
[25] Marlin, T.E.: Process Control. McGraw-Hill, New York (1995)
[26] Marlin, T.E., Hrymak, A.N.: Real-time operations optimization of continuous processes. In: Kantor, J.C., García, C.E., Carnahan, B. (eds.) Chemical Process Control–V, pp. 156–164. CACHE, AIChE (1997)
[27] Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: Stability and optimality. Automatica 36(6), 789–814 (2000)
[28] McKenzie, L.W.: Accumulation programs of maximum utility and the von Neumann facet. In: Wolfe, J.N. (ed.) Value, Capital, and Growth, pp. 353–383. Edinburgh University Press/ Aldine Publishing Company (1968)
[29] McKenzie, L.W.: Turnpike theory. Econometrica 44(5), 841–865 (1976)
[30] Miletic, I., Marlin, T.E.: Results analysis for real-time optimization (RTO): deciding when to change the plant operation. Comput. Chem. Eng. 20, 1077 (1996)
[31] Morari, M., Arkun, Y., Stephanopoulos, G.: Studies in the synthesis of control structures for chemical processes. Part I: Formulation of the problem. Process decomposition and the classification of the control tasks. Analysis of the optimizing control structures. AIChE J. 26(2), 220–232 (1980)
[32] Morshedi, A.M., Cutler, C.R., Skrovanek, T.A.: Optimal solution of dynamic matrix control with linear programming techniques (LDMC). In: Proceedings of the 1985 American Control Conference, pp. 199–208 (June 1985)
[33] Muske, K.R.: Steady-state target optimization in linear model predictive control. In: Proceedings of the American Control Conference, Albuquerque, NM, pp. 3597–3601 (June 1997)
[34] Muske, K.R., Rawlings, J.B.: Linear model predictive control of unstable processes. J. Proc. Cont. 3(2), 85–96 (1993)
[35] Ramsey, F.P.: A mathematical theory of saving. Econ. J. 38(152), 543–559 (1928)
[36] Rao, C.V., Rawlings, J.B.: Steady states and constraints in model predictive control. AIChE J. 45(6), 1266–1278 (1999)
[37] Rawlings, J.B., Bonné, D., Jørgensen, J.B., Venkat, A.N., Jørgensen, S.B.: Unreachable setpoints in model predictive control. IEEE Trans. Auto. Cont. 53(9), 2209–2215 (2008)
[38] Rotava, O., Zanin, A.: Multivariable control and real-time optimization — an industrial practical view. Hydrocarbon Processing, pp. 61–71 (June 2005)
[39] Sakizlis, V., Perkins, J.D., Pistikopoulos, E.N.: Recent advances in optimization-based simultaneous process and control design. Comput. Chem. Eng. 28, 2069–2086 (2004)
[40] Scokaert, P.O.M., Mayne, D.Q., Rawlings, J.B.: Suboptimal model predictive control (feasibility implies stability). IEEE Trans. Auto. Cont. 44(3), 648–654 (1999)
[41] Sequeira, E., Graells, M., Puigjaner, L.: Real-time evolution of online optimization of continuous processes. Ind. Eng. Chem. Res. 41, 1815–1825 (2002)
[42] Sincic, D., Bailey, J.: Analytical optimization and sensitivity analysis of forced periodic chemical processes. Chem. Eng. Sci. 35, 1153–1161 (1980)
[43] Skogestad, S.: Plantwide control: the search for the self-optimizing control structure. J. Proc. Cont. 10, 487–507 (2000)
[44] Watanabe, N., Onogi, K., Matsubara, M.: Periodic control of continuous stirred tank reactors-II: The Pi criterion and its applications to isothermal cases. Chem. Eng. Sci. 36, 809–818 (1981)
[45] Watanabe, N., Matsubara, M., Kurimoto, H., Onogi, K.: Periodic control of continuous stirred tank reactors-I: Cases of a nonisothermal single reactor. Chem. Eng. Sci. 37, 745–752 (1982)

[46] Yip, W.S., Marlin, T.E.: Designing plant experiments for real-time optimization systems. Control Eng. Prac. 11, 837–845 (2003)
[47] Yip, W.S., Marlin, T.E.: The effect of model fidelity on real-time optimization performance. Comput. Chem. Eng. 28, 267–280 (2004)
[48] Yousfi, C., Tournier, R.: Steady-state optimization inside model predictive control. In: Proceedings of American Control Conference, p. 1866 (1991)
[49] Zanin, A.C., Tvrzská de Gouvêa, M., Odloak, D.: Industrial implementation of a real-time optimization strategy for maximizing production of LPG in a FCC unit. Comput. Chem. Eng. 24, 525–531 (2000)
[50] Zanin, A.C., Tvrzská de Gouvêa, M., Odloak, D.: Integrating real-time optimization into the model predictive controller of the FCC system. Control Eng. Practice 10, 819–831 (2002)
[51] Zhang, Y., Forbes, J.F.: Extended design cost: a performance criterion for real-time optimization. Comput. Chem. Eng. 24, 1829–1841 (2000)

Hierarchical Model Predictive Control of Wiener Models

Bruno Picasso, Carlo Romani, and Riccardo Scattolini

Abstract. A hierarchical two-layer control structure is designed with robust Model Predictive Control. The system at the upper level is described by a Wiener model, while the systems at the lower level, which represent the fast actuators dynamics, are described by general nonlinear models. The proposed control structure guarantees steady-state zero error regulation for constant reference signals and allows one to largely decouple the control design phase at the two levels while still guaranteeing the overall stability under mild assumptions.

Keywords: Wiener systems, hierarchical control, nonlinear MPC, robustness, stability, asymptotic tracking.

1 Introduction

Due to the ever increasing complexity and to the higher and higher levels of automation of industrial plants and of production and distribution networks, after many years, see e.g. [5, 11, 12], research in large scale systems is going to have a renaissance. Therefore, there is a growing interest in the development of distributed and multi-level control structures. In particular, this paper deals with the design of two layers hierarchical control systems with Model Predictive Control (MPC), see figure 1. The system to be controlled is described by the discrete-time, multi-input, multi-output Wiener model

$$\mathscr{S}: \begin{cases} x^f(h+1) = A^f x^f(h) + B^f u^f(h) = \\ \qquad\qquad = A^f x^f(h) + \sum_{i=1}^{m} b_i^f u_i^f(h) \\ y^f(h) = c(x^f(h)), \end{cases} \tag{1}$$

Bruno Picasso, Carlo Romani, and Riccardo Scattolini
Politecnico di Milano, Dipartimento di Elettronica e Informazione,
via Ponzio 34/5, 20133 Milano, Italy
e-mail: picasso-romani-scattolini@elet.polimi.it

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 139–152.
springerlink.com

where b_i^f is the i-th column of B^f. At the upper level of the considered hierarchical structure, an MPC regulator is designed at a slow time scale to guarantee robust steady-state zero error regulation for constant reference signals by including a suitable integral action in the control law. The control variables u_i computed by this regulator are the desired outputs of the systems at the lower level, described by the nonlinear models

$$\mathscr{S}_{\text{act}} : \begin{cases} \zeta_i(h+1) = s_i(\zeta_i(h), \mu_i(h)) \\ \tilde{u}_i(h) = m_i(\zeta_i(h)), \end{cases} \tag{2}$$

which represent the available actuators operating at a faster time scale. In turn, also these actuators are controlled with the MPC approach. The use of MPC at both levels allows one to cope with control and/or state constraints.

The discrepancy between the control actions (u_i) computed by the regulator at the upper level and the ones ($\tilde{u}_i$) provided by the systems at the lower level can be tolerated in view of the robustness properties guaranteed by the min-max MPC law adopted at the upper level, see [6]. Specifically, the term $w^f = \tilde{u} - u$ is viewed as a state-dependent disturbance to be rejected. This approach differs from the classical sequential design where the lower level control loop is considered as part of the model seen by the upper level controller. With the proposed technique, instead, it is possible to largely decouple the design phase and the related optimization problems at the two levels of the hierarchical structure, while still guaranteeing stability properties for the overall control system under suitable assumptions on the actuators' dynamics.

For linear systems, this approach has already been considered in [9, 10], where it has been shown how: (i) the robustness of the control law at the upper level can be improved to achieve stability also when the actuators undergo dynamic limitations, (ii) different configurations can be considered at the lower level when redundancy in the actuators allows for some degrees of freedom.

Notation. In order to cope with a multirate implementation typical of hierarchical control structures, where the upper layer acts at a slower rate than the lower layer, two time scales are considered: the fast discrete-time index is denoted by h, while the slow discrete-time index is represented by k. Then, given a signal $\phi^f(h)$ in the fast time scale, its sampling in the slow time scale is $\phi(k) = \phi^f(\nu k)$, where ν is a fixed positive integer.

By $\|\cdot\|$ we denote the Euclidean vector or matrix norm.

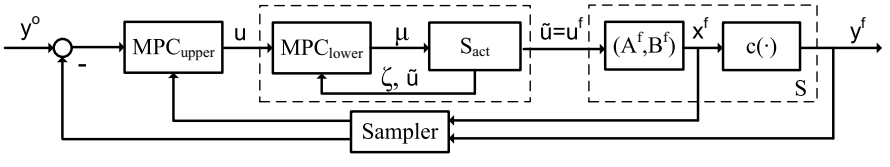


Fig. 1 The hierarchical control structure considered in the paper

2 System and Control at the Upper Level

2.1 System and Control Problem

In the fast discrete time index, the system at the upper level is described by equation (1). It is supposed that $c(\cdot)$ is a C^2 function with $c(0)=0$, the measurable state $x^f \in X_p \subset \mathbb{R}^{n_x}$ and the control variables $u_i^f \in U_{pi} \subset \mathbb{R}$, $i=1,\ldots,m$, belong to given compact sets, $y^f \in \mathbb{R}^m$ is the output.

For system (1), the tracking problem considered here consists of finding a feedback control law such that $\lim_{h\to+\infty} y^f(h) = y^o$, where $y^o \in Y^o \subseteq c(X_p)$ is a given reference signal.

Assumption 1

i) *the pair (A^f, B^f) is stabilizable;*
ii) *the matrix A^f has no eigenvalues λ with $|\lambda| = 1$;*
iii) *for a given reference signal $y^o \in Y^o$:* (a) *there exists at least one equilibrium pair $\bar{x}^f(y^o)$, $\bar{u}^f(y^o)$ (for short $\bar{x}^f$ and $\bar{u}^f$ in the following) strictly inside X_p and U_p respectively, such that $c(\bar{x}^f) = y^o$;* (b) *letting $C^f(y^o) = \frac{\partial c(x^f)}{\partial x^f}|_{\bar{x}^f}$ (for short C^f), the pair (A^f, C^f) is detectable;* (c) *the linearized system (A^f, B^f, C^f) has no invariant zeros in* 1.

For a fixed integer $\nu \geq 1$, let us decompose the input of system (1) in the form $u^f(h) = \hat{u}(h) + (u^f(h) - \hat{u}(h))$, where $\hat{u}(h) \in U_p$ is some piecewise constant signal such that, $\forall k = 0, 1, \ldots$ and $\forall j = 0, \ldots, \nu - 1$, it holds that $\hat{u}(\nu k + j) = \hat{u}(\nu k)$. Then system (1) can be rewritten as

$$\begin{cases} x^f(h+1) = A^f x^f(h) + B^f \hat{u}(h) + B^f w^f(h) \\ y^f(h) = c(x^f(h)), \end{cases} \tag{3}$$

where $w^f(h) = u^f(h) - \hat{u}(h)$ is considered as a matched disturbance term. Letting $x(k) = x^f(\nu k)$, $u(k) = \hat{u}(\nu k)$, $y(k) = y^f(\nu k)$ and

$$A = (A^f)^\nu, \quad B = \sum_{j=0}^{\nu-1} (A^f)^{\nu-j-1} B^f, \quad w(k) = \sum_{j=0}^{\nu-1} (A^f)^{\nu-j-1} B^f w^f(\nu k + j) \tag{4}$$

system (3) can be written in the slow sampling rate as:

$$\mathscr{S}_s : \begin{cases} x(k+1) = Ax(k) + Bu(k) + w(k) \\ y(k) = c(x(k)) \end{cases} \tag{5}$$

Hence, at the upper level, the problem consists of solving a robust zero error regulation for system (5), thus yielding the piecewise constant signal $\hat{u}(h)$ in (3).

In view of (4), if $(\bar{x}^f, \bar{u}^f)$ is an equilibrium pair for system (1) such that $c(\bar{x}^f) = y^o$, then the pair $\bar{x} = \bar{x}^f$, $\bar{u} = \bar{u}^f$ is also an equilibrium for system (5), with $w = 0$, such that $c(\bar{x}) = y^o$. Moreover, it is easy to prove the following result:

Proposition 1. *Under Assumption 1, the linearization of system (5) at* $(\bar{x},\bar{u})$ *has no invariant zeros in* 1.

In order to guarantee robust asymptotic zero error regulation, the above stated tracking problem is solved by forcing an integral action on each component of the error $e(k) = y^o - y(k)$. Moreover, an additional delay function is applied on each control variable for an easy computation of the control increment $u(k) - u(k-1)$ in the design of the MPC law. Then, the overall system to be controlled is

$$\mathscr{S}_{\text{all}} : \begin{cases} x(k+1) = Ax(k) + Bu(k) + w(k) \\ v(k+1) = v(k) + y^o - c(x(k)) \\ \phi(k+1) = u(k) \\ e(k) = y^o - c(x(k)) \end{cases} \tag{6}$$

which can be written in the more compact form

$$\begin{cases} \chi(k+1) = \widehat{f}(\chi(k), y^o) + \hat{B}u(k) + \hat{B}_1 w(k) \\ e(k) = \widehat{\eta}(\chi(k), y^o) \end{cases} \tag{7}$$

where $\chi = [x' v' \phi']'$, while the matrices $\hat{B}$, $\hat{B}_1$ and the C^2 functions $\widehat{f}$, $\widehat{\eta}$ can be easily derived from (6). Associated to (7), let us define the auxiliary output

$$z(k) = \begin{bmatrix} e(k) \\ M\chi(k) \\ u(k) - \hat{C}\chi(k) \end{bmatrix} \tag{8}$$

where

$$\hat{C} = \begin{bmatrix} 0_{m,n_x} & 0_{m,m} & I_{m,m} \end{bmatrix} \quad \text{and} \quad \mathbb{R}^{(n_x+2m)\times(n_x+2m)} \ni M > 0.$$

Let $\overline{\chi} = \begin{bmatrix} \bar{x}' & \bar{v}' & \bar{u}' \end{bmatrix}'$ be an equilibrium of (7) with $w = 0$ such that $\widehat{\eta}(\overline{\chi}, y^o) = 0$, and $\bar{z} = [0 \; M\overline{\chi} \; 0]$. Rewrite (7), (8) as

$$\begin{cases} \delta\chi(k+1) = f(\delta\chi(k), y^o) + \hat{B}\delta u(k) + \hat{B}_1 w(k) \\ e(k) = \eta(\delta\chi(k), y^o) \end{cases} \tag{9}$$

$$\delta z(k) = \begin{bmatrix} \eta(\delta\chi(k), y^o) \\ M\delta\chi(k) \\ \delta u(k) - \hat{C}\delta\chi(k) \end{bmatrix}, \tag{10}$$

where $\delta\chi = \chi - \overline{\chi}$, $\delta u = u - \overline{u}$ and $\delta z = z - \bar{z}$. Moreover, let $X \subset \mathbb{R}^{n_x+2m}$ and $U \subset \mathbb{R}^m$ be the two largest sets such that, for any $\delta\chi \in X$ and $\delta u \in U$ one has $x \in X_p$ and $u_i \in U_{pi}$, $i = 1,\ldots,m$. It holds that $U \subset \mathbb{R}^m$ is a compact set and $X \subset \mathbb{R}^{n_x+2m}$ is closed (and bounded with respect to $x(k), \phi(k)$) both containing the origin in their interior.

Remark 1. Letting $F_1 = \partial f / \partial \delta\chi|_{\delta\chi=0}$, from Proposition 1 it can be easily checked that the pair $(F_1, \hat{B})$ is stabilizable.

Assumption 2. *For a given $\gamma_\Delta > 0$, the disturbance w in system (9), (10) is such that $\forall k \geq 0$, $w(k) \in W(\gamma_\Delta, \delta z(k))$, where*

$$W(\gamma_\Delta, \delta z) := \{w \in \mathbb{R}^{n_x} : \|w\| \leq \gamma_\Delta \|\delta z\|\}$$

2.2 Robust MPC for the Upper Level

Consider system (9) and let γ be a positive real: according to the results reported in [6], an auxiliary control law guaranteeing robust stability for any disturbance in $W(\gamma_\Delta, \delta z)$, with $\gamma\gamma_\Delta < 1$, is first derived. Then, the goal is to determine an MPC state-feedback control law such that the corresponding closed loop system with input w and output δz given by (10) has performance and a region of attraction greater than those provided by the auxiliary control law.

Auxiliary control law

For system (9), (10), let us synthesize a state-feedback controller which guarantees robust stability for disturbances satisfying Assumption 2. To this end, given a positive γ such that $\gamma\gamma_\Delta < 1$ and a square $(n_x + 2m) \times (n_x + 2m)$ matrix P, letting

$$R = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} = \begin{bmatrix} \hat{B}'P\hat{B} + I & \hat{B}'P\hat{B}_1 \\ \hat{B}_1'P\hat{B} & \hat{B}_1'P\hat{B}_1 - \gamma^2 I \end{bmatrix}$$

and defining the quadratic function

$$V_f(\delta\chi) = \delta\chi' P \delta\chi$$

the following result holds:

Proposition 2. *Suppose that there exists a positive definite matrix P such that:*

a) $r_{22} < 0$

b) *letting* $H = \begin{bmatrix} \partial\eta/\partial\delta\chi|_{\delta\chi=0} \\ M \end{bmatrix}$,

$$-P + F_1'PF_1 + H'H + \hat{C}'\hat{C} - [F_1'P\hat{B} - \hat{C}' \quad F_1'P\hat{B}_1]R^{-1}\begin{bmatrix} \hat{B}'PF_1 - \hat{C} \\ \hat{B}_1'PF_1 \end{bmatrix} < 0$$

Consider the control law

$$\delta u = \kappa_f(\delta\chi, y^o), \tag{11}$$

where

$$\begin{bmatrix} \kappa_f(\delta\chi, y^o) \\ \xi^*(\delta\chi, y^o) \end{bmatrix} = -R^{-1}\begin{bmatrix} \hat{B}'Pf(\delta\chi, y^o) - \hat{C}\delta\chi \\ \hat{B}_1'Pf(\delta\chi, y^o) \end{bmatrix}, \tag{12}$$

and the closed-loop system (9), (11). Then there exist $\alpha > 0$ and $\varepsilon > 0$ such that, letting $X_f := \{\delta\chi : V_f(\delta\chi) \leq \alpha\}$, it holds that: **i)** *$X_f \subseteq X$;* **ii)** *$\forall \delta\chi \in X_f$ one has $\kappa_f(\delta\chi, y^o) \in U$;* **iii)** *$\forall \delta\chi \in X_f$, $\delta\chi \neq 0$, and $\forall w \in W(\gamma_\Delta, \delta z)$ the following holds:*

$$V_f(f(\delta\chi,y^o)+\hat{B}\kappa_f(\delta\chi,y^o)+\hat{B}_1 w)-V_f(\delta\chi)<-(\|\delta z\|^2-\gamma^2\|w\|^2+\varepsilon\|\delta\chi\|^2) \tag{13}$$

In particular, under Assumption 2, X_f is a positively invariant set.

Proof. To simplify the notation, in the proof let $\delta\chi\longrightarrow\chi$, $\delta u\longrightarrow u$, $f(\delta\chi,y^o)\longrightarrow f(\chi)$, $\eta(\delta\chi,y^o)\longrightarrow\eta(\chi)$ and $\delta z\longrightarrow z$. Following [6], define:

$$\Psi(\chi,u,w)=V_f(f(\chi)+\hat{B}u+\hat{B}_1 w)-V_f(\chi)+(\|z\|^2-\gamma^2\|w\|^2)$$

Then:

$$\begin{aligned}\Psi(\chi,u,w)&=(f(\chi)+\hat{B}u+\hat{B}_1 w)'P(f(\chi)+\hat{B}u+\hat{B}_1 w)-\chi'P\chi+\\&+(\eta(\chi)'\eta(\chi)+\chi'M'M\chi+u'u+\chi'\hat{C}'\hat{C}\chi-2u'\hat{C}\chi-\gamma^2 w'w)=\\&=f'(\chi)Pf(\chi)-\chi'P\chi+\eta'(\chi)\eta(\chi)+\chi'M'M\chi+\chi'\hat{C}'\hat{C}\chi-\\&-2\begin{bmatrix}u' & w'\end{bmatrix}\begin{bmatrix}\hat{C}\chi\\0\end{bmatrix}+\begin{bmatrix}u' & w'\end{bmatrix}R\begin{bmatrix}u\\w\end{bmatrix}+2\begin{bmatrix}u' & w'\end{bmatrix}\begin{bmatrix}\hat{B}'Pf(\chi)\\\hat{B}_1'Pf(\chi)\end{bmatrix}\end{aligned} \tag{14}$$

and, computing $\Psi(\chi,u,w)$ for $u=\kappa_f(\chi)$ and $w=\xi^*(x)$ from (12),

$$\begin{aligned}\Psi(x,\kappa_f(\chi),\xi^*(\chi))&=f'(\chi)Pf(\chi)-\chi'P\chi+\eta'(\chi)\eta(\chi)+\chi'M'M\chi+\chi'\hat{C}'\hat{C}\chi-\\&-\begin{bmatrix}f'(\chi)P\hat{B}-\chi'\hat{C}' & f'(\chi)P\hat{B}_1\end{bmatrix}R^{-1}\begin{bmatrix}\hat{B}'Pf(\chi)-\hat{C}\chi\\\hat{B}_1'Pf(\chi)\end{bmatrix}\end{aligned}$$

From hypothesis **b)** it follows that there exist $\varepsilon>0$ and a neighborhood $\widetilde{X}_0$ of $\chi=0$ such that

$$\Psi(\chi,\kappa_f(\chi),\xi^*(\chi))<-\varepsilon\|\chi\|^2\qquad\forall\chi\in\widetilde{X}_0,\ \chi\neq 0 \tag{15}$$

By the Taylor expansion of (14):

$$\begin{aligned}\Psi(\chi,u,w)&=\Psi(\chi,\kappa_f(\chi),\xi^*(\chi))+\tfrac{1}{2}\begin{bmatrix}u'-\kappa_f'(\chi) & w'-\xi'^*(\chi)\end{bmatrix}R\begin{bmatrix}u-\kappa_f(\chi)\\w-\xi^*(\chi)\end{bmatrix}+\\&+o\left(\left\|\begin{matrix}u-\kappa_f(\chi)\\w-\xi^*(\chi)\end{matrix}\right\|^2\right)\end{aligned}$$

If the system is controlled by $u=\kappa_f(\chi)$ then:

$$\Psi(\chi,\kappa_f(\chi),w)=\Psi(\chi,\kappa_f(\chi),\xi^*(\chi))+\tfrac{1}{2}(w-\xi^*(\chi))'r_{22}(w-\xi^*(\chi))+o(\|w-\xi^*(\chi)\|^2)$$

Since $r_{22}<0$, there exist a neighborhood X_0 of $\chi=0$ and $W_0:=\{w\in\mathbb{R}^{n_x}:\|w\|\leq r_w\}$ (for some $r_w>0$) such that $\forall\chi\in X_0$ and $\forall w\in W_0$, one has $\Psi(\chi,\kappa_f(\chi),w)\leq\Psi(\chi,\kappa_f(\chi),\xi^*(\chi))$. In view of (15), $\forall\chi\in\widetilde{X}_0\cap X_0$, $\chi\neq 0$, and $\forall w\in W_0$, it holds that $\Psi(\chi,\kappa_f(\chi),w)\leq\Psi(\chi,\kappa_f(\chi),\xi^*(\chi))<-\varepsilon\|\chi\|^2$ that is,

$$V_f(f(\chi)+\hat{B}\kappa_f(\chi)+\hat{B}_1 w)<V_f(\chi)-(\|z\|^2-\gamma^2\|w\|^2)-\varepsilon\|\chi\|^2$$

It is then possible to choose a sufficiently small $\alpha>0$ so that

$$X_f := \{\chi : V_f(\chi) \leq \alpha\}$$

is such that $X_f \subseteq X$, $X_f \subseteq \widetilde{X}_0 \cap X_0$ and $\forall \chi \in X_f$, $\|z(\chi)\| \cdot \gamma_\Delta \leq r_w$, where $z(\chi) = \begin{bmatrix} \eta(\chi, y^o)' & \chi' M' & \kappa_f(\chi)' - \chi' \hat{C}' \end{bmatrix}'$. Hence, from Assumption 2, $\forall \chi \in X_f$ it holds that $w \in W(\gamma_\Delta, z(\chi)) \subset W_0$. Therefore, $\forall \chi \in X_f$ with $\chi \neq 0$ and $\forall w \in W(\gamma_\Delta, z(\chi))$,

$$V_f(f(\chi) + \hat{B}\kappa_f(\chi) + \hat{B}_1 w) < V_f(\chi) - (\|z\|^2 - \gamma^2 \|w\|^2) - \varepsilon \|\chi\|^2 < V_f(\chi) - \varepsilon \|\chi\|^2$$

which proves both (13) and the positive invariance of X_f for the closed-loop system (9), (11). Finally, the control constraint $\kappa_f(\chi, y^o) \in U$ $\forall \chi \in X_f$ follows by the choice of $X_f \subseteq X$ and by the invariance of X_f. ■

Model predictive control

The region of attraction X_f and the performance provided by the auxiliary control law (11) are now improved with the MPC approach. To this end, consider system (9), (10), let $N_p \in \mathbb{N}$ be the length of the prediction horizon and $N_c \in \mathbb{N}$, $N_c \leq N_p$, be the length of the control horizon. Define by

$$D(k, N_p) = [w(k), w(k+1), \ldots, w(k+N_p-1)]$$

the sequence of disturbances over the prediction horizon and by

$$K(\delta\chi(k), y^o, N_c) = [\kappa_o(\delta\chi(k), y^o), \kappa_1(\cdot, y^o), \ldots, \kappa_{N_c-1}(\cdot, y^o)],$$

where $\kappa_i : \mathbb{R}^{n_x+2m} \to \mathbb{R}^m$, the vector of state-feedback control laws to be synthesized by the MPC control algorithm. At any time instant k, the control problem consists of solving the following min-max optimization problem:

$$\min_{K(\delta\chi(k), y^o, N_c)} \max_{D(k, N_p)} J(\delta\chi(k), K, D, N_c, N_p) \tag{16}$$

$$J(\delta\chi(k), K, D, N_c, N_p) = \sum_{i=k}^{k+N_p-1} (\|\delta z(i)\|^2 - \gamma^2 \|w(i)\|^2) + V_f(\delta\chi(k+N_p))$$

subject to:

i) system (9), (10) and the control law

$$\delta u(k+j) = \begin{cases} \kappa_j(\delta\chi(k+j), y^o), & j = 0, \ldots, N_c - 1 \\ \kappa_f(\delta\chi(k+j), y^o), & j = N_c, \ldots, N_p - 1; \end{cases}$$

ii) the disturbance constraint $w(k+j) \in W(\gamma_\Delta, \delta z(k+j))$, $j = 0, \ldots, N_p - 1$;

iii) the state and control constraints $\delta\chi(k+j) \in X$ and $\delta u(k+j) \in U$, $\forall j = 0, \ldots, N_p - 1$ and $\forall D(k, N_p)$ such that ii) holds;

iv) the terminal constraint $\delta\chi(k+N_p) \in X_f$, $\forall D(k, N_p)$ such that ii) holds.

If $(\bar{K}(\delta\chi(k),y^o,N_c),\bar{D}(k,N_p))$ is the optimal solution of this min-max problem, according to the Receding Horizon principle, define the MPC control law as

$$\delta u(k) = \bar{\kappa}_o(\delta\chi(k),y^o) = \kappa_{RH}(\delta\chi(k),y^o) \tag{17}$$

Under the previous assumptions, the following theorem can be proved:

Theorem 1. *Let $X^{MPC}(N_c,N_p) \subseteq X$ be the set of states $\delta\chi$ such that the min-max problem (16) admits a solution. Then $\forall N_p \in \mathbb{N}$ and $\forall N_c \in \mathbb{N}$, $N_c \leq N_p$:*

i) $X^{MPC}(N_c,N_p)$ is a positively invariant set for the closed-loop system (9), (17);
ii) $X_f \subseteq X^{MPC}(N_c,N_p)$;
iii) The state of the closed-loop system (9), (17) asymptotically converges to the origin with region of attraction $X^{MPC}(N_c,N_p)$, that is the controlled system (5) converges to its equilibrium $(\bar{x},\bar{u})$.

Proof. To simplify the notation, in the proof let $\delta\chi \longrightarrow \chi$ and $\delta z \longrightarrow z$. The theorem is proved for $N_c \geq 1$ (the case $N_c = 0$ directly follows by Proposition 2).

Let $V(\chi(k),N_c,N_p) = J(\chi(k),\bar{K},\bar{D},N_c,N_p)$ be the optimal performance at time k. By the disturbance constraint ii) (i.e., Assumption 2) and the condition $\gamma\gamma_\Delta < 1$, the function $V(\chi(k),N_c,N_p)$ is positive definite.

Let us prove that $X^{MPC}(N_c,N_p)$ is a positively invariant set for the closed loop system (9), (17). Indeed, if $\chi(k) \in X^{MPC}$, then there exists $\bar{K}$ such that $\chi(k+N_p) \in X_f$. Hence, at time $k+1$, consider the following policy:

$$\hat{K}(\chi(k+1),y^o,N_c) = \\ = [\bar{\kappa}_1(\chi(k+1),y^o),\ldots,\bar{\kappa}_{N_c-1}(\chi(k+N_c-1),y^o),\kappa_f(\chi(k+N_c),y^o)]$$

Since, under Assumption 2, X_f is a positively invariant set with respect to the auxiliary law, the new policy is feasible and X^{MPC} is a positively invariant set. Moreover, being the auxiliary law feasible, it follows that $X_f \subseteq X^{MPC}$ (in particular, the equilibrium is included in X^{MPC}).

To prove iii), let us show that $V(\chi(k),N_c,N_p)$ is decreasing along the state trajectories. Indeed, consider the control and prediction horizons of length N_c+1, N_p+1 and the control policy

$$\widetilde{K}(\chi(k),y^o,N_c+1) = [\bar{K}(\chi(k),y^o,N_c) \quad \kappa_f(\chi(k+N_c),y^o)]$$

Therefore,

$$J(\chi(k),\widetilde{K}(\chi(k),y^o,N_c+1),D(k,N_p+1),N_c+1,N_p+1) = \\ = \textstyle\sum_{i=k}^{k+N_p-1}(\|z(i)\|^2 - \gamma^2\|w(i)\|^2) + V_f(\chi(k+N_p+1)) + \\ + V_f(\chi(k+N_p)) - V_f(\chi(k+N_p)) + (\|z(k+N_p)\|^2 - \gamma^2\|w(k+N_p)\|^2)$$

Since the value of the output $z(k+N_p)$ is obtained with the auxiliary control law used at time $k+N_p$, using inequality (13), one has

$$J(\chi(k),\widetilde{K}(\chi(k),y^o,N_c+1),D(k,N_p+1),N_c+1,N_p+1)<$$
$$<\sum_{i=k}^{k+N_p-1}(\|z(i)\|^2-\gamma^2\|w(i)\|^2)+V_f(\chi(k+N_p))$$

Consequently

$$V(\chi(k),N_c+1,N_p+1)\leq$$
$$\leq\max_{D(k,N_p+1)}J(\chi(k),\widetilde{K}(\chi(k),y^o,N_c+1),D(k,N_p+1),N_c+1,N_p+1)< \tag{18}$$
$$<V(\chi(k),N_c,N_p)$$

Now observe that

$$V(\chi(k),N_c,N_p)\geq\|z(k)\|^2-\gamma^2\|w(k)\|^2+V(\chi(k+1),N_c-1,N_p-1)$$

and, thanks to (18),

$$V(\chi(k+1),N_c,N_p)-V(\chi(k),N_c,N_p)<-(\|z(k)\|^2-\gamma^2\|w(k)\|^2)$$

Finally, by Assumption 2, it follows that

$$V(\chi(k+1),N_c,N_p)-V(\chi(k),N_c,N_p)<-(1-\gamma^2\gamma_\Delta^2)\|z(k)\|^2 \tag{19}$$

which proves that $V(\chi(k),N_c,N_p)$ is a decreasing function since $\gamma\gamma_\Delta<1$.

From (19), it follows that $\lim_{k\to+\infty}z(k)=0$ and, since $M>0$ (see equation (8)), $\lim_{k\to+\infty}\chi(k)=0$. Therefore, by returning to the original coordinates, $\lim_{k\to+\infty}x(k)=\bar{x}$, $\lim_{k\to+\infty}u(k)=\bar{u}$ and $\lim_{k\to+\infty}v(k)=\bar{v}$. ■

3 System and Control at the Lower Level

3.1 System

The real inputs u_i^f of system (1) coincide with the outputs $\tilde{u}_i$, $i=1,\ldots,m$, of the i-th single-input, single-output system at the lower level described in the short sampling time by the nonlinear model in equation (2). There, $\zeta_i\in\mathbb{R}^{n_{\zeta_i}}$ is the measurable state and $\mu_i\in\mathbb{R}$ is the manipulated input. Moreover, the following state and input constraints are considered:

$$\zeta_i\in Z_{pi},\quad \mu_i\in M_{pi},$$

where Z_{pi}, M_{pi} are compact sets with nonempty interior and Z_{pi} is such that $m_i(Z_{pi})=U_{pi}$. For systems (2), we consider the following assumption:

Assumption 3. *For any $\hat{u}_i\in U_{pi}$ there exists an unique equilibrium pair ($\hat{\zeta}_i\in Z_{pi}$, $\hat{\mu}_i\in M_{pi}$) such that:*

$$\begin{cases}\hat{\zeta}_i=s_i(\hat{\zeta}_i,\hat{\mu}_i)\\ \hat{u}_i=m_i(\hat{\zeta}_i)\end{cases}$$

For a given reference $\hat{u}_i \in U_{pi}$, according to Assumption 3, system (2) can be rewritten as

$$\begin{cases} \delta\zeta_i(h+1) = \widetilde{s}_i(\delta\zeta_i(h), \delta\mu_i(h), \hat{\zeta}_i, \hat{\mu}_i) \\ \delta u_i(h) = \widetilde{m}_i(\delta\zeta_i(h), \hat{\zeta}_i) - \hat{u}_i \end{cases} \tag{20}$$

where $\delta\zeta_i(h) = \zeta_i(h) - \hat{\zeta}_i$, $\delta\mu_i(h) = \mu_i(h) - \hat{\mu}_i$ and $\delta u_i(h) = \widetilde{u}_i(h) - \hat{u}_i$. Define by $M_i(\hat{\mu}_i) \subseteq \mathbb{R}$ and $Z_i(\hat{\zeta}_i) \subseteq \mathbb{R}^{n_{\zeta_i}}$ the largest compact sets such that $\mu_i \in M_{pi}$ for any $\delta\mu_i \in M_i(\hat{\mu}_i)$ and $\zeta_i \in Z_{pi}$ for any $\delta\zeta_i \in Z_i(\hat{\zeta}_i)$.

3.2 MPC for the Lower Level

For system (20) it is now possible to design a stabilizing MPC algorithm by means of one of the many techniques proposed in the literature, see e.g. [7, 1, 3, 4]. Additionally, with the same kind of developments reported in the previous section, one could include integral actions to achieve robust steady-state zero error regulation for constant references. However, for simplicity, in the following a *zero terminal constraint* algorithm without integral action will be used, see e.g. [2, 7]. Specifically, for the i-th subsystem, it is possible to minimize with respect to the sequence of future control variables

$$\Theta_i(h,\nu) = [\delta\mu_{i,h}(h), \ldots, \delta\mu_{i,h}(h+\nu-1)]$$

the performance index

$$J_i(\delta\zeta_i(h), \Theta_i, \nu) = \textstyle\sum_{j=0}^{\nu-1}(\|\delta u_i(h+j)\|^2 + \|\delta\zeta_i(h+j)\|^2_{Q_i}) \quad i = 1, \ldots, m,$$

where $\|\delta\zeta_i\|^2_{Q_i} = \zeta_i' Q_i \zeta_i$ with $Q_i > 0$, subject to (20), to the input and state constraints $\delta\mu_{i,h}(h+j) \in M_i$, $\delta\zeta_i(h+j) \in Z_i$, $j = 0, \ldots, \nu-1$, and to the terminal constraint $\delta\zeta_i(h+\nu) = 0$. Letting $\Theta_i^o(h,\nu) = [\delta\mu_{i,h}^o(h), \ldots, \delta\mu_{i,h}^o(h+\nu-1)]$ be the optimal future control sequence, only its first value $\delta\mu_{i,h}^o(h)$ is applied and the overall procedure is repeated at any short time instant. This implicity defines a state-feedback control law

$$\delta\mu_i(h) = \gamma_i(\delta\zeta_i(h), \hat{u}_i) \tag{21}$$

where, for later use, we find convenient to make explicit the dependence of the control law from the reference $\hat{u}_i$.

Assumption 4. *For any equilibrium pair $(\hat{\zeta}_i \in Z_{pi}, \hat{\mu}_i \in M_{pi})$, the set of states $\delta\zeta_i$ such that the above optimization problem admits a solution is $Z_i(\hat{\zeta}_i)$.*

Under this assumption, the following result holds true:

Theorem 2. *The state of the closed loop system (20), (21) asymptotically converges to the origin with region of attraction $Z_i(\hat{\zeta}_i)$ and $\lim_{h\to+\infty} \delta u_i(h) = 0$.*

The proof follows standard arguments, see [7].

4 The Hierarchical Control System

In a hierarchical implementation, at any long sampling time k, the controller at the upper level computes the value $u(k)$ of the input for system (9), the components of $u(k)$ are then taken as the references for the systems at the lower level (2). In details, consider the input signal $u(k)$, $k = 0, 1, \ldots$ generated by the MPC at the upper level and define

$$\hat{u}(h) = u\left(\lfloor \tfrac{h}{\nu} \rfloor\right), \tag{22}$$

where $\lfloor \cdot \rfloor$ is the floor function. Let the i-th component of $\hat{u}(h)$ be the reference signal for (2), the input for system (1) is then defined by the corresponding outputs of the systems (2), (21), that is $u^f(h) = \tilde{u}(h)$ where

$$\begin{cases} \zeta_i(h+1) = s_i(\zeta_i(h), \mu_i(h)) \\ \mu_i(h) = \gamma_i(\zeta_i(h) - \hat{\zeta}_i(h), \hat{u}_i(h)) + \hat{\mu}_i(h) \\ \tilde{u}_i(h) = m_i(\zeta_i(h)), \end{cases} \tag{23}$$

with $(\hat{\zeta}_i(h), \hat{\mu}_i(h))$ being the equilibrium pair associated to $\hat{u}_i(h)$.

Notice that system (23) is well-defined thanks to Assumption 4 which guarantees that the optimization problem defining the feedback law (21) remains feasible at any switching instant $h = \nu k$. Notice also that, although the references $\hat{u}_i(h)$ are piecewise constant functions, the values of $\mu_i(h)$ provided by (21) are computed considering constant references $\hat{u}_i(h)$ over the prediction horizon.

Because of the state and control constraints of the systems at the lower level, as well as to the initial conditions $\zeta_i(\nu k)$ of the states of the systems (2) at the beginning of any slow sampling period, the control variables $\tilde{u}_i(\nu k + j)$, $j = 0, \ldots, \nu - 1$, are in general different from the corresponding desired components $u_i(k)$. The difference between the input value provided by the controller at the upper level and the one provided by the subsystems at the lower level is the matched disturbance term appearing in (3), namely $w_i^f(h) = \tilde{u}_i(h) - \hat{u}_i(h)$. Hence, according to (4), the norm of the disturbance $w(k)$ acting on system (9) is such that

$$\|w(k)\| \leq \sum_{i=1}^{m} \sum_{j=0}^{\nu-1} \|(A^f)^{\nu-j-1} b_i^f w_i^f(\nu k + j)\| \tag{24}$$

In order to test the validity of Assumption 2, at any long time instant $h = \nu k$, associated with the optimal future control sequence $\Theta_i^o(\nu k, \nu)$, it is possible to compute the function

$$\tilde{J}_i^o(\delta\zeta_i(\nu k), \Theta_i^o(\nu k, \nu), \nu) = \textstyle\sum_{j=0}^{\nu-1} \|(A^f)^{\nu-j-1} b_i^f \delta u_i^o(\nu k + j)\|$$

(denoted, for short, by $\tilde{J}_i^o(\nu k)$) where $\delta u_i^o(\nu k + j)$, $j = 0, \ldots, \nu - 1$, is the output of (20) with the sequence $\Theta_i^o(\nu k, \nu)$. However, since system (20) is controlled with a receding horizon policy, the function $\tilde{J}_i^o(\nu k)$ is not directly related to an upper bound for $\|w(k)\|$. Nevertheless, if at any short sampling time $h = \nu k + l$, $l = 1, \ldots, \nu - 1$,

the optimization problem defining the feedback law (21) is solved under the additional constraint

$$\sum_{j=l}^{\nu-1} \|(A^f)^{\nu-j-1} b_i^f \delta u_{i,l}(\nu k+j)\| \le \sum_{j=l}^{\nu-1} \|(A^f)^{\nu-j-1} b_i^f \delta u_i^o(\nu k+j)\|,$$

where the $\delta u_{i,l}$ are the outputs of (20) with the sequence $\Theta_i(\nu k+l,\nu)$, then

$$\sum_{j=0}^{\nu-1} \|(A^f)^{\nu-j-1} b_i^f w_i^f(\nu k+j)\| \le \tilde{J}_i^o(\nu k) \tag{25}$$

so that, by (24),

$$\|w(k)\| \le \sum_{i=1}^{m} \tilde{J}_i^o(\nu k).$$

This modified optimization problem is always feasible in view of the feasibility at time $h = \nu k$. Thus, it is assumed that the feedback law (21), and hence system (23), are changed accordingly. In the end, the following result holds:

Theorem 3. *Under the assumptions of Theorems 1 and 2, consider the closed loop system (1), (22) and (23), where the MPC controller at the upper level is initialized with* $\delta\chi(0) = [x^{f\prime}(0) - \bar{x}' \;\; v'(0) - \bar{v}' \;\; \phi'(0) - \bar{u}']' \in X^{MPC}(N_c, N_p)$ *and, at the lower level,* $\zeta_i(0) \in Z_{pi}$, $i = 1,\dots,m$. *Assume that* $\forall k \in \mathbb{N}$, *the following condition is verified:*

$$\sum_{i=1}^{m} \tilde{J}_i^o(\nu k) \le \gamma_\Delta \|\delta z(k)\| \tag{26}$$

Assume also that, $\forall j = 0,\dots,\nu-1$, *the columns of the matrix* $(A^f)^j B^f$ *are all different from 0. Then it holds that* $\lim_{h\to+\infty} x^f(h) = \bar{x}$ *and, in particular,* $\lim_{h\to+\infty} y^f(h) = y^o$.

Proof. Inequality (26) guarantees that Assumption 2 is satisfied for system (9), (10). Hence, by Theorem 1, it holds that $\lim_{k\to+\infty} \delta z(k) = 0$ (for, $\lim_{h\to+\infty} \hat{u}(h) = \bar{u}$ and $\lim_{k\to+\infty} \delta x^f(\nu k) = 0$, where $\delta x^f = x^f - \bar{x}$).

We claim that $\lim_{h\to+\infty} \tilde{u}(h) = \bar{u}$: since $\lim_{k\to+\infty} \delta z(k) = 0$, then by (26) and (25), $\forall i = 1,\dots,m$, one has $\lim_{k\to+\infty} \sum_{j=0}^{\nu-1} \|(A^f)^{\nu-j-1} b_i^f w_i^f(\nu k+j)\| = 0$, namely

$$\lim_{k\to+\infty} \sum_{j=0}^{\nu-1} \|(A^f)^{\nu-j-1} b_i^f (\tilde{u}_i(\nu k+j) - \hat{u}_i(\nu k+j))\| = 0.$$

Thus, $\forall j = 0,\dots,\nu-1$, $\lim_{k\to+\infty} \|(A^f)^{\nu-j-1} b_i^f (\tilde{u}_i(\nu k+j) - \hat{u}_i(\nu k+j))\| = 0$ and, since $(A^f)^{\nu-j-1} b_i^f \neq 0$, $\lim_{k\to+\infty} (\tilde{u}_i(\nu k+j) - \hat{u}_i(\nu k+j)) = 0$ which is equivalent to $\lim_{h\to+\infty} (\tilde{u}_i(h) - \hat{u}_i(h)) = 0$. The claim is proved because $\lim_{h\to+\infty} \hat{u}_i(h) = \bar{u}_i$.

To complete the proof, let us show that, $\forall l = 1,\dots,\nu-1$, it holds that $\lim_{k\to+\infty} \|\delta x^f(\nu k+l)\| = 0$. To this end, since by (1)

$$\delta x^f(\nu k+l) = (A^f)^l \delta x^f(\nu k) + \sum_{j=0}^{l-1} (A^f)^{l-j-1} B^f (\tilde{u}(\nu k+j) - \bar{u}),$$

then, $\forall l = 1,\dots,\nu-1$, it holds that

$$\|\delta x^f(\nu k+l)\| \leq \|(A^f)^l\| \cdot \|\delta x^f(\nu k)\| + \sum_{j=0}^{l-1} \|(A^f)^{l-j-1} B^f\| \cdot \|\tilde{u}(\nu k+j) - \bar{u}\|.$$

The thesis follows because $\lim_{k\to+\infty} \delta x^f(\nu k) = 0$ and $\lim_{h\to+\infty} \tilde{u}(h) = \bar{u}$. ■

Remark 2. Theorem 2 ensures that, for any constant reference $\hat{u}_i$, the state ζ_i of the i-th actuator converges to the corresponding equilibrium $\hat{\zeta}_i$. However, as it is clarified in [8], it is not guaranteed in general that the following CICS *(converging-input converging-state)* property holds: if $\lim_{h\to+\infty} \hat{u}_i(h) = \bar{u}_i$, then $\lim_{h\to+\infty} \zeta_i(h) = \bar{\zeta}_i$. Hence, in Theorem 3, it is not possible to conclude that also the internal states of the actuators converge to the equilibrium $\bar{\zeta}$. Nevertheless, by definition of the controller (21) at the lower level, it is guaranteed that the state ζ_i of the i-th actuator belongs to the compact set Z_{pi}.

If instead, for the system at the lower level, one is able to prove stability rather than mere convergence, then also the convergence of the states of the actuators to $\bar{\zeta}_i$ is guaranteed for the overall system [8].

Remark 3. If condition (26) is not fulfilled, one may relax the constraint by modifying the robustness properties of the system at the upper level, i.e. by reducing the attenuation level γ, so that a larger value of γ_Δ is allowed. With reference to a pure regulation problem for linear systems, in [9] it has been shown that combining this strategy with a suitable switching procedure, the convergence property provided by Theorem 3 can be achieved under milder assumptions. If there is some redundancy in the available actuators at the lower level, also these further degrees of freedom can be exploited to satisfy at any time instant the robustness constraint. This approach is described in [10] for linear systems.

5 Conclusions

A robust MPC approach has been derived to solve a hierarchical control problem characterized by two layers. The upper one, working at a slow time scale, decides the ideal control inputs by solving an asymptotic zero error regulation problem for Wiener systems with constant reference signals and sends its requirements to the set of the available actuators at the lower level, which in turn compute the actual control action. The discrepancy between the ideal and actual control values justifies the use of the robust paradigm, whereas the presence of hard bounds on the control and state variables enforces the use of the MPC approach.

Acknowledgements. This research has received funding from European Community through FP7/2007-2013 under grant agreement n. 223854 (“Hierarchical and Distributed Model Predictive Control of large scale systems”, HD-MPC project).

References

1. Chen, H., Allgöwer, F.: A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. Automatica 34, 1205–1217 (1998)
2. Clarke, D.W., Scattolini, R.: Constrained Receding Horizon Predictive Control. In: 17IEE Proc. Part D, vol. 138, pp. 347–354 (1991)
3. De Nicolao, G., Magni, L., Scattolini, R.: Stabilizing Receding-Horizon Control of Nonlinear Time-Varying Systems. In: IEEE Trans. on Automatic Control, vol. AC-43, pp. 1030–1036 (1998)
4. De Nicolao, G., Magni, L., Magnani, L., Scattolini, R.: A stabilizing model-based predictive control for nonlinear systems. Automatica 37, 1351–1362 (2001)
5. Findeisen, W., Bailey, F.N., Brdyś, M., Malinowski, K., Tatjevski, P., Woźniak, A.: Control and Coordination in Hierarchical Systems. John Wiley & Sons, Chichester (1980)
6. Magni, L., De Nicolao, G., Scattolini, R., Allgöwer, F.: Robust Model Predictive Control of Nonlinear Discrete-Time Systems. International Journal of Robust and Nonlinear Control 13, 229–246 (2003)
7. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: Stability and Optimality. Automatica 36, 789–814 (2000)
8. Ryan, E.P., Sontag, E.D.: Well-defined steady-state response does not imply CICS. Systems & Control Letters 55, 707–710 (2006)
9. Scattolini, R., Colaneri, P.: Hierarchical Model Predictive Control. In: IEEE Control and Decision Conference, New Orleans (2007)
10. Scattolini, R., Colaneri, P., De Vito, D.: A Switched MPC Approach to Hierarchical Control. In: 17th IFAC World Congress, Seoul (2008)
11. Siljak, D.D.: Large-scale dynamic systems: stability and structure. North-Holland, New York (1978)
12. Siljak, D.D.: Decentralized control of complex systems. Academic Press, Boston (1991)

Multiple Model Predictive Control of Nonlinear Systems

Matthew Kuure-Kinsey and B. Wayne Bequette

Abstract. An augmented state space formulation for multiple model predictive control (MMPC) is developed to improve the regulation of nonlinear and uncertain process systems. By augmenting disturbances as states that are estimated using a Kalman filter, improved disturbance rejection is achieved compared to an additive output disturbance assumption. The approach is applied to a Van de Vusse reactor example, which has challenging dynamic behavior in the form of a right half plane zero and input multiplicity.

Keywords: model predictive control, state estimation, multiple model.

1 Introduction

With chemical processes moving towards specialty chemicals and batch processes in recent years, advanced control strategies are required that handle the resulting wide range of operating conditions from, for example, different product specifications for different customers. One such advanced control strategy is multiple model control. In its simplest form, known as gain scheduling, a bank of controllers is generated, with each representing a discrete region of the operating space. A decision variable, such as the value of the output variable, is used to trigger a switch between active controllers.

In gain scheduling, the switching point between controllers is often pre-determined and there is no flexibility in determining if that controller is still the best suited to the particular system conditions. This flexibility is provided by

Matthew Kuure-Kinsey
Rensselaer Polytechnic Institute, Troy, New York
e-mail: kuurem@rpi.edu

B. Wayne Bequette
Rensselaer Polytechnic Institute, Troy, New York
e-mail: bequette@rpi.edu

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 153–165.
springerlink.com

not pre-determining switching points, and instead choosing as the active controller the controller that minimizes a desired objective function. Athans *et al.* [1] use this approach to control F-8C fighter aircraft. The approach uses linear quadratic controllers, each of which is generated to nominally represent a different aspect of a typical flight profile. The objective function is an *a posteriori* calculation based on the residuals of each model output, with the control action a weighted linear combination based on the magnitude of the residuals. This results in blending of controller outputs, in contrast to switching strategies that select a single controller output.

Similar to the work of Athans *et al.* [1], the motivation for this research is the development of a unified control strategy capable of controlling a nonlinear system at different steady state operating conditions. There are two primary contributions in this paper. The first is the extension of multiple model predictive control to the use of augmented state space matrices, and the second addresses a lack of detail in the existing literature by providing a tutorial overview of model averaging, state estimation and state variable transitions.

2 Control Structure

The control structure for the multiple model predictive control strategy is shown in the control block diagram in figure 1.

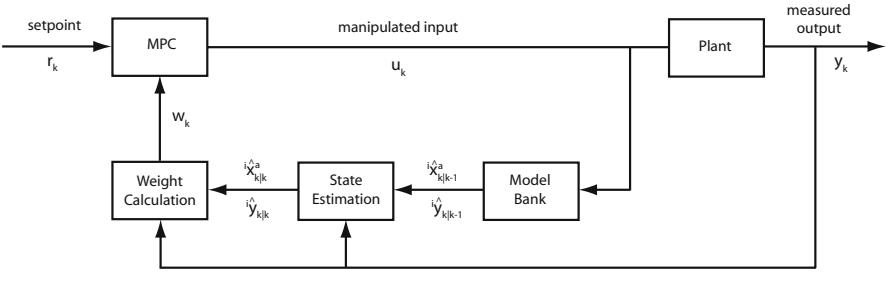


Fig. 1 Control block diagram of multiple model predictive control strategy

2.1 Model Bank

The multiple model predictive control strategy is based on the use of n models in the model bank that have the general form given in (1).

$$\begin{aligned} {}^i\hat{x}_k &= {}^i\boldsymbol{\Phi}\,{}^i\hat{x}_{k-1} + {}^i\Gamma\, u_{k-1} \\ {}^i\hat{y}_k &= {}^iC\,{}^i\hat{x}_k \end{aligned} \tag{1}$$

The left superscript i denotes the model number, with i ranging from 1 to n models. Although the plant being controlled in practice is likely nonlinear, the models in (1) are all linear. It is important to note that there are no inherent restrictions in this

method to linear models, and extension of this method to nonlinear models follows a similar approach. Each of the n linear models is selected to represent a specific expected operating condition of the nonlinear system. The use of local linear models simplifies control design while maintaining an accurate representation of each system operating condition. The models in the model bank are updated in parallel with the plant by passing the manipulated input, u_{k-1}, into the model bank and updating each model.

2.2 State Estimation

The linear state space model defined in (1) as the basis for the model bank uses "hat" notation to designate that the states and outputs are predicted rather than measured. Taking into account the presence of process and measurement noise, the state space model in (1) is rewritten in (2).

$$\begin{aligned} {}^{i}x_k &= {}^{i}\Phi\, {}^{i}x_{k-1} + {}^{i}\Gamma\, u_{k-1} \\ {}^{i}y_k &= {}^{i}C\, {}^{i}x_k + \nu_k \end{aligned} \tag{2}$$

The ν_k term in (2) represents measurement noise. None of the n linear models in the model bank perfectly describes the plant at all times. There is always a degree of parameter uncertainty, system characteristics change with time and disturbances enter and affect the system. To handle this, the model predicted outputs, ${}^{i}y_k$, are corrected by estimating a disturbance term. This disturbance term, ${}^{i}d_k$, is used to account for all uncertainties between the model and plant being controlled.

To estimate this disturbance term, each model in the model bank is augmented with a disturbance estimation model. The simplest and most frequently used model is the additive output disturbance model. The additive output disturbance enters the linear model through the output equation, resulting in the augmented model in (3).

$$\begin{aligned} {}^{i}x_k &= {}^{i}\Phi\, {}^{i}x_{k-1} + {}^{i}\Gamma\, u_{k-1} \\ {}^{i}d_k &= {}^{i}d_{k-1} + \omega_{k-1} \\ {}^{i}y_k &= {}^{i}C\, {}^{i}x_k + {}^{i}d_k + \nu_k \end{aligned} \tag{3}$$

The ω_{k-1} term is process noise associated with the disturbance estimation and the ν_k term is again measurement noise. An alternative correction method is the use of a step input disturbance model. The step input disturbance enters the linear model through the state equation, resulting in the augmented model in (4).

$$\begin{aligned} {}^{i}x_k &= {}^{i}\Phi\, {}^{i}x_{k-1} + {}^{i}\Gamma\, u_{k-1} + {}^{i}\Gamma^{d}\, {}^{i}d_{k-1} \\ {}^{i}d_k &= {}^{i}d_{k-1} + \omega_{k-1} \\ {}^{i}y_k &= {}^{i}C\, {}^{i}x_k + \nu_k \end{aligned} \tag{4}$$

The model states ${}^{i}x_k$ and disturbance term ${}^{i}d_k$ for both disturbance models are "states" being estimated by the model, allowing for combination into a single augmented state vector, defined in (5).

$$ {}^{i}x_k^a = \begin{bmatrix} {}^{i}x_k \\ {}^{i}d_k \end{bmatrix} \tag{5} $$

Using the augmented state vector, the two disturbance models are generalized into the common augmented model in (6).

$$ \begin{aligned} \underbrace{\begin{bmatrix} {}^{i}x_k \\ {}^{i}d_k \end{bmatrix}}_{{}^{i}x_k^a} &= \underbrace{\begin{bmatrix} {}^{i}\Phi & G_x \\ 0 & I \end{bmatrix}}_{{}^{i}\Phi^a} \underbrace{\begin{bmatrix} {}^{i}x_{k-1} \\ {}^{i}d_{k-1} \end{bmatrix}}_{{}^{i}x_{k-1}^a} + \underbrace{\begin{bmatrix} {}^{i}\Gamma \\ 0 \end{bmatrix}}_{{}^{i}\Gamma^a} u_{k-1} + \underbrace{\begin{bmatrix} 0 \\ I \end{bmatrix}}_{{}^{i}\Omega^a} \omega_{k-1} \\ {}^{i}y_k &= \underbrace{\begin{bmatrix} {}^{i}C & G_y \end{bmatrix}}_{{}^{i}C^a} \underbrace{\begin{bmatrix} {}^{i}x_k \\ {}^{i}d_k \end{bmatrix}}_{{}^{i}x_k^a} + \nu_k \end{aligned} \tag{6} $$

The two general terms in (6) are G_x and G_y, defined as 0 and I for the additive output disturbance model and ${}^{i}\Gamma^d$ and 0 for the step input disturbance model. The augmented model in (6) is further simplified using the appropriate augmented state matrices, resulting in:

$$ \begin{aligned} {}^{i}x_k^a &= {}^{i}\Phi^a \, {}^{i}x_{k-1}^a + {}^{i}\Gamma^a u_{k-1} + {}^{i}\Omega^a \omega_{k-1} \\ {}^{i}y_k &= {}^{i}C^a \, {}^{i}x_k^a + \nu_k \end{aligned} \tag{7} $$

The augmented state and output terms in (7) are calculated based on information available at the previous timestep — that is, before a plant measurement is taken. This is clear when (7) is rewritten to include timestep notation.

$$ \begin{aligned} {}^{i}\hat{x}_{k|k-1}^a &= {}^{i}\Phi^a \, {}^{i}\hat{x}_{k-1|k-1}^a + {}^{i}\Gamma^a u_{k-1} \\ {}^{i}\hat{y}_{k|k-1} &= {}^{i}C^a \, {}^{i}\hat{x}_{k|k-1}^a \end{aligned} \tag{8} $$

The augmented states and outputs in (8) are predicted using information from the previous timestep. Because they are predictions, and as the process and measurement noise terms in (7) are not known in practice, the augmented states and outputs use "hat" notation to denote a predicted value.

To make the predictions from (8) as accurate as possible, information from the plant measurement at the current timestep is used to update the model. The plant measurement is incorporated using a set of predictor / corrector equations, given in (9) – (11).

$$ {}^{i}\hat{x}_{k|k-1}^a = {}^{i}\Phi^a \, {}^{i}\hat{x}_{k-1|k-1}^a + {}^{i}\Gamma^a u_{k-1} \tag{9} $$

$$ {}^{i}\hat{x}_{k|k}^a = {}^{i}\hat{x}_{k|k-1}^a + {}^{i}L_k (y_k - {}^{i}C^a \, {}^{i}\hat{x}_{k|k-1}^a) \tag{10} $$

$$ {}^{i}\hat{y}_{k|k} = {}^{i}C^a \, {}^{i}\hat{x}_{k|k}^a \tag{11} $$

The predictor / corrector equations first predict the augmented states without the measurement, then update the augmented states based on the difference between the

plant measurement and the uncorrected model prediction. In the limit of a perfect model, this difference is zero and there is no state update in (10). The augmented state update is dependent on L_k, which is an appropriate observer gain based on the disturbance model. For the additive output disturbance model, L_k is defined in (12).

$$L_k = \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{12}$$

This is equivalent to the use of a deadbeat observer [2]. For the step input disturbance model, L_k is defined by the solution to the Riccati equation in (13) and (14).

$$\begin{aligned} {}^{i}P_k = {}^{i}\Phi^{a\,i}P_{k-1}{}^{i}\Phi^{a^T} + {}^{i}\Omega^{a\,i}Q^{i}\Omega^{a^T} - {}^{i}\Phi^{a\,i}P_{k-1}{}^{i}C^{a^T} \\ [{}^{i}C^{a\,i}P_{k-1}{}^{i}C^{a^T} + {}^{i}R]^{-1}{}^{i}C^{a\,i}P_{k-1}{}^{i}\Phi^{a^T} \end{aligned} \tag{13}$$

$${}^{i}L_k = {}^{i}P_k{}^{i}C^{a^T}[{}^{i}C^{a\,i}P_k{}^{i}C^{a^T} + {}^{i}R] \tag{14}$$

The solution to the Riccati equation in (14), ${}^{i}L_k$, is known as the Kalman gain. The Q and R terms in the Riccati equation are stochastic terms representing the variance on the input disturbance and output measurement respectively. For the example studied in this paper, both are diagonal matrices, Q with dimensions equal to the number of disturbances being estimated and R with dimensions equal to the number of measured outputs. Muske and Badgwell [2] show that for unbiased estimates, the number of estimated disturbances cannot exceed the number of measured outputs. In practice, the variances are not known, and Q and R become tuning parameters, conventionally expressed as the ratio Q/R, for scalar noise terms. Each model i in the model bank has a Q and R matrix. Since Q/R is a tuning parameter and all models are independent of one another, there is no restriction on the magnitude and values of ${}^{i}Q$ and ${}^{i}R$, with ${}^{i}Q$ and ${}^{i}R$ tuned online for each system being controlled.

2.3 Model Weighting Calculation

Once the models are updated with information from the most recent measurement, the predicted outputs are passed to the model weighting calculation. For each model and associated predicted output, a corresponding weight is calculated. The weights are normalized so that the sum of all n weights is unity, and the closer a models' weight is to unity, the better that model is at representing the plant, relative to the other models in the model bank. The weights are calculated based on residuals of each model in the model bank, with the residuals defined in (15).

$${}^{i}\varepsilon_k = y_k - {}^{i}\hat{y}_{k|k} \tag{15}$$

The model predicted outputs ${}^{i}\hat{y}_{k|k}$ are calculated based on state estimation updates in (11). The formula used to calculate the model weights is based on Bayes probability theorem, given in (16).

$$ {}^{i}\rho_k = \frac{\exp(-0.5\,{}^{i}\varepsilon_k^T\,{}^{i}\Lambda\,{}^{i}\varepsilon_k)\,{}^{i}\rho_{k-1}}{\sum_{j=1}^{n}\exp(-0.5\,{}^{j}\varepsilon_k^T\,{}^{i}\Lambda\,{}^{j}\varepsilon_k)^{j}\,\rho_{k-1}} \tag{16} $$

The exponential term in the Bayesian probability comes from the Gaussian probability distribution function and is based on the assumption that the underlying model weight probabilities are Gaussian and stochastic in nature. The exponential term in (16) also has the added benefit of rejecting models that do not represent the current plant state exponentially fast. The ${}^{i}\Lambda$ in (16) is a diagonal scaling matrix for the residuals, and is based on the covariances of each model. As the covariances are unknown in practice, the ${}^{i}\Lambda$ matrix is a tuning parameter that is adjusted to achieve desired control behavior. The difference in residuals due to different underlying linear models drives the calculation in (16) while ${}^{i}\Lambda$ affects the speed at which models evolve in the model bank. The term ${}^{i}\rho_k$ represents the probability of the i^{th} model representing the plant at the k^{th} time step. The probability calculation is recursive, as it relies on information from the previous time step ${}^{i}\rho_{k-1}$. Due to this recursion, if the probability of any model reaches zero, there is no way for that probability to become non-zero at a future time step, even if the model is a more accurate representation of the plant. To account for this and allow every model to remain active in the calculation, an artificial lower limit on the probability is enforced. Any probability that drops below this limit, represented by δ, is set equal to δ. The model weights are calculated by normalizing the probabilities, according to the formula in (17).

$$ {}^{i}w_k = \begin{cases} \frac{{}^{i}\rho_k}{\sum_{j=1}^{n}{}^{j}\rho_k} & {}^{i}\rho_k > \delta \\ 0 & {}^{i}\rho_k < \delta \end{cases} \tag{17} $$

The probability calculation in (16) is recursive, so an initial value for the probability of each model is required prior to the first control calculation. Without *a priori* knowledge of the system, each disturbance model starts with the same initial weight.

In multiple model control strategies that "hard switch" between controllers, bumpless transfer is a critical consideration when switching between controllers. Proper initialization of the controller switching on is required to avoid a discontinuity in manipulated control action, resulting in a continuous, or bumpless, transition between controllers. It is important to note that bumpless transfer is not a concern in the developed multiple model predictive control strategy. The evolution of weights according to (16) and (17) represent a continuously evolving model and the multiple model strategy formulated in this paper does not contain multiple distinct controllers. The strategy uses the evolving weights to calculate manipulated control action without discontinuities.

3 Model Predictive Control

The fundamental control strategy used in multiple model predictive control is based on linear model predictive control. At each time step, an optimization problem is

formulated and solved. The objective function is to minimize control action over a prediction horizon of p timesteps. The decision variables are m control moves, where m is the control horizon. Only the first control move is applied to the system, the model is updated, and the entire process is repeated at the next time step.

The model that is used in the control calculation is the average linear model defined in (18).

$$\overline{y}_{k+j|k} = \sum_{i=1}^{n} {}^{i}w_k \, {}^{i}\hat{y}_{k+j|k} \tag{18}$$

The vector ${}^{i}w_k$ is the weight vector defined in (17), and the ${}^{i}\hat{y}_{k+j|k}$ terms are the individual model predicted outputs generated from the models defined in (1) and calculated with the state estimation in (11). From the average linear model in (18), the next step is to derive a model predictive control solution. In the objective function that follows, the first term represents the error over the prediction horizon and the second term is a penalty on control actions.

$$\min \; \Phi = (Y_{sp} - \overline{Y})^T \, W_y \, (Y_{sp} - \overline{Y}) + \Delta U^T W_u \Delta U \tag{19}$$

In (19), Y_{sp} is a vector of setpoints, Δ U is a vector of optimal control moves, and $\overline{Y}$ is the vector of predicted outputs from $\overline{y}_{k+1|k}$ to $\overline{y}_{k+p|k}$, defined in (20).

$$\overline{Y} = \begin{bmatrix} \overline{y}_{k+1|k} \\ \overline{y}_{k+2|k} \\ \vdots \\ \overline{y}_{k+p|k} \end{bmatrix} \tag{20}$$

To solve for a control action for the unconstrained problem in (19), the propagation of $\overline{y}$ from $\overline{y}_{k+1|k}$ to $\overline{y}_{k+p|k}$ is needed. The propagation results in the definition of $\overline{Y}$ in (21).

$$\overline{Y} = S_x^a + S_c^a S_e \Delta U + S_c^a U_0 \tag{21}$$

The vectors ΔU and U_0 and matrices S_x^a, S_c^a, and S_e are defined as (22)-(26).

$$\Delta U = \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+m-1} \end{bmatrix} \tag{22}$$

$$U_0 = \begin{bmatrix} u_{k-1} \\ u_{k-1} \\ \vdots \\ u_{k-1} \end{bmatrix} \tag{23}$$

$$S_x^a = \sum_{j=1}^{n} {}^j w_k \begin{bmatrix} {}^jC^a\,{}^j\Phi^a \\ {}^jC^a\,{}^j\Phi^{a^2} \\ \vdots \\ {}^jC^a\,{}^j\Phi^{a^n} \end{bmatrix} {}^j\hat{x}_k^a \tag{24}$$

$$S_e = \begin{bmatrix} I & 0 \\ \vdots & \ddots \\ I & \dots\ I \end{bmatrix} \tag{25}$$

$$S_c^a = \sum_{j=1}^{n} {}^j w_k \begin{bmatrix} {}^jC^a\,{}^j\Gamma^a & & 0 \\ \vdots & \ddots & \\ {}^jC^a\,{}^j\Phi^{a^{p-1}}\,{}^j\Gamma^a & \dots & {}^jC^a\,{}^j\Gamma^a \end{bmatrix} \tag{26}$$

The optimization problem in (19) is solved analytically for ΔU.

$$\Delta U = (S_e^T S_c^{a^T} S_c^a S_e + W_u)^{-1} S_e^T S_c^{a^T} (Y_{sp} - S_x^a - S_c^a U_0) \tag{27}$$

4 Example System

The van de Vusse reactor is chosen for its challenging nonlinear behavior, including input multiplicity and nonminimum phase behavior [3] and is described by (28) and (29).

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{Af} - C_A) - k_1 C_A - k_3 C_A^2 \tag{28}$$

$$\frac{dC_B}{dt} = -\frac{F}{V} C_B + k_1 C_A - k_2 C_B \tag{29}$$

The measured output in (28) and (29) is the concentration of species B. For the van de Vusse reactor, dilution rate ($\frac{F}{V}$) is considered manipulated and feed concentration is a likely disturbance, so the disturbance input estimated in ${}^i\hat{d}_k$ is the feed concentration, C_{Af}.

Table 1 Van de Vusse parameters and steady state values (Bequette [4]

Parameter	Value	Parameter	Value
k_1	5/6 min^{-1}	$k1_2$	5/3 min^{-1}
k_3	1/6 $mol/L - min$	$C_{A,ss}$	3 $gmol/L$
$C_{B,ss}$	1.117 $gmol/L$	$C_{Af,ss}$	10.0 $gmol/L$

The next step is to populate the model bank. The values in table 1 represent a nominal model of operation on the left hand side (LHS) of the input multiplicity curve. As system behavior changes with time and disturbances affect the system, additional models are needed in the model bank. These models are based on either

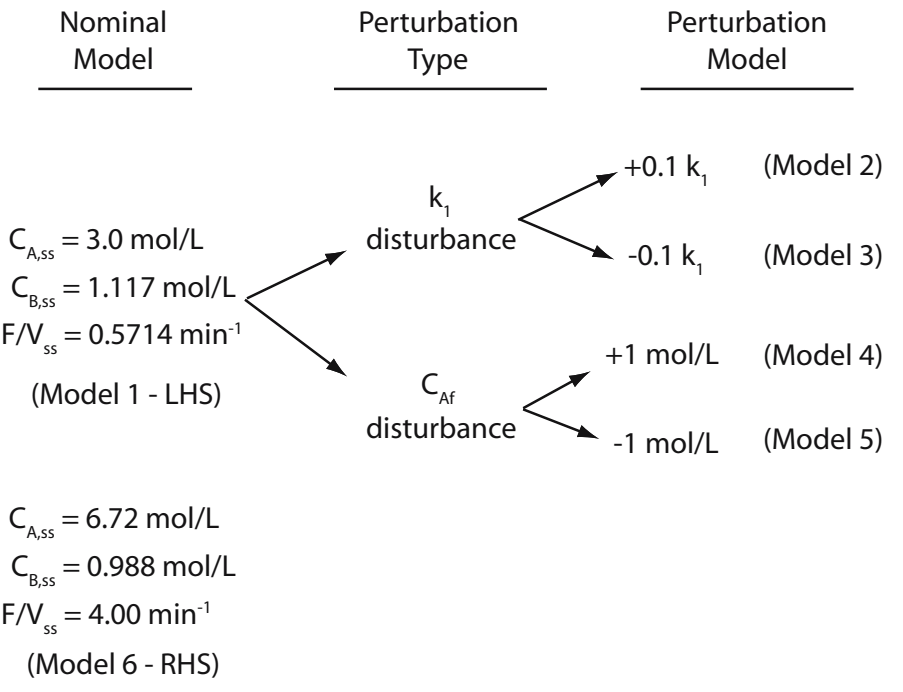


Fig. 2 Tree and branch diagram for the van de Vusse reactor illustrating the relationship between linear models in the model bank

perturbations of nominal parameters or a different steady state, and are illustrated in figure 2.

The branching in figure 2 shows how different models in the model bank are based on the same nominal model. The perturbations in figure 2 are the likely sources of disturbance, k_1 and C_{Af}. In addition, model 6 is based on a different set of nominal parameters, ones that represent behavior on the opposite, or right hand side (RHS), of the input multiplicity curve. This demonstrates how additional expected operating conditions are incorporated into the model bank.

5 Simulation Results

The first test of the van de Vusse reactor is a comparison of the two disturbance models used for state estimation, additive output and step input, for a setpoint regulation problem. There is Gaussian measurement noise with a mean value equal to five percent of the steady state measured output value added to the measured output. This is done to provide realistic conditions for the comparison, and is typical of what is expected in practice. The disturbance estimated by ${}^{i}\hat{d}_k$ in the step input disturbance model is the feed concentration, C_{Af}. The setpoint regulation results and weight evolutions are shown in figure 3.

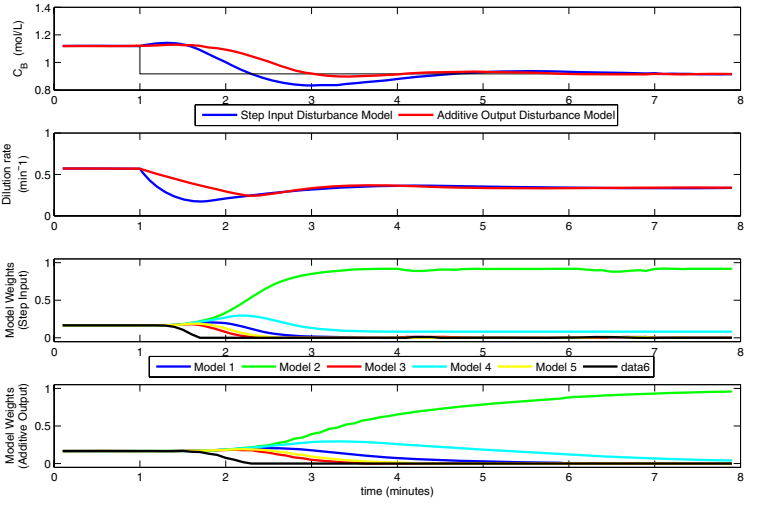


Fig. 3 Comparison of additive output and step input disturbance models for setpoint regulation; p = 30, m = 1, $\Lambda = 500$, $\delta = 0.01$, $\Delta t = 0.1$ minutes, $Q/R = 1.0$

There are several important details to note from figure 3. First is the relatively similar performance between the two disturbance models. Using mean absolute deviation (MAD) as a performance criterion, the step input disturbance model is 12 percent lower at a MAD of 0.4749 $\frac{mol}{L}$ than the 0.540 $\frac{mol}{L}$ for the additive output disturbance model. In addition, there is a marked difference in weight evolutions for the two disturbance models. The weights evolve significantly faster for the step input disturbance model. Setpoint regulation is not the only metric for comparing the two disturbance models. The purpose of the disturbance models is to estimate and reject disturbances, so it therefore makes sense to also compare the two models for their disturbance rejection capability. One common disturbance for industrial reactors is to the dilution rate, due to variations and uncertainty in flow from upstream processes. For a $+0.1\ min^{-1}$ change in dilution rate at time 6 minutes, the disturbance rejection results and weight evolutions are shown in figure 4.

The results in figure 4 show the strength of the step input disturbance model. While the additive output disturbance model is unable to reject the input disturbance, the step input disturbance model quickly rejects the disturbance. This disturbance capability, along with the lower MAD for setpoint regulation, is the justification for using the step input disturbance model in the remainder of simulation in this paper.

The parameters given in table 1 are taken from Bequette [4], and are representative of the series-parallel reaction scheme described in (28) — (29). The parameters that describe reaction kinetics: k_1, k_2 and k_3, have an initial value that is estimated in lab bench scale experiments. Once the process is online and running in a scaled up, continuous plant setting, the values of these parameters change with time and are especially sensitive during plant start-up. These changes are in response to a

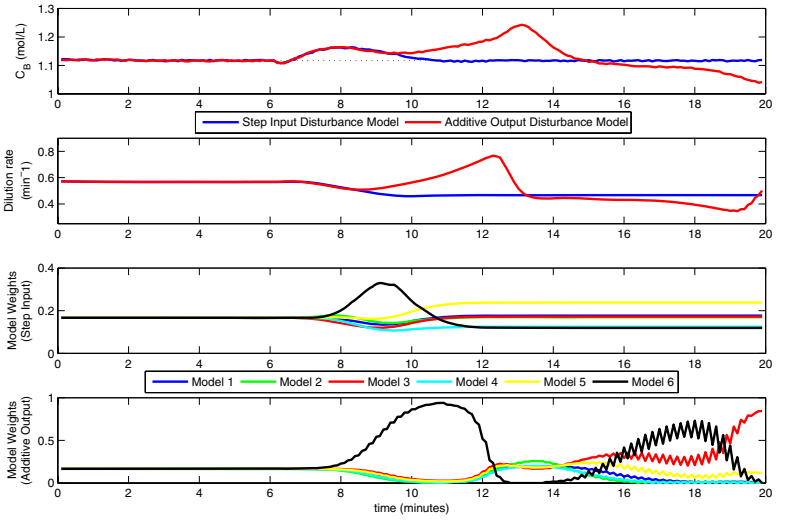


Fig. 4 Comparison of additive output and step input disturbance models for rejection of $+0.1 min^{-1}$ disturbance to the dilution rate at t = 6 minutes; p = 30, m = 1, $\Lambda = 500$, $\delta = 0.01$, $\Delta t = 0.1$ minutes, $Q/R = 1.0$

spike or surge in temperature, flowrate, concentration, etc. from an upstream process. It is important to see how well the multiple model predictive control strategy is able to handle this change in reaction kinetic parameters. Additionally, since the disturbance estimated in (4) is the feed concentration, this simulation also looks at how the multiple model predictive control strategy handles disturbances that do not match the estimated disturbance in (4). At time 10 minutes, the k_1 parameter is increased by 10 percent. The result of this sudden change in reaction kinetic parameter is shown in figure 5.

The results in figure 5 show how sensitive the multiple model predictive control strategy is to small, sudden changes in reaction kinetic parameters. Not only does the strategy bring the concentration back to setpoint, but the weighting of the models undergoes a drastic shift in response to the 10 percent change in k_1, with model 2 (representing an increase in k_1) becoming the dominant model after the introduction of the disturbance.

One of the more challenging features of the van de Vusse reactor is the presence of input multiplicity, where multiple distinct values of the manipulated input result in the same steady state measured output. While two distinct dilution rates can lead to the same concentration, they do so on opposite sides of the curve. The sign of the gain on each side of the curve is different. A large enough disturbance in the dilution rate can force the process to "cross the peak" and operate on a side of the curve with the opposite gain sign. A linear controller has a fixed gain and is unable to handle this disturbance. By creating a model bank that has models representing both sides of the curve, the multiple model predictive control strategy is able to handle this

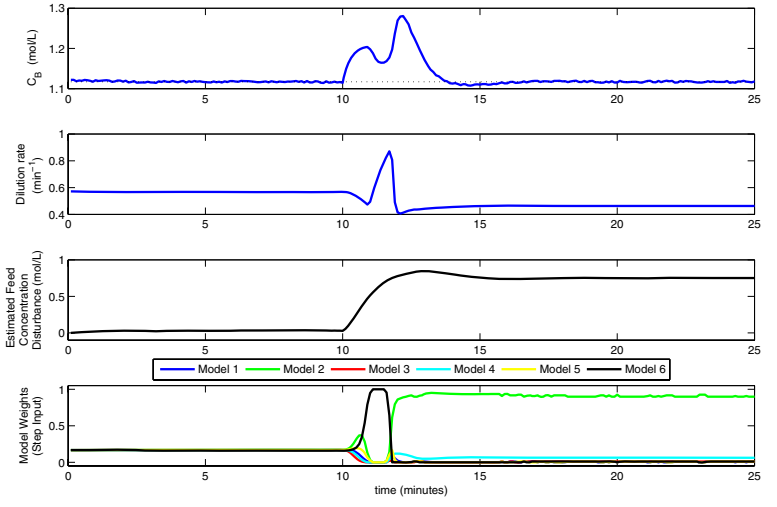


Fig. 5 Simulation of a change in reaction kinetic parameters; 10 percent change in k_1 at t = 10 minutes; p = 30, m = 1, $\Lambda = 500$, $\delta = 0.01$, $\Delta t = 0.1$ minutes, $Q/R = 1.0$

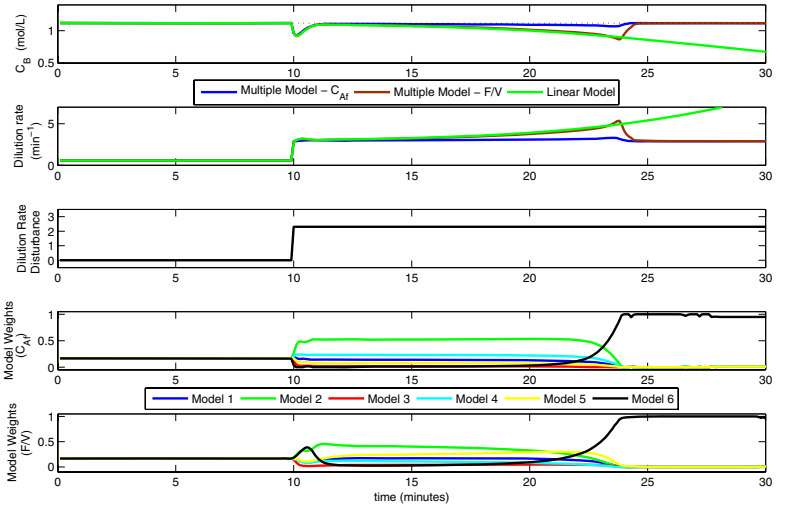


Fig. 6 Simulation of a sudden change in dilution rate causing operation to shift from the left hand side of the peak to the right hand side; p = 30, m = 1, $\Lambda = 500$, $\delta = 0.01$, $\Delta t = 0.1$ minutes, $Q/R = 1.0$

type of disturbance. The model bank has models that represent both sides of the peak: model 1 on the left hand side and model 6 on the right hand side. The results to this point are for a step input disturbance model that estimates a disturbance in feed concentration. This is an appropriate disturbance to estimate when the feed

concentration is perturbed. The step input disturbance model is also able to estimate a disturbance in the dilution rate, which is the disturbance that causes the van de Vusse reactor to "cross the peak". Performance of the two disturbance estimations for a dilution rate disturbance that forces the van de Vusse reactor onto the other side of the curve are compared with a linear model predictive controller, and the results are shown in figure 6.

The results in figure 6 are exactly what is expected. The linear model predictive controller is unable to handle the disturbance while the multiple model predictive control strategy successfully rejects the peak crossing with both disturbance estimates. The model weight evolution shows how the dominant model switches to the right hand side of the curve model after the disturbance is introduced for both disturbance estimates.

6 Summary

This paper has developed an extension of multiple model predictive control using augmented state space models. The detailed development of an unconstrained solution to the multiple model predictive control problem addresses the lack of detail in the existing literature, including model updating, state variable transition and model bank generation. The analytical unconstrained solution is tested on a challenging problem with input multiplicity.

References

1. Athans, M., Castanon, D., Dunn, K., Greene, C., Lee, W., Sandell, N., Willsky, A.: The stochastic control of the F-8C aircraft using a multiple model adaptive control method. IEEE Trans. Auto Cont. 22, 768–780 (1977)
2. Muske, K., Badgwell, T.: Disturbance modeling for offset-free linear model predictive control. J. Process Control 12, 617–632 (2002)
3. Sistu, P., Bequette, B.W.: Model predictive control of processes with input multiplicities. Chem. Eng. Sci. 50, 921–936 (1995)
4. Bequette, B.W.: Process Control: Modeling, Design and Simulation. Prentice Hall, Upper Saddle River (2003)

Stabilizing Nonlinear Predictive Control over Nondeterministic Communication Networks

R. Findeisen and P. Varutti

Abstract. Networked control systems are systems in which distributed controllers, sensors, actuators and plants are connected via a shared communication network. The use of nondeterministic networks introduces two major issues: communication delays and packet dropouts. These problems cannot be avoided and they might lead to a degradation in performance, or, even worse, to instability of the system. Thus, it is important to take network effects directly into account. In this paper, nonlinear continuous time networked control systems are considered and a nonlinear model predictive controller that is able to compensate the network nondeterminism is outlined.

Keywords: nonlinear continuous time systems, networked control systems, time-varying delays, packet dropouts, nonlinear model predictive control, stability.

1 Introduction

In recent years, the attention of the control community has focused on networked control systems (NCSs), i.e. systems where controllers, actuators, and sensors are located remotely and connected through a shared communication network. Although the idea is not new, e.g. CAN for automotive purposes, the availability of cheap and universal shared communication media like Internet has cast a new light on the way of interpreting NCSs. In fact, it can be convenient to re-use the preexisting standardized infrastructure to control systems remotely, since it allows to reduce deployment costs and to cope more efficiently with component failures, allowing for redundancies and cheap replacement. Unfortunately, using shared communication networks instead of dedicated ones introduces new control issues and challenges. Digital networks are usually nondeterministic in their behavior and the communicated information can be delayed or lost. This not only might reduce the overall system performance but also lead to instability.

R. Findeisen and P. Varutti
Institute for Automation Engineering, Otto-von-Guericke Universität, Magdeburg, Germany
e-mail: {rolf.findeisen,paolo.varutti}@ovgu.de

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 167–179.
springerlink.com

Though great attention on linear NCSs has been paid, e.g. [7, 13, 18], only a few works consider nonlinear continuous time systems, e.g. [2, 10, 16]. Often either measurement or actuator delays are taken into account. Moreover, only a few works consider packet losses, as in [12, 14]. Furthermore, specific communication protocols are usually contemplated in the controller design, as for example in [6, 7, 15, 17].

In this paper, a predictive control approach for nonlinear NCSs is presented. As shown in [4], predictive control has already demonstrated to be an effective way to deal with computational delays. In the frame of this work, we expand these results to delays in the sensor and actuator channel. In comparison to other papers, the protocol stack is abstracted and modeled as an additive delay. By labeling the information packets with time-stamps, the measurement delay can be compensated. In a similar way, by considering a worst case scenario and using smart-actuators, i.e. special actuators equipped with playback buffers, the actuation delay, as well as limited packet losses, can be solved in a simple, yet conservative way. Finally, if prediction consistent feedbacks are used, it is possible also to counteract larger packet losses. In this case, instead of sending a short piece of the input trajectory, the whole predicted control input is dispatched. Consistence between consecutive sequences should be guaranteed to ensure stability. It is proved that the proposed method is able to stabilize the closed-loop system. Simulation results for an inverted pendulum on a cart are provided.

In Section 2, the nonlinear networked control problem is presented. Section 3 introduces the new proposed method. In particular, in 3.1-3.2, the problem of measurement and actuation delays are solved by using a compensation approach, while in Section 3.3 a *prediction consistent* method to solve packet losses is outlined. Finally, to support the effectiveness of the new approach, simulation results for an inverted pendulum on a cart are presented in Section 4.

2 Problem Statement

Consider the nonlinear continuous time system

$$\dot{x}(t) = f(x(t), u(t)), \;\; x(0) = x_0, \tag{1}$$

where $x(t) \in \mathscr{X} \subseteq \mathbb{R}^n$, and $u(t) \in \mathscr{U} \subset \mathbb{R}^m$ denote respectively the state and the input vectors. The set of admissible inputs is denoted by $\mathscr{U}$, while the set of feasible states is denoted by $\mathscr{X}$. It is assumed that $\mathscr{U}$ is compact, $\mathscr{X}$ is connected, and $(0,0) \in \mathscr{X} \times \mathscr{U}$. Moreover, the vector field $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is locally Lipschitz continuous and satisfies $f(0,0) = 0$.

The following definition of partition will be used through the paper.

Definition (Partition). Every series $\pi = (t_i)$, $i \in \mathbb{N}$ of positive real numbers such that $t_0 = 0$, $t_i < t_{i+1}$ and $t_i \to \infty$ is called *partition*.

Figure 1 shows a sketch of an NCS. Here, $\tau_{sc}(t) \in [0, \overline{\tau_{sc}}]$ represents the measurement delay, while $\tau_{ca}(t) \in [0, \overline{\tau_{ca}}]$ the actuation delay, both assumed to be

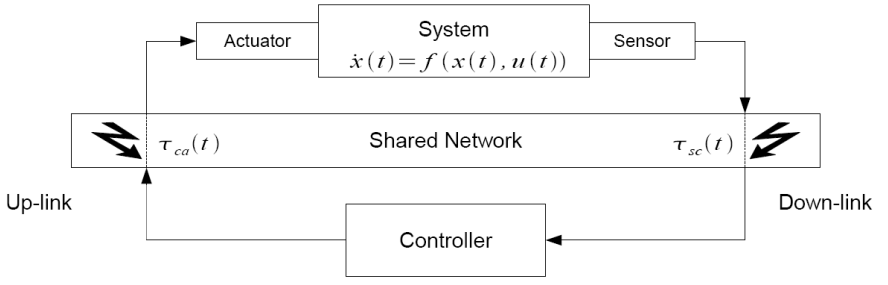


Fig. 1 Sketch of an NCS with packet dropouts, delays $\tau_{sc}(t)$, $\tau_{ca}(t)$ and random packet losses

nondeterministic. For sake of simplicity, they will be referred as τ_{sc}, and τ_{ca}. No assumption on their probability distributions is made. We assume that both are ultimately bounded by $\overline{\tau_{sc}}$ and $\overline{\tau_{ca}}$ respectively.

The communication medium is shared with other processes, such as in the Internet case. Although only one system and one controller are considered, as well as one sensor/actuator, the idea can be easily extended also to hierarchical or distributed control problems, considering multiple controller, plant, actuator, and sensor configurations. In comparison to other works, such as [6, 11], the underlying network protocols are not taken directly into account but abstracted as delays and packet dropouts.

In order to avoid ambiguities, a clear definition of packet loss is commonly known. Finally, every component - sensor, actuator, controller - has an inner clock, synchronized with all the others. Although limiting, this is a common assumption in NCSs since it provides a common time-frame, which is used to compensate the delays.

On the sensor side, periodically possibly-not-equally-distributed instances of the state $x(t)$ are measured at time $t_i \in \pi$. The information is sent to the controller through the network. A time-stamp $ts = t_i$ is associated to every packet. The controller receives this information and solves the optimization problem. Once the optimal control input is obtained, the control information is sent to the actuator, attached again with a time-stamp. The network is subject to random packet losses, which can be, for example, modeled as Bernoullian variables $S_i \sim \mathscr{B}(1-p_s)$, $A_i \sim \mathscr{B}(1-p_a)$ with $p_s, p_a \in [0,1]$, loss probabilities in the sensor/actuator link, s.t.

$$S_i = \begin{cases} 0, \text{ if a measurement packet is lost at } t_i \in \pi \\ 1, \text{ otherwise} \end{cases},$$

$$A_i = \begin{cases} 0, \text{ if an actuation packet is lost at } t_i \in \pi \\ 1, \text{ otherwise} \end{cases}.$$

Controller and actuator can either be event-driven or time-driven components. For simplicity of presentation, it is assumed that no computational delay is present.

In the next section, a method to compensated the nondeterministic network behavior is presented. Section 3.1 shows how to counteract the measurement delay, while in Section 3.2 a procedure to deal with the actuation delay is presented. Finally, in Section 3.3, a solution for packet losses is described.

3 Proposed Method

In this section, a predictive control strategy to solve the network nondeterminism is presented. Predictive techniques have demonstrated to be effective against delays, as in [4], where a compensation approach for computational delays was presented. In fact, if a model of the system is available, it can be used to forecast the current state of the plant and consequently to counteract the delays. The key challenge for a sufficiently good prediction, however, is that the input should be known at any time.

3.1 Compensation of Measurement Delays

Assume for the moment that only the sensor-to-controller channel is subject to a nondeterministic delay τ_{sc}. Since the information measured at the time t_i is available at the controller side only at $(t_i + \tau_{sc})$, the actual state of the system differs from the received piece of information. One can compensate this delay by means of suitable prediction/simulation.

In particular, consider the nonlinear system (1). Sampled-Data Nonlinear Model Predictive Control (SD-NMPC) is based on the repeated solution of an open-loop control problem, subject to the system dynamics and constraints. Based on the state at time t_i, the controller predicts the system behavior in the future over a prediction horizon T_p, such that a certain objective cost functional is minimized. The procedure is repeated at every recalculation instant $t_i \in \pi$. This is mathematically formulated as:[1]

$$\begin{aligned} &\min_{\bar{u}(\cdot)} J(\bar{u}(\cdot),\bar{x}(\cdot)) \\ s.t.\ &\dot{\bar{x}} = f(\bar{x}(\tau),\bar{u}(\tau)),\ \bar{x}(t_i) = x(t_i) \\ &\bar{u}(\tau) \in \mathscr{U}, \bar{x}(\tau) \in \mathscr{X}, \tau \in [t_i, t_i + T_p] \\ &\bar{x}(t_i + T_p) \in \mathbb{E}, \end{aligned} \tag{2}$$

where the cost functional J is typically given by

$$J(\cdot) = \int_{t_i}^{t_i+T_p} F(\bar{x}(\tau),\bar{u}(\tau))d\tau + E(\bar{x}(t_i + T_p)),$$

[1] Assuming for simplicity that the minimum is obtained.

and $u(\tau) = u^*(\tau; x(t_i))$, for $\tau \in [t_i, t_{i+1})$. It is assumed that the cost function $F : \mathscr{X} \times \mathscr{U} \to \mathbb{R}$ is locally Lipschitz continuous with $F(0,0) = 0$ and $F(x,u) > 0$, $\forall (x,u) \in \mathscr{X} \times \mathscr{U} \setminus (0,0)$.

Under certain rather mild conditions, stability in the nominal case - no delays and no packet losses - can be guaranteed, see [3, 5, 8, 9]. The idea is to consider the available model of the plant and use it to compensate the measurement delay by predicting $\hat{x}(t_i + \tau_{sc})$ from the last available measurement. The time-stamp can be used to calculate the delay just by comparison with the controller's inner clock. In this way, the optimization problem can be solved for the predicted state. The proof of stability for this compensation approach can be found in the next section.

3.2 Compensation of Actuation Delays

Assume now that an additional controller-to-actuator delay τ_{ca} affects the communication. While the measurement delays can be solved as shown in Section 3.1, actuation delays are less trivial to deal with. In fact, the presence of (stochastic) delays on the actuation side does not allow to conclude for sure which control input is applied to the plant at a specific moment, therefore compromising the state prediction.

As stated in several works, e.g. [1], nondeterministic delays can be rendered deterministic by using playback buffers. However, no former work has ever solved the control problem for nonlinear continuous time systems affected simultaneously by both input and output delays. By introducing special actuators which are able to store the control information and apply it at the proper moment - commonly called *smart actuators* -, it is possible to solve the up-link delay by considering $\overline{\tau_{ca}}$. In fact, since $\overline{\tau_{ca}}$ is assumed to be known, the state prediction $\hat{x}(t_i + \tau_{sc} + \overline{\tau_{ca}})$ can be calculated and used to solve the optimization problem. The time-stamp of the corresponding control packet is set equal to $(t_i + \tau_{sc} + \overline{\tau_{ca}})$, since it cannot be used before that time. It is responsibility of the actuator to apply the input when the time-stamp matches the actuator's inner clock. Although performance might slightly worsen, it is possible for the controller to reconstruct exactly the applied input, since the actuation delays is made deterministic.

The overall procedure can be summarized by the following algorithm.

Algorithm 1(Worst-case compensation). For all $t_i \in \pi$; t = current time; **Sensor:**

1. Measure $x(t_i)$.
2. Send the packet $[x(t_i)|ts]$, with $ts = t_i$.
3. Go to 1.

Controller:
$\texttt{buffer} = [x(t_i)|ts]_{old}$;
$\texttt{control\_input} = \{[u(\cdot)|ts_0]\}$;

1. If a new packet $[x(t_i)|ts]_{new}$ arrives
   a) If $ts_{new} \leq ts_{old}$, then discard.
   b) Else $\texttt{buffer} = [x(t_i)|ts]_{new}$.

2. $\tau_{sc} = (t - ts)$.
3. $\hat{x}(t + \overline{\tau_{ca}}) = x_{ts} + \int_{ts}^{t+\overline{\tau_{ca}}} f(x(\tau), u(\tau))d\tau$, where $u(\tau) \in$ `control_input`.
4. Solve the o.c.p. for $\hat{x}(t + \overline{\tau_{ca}}) \longrightarrow$
   $u^*(\tau; x(t_i))$, for $\tau \in (t_i + \tau_{sc} + \overline{\tau_{ca}}, t_{i+1} + \tau_{sc} + \overline{\tau_{ca}}]$, with $t_i = ts$.
5. Send $[u^*(\tau; x(t_i))|ts]$, with $ts = (t + \overline{\tau_{ca}})$.
6. Insert $[u^*(\tau; x(t_i))|ts]$ in `control_input`.
7. Go to 1.

Actuator:
`buffer` $= \{[u(\cdot)|ts_0], \ldots, [u(\cdot)|ts_n]\}$, for $ts_0 < \ldots < ts_n$;
`applied_input` $= [u(\cdot)|ts_0]$;

1. If a new packet $[u(\cdot)|ts]_{new}$ arrives
   a) Insert $[u(\cdot)|ts]_{new}$ in `buffer`.
   b) Sort `buffer` by increasing ts.
2. `temp` = first element of `buffer`.
3. If $ts_{\texttt{temp}} = t$
   a) `applied_input` = `temp`.
   b) Remove first element from `buffer`.
4. Go to 1.

By using the former method, even though sensor and actuator delays are present, it can be proved that stability can be established.

Theorem 1 (Worst-case compensation). *Given the nonlinear system (1), the closed loop using the compensation approach is stable, in the sense of asymptotic convergence, if the nominal controller, i.e. the controller subject to no delays, stabilizes the system.*

Proof. The proof is composed by two parts, feasibility and convergence, and it can be re-conducted to the nominal case (see [3]).

Feasibility: Consider any time t_i such that the optimal continuous problem has a solution. The corresponding optimal input $u^*(\tau; x(t_i))$ for $x(t_i)$ is implemented for $\tau \in (t_i + \tau_{sc} + \overline{\tau_{ca}}, t_{i+1} + \tau_{sc} + \overline{\tau_{ca}}]$, by storing the corresponding piece of trajectory in the actuator's buffer. Since it is assumed that there is no model mismatch, the prediction state $\hat{x}(t_{i+1})$ at t_{i+1} must be equal to the actual state $x(t_{i+1})$. Therefore, the remaining piece of optimal input $u^*(\tau; x(t_i))$, $\tau \in [t_{i+1}, t_i + T_p]$, where T_p is the prediction horizon, satisfies the state and input constraints. By considering the input

$$\bar{u}(\tau) = \begin{cases} u^*(\tau; x(t_i)), \tau \in [t_{i+1}, t_i + T_p] \\ u_{\mathbb{E}}(\tau), \tau \in (t_i + T_p, t_{i+1} + T_p] \end{cases}, \tag{3}$$

satisfying the constraints, the state $x(t_{i+1} + T_p; x_{t+1}, \bar{u}(\cdot))$ is reached. Thus, feasibility for t_i implies feasibility at t_{i+1}.

Convergence: By denoting the optimal cost function at t_i as the value function $V(x(t_i)) = J^*(u^*(\cdot), x(t_i))$, it can be shown that the value function is strictly decreasing. Therefore, the state converges to the origin. In fact, the value function at t_i is given by

$$V(x(t_i)) = \int_{t_i}^{t_i+T_p} F(\hat{x}(\tau;x(t_i)),u^*(\cdot;x(t_i))),u^*(\tau;x(t_i))d\tau + E(\hat{x}(t_i+T_p;x(t_i),\overline{u}(\cdot))).$$

Now, consider the cost resulting from the application of $\overline{u}$ starting from $x(t_{i+1})$:

$$J(\overline{u}(\cdot),x(t_{i+1})) = \int_{t_{i+1}}^{t_{i+1}+T_p} F(\hat{x}(\tau;x_{t_{i+1}}),\overline{u}(\cdot),\overline{u}(\tau))d\tau + E(\hat{x}(t_{i+1}+T_p;x(t_{i+1}),\overline{u}(\cdot))).$$

It can be reformulated as

$$\begin{aligned} J(\overline{u}(\cdot),x(t_{i+1})) = V(x(t_i)) &- \int_{t_i}^{t_{i+1}} F(\hat{x}(\tau;x(t_i)),u^*(\cdot;x(t_i)))d\tau \\ &- E(\hat{x}(t_i+T_p;x(t_i),u^*(\cdot;x(t_i)))) \\ &+ \int_{t_i+T_p}^{t_{i+1}+T_p} F(\hat{x}(\tau;x(t_{i+1}),\overline{u}(\cdot)),\overline{u}(\tau))d\tau \\ &+ E(\hat{x}(t_{i+1}+T_p;x(t_{i+1}),\overline{u}(\cdot))). \end{aligned}$$

By integrating the inequality $\frac{\partial E}{\partial x} f(x(\tau),u_{\mathbb{E}}(\tau)) + F(x(\tau),u_{\mathbb{E}}(\tau)) \leq 0$, where, as previously mentioned, $u_{\mathbb{E}}(\tau)$ is such that the set $\mathbb{E}$ is an invariant region, the last three terms of the former cost function can be upper bounded by zero. Thus,

$$V(x(t_i)) - J(\overline{u}(\cdot),x(t_{i+1})) \leq -\int_{t_i}^{t_{i+1}} F(\hat{x}(\tau;x(t_i)),u^*(\cdot;x(t_i)))d\tau.$$

But this is equal to

$$V(x(t_i)) - V(x(t_{i+1})) \leq -\int_{t_i}^{t_{i+1}} F(\hat{x}(\tau;x(t_i)),u^*(\cdot;x(t_i)))d\tau,$$

which is strictly decreasing for $(x,u) \neq (0,0)$. Since this last condition is verified, it is possible to apply a variant of the Barbalat's lemma to establish convergence of the state to the origin for $t \to \infty$. □

Remark 1. Only nominal asymptotic converge is ensured. This is a weaker property than asymptotic stability, meaning that not only disturbances cannot be rejected immediately, but also that, if disturbances are present, the system can temporary drift from the equilibrium point before converging. Proving stability would in the first step require to rigorously define stability for systems of discrete even nature, since the partition π is not necessarily known a priori. This is way beyond the available space and the focus of the paper.

Remark 2. Although NMPC has been adopted in the previously described method, any predictive/model-based control approach can be used. NMPC is only well suited for the purpose.

Remark 3. The compensation approach presented in the paper requires the presence of synchronized inner clocks in order to have a common time-frame among the components. This can represent a problem for fast dynamic systems, since the state-of-the-art synchronization algorithms cannot guarantee high precision.

3.3 Packet Dropouts

In the presence of packet losses, the former method is too conservative. When $S_i = 0$, i.e. a measurement packet is dropped, the local model can still be used to predict the current state of the plant by utilizing the last available measurement. Instead, if $A_i = 0$, i.e. no control input arrives, the actuator does not have any new information to use and the controller cannot be sure about what is applied to the system. The idea is to keep compensating the delays as described in Algorithm 1, and at the same time use *prediction consistent feedbacks*, i.e. $\forall t_i \in \pi$, $u^*(\tau; x(\cdot))$, for $\tau \in [t_i, t_i + T_p)$ is calculated and dispatched. When a packet is lost, the smart actuator can continue to apply the remainder of the old input, i.e. if $A_i = 0$, $u^*(\tau) = u^*(\tau; x(\cdot))$, for $\tau \in [t_{i-1}, t_{i+1})$. However, to guarantee stability a more elaborated version of NMPC is required, such that the input trajectories are able to keep consistency between consecutive state predictions even if some information is lost. The following definitions are useful to delineate prediction consistency. They are formulated in terms of a discrete system representing the dynamics in between sampling instances.

Definition (Control Invariant Set). The non-empty set $\Omega \subset \mathbb{R}^n$ is a *control invariant set* for the discrete system $x_{k+1} = \tilde{f}(x_k, u_k)$ if and only if there exist a feedback control law $u_k = g(x_k)$ such that Ω is a positive invariant set for the closed-loop system $x_{k+1} = \tilde{f}(x_k, g(x_k))$, and $u_k \in \mathscr{U}, \forall x_k \in \Omega$.

Definition (Reachable Set). The set $S_i(\Omega, T)$ is the *i-step reachable set* contained in Ω for the systems $x_{k+1} = \tilde{f}(x_k, u_k)$ if and only if T is a control invariant subset of Ω and $S_i(\Omega, T)$ contains all the states in Ω for which there exists an admissible control sequence of length i which will drive the state of the system in T in i or less steps, while keeping the evolution of the state inside Ω.

Definition (Prediction consistent feedback). Given the recalculation partition π, and the terminal set $T \subseteq \mathscr{X} \subseteq \mathbb{R}^n$, the feedback $u(\cdot)$ is called *prediction consistent* if for every recalculation time $t_i \in \pi$, $u(\cdot; x(t_i)) \in \mathscr{U}$, and, given two input trajectories $u_k(\cdot; x(t_k))$, $u_h(\cdot; x(t_h))$ obtained at the recalculation times $t_k < t_h$, the predicted states of the system (1), $\hat{x}(t_j; u_k(\cdot; x(t_k)))$, $\hat{x}(t_j; u_h(\cdot; x(t_h)))$ at the recalculation time $t_j > t_h$, obtained by applying the former inputs, belong to the same reachable set $S_j(\mathscr{X}, T)$ of the corresponding sampled-data system, $\forall t_j \in \pi$.

Essentially, this can be seen as the requirements that under information loss, the actual state of the system under the old input trajectory, drift negligibly from the controller's prediction (see Figure 2).

By using a prediction consistent feedback and the compensation approaches presented formerly, the following theorem holds.

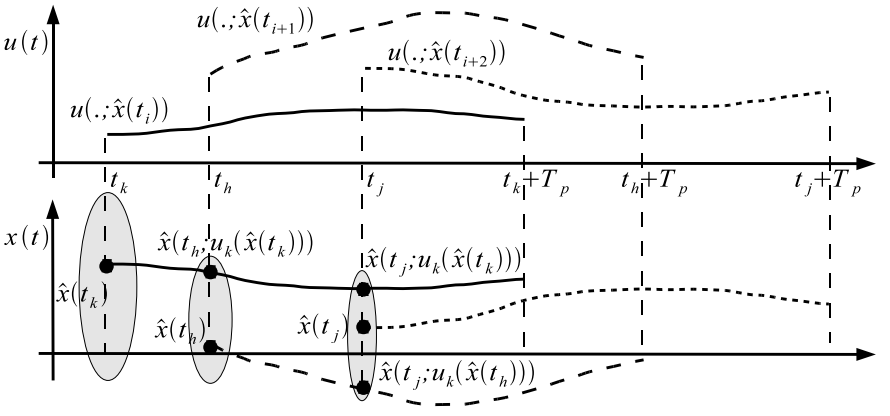


Fig. 2 Example of prediction consistent feedbacks

Theorem 2 (Dropouts compensation). *Given the system (1), the closed loop is stable if*

- *The feedback prediction is consistent.*
- *The feedback stabilizes the nominal system, i.e. the system with no delays.*
- *Compensation and smart actuators are used to counteract delays.*
- *The number of consecutive losses is less that the prediction horizon.*

Proof. Since, the prediction consistent feedbacks ensure by definition that $\hat{x}(t_i;u_i^*(\cdot)) \in S_i(\mathscr{X},\mathscr{T})$, $\forall t_i \in \pi$, from the Theorem 1 feasibility and convergence are guaranteed. □

Remark 1. Special control schemes are needed, i.e. they must be *prediction consistent*. This means that the concatenation of different input sequences under information loss must be such that the actual state of the system under the old input trajectory drift only negligibly from the controller's prediction. By using prediction consistent feedbacks, even though the applied input is not known precisely at the controller side, the closed-loop system is not destabilized.

Remark 2. Although here sample-data NMPC was used, other finite horizon NMPC approaches can be utilized, e.g. zero terminal constraint, terminal penalty constraint, terminal region constraint. Moreover, note that prediction consistency is one possible solution for the packet dropouts. In practice, every method which is able to lead to the same control sequence even though control packets are lost can be used.

Remark 3. If the number of consecutive losses A_i is longer than the prediction horizon, the system cannot be controlled anymore.

4 Simulation Results

An inverted pendulum on a cart is used as benchmark problem. Attention was focused solely on the pendulum dynamics. The mathematical model is given by the system

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = \frac{mL\cos(x_1(t))\sin(x_1(t))x_2^2(t) - g(M+m)\sin(x_1(t)) + \cos(x_1(t))u(t)}{mL\cos^2(x_1(t)) - \frac{4}{3}(m+M)L} \end{cases},$$

where $x_1(t)$ is the angular position of the pendulum, and $x_2(t)$ is its angular speed. The control input is represented by $u(t)$. The values of the coefficients m, M, and L, and the initial conditions are reported in Table 1.

Table 1 Coefficients and initial conditions for the inverted pendulum

Pendulum mass	Cart mass	Pendulum length	Initial condition	Initial prediction
$m = 0.2$ Kg	$M = 1$ Kg	$L = 1$ m	$x(0) = [0.2\ \ 0]^T$	$\hat{x}(0) = [0\ \ 0]^T$

The simulations were made in Matlab and Simulink. Sensor, actuator and controller were implemented as time-driven components, with a sampling and recalculation time equal to 0.025 seconds. Two setups are considered: one without packet losses and one with random dropouts. For sake of simplicity, both delays are assumed to be constant and multiples of the sampling time.

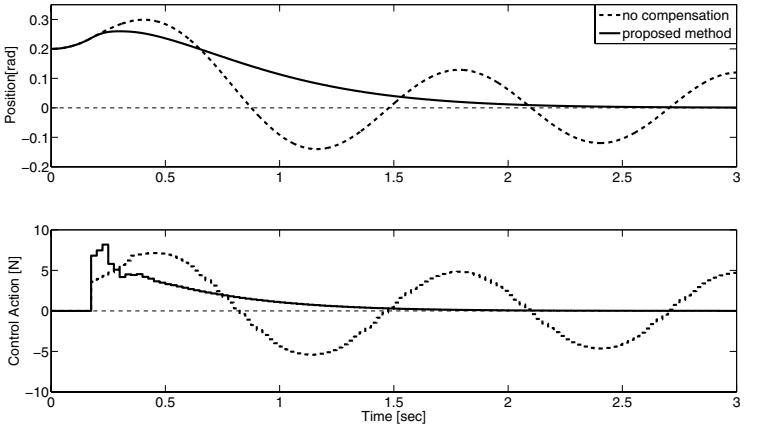


Fig. 3 Simulation results in absence of packet dropouts

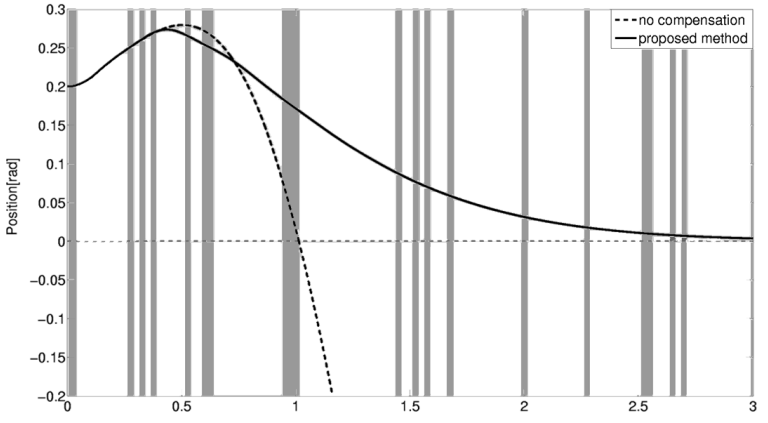


Fig. 4 Simulation results in presence of both packet losses and delays

In the first simulation, τ_{sc} is set equal to 0.1 seconds, while $\tau_{ca} = 0.05$. The corresponding results are depicted in Figure 3. Here, the dashed line represents the non-compensated case, while the solid line shows the system when the formerly proposed method is applied. The controller not only stabilizes the system, but it exhibits performance comparable to the nominal case.

In the second simulation, packet dropouts are included. A loss probability of 20% is considered, while $\tau_{sc} = 0.2$ and $\tau_{ca} = 0.1$ seconds. As shown in Figure 4, where the gray bars represents the intervarls for which the communication fails, the controller obtained with the proposed method is still able to stabilize the system under packet losses.

5 Conclusions

In this paper the problem of controlling nonlinear NCSs has been analyzed. A compensation approach to counteract measurement and actuation delays has been presented. The measurement delay can be evaluated by using a system of synchronized clocks and by labeling the sensor information with a time-stamp, representing the instant in which the information is collected from the plant. By using a predictive approach it is possible to forecast the actual state of the system, even though the communication channel is affected by random delays. Moreover, a worst-case compensation approach for actuation delays has been introduced. Solving the delays in the actuation channel is in general less trivial than the sensor one, since it is not possible a priori to be sure about the input that is applied to the system. However, whenever the maximum actuation delay is know, it is possible to use again predictive control techniques to solve the delay. This requires the use of smart actuators. Finally, packet losses have been solved by using prediction consistent techniques,

i.e. techniques that guarantee that two consecutive sequences of optimal control inputs do not differ extremely from each other. In this way, packet dropouts which are shorter that the prediction horizon can be handled. It has been proved that the proposed method is able to stabilize the close loop system, in sense of asymptotical convergence. An inverted pendulum on a cart has been used as simulation benchmark, demonstrating that the method is able to stabilize the NCS and to increase its performance.

Acknowledgements. This work was supported by the German reseach fundation in the frame of the SPP 1309 grant, and the project *Entwicklung asynchroner prädiktiver Regelungsverfahren für digital vernetzte Systeme.*

References

1. Alldredge, G., Liberatore, V., Branicky, M.S.: Play-back buffers in networked control systems, evaluation and design. In: Amer. Cont. Conf, pp. 3106–3113 (2008)
2. Carnevale, D., Teel, A.R., Nešić, D.: A lyapunov proof of an improved maximum allowable transfer interval for networked control systems. IEEE Trans. Automat. Contr. 52(5), 892–897 (2007)
3. Findeisen, R.: Nonlinear Model Predictive Control: A Sampled-Data Feedback Perspective. PhD thesis, University of Stuttgart (2006)
4. Findeisen, R., Allgöwer, F.: Computational delay in nonlinear model predictive control. In: Proc. Int. Symp. Adv. Contr. of Chem. Proc., Hong Kong, PRC, pp. 427–432 (2004)
5. Fontes, F.: A general framework to design stabilizing nonlinear model predictive controllers. Sys. Contr. Lett. 42(2), 127–143 (2001)
6. Garone, E., Sinopoli, B., Casavola, A.: LQG control for distributed systems over TCP-like erasure channels. In: Proc. 46th IEEE Conf. on Dec. and Cont., December 12–14, pp. 44–49 (2007)
7. Hespanha, J.P., Naghshtabrizi, P., Xu, Y.: A survey of recent results in networked control systems. Proc. of the IEEE 95(1), 138–162 (2007)
8. Magni, L., Scattolini, R.: Stabilizing model predictive control of nonlinear continuous time systems. Ann. Rev. in Cont. 28, 1–11 (2004)
9. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: Stability and optimality. Automatica 36(6), 789–814 (2000)
10. Nešić, D., Teel, A.R.: Input-output stability properties of networked control systems. IEEE Trans. Automat. Contr. 49(10), 1650–1667 (2004)
11. Schenato, L., Sinopoli, B., Franceschetti, M., Poolla, K., Sastry, S.S.: Foundations of control and estimation over lossy networks. Proc. of the IEEE 95(1), 163–187 (2007)
12. Sinopoli, B., Schenato, L., Franceschetti, M., Poolla, K., Sastry, S.: An LQG optimal linear controller for control systems with packet losses. In: Proc. and 2005 Eur. Cont. Conf. Dec. and Cont. CDC-ECC 2005, 44th IEEE Conf. on, December 12–15, vol. 463, pp. 458–463 (2005)
13. Sinopoli, B., Schenato, L., Franceschetti, M., Poolla, K., Sastry, S.: Optimal linear LQG control over lossy networks without packet acknowledgment. In: Proc. 45th IEEE Conf. on Dec. and Cont., December 13–15, pp. 392–397 (2006)
14. Sun, S., Xie, L., Xiao, W., Soh, Y.C.: Optimal linear estimation for systems with multiple packet dropouts. Automatica 44, 1333–1342 (2008)

15. Tipsuwan, Y., Chow, M.Y.: Control methodologies in networked control systems. Cont. Eng. Pract. 11, 1099–1111 (2003)
16. Walsh, G.C., Beldiman, O., Bushnell, L.G.: Asymptotic behavior of nonlinear networked control systems. IEEE Trans. Automat. Contr. 46(7), 1093–1097 (2001)
17. Zhang, W., Branicky, M.S., Phillips, S.M.: Stability of networked control systems. IEEE Contr. Sys. Mag, 84–99 (2001)
18. Zhu, X.L., Yang, G.H.: New results on stability analysis of networked control systems. In: Amer. Cont. Conf., pp. 3792–3797 (2008)

Distributed Model Predictive Control System Design Using Lyapunov Techniques

Jinfeng Liu, David Muñoz de la Peña, and Panagiotis D. Christofides

Abstract. In this work, we introduce a distributed control method for nonlinear process systems in which two different controllers have to coordinate their actions to regulate the state of the system to a given equilibrium point. This class of systems arises naturally when new sensors, actuators and controllers are added to already operating control loops to improve closed-loop performance and fault tolerance, taking advantage from the latest advances in wireless technology. Assuming that there exists a Lyapunov-based controller that stabilizes the closed-loop system using the pre-existing control loops, we propose to use Lyapunov-based model predictive control to design two different predictive controllers that coordinate their actions in an efficient fashion. Specifically, the proposed distributed control design preserves the stability properties of the Lyapunov-based controller, improves the closed-loop performance and is computationally more efficient compared to the corresponding centralized MPC design. The theoretical results are demonstrated using a chemical process example.

Keywords: Networked control systems; Distributed model predictive control; Nonlinear systems; Process control.

1 Introduction

Increasingly faced with the requirements of safety, environmental sustainability, and profitability, chemical process operation is relying extensively on highly automated

Jinfeng Liu
Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095-1592, USA
e-mail: jinfeng@ucla.edu

David Muñoz de la Peña
Departamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla, Camino de los Descubrimientos S/N, 41092, Sevilla, Spain
e-mail: davidmps@cartuja.us.es

Panagiotis D. Christofides
Department of Chemical and Biomolecular Engineering and Department of Electrical Engineering, University of California, Los Angeles, CA 90095-1592, USA
e-mail: pdc@seas.ucla.edu

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 181–194.
springerlink.com

control systems. The operation of chemical processes could benefit from the deployment of control systems using hybrid communication networks that take advantage of an efficient integration of the existing, point-to-point communication networks (wire connections from each actuator/sensor to the control system using dedicated local area networks) and additional networked (wired or wireless) actuator/sensor devices. Such an augmentation in sensor information and network-based availability of wired and wireless data is now well underway in the process industries [1, 2] and clearly has the potential to be transformative in the sense of dramatically improving the ability of the single-process and plantwide model-based control systems to optimize process and plant performance (in terms of achieving performance objectives that go well beyond the ones that can be achieved with control systems using wired, point-to-point connections) and prevent or deal with adverse and emergency situations more quickly and effectively (fault-tolerance).

Augmenting existing control systems with additional sensors and actuators gives rise to the need to develop coordination schemes between the different controllers that operate on a process. Lately, several distributed model predictive control (MPC) schemes have been proposed in the literature that deal with the coordination of different MPC controllers that communicate in order to obtain an optimal input trajectory in a distributed manner, see [3] for a review of results in this area. In [4], the problem of distributed control of dynamically coupled nonlinear systems that are subject to decoupled constraints was considered. In [5, 6], the effect of the coupling is modeled as a bounded disturbance compensated using a robust MPC formulation. In [7], it was proven that through multiple communications between each controller and using system-wide control objective functions, stability of the closed-loop system can be guaranteed. This problem has also been studied from a decentralized point of view in [8]. In [9] distributed MPC control of decoupled systems (a class of systems of relevance in the framework of multi-agents systems) has been studied. Within process control, important recent work on the subject of networked process control includes the development of a quasi-decentralized control framework for multi-unit plants that achieves the desired closed-loop objectives with minimal cross communication between the plant units [10].

In this paper, we continue our recent efforts [11] on the development of two-tier control architectures for networked process control problems. Specifically, we focus on networked process control problems in which two different controllers have to coordinate their actions to regulate the state of the system to a given equilibrium point. Assuming that there exists a Lyapunov-based controller that stabilizes the closed-loop system using the pre-existing control loops, we propose to use Lyapunov-based model predictive control (LMPC) theory [12, 13, 14] to design two different predictive controllers that coordinate their actions in an efficient fashion. LMPC is based on the concept of uniting model predictive control with control Lyapunov functions as a way of guaranteeing closed-loop stability. The main idea is to formulate appropriate constraints in the predictive controller's optimization problem based on an existing Lyapunov-based controller, in such a way that the MPC controller inherits the robustness and stability properties of the Lyapunov-based controller. The proposed distributed model predictive control design preserves the stability properties of the Lyapunov-based controller, improves the closed-loop

performance and is computationally more efficient compared to the corresponding centralized MPC design. The applicability and effectiveness of the proposed control method is demonstrated using a chemical process example.

2 Preliminaries

2.1 Problem Formulation

We consider nonlinear systems described by the following state-space model

$$\dot{x}(t) = f(x(t), u_1(t), u_2(t), w(t)) \tag{1}$$

where $x(t) \in R^{n_x}$ denotes the vector of state variables, $u_1(t) \in R^{n_{u1}}$ and $u_2(t) \in R^{n_{u2}}$ are two separate sets of control (manipulated) inputs and $w(t) \in R^{n_w}$ denotes the vector of disturbance variables. The disturbance vector is bounded, i.e., $w(t) \in W$ where $W := \{w \in R^{n_w} \text{ s.t. } |w| \leq \theta, \theta > 0\}$[1].

We assume that f is a locally Lipschitz vector function and $f(0,0,0,0) = 0$ which means that the origin is an equilibrium point for the nominal system (system of Eq. 1 with $w(t) = 0$ for all t) with $u_1 = 0$ and $u_2 = 0$. System of Eq. 1 is controlled with the two sets of control inputs u_1 and u_2, which could be multiple inputs of a system or a single input divided artificially into two terms (e.g., $\dot{x}(t) = \hat{f}(x(t), u(t), w(t))$ with $u(t) = u_1(t) + u_2(t)$). We also assume that the state x of the system is sampled synchronously and continuously and the time instants that we have measurement samplings are indicated by the time sequence $\{t_{k\geq 0}\}$ with $t_k = t_0 + k\Delta$, $k = 0, 1, \ldots$ where t_0 is the initial time and Δ is the sampling time.

2.2 Lyapunov-Based Controller

We assume that there exists a Lyapunov-based controller $u_1(t) = h(x(t))$, such that $f(x, h(x), 0, 0)$ is locally Lipschitz, which renders the origin of the nominal closed-loop system asymptotically stable with $u_2(t) = 0$. Using converse Lyapunov theorems [15, 16], this assumption implies that there exist functions $\alpha_i(\cdot)$, $i = 1, 2, 3, 4$ of class $\mathcal{K}$[2] and a continuously differentiable Lyapunov function $V(x)$ for the nominal closed-loop system that satisfy the following inequalities

$$\begin{aligned} \alpha_1(|x|) \leq V(x) \leq \alpha_2(|x|) \\ \frac{\partial V(x)}{\partial x} f(x, h(x), 0, 0) \leq -\alpha_3(|x|) \\ \left| \frac{\partial V(x)}{\partial x} \right| \leq \alpha_4(|x|) \end{aligned} \tag{2}$$

for all $x \in \Omega_\rho$[3] $\subseteq D$ where D is an open neighborhood of the origin and Ω_ρ denotes the stability region of the nominal closed-loop system under the control $u_1 = h(x)$ and $u_2 = 0$.

[1] $|\cdot|$ denotes Euclidean norm of a vector.

[2] Class $\mathcal{K}$ functions are strictly increasing functions of their argument and satisfy $\alpha(0) = 0$.

[3] We use Ω_r to denote the set $\Omega_r := \{x \in R^{n_x} | V(x) \leq r\}$.

By the Lipschitz property of f and the continuous differentiable property of the Lyapunov function V, there exist positive constants L_x, L_w such that

$$\left|\frac{\partial V}{\partial x}|_x f(x,u_1,u_2,w) - \frac{\partial V}{\partial x}|_{x'} f(x',u_1,u_2,0)\right| \leq L_x|x-x'| + L_w|w|. \tag{3}$$

for al $x,x' \in \Omega_\rho$ and $w \in W$. These constants will be used in the proof of the main results of this work.

2.3 Centralized Lyapunov-Based MPC

Lyapunov-based MPC (LMPC) is based on uniting receding horizon control with Lyapunov functions and computes the manipulated input trajectory solving a finite horizon constrained optimal control problem. The main advantage of LMPC approaches with respect to the Lyapunov-based controller is that optimality considerations can be taken explicitly into account (as well as constraints on the inputs and the states [13]). In particular, we propose to use the LMPC scheme proposed in [12] which guarantees practical stability of the closed-loop system. The LMPC controller is based on the previously designed Lyapunov-based controller h. The controller h is used to define a contractive constraint for the LMPC scheme which guarantees that the LMPC inherits the stability and robustness properties of the Lyapunov-based controller. The LMPC scheme introduced in [12] is based on the following optimization problem

$$\min_{u_{c1},u_{c2}\in S(\Delta)} \int_0^{N\Delta} [\tilde{x}^T(\tau)Q_c\tilde{x}(\tau) + u_{c1}^T(\tau)R_{c1}u_{c1} + u_{c2}^T(\tau)R_{c2}u_{c2}(\tau)]d\tau \tag{4a}$$

$$\text{s.t. } \dot{\tilde{x}}(\tau) = f(\tilde{x}(\tau),u_{c1}(\tau),u_{c2}(\tau),0) \tag{4b}$$

$$\tilde{x}(0) = x(t_k) \tag{4c}$$

$$\frac{\partial V(x)}{\partial x}|_{x(t_k)} f(x(t_k),u_{c1}(0),u_{c2}(0),0) \leq \frac{\partial V(x)}{\partial x}|_{x(t_k)} f(x(t_k),h(x(t_k)),0,0) \tag{4d}$$

where $S(\Delta)$ is the family of piece-wise constant functions with sampling period Δ, Q_c, R_{c1} and R_{c2} are positive definite weight matrices that define the cost, $x(t_k)$ is the state measurement obtained at t_k, $\tilde{x}$ is the predicted trajectory of the nominal system for the input trajectory computed by the LMPC, N is the prediction horizon and V is the Lyapunov function corresponding to the controller $h(x)$. The optimal solution to this optimization problem is denoted by $u_{c1}^*(\tau|t_k)$ and $u_{c2}^*(\tau|t_k)$. The LMPC controller is implemented with a receding horizon scheme; that is, at time t_k, the optimization problem is solved for $x(t_k)$ and $u_{c1}^*(0|t_k), u_{c2}^*(0|t_k)$ is applied to the system for $t \in [t_k, t_{k+1}]$. In what follows, we refer to this controller as the centralized LMPC. The optimization problem of Eq. 4 does not depend on the uncertainty and assures that the system in closed-loop with the LMPC controller of Eq. 4 maintains the stability properties of the Lyapunov-based controller $u_1 = h(x)$ with $u_2 \equiv 0$. The contractive constraint of Eq. 4d guarantees that the value of the time derivative of

the Lyapunov function at the initial evaluation time of the LMPC is lower or equal to the value obtained if only the Lyapunov-based controller $u_1 = h(x)$ is implemented in the closed-loop system. This is the contractive constraint that allows one to prove that the LMPC inherits the stability and robustness properties of the Lyapunov-based controller h.

Note that a major difference of the above LMPC scheme of Eq. 4 from some other existing Lyapunov-based MPC approaches (see for example [17, 18]) is that the constraint of Eq. 4d requires the Lyapunov function value to decay, not at the end of the prediction horizon, but only during the first time step.

3 Distributed LMPC

3.1 Distributed LMPC Formulations

It is well known that the computation time of a nonlinear MPC optimization problem grows significantly with the increase of the iteration times and the number of optimization (decision) variables. The main objective of the proposed distributed LMPC scheme is to reduce the computation burden in the evaluation of the optimal manipulated inputs while maintain the performance of the closed-loop system at a close level to the one attained when a centralized LMPC is used. In the present work, we design two separate LMPCs to compute u_1 and u_2 and refer to the LMPC computing the trajectories of u_1 and u_2 as LMPC 1 and LMPC 2, respectively. Figure 1 shows a schematic of the proposed distributed scheme. We propose to use the following implementation strategy:

1. At sampling instant t_k, both LMPC 1 and LMPC 2 receive the state measurement $x(t_k)$ from the sensors.
2. LMPC 2 computes the future input trajectory of u_2 based on $x(t_k)$ and sends the first step input value to its corresponding actuators and the entire optimal input trajectory to LMPC 1.
3. Once LMPC 1 receives the entire optimal input trajectory for u_2 from LMPC 2, it evaluates the future input trajectory of u_1 based on $x(t_k)$ and the entire optimal input trajectory of u_2 computed by LMPC 2.
4. LMPC 1 sends the first step input value of u_1 to corresponding actuators.

Remark 1. *The key idea of the proposed formulation is to impose a hierarchy on the order in which the controllers are evaluated. In this work, we assume flawless communication. If data losses are taken into account, the control scheme has to be modified because the coordination between the two LMPCs is not guaranteed.*

First we define the LMPC 2 optimization problem. This optimization problem depends of the latest state measurement $x(t_k)$, however, LMPC 2 does not have any information on the value that u_1 will take. In order to make a decision, LMPC 2 must assume a trajectory for u_1 along the prediction horizon. To this end, the Lyapunov-based controller $u_1 = h(x)$ is used. In order to inherit the stability properties of this controller, u_2 must satisfy a contractive constraint that guarantees a given minimum

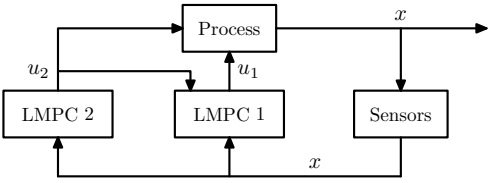


Fig. 1 Distributed LMPC control system

decrease rate of the Lyapunov function V. The LMPC 2 controller is based on the following optimization problem:

$$\min_{u_{d2}\in S(\Delta)} \int_0^{N\Delta} [\tilde{x}^T(\tau)Q_c\tilde{x}(\tau) + u_{d1}^T(\tau)R_{c1}u_{d1}(\tau) + u_{d2}^T(\tau)R_{c2}u_{d2}(\tau)]d\tau \tag{5a}$$

$$\dot{\tilde{x}}(\tau) = f(\tilde{x}(\tau), u_{d1}(\tau), u_{d2}(\tau), 0) \tag{5b}$$

$$u_{d1}(\tau) = h(\tilde{x}(j\Delta)) \;\; \forall\, \tau \in [j\Delta, (j+1)\Delta),\; j = 1, ..., N-1 \tag{5c}$$

$$\tilde{x}(0) = x(t_k) \tag{5d}$$

$$\frac{\partial V(x)}{\partial x}|_{x(t_k)} f(x(t_k), h(x(t_k)), u_{d2}(0), 0) \leq \frac{\partial V(x)}{\partial x}|_{x(t_k)} f(x(t_k), h(x(t_k)), 0, 0) \tag{5e}$$

where $\tilde{x}$ is the predicted trajectory of the nominal system with u_2 being the input trajectory computed by the LMPC of Eq. 5 (i.e., LMPC 2) and u_1 being the Lyapunov-based controller $h(x(t_k))$ applied in a sample and hold fashion. The optimal solution to this optimization problem is denoted by $u_{d2}^*(\tau|t_k)$. The first move of this trajectory is sent to LMPC 1. The contractive constraint of Eq. 5e guarantees that the value of the time derivative of the Lyapunov function at the initial evaluation time, if $u_1 = h(x(t_k)), u_2 = u_{d2}^*(0|t_k)$ are applied, is lower than or equal to the value obtained when $u_1 = h(x(t_k)), u_2 = 0$ are applied.

The LMPC 1 optimization problem depends on the latest state measurement $x(t_k)$, and the decision taken by LMPC 2 (i.e., $u_{d2}^*(\tau|t_k)$). This allows LMPC 1 to compute an input u_1 such that the closed-loop performance is optimized, while guaranteeing that the stability properties of the Lyapunov-based controller are preserved. LMPC 1 is based on the following optimization problem:

$$\min_{u_{d1}\in S(\Delta)} \int_0^{N\Delta} [\tilde{x}^T(\tau)Q_c\tilde{x}(\tau) + u_{d1}^T(\tau)R_{c1}u_{d1}(\tau) + u_{d2}^{*T}(\tau|t_k)R_{c2}u_{d2}^*(\tau|t_k)]d\tau \tag{6a}$$

$$\dot{\tilde{x}}(\tau) = f(\tilde{x}(\tau), u_{d1}(\tau), u_{d2}^*(\tau|t_k), 0) \tag{6b}$$

$$\tilde{x}(0) = x(t_k) \tag{6c}$$

$$\frac{\partial V(x)}{\partial x}|_{x(t_k)} f(x(t_k), u_{d1}(0), u_{d2}^*(0|t_k), 0) \leq \frac{\partial V(x)}{\partial x}|_{x(t_k)} f(x(t_k), h(x(t_k)), u_{d2}^*(0|t_k), 0) \tag{6d}$$

where $\tilde{x}$ is the predicted trajectory of the nominal system with u_2 being the optimal input trajectory $u_{d2}^*(\tau|t_k)$ computed by LMPC 2 and u_1 being the input trajectory

computed by LMPC 1. The optimal solution to this optimization problem is denoted by $u_{d1}^*(\tau|t_k)$. The contractive constraint of Eq. 6d guarantees that the value of the time derivative of the Lyapunov function at the initial evaluation time, if $u_1 = u_{d1}^*(0|t_k), u_{d2}^*(0|t_k)$ are applied, is lower than or equal to the value obtained if $u_1 = h(x(t_k)), u_2 = u_{d2}^*(0|t_k)$ is applied.

The manipulated inputs of the proposed distributed LMPC design based on the above LMPC 1 and LMPC 2 are defined as follows

$$\begin{aligned} u_1(t) &= u_{d1}^*(0|t_k),\ \forall t \in [t_k, t_{k+1}) \\ u_2(t) &= u_{d2}^*(0|t_k),\ \forall t \in [t_k, t_{k+1}). \end{aligned} \tag{7}$$

Referring to the behavior (boundedness) of the optimal inputs of Eq. 7 for $x(t_k) \in \Omega_\rho$, a feasible solution to LMPC 2 and LMPC 1 is $u_2 = 0$ and $u_1 = h(x(t_{k+j}))$ $(j = 0, 1, \ldots, N-1)$, respectively, which results in finite values for the costs of Eqs. 5a and 6a. Since LMPC 2 and LMPC 1 try to minimize these costs further while satisfying the constraints of Eqs. 5e and 6d, the resulting closed-loop costs will also be finite for the optimal policies of Eq. 7, which implies $u_{d1}^*(0|t_k)$ and $u_{d2}^*(0|t_k)$ are bounded. Therefore, when $u_1 = u_{d1}^*(0|t_k)$ and $u_2 = u_{d2}^*(0|t_k)$ are applied to the closed-loop system, by the local Lipschitz property assumed for f, there exists a positive constant M such that

$$\max_{t \in [t_k, t_{k+1})} |f(x(t), u_1(t_k), u_2(t_k), w(t))| \leq M \tag{8}$$

for all $x(t) \in \Omega_\rho$ and $w(t) \in W$.

Remark 2. *In order to achieve that LMPC 2 and LMPC 1 can be solved in parallel, LMPC 1 can use old input trajectories of LMPC 2, that is, at t_k, LMPC 1 uses $u_2^*(t - t_{k-1}|t_{k-1})$ to define its optimization problem. This strategy may introduce extra errors in the optimization problem, however, and may not guarantee closed-loop stability.*

3.2 Distributed LMPC Stability

The stability property of the distributed LMPC scheme (7) is stated below.

Theorem 1. *Consider system of Eq. 1 in closed-loop with the distributed LMPC design of Eq. 7 based on a controller $u_1 = h(x)$ that satisfies the conditions of Eq. 2. Let $\varepsilon_w > 0$, $\Delta > 0$ and $\rho > \rho_s > 0$ satisfy the following constraint:*

$$-\alpha_3(\alpha_2^{-1}(\rho_s)) + L_x M \Delta + L_w \theta \leq -\varepsilon_w / \Delta. \tag{9}$$

If $x(t_0) \in \Omega_\rho$ and if $\rho^* \leq \rho$ where

$$\rho^* = \max\{V(x(t+\Delta)) : V(x(t)) \leq \rho_s\},$$

then the state $x(t)$ of the closed-loop system is ultimately bounded in Ω_{ρ^*}.

Proof: The proof consists of two parts. We first prove that the optimization problems of Eqs. 5 and 6 are feasible for all states $x \in \Omega_\rho$. Then we prove that, under the proposed distributed LMPC of Eq. 7, the state of system of Eq. 1 is ultimately bounded in a region that contains the origin.

Part 1: We prove the feasibility of LMPC 2 first, and then the feasibility of LMPC 1. All input trajectories $u_2(\tau)$ such that $u_2(\tau) = 0$, $\forall \tau \in [0, \Delta)$ satisfy the

constraints of Eq. 5, thus feasibility of LMPC 2 is guaranteed. The feasibility of the optimization problem of Eq. 6 follows because input trajectories $u_1(\tau)$ such that $u_1(\tau) = h(x(t_k))$, $\forall \tau \in [0, \Delta)$ satisfy the constraints of Eq. 6.

Part 2: From conditions of Eq. 2 and the constraints of Eqs. 5e and 6d, if $x(t_k) \in \Omega_\rho$ it follows that

$$\begin{aligned} &\tfrac{\partial V(x)}{\partial x}|_{x(t_k)} f(x(t_k), u^*_{d1}(0|t_k), u^*_{d2}(0|t_k), 0) \\ &\quad \leq \tfrac{\partial V(x)}{\partial x}|_{x(t_k)} f(x(t_k), h(x(t_k)), u^*_{d2}(0|t_k), 0) \\ &\quad \leq \tfrac{\partial V(x)}{\partial x}|_{x(t_k)} f(x(t_k), h(x(t_k)), 0, 0) \\ &\quad \leq -\alpha_3(x(t_k)). \end{aligned} \tag{10}$$

The time derivative of the Lyapunov function along the actual state trajectory $x(t)$ of system of Eq. 1 in $t \in [t_k, t_{k+1})$ is given by

$$\dot{V}(x(t)) = \tfrac{\partial V}{\partial x}|_{x(t)} f(x(t), u^*_{d1}(0|t_k), u^*_{d2}(0|t_k), w(t)).$$

Adding and subtracting $\frac{\partial V}{\partial x}|_{x(t_k)} f(x(t_k), u^*_{d1}(0|t_k), u^*_{d2}(0|t_k), 0)$ and taking into account the Eq. 10, we obtain the following inequality

$$\begin{aligned} \dot{V}(x(t)) \leq -\alpha_3(|x(t_k)|) \; &+ \; \tfrac{\partial V(x)}{\partial x}|_{x(t)} f(x(t), u^*_{d1}(0|t_k), u^*_{d2}(0|t_k), w(t)) \\ &- \; \tfrac{\partial V}{\partial x}|_{x(t_k)} f(x(t_k), u^*_{d1}(0|t_k), u^*_{d2}(0|t_k), 0). \end{aligned} \tag{11}$$

From Eqs. 2, 3 and 11, the following inequality is obtained for all $x(t_k) \in \Omega_\rho / \Omega_{\rho_s}$

$$\dot{V}(x(t)) \leq -\alpha_3(\alpha_2^{-1}(\rho_s)) + L_x|x(t) - x(t_k)| + L_w|w|.$$

By Eq. 8 and the continuity of $x(t)$, the following bound can be written for all $t \in [t_k, t_{k+1})$

$$|x(t) - x(t_k)| \leq M\Delta.$$

Using this expression, we obtain the following bound on the time derivative of the Lyapunov function for $t \in [t_k, t_{k+1})$, for all initial states $x(t_k) \in \Omega / \Omega_{\rho_s}$

$$\dot{V}(x(t)) \leq -\alpha_3(\alpha_2^{-1}(\rho_s)) + L_x M\Delta + L_w\theta.$$

If Eq. 9 is satisfied, then there exists $\varepsilon_w > 0$ such that the following inequality holds for $x(t_k) \in \Omega / \Omega_{\rho_s}$

$$\dot{V}(x(t)) \leq -\varepsilon_w / \Delta$$

in $t = [t_k, t_{k+1})$. Integrating this bound on $t \in [t_k, t_{k+1})$, we obtain that

$$\begin{aligned} &V(x(t_{k+1}) \leq V(x(t_k)) - \varepsilon_w \\ &V(x(t)) \leq V(x(t_k)), \; \forall t \in [t_k, t_{k+1}). \end{aligned} \tag{12}$$

If $x(t_0) \in \Omega / \Omega_{\rho_s}$, using Eq. 12 recursively it is proved that the state converges to Ω_{ρ_s} in finite number of sampling times without leaving the stability region. Once

the state converges to $\Omega_{\rho_s} \subset \Omega_{\rho^*}$, it remains inside Ω_{ρ^*} for all times. This statement holds because of the definition of ρ^*. This proves that the closed-loop system under the proposed distributed LMPC is ultimately bounded in Ω_{ρ^*}.

4 Application to a Reactor-Separator Process

The process considered in this example is a three vessel, reactor-separator process consisting of two continuously stirred tank reactors (CSTRs) and a flash tank separator. A feed stream to the first CSTR F_{10} contains the reactant A which is converted into the desired product B. The desired product B can then further react into an undesired side-product C. The effluent of the first CSTR along with additional fresh feed F_{20} makes up the inlet to the second CSTR. The reactions $A \rightarrow B$ and $B \rightarrow C$take place in the two CSTRs.The overhead vapor from the flash tank is condensed and recycled to the first CSTR, and the bottom product stream is removed. A small portion of the overhead is purged before being recycled to the first CSTR. All the three vessels are assumed to have static holdup. The equations describing the behavior of the system are given below:

$$
\begin{aligned}
\frac{dx_{A1}}{dt} &= \frac{F_{10}}{V_1}(x_{A10}-x_{A1}) + \frac{F_r}{V_1}(x_{Ar}-x_{A1}) - k_1 e^{\frac{-E_1}{RT_1}} x_{A1} \\
\frac{dx_{B1}}{dt} &= \frac{F_{10}}{V_1}(x_{B10}-x_{B1}) + \frac{F_r}{V_1}(x_{Br}-x_{B1}) + k_1 e^{\frac{-E_1}{RT_1}} x_{A1} - k_2 e^{\frac{-E_2}{RT_1}} x_{B1} \\
\frac{dT_1}{dt} &= \frac{F_{10}}{V_1}(T_{10}-T_1) + \frac{F_r}{V_1}(T_3-T_1) + \frac{-\Delta H_1}{C_p} k_1 e^{\frac{-E_1}{RT_1}} x_{A1} + \frac{-\Delta H_2}{C_p} k_2 e^{\frac{-E_2}{RT_1}} x_{B1} + \frac{Q_1}{\rho C_p V_1} \\
\frac{dx_{A2}}{dt} &= \frac{F_1}{V_2}(x_{A1}-x_{A2}) + \frac{F_{20}}{V_2}(x_{A20}-x_{A2}) - k_1 e^{\frac{-E_1}{RT_2}} x_{A2} \\
\frac{dx_{B2}}{dt} &= \frac{F_1}{V_2}(x_{B1}-x_{B2}) + \frac{F_{20}}{V_2}(x_{B20}-x_{B2}) + k_1 e^{\frac{-E_1}{RT_2}} x_{A2} - k_2 e^{\frac{-E_2}{RT_2}} x_{B2} \\
\frac{dT_2}{dt} &= \frac{F_1}{V_2}(T_1-T_2) + \frac{F_{20}}{V_2}(T_{20}-T_2) + \frac{-\Delta H_1}{C_p} k_1 e^{\frac{-E_1}{RT_2}} x_{A2} + \frac{-\Delta H_2}{C_p} k_2 e^{\frac{-E_2}{RT_2}} x_{B2} + \frac{Q_2}{\rho C_p V_2} \\
\frac{dx_{A3}}{dt} &= \frac{F_2}{V_3}(x_{A2}-x_{A3}) - \frac{F_r+F_p}{V_3}(x_{Ar}-x_{A3}) \\
\frac{dx_{B3}}{dt} &= \frac{F_2}{V_3}(x_{B2}-x_{B3}) - \frac{F_r+F_p}{V_3}(x_{Br}-x_{B3}) \\
\frac{dT_3}{dt} &= \frac{F_2}{V_3}(T_2-T_3) + \frac{Q_3}{\rho C_p V_3}
\end{aligned}
\tag{13}
$$

where the definitions for the variables can be found in [11]. The model of the flash tank separator was derived under the assumption that the relative volatility for each of the species remains constant within the operating temperature range of the flash tank. It has been assumed that there is a negligible amount of reaction taking place in the separator. The following algebraic equations model the composition of the overhead stream relative to the composition of the liquid holdup in the flash tank:

Table 1 Noise parameters

	σ_p	ϕ	θ_p		σ_p	ϕ	θ_p		σ_p	ϕ	θ_p
x_{A1}	1	0.7	0.25	x_{A2}	1	0.7	0.25	x_{A3}	1	0.7	0.25
x_{B1}	1	0.7	0.25	x_{B2}	1	0.7	0.25	x_{B3}	1	0.7	0.25
T_1	10	0.7	2.5	T_2	10	0.7	2.5	T_3	10	0.7	2.5

$$
\begin{aligned}
x_{Ar} &= \frac{\alpha_A x_{A3}}{\alpha_A x_{A3} + \alpha_B x_{B3} + \alpha_C x_{C3}} \\
x_{Br} &= \frac{\alpha_B x_{B3}}{\alpha_A x_{A3} + \alpha_B x_{B3} + \alpha_C x_{C3}} \\
x_{Cr} &= \frac{\alpha_C x_{C3}}{\alpha_A x_{A3} + \alpha_B x_{B3} + \alpha_C x_{C3}}
\end{aligned}
\tag{14}
$$

Each of the tanks has an external heat input. The manipulated inputs to the system are the heat inputs to the three vessels, Q_1, Q_2 and Q_3, and the feed stream flow rate to vessel 2, F_{20}.

System of Eq. 13 was simulated using a standard Euler integration method. Process noise was added to the right-hand side of each equation in system of Eq. 13 and it was generatedof the form $w_k = \phi w_{k-1} + \xi_k$ where $k = 0, 1, \ldots$ is the discrete time step of 0.001 hr, ξ_k is generated by a normally distributed random variable with standard deviation σ_p, and ϕ is the autocorrelation factor and w_k is bounded by θ_p, that is $|w_k| \leq \theta_p$. Table 1 contains the parameters.

We assume that the measurements of the temperatures T_1, T_2, T_3 and mass fractions x_{A1}, x_{B1}, x_{A2}, x_{B2}, x_{A3}, x_{B3} are available at time instants $\{t_{k\geq 0}\}$ with $t_k = t_0 + k\Delta$, $k = 0, 1, \ldots$ where t_0 is the initial time and Δ is the sampling time. For the simulations carried out in this section, we pick the initial time $t_0 = 0$ and the sampling time $\Delta = 0.02\ hr = 1.2\ min$.

Table 2 Steady-state values for u_{1s} and u_{2s}

Q_{1s}	$12.6\times10^5[KJ/hr]$	Q_{3s}	$11.88\times10^5[KJ/hr]$
Q_{2s}	$13.32\times10^5[KJ/hr]$	F_{20s}	$5.04[m^3/hr]$

Table 3 Steady-state values for x_s

x_{A1s}	0.605	x_{A2s}	0.605	x_{A3s}	0.346
x_{B1s}	0.386	x_{B2s}	0.386	x_{B3s}	0.630
T_{1s}	425.9[K]	T_{2s}	422.6[K]	T_{3s}	427.3[K]

The control objective is to regulate the system to the steady state x_s corresponding to the the operating point defined by Q_{1s}, Q_{2s}, Q_{3s} of u_{1s} and F_{20s} of u_{2s}. The steady-state values for u_{1s} and u_{2s} and the values of the steady-state are given in Table 2 and Table 3, respectively. The process model of Eq. 13 belongs to the following class of nonlinear systems

$$\dot{x}(t) = f(x(t)) + g_1(x(t))u_1(t) + g_2(x(t))u_2(t) + w(x(t))$$

where $x^T = [x_1\, x_2\, x_3\, x_4\, x_5\, x_6\, x_7\, x_8\, x_9] = [x_{A1} - x_{A1s}\, x_{B1} - x_{B1s}\, T_1 - T_{1s}\, x_{A2} - x_{A2s}\, x_{B2} - x_{B2s}\, T_2 - T_{2s}\, x_{A3} - x_{A3s}\, x_{B3} - x_{B3s}\, T_3 - T_{3s}]$ is the state, $u_1^T = [u_{11}\, u_{12}\, u_{13}] = [Q_1 - Q_{1s}\, Q_2 - Q_{2s}\, Q_3 - Q_{3s}]$ and $u_2 = F_{20} - F_{20s}$ are the manipulated inputs which are subject to the constraints $|u_{1i}| \leq 10^6\ KJ/hr$ $(i = 1,\ 2,\ 3)$ and $|u_2| \leq 3\ m^3/hr$, and $w = w_k$ is a time varying bounded noise. To demonstrate the theoretical results, we first design the Lyapunov-based controller $u_1 = h(x)$, which stabilizes the closed-loop system, as follows [19, 20]

$$h(x) = \begin{cases} -\frac{L_f V + \sqrt{L_f V^2 + L_{g1} V^4}}{L_{g1} V^2} L_{g1} V & \text{if } L_{g_1} V \neq 0 \\ 0 & \text{if } L_{g_1} V = 0 \end{cases} \tag{15}$$

where $L_f V$ and $L_{g1} V$ denote the Lie derivatives of the scalar function V with respect to the vectors fields f and g_1 respectively. We consider a Lyapunov function $V(x) = x^T P x$ with P being the following weight matrix

$$P = diag[4](5.2 \times 10^{12} \left[4\ 4\ 10^{-4}\ 4\ 4\ 10^{-4}\ 4\ 4\ 10^{-4}\right]).$$

The values of the weights in P have been chosen in such a way that the Lyapunov-based controller of Eq. 15 stabilizes the closed-loop system and provides good closed-loop performance.

Based on the Lyapunov-based controller of Eq. 15, we design the centralized and the proposed distributed LMPC schemes. In the simulations, the same parameters are used for both controllers. The prediction step is the same as the sampling time, that is $\Delta = 0.02\ hr = 1.2\ min$; the prediction horizon is chosen to be $N = 6$; and the weight matrices for the LMPC schemes are chosen as

$$Q_c = diag(\left[2 \cdot 10^3\ 2 \cdot 10^3\ 2.5\ 2 \cdot 10^3\ 2 \cdot 10^3\ 2.5\ 2 \cdot 10^3\ 2 \cdot 10^3\ 2.5\right])$$

and $R_{c1} = diag(\left[5 \cdot 10^{-12}\ 5 \cdot 10^{-12}\ 5 \cdot 10^{-12}\right])$ and $R_{c2} = 100$. The state and input trajectories of system of Eq. 13 under the proposed distributed LMPC and the centralized LMPC are shown in figures 2 and 3 from the initial state

$$x(0)^T = [0.890\ 0.110\ 388.7\ 0.886\ 0.113\ 386.3\ 0.748\ 0.251\ 390.6].$$

Figure 2 shows that both the distributed and the centralized LMPC schemes give similar closed-loop performance and drive the temperatures and the mass fractions in the closed-loop system close to the desired steady-state in about 0.3 *hr* and 0.5 *hr*, respectively.

We carried out a set of simulations to compare the proposed distributed LMPC scheme with the centralized LMPC scheme with the same parameters from a performance index point of view. Table 4 shows the total cost computed for 15 different closed-loop simulations under the proposed distributed LMPC scheme and the centralized LMPC scheme. To carry out this comparison, we have computed the total cost of each simulation with different operating conditions (different initial states and process noise) based on the index of the form $\sum_{i=0}^{i=M} x(t_i)^T Q_c x(t_i) + u_1(t_i)^T R_{c1} u_1(t_i) + u_2(t_i)^T R_{c2} u_2(t_i)$, where t_0 is the initial time of the simulations

[4] $diag(v)$ denotes a matrix with its diaganol elements being the elements of vector v and all the other elements being zeros.

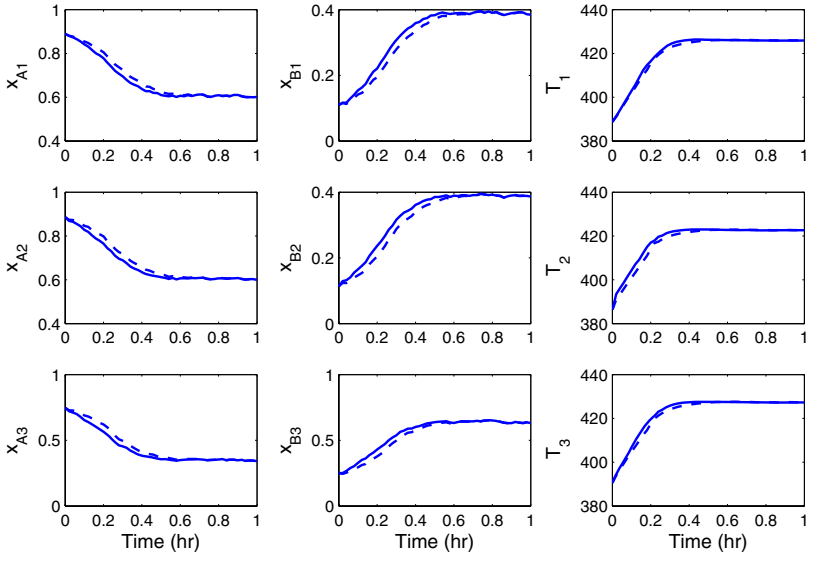


Fig. 2 State trajectories of system of Eq. 13 under the distributed LMPC (solid lines) and the centralized LMPC (dashed lines)

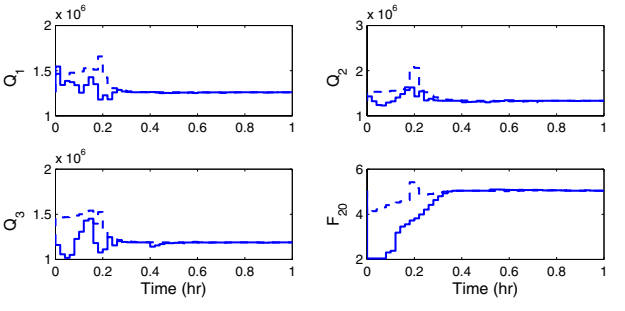


Fig. 3 Input trajectories of system of Eq. 13 under the distributed LMPC (solid lines) and the centralized LMPC (dashed lines)

and $t_M = 1$ *hr* is the end of the simulations. As we can see in Table 4, the proposed distributed LMPC scheme has a cost lower than the centralized LMPC in 10 out of 15 simulations. This demonstrates that in this example, the closed-loop performance of the distributed LMPC is comparable to the one of the centralized LMPC.

We have also carried out simulations to compare the computation time needed to evaluate the distributed LMPC scheme with that of the centralized LMPC. The simulations have been carried out using Matlab in a Pentium 3.20G Hz. The nonlinear optimization problem has been solved using the built-in function `fmincom` of Matlab. To solve the ODEs model of Eq. 13, both in the simulations and in the optimization algorithm, an Euler method with a fixed integration time of 0.001 *hr* has

Table 4 Total performance cost along the closed-loop system trajectories

sim.	Distr.	Centr.	sim.	Distr.	Centr.	sim.	Distr.	Centr.
1	65216	70868	6	83776	66637	11	62714	70951
2	70772	73112	7	61360	68897	12	76348	70547
3	57861	67723	8	47070	66818	13	49914	66869
4	62396	70914	9	79658	64342	14	89059	72431
5	60407	67109	10	65735	72819	15	78197	70257

been implemented in a mex DLL using the C programming language. For 50 evaluations, the mean time to solve the centralized LMPC optimization problem is 9.40 *s*; the mean times to solve LMPC 1 and LMPC 2 are 3.19 *s* and 4.53 *s*, respectively. From this set of simulations, we see that the computation time needed to solve the centralized LMPC is larger than the sum of the values for LMPC 1 and LMPC 2 even though the closed-loop performance in terms of the total performance cost is comparable to the one of the distributed LMPC. This is because the centralized LMPC has to optimize u_1 and u_2 in one optimization problem and the distributed LMPC has to solve two smaller (in terms of decision variables) optimization problems.

5 Conclusion

In this work, a distributed control design for nonlinear process systems was proposed. Assuming that there exists a Lyapunov-based controller that stabilizes the closed-loop system, Lyapunov-based model predictive control techniques were used to design two different predictive controllers that coordinate their actions in an efficient fashion. The proposed distributed control design preserves the stability properties of the Lyapunov-based controller, improves the closed-loop performance and is computationally more efficient compared to the corresponding centralized MPC. Simulations using a chemical plant example illustrated the applicability and effectiveness of the proposed control method.

References

1. Neumann, P.: Communication in industrial automation - what is going on? Control Engineering Practice 15, 1332–1347 (2007)
2. Christofides, P.D., Davis, J.F., El-Farra, N.H., Clark, D., Harris, K.R.D., Gipson, J.N.: Smart plant operations: Vision, progress and challenges. AIChE Journal 53, 2734–2741 (2007)
3. Camponogara, E., Jia, D., Krogh, B.H., Tulukdar, S.: Distributed model predictive control. IEEE Control System Magazine 22, 44–52 (2002)
4. Dunbar, W.B., Murray, R.M.: Distributed receding horizon control for multi-vehicle formation stabilization. Automatica 42, 549–558 (2006)
5. Richards, A., How, J.P.: Robust distributed model predictive control. International Journal of Control 80, 1517–1531 (2007)

6. Jia, D., Krogh, B.: Min-max feedback model predictive control for distributed control with communication. In: Proceedings of the American Control Conference, Anchorage, pp. 4507–4512 (2002)
7. Venkat, A.N., Rawlings, J.B., Wright, S.J.: Stability and optimality of distributed model predictive control. In: Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference ECC 2005, Seville, Spain, pp. 6680–6685 (2005)
8. Raimondo, D.M., Magni, L., Scattolini, R.: Decentralized MPC of nonlinear systems: An input-to-state stability approach. International Journal of Robust and Nonlinear Control 17, 1615–1667 (2007)
9. Keviczky, T., Borrelli, F., Balas, G.J.: Decentralized receding horizon control for large scale dynamically decoupled systems. Automatica 42, 2105–2115 (2006)
10. Sun, Y., El-Farra, N.H.: Quasi-decentralized model-based networked control of process systems. Computers and Chemical Engineering 32, 2016–2029 (2008)
11. Liu, J., Muñoz de la Peña, D., Ohran, B., Christofides, P.D., Davis, J.F.: A two-tier architecture for networked process control. Chemical Engineering Science 63, 5394–5409 (2008)
12. Mhaskar, P., El-Farra, N.H., Christofides, P.D.: Predictive control of switched nonlinear systems with scheduled mode transitions. IEEE Transactions on Automatic Control 50, 1670–1680 (2005)
13. Mhaskar, P., El-Farra, N.H., Christofides, P.D.: Stabilization of nonlinear systems with state and control constraints using Lyapunov-based predictive control. Systems and Control Letters 55, 650–659 (2006)
14. Muñoz de la Peña, D., Christofides, P.D.: Lyapunov-based model predictive control of nonlinear systems subject to data losses. IEEE Transactions on Automatic Control 53, 2076–2089 (2008)
15. Massera, J.L.: Contributions to stability theory. Ann. of Math 64, 182–206 (1956)
16. Lin, Y., Sontag, E.D., Wang, Y.: A Smooth Converse Lyapunov Theorem for Robust Stability. SIAM J. Control and Optimization 34, 124–160 (1996)
17. Primbs, J.A., Nevistic, V., Doyle, J.C.: A receding horizon generalization of pointwise min-norm controllers. IEEE Transactions on Automatic Control 45, 898–909 (2000)
18. Kothare, S.L.D., Morari, M.: Contractive model predictive control for constrained nonlinear systems. IEEE Transactions on Automatic Control 45, 1053–1071 (2000)
19. Sontag, E.: A 'universal' construction of Artstein's theorem on nonlinear stabilization. System and Control Letters 13, 117–123 (1989)
20. Christofides, P.D., El-Farra, N.H.: Control of nonlinear and hybrid process systems: Designs for uncertainty, constraints and time-delays. Springer, Heidelberg (2005)

Stabilization of Networked Control Systems by Nonlinear Model Predictive Control: A Set Invariance Approach

Gilberto Pin and Thomas Parisini

Abstract. The present paper is concerned with the robust state feedback stabilization of uncertain discrete-time constrained nonlinear systems in which the loop is closed through a packet-based communication network. In order to cope with model uncertainty, time-varying transmission delays and and packet dropouts which typically affect networked control systems, a robust control policy, which combines model predictive control with a network delay compensation strategy, is proposed.

Keywords: Networked Control Systems, Nonlinear Model Predictive Control.

1 Introduction

In the past few years, control applications in which sensor data and actuator commands are sent through a shared communication network have attracted increasing attention in control engineering, since network technologies provide a convenient way to remotely control large distributed plants [1]. These systems, usually referenced as Networked Control Systems (NCS's), are affected by the dynamics introduced by both the physical link and the communication protocol, that, in general, need to be taken in account in the design of the NCS. Various control schemes have been presented in the current literature to design effective NCS's for linear time-invariant systems [2, 3, 4, 5, 6]. Moreover, if the system to be controlled is subjected to constraints and nonlinearities, the formulation of an effective networked control strategy becomes a really hard task [7]. In this framework, the present paper provides theoretical results that motivate, under suitable assumptions, the combined use of nonlinear Model Predictive Control (MPC) with a Network Delay Compensation (NDC) strategy [8, 9], in order to cope with the simultaneous presence of model uncertainties, time-varying transmission delays and data-packet losses. In the

Gilberto Pin and Thomas Parisini
Dept. of Electrical, Electronic and Computer Engineering (DEEI),
University of Trieste, Trieste, Italy
e-mail: gilberto.pin@units.it, parisini@units.it

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 195–204.
springerlink.com

current literature, for the specific class of MPC schemes which impose a fixed terminal constraint set, X_f, as a stabilizing condition, the robust stability properties of the overall c-l system, in absence of transmission delays, has been shown to depend on the invariance properties of X_f, [10, 11]. In this regard, by resorting to invariant set theoretic arguments [12, 13], this paper aims to show that the devised NCS retains some degree of robustness also in presence of data transmission delays.

2 Main Notations

Let $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{Z}$, and $\mathbb{Z}_{\geq 0}$ denote the real, the non-negative real, the integer, and the non-negative integer sets of numbers, respectively. The Euclidean norm is denoted as $|\cdot|$. Given a compact set $A \subseteq \mathbb{R}^n$, let ∂A denote the boundary of A. Given a vector $x \in \mathbb{R}^n$, $\mathrm{d}(x,A) \triangleq \inf\{|\xi - x|, \xi \in A\}$ is the point-to-set distance from $x \in \mathbb{R}^n$ to A. Given two sets $A \subseteq \mathbb{R}^n$, $B \subseteq \mathbb{R}^n$, $\mathrm{dist}(A,B) \triangleq \inf\{\mathrm{d}(\zeta, A), \zeta \in B\}$ is the minimal set-to-set distance. The difference between two given sets $A \subseteq \mathbb{R}^n$ and $B \subseteq \mathbb{R}^n$, with $B \subseteq A$, is denoted as $A \backslash B \triangleq \{x : x \in A, x \notin B\}$. Given two sets $A \in \mathbb{R}^n$, $B \in \mathbb{R}^n$, the Pontryagin difference set C is defined as $C = A \sim B \triangleq \{x \in \mathbb{R}^n : x + \xi \in A, \forall \xi \in B\}$. Given a vector $\eta \in \mathbb{R}^n$ and a scalar $\rho \in \mathbb{R}_{>0}$, the closed ball centered in η of radius ρ is denoted as $\mathscr{B}(\eta, \rho) \triangleq \{\xi \in \mathbb{R}^n : |\xi - \eta| \leq \rho\}$. The shorthand $\mathscr{B}(\rho)$ is used when $\eta = 0$. A function $\alpha : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ belongs to class $\mathscr{K}$ if it is continuous, zero at zero, and strictly increasing.

3 Problem Formulation

Consider the nonlinear discrete-time dynamic system

$$x_{t+1} = \hat{f}(x_t, u_t) + d_t \;\; x_0 = \overline{x}, t \in \mathbb{Z}_{\geq 0}. \tag{1}$$

where $x_t \in \mathbb{R}^n$ denotes the state vector, $u_t \in \mathbb{R}^m$ the control vector and $d_t \in \mathbb{R}^n$, is an *additive transition uncertainty*. Assume that state and control variables are subject to the following constraints

$$x_t \in X, \; t \in \mathbb{Z}_{\geq 0}, \tag{2}$$

$$u_t \in U, \; t \in \mathbb{Z}_{\geq 0}, \tag{3}$$

where X and U are compact subsets of $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively, containing the origin as an interior point. With regard to the *nominal* transition map $\hat{f}(x_t, u_t)$, assume that $\hat{f}(0,0) = 0$. Moreover, let $\hat{x}_{t+j|t}$, $j \in \mathbb{Z}_{>0}$ denote the state prediction generated by means of the nominal model on the basis of the state information at time t with the input sequence $\mathbf{u}_{t,t+j-1} = \mathrm{col}[u_t, \ldots, u_{t+j-1}]$

$$\hat{x}_{t+j|t} = \hat{f}(\hat{x}_{t+j-1|t}, u_{t+j-1}), \; \hat{x}_{t|t} = x_t, t \in \mathbb{Z}_{\geq 0}, j \in \mathbb{Z}_{>0}. \tag{4}$$

Throughout the paper, all the quantities computed by the control algorithm (i.e., input sequences, state estimates and predictions) will be double-indexed, in order to denote *i*) the time instant which they refer to and *ii*) the time instant in which the information used for the computation was collected (typically it consists in a measurement of the process state).

Assumption 1 (Lipschitz). *The nominal map $\hat{f}(x,u)$ is Lipschitz with respect to x in X, with Lipschitz constant $L_{f_x} \in \mathbb{R}_{>0}$.* □

Assumption 2 (Uncertainty). *The transition uncertainty vector d_t belongs to the compact ball $D \triangleq \mathscr{B}(\bar{d})$, with $\bar{d} \in [0, +\infty)$.* □

Under the posed assumptions, the control objective consists in guaranteeing, in absence of delays, the Input-to-State Stability (ISS) of the c-l system with respect to the prescribed class of uncertainties, while keeping, also in presence of data packet dropouts and transmission delays, the state variables in the set X (i.e., achieving constraint satisfaction) despite of bounded disturbances.

With regard to the network dynamics and communication protocol, it is assumed that a set of data (packet) can be transmitted at the same time by a node. Moreover, both the sensor-to-controller and controller-to-actuator links are supposed to be affected by delays and dropouts due to the unreliable nature of networked communications. In order to cope with network delays, the data packets are Time-Stamped (TS) [5], that is, they contain the information on when they were delivered by the transmission node. In this regard, a common system clock is assumed to be accessible by sensor and actuator nodes, which are usually physically connected to the process. In general, a proper synchronization mechanism should be set up to make the time-stamping strategy work properly; the synchronization issue will not be discussed in detail here, since we will focus on the control design problem. We restrict our analysis on communication protocols in which the destination node sends an acknowledgment of successful packet reception to the source node (e.g., TCP networks [3]). The acknowledgment messages are assumed to have the highest priority, such that, after each successful packet reception, the source node receives the notification within a single time-interval. The mechanism of acknowledgment is intended for the controller to know when a packeted control sequence has been successfully transmitted to the actuator, in order to internally reconstruct the true sequence of controls which have been applied to the plant [9]. A graphical representation of the overall NCS layout is depicted in Figure 1, while the NDC and the controller will be described in the next sections.

3.1 Network Delay Compensation

In the sequel, $\tau_{ca}(t)$ and $\tau_{sc}(t)$ will denote the delays occurred respectively in the controller-to-actuator and in the sensor-to-controller links, while $\tau_a(t)$ is the age (in discrete time instant) of the control sequence used by the smart actuator to compute the current input and $\tau_c(t)$ the age of the state measurement used by the controller to compute the control actions at time t. Finally, $\tau_{rt}(t) \triangleq \tau_a(t) + \tau_c(t - \tau_a(t))$ is the so called *round trip time*, i.e., the age of the state measurement used to compute the input applied at time t.

The NDC strategy adopted, devised in [9], is based on exploiting the time stamps of the data packets in order to retain only the most recent informations at each destination node: when a novel packet is received, if it carries a more recent time-stamp than the one already in the buffer, then it takes the place of the older one, and an acknowledgment of successful packet reception is sent to the node which

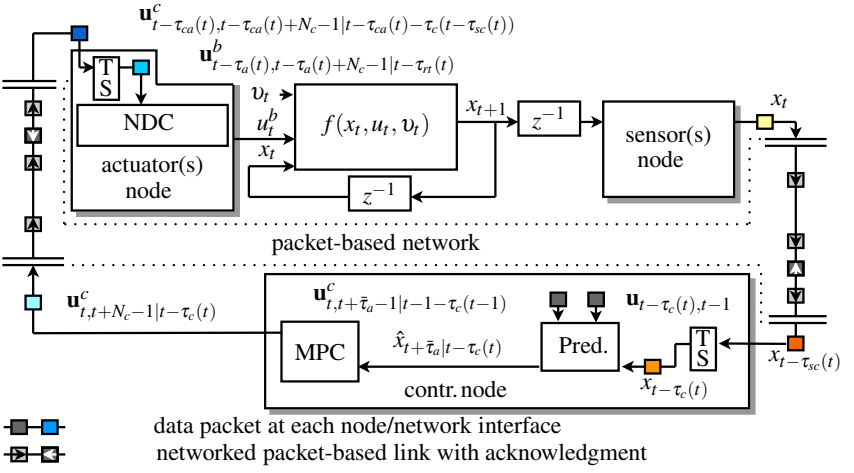


Fig. 1 Scheme of the NCS, with emphasis on the MPC controller, on Time Stamped data buffering (TS) and on the NDC components

transmitted the packet. Such a mechanism implies $\tau_a(t) \leq \tau_{ca}(t)$ and $\tau_c(t) \leq \tau_{sc}(t)$. Moreover, the adopted NDC strategy uses an input buffering mechanism [6], which requires that the controller node send a sequence of N_c control actions (with N_c to satisfy Assumption 3) to the actuator node, relying on a model-based prediction (in this case generated by the MPC), and that the smart actuator choose, at each time instant t, the correspondent input value, by setting $u_t = u^b_t$, where u^b_t is the $\tau_a(t)$-th subvector of the buffered sequence $\mathbf{u}^b_{t-\tau_a(t),t-\tau_a(t)+N_c-1}$, given by

$$\begin{aligned}\mathbf{u}^b_{t-\tau_a(t),t+N_c-1} &= \mathrm{col}[u^b_{t-\tau_a(t)},\ldots,u^b_t,\ldots,u^b_{t-\tau_a(t)+N_c-1}]\\ &= \mathbf{u}^c_{t-\tau_a(t),t+N_c-1|t-\tau_{rt}(t)}.\end{aligned}$$

where the sequence $\mathbf{u}^c_{t-\tau_a(t),t+N_c-1|t-\tau_{rt}(t)}$ had been computed at time $t-\tau_a(t)$ by the controller on the basis of the state measurement collected at time $t-\tau_{rt}(t) = t-\tau_a(t)-\tau_c(t-\tau_a(t))$. In most situations, it is natural to assume that the age of the data-packets available at the controller and actuator nodes subsumes an upper bound [9], as specified by the following assumption.

Assumption 3 (Network reliability). *The quantities $\tau_c(t)$ and $\tau_a(t)$ verify $\tau_c(t) \leq \bar\tau_c$ and $\tau_a(t) \leq \bar\tau_a$, $\forall t \in \mathbb{Z}_{>0}$, with $\bar\tau_c + \bar\tau_a + 1 < N_c$. Notably, we don't impose bounds on $\tau_{sc}(t)$ and $\tau_{ca}(t)$, allowing the presence of packet losses (infinite delay). Under these conditions, the round trip time verifies $\tau_{rt}(t) \leq \bar\tau_{rt} = \bar\tau_c + \bar\tau_a \leq N_c - 1$, $\forall t \in \mathbb{Z}_{>0}$.* □

3.2 Current State Reconstruction and Prediction

As already specified in Section 3, at time t the computation of the control actions must rely on a state measurement performed at time $t-\tau_c(t)$, $x_{t-\tau_c(t)}$. In order to recover the standard MPC formulation, the current (unavailable) state x_t has to be

reconstructed by means of the nominal model (4) and of the true input sequence $\mathbf{u}_{t-\tau_c(t),t-1}$ applied by the smart actuator to the plant, $\mathbf{u}_{t-\tau_c(t),t-1} \triangleq \mathrm{col}[u_{t-\tau_c(t)}, \ldots, u_{t-1}]$ from time $t-\tau_c(t)$ to $t-1$. In this regard, the benefits due to the use of a state predictor in NCS's are deeply discussed in [9] and [5, 6]. The sequence $\mathbf{u}_{t-\tau_c(t),t-1}$ can be internally reconstructed by the controller thanks to the acknowledgment-based protocol. Moreover, in presence of delays in the controller-to-actuator path, we must consider that the computed control sequence may not be applied entirely to the plant, and that only a subsequence of future control actions can be effectively used. In order to ensure that the sequence used for prediction would coincide with the one that will be applied to the plant, we can retain some of the elements of the control sequence computed at time $t-1$, i.e. the subsequence $\mathbf{u}^b_{t,t+\bar{\tau}_a-1|t-1-\tau_c(t-1)}$, and optimize only over the remaining elements, i.e. the sequence $\mathbf{u}_{t+\bar{\tau}_a,t+N_c-1}$, initiating the MPC with the state prediction $\hat{x}_{t+\bar{\tau}_a}$. We will show that the recursive feasibility of such a scheme can be guaranteed w.r.t. (suitably small) model uncertainty.

3.3 *Finite Horizon Predictive Controller*

In the following, we will describe the mechanism used by the controller to compute the sequence of control actions to be forwarded to the smart actuator. It relies on the solution, at each time instant t, of a Finite Horizon Optimal Control Problem (FHOCP), which uses a constraint tightening technique [10] to robustly enforce the constraints.

Problem 1 (FHOCP). Given a positive integer $N_c \in \mathbb{Z}_{\geq 0}$, at any time $t \in \mathbb{Z}_{\geq 0}$, let $\hat{x}_{t|t-\tau_c(t)}$ be the estimate of the current state obtained from the last available plant measurement $x_{t-\tau_c(t)}$ with the controls $\mathbf{u}_{t-\tau_c(t),t-1}$ already applied to the plant; moreover let $\hat{x}_{t+\bar{\tau}_a|t-\tau_c(t)}$ be the state computed from $\hat{x}_{t|t-\tau_c(t)}$ by extending the prediction using the input sequence computed at time $t-1$, $\mathbf{u}^c_{t,t+\bar{\tau}_a-1}$. Then, given a stage-cost function h, the constraint sets $X_i(\bar{d}) \subseteq X, i \in \{\bar{\tau}_a(t)+1, \ldots, N_c\}$, a terminal cost function h_f and a terminal set X_f, the *Finite Horizon Optimal Control Problem* (FHOCP) consists in minimizing, with respect to a $N_c - \bar{\tau}_a$ steps input sequence, $\mathbf{u}_{t+\bar{\tau}_a,t+N_c-1|t-\tau_c(t)} \triangleq \mathrm{col}[u_{t+\bar{\tau}_a|t-\tau_c(t)}, \ldots, u_{t+N_c-1|t-\tau_c(t)}]$, the cost function

$$J^\circ_{FH}(\hat{x}_{t+\bar{\tau}_a|t-\tau_c(t)}, \mathbf{u}^\circ_{t+\bar{\tau}_a,t+N_c-1|t-\tau_c(t)}, N_c) \triangleq \min_{\mathbf{u}_{t+\bar{\tau}_a,t+N_c-1|t-\tau_c(t)}} \sum_{l=t+\bar{\tau}_a}^{t+N_c-1} h(\hat{x}_{l|t-\tau_c(t)}, u_{l|t-\tau_c(t)}) + h_f(\hat{x}_{t+N_c|t-\tau_c(t)})$$

subject to the

i) nominal dynamics (4);

ii) input constr. $u_{t-\tau_c(t)+i|t-\tau_c(t)} \in U, i \in \{\tau_c(t)+\bar{\tau}_a, \ldots, \tau_c(t)+N_c-1\}$;

iii) state constr. $\hat{x}_{t-\tau_c(t)+i|t-\tau_c(t)} \in X_i(\bar{d}), i \in \{\tau_c(t)+\bar{\tau}_a+1, \ldots, \tau_c(t)+N_c\}$;

iv) terminal state constr. $\hat{x}_{t+N_c|t-\tau_c(t)} \in X_f$.

Finally, the sequence of controls forwarded by the controller to the actuator is constructed as $\mathbf{u}^c_{t,t+N_c-1|t-\tau_c(t)} \triangleq \mathrm{col}[\mathbf{u}^c_{t,t+\bar{\tau}_a-1|t-1-\tau_c(t-1)}, \mathbf{u}^\circ_{t+\bar{\tau}_a,t+N_c-1|t-\tau_c(t)}]$ (i.e., it is obtained by appending the solution of the FHOCP a subsequence computed at time $t-1$). In the following, we will say that a sequence $\bar{\mathbf{u}}^c_{t,t+N_c-1|t-\tau_c(t)} \triangleq$

$\mathrm{col}[\mathbf{u}^c_{t,t+\bar{\tau}_a-1|t-1-\tau_c(t-1)}, \bar{\mathbf{u}}_{t+\bar{\tau}_a,t+N_c-1|t-\tau_c(t)}]$ is feasible if the first subsequence yields to $\hat{x}_{t-\tau_c(t)+i|t-\tau_c(t)} \in X_i(\bar{d})$, $\forall i \in \{\tau_c(t)+1,\ldots,\tau_c(t)+\bar{\tau}_a$ and if the second subsequence (possibly suboptimal) satisfies all the constraints of the FHOCP set up at time t. □

By accurately choosing the stage cost h, the constraints $X_i(\bar{d})$, the terminal cost function h_f, and by imposing a terminal constraint X_f at the end of the control horizon, it is possible to show that the recursive feasibility of the scheme can be guaranteed for $t \in \mathbb{Z}_{>0}$, also in presence of norm-bounded additive transition uncertainties and network delays. Moreover, in absence of transmission delays, this class of controllers has been proven to achieve the ISS property if the the following assumption is verified [11].

Assumption 4 (κ_f,h_f,X_f). *There exist an auxiliary control law $\kappa_f(x) : X \to U$, a function $h_f(x) : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, a positive constant $L_{h_f} \in \mathbb{R}_{>0}$, a level set of h_f, $X_f \subset X$ and a positive constant $\nu \in \mathbb{R}_{>0}$ such that the following properties hold:*

i) $X_f \subset X$, *X_f closed,* $\{0\} \in X_f$;
ii) $\kappa_f(x) \in U, \ \forall x \in X_f$;
iii) $h_f(x)$ *Lipschitz in* X_f, *with L. constant* $L_{h_f} \in \mathbb{R}_{>0}$;
iv) $h_f(\hat{f}(x,\kappa_f(x))) - h_f(x) < -h(x,\kappa_f(x)), \forall x \in X_f \setminus \{0\}$; □

Now, the following Lemma (proven in the Appendix) describes how the constraint sets of the FHOCP can be computed in order to enforce the satisfaction of original constraints under the perturbed networked c-l dynamics.

Lemma 1 (State Constraints Tightening). *Under Assumptions 1 and 2, suppose (the very special case $L_{f_x}=1$ can be trivially addressed by a few suitable modifications to the proof of Lemma 1), without loss of generality, $L_{f_x} \neq 1$. If the state constraints $X_i(\bar{d})$, are computed as follows*

$$X_i(\bar{d}) \triangleq X \sim \mathscr{B}\left((L^i_{f_x}-1)/(L_{f_x}-1)\bar{d}\right), \ \forall i \in \{1,\ldots,N_c+\tau_c(t)\} \tag{5}$$

then, each feasible control sequence $\bar{\mathbf{u}}^c_{t,t+N_c-1|t-\tau_c(t)}$ guarantees that the true state will satisfy $x_{t+j} \in X$, $\forall j \in \{1,\ldots,N_c\}$, under the c-l dynamics. □

In the next section, the robust stability properties of the control policy described will be analyzed in presence of transmission delays and model uncertainty.

4 Set Invariance Theory and Robust Stability

In the following, the robust stability properties of the developed NCS with respect to model uncertainties, data transmission delays and packet-dropouts will be analyzed. To this end, the interplay between set invariance [12] and the robust stability of the c-l system will be addressed. The following definition will be used .

Definition 1 ($\mathscr{C}_i(X,\Xi)$). *Given a set $\Xi \subseteq X$, the i-step Controllability Set to Ξ, $\mathscr{C}_i(X,\Xi)$, is the set of states which can be steered to Ξ by an admissible control sequence of length i, $\mathbf{u}_{0,i-1}$, under the nominal map $\hat{f}(x,u)$, subject to constraints (2) and (3), i.e.*

$$\mathscr{C}_i(X,\Xi)\triangleq\left\{\begin{array}{l} x_0\in X\colon \exists \mathbf{u}_{0,i-1}\in U\times\ldots\times U \;\; \textit{such that} \\ \hat{x}(x_0,\mathbf{u}_{0,i-1},t)\in X,\,\forall t\in\{1,\ldots,i-1\},\hat{x}(x_0,\mathbf{u}_{0,i-1},i)\in\Xi. \end{array}\right\} \quad \square$$

In the sequel, the shorthand $\mathscr{C}_1(\Xi)$ will be used in place of $\mathscr{C}_1(\mathbb{R}^n,\Xi)$ to denote the one-step controllability set to Ξ.

Resorting to feasibility arguments, the main stability result for the designed robust networked RH scheme is asserted by the following Theorem (see Appendix 4 for the proof).

Theorem 1 (Robust Stability). *Assume that at time instant t the control sequence computed by the controller, $\bar{\mathbf{u}}^c_{t,t+N_c-1|t-\tau_c(t)}$, is feasible. Then, in view of Assumptions 1-4, if the norm bound on the uncertainty satisfies*

$$\bar{d}\leq \min_{k\in\{0,\bar{\tau}_c\}}\left\{\min\left(\begin{array}{c}(L_{f_x}-1)/(L_{f_x}^{N_c+k}-L_{f_x}^{N_c-1})\mathrm{dist}\left(\mathbb{R}^n\backslash\mathscr{C}_1(X_f),X_f\right),\\ (L_{f_x}-1)/(L_{f_x}^{N_c+k}-1)\mathrm{dist}\left(\mathbb{R}^n\backslash\hat{X}_{k+N_c}(\bar{d}),X_f\right)\end{array}\right)\right\},$$

then, the recursive feasibility of the scheme in ensured for every time instant $t+i,\forall i\in\mathbb{Z}_{>0}$, while the closed-loop trajectories are confined into X. $\square$

Conclusion

In this paper, a networked control system, based on the combined use of a model predictive controller with a network delay compensation strategy, has been designed with the aim to stabilize in a compact set a nonlinear discrete-time system, affected by unknown perturbations and subject to delayed packet-based communications in both sensor-to-controller and controller-to-actuator links. The characterization of the robust stability properties of the devised scheme represents a significant contribution in the context of nonlinear networked control systems, since it establishes the possibility to guarantee the robust enforcement of constraints under unreliable networked communications in the feedback and command channels, even in presence of model uncertainty.

Appendix

Proof (Lemma 1). Given the state measurement $x_{t-\tau_c(t)}$ available at time t at the controller node, let us consider the combined sequence of control actions formed by i) the subsequence used for estimating $\hat{x}_{t|t-\tau_c(t)}$ (i.e., the true control sequence applied by the NDC to the plant from $t-\tau_c(t)$ to t) and ii) by a feasible control sequence $\bar{\mathbf{u}}^c_{t,t+N_c-1|t-\tau_c(t)}$,

$$\mathbf{u}^*_{t-\tau_c(t),t+N_c-1|t-\tau_c(t)}\triangleq\mathrm{col}\left[\mathbf{u}_{t-\tau_c(t),t-1},\bar{\mathbf{u}}^c_{t,t+N_c-1|t-\tau_c(t)}\right], \tag{6}$$

then the prediction error $\hat{e}_{t-\tau_c(t)+i|t-\tau_c(t)}\triangleq x_{t-\tau_c(t)+i}-\hat{x}_{t-\tau_c(t)+i|t-\tau_c(t)}$, with $i\in\{1,\ldots,N_c+\tau_c(t)\}$, and $x_{t-\tau_c(t)+i}$ obtained applying $\mathbf{u}^*_{t-\tau_c(t),t+N_c-1|t-\tau_c(t)}$ in open loop to the uncertain system (1), is upper bounded by

$$|\hat{e}_{t-\tau_c(t)+i|t-\tau_c(t)}| \leq (L_{f_x}^i - 1)/(L_{f_x} - 1)\bar{d},\ \forall i \in \{1,\ldots,N_c + \tau_c(t)\}$$

where $\bar{d}$ is defined as in Assumption 2. Being $\bar{\mathbf{u}}^c_{t,t+N_c-1|t-\tau_c(t)}$ feasible, it holds that $\hat{x}_{t-\tau_c(t)+i|t-\tau_c(t)} \in X_i(\bar{d}), \forall i \in \{\tau_c(t)+1,\ldots,N_c+\tau_c(t)\}$, then it follows immediately that $x_{t-\tau_c(t)+i} = \hat{x}_{t-\tau_c(t)+i|t-\tau_c(t)} + \hat{e}_{t-\tau_c(t)+i|t-\tau_c(t)} \in X$,. ■

Appendix II

Proof (Theorem 1). The proof consists in showing that if, at time t, the input sequence computed by the controller $\bar{\mathbf{u}}^c_{t,t+N_c-1|t-\tau_c(t)}$ is feasible, then, under the perturbed c-l dynamics, there exists a feasible control sequence at time instant $t+1$ (i.e., the FHOCP is solvable and the overall sequence verifies the prescribed constraints).Finally, the recursive feasibility follows by induction. The proof will be carried out in four steps.

i) $\hat{x}_{t+N_c|t-\tau_c(t)} \in X_f \Rightarrow \hat{x}_{t+N_c+1|t+1-\tau_c(t+1)} \in X_f$: Let us consider the sequence $\mathbf{u}^*_{t-\tau_c(t),t+N_c-1|t-\tau_c(t)}$ defined in (6). It is straightforward to prove that the norm difference between the predictions $\hat{x}_{t-\tau_c(t)+j|t-\tau_c(t)}$ and $\hat{x}_{t-\tau_c(t)+j|t+1-\tau_c(t+1)}$ (initiated respectively by $x_{t-\tau_c(t)}$ and $x_{t+1-\tau_c(t+1)}$), respectively obtained by applying to the nominal model the sequence $\mathbf{u}^*_{t-\tau_c(t),t-\tau_c(t)+j-1|t-\tau_c(t)}$ and its subsequence $\mathbf{u}^*_{t+1-\tau_c(t+1),t-\tau_c(t)+j-1|t-\tau_c(t)}$, can be upper bounded by

$$|\hat{x}_{t-\tau_c(t)+j|t-\tau_c(t)+i} - \hat{x}_{t-\tau_c(t)+j|t-\tau_c(t)}| \leq \frac{1}{L_{f_x}} \sum_{l=1}^{i} L_{f_x}^{j-l+1} \bar{d} = \frac{L_{f_x}^j - L_{f_x}^{j-i}}{L_{f_x} - 1} \bar{d} \tag{7}$$

where we have posed $i = \tau_c(t) - \tau_c(t+1) + 1$ and with $j \in \{i,\ldots,N_c+\tau_c(t)\}$. Considering now the case $j = N_c + \tau_c(t)$, then (7) yields to $|\hat{x}_{t+N_c|t-\tau_c(t)+i} - \hat{x}_{t+N_c|t-\tau_c(t)}| = |\hat{x}_{t+N_c|t+1-\tau_c(t+1)} - \hat{x}_{t+N_c|t-\tau_c(t)}| \leq (L_{f_x}^{N_c+\tau_c(t)} - L_{f_x}^{N_c+\tau_c(t)-i})/(L_{f_x}-1)\bar{d}$. If the following inequality holds for all the possible values of the quantity $\tau_c(t)$ and $\tau_c(t+1)+1$, i.e. $\forall k \in \{1,\ldots,\bar{\tau}_c\}$ (the case $i = 0$ is trivial)

$$\bar{d} \leq (L_{f_x} - 1)/(L_{f_x}^{N_c+k} - L_{f_x}^{N_c-1}) \mathrm{dist}\left(\mathbb{R}^n \backslash \mathscr{C}_1(X_f), X_f\right),$$

then $\hat{x}_{t+N_c|t+1-\tau_c(t+1)} \in \mathscr{C}_1(X_f)$, whatever be the value of $\tau_c(t+1)$. Hence, there exists a control move $\bar{u}_{t+N_c|t+1-\tau_c(t+1)} \in U$ which can steer the state vector from $\hat{x}_{t+N_c|t+1-\tau_c(t+1)}$ to $\hat{x}_{t+N_c+1|t+1-\tau_c(t+1)} \in X_f$.

ii) $\hat{x}_{t-\tau_c(t)+j|t-\tau_c(t)} \in X_j(\bar{d}) \Rightarrow \hat{x}_{t-\tau_c(t)+j|t+1-\tau_c(t+1)} \in X_{j-i}(\bar{d})$, with $i = \tau_c(t) - \tau_c(t+1) + 1$ and $\forall j \in \{\tau_c(t)+1,\ldots,N_c+\tau_c(t)\}$: Consider the predictions $\hat{x}_{t-\tau_c(t)+j|t-\tau_c(t)}$ and $\hat{x}_{t-\tau_c(t)+j|t-\tau_c(t)+i}$(initiated respectively by $x_{t-\tau_c(t)}$ and $x_{t-\tau_c(t)+i}$), respectively obtained with the sequence $\mathbf{u}^*_{t-\tau_c(t),t-\tau_c(t)+j-1|t-\tau_c(t)}$ and its subsequence $\mathbf{u}^*_{t-\tau_c(t)+i,t-\tau_c(t)+j-1|t-\tau_c(t)}$. Assuming that $\hat{x}_{t-\tau_c(t)+j|t-\tau_c(t)} \in X \backsim \mathscr{B}((L_{f_x}^j - 1)/(L_{f_x} - 1)\bar{d})$, let us introduce $\eta \in \mathscr{B}((L_{f_x}^{j-i} - 1)/(L_{f_x} - 1)\bar{d})$. Let $\xi \triangleq \hat{x}_{t-\tau_c(t)+j|t-\tau_c(t)+i} - \hat{x}_{t-\tau_c(t)+j|t-\tau_c(t)} + \eta$, then, in view of Assumption 1 and thanks to (7), it follows that

$$|\xi| \leq |\hat{x}_{t-\tau_c(t)+j|t-\tau_c(t)+i} - \hat{x}_{t-\tau_{rt}(t)+j|t-\tau_c(t)}| + |\eta| \leq (L_{f_x}^j - 1)/(L_{f_x} - 1)\bar{d},$$

and hence, $\xi \in \mathscr{B}((L_{f_x}^j - 1)/(L_{f_x} - 1)\bar{d})$. Since $\hat{x}_{t-\tau_c(t)+j|t} \in X_j(\bar{d})$, it follows that $\hat{x}_{t-\tau_c(t)+j|t-\tau_c(t)} + \xi = \hat{x}_{t-\tau_c(t)+j|t-\tau_c(t)+i} + \eta \in X$, $\forall \eta \in \mathscr{B}((L_{f_x}^{j-i} - 1)/(L_{f_x} - 1)\bar{d})$, yielding to $\hat{x}_{t-\tau_c(t)+j|t+1-\tau_c(t+1)} \in X_{j-\tau_c(t)+\tau_c(t+1)-1}(\bar{d})$. Posing $k = j - \tau_c(t)$ the last inclusion can be rearranged as $\hat{x}_{t+k|t+1-\tau_c(t+1)} \in X_{k+\tau_c(t+1)}(\bar{d})$, $\forall k \in \{1, \ldots, N_c\}$

iii) $\hat{x}_{t+N_c|t-\tau_c(t)} \in X_f \Rightarrow \hat{x}_{t+1+N_c|t+1-\tau_c(t+1)} \in X_{N_c+\tau_c(t+1)}(\bar{d})$; Thanks to Point i), there exists a feasible control sequence at time $t+1$ which yields to $\hat{x}_{t+1+N_c|t+1-\tau_c(t+1)} \in X_f$. If $\bar{d}$ satisfies

$$\bar{d} \leq \min_{j \in \{N_c, \ldots, N_c + \bar{\tau}_c\}} \left\{ (L_{f_x} - 1)/(L_{f_x}^j - 1) \mathrm{dist}(\mathbb{R}^n \backslash X_j(\bar{d}), X_f) \right\},$$

it follows that the statement holds whatever be the value of $\tau_c(t+1)$.

Then, under the assumptions posed in the statement of Theorem 1, given $x_{t=0} \in X_{RH}$, and being $\tau_c(0) = 0$ (i.e. at the first time instant the actuator buffer is initiated with a feasible sequence)in view of Points i)–iii) it holds that at any time $t \in \mathbb{Z}_{>0}$ a feasible control sequence does exist and can be chosen as $\bar{\mathbf{u}}_{t+1,t+N_c+1|t+1-\tau_c(t+1)}^c = \mathrm{col}[\bar{\mathbf{u}}_{t+1,t+N_c-1|t-\tau_c(t)}^c, \bar{u}_{t+N_c|t+1-\tau_c(t+1)}]$. Therefore the recursive feasibility of the scheme is ensured. ■

References

1. Antsaklis, P., Baillieul, J.: Guest editorial: Special issue on networked control systems. IEEE Transaction on Automatic Control 49 (2006)
2. Casavola, A., Mosca, F., Papini, M.: Predictive teleoperation of constrained dynamic system via internet-like channels. In: IEEE Trans Contr. Syst. Technol., pp. 681–694 (2006)
3. Imer, O., Yuksel, S., Basar, T.: Optimal control of lti systems over unreliable communication links. Automatica 42, 1429–1439 (2006)
4. Liu, G., Xia, Y., Che, J., Rees, D., Hu, W.: Networked predictive control of systems with random network delays in both forward and feedback channels. IEEE Transactions on Industrial Electronics 54, 1282–1297 (2007)
5. Tang, P., de Silva, C.: Compensation for transmission delays in an ethernet-based control network using variable horizon predictive control. IEEE Trans. on Control Systems Technology 14, 707–716 (2006)
6. Tang, P., de Silva, C.: Stability validation of a constrained model predictive control system with future input buffering. International Journal of Control 80, 1954–1970 (2007)
7. Quevedo, D., Silva, E.I., Goodwin, G.C.: Packetized predictive control over erasure channels. In: Proc. American Control Conference, pp. 1003–1008 (2007)
8. Bemporad, A.: Predictive control of teleoperated constrained systems with unbounded communication delays. In: Proc. IEEE Conf. on Decision and Control, pp. 2133–2138 (1998)
9. Polushin, I., Liu, P., Lung, C.: On the model-based approach to nonlinear networked control systems. In: Proc. American Control Conference (2007)

10. Limón, D., Alamo, T., Camacho, E.F.: Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties. In: Proc. IEEE Conf. Decision Control, pp. 4619–4624 (2002)
11. Pin, G., Parisini, T., Magni, L., Raimondo, D.: Robust receding - horizon control of nonlinear systems with state dependent uncertainties: an input-to-state stability approach. In: Proc. American Control Conference (2008)
12. Blanchini, F.: Set invariance in control. Automatica 35(11), 1747–1767 (1999)
13. Kerrigan, E., Maciejowski, J.M.: Invariant sets for constrained nonlinear discrete-time systems with application to feasibility in model predictive control. In: Proc. IEEE Conf. Decision and Control (2000)

Nonlinear Model Predictive Control for Resource Allocation in the Management of Intermodal Container Terminals

A. Alessandri, C. Cervellera, M. Cuneo, and M. Gaggero

Abstract. Nonlinear model predictive control is proposed to allocate the available transfer resources in the management of container terminals by minimizing a performance cost function that measures the lay times of carriers over a forward horizon. Such an approach to predictive control is based on a model of the container flows inside a terminal as a system of queues. Binary variables are included into the model to represent the events of departure or stay of a carrier, thus the proposed approach requires the on-line solution of a mixed-integer nonlinear programming problem. Different techniques for solving such problem are considered that account for the presence of binary variables as well as nonlinearities into the model and cost function. The first relies on the application of a standard branch-and-bound algorithm. The second is based on the idea of dealing with the decisions associated with the binary variables as step functions. In this case, real nonlinear programming techniques are used to find a solution. Finally, a third approach is proposed that is based on the idea of approximating off line the feedback control law that results from the application of the second one. The approximation is made using a neural network that allows to construct an approximate suboptimal feedback control law by optimizing the neural weights. Simulation results are reported to compare such methodologies.

Keywords: nonlinear model predictive control, branch-and-bound, approximation, neural networks.

A. Alessandri and M. Gaggero
Department of Production Engineering, Thermoenergetics, and Mathematical Models (DIPTEM), University of Genoa, P.le Kennedy Pad. D, I-16129 Genoa, Italy
e-mail: alessandri@diptem.unige.it, gaggero@diptem.unige.it

C. Cervellera and M. Cuneo
Institute of Intelligent Systems for Automation (ISSIA-CNR), National Research Council of Italy, Via De Marini 6, I-16149 Genoa, Italy
e-mail: cervellera@ge.issia.cnr.it, marta@ge.issia.cnr.it

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 205–213.
springerlink.com

1 Introduction

The considerable growth of container shipping makes the problem of an efficient management of container terminals crucial. Toward this end, queuing theory can be used for performance evaluation, but it suffers from a poor capability of describing the dynamic behavior of the container flows inside a terminal (see, e.g., [1]). As a consequence, more powerful modeling paradigms, like, for example, discrete-event systems, were proposed that allow one to account for dynamic aspects [2].

Discrete-event tools allow one to construct very precise models of the logistics operations carried out in container terminals. Unfortunately, they are quite demanding from the computational point of view, which may become critical if the model is used for the purpose of controlling a container terminal in real time. In this paper, in line with previous works (see [3, 4]), a novel approach to the modeling of container terminals is proposed that is based on a nonlinear discrete-time dynamic model of the terminal activities, and consists in optimizing the container flows using a limited amount of resources (i.e., cranes, trucks, and other transfer machines employed inside the terminal). Using this model, an optimal allocation of such resources is searched that minimizes a given performance index according to a predictive-control strategy.

In [5], the modeling framework presented in [4] has been improved by adding binary variables that represent the events of departure or stay of a carrier. Thus, the predictive control action results from the solution of a mixed-integer nonlinear programming problem, for which we investigated the use of two methodologies. The first approach consists in using a standard branch-and-bound technique. The second one is based on the idea of treating the decisions associated with the binary variables as nondifferentiable step functions. Thus, real mathematical programming tools are required to perform the optimization. In this paper, a third approach is presented that consists in applying an approximation scheme to find suboptimal feedback control laws that result from the approximation of the optimal ones obtained via the second methodology previously described. Among the various choices for the family of nonlinear approximators, we focused on neural networks. Such class of approximators includes one-hidden-layer neural networks, which exhibit another powerful feature that consists in requiring a small number of parameters (i.e., the neural weights) to ensure a fixed approximation accuracy, especially in high-dimensional settings (see, e.g., [6, 7] and the references therein). According to such an approach, suboptimal controllers are constructed that take on the structure of a neural network, whose parameters are chosen via suitable training algorithms. The learning process is computationally demanding, but it is made off line. By contrast, the on-line use of the resulting neural controller requires a very small computation.

2 A Dynamic Model of Terminal Operations

A general maritime container terminal takes up a large storage yard, usually with three subterminals: the first for ships, the second for trucks, and the third for trains. The terminal yard is usually divided into an import area and an export area. We shall

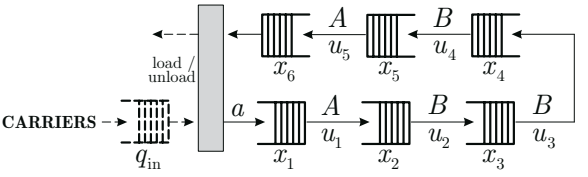


Fig. 1 Submodel of container flows related to a single transportation mode

focus on loading/unloading operations, which are critically connected to the exploitation of the available transfer resources, e.g., quay cranes (QCs), rail mounted gantry cranes (RMGCs), reach stackers (RSs), rubber tyred gantry cranes (RTGCs), and yard trucks (YTs).

In line with previous works [3, 4], we introduce a discrete-time dynamic model to describe the flows of containers that enter or leave a maritime terminal upon the arrival of carriers or that are scheduled for departure aboard other carriers, respectively. In practice, it is a system composed of container queues, each served by specific transfer resources. Such queues model the waiting times of containers before they are moved by any transfer resource. The resulting queue system can be split into three submodels, each related to a specific transportation mode, i.e., ship, truck, or train (see [5]). Without loss of generality, we shall focus only on a generic carrier type that represents a single transportation mode. To this purpose, we shall use the generic term "carrier" to indicate a ship or a truck or a train. The combination of the three single-mode submodels provides the overall model. A sketch of one of such submodels is depicted in Figure 1.

The arrival of carriers is modeled by means of the waiting queue q_{in}, where the carriers stay until the area for loading/unloading operations becomes free. We shall suppose that there exists only one area for loading/unloading, and thus that only one carrier per time may be served. We shall adopt a discrete-time representation of the queue dynamics with a sampling time equal to ΔT. For the sake of simplicity, we refer only to two generic types of transfer resources, i.e., type A and type B. We shall denote the queues by means of the corresponding lengths at time t, i.e., $x_i(t)$, $i = 1, 2, \ldots, 6$ (we measure these lengths in TEU, i.e., Twenty-foot Equivalent Unit). The queue x_1 represents the containers to be unloaded from the carrier that entered the terminal, whereas the queue x_6 represents the containers that have been loaded into that carrier; such containers are ready to leave the terminal on carrier departure. The queues x_3 and x_4 account for the container stay in the import and export area of the storage yard, respectively. The queues x_2 and x_5 model the ongoing transfer without storage during unloading operations (the import flow, i.e., the flow from x_1 to x_3) and loading operations (the export flow, i.e., the flow from x_4 to x_6), respectively. The submodel in Figure 1 can be generalized to represent the loading/unloading of more than one carrier per time. In this case, we should add more pairs of queues like x_1 and x_6, exactly as the number of contemporarily served carriers.

The control inputs $u_i(t) \in [0,1]$, $i = 1,2,\ldots,5$, are the percentages of server capacities used for container transfers at time $t = 0,1,\ldots$. The exogenous input $a(t) \geq 0$ is the quantity of containers (in TEU) that enters the terminal at time $t = 0,1,\ldots$ via the carrier that is in the loading/unloading area. Such variable is normally equal to zero, except when a carrier enters the terminal; in this case the variable becomes equal to the amount of containers that has to be unloaded from that carrier. The time-varying parameters $\mu_i(t) \geq 0$, $i = 1,\ldots,5$, are the maximum container handling capacities (in TEU/h) for the various queues at time $t = 0,1,\ldots$. Such parameters are related to the corresponding number and hourly handling rate of the available resources. They become time-invariant if we assume, for example, that there exist n_A and n_B transfer machines with the same ideal capacity given by r_A and r_B, respectively; in this case they turn out to be $\mu_i(t) = r_A n_A$, $i = 1,5$, and $\mu_i(t) = r_B n_B$, $i = 2,3,4$, for all $t = 0,1,\ldots$. Finally, let us introduce the binary variable $y(t) \in \{0,1\}$ to model the departure or stay of a carrier depending on whether the planned loading/unloading operations are finished or not at time $t = 0,1,\ldots$. More specifically, $y(t)$ is equal to 0 when the carrier in the loading/unloading area has finished all the operations, and thus can leave the terminal. Otherwise, it is equal to 1. In other words, the variable $y(t)$ takes on its values as follows:

$$y(t) = \begin{cases} 0 & \text{if} \quad x_6(t) + \Delta T \mu_5(t) u_5(t) = s(t) \quad \text{and} \quad x_1(t) = 0 \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

where $t = 0,1,\ldots$ and $s(t)$ is defined as the amount of containers (in TEU) scheduled, at time t, for loading before departure. The first condition in (1) occurs after the completion of the loading/unloading, whereas the second one is true if the loading/unloading is not complete.

The dynamic equations that result from the balance of input and output container flows for all the queues in Figure 1 are the following:

$$x_1(t+1) = x_1(t) + a(t) - \Delta T \mu_1(t) u_1(t) \tag{2a}$$
$$x_i(t+1) = x_i(t) + \Delta T [\mu_{i-1}(t) u_{i-1}(t) - \mu_i(t) u_i(t)], \; i = 2,3,4,5 \tag{2b}$$
$$x_6(t+1) = y(t)[x_6(t) + \Delta T \mu_5(t) u_5(t)] \tag{2c}$$

where $t = 0,1,\ldots$.

To account for the boundedness of the type A and type B resources, we need to impose constraints on the control variables as follows:

$$u_1(t) + u_5(t) \leq 1 \tag{3a}$$
$$u_2(t) + u_3(t) + u_4(t) \leq 1 \tag{3b}$$

where $t = 0,1,\ldots$.

In the next section, we shall use the vectors $\underline{x}(t)$, $\underline{u}(t)$, and $\underline{y}(t)$ to represent the state vector, the control vector, and the binary vector of the overall model at time t, as they result from the combination of three submodels of type (1)–(3) for the transportation modes based on ships, trucks, and trains.

3 Predictive Control of Container Flows

In this section, we shall use the above-described modeling framework to develop an optimal resource allocation strategy based on predictive control. Here the goal of predictive control is to allocate the transfer resources inside a terminal in order to optimize an objective function that is related to the performance of the terminal over a certain time horizon from the current instant.

Let us assume to know the container demands for loading/unloading with early forecasts. On the basis of such information, we can foresee the queue lengths and devise the control actions for the allocation of the resources from the current time instant for a given number of forward time steps that optimize the considered performance index. Toward this end, we refer to a function $h[\underline{x}(t),\underline{y}(t),\underline{u}(t)]$ (in general nonlinear), which provides a measure of performance that depends on $\underline{x}(t)$, $\underline{y}(t)$, and $\underline{u}(t)$.

We chose the minimization of the lay times of carriers as a management goal since a carrier that remains for a longer time in the terminal results in higher costs for both a useless employment of resources and a poor service to customers. As in [4, 5], we focused on the following function h for the submodel (1)–(3):

$$h[\underline{x}(t),\underline{y}(t),\underline{u}(t)] = c_1[1-u_1(t)]y(t) + c_2[1-u_5(t)]y(t) \tag{4}$$

where the coefficients c_1 and c_2 are positive constants. The task of reducing the lay times of carriers is indirectly accomplished by using a performance index based on (4). Indeed, given the time a carrier spends in the terminal tightly depends on how fast the containers to load/unload into/from carriers are moved, one can reduce the overall delay by keeping the control inputs as higher as possible.

Since no specific final penalty is required for the last time step in our context, a predictive control approach can be obtained by minimizing, at each time $t = 0, 1, \ldots$, a cost function of the form

$$J_t[\underline{x}(t),\underline{y}(t),\ldots,\underline{y}(t+T-1),\underline{u}(t),\ldots,\underline{u}(t+T-1)] = \sum_{k=t}^{t+T-1} h[\underline{x}(k),\underline{y}(k),\underline{u}(k)]$$

where T is the length of the forward horizon. If the predictions concerning the future are either unknown or known with large uncertainty, a one-step-ahead strategy may be adopted that consists in minimizing the previous cost function with T equal to 1, as shown in [4]. If the arrivals of carriers and the amounts of containers to load/unload are known with sufficient accuracy, one can set up the terminal operations over a forward horizon that may be longer, as a larger value of T enables one to adopt strategies that are more effective since a larger amount of information about the future is taken into account. Clearly, an increase in T entails much more computation, and hence a tradeoff between effectiveness and computational effort is needed.

To sum up, given $\underline{x}(t)$ at each time $t = 0, 1, \ldots$, we need to solve the following optimization problem:

$$\min_{\substack{\underline{u}(t),\ldots,\underline{u}(t+T-1)\\ \underline{y}(t),\ldots,\underline{y}(t+T-1)}} J_t[\underline{x}(t),\underline{y}(t),\ldots,\underline{y}(t+T-1),\underline{u}(t),\ldots,\underline{u}(t+T-1)] \quad (5)$$

subject to $\underline{u}(k) \in [0,1]^{\dim(\underline{u})}$, $\underline{y}(k) \in \{0,1\}^{\dim(\underline{y})}$, for $k = t,\ldots,t+T-1$, as well as to the extensions to the complete three-mode model of the constraints that correspond to the following: right-hand side of (2) $\geq \underline{0}$ and (3). Once the optimal values $\underline{u}^\circ(t),\ldots,\underline{u}^\circ(t+T-1)$ and $\underline{y}^\circ(t),\ldots,\underline{y}^\circ(t+T-1)$ have been found by the minimization of J_t, only the first values $\underline{u}^\circ(t)$ and $\underline{y}^\circ(t)$ are retained and applied; such a procedure is repeated at time $t+1$, and so on.

A possible approach to finding a solution to problem (5) is based on Branch-and-Bound Mixed-Integer NonLinear Programming techniques (we shall refer to such methodologies as BBMINLP) because of the binary variables that are involved in both the state equation and constraints. Thus, the search for a solution may be too computational demanding. As a matter of fact, we note that we can avoid dealing with the binary variables by using nonsmooth step functions in the constraints involving such variables. For example, if we refer to the single-mode model (1)–(3), we note that (2c) can be rewritten as follows:

$$x_6(t+1) = \begin{cases} 0 & \text{if} \quad x_6(t) + \Delta T \mu_5(t) u_5(t) = s(t) \quad \text{and} \quad x_1(t) = 0 \\ x_6(t) + \Delta T \mu_5(t) u_5(t) & \text{otherwise.} \end{cases}$$

Such a constraint can be expressed by the following relationship:

$$x_6(t+1) = [x_6(t) + \Delta T \mu_5(t) u_5(t)]\, \chi\, [s(t) - x_6(t) - \Delta T \mu_5(t) u_5(t) + x_1(t)]$$

where χ is the step function [i.e., $\chi(z) = 1$ if $z > 0$, $\chi(z) = 0$ otherwise]. By replacing the components of $\underline{y}(t)$ with the corresponding functions $\chi(\cdot)$ in all the constraints and cost function, the problem reduces to a nonlinear programming one with no integer variables but with nonsmooth functions; thus, we resort to Real NonLinear Programming techniques (RNLP, for short) to find a solution. Following such approach, the predictive control problem (5) reduces to searching for only the optimal inputs $\underline{u}^\circ(t),\ldots,\underline{u}^\circ(t+T-1)$.

Unfortunately, it may be quite demanding to find a solution of RNLP and particularly BBMINLP problems in a real-time context, especially with large predictive horizons and numerous control and state variables. An alternative method to construct a predictive control strategy can be derived from the RNLP approach. Since only the first optimal control input [i.e., $\underline{u}^\circ(t)$] is retained and applied, a different methodology is proposed that consists in approximating off line the optimal RNLP feedback control law $\underline{x}(t) \mapsto \underline{u}^\circ(t)$ by means of some approximator in order to generate the control action on line almost instantaneously. In particular, once T has been selected, we can (i) solve off line many optimization problems of type RNLP, (ii) collect the solution pairs given by state and optimal control vectors, and (iii) apply some training method to approximate such pairs. The optimal weights that result from (iii) are used to compute the control action on line. Toward this end, we shall employ

one-hidden-layer feedforward neural networks (OHLFFNNs) with sigmoidal activation functions in the hidden layer.

It is worth noting that the resulting approximate optimal control law may not satisfy some constraints exactly. However, to deal with this difficulty, some simple solutions can be devised. For example, if the inequality constraints (3) were not satisfied, we could normalize the outputs of the neural network in such a way to impose the satisfaction of such inequalities. Clearly, the more precise the approximation, the smaller the correction. Thus, in order to ensure a sufficient precision, neural networks of adequate complexity should be used.

4 Simulation Results

In this section, we present the simulation results that refer to a three-berth medium-size container terminal in northwest Italy (details can be found in [4]). The overall model is made up of a number of state, binary, and input variables equal to 28, 5, and 25, respectively. In the following, a description of the simulation setup and results is provided.

The interarrival times of ships were assumed to be exponentially distributed with a mean of 11 h; the quantity of containers to load/unload was taken according to a uniform distribution in the [60, 750] TEU range. The arrivals of trucks were generated according to a criterion that takes into consideration the existence of peaks at certain hours of the day. More specifically, we assumed that the number of containers to load/unload were uniformly distributed over different ranges given by [1, 6] TEU for "quiet" time periods and [50, 70] TEU for "busy" ones. The interarrival times of trains were taken according to an exponential distribution with mean equal to 2.5 h and standard deviation equal to 0.25 h; the quantity of containers to load/unload was taken with a uniform distribution in the [50, 70] TEU range. Concerning the sampling time, we chose $\Delta T = 0.5$ h, which is a sampling time consistent with the dynamics of the terminal and its operations, and allows one to obtain a compromise between effectiveness of the resulting control strategy and computational burden. Larger values of ΔT might be used with the advantage of having at disposal more time to perform the optimization, but this would compromise the capability of applying the proposed approaches to control the operations in real time. The handling capacities of the resources were generated as random variables with truncated Gaussian distributions. The means and standard deviations of such random variables were chosen to be equal to the following: 25 and 2.5 TEU/h for QCs; 20 and 6 TEU/h for RMGCs; 10 and 3 TEU/h for RSs; 17 and 5 TEU/h for RTGCs; 9 and 3 TEU/h for YTs. As regards the number of transfer resources, we used 5 QCs, 2 RMGCs, 15 RSs, 16 RTGCs, and 30 YTs.

The solutions of the various predictive control problems at $t = 0, 1, \ldots$ with a function like in (4) having coefficients equal to 100 for ships, 25 for trucks, and 60 for trains were found using either Matlab nonlinear programming techniques without using the derivatives of both the cost function and constraints according to the RNLP approach, or a standard branch-and-bound algorithm to perform

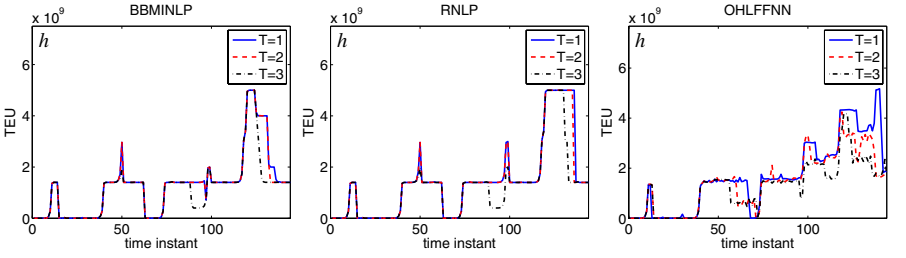


Fig. 2 Performances over time of BBMINLP, RNLP, and OHLFFNN predictive controllers

Table 1 Summary of the performances of BBMINLP, RNLP, and OHLFFNN controllers for the prediction horizons $T = 1$ and $T = 3$

	Control horizon	$T = 1$			$T = 3$		
	Carrier type	Ship	Truck	Train	Ship	Truck	Train
BBMINLP	TNSC	6	71	13	6	72	13
	NTC [TEU]	4096	4860	2110	4096	4866	2110
	MLTC [h]	13.2	0.33	3.76	12.5	0.35	3.90
	MOCT [min]	7.6			154.1		
RNLP	TNSC	6	71	13	6	72	13
	NTC [TEU]	4096	4860	2110	4096	4866	2110
	MLTC [h]	13.6	0.32	3.84	12.8	0.36	3.88
	MOCT [min]	0.16			3.5		
OHLFFNN	TNSC	6	66	13	6	67	13
	NTC [TEU]	4096	4746	2110	4096	4780	2110
	MLTC [h]	15.3	0.50	4.75	13.7	0.45	4.65
	MOCT [min]	0.0011			0.0011		

a BBMINLP optimization. As regards the neural approach, we used a OHLFFNN with 10 sigmoidal activation functions in the hidden layer. The neural training was done using the Levenberg-Marquardt algorithm available in Matlab with a data set made up of 1500 state-control pairs obtained by the RNLP approach.

The performances of the terminal were evaluated for different predictive horizons T (we chose $T = 1$, $T = 2$, and $T = 3$) and computing the total number of served carriers (TNSC), the number of transferred containers (NTC), and the mean lay times of carriers (MLTC) for $t = 0, 1, \ldots, 144$. The simulations were performed on a 3.2 GHz Pentium 4 PC with 1 GB of RAM.

As expected, the longer the control horizon T, the smaller the values of the function h resulting from the optimization, as pictorially shown in Figure 2 for the BBMINLP, RNLP, and OHLFFNN approaches. As shown in Table 1 (where MOCT stands for mean on-line computational time), BBMINLP provides the best results in terms of the final minimization cost, while RNLP demand a reduced computational burden with respect to BBMINLP. This reduction is payed with a small decay of performances. The OHLFFNN methodology performs worse than the other

approaches, but it has the great advantage of a very low on-line computational burden. Clearly, the neural network that provides the approximate optimal control law of the OHLFFNN approach has to be trained again if there are changes in the terminal configuration (e.g., number or capacity of resources and number of loading/unloading areas).

Note that, looking at Table 1, the use of predictive control to devise management strategies can be practically accomplished in real time only using the RNLP and OHLFFNN approaches because of their reduced computational burden. By contrast, the solution based on BBMINLP cannot be applied with success because of the too high computational effort that prevents it from being used on line. The RNLP approach appears to be a good compromise between efficiency and computational effort to determine it. However, note that the simulations were run in Matlab, thus the use of a more efficient development framework employing, e.g., C code, would give a considerable reduction of the on-line computational time.

References

1. Van Hee, K.M., Wijbrands, R.J.: Decision support system for container terminal planning. Eur. J. Oper. Res. 34, 262–272 (1988)
2. Gambardella, L.M., Mastrolilli, A., Rizzoli, A.E., Zaffalon, M.: An optimization methodology for intermodal terminal management. J. Intell. Manuf. 12(5-6), 521–534 (2001)
3. Alessandri, A., Sacone, S., Siri, S.: Modelling and optimal receding-horizon control of maritime container terminals. J. Math. Model. Algorithms 6(1), 109–133 (2007)
4. Alessandri, A., Cervellera, C., Cuneo, M., Gaggero, M., Soncin, G.: Modeling and feedback control for resource allocation and performance analysis in container terminals. IEEE Trans. Intell. Transp. Syst. 9(4), 601–614 (2008)
5. Alessandri, A., Cervellera, C., Cuneo, M., Gaggero, M., Soncin, G.: Management of logistics operations in intermodal terminals by using dynamic modelling and nonlinear programming. Maritime Economics and Logistics 11(1), 58–76 (2009)
6. Barron, A.: Universal approximation bounds for superpositions of a sigmoidal function. IEEE Trans. Inf. Theory 39, 930–945 (1993)
7. Zoppoli, R., Sanguineti, M., Parisini, T.: Approximating networks and extended Ritz method for the solution of functional optimization problems. J. Optim. Theory Appl. 112(2), 403–439 (2002)

Predictive Power Control of Wireless Sensor Networks for Closed Loop Control

Daniel E. Quevedo, Anders Ahlén, and Graham C. Goodwin

Abstract. We study a networked control architecture where wireless sensors are used to measure and transmit plant outputs to a remote controller. Packet loss probabilities depend upon the time-varying communication channel gains and the transmission powers of the sensors. Within this context, we develop a centralized stochastic nonlinear model predictive controller. It determines the sensor power levels by trading energy expenditure for expected plant state variance. To further preserve sensor energies, the power controller sends coarsely quantized power increment commands only when necessary. Simulations on measured channel data illustrate the performance achieved by the proposed controller.

Keywords: Wireless Sensor Networks, Predictive Control.

1 Introduction

Wireless Sensor Networks (WSNs) are becoming an interesting alternative for closed loop control [1, 2]. WSNs can be placed where wires cannot go and where power sockets are not available. A drawback of using WSNs is that channel fading and interference may lead to packet errors and, thus, performance degradation. Whilst communication reliability and, thus, control accuracy, can certainly be improved by

Daniel E. Quevedo and Graham C. Goodwin
School of Electrical Engineering and Computer Science, The University of Newcastle, NSW 2308, Australia
e-mail: dquevedo@ieee.org, graham.goodwin@newcastle.edu.au

Anders Ahlén
Department of Engineering Sciences, Signals and Systems, Uppsala University, P.O. Box 534 SE-751 21, Uppsala, Sweden
e-mail: Anders.Ahlen@signal.uu.se

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 215–224.
springerlink.com

increasing transmission power levels, saving energy in WSNs is uppermost to avoid unnecessary maintenance, such as the replacement of batteries, see also [3, 4].

In the present work we examine a Networked Control System (NCS) architecture where sensor measurements are sent over wireless fading channels. In contrast to other approaches, see, e.g., [5], in the topology studied here, sensors do not communicate with each other. Instead, sensor measurements are sent to a single gateway for state estimation and subsequent plant input calculation. In addition, the gateway decides upon the power levels to be used by the sensors. Within this setting, we show how the sensor power levels can be designed via nonlinear predictive control. The proposed controller trades expected plant state variance for energy expenditure. The present work extends our recent conference contribution [6] (on state estimation) to NCS's.

2 WSNs for Networked Control

We consider an LTI n-dimensional plant with input $\{\mu(k)\}_{k\in\mathbb{N}_0}$:

$$x(k+1) = Ax(k) + B\mu(k) + w(k), \quad k \in \mathbb{N}_0 \triangleq \{0,1,\dots\}, \tag{1}$$

where the initial state is Gaussian distributed with mean x_0 and covariance $P_0 \in \mathbb{R}^{n\times n}$, i.e., $x(0) \in \mathcal{N}(x_0, P_0)$. Similarly, the driving noise process $w = \{w(k)\}_{k\in N_0}$ is i.i.d., where each $w(k) \in \mathcal{N}(0, Q)$.

A collection of M sensors is used to measure and transmit plant output information via wireless links to a single gateway. Each sensor provides a noisy measurement signal, say $\{y_m(k)\}_{k\in N_0}$:

$$y_m(k) = C_m x(k) + v_m(k), \quad m \in \{1,2,\dots,M\}, \tag{2}$$

and where $v = \{v_m(k)\}_{k\in N_0}$ is an i.i.d. process with $v_m(k) \in \mathcal{N}(0, R_m)$.[1]

The signals received at the gateway are then used to calculate the control input μ. The aim is to steer the system state $x(0)$ to the origin. In the present work, we will assume that the associated control policy has already been designed and is given by linear state estimate feedback:

$$\mu(k) = -L(k)\hat{x}(k), \quad k \in \mathbb{N}_0 \triangleq \{0,1,\dots\}, \tag{3}$$

where $\hat{x}(k)$ is an estimate of $x(k)$ and where $L(k)$ are given state feedback matrices of appropriate dimensions. Thus, the gateway needs to remotely estimate the state of the system (1). The situation is depicted in Fig. 1 for a networked control system (NCS) having $M = 2$ wireless sensors.

The distinguishing aspect of the situation at hand is that, since the M links between sensors and gateway are wireless, transmission errors are likely to occur.

[1] In addition to measurement noise, v may also describe quantization effects.

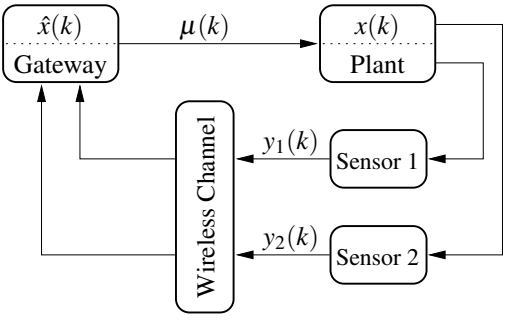


Fig. 1 Networked Control with a WSN having $M = 2$ sensors

This leads to loss of packets and control performance degradation.[2] Packet loss probabilities depend upon the time-varying channel gains and upon the transmission power used by the sensors, higher power providing less transmission errors. However, in wireless sensor networks, it is of fundamental importance to save energy: Sensor nodes are expected to be operational for several years without maintenance. Thus, available energy resources have to be used with care. The main purpose of the present work lies in designing a centralized predictive power controller for the WSN used in the NCS of Fig. 1. Before presenting our proposal, we will first set the work in context by briefly elucidating the trade-off between power use and control accuracy.

3 Trading Energy Use for Control Performance

To describe the interplay between energy consumption and transmission reliability, we quantify the energy used by each sensor m at time $k \in \mathbb{N}_0$ via:

$$g_m(u_m(k)) \triangleq \begin{cases} \dfrac{u_m(k) b_m}{r} + E_{\mathrm{P}} & \text{if } u_m(k) > 0, \\ 0 & \text{if } u_m(k) = 0. \end{cases} \tag{4}$$

Here, $u_m(k)$ is the transmission power used by the m-th sensor radio power amplifier, b_m is the number of bits used per measurement value $y_m(k)$, r is bit-rate of the channels and E_{P} is the total energy needed (per measurement value) for power-up, sensing and circuitry. As we will see in Section 5, the choice of b_m depends on the required accuracy and the energy available. A large value of b_m will lead to an improved accuracy, but at the expense of a higher energy expenditure and an increased probability of packet error. In general, the number of bits per transmitted packet will be governed by the protocol used. In the present work, we focus on a

[2] We will assume that sensor data is not affected by delays. Extensions to include time-delay issues, and also irregular sampling, does not present conceptual difficulties, provided sensor data are time-stamped.

simple scheme, where one measurement is transmitted at a time.[3] The selected bit rate, r, will thus depend on the application as will the number of channels used.

Due to physical limitations of the radio power amplifiers, we assume that the power levels are constrained in magnitude according to:

$$0 \leq u_m(k) \leq u_m^{\max}, \quad \forall k \in \mathbb{N}_0, \quad \forall m \in \{1,2,\ldots,M\}, \tag{5}$$

for given values $\{u_m^{\max}\}$. Thus, the maximum transmission energy per measurement value at each node is given by:

$$(E_{\mathrm{TX}}^{\max})_m \triangleq (b_m/r)u_m^{\max}, \quad m \in \{1,2,\ldots,M\}. \tag{6}$$

We model transmission effects by introducing the M binary stochastic arrival processes $\{\gamma_m(k)\}_{k\in N_0}$, $m \in \{1,2,\ldots,M\}$, where:

$$\gamma_m(k) = \begin{cases} 1 & \text{if } y_m(k) \text{ arrives error-free at time } k, \\ 0 & \text{if } y_m(k) \text{ does not arrive error-free at time } k. \end{cases} \tag{7}$$

The associated success probabilities satisfy

$$\mathscr{P}\{\gamma_m(k) = 1\} = f_m\big(u_m(k)h_m(k)\big), \quad m \in \{1,2,\ldots,M\}, \tag{8}$$

where $f_m(\cdot)\colon [0,\infty) \to [0,1]$ is a monotonically increasing function, which depends upon the communication scheme employed, and where $h_m(k)$ denotes the square of the magnitude of the channel gain.

To calculate the plant input $\mu(k)$ in (3), the gateway needs to obtain plant state estimates. Here, we will assume that the data transmitted incorporates error detection coding [7]. Hence, at any time k, past and present realizations of the transmission processes (7), say

$$\gamma^k \triangleq \bigcup_{m\in\{1,2,\ldots,M\}} \Big\{\gamma_m(0), \gamma_m(1), \ldots, \gamma_m(k)\Big\} \tag{9}$$

are available at the gateway. Faulty packets will be discarded when estimating the system state. This amounts to sampling (1)-(2) only at the successful transmission instants of each sensor link. Indeed, the conditional probability distribution of the system state at any time k, given x_0, P_0, γ^k and correctly received sensor measurements up to time k, say y^k, is Gaussian. The conditional mean and covariance[4] of the state, i.e.,

$$\hat{x}(k) \triangleq \mathscr{E}_{w,v,x(0)}\Big\{x(k)\big|\, y^k, \gamma^k\Big\}$$
$$\bar{P}(k) \triangleq \mathscr{E}_{w,v,x(0)}\Big\{\big(\hat{x}(k)-x(k)\big)\big(\hat{x}(k)-x(k)\big)^T \big|\, y^k, \gamma^k\Big\},$$

[3] Alternatively, one could also aggregate measurements.

[4] Here, $\mathscr{E}_{w,v,x(0)}$ denotes expectation taken w.r.t. the noise sequences w and v and the initial state $x(0)$.

satisfy the Kalman Filter recursions (see, e.g., [8]):

$$\begin{aligned}\hat{x}(k+1) &= A\hat{x}(k) + B\mu(k) + K(k+1)\big(y(k+1) - C(k+1)(A\hat{x}(k) + B\mu(k))\big)\\ P(k+1) &= AP(k)A^T + Q - AK(k)C(k)P(k)A^T\\ \bar{P}(k) &= P(k) - K(k)C(k)P(k),\end{aligned} \tag{10}$$

with initial values $P(0) = P_0$ and $\hat{x}(0) = x_0$ and where:[5]

$$\begin{aligned}C(k) &\triangleq \left[\gamma_1(k)(C_1)^T \;\; \gamma_2(k)(C_2)^T \;\; \ldots \;\; \gamma_M(k)(C_M)^T\right]^T\\ K(k) &\triangleq P(k)C(k)^T\left(C(k)P(k)C(k)^T + R\right)^{-1}\\ R &\triangleq \mathrm{diag}\big(R_1, R_2, \ldots, R_M\big).\end{aligned} \tag{11}$$

The state estimate in (10) is used to calculate the plant input, see (3). The controlled plant (1)-(3) is, thus, described via:

$$x(k+1) = \bar{A}(k)x(k) + e(k) + w(k),$$

where $\bar{A}(k) \triangleq A - BL(k)$, whereas

$$e(k) \triangleq BL(k)(x(k) - \hat{x}(k)) \tag{12}$$

denotes the effect of the state estimation error on the successor plant state.

We note that $e(k)$ in (12) depends upon the transmission processes $\gamma_m(k)$ through the matrices $C(k)$ in the state estimate $\hat{x}(k)$, see (10) and (11). As seen in (8), transmission reliability can be improved by using larger power levels $u_m(k)$, however, this occurs at the expense of more energy consumption, see (4). This trade-off between energy consumption at the sensors and resulting control accuracy forms the background to the power control scheme proposed in the following section.

4 Predictive Power Control

In the NCS architecture under study, the gateway not only calculates the plant inputs, but also determines the power levels to be used by the sensors. For that purpose, the gateway is equipped with a model predictive controller which trades energy consumption for control quality over a future prediction horizon. Power control signals are sent over wireless links to the sensors.[6]

Power Control Signal Coding

To keep processing and associated power consumption at the sensors to a minimum, in our approach the power control signals have short word-lengths. Here, we will use

[5] Properties of this (and related) estimators have been studied, e.g., in [9, 10, 11].

[6] At the gateway saving energy is of less importance than at the sensors. We, thus, assume that communication from the gateway to the sensors is error-free.

coding ideas frequently used in power control architectures for cellular networks, see, e.g., [12] (and compare also to our work on NCS's in [13]) and send coarsely quantized power increments, say $\Delta u_m(k)$, rather than actual power values, $u_m(k)$, to each sensor $m \in \{1,2,\ldots,M\}$. We thus have:

$$\Delta u_m(k) \in \mathbb{U}_m, \quad \forall k \in \mathbb{N}_0, \quad \forall m \in \{1,2,\ldots,M\}, \tag{13}$$

where $\{\mathbb{U}_m\}$ are given finite sets, each having a small number of elements.

Upon reception of $\Delta u_m(k)$, each sensor m reconstructs the power level to be used by its radio power amplifier by simply setting

$$u_m(k) = u_m(k-1) + \Delta u_m(k). \tag{14}$$

For further reference, we define the signal

$$\Delta u(k) \triangleq \left[\Delta u_1(k) \ \ldots \ \Delta u_M(k)\right]^T, \quad k \in \mathbb{N}_0 \tag{15}$$

and note that the quantization constraint (13) imposes:

$$\Delta u(k) \in \mathbb{U} \triangleq \mathbb{U}_1 \times \mathbb{U}_2 \times \cdots \times \mathbb{U}_M, \quad \forall k \in \mathbb{N}_0.$$

Predictive Power Controller

At every time instant $k \in \mathbb{N}_0$, the predictive power controller first calculates $\bar{P}(k)$, which results from iterating (10) for the (known) past arrival process realizations γ^k, see (9). It also obtains channel gain predictions over a finite horizon of fixed length N, namely:

$$\{\hat{h}_m(k+1|k), \hat{h}_m(k+2|k), \ldots, \hat{h}_m(k+N|k)\}, \quad \forall m \in \{1,2,\ldots,M\},$$

which can be estimated by using previous channel estimates, see, e.g., [14, 15]. Given this information, the controller minimizes the finite-set constrained cost[7]

$$J(\Delta U) \triangleq \sum_{\ell=k+1}^{k+N} \left\{ \mathscr{E}_{\Gamma(k)} \left\{ \mathrm{trace}\left(\Sigma'(\ell)\right) \right\} + \rho \sum_{m=1}^{M} g_m(u'_m(\ell)) \right\}, \tag{16}$$

where $g_m(u'_m(\ell))$ is as in (4) and where

$$\begin{aligned} \Sigma'(\ell) &\triangleq \mathscr{E}_{w,v,x(0)} \left\{ e'(\ell) e'(\ell)^T \,\middle|\, y^k, \gamma^k \right\} = BL(\ell)\bar{P}'(\ell)L(\ell)^T B^T, \\ \Delta U &\triangleq \{\Delta u'(k+1), \Delta u'(k+2), \ldots, \Delta u'(k+N)\}, \end{aligned} \tag{17}$$

see (10) and (12). The scalar $\rho \geq 0$ is a design parameter which allows one to trade control accuracy for energy consumption. The expectation operator $\mathscr{E}_{\Gamma(k)}$ is taken with respect to the distribution of future transmission outcomes in[8]

[7] Primed variables refer to tentative values of the corresponding physical variables.

[8] Compare to the scenario based approaches taken in [16].

$$\Gamma(k) \triangleq \left[\gamma(k+1)\ \gamma(k+2)\ \ldots\ \gamma(k+N)\right]^T.$$

This distribution depends upon the power level, see (8). The decision variables, i.e., the tentative future power value increments, are collected in ΔU, see (15) and (17). These determine the tentative future power levels $u'_m(\ell)$ in (16) via

$$u'_m(\ell) = u'_m(\ell-1) + \Delta u'_m(\ell), \quad \ell \in \{k+1,\ldots,k+N\},\ m \in \{1,2,\ldots,M\},$$

starting from the current values, $u'_m(k) = u_m(k)$, $\forall m \in \{1,2,\ldots,M\}$.

Minimization of (16) subject to the constraints:

$$\Delta U \in \mathbb{U}^N \triangleq \mathbb{U} \times \mathbb{U} \times \cdots \times \mathbb{U}$$
$$0 \le u'_m(\ell) \le u_m^{\max}, \quad \forall \ell \in \{k+1,\ldots,k+N\},\ \forall m \in \{1,2,\ldots,M\}$$

gives the sequence of control increments:

$$\Delta U^{\text{opt}} \triangleq \arg\min J(\Delta U). \tag{18}$$

Following the moving horizon principle, only the M power updates in[9]

$$\Delta u(k+1)^{\text{opt}} \triangleq \left[I_M\ 0_M\ \ldots\ 0_M\right] \Delta U^{\text{opt}}$$

are sent to the corresponding sensors. At the next time step, namely $k+1$, the optimization procedure is repeated, giving rise to power level increments $\Delta u(k+2)^{\text{opt}}$. This procedure is repeated *ad infinitum*.

We emphasize that, despite (18) being a stochastic nonlinear programme, ΔU^{opt} can be found via simple exhaustive search over the 2^{MN} possible transmission scenarios $\Gamma(k)$ and, in a worst complexity case, $|\mathbb{U}|^N$ values of ΔU.

The proposed controller jointly decides upon the power levels of all M sensors to achieve the best trade-off between energy use and control accuracy.

5 Simulation Study

To illustrate basic features of the model predictive power controller presented in the previous section, we consider a NCS as in Fig. 1 with $M = 2$ channels and use measured channel data. Measurements were acquired in the 2.4 GHz ISM band in an office area at the Signals and Systems group at Uppsala University.

We assume that Binary Phase Shift Keying is employed and use the transmission error model [7]:

$$f_m(u_m(k)h_m(k)) = \begin{cases} \left(1 - \tilde{Q}\left(\sqrt{2u_m(k)h_m(k)/(rk_BT)}\right)\right)^{b_m} & \text{if } u_m(k) > 0 \\ 0 & \text{if } u_m(k) = 0, \end{cases}$$

[9] I_M denotes the $M \times M$ identity matrix and 0_M the all zeros $M \times M$ matrix.

where $\tilde{Q}(z) \triangleq 1/\sqrt{2\pi}\int_z^\infty exp(-v^2/2)dv$ is the Q-function, k_B is the Boltzmann constant and $T = 300\,[K]$ is the temperature. The selected bit-rate of the channels is $r = 40\,[kbits/s]$, and the number of bits per measurement value are $b_1 = b_2 = 8\,[bits]$.

The (unstable) plant is characterized via the model of Section 2 with:

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, P_0 = \frac{1}{3}I_2, Q = \frac{1}{2}I_2, \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = I_2, R_1 = R_2 = \frac{1}{100}.$$

The plant input in (3) is provided by the LQG policy: $\mu(k) = -\begin{bmatrix} 1.452 & 0.898 \end{bmatrix} \hat{x}(k)$.

The constraints on the power values, see (5), are $u_1^{\max} = u_2^{\max} = 0.5\,[mW]$. The power controller parameters are chosen as $N = 1$, $\rho = 10\,[1/\mu J]$. Power increments are restricted to belong to the finite sets $\mathbb{U}_1 = \mathbb{U}_2 = \{0, \pm 50\}\,[\mu W]$. We assume that the gateway has perfect one-step-ahead channel predictions.

To investigate the impact of different kinds of sensor nodes, we introduce the energy ratio $\eta \triangleq E_{\mathrm{P}}/E_{\mathrm{TX}}^{\max}$, see (4) and (6).

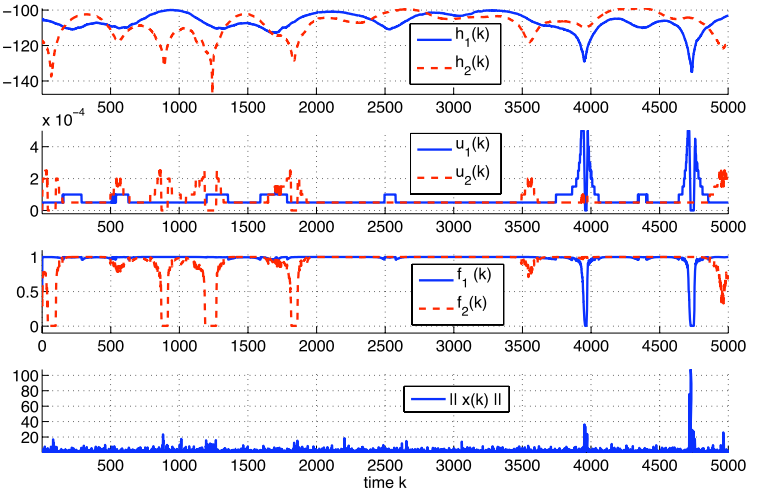


Fig. 2 System performance for sensor nodes with energy ratio $\eta = 0.5$

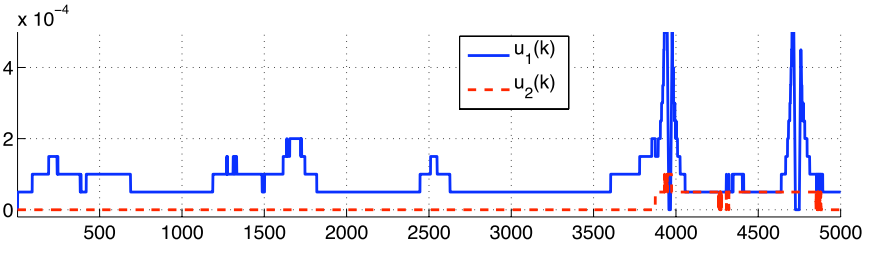


Fig. 3 Power levels for sensor nodes with energy ratio $\eta = 1.7$

Fig. 2 illustrates that the proposed predictive controller tries to find the best compromise between the two sensor links. In particular, the controller at times approximately inverts the channel gains. At other times, it decides to send one of the sensors to sleep, i.e., to set $u_1(k) = 0$ or $u_2(k) = 0$. This scheduling aspect is even more apparent in Fig. 3, as a consequence of a higher value of η. It is important to notice that, in the present case, the first component of $x(k)$ is more important than the second component. Consequently, the predictive controller favors Sensor 1 over Sensor 2.

If power levels would be kept constant, such that the same total amount of energy is used, then, for $\eta = 0.5$, the trace of the empirical covariance of the state would be 25% larger. For $\eta = 1.7$, no such constant power levels exist, since E_P is too large.

6 Conclusions

We have developed a stochastic nonlinear model predictive power controller for wireless sensor networks used within a networked control system. Due to (time-varying) fading on the wireless channels, transmission errors of sensor measurements are likely to occur. The proposed controller trades energy expenditure at the sensors for expected plant state variance to compensate for channel fading in an optimal manner, as illustrated on measured channel data. Key to keeping the computational burden limited is the fact that the occurrence of transmission errors constitutes a binary random variable. Thus, expectations can be exactly evaluated via finite sums, i.e., no integrals need to be evaluated or approximated.

References

1. Antsaklis, P., Baillieul, J.: Special issue on technology of networked control systems. Proc. IEEE 95, 5–8 (2007)
2. Shen, X., Zhang, Q., Caiming Qiu, R.: Wireless sensor networking [guest ed.]. IEEE Wireless Commun 14, 4–5 (2007)
3. Xiao, J.-J., Ribeiro, A., Luo, Z.-Q., Giannakis, G.B.: Distributed compression-estimation using wireless sensor networks. IEEE Trans. Signal Processing 23, 27–41 (2006)
4. Luo, X., Giannakis, G.B.: Energy-constrained optimal quantization for wireless sensor networks. In: EURASIP J. Adv. Signal Processing (2008)
5. Speranzon, A., Fischione, C., Johansson, K.H., Sangiovanni-Vincentelli, A.: A distributed minimum variance estimator for sensor networks. IEEE J. Select. Areas Commun. 26, 609–621 (2008)
6. Quevedo, D.E., Ahlén, A.: A predictive power control scheme for energy efficient state estimation via wireless sensor networks. In: Proc. IEEE Conf. Decis. Contr. (2008)
7. Proakis, J.G.: Digital Communications, 3rd edn. McGraw-Hill, New York (1995)
8. Anderson, B.D.O., Moore, J.: Optimal Filtering. Prentice-Hall, Englewood Cliffs (1979)
9. Sinopoli, B., Schenato, L., Franceschetti, M., Poolla, K., Jordan, M.I., Sastry, S.S.: Kalman filtering with intermittent observations. IEEE Trans. Automat. Contr. 49, 1453–1464 (2004)
10. Huang, M., Dey, S.: Stability of Kalman filtering with Markovian packet losses. Automatica 43, 598–607 (2007)

11. Liu, X., Goldsmith, A.: Kalman filtering with partial observation losses. In: Proc. IEEE Conf. Decis. Contr. Paradise Island, Bahamas, pp. 4180–4186 (2004)
12. Holma, H., Toskala, A. (eds.): WCDMA for UMTS, 3rd edn. John Wiley & Sons, West Sussex (2004)
13. Goodwin, G.C., Haimovich, H., Quevedo, D.E., Welsh, J.S.: A moving horizon approach to networked control system design. IEEE Trans. Automat. Contr. 49, 1427–1445 (2004)
14. Ekman, T., Sternad, M., Ahlén, A.: Unbiased power prediction of Rayleigh fading channels. In: Proc. Vehicular Technology Conf., vol. 1, pp. 280–284 (2002)
15. Ekman, T.: Prediction of Mobile Radio Channels: Modeling and Design. PhD thesis, Uppsala University (October 2002)
16. Quevedo, D.E., Silva, E.I., Goodwin, G.C.: Control over unreliable networks affected by packet erasures and variable transmission delays. IEEE J. Select. Areas Commun. 26, 672–685 (2008)

On Polytopic Approximations of Systems with Time-Varying Input Delays

Rob Gielen, Sorin Olaru, and Mircea Lazar

Abstract. Networked control systems (NCS) have recently received an increasing attention from the control systems community. One of the major problems in NCS is how to model the highly nonlinear terms caused by uncertain delays such as time-varying input delays. A straightforward solution is to employ polytopic approximations. In this paper we develop a novel method for creating discrete-time models for systems with time-varying input delays based on polytopic approximations. The proposed method is compared to several other existing approaches in terms of quality, complexity and scalability. Furthermore, its suitability for model predictive control is demonstrated.

Keywords: input delay, networked control systems, polytopic uncertainty.

1 Introduction

Recently networked control systems (NCS) have become one of the topics in control that receives a continuously increasing attention. This is due to the important role that transmission and propagation delay play in nowadays modern control applications. In [8], a survey on future directions in control, NCS where even indicated to be one of the emerging key topics in control. In NCS the connection between plant and controller is a network that is in general shared with other applications. The motivations for using NCS are mostly cost and efficiency related. In [6, 10] a comprehensive overview of the main difficulties within NCS and the recent developments in this field is given. Both papers present different setups and solutions

Rob Gielen and Mircea Lazar
Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands
e-mail: r.h.gielen@tue.nl, m.lazar@tue.nl

Sorin Olaru
Automatic Control Department, SUPELEC, Gif-sur-Yvette, France
e-mail: sorin.olaru@supelec.fr

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 225–233.
springerlink.com

for stabilizing controller design. In general two main issues can be distinguished: time-varying input delay and data packet dropout. The present paper focusses on the first issue. A general study on stability of NCS can be found in [1] and the references therein. More recently, in [3, 5] the problem of time-varying input delay was reformulated as a robust control problem and a static feedback controller was then synthesized by means of linear matrix inequalities (LMI). In [9], also by means of LMI, a model predictive control (MPC) scheme has been designed for systems with time-varying input delay.

One of the biggest challenges in stabilization and predictive control of NCS is to find a modeling framework that can handle time-varying input delays effectively. One of the most popular solutions to this problem, already employed in [3, 5, 9], is to model the delay-induced nonlinearity using a polytopic approximation. The advantage of this approach is that the resulting model is a linear parameter varying system for which efficient stabilization methods and control design techniques exist, see, for example, [7]. This paper proposes a new approach for deriving a polytopic approximation, based on the Cayley-Hamilton theorem. The method is compared with the above-mentioned techniques in terms of scalability, complexity and conservativeness. The suitability of all methods for predictive control is analyzed using the MPC strategy of [7].

2 Preliminaries

2.1 Basic Notation and Definitions

Let $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{Z}$ and $\mathbb{Z}_+$ denote the field of real numbers, the set of non-negative reals, the set of integers and the set of non-negative integers respectively. We use the notation $\mathbb{Z}_{\geq c_1}$ and $\mathbb{Z}_{(c_1,c_2]}$ to denote the sets $\{k \in \mathbb{Z}_+ | k \geq c_1\}$ and $\{k \in \mathbb{Z}_+ | c_1 < k \leq c_2\}$, respectively, for some $c_1, c_2 \in \mathbb{Z}_+$. A polyhedron, or a polyhedral set, in $\mathbb{R}^n$ is a set obtained as the intersection of a finite number of open and/or closed half-spaces, a polytope is a compact polyhedron. Let $Co(\cdot)$ denote the convex hull. Let $\|A\|_2 := \sup_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2}$ denote the induced matrix 2-norm. A well-known property is that $\|A\|_2^2 = \lambda_{max}(A^T A)$, where $\lambda_{max}(M)$ is the largest eigenvalue of $M \in \mathbb{R}^{n \times n}$.

2.2 Problem Definition

Consider the continuous time system with input delay

$$\begin{aligned} \dot{x}(t) &= A_c x(t) + B_c u(t) \\ u(t) = u_k, \quad \forall t \in [t_k + \tau_k, t_{k+1} + \tau_{k+1}] \quad &\text{and} \quad u(t) = u_{initial}, \quad \forall t \in [0, \tau_0], \end{aligned} \tag{1}$$

where $t_k = kT_s$, $k \in \mathbb{Z}_+$ and $T_s \in \mathbb{R}_+$ is the sampling time. Furthermore $A_c \in \mathbb{R}^{n \times n}$, $B_c \in \mathbb{R}^{n \times m}$, $\tau_k \in \mathbb{R}_{[0,T_s)}$, $\forall k \in \mathbb{Z}_+$ is the network delay, $u_k \in \mathbb{R}^m$, $k \in \mathbb{Z}_+$ is the control action generated at $t = t_k$, $u(t) \in \mathbb{R}^m$ is the system input and $x(t) \in \mathbb{R}^n$ is the system state. The time-varying delay that affects the input signal is one of the

most important aspects of NCS. As in NCS the controller only has discrete time information, we employ next several algebraic manipulations to obtain a discrete time description of the system, i.e.

$$x_{k+1} = e^{A_c T_s} x_k + \int_0^{\tau_k} e^{A_c(T_s-\theta)} d\theta B_c u_{k-1} + \int_{\tau_k}^{T_s} e^{A_c(T_s-\theta)} d\theta B_c u_k. \tag{2}$$

The goal is to design a stabilizing controller that is robust in the presence of time-varying delays. However this is a highly nonlinear system and is in general not suitable for controller synthesis. To obtain a model more suitable for control design define

$$\Delta_k := \int_0^{\tau_k} e^{A_c(T_s-\theta)} d\theta B_c, \quad k \in \mathbb{Z}_+. \tag{3}$$

Furthermore, by manipulating (2) and introducing a new augmented state vector of the form $\xi_k^T = [x_k^T \; u_{k-1}^T]$ we obtain:

$$\xi_{k+1} = A(\Delta_k)\xi_k + B(\Delta_k)u_k, \tag{4}$$

with $A(\Delta_k) := \left[\begin{smallmatrix} A_d & \Delta_k \\ 0 & 0 \end{smallmatrix}\right]$, $B(\Delta_k) := \left[\begin{smallmatrix} B_d - \Delta_k \\ I_m \end{smallmatrix}\right]$, $B_d = \int_0^{T_s} e^{A_c(T_s-\theta)} d\theta B_c$ and $A_d = e^{A_c T_s}$. Here (4) is a *nonlinear parameter varying* system with unknown parameter τ_k. The challenge that remains is to find a polytopic approximation of this nonlinear uncertainty in order to reformulate (4) into a *linear parameter varying* system with unknown parameter Δ_k. To achieve this we define the following set of matrices:

$$\Delta := Co(\{\overline{\Delta}_l\}), \quad \overline{\Delta}_l \in \mathbb{R}^{n\times m}, \quad l \in \mathbb{Z}_{[0,L]}, \quad L \in \mathbb{Z}_+, \tag{5}$$

such that $\Delta_k \in \Delta, \forall \tau_k \in [0, \overline{\tau}]$, where $\overline{\tau}$ is the maximum input delay that can be introduced by the network. This is a model that can be handled by most robust control techniques, including MPC, as it will be shown later.

2.3 Existing Solutions

In [3, 5, 9] several methods for finding the generators of the set in (5) were derived. Here these methods are only explained briefly to obtain a self-contained assessment; further details and proofs can be found in the corresponding articles.

In [3] and the references therein an elementwise maximization is proposed where $\overline{\Delta}_l$ contain all possible combinations of maxima and minima for all entries of Δ_k. This approach will be referred to as the ME method.

Other methods, as the ones in [3] and [9], are based on the Jordan normal form (JNF), i.e. $A_c = VJV^{-1}$ with J block diagonal. Starting from (3), with a mild assumption on A_c and using the JNF yields:

$$\Delta_k = \sum_{i=1}^{n} A_c^{-1} V (e^{J_i T_s} - e^{J_i(T_s-\tau_k)}) V^{-1} B_c. \tag{6}$$

Filling in $\tau_k = 0$ and $\tau_k = \overline{\tau}$ gives the generators of the set Δ. The two papers differ in so far that in [9] a method is proposed to reduce the number of generators at the cost of a larger polytope. The method as presented in [3] will be referred to as JNF1 and the method from [9] as JNF2.

Another option was proposed in [5], which makes use of a Taylor series expansion of (3), i.e.:

$$\Delta_k = \left(-\sum_{i=1}^{\infty} \frac{(-\tau_k)^i}{i!} A_c^{i-1} e^{A_c T_s} \right) B_c. \tag{7}$$

The generators of Δ are also obtained for $\tau_k = 0$ and $\tau_k = \overline{\tau}$. The infinite sum is approximated by a finite number of terms p, which is also the number of generators for Δ, i.e. $L = p$. This method will be referred to as TA. Next we present a novel method for finding the generators of the set Δ.

3 Main Result

The method presented in this paper is based upon the Cayley-Hamilton theorem.

Theorem 1 (Cayley-Hamilton theorem). *If $p(\lambda) := \det(\lambda I_n - A)$ is the characteristic polynomial of a matrix $A \in \mathbb{R}^{n \times n}$ then $p(A) = 0$.*

The original proof of this theorem can be found in [2] and further details on the theorem are given in [4]. Using this theorem it is possible to express all powers of A of order n and higher as a combination of the first n powers, i.e.

$$A^i = c_{i,0} I + \ldots + c_{i,n-1} A^{n-1}, \quad \forall i \in \mathbb{Z}_{\geq n}, \tag{8}$$

for some $c_{i,j} \in \mathbb{R}$, $j = 0, \ldots, n-1$. Define now the functions

$$f_j(T_s - \theta) := \sum_{i=0}^{\infty} a_{i,j} (T_s - \theta)^i, \tag{9}$$

where $a_{i,j} := \frac{c_{i,j}}{i!}$. By Theorem 1 we can derive the following expression for Δ_k.

Lemma 1. *Let*

$$g_j(\tau_k) := \int_0^{\tau_k} f_j(T_s - \theta) d\theta, \tag{10}$$

for some $c_{i,j} \in \mathbb{R}$ and $f_j(T_s - \theta)$ as in (8) and (9). Then

$$\Delta_k = \sum_{j=0}^{n-1} g_j(\tau_k) A_c^j B_c. \tag{11}$$

Proof: Starting from (3) and using (8) we obtain:

$$\begin{aligned}\Delta_k &= \int_0^{\tau_k} \sum_{k=0}^{\infty} \frac{(T_s-\theta)^k}{k!} A_c^k d\theta B_c \\ &= \int_0^{\tau_k} \left(I_n + A_c(T_s-\theta) + \ldots + A_c^n \frac{(T_s-\theta)^n}{n!} + \ldots \right) d\theta B_c \\ &= \int_0^{\tau_k} \Big(I_n + A_c(T_s-\theta) + \ldots \\ &\quad \ldots + (c_{n,0} I_n + \ldots + c_{n,n-1} A_c^{n-1}) \frac{(T_s-\theta)^n}{n!} + \ldots \Big) d\theta B_c. \end{aligned} \tag{12}$$

Gathering all terms before the same matrices, writing them as a function of $T_s - \theta$ and using (9) yields:

$$\Delta_k = \int_0^{\tau_k} \left(f_0(T_s-\theta) I_n + \ldots + f_{n-1}(T_s-\theta) A_c^{n-1} \right) d\theta B_c, \tag{13}$$

which concludes the proof. □

Filling in the corresponding values for τ_k in $g_j(\tau_k)$ gives $g_{j,l} \in \mathbb{R}$ and $g_{j,u} \in \mathbb{R}$ such that $g_{j,l} \leq g_j(\tau_k) \leq g_{j,u}, \forall \tau_k \in [0, \overline{\tau}]$. By Lemma 1 it is possible to write all realizations of Δ_k as a convex combination of a finite number of matrices $\overline{\Delta}_l$, as stated in the next theorem.

Theorem 2. *For any $\tau_k \in [0, \overline{\tau}]$, Δ_k satisfies:*

$$\Delta_k \in Co(n\Delta_0, \ldots, n\Delta_{2n-1}), \tag{14}$$

where

$$\Delta_j := g_{j,l} A_c^j B_c, \quad \Delta_{j+n} := g_{j,u} A_c^j B_c, \quad \forall j = 0, \ldots, n-1. \tag{15}$$

Proof: Starting from Lemma 1, for any $\tau_k \in [0, \overline{\tau}]$ and $g_j(\tau_k)$ there exists a $\nu_j \in \mathbb{R}_{[0,1]}$ and $\mu_j = 1 - \nu_j$ such that:

$$\begin{aligned}\Delta_k &= \left(g_0(\tau_k) I_n + g_1(\tau_k) A_c + \ldots + g_{n-1}(\tau_k) A_c^{n-1} \right) B_c \\ &= \left((\nu_0 g_{0,l} + \mu_0 g_{0,u}) I_n + \ldots + (\nu_{n-1} g_{n-1,l} + \mu_{n-1} g_{n-1,u}) A_c^{n-1} \right) B_c, \\ &= \left(\sum_{j=0}^{n-1} \left(\frac{\nu_j}{n} n g_{j,l} + \frac{\mu_j}{n} n g_{j,u} \right) A_c^j \right) B_c, \\ &= \left(\delta_0 n g_{1,l} I + \delta_n n g_{1,u} I + \ldots + \delta_{n-1} n g_{n-1,l} A_c^{n-1} + \delta_{2n-1} n g_{n-1,u} A_c^{n-1} \right) B_c. \end{aligned} \tag{16}$$

As $\delta_i = \frac{\nu_j}{n}$, $\delta_{i+n} = \frac{\mu_j}{n}$ and hence, $\sum_{i=0}^{2n-1} \delta_i = 1$, concludes the proof. □

Thus we have now found again the generators for the convex set as defined in (5). Throughout the remainder of the paper we will refer this approach as CH2, with the observation that the resulting polytope is spanned by $2n$ generators.

Remark 1. In Section 2.3 it was pointed out that both [3, 9] propose methods based upon the Jordan Normal Form with the difference that the method of [9] reduces the number of generators at the cost of a larger polytope, e.g. a square can always be contained in a triangle thus reducing the number of points spanning the polytope. A similar reasoning can also be applied to the method CH2 presented above, to obtain a polytope smaller than the one obtained via Theorem 2, but now with 2^n generators instead of $2n$. The method corresponding to this modification of CH2 will be referred to as CH1, to be consistent with the method JNF2 versus JNF1. □

Observe that (9) is of infinite length and will in practice be approximated by a function of finite length p. The resulting polytopic embedding therefore has an error. Next, we provide an explicit upper bound on the 2-norm of the approximation error.

Theorem 3. *Let*

$$\rho := \frac{3\sqrt{\lambda_{max}(A_c^T A_c)}T_s}{p}, \tag{17}$$

and suppose[1] $\rho < 1$. *Then:*

$$\left\| \int_0^{\tau_k} \sum_{k=p}^{\infty} \frac{A_c^k (T_s - \theta)^k}{k!} B_c d\theta \right\|_2 \leq \frac{\rho^p}{1-\rho} \overline{\tau} \sqrt{\lambda_{max}(B_c^T B_c)}. \tag{18}$$

Proof:

$$\begin{aligned}
&\left\| \int_0^{\tau_k} \sum_{k=p}^{\infty} \frac{A_c^k (T_s - \theta)^k}{k!} B_c d\theta \right\|_2 \leq \sum_{k=p}^{\infty} \left\| \int_0^{\tau_k} \frac{A_c^k (T_s - \theta)^k}{k!} B_c d\theta \right\|_2 \\
&\leq \sum_{k=p}^{\infty} \left\| \frac{A_c^k T_s^k}{(\frac{k}{3})^k} B_c \overline{\tau} \right\|_2 \leq \sum_{k=p}^{\infty} \left(\frac{3T_s}{k} \right)^k \overline{\tau} \|A_c^k\|_2 \|B_c\|_2 \\
&\leq \sum_{k=p}^{\infty} \left(\frac{3\sqrt{\lambda_{max}(A_c^T A_c)}T_s}{p} \right)^k \overline{\tau} \sqrt{\lambda_{max}(B_c^T B_c)} = \frac{\rho^p}{1-\rho} \overline{\tau} \sqrt{\lambda_{max}(B_c^T B_c)},
\end{aligned} \tag{19}$$

where the triangle and the Cauchy-Schwarz inequality were used. The inequality $\|A^k\|_2^2 \leq \|A\|_2^2 \times \ldots \times \|A\|_2^2 = \lambda_{max}^k(A^T A)$, which follows from the Cauchy-Schwarz inequality, was also employed. □

Using Theorem 3 one can choose p such that the approximation error is small enough and then correct the resulting polytope accordingly. This can be done by performing a Minkowsky addition of the resulting polytope with the unit ball proportional to the size of the error bound.

[1] Note that p and T_s can always be chosen such that this requirement is satisfied.

4 Suitability for MPC

In this section we present an assessment of all modeling methods considered in this paper with focus on suitability for MPC. To do so, consider system (1)-(2) with $A_c = \left[\begin{smallmatrix} 1 & -1.2 \\ 4 & 6 \end{smallmatrix}\right]$, $B_c = \left[\begin{smallmatrix} 0 \\ -1 \end{smallmatrix}\right]$, $T_s = 0.05$ and $\overline{\tau} = 0.045$. For CH1 and CH2 we chose $p = 15$ and thus, (18) yields $\frac{\rho^p}{1-\rho}\overline{\tau}\sqrt{\lambda_{max}(B_c^T B_c)} \approx 4 \times 10^{-19}$. The approximation order needed by TA was $p = 8$. Each method has its polytope, as defined in (5), and generators spanning the polytope. In Figure 1 these polytopes are plotted. Notice that the accuracy of the methods ME, CH1, CH2 and TA is of the same order of magnitude, whereas for JNF1 and JNF2 the polytope is much larger (different axes).

We will now discuss the methods in terms of scalability, computational aspects and control performance. Firstly, note that the LMI used in [7] for stabilizing controller synthesis scales linearly with the number of generators of Δ. In Table 1 the number of generators for each method is shown.

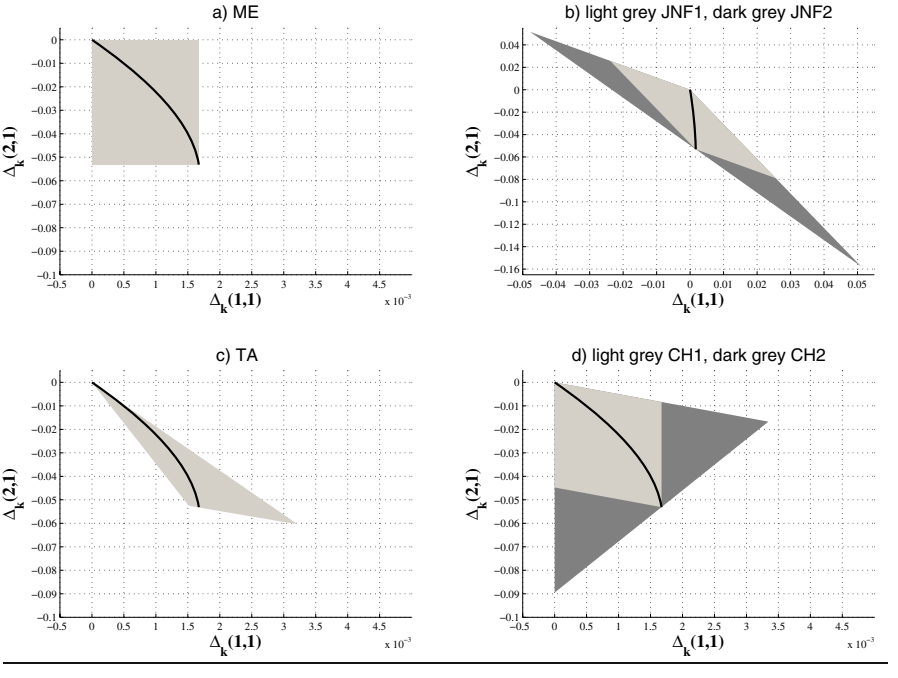


Fig. 1 Different polytopic approximations: along the axes are the values of $\overline{\Delta}_l(1,1)$ and $\overline{\Delta}_l(2,1)$ for $l = 1,2$, in black all the possible realizations of Δ_k and the grey areas are the polytopes

Table 1 The number of generators per method (L)

method:	ME	JNF1	JNF2	TA	CH1	CH2
number of generators:	2^{nm}	2^n	$n+1$	p	2^n	$2n$

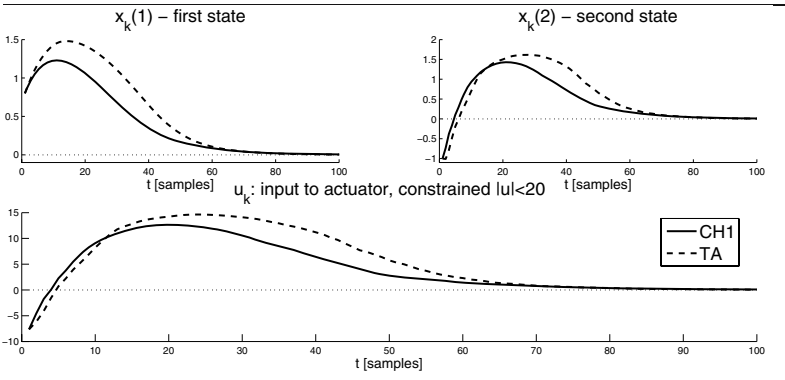


Fig. 2 Simulation of the same MPC scheme for two different models

A few further observations about the various methods are worth noticing:

- The number of generators yielded by the TA method does not depend on the state dimension. Hence, the TA approach seems well suited for large dimension systems, while it can be less efficient for low dimension systems.
- The number of generators for the methods ME, JNF1 and CH1 is an exponential function of the state dimension, which makes these approaches not suitable for large dimension systems.
- The ME method is not implementable because the extreme realizations of Δ_k are not necessarily obtained for $\tau_k = 0$ and $\tau_k = \overline{\tau}$.
- Both JNF methods have the disadvantage that they become complex when the JNF becomes complex, e.g. when A_c has complex values on the diagonal of J. Also, the JNF methods become more complex when A_c is not invertible.
- The TA method does not provide an upper bound on the estimation error due to the finite order approximation of the Taylor series. This means that one has to check stability of the closed-loop system a posteriori. If this stability test fails there is no systematic approach for finding a solution.
- CH1 and CH2 use an algorithm which calculates the determinant of a possibly large matrix and the roots of a high order polynomial.
- For CH1 and CH2, if p is chosen small this increases the number of generators, while if p is chosen very large a correction of the polytope becomes superfluous, but the influence on the computational complexity is insignificant.

Finally we can, by means of the MPC law from [7], compare the performance of the different approaches to see how they perform in the MPC context. At each time instant t_k a feedback gain $K(x_k)$ is calculated by solving a semi-definite programming problem, which yields the control action $u_k := K(x_k)x_k$. JNF2 and CH2 were not considered because they will never outperform their corresponding variants JNF1 and CH1, respectively. ME was not considered because this method is not really applicable due to numerical reasons mentioned above. In Figure 2 we plot the results of a simulation for the system under observation. The resulting closed

loop state and input trajectories of the simulations corresponding to CH1 and TA are plotted. Note that in the simulation CH1 uses less control effort and has less overshoot, even though the two methods achieve the same settling time. In the simulation corresponding to JNF1 the resulting MPC problem was not feasible. This indicates that overestimating the nonlinearity can lead to infeasibility.

5 Conclusions

A novel method for modeling uncertain time-varying input delays was presented. It has been shown that this method indeed creates a polytope that contains all possible realizations of the nonlinear terms induced by delays. Then it was shown how to upper bound the error made in the approximation of an infinite length polynomial and how to compensate for this error. It has been demonstrated that the approach presented in this paper can be more efficient compared with earlier presented methods, also in terms of suitability for MPC. Furthermore, the presented modeling method can be modified to allow for delays larger then the sampling time using techniques similar to the ones employed in e.g., [3, 9], which makes the developed method appealing for control of networked control systems.

References

1. Branicky, M.S., Phillips, S.M., Zhang, W.: Stability of Networked Control Systems: Eplicit Analysis of Delay. In: Proceedings of the American Control Conference, vol. 4, pp. 2352–2357 (2000)
2. Cayley, A.: A memoir on the theory of matrices. Philosophical Transactions London 158, 17–37 (1857)
3. Cloosterman, M.B.G.: Control of Systems over Communication Networks: Modelling, Analysis and Design. Ph.D. dissertation, Eindhoven University of Technology, The Netherlands (2008)
4. Gantmacher, F.R.: Theory of Matrices, vols. 1, 2. Chelsea Publishing Company, New York (1959)
5. Hetel, L., Daafouz, J., Iung, C.: Stabilization of arbitrary switched linear systems with unknown time-varying delays. IEEE Transactions on Automatic Control 51(10), 1668–1674 (2006)
6. Hespanha, J., Naghshtabrizi, P., Xu, Y.: A survey of Recent Results in Networked Control Systems. Proceedings of the IEEE 95, 138–161 (2007)
7. Kothare, M.V., Balakrishnan, V., Morari, M.: Robust Constrained Model Predictive Control using Linear Matrix Inequalities. Automatica 32(10), 1361–1379 (1996)
8. Murray, R.M., Åström, K.J., Boyd, S.P., Brockett, R.W., Stein, G.: Future Directions in Control in an Information-Rich World. IEEE Control Systems Magazine 23(2), 20–33 (2003)
9. Olaru, S., Niculescu, S.I.: Predictive control for linear systems with delayed input subject to constraints. IFAC World Congress, Seoul, Korea (2008)
10. Tipsuwan, Y., Chow, M.-Y.: Control methodologies in networked control systems. Control Engineering Practice 11, 1099–1111 (2003)

A Vector Quantization Approach to Scenario Generation for Stochastic NMPC

Graham C. Goodwin, Jan Østergaard, Daniel E. Quevedo, and Arie Feuer

Abstract. This paper describes a novel technique for scenario generation aimed at closed loop stochastic nonlinear model predictive control. The key ingredient in the algorithm is the use of vector quantization methods. We also show how one can impose a tree structure on the resulting scenarios. Finally, we briefly describe how the scenarios can be used in large scale stochastic nonlinear model predictive control problems and we illustrate by a specific problem related to optimal mine planning.

Keywords: Scenario generation, closed loop control, stochastic nonlinear model predictive control, vector quantization.

1 Introduction

The motivation for the research described in the current paper arises from large scale optimization problems having a temporal component. A specific example of such a problem is open-cut mine planning. In this example, the goal is to determine the value of an asset by carrying out an optimization of possible future actions over a suitable planning horizon (typically 20 years for a mine). Such problems can be converted into nonlinear model predictive control problems by appropriate choice of variables. An important feature of such problems is that they contain a large number of inputs and states. Indeed, a simplified version of the mine optimization problem involves tens of thousands of state variables. Hence, even after the application

Graham C. Goodwin, Jan Østergaard, and Daniel E. Quevedo
School of Electrical Engineering and Computer Science,
The University of Newcastle, NSW 2300, Australia
e-mail: graham.goodwin@newcastle.edu.au,
janoe@ieee.org, dquevedo@ieee.org

Arie Feuer
The Technion, Haifa 32000, Israel
e-mail: feuer@ee.technion.ac.il

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 235–248.
springerlink.com

of spatial and temporal aggregation, it typically takes many hours to carry out the required optimization on a high speed computer. Another important feature of such problems is that there are usually variables whose future values cannot be accurately predicted. For example, in the case of mining, one does not know the future price that the ore will bring. Hence it is desirable to treat such problems in a stochastic setting. Alas, the issue of computational complexity now becomes critical.

The usual model predictive control (MPC) paradigm used in industrial control to deal with uncertainty is the so called "receding horizon" approach. Here one typically uses open loop optimization to determine the control sequence over some horizon and then one applies the first control action. At the next time instant, one measures (or estimates) the state and then recomputes the control over a future control horizon and the first control step is again implemented. This strategy is very well known to the control community and has been extremely successful in practice [2, 3, 4, 5]. This kind of strategy "reacts" to disturbances when they occur (since the input is recalculated based on the measured state). However, no explicit account is taken of the fact that, in the future, we will have more information about the uncertain states than we do at the present time. Control strategies which implement the latter policy are usually termed "closed loop" and lead to so-called scenario trees [6, 7]. Also, there exist intermediate strategies in which a restricted form of feedback is allowed; see, e.g., [8], which studies robust constraint satisfaction and closed loop stability for a class of uncertain linear stochastic models.

There exists a substantial literature[1] on "closed loop" optimization in the stochastic programming literature [9]. There has also been some interest in the topic in recent control literature. For example, the work [10] considers closed loop policies based on the vertices of an assumed set.

Our particular interest in the current paper resides in cases where the state dimension is very large and the underlying system is highly nonlinear. Clearly, in such a problem one needs to be extremely careful with stochastic optimization since the associated computations can easily become intractable. The end result of this line of reasoning is that one can, at best, deal with a "handful" (say several hundred) possible realizations of the uncertain elements in the problem. This, in turn, raises the issue of how one should choose this "handful" of realizations (which we term "scenarios") so they give representative "coverage" of the likely outcomes. To illustrate the difficulty of this problem, we note that if we utilize an optimization horizon of 20 steps and we consider just 10 values for the uncertain variables at each step then this gives 10^{20} realizations of the uncertain process. Since in non-convex stochastic optimization the computational time grows linearly with the number of scenarios, then if it takes several hours to deal with one realization, then clearly 10^{20} realizations is completely impossible. (It would take 10^{17} years!). Obviously, careful choice of the scenarios is an important question in this context.

The topic of scenario generation has been addressed extensively in the stochastic programming literature. For example, in [11, 12, 13], scenario tree generation for specific multi-stage optimization problems is considered. In these works,

[1] The policies are sometimes said to be "with recourse" in the stochastic programming literature.

algorithms are proposed which in certain cases and for the Wasserstein (transportational) distance metric, lead to optimal scenario trees. For alternative approaches, see, e.g. [14, 15].

One obvious recommendation made in the literature is that one should ideally design the scenarios taking into account the "true problem". However, this is not sensible in the context of complex problems since it is computationally intractable to compare different scenario patterns, e.g., via a Monte-Carlo study. Indeed, if this were possible, then one could simply carry out the intended design directly.

Our strategy will be to divide the problem into two stages. In the first stage, we will carry out scenario design based on a simple measure of scenario performance. In this stage, we rely upon the fact that the number of uncertain variables is typically small (say 2 or 3 in the case of the mining problem). Then, in a second stage, we will utilize the scenarios on the "true problem". This "divide-and-conquer" strategy is aimed at making the overall problem computationally tractable. The novel contribution in the current paper is to link the problem of scenario generation to code book design in vector quantization. This link allows us to develop a new strategy for scenario generation. We also explain several embellishments of the basic scheme including how to enforce a tree structure on the scenarios. The latter is used for closed loop stochastic control.

The layout of the remainder of the paper is as follows: In section 2, we give a brief overview of the mine planning problem so as to place the subsequent work in a practical context. In section 3 we briefly review different stochastic optimization strategies. Section 4 contains the key result of the paper, namely, the scenario generation algorithm. In section 5 we briefly return to the mine planning problem and conclusions are given in section 6.

2 Motivational Problem

Before describing the scenario generation strategy, we will first set the work in a practical context by briefly describing the optimal mine planning problem [1, 5].

The key idea is as follows: given geological data based on preliminary exploration, determine where and when to dig. The optimization problem can be cast as a mixed integer linear programming (MILP) problem. A host of constraints need to be satisfied e.g. mining capacity in each year, slope constraints on the walls of the mine, precedent constraints on the order in which material is removed, processing plant constraints etc.

If one adopts the, so called, block model approach, then one divides the mine into blocks say 100×100 on the surface and 10 vertically. This gives 10^5 blocks. Over a 15 year horizon, this gives 10^{75} decisions on when to remove a block. Interestingly, 10^{75} is approximately the number of atoms in the known universe, so clearly some simplifications are necessary.

The basic problem can be given a nice interpretation in the NMPC framework. To see how this can be done, we divide the surface into rectangular blocks $\{j = 1, \ldots, M; k = 1, \ldots, M\}$ as shown in Fig. 1.

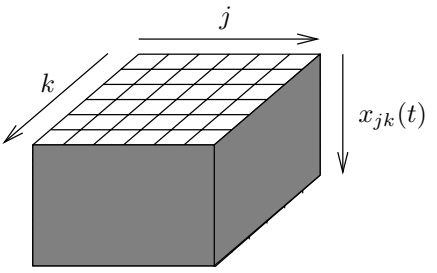


Fig. 1 Block model of a mine

Let $x_{jk}(t)$ denote the depth at location j, k at time t, $u_{jk}(t) \in \{0, 1\}$ denote the decision to mine (1) or not to mine (0) at time t. Then a simple state space model is

$$x_{jk}(t) = x_{jk}(t-1) + bu_{jk}(t). \tag{1}$$

The various constraints take the form $\sum_{jk} h^l_{jk} x_{jk} \leq b_l$. The cost function can be expressed as $J = \sum_{t=1}^{N} d_t c_t \sum_{j,k} V_{j,k}(x_{jk}(t)) u_{j,k}$, where $V_{jk}(x_{jk}(t))$ denotes the amount of ore at depth $x_{jk}(t)$ in location (j, k), d_t denotes a discount factor and c_t denotes the value of ore at time t.

3 Stochastic Optimization Strategies

The simplified description of the mine planning problem given above implicitly assumes that the value of the ore is known. However, future values of this variable are certainly not exactly known. Several strategies can be adopted to deal with this uncertainty as described below:

Open Loop Policies. Here one carries out the design based on some nominal trajectory (say the expected value) for the uncertain variables. Then one applies the strategy irrespective of what actually happens. This may sound rather strange to the control community but, in mine planning, certain decisions (e.g. how large to make the processing plant) cannot easily be changed in the light of updated information.

Receding Horizon (or Reactive) Policies. Here one bases the original design on some nominal trajectory for the uncertain variables. However, one only implements the first stage. One then re-does the optimization when new information is obtained (i.e. one "reacts" to incoming data). This idea is central to model predictive control and will be very familiar to the control community.

Closed Loop Policies. These policies take account of the fact that, in the future, we will have additional information not available now. Closed loop policies typically lead to function optimization problems in which one designs a mapping from the future information state to the control. We give a brief overview of the dynamic programming (DP) approach to these policies.

Let I_k denote the information available to the controller at time k, that is

$$I_k = (y_0, \ldots, y_k, u_0, \ldots, u_{k-1}). \tag{2}$$

The required control policy $\pi(\mu_0, \ldots, \mu_{N-1})$ maps I_k into the control space c_k. A key feature is the non-anticipatory constraint i.e. decisions can only be based on the information that has been revealed so far.

The cost function takes the form:

$$J_\pi = \underset{z_0,\{\omega_k\},\{\nu_k\}}{\mathbb{E}} \left\{ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(I_k), \omega_k) \right\} \tag{3}$$

where the state evolution satisfy $x_{k+1} = f_k(x_k, \mu_k(I_k), \omega_k)$. The available measurements are $y_0 = h_0(x_0, \nu_0)$ and $y_k = h_k(x_k, \mu_{k-1}(I_{k-1}), \nu_k)$, where $\{\omega_k\}, \{\nu_k\}$ are i.i.d. sequences (typically Gaussian distributed).

The associated DP equations are

$$\begin{aligned} J_{N-1}(I_{N-1}) = \min_{u_{N-1} \in \mathcal{U}_{N-1}} \Bigg\{ \underset{x_{N-1}, \omega_{N-1}}{\mathbb{E}} \Big[& g_N(f_{N-1}(x_{N-1}, u_{N-1}, \omega_{N-1})) \\ & + g_{N-1}(x_{N-1}, u_{N-1}, \omega_{N-1}) | I_{N-1}, \mu_{N-1} \Big] \Bigg\} \end{aligned} \tag{4}$$

and for $k = 0, \ldots, N-2$

$$J_k(I_k) = \min_{u_k \in \mathcal{U}_k} \left\{ \underset{x_k, \omega_k, y_{k+1}}{\mathbb{E}} \Big[g_k(x_k, u_k, \omega_k) + J_{k+1}(I_k, y_{k+1}, u_k) | I_k, \mu_k \Big] \right\}. \tag{5}$$

A Simple Example. To illustrate closed loop planning, we consider the simple two-stage stochastic decision problem in Fig. 2. We see in this figure that there is only one random variable, ω, which takes one of two values, ω_1, ω_2 with equal probability. There are two stages in the problem and two decisions for the control at each stage. Thus, at stage 1, u_0 can be chosen as a or b and at stage 2, u_1 can be chosen as a or b. The final rewards (cost function) are shown on the right of the diagram.

Optimal open loop and reactive policies do not use the fact that the state will be known after stage 1 has been completed. Thus open loop and reactive policies both lead to the same return of \$2. This can be seen from the following simple argument:

Say we apply $u_0 = a$; then whatever we do next gives $\pm\$50,000$ with equal probability. Hence, the expected return is \$0. However, $u_0 = b, u_1 = a$ returns \$1 and $u_0 = b, u_1 = b$ returns \$2. Thus, in conclusion, the best open loop strategy is $u_0 = b, u_1 = b$ yielding \$2 (we would get the same answer with a reactive policy).

For the closed loop case, we add the extra information that we will know where we have reached at the end of stage 1. The obvious closed loop policy is; $u_0 = a$,

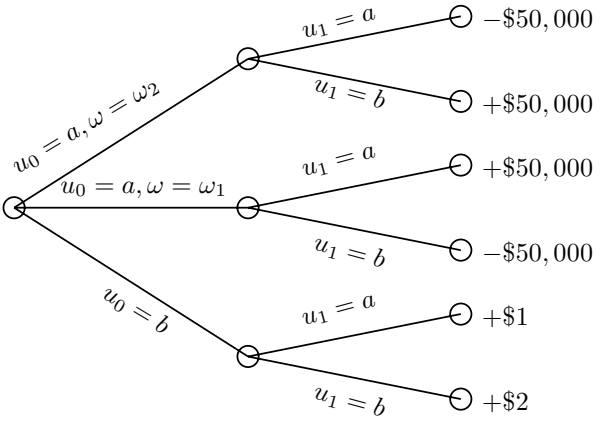


Fig. 2 Simple example

then $u_1 = a$ if $\omega = \omega_1$ which implies a return of $\$50,000$ and $u_0 = a$, then $u_1 = b$ if $\omega = \omega_2$ which implies a return of $\$50,000$.

We see from the above that a closed loop strategy can give significant benefits compared with open loop or reactive. At a heuristic level the closed loop policy keeps "all options open" and avoids being "painted into a corner" by the first move.

Computational Issues. The computational burden associated with the design of closed loop policies grows linearly with the number of alternatives considered for the uncertain variables. Hence, it is usually essential to restrict the cardinality of the set of alternatives for the uncertain variables. The philosophical basis of the strategy is to carry out extensive off-line calculations (by Monte Carlo-like techniques) so as to design a small set of scenarios which are then used in the on-line solution of the optimization problem. This amounts to replacing a very complex on-line problem by a simpler on-line problem based on the pre-computed scenarios. The issue of how to choose the representative set of alternatives is addressed in the next section.

4 Scenario Generation

The goal of scenario generation is to come up with a (relatively small) set of representative trajectories for a stochastic process. A specific example is that of ore prices as described in section 2.

A "brute force" method is to use Monte Carlo type methods to simulate a set of trajectories using different "seeds" for the underlying innovation process. This is known to perform well for a large number of scenarios. However, in the case where the cardinality of the scenario set is severely restricted, then it is prudent to exercise some care in the scenario selection.

In the next subsection we describe the key novel contribution of this paper, namely, linking the problem of scenario generation to vector quantization.

Vector Quantization. Let us first briefly review some important properties of vector quantization (VQ). For a thorough introduction to VQ we refer the reader to [16, 17].

An L-dimensional vector quantizer Q_L is a (nonlinear and non-invertible) map, say $Q_L : \mathbb{R}^L \to \mathcal{C}$, where $\mathcal{C}$ is a discrete set of M distinct elements given by

$$\mathcal{C} \triangleq \{c^i \in \mathbb{R}^L : i = 1, \ldots, M\}. \tag{6}$$

The set $\mathcal{C}$ is also known as a codebook and the element $c^i \in \mathcal{C}$ is usually referred to as the ith codeword or ith reconstruction point.

It is convenient to decompose Q_L into a cascade of two functions, e.g. $Q_L(\cdot) = \beta(\alpha(\cdot))$ where α is referred to as the encoder and β is the decoder. The encoder is a many-to-one function which maps points in $\mathbb{R}^L$ to indices, i.e. $\alpha : \mathbb{R}^L \to \mathcal{I}$, where $\mathcal{I}$ is an index set defined as

$$\mathcal{I} \triangleq \{i \in \mathbb{N} : i = 1, \ldots, M\}. \tag{7}$$

We then define $\alpha(x) \triangleq i$ if and only if $x \in S^i$ where $S^i \in \mathcal{S}$ and where

$$\mathcal{S} \triangleq \{S^i \subset \mathbb{R}^L : i = 1, \ldots, M\}. \tag{8}$$

We generally require that $\mathcal{S}$ "cover" $\mathbb{R}^L$, i.e. $\mathbb{R}^L \subseteq \mathcal{S}$ and moreover that any pair of subsets $(S^i, S^j), i \neq j$, do not overlap except possibly at their boundaries.

The decoder is given by $\beta : \mathcal{I} \to \mathcal{C}$, where $\mathcal{I}$ is given by (7) and $\mathcal{C}$ by (6). With this, we establish the following:

$$Q_L : x \mapsto c^i \Leftrightarrow x \in S^i \Leftrightarrow \alpha(x) = i, \beta(i) = c^i \quad \text{so that} \quad Q_L(x) = \beta(\alpha(x)). \tag{9}$$

Given a distortion measure (or cost function) $\rho : \mathbb{R}^L \times \mathcal{C} \to \mathbb{R}^+$ we define a Voronoi cell V^i :

$$V^i \triangleq \{x \in \mathbb{R}^L : \rho(x, c^i) \leq \rho(x, c^j), \quad j = 1, \ldots, M\}, \quad i = 1, \ldots, M. \tag{10}$$

It follows that if Q_L is a *nearest neighbor* quantizer, then $S^i = V^i$ and this is in fact an optimal encoder for the given decoder, i.e. for the given set of codewords $\mathcal{C}$ [16].

Let $\phi_X(x)$ denote the probability density function for the random variable X. Then, an optimal quantizer is one that, for a given M, minimizes the expected cost J where

$$\begin{aligned} J &= \mathbb{E}\rho(X, Q_L(X)) \\ &= \sum_{i=1}^{M} \int_{x \in S^i} \phi_X(x) \rho(x, c^i)\, dx = \sum_{i=1}^{M} P(X \in S^i) \mathbb{E}[\rho(x, c^i) | X \in S^i]. \end{aligned} \tag{11}$$

In simple cases, the codeword c^i only appears in one of the terms of the sum in (11). It then follows that the optimal codeword, given the set S^i, is the generalized centroid of S^i. Specifically, given $S^i \in \mathcal{S}$

$$\hat{c}^i = \arg\min_{c^i \in \mathbb{R}^L} \mathbb{E}\left[\rho(X, c^i) | X \in S^i\right] = \arg\min_{c^i \in \mathbb{R}^L} \frac{\int_{x \in S_i} \phi_X(x) \rho(x, c)\, dx}{\int_{x \in S_i} \phi_X(x)\, dx}. \quad (12)$$

In other words, given the encoder, or equivalently, given the set $\mathcal{S}$, the optimal decoder is defined by the set of reconstructions points $\mathcal{C} = \{\hat{c}^i : i = 1, \ldots, M\}$, where $\hat{c}^i$ is given by (12).

An optimal quantizer is therefore a nearest neighbor quantizer having centroids as codewords [16]. For example, if ρ is the squared error distortion measure, i.e. $\rho(x, x') = \|x - x'\|^2 = \sum_{n=0}^{L-1} |x_n - x'_n|^2$ then it is easy to show that

$$\hat{c}^i = \arg\min_{c^i \in \mathbb{R}^L} \mathbb{E}\left[\rho(X, c^i) | X \in S^i\right] \quad (13)$$

$$= \mathbb{E}\left[X | X \in S^i\right] \quad (14)$$

$$= \frac{\int_{x \in S_i} \phi_X(x) x\, dx}{\int_{x \in S_i} \phi_X(x)\, dx}. \quad (15)$$

Furthermore, if X is stationary and ergodic, then one can approximate the centroid by the sample average obtained simply by drawing a large number of points from S^i and taking their average [16].

Unfortunately, it is generally hard to design a jointly optimal encoder and decoder pair $(\alpha(\cdot), \beta(\cdot))$. However, there exist iterative design algorithms which yield locally optimal quantizers. One such algorithm is Lloyd's algorithm, which was originally defined for the scalar case [18] and later extended to the vector case [19].

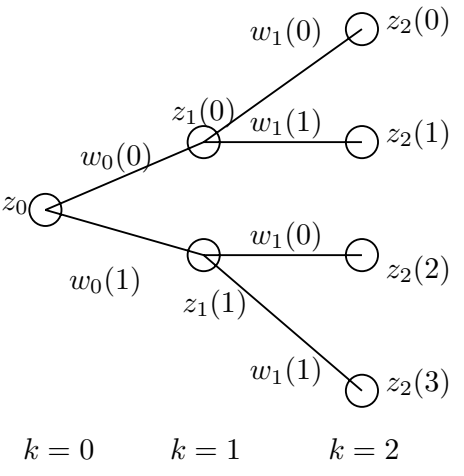
Lloyd's algorithm (and its extension to the vector case) is basically a cyclic minimizer that alternates between two stages; given an optimal encoder α, find the optimal decoder β and given an optimal decoder find an optimal encoder. More specifically, we first construct a random set of codewords $\mathcal{C}$. Then we repeatedly apply the following two steps:

1. Given a set of centroids $\mathcal{C} = \{c^i\}_{i=1}^M$, find the Voronoi cells $\mathcal{S} = \{S^i\}_{i=1}^M$ by use of (10).
2. Given a set of decision cells $\mathcal{S} = \{S^i\}_{i=0}^{M-1}$ find the centroids $\mathcal{C} = \{c^i\}_{i=1}^M$ by use of (12).

This approach guarantees convergence to a (local) minimum [20].[2]

[2] In practice, one can run the codebook design algorithm for several initial guesses and choose the one that yields the lowest distortion (on the same test data).

Fig. 3 Binary tree



Scenario Generation by Vector Quantization Techniques. We will now establish a connection between the extended Lloyd's VQ design algorithm and scenario generation in stochastic MPC.[3]

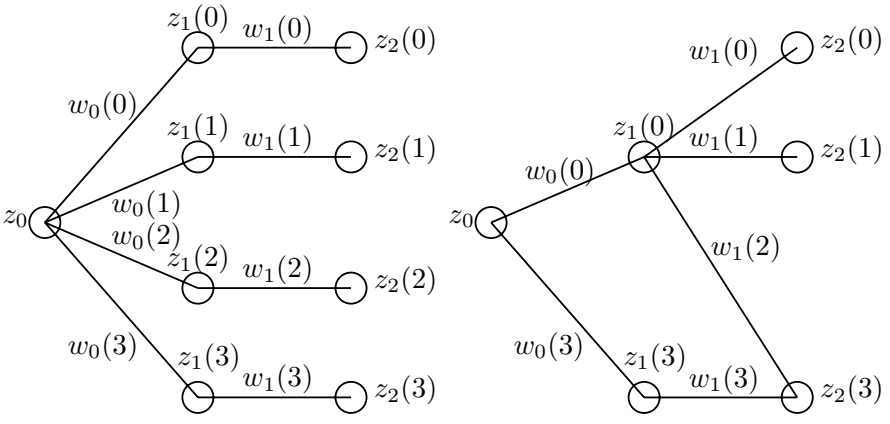
Let $z_k \in \mathbb{R}^L$ be a state vector that satisfies the Markovian recursion given by

$$z_{k+1} = f(z_k, \omega_k) \tag{16}$$

where $\omega_k \in \mathbb{R}^L$ is an arbitrary distributed random vector process.

In the special case where $\omega_k \in \{w_k(0), \omega_k(1)\}$, i.e. the disturbance, can take on only two distinct values at every time instant k, then the evolving state sequence describes a binary tree as shown in Fig. 3. The root of the tree describes the initial state z_0 at time $k = 0$. Then, at time $k = 1$, the next state, i.e. z_1 will take on the value $z_1(0)$ or $z_1(1)$ depending upon whether the event $\omega_0(0)$ or $\omega_0(1)$ happens. At time $k = 2$, if the previous state was $z_1(0)$, the current state will be either $z_2(0)$ or $z_2(1)$. Similarly, if the previous state was $z_1(1)$ then the current state will be either $z_2(2)$ or $z_2(3)$ depending on the actual realization of the uncertain disturbance ω_k. Thus, four different state trajectories are possible and at time $k = 0$ it is not known in advance which one of them will eventually happen. The only information that is available at time $k = 0$ is the statistics, i.e. the probability of each of the trajectories. Notice that we can describe each trajectory, i.e. each path in the tree, by a sequence of disturbances. Specifically, the ith path can be described by the sequence $\omega^i = (\omega_0(i), \omega_1(i))$, where $i \in \{0, \ldots, 3\}$. In the general case where we have $N + 1$ stages in the tree and M distinct end nodes $\{z_N(i)\}, i = 1, \ldots, M$, which (in the case of the binary tree) corresponds to M distinct paths in the tree, we have $\omega^i = (\omega_0(i), \omega_1(i), \ldots, \omega_{N-1}(i))$. Furthermore, there is a one-to-one correspondence between the sequence of disturbances ω^i and the sequence of

[3] It is worth emphasizing that the related k-means algorithm [21] has been adapted for a specific instance of scenario generation in [11]. Our extension includes allowing arbitrary cost functions and imposing desired tree structures.



(a) No node sharing (b) Partial node sharing

Fig. 4 Different scenario trees

state vectors $z^i = (z_0(i), z_1(i), \ldots, z_N(i))$. To see this, recall that $z_{k+1}(i) = f_k(z_k(i), \omega_k(i))$.

With the above, we will refer to a sequence of disturbances, say ω^i, as a scenario. In particular, ω^i denotes the ith scenario. We are interested in scenario generation for finite-horizon stochastic MPC. If the disturbances takes on only a finite number of possible values at each time instant, then we can form a scenario tree, e.g. as the one shown in Fig. 3. Of course, many other trees are possible, cf. Figs. 4(a) and 4(b).

It is often the case that the disturbances take on a continuum of values. In this case, we seek to form a finite number of scenarios by discretizing the set of possible disturbances. Specifically, we wish to design M distinct scenarios, whose trajectories capture the evolution of the the most likely state sequences. In other words, the set of M candidate scenarios should (on average) be a good approximation of all possible sequences of disturbances. Thus, we actually wish to design a codebook $\mathcal{C}$ in the ω-space having M codewords where the codewords $\{\omega^i \in \mathcal{C}\}_{i=1}^M$ are themselves scenarios.

Let J_N be the N-horizon cost function defined by

$$J_N \triangleq \mathbb{E} \min_{\omega^i \in \mathcal{C}} \rho_N(z, z^i) \tag{17}$$

$$= \min_{\mathcal{S}} \sum_{i=1}^{M} P(\omega \in S^i) \mathbb{E}[\rho_N(z, z^i) | \omega \in S^i] \tag{18}$$

where

$$\hat{\omega}^i = \arg \min_{\omega^i \in \mathbb{R}^{L(N+1)}} \mathbb{E}[\rho_N(z, z^i) | \omega \in S^i] \tag{19}$$

is the generalized centroid of an optimal nearest neighbor quantizer, $S^i \subset \mathbb{R}^{L(N+1)}$ and $\rho_N(\cdot,\cdot)$ is given by

$$\rho_N(z, z^i) = \|z - z^i\|_Q^2 \tag{20}$$

$$= \sum_{k=0}^{N} \|z_k - z_k(i)\|_{Q_k}^2 \tag{21}$$

where $\{Q_k \in \mathbb{R}^{L\times L}\}, k = 0, \ldots, N$ is a sequence of weighting matrices and $Q = \text{diag}(Q_0, \ldots, Q_N)$.

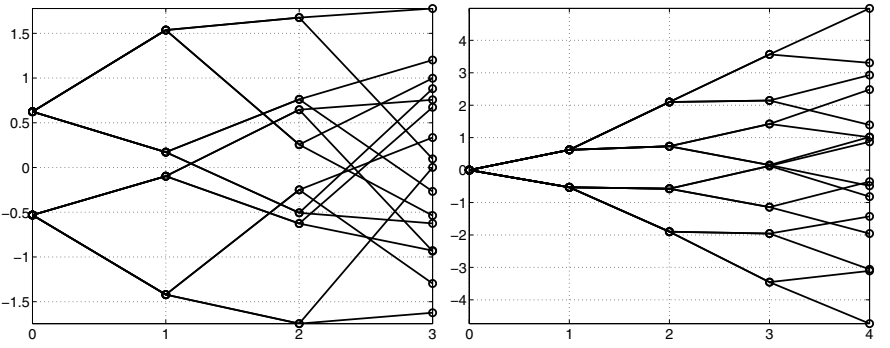
Interestingly, this approach yields a jointly (locally) optimal distribution of codewords over the temporal as well as spatial dimensions. Note, however, that one has no control over the resulting structure of the scenario tree. In fact, a likely outcome is a scenario tree with a root node that branches into M separate *deterministic* paths (similar to Fig. 4(a) for the case of $M = 4$).

Imposing a Tree Structure on the Scenarios. In the previous section we allowed arbitrary scenario trees. Clearly, this yields the lowest possible cost. Nonetheless, in stochastic closed loop planning one requires that the future uncertainty be progressively reduced as the stages proceed. Thus, a tree like structure is required in the scenario space, see also [12, 14, 15, 11]. We will impose a particular tree structure which is especially suited to the Dynamic Programming formulation of stochastic optimal control. In the chosen tree structure the nodes of the tree share common points. We enforce this by adding linear equality constraints at each node in the code book generation algorithm. Let Υ_N^M denote the set of all possible tree structures containing exactly M distinct *paths* each having $N + 1$ nodes. For example, Figs. 3, 4(a), and 4(b), all belong to Υ_2^4. It should be clear that any codebook $\mathcal{C}$ having M codewords $\{\omega^i\}_{i=1}^M$ (each having N elements $\omega_k^i, k = 0, \ldots, N-1$) *admits* a tree $\Gamma \in \Upsilon_N^M$. We write $\mathcal{C} \triangleright \Gamma$ if $\mathcal{C}$ admits the specific scenario tree described by Γ.

When we restrict the codebook to admit a specific scenario tree, the codeword separation described in (19) does not apply and one needs to design the full codebook simultaneously. Specifically, given a scenario tree $\Gamma \in \Upsilon_N^M$ and a set of decision cells $\mathcal{S} = \{S^i \in \mathbb{R}^L : i = 1, \ldots, M\}$, the optimal set $\hat{C}$ of codewords must jointly satisfy:

$$\hat{\mathcal{C}} = \arg\min_{\mathcal{C}\triangleright\Gamma} \sum_{i=1}^{M} P(\omega \in S^i)\mathbb{E}[\rho_N(z, z^i)|\omega \in S^i] \tag{22}$$

where the minimization is now over discrete sets $\mathcal{C} \subset \mathbb{R}^{L(N+1)}$ satisfying $|\mathcal{C}| = M$ in addition to $\mathcal{C} \triangleright \Gamma$, which has the equivalent interpretation of minimizing over points in a higher dimensional vector space, i.e. $\mathcal{C} \in \mathbb{R}^{ML(N+1)}$ subject to $\mathcal{C} \triangleright \Gamma$.



(a) 16 (4-dimensional) codewords (b) The resulting binary scenario tree

Fig. 5 The resulting codebook and scenario tree after five iterations of the modified Lloyd's algorithm

We thus modify Lloyd's algorithm to alternate between updating the decision cells $\mathcal{S}$ using (10) keeping the codebook $\mathcal{C}$ fixed and updating the codebook using (22) keeping the decision cells fixed.

Example of Scenario Generation. To illustrate the principle behind the proposed scenario generation technique, we carry out a simple simulation. Let $z_{k+1} = 0.9z_k + \omega_k$ where $z_k, \omega_k \in \mathbb{R}$ and ω_k is a zero-mean unit-variance Gaussian distributed random variable. Let the horizon length be $N = 4$ and let the number of codewords be $M = 16$. It follows that we are interested in finding 16 "good" 4-dimensional codewords $\{\omega^i\}_{i=1}^4$ defined in the ω-space, which admit a specific scenario tree, say a binary tree. For simplicity, we will minimize the squared error in the state-space domain, i.e. $\rho_N(z, z^i) = \|z - z^i\|^2$.

We now first randomly pick a set of 16 codewords (from the distribution of ω) which admits a binary scenario tree. We then randomly draw $20,000$ 4-dimensional vectors (also from the distribution of ω) to be used as "training" vectors.[4] Finally, we alternate between numerically evaluating (10) and (22) given the training set. The resulting codebook and scenario tree after five iterations is illustrated in Figs. 5(a) and 5(b), respectively.

5 Return to the Motivational Problem

Finally, we return to the optimal mine planning example. We recall that the state for this problem has two decoupled components; namely the mine depth at various locations and the ore price. For simplicity we assume that the current ore price states can be measured.

[4] In the case of simple distributions in the ω-space, it might be possible to explicitly derive the associated distribution in the z-space. This is convenient, since it eliminates the need for "training" vectors.

We can set the problem up as a large problem in which we assign a different control action to each node of the scenario tree. This implicitly imposes a mapping from the measured state to the control input [5].

We utilize the scenario tree for ore price to evaluate the current control as a function of the measured state x_0. Now time is advanced 1 step to $k = 1$. Since we have discretized the scenario space, the measured value of x_1 will, with probability one, not coincide with any of the values used in evaluating the current input. Thus, the scenario tool is really only a computational device to allow us to compute the current control action μ_0 whilst accounting (in some way) for the fact that, in the future, we will actually know more than we do now. In other words, exactly as in the simple example of Section 3, we utilize scenarios to ensure that the first step is made in the knowledge that more will be known in the future.

Of course, the fact that the measured value of x_1 is not exactly equal to any of the values used the calculation should not be of great concern to us. All we need to do is to react to the measured value of x_1 at time 1 and recalculate u_1 by the same procedure as was used to evaluate μ_0.

We call the above strategy a receding horizon closed loop policy.

6 Conclusion

This paper has described a novel approach to scenario generation aimed at complex nonlinear model predictive control problems. We have shown that the problem can be formulated in the framework of codebook design for vector quantization. We have also shown how the method can be embellished in several ways, e.g. by imposing a tree structure on the scenarios. A crucial point to note is that extensive off-line calculations (needed to generate the scenarios) are used to simplify the necessary on-line computational burden.[5]

References

[1] Goodwin, G.C., Serón, M.M., Mayne, D.Q.: Optimization opportunities in mining, metal and mineral processing. Annual Reviews in Control 32(1), 17–32 (2008)

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

[3] Goodwin, G.C., Serón, M.M., De Doná, J.A.: Constrained Control & Estimation – An Optimization Perspective. Springer, London (2005)

[4] Camacho, E.F., Bordons, C.: Model Predictive Control. Springer, New York (1999)

[5] Rojas, C.R., Goodwin, G.C., Serón, M.M.: Open-cut mine planning via closed-loop receding horizon optimal control. In: Sánchez-Peña, R., Quevedo, J., Puig Cayuela, V. (eds.) Identification and control: The gap between theory and practice. Springer, Heidelberg (2007)

[6] Dreyfus, S.E.: Some types of optimal control of stochastic systems. J. SIAM, Series A: Control 2(1), 120–134 (1964)

[5] We have verified the advantages of this approach in the context of optimal mine planning in collaborative work with BHP-Billiton.

[7] Bertsekas, D.P.: Dynamic programming and suboptimal control: A survey from adp to mpc. European J. Contr. 11(4–5), 310–334 (2005)
[8] Cannon, M., Kouvaritakis, B., Wu, X.: Stochastic predictive control with probabilistic constraints. Automatica (to appear)
[9] Pflug, G.C.: Optimization of Stochastic Models: The Interface Between Simulation and Optimization. Kluwer Academic Publishers, Boston (1996)
[10] Muñoz de la Peña, D., Bemporad, A., Alamo, T.: Stochastic programming applied to model predictive control. In: Proc. IEEE Conf. Decis. Contr. and Europ. Contr. Conf., Seville, Spain, pp. 1361–1366 (December 2005)
[11] Pflug, G.C.: Scenario tree generation for multiperiod financial optimization by optimal discretization. Math. Program., Ser. B 89, 251–271 (2001)
[12] Mirkov, R., Pflug, G.C.: Tree approximations of stochastic dynamic programs. SIAM Journal on Optimization 18(3), 1082–1105 (2007)
[13] Hochreiter, R., Pflug, G.C.: Financial scenario generation for stochastic multi-stage decision processes as facility location problems. Ann. Oper. Res. 152(1), 257–272 (2007)
[14] Gülpınar, N., Rustem, B., Settergren, R.: Simulation and optimization approaches to scenario tree generation. Journal of Economic Dynamics & Control 28, 1291–1315 (2004)
[15] Kuhn, D.: Aggregation and discretization in multistage stochastic programming. Math. Program., Ser. A 113, 61–94 (2008)
[16] Gersho, A., Gray, R.M.: Vector Quantization and Signal Compression. Kluwer Academic Publishers, Dordrecht (1992)
[17] Gray, R.M., Neuhoff, D.: Quantization. IEEE Trans. Inf. Theory 44(6), 2325–2383 (1998)
[18] Lloyd, S.P.: Least squares quantization in PCM (unpublished Bell Laboratories technical note) (1957)
[19] Linde, Y., Buzo, A., Gray, R.M.: An algorithm for vector quantizer design. IEEE Trans. Commun. 28, 84–95 (1980)
[20] Gray, R., Kieffer, J., Linde, Y.: Locally optimal block quantizer design. Information and Control 45, 178–198 (1980)
[21] MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proc. 5th Berkeley Symp., vol. 1, pp. 281–297 (1967)

Successive Linearization NMPC for a Class of Stochastic Nonlinear Systems

Mark Cannon, Desmond Ng, and Basil Kouvaritakis

Abstract. A receding horizon control methodology is proposed for systems with nonlinear dynamics, additive stochastic uncertainty, and both hard and soft (probabilistic) input/state constraints. Jacobian linearization about predicted trajectories is used to derive a sequence of convex optimization problems. Constraints are handled through the construction of tubes and an associated Markov chain model. The parameters defining the tubes are optimized simultaneously with the predicted future control trajectory via online linear programming.

Keywords: Nonlinear systems, stochastic optimal control, constrained control.

1 Introduction

Constraints handled by predictive control strategies are typically treated as hard (inviolable) constraints, or as soft constraints, in which case the degree of violation is to be minimized in some sense. This paper considers probabilistic constraints in the form of soft input/state constraints, for which the probability of violation is subject to hard limits. This form of constraint can account for the distribution of model or measurement uncertainty, and thus avoid the conservativeness of a hard-constraint strategy based on the worst-case uncertainty [2], which may be highly unlikely. The approach also provides statistical guarantees of closed-loop constraint satisfaction, unlike approximate methods [8, 6] based on constraints on the means and variances of predicted variables.

The difficulties of predicting the distributions of model states over a horizon and of ensuring recursive feasibility in closed-loop operation have limited MPC based on probabilistic constraints to highly computationally intensive Monte Carlo methods (e.g. [1]) or to limited problem classes (e.g. linear dynamics [7]). This paper

Mark Cannon, Desmond Ng, and Basil Kouvaritakis
Department of Engineering Science, University of Oxford, OX1 3PJ, UK
e-mail: mark.cannon@eng.ox.ac.uk

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 249–262.
springerlink.com

considers nonlinear systems with stochastic disturbances, and proposes a receding horizon control law subject to probabilistic and hard constraints based on tubes [4, 3]. Analysis of a simplified Markov chain model verifies that the probability of constraint violation is within the limits specified by the soft constraints. Linearizations about predicted trajectories allow for an efficient online optimization which may be terminated after a single iteration. The approach is illustrated by a numerical example in Section 5.

1.1 Problem Statement

The system to be controlled is described by a discrete-time nonlinear model with state $x_k \in \mathbb{R}^{n_x}$ and input $u_k \in \mathbb{R}^{n_u}$:

$$x_{k+1} = f(x_k, u_k) + d_k, \quad k = 0, 1, \ldots \tag{1}$$

and with $f(0,0) = 0$. Here d_k is a random disturbance with a finitely supported, stationary distribution satisfying

$$\mathbb{E}(d_k) = 0, \quad \forall k$$

(where $\mathbb{E}(\cdot)$ denotes expectation). Furthermore d_j, d_k are assumed to be independent for all $j \neq k$. We assume that x_k is available for measurement at time k. The dynamics of (1) are assumed to be continuous throughout the operating region for the state (denoted $\mathscr{X}$) and input (denoted $\mathscr{U}$) in the following sense.

Assumption 1. *$f(x,u)$ is Lipschitz continuous for all $(x,u) \in \mathscr{X} \times \mathscr{U}$.*

The system is subject to two types of constraint on state and input variables. Hard constraints of the form

$$F_H x_k + G_H u_k \leq h_H, \quad h_H \in \mathbb{R}^{n_H} \tag{2}$$

must be satisfied at all times $k = 0, 1, \ldots$. Thus, for example, we require the set of feasible (x,u) for (2) to be a subset of the operating region, i.e.

$$\{(x,u) : F_H x + G_H u \leq h\} \subset \mathscr{X} \times \mathscr{U}.$$

In addition, we consider soft input/state constraints:

$$F_S x_k + G_S u_k \leq h_S, \quad h_S \in \mathbb{R}^{n_S} \tag{3}$$

which may be violated at any given time k, but which are subject to hard bounds on the expected number of constraint violations over a given horizon. To simplify presentation (but with no loss of generality), we consider the case of a single soft constraint ($n_S = 1$). The bound on the expected number of constraint violations can therefore be expressed as a hard constraint:

$$\frac{1}{N_c}\sum_{i=1}^{N_c}\Pr\{F_S x_{k+i}+G_S u_{k+i}>h_S\}\le\frac{N_{\max}}{N_c} \tag{4}$$

which must hold for all $k=0,1,\ldots$. Here $\Pr\{A\}$ denotes the probability of event A, and $N_{\max}/N_c$ is the maximum allowable rate of violation of soft constraints averaged over an interval of N_c samples.

The control objective is the optimal regulation of x_k about the origin with respect to the performance index

$$J(\{u_0,u_1,\ldots\},x_0)=\sum_{k=0}^{\infty}\mathbb{E}_0\left(\mathbf{1}^T|x_k|+\lambda\mathbf{1}^T|u_k|\right) \tag{5}$$

subject to constraints (2) and (4). Here $\mathbb{E}_k(\cdot)$ denotes expectation conditional on information available to the controller at time k, namely the measured state x_k; $\mathbf{1}$ is a vector, $\mathbf{1}=\begin{bmatrix}1&\cdots&1\end{bmatrix}^T$, with dimension dependent on the context and $\lambda>0$ is a control weighting. The 1-norm cost defined in (5) is employed for computational convenience but the paper's approach is easily extended to more general stage costs that are convex in (x_k,u_k).

2 Successive Linearization MPC

This section describes in outline a method of solving the receding horizon formulation of the control problem defined in Section 1.1. Let $\{u_{k|k},u_{k+1|k},\ldots\}$ denote a predicted input sequence at time k and denote $\{x_{k|k},x_{k+1|k},\ldots\}$ as the corresponding state trajectory, with $x_{k|k}=x_k$. Following the dual mode prediction paradigm [5], we define the infinite horizon predicted input sequence in terms of a finite number of free variables, $\mathbf{c}_k=\{c_{0|k},\ldots,c_{N-1|k}\}$, as:

$$u_{k+i|k}=Kx_{k+i|k}+c_{i|k} \tag{6}$$

with

$$c_{i|k}=0,\quad i=N,N+1,\ldots$$

The linear feedback law $u=Kx$ is assumed to stabilize the model (1) in a neighbourhood of $x=0$ (the paper's approach allows this to be replaced by a stabilizing nonlinear feedback law if available). Note that this formulation contains a degree of conservativeness since it leads to an optimization over the variables $\mathbf{c}_k$ rather than closed-loop policies, however it provides a convenient balance of computation and conservativeness.

Under the control law of (6), state predictions are governed by the model

$$x_{k+i+1|k}=\phi(x_{k+i|k},c_{i|k})+d_{k+i},\qquad x_{k|k}=x_k \tag{7}$$

where $\phi:\mathbb{R}^{n_x\times n_u}\to\mathbb{R}^{n_x}$ is defined by the identity

$$\phi(x,c)=f(x,Kx+c),\quad \forall x\in\mathbb{R}^{n_x},\ c\in\mathbb{R}^{n_u}.$$

In order to account efficiently for the nonlinearity and uncertainty in the prediction system (7), the proposed receding horizon optimization is based on linear models obtained from the Jacobian linearization of (7) around nominal trajectories for the predicted state. Let $\{x^0_{k|k},\ldots,x^0_{k+N|k}\}$ denote a trajectory for the nominal system associated with the expected value of uncertainty in (7) and $\mathbf{c}^0_k = \{c^0_{0|k},\ldots,c^0_{N-1|k}\}$, so that $x^0_{k+i|k}$ evolves according to

$$x^0_{k+i+1|k} = \phi(x^0_{k+i|k}, c^0_k), \qquad x^0_{k|k} = x_k. \tag{8}$$

The combined effects of approximation errors and unknown disturbances can be taken into account through the definition of a sequence of sets centred on a nominal trajectory at prediction times $i = 1,\ldots,N$ and a terminal set centred at the origin for $i > N$. For computational convenience we define these sets as low complexity polytopes of the form $\{x : |V(x-\hat{x}_{i|k})| \le \bar{z}_{i|k}\}$ for $i = 1,\ldots,N$, and $\{x : |Vx| \le \bar{z}_t\}$ for the terminal set. Here V is a square full-rank matrix and the parameters $\bar{z}_{i|k}, \bar{z}_t$ determine the relative scaling of the sets. Possible choices for V and K are discussed in Section 3.2.

To simplify presentation, we define a transformed variable $z = Vx$ and denote z^δ and c^δ as the deviations from the nominal trajectories for z and c:

$$z^\delta_{k+i|k} = z_{k+i|k} - z^0_{k+i|k}, \qquad z^0_{k+i|k} = Vx^0_{k+i|k}$$
$$c^\delta_{i|k} = c_{i|k} - c^0_{i|k}.$$

The transformed state evolves according to

$$z^0_{k+i+1|k} + z^\delta_{k+i+1|k} = V\phi\left(V^{-1}(z^0_{k+i|k} + z^\delta_{k+i|k}),\ c^0_{i|k} + c^\delta_{i|k}\right) + \varepsilon_{k+i} \tag{9}$$

where $\varepsilon_k = Vd_k$. The linearization of (7) about $\{x^0_{k|k},\ldots,x^0_{k+N|k}\}$ and $\mathbf{c}^0_k$ can therefore be expressed

$$z^\delta_{k+i+1|k} = \Phi_{k+i|k} z^\delta_{k+i|k} + B_{k+i|k} c^\delta_{i|k} + \varepsilon_{k+i} + e_{k+i|k}, \qquad z^\delta_{k|k} = 0 \tag{10}$$

where

$$\Phi_{k+i|k} = V\frac{\partial \phi}{\partial x}\bigg|_{(x^0_{k+i|k}, c^0_{i|k})} V^{-1} \qquad B_{k+i|k} = V\frac{\partial \phi}{\partial c}\bigg|_{(x^0_{k+i|k}, c^0_{i|k})}$$

Similarly, for $i \ge N$ we have $z_{k+i|k} = Vx_{k+i|k}$ where

$$z_{k+i+1|k} = V\phi\left(V^{-1}z_{k+i|k},\ 0\right) + \varepsilon_{k+i} \tag{11}$$

and the Jacobian linearization about $z = 0$ therefore gives

$$z_{k+i+1|k} = \Phi z_{k+i|k} + \varepsilon_{k+i} + e_{k+i|k}\ , \qquad \Phi = V\frac{\partial \phi}{\partial x}\bigg|_{(0,0)} V^{-1}. \tag{12}$$

Remark 1. *From Assumption 1 it follows that the linearization error in (10):*

$$e_{k+i|k} = V\phi\left(V^{-1}(z^0_{k+i|k} + z^\delta_{k+i|k}),\, c^0_{i|k} + c^\delta_{i|k}\right) - V\phi\left(V^{-1}z^0_{k+i|k}, c^0_{i|k}\right) - \Phi_{k+i|k}z^\delta_{k+i|k} - B_{k+i|k}c^\delta_{i|k}$$

necessarily satisfies the Lipschitz condition

$$|e_{k+i|k}| \leq \Gamma_z|z^\delta_{k+i|k}| + \Gamma_c|c^\delta_{i|k}| \tag{13}$$

for some positive matrices Γ_z, Γ_c, for all $(z^\delta_{k+i|k}, c^\delta_{i|k})$ such that

$$\left(V^{-1}(z^0_{k+i|k} + z^\delta_{k+i|k}),\, KV^{-1}(z^0_{k+i|k} + z^\delta_{k+i|k}) + c^0_{i|k} + c^\delta_{i|k}\right) \in \mathscr{X} \times \mathscr{U}.$$

Similarly, for $i \geq N$, the linearization error in (12):

$$e_{k+i|k} = V\phi\left(V^{-1}(z_{k+i|k}),\, 0\right) - \Phi z_{k+i|k}$$

is Lipschitz continuous, with

$$|e_{k+i|k}| \leq \Gamma_t|z_{k+i|k}| \tag{14}$$

for some positive matrix Γ_t, for all $z_{k+i|k}$ such that

$$\left(V^{-1}z_{k+i|k},\, KV^{-1}z_{k+i|k}\right) \in \mathscr{X} \times \mathscr{U}.$$

In Section 3 the bounds (13) and (14) are combined with bounds on ε_k to construct sets $\mathscr{Z}_{i|k}, i = 0, \ldots, N$ that depend on $\mathbf{c}^\delta_k = \{c^\delta_{0|k}, \ldots, c^\delta_{N-1|k}\}$, thus defining tubes centred on a nominal trajectory containing the predictions of (7). These tubes provide a means of bounding the receding horizon performance cost and of ensuring satisfaction of constraints. As a result, the process of successively linearizing about $(\{x^0_{k+i|k}\}, \mathbf{c}^0_k)$, optimizing $\mathbf{c}^\delta_k$, and then redefining $(\{x^0_{k+i|k}\}, \mathbf{c}^0_k)$ by setting $\mathbf{c}^0_k \leftarrow \mathbf{c}^0_k + \mathbf{c}^\delta_k$ necessarily converges to a (local) optimum for the original nonlinear dynamics, as discussed in Section 4.

3 Probabilistic Tubes

This section describes a method of constructing a series of tubes around a nominal predicted trajectory so that each tube contains the future predicted state with a prescribed probability. This process provides a means of bounding the predicted value of the cost (5) and of ensuring satisfaction of hard constraints (2) and probabilistic constraints (4) along future predicted trajectories. The probabilities of transition between tubes from one sampling instant to the next and the probability of constraint violation within each tube are governed by fixed probabilities that are determined offline. However the parameters determining the size of each tube are retained as

optimization variables, and this allows the effects of stochastic model uncertainty and linearization errors (which depend on the predicted input trajectory) to be estimated non-conservatively over the prediction horizon.

Let $\{\mathscr{S}_t^{(1)},\ldots,\mathscr{S}_t^{(r)}\}$ and $\{\mathscr{S}_{i|k}^{(1)},\ldots,\mathscr{S}_{i|k}^{(r)}\}$ for $i=1,\ldots,N$ denote collections of sets in $\mathbb{R}^{n_x}$ with

$$\mathscr{S}_t^{(j)}\cap\mathscr{S}_t^{(m)}=\emptyset,\quad \mathscr{S}_{i|k}^{(j)}\cap\mathscr{S}_{i|k}^{(m)}=\emptyset\quad\forall j\neq m, \tag{15}$$

and let $\mathscr{S}_{0|k}^{(j)}=0$ for all j. Let $p_{jm}\in[0,1]$ for $j,m=1,\ldots,r$, with

$$\sum_{j=1}^{r}p_{jm}=1\quad j=1,\ldots,r, \tag{16}$$

and assume that the sequences $\{z_{k+i|k}^{\delta},\ i=0,\ldots,N\}$ and $\{z_{k+i|k},\ i\geq N\}$ generated respectively by the prediction models of (10) and (12) satisfy

$$\Pr\big(z_{k+i+1|k}^{\delta}\in\mathscr{S}_{i+1|k}^{(j)}\mid z_{k+i|k}^{\delta}\in\mathscr{S}_{i|k}^{(m)}\big)=p_{jm}\quad i=0,\ldots,N \tag{17a}$$

$$\Pr\big(z_{k+i+1|k}\in\mathscr{S}_{t}^{(j)}\mid z_{k+i|k}\in\mathscr{S}_{t}^{(m)}\big)=p_{jm}\quad i=N,N+1,\ldots \tag{17b}$$

(note that the requirement for these probabilities to hold with equality is relaxed in Section 3.1). Let the sets $\mathscr{S}_{N|k}^{(j)}$ be linked to the terminal sets $\mathscr{S}_t^{(j)}$ through

$$z_{k+N|k}^{\delta}\in\mathscr{S}_{N|k}^{(j)}\implies z_{k+N|k}=z_{k+N|k}^{0}+z_{k+N|k}^{\delta}\in\mathscr{S}_t^{(j)}\quad j=1,\ldots,r. \tag{18}$$

Then the probabilities of $z_{k+i|k}^{\delta}\in\mathscr{S}_{i|k}^{(j)}$ for $i=1,\ldots,N$ and of $z_{k+i|k}^{\delta}\in\mathscr{S}_t^{(j)}$ for $i>N$ are governed by a Markov chain model with transition matrix Π:

$$\Pi=\begin{bmatrix}p_{11}&\cdots&p_{1r}\\ \vdots&\ddots&\vdots\\ p_{r1}&\cdots&p_{rr}\end{bmatrix}$$

and the distribution of predicted states can be approximated using the property

$$\begin{bmatrix}p_i^{(1)}\\ p_i^{(2)}\\ \vdots\\ p_i^{(r)}\end{bmatrix}=\Pi^i e_1,\quad p_i^{(j)}=\begin{cases}\Pr\big(z_{k+i|k}^{\delta}\in\mathscr{S}_{i|k}^{(j)}\big) & i=1,\ldots,N\\ \Pr\big(z_{k+i|k}\in\mathscr{S}_t^{(j)}\big) & i=N,N+1,\ldots\end{cases} \tag{19}$$

where $e_1=\begin{bmatrix}1&0&\cdots&0\end{bmatrix}^T$.

Define p_j as a bound on the one-step-ahead conditional probability of violating the soft constraint (3):

$$\Pr\left(F_S x_{k+i+1|k} + G_S u_{k+i+1|k} > h_S \mid z^{\delta}_{k+i|k} \in \mathscr{S}^{(j)}_{i|k}\right) \le p_j \quad i < N \tag{20a}$$

$$\Pr\left(F_S x_{k+i+1|k} + G_S u_{k+i+1|k} > h_S \mid z_{k+i|k} \in \mathscr{S}^{(j)}_{t}\right) \le p_j \quad i \ge N \tag{20b}$$

Then, from (15), (16) and (19), we have

$$\Pr(F_S x_{k+i+1|k} + G_S u_{k+i+1|k} > h_S) \le \begin{bmatrix} p_1 & p_2 & \cdots & p_r \end{bmatrix} \Pi^i e_1 \ \ \forall i. \tag{21}$$

Assume also that the hard constraints (2) are satisfied within $\mathscr{S}^{(j)}_{i|k}$ and $\mathscr{S}^{(j)}_{t}$:

$$z^{\delta}_{k+i|k} \in \mathscr{S}^{(j)}_{i|k} \implies F_H x_{k+i|k} + G_H u_{k+i|k} \le h_H \quad i < N \tag{22a}$$

$$z_{k+i|k} \in \mathscr{S}^{(j)}_{t} \implies F_H x_{k+i|k} + G_H u_{k+i|k} \le h_H \quad i \ge N \tag{22b}$$

for $j = 1, \ldots, r$. Then sufficient conditions for satisfaction of both hard and probabilistic constraints are given by the following lemma.

Lemma 3.1. *The constraints of (2) and (4) are necessarily satisfied along predicted state and input trajectories of (6)-(7) if the conditions on: transition probabilities (17a,b), terminal sets (18), probabilities of soft constraint violation (20a,b), and hard constraints (22a,b), are satisfied for Π and p_j, $j = 1, \ldots, r$ such that:*

$$\frac{1}{N_c} \sum_{i=0}^{N_c - 1} \begin{bmatrix} p_1 & p_2 & \cdots & p_r \end{bmatrix} \Pi^i e_1 \le \frac{N_{\max}}{N_c}. \tag{23}$$

Proof. This is a direct consequence of (21) and (22a,b). ■

Throughout the following development we assume that Π and p_j satisfy (23).

3.1 Tube Constraints

We next construct constraints that ensure satisfaction of (2) and (4), and which are suitable for a receding horizon control law. Consider the nested sets:

$$\mathscr{Z}^{(1)}_{i|k} \subseteq \mathscr{Z}^{(2)}_{i|k} \subseteq \cdots \subseteq \mathscr{Z}^{(r)}_{i|k}, \quad \mathscr{Z}^{(1)}_{t} \subseteq \mathscr{Z}^{(2)}_{t} \subseteq \cdots \subseteq \mathscr{Z}^{(r)}_{t} \tag{24}$$

The effects of model uncertainty and linearization error cause the uncertainty in $z^{\delta}_{k+i|k}$ to be symmetric about the state $\hat{z}_{i|k}$ of the linear model:

$$\hat{z}_{i+1|k} = \Phi_{k+i|k} \hat{z}_{i|k} + B_{k+i|k} c^{\delta}_{i|k} \qquad \hat{z}_{0|k} = 0. \tag{25}$$

Therefore define $\{\mathscr{Z}^{(j)}_{i|k}\}$ and $\{\mathscr{Z}^{(j)}_{t}\}$ as the low-complexity polytopic sets:

$$\mathscr{Z}^{(j)}_{i|k} = \{z^{\delta} = \hat{z}_{i|k} + v : |v| \le \bar{z}^{(j)}_{i|k}\} \qquad \mathscr{Z}^{(j)}_{t} = \{z : |z| \le \bar{z}^{(j)}_{t}\}$$

and define $\mathscr{S}^{(j)}_{t}$ and $\mathscr{S}^{(j)}_{i|k}$ in terms of $\mathscr{Z}^{(j)}_{t}$ and $\mathscr{Z}^{(j)}_{i|k}$ via the relations:

$$\mathscr{S}_t^{(j)} = \begin{cases} \mathscr{Z}_t^{(1)} & j=1 \\ \mathscr{Z}_t^{(j)} - \mathscr{Z}_t^{(j-1)} & j=2,\ldots,r \end{cases} \qquad \mathscr{S}_{i|k}^{(j)} = \begin{cases} \mathscr{Z}_{i|k}^{(1)} & j=1 \\ \mathscr{Z}_{i|k}^{(j)} - \mathscr{Z}_{i|k}^{(j-1)} & j=2,\ldots,r \end{cases}$$

for $i=1,\ldots,N$. The transition probabilities in (17a,b) are assumed to hold with equality, which places a strong and unrealistic restriction on the distribution of the uncertain disturbance in (7). Here we remove this assumption by instead imposing constraints on transition probabilities for $\mathscr{Z}_t^{(j)}$ and $\mathscr{Z}_{i|k}^{(j)}$. These constraints have the additional advantage over constraints invoked directly on $\mathscr{S}_t^{(j)}$ and $\mathscr{S}_{i|k}^{(j)}$ that they are convex (in fact linear in the degrees of freedom). We show that, when combined with conditions on the violation of system constraints (2) and (3), this formulation provides sufficient conditions for the conditions of Lemma 3.1 for disturbances d_k with general (continuous, finitely supported) distributions.

Accordingly, let

$$\tilde{p}_{jm} = \sum_{l=1}^{j} p_{lm}, \quad j,m=1,\ldots,r$$

(so that $\tilde{p}_{rm}=1$, $m=1,\ldots,r$) and define

$$\tilde{\Pi} = T\Pi, \quad \tilde{\Pi} = \begin{bmatrix} \tilde{p}_{11} & \cdots & \tilde{p}_{1r} \\ \vdots & \ddots & \vdots \\ \tilde{p}_{r1} & \cdots & \tilde{p}_{rr} \end{bmatrix}, \quad T = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & & 0 \\ \vdots & \vdots & \ddots & \\ 1 & 1 & \cdots & 1 \end{bmatrix}.$$

For $j=1,\ldots,r-1$ and $m=1,\ldots,r$, we impose the transition probabilities

$$\Pr\left(z^{\delta}_{k+i+1|k} \in \mathscr{Z}^{(j)}_{i+1|k} \mid z^{\delta}_{k+i|k} \in \mathscr{Z}^{(m)}_{i|k}\right) \geq \tilde{p}_{jm} \quad i=0,\ldots,N \tag{26a}$$

$$\Pr\left(z_{k+i+1|k} \in \mathscr{Z}^{(j)}_{t} \mid z_{k+i|k} \in \mathscr{Z}^{(m)}_{t}\right) \geq \tilde{p}_{jm} \quad i=N,N+1,\ldots \tag{26b}$$

whereas for $m=1,\ldots,r$ we require

$$\Pr\left(z^{\delta}_{k+i+1|k} \in \mathscr{Z}^{(r)}_{i+1|k} \mid z^{\delta}_{k+i|k} \in \mathscr{Z}^{(m)}_{i|k}\right) = 1 \quad i=0,\ldots,N \tag{27a}$$

$$\Pr\left(z_{k+i+1|k} \in \mathscr{Z}^{(r)}_{t} \mid z_{k+i|k} \in \mathscr{Z}^{(m)}_{t}\right) = 1 \quad i=N,N+1,\ldots \tag{27b}$$

The required probabilities on soft constraints are invoked for $j=1,\ldots,r$ by

$$\Pr\left(F_S x_{k+i+1|k} + G_S u_{k+i+1|k} > h_S \mid z^{\delta}_{k+i|k} \in \mathscr{Z}^{(j)}_{i|k}\right) \leq p_j,\ i=0,\ldots,N-1 \tag{28a}$$

$$\Pr\left(F_S x_{k+i+1|k} + G_S u_{k+i+1|k} > h_S \mid z_{k+i|k} \in \mathscr{Z}^{(j)}_{t}\right) \leq p_j,\ i=N,N+1,\ldots \tag{28b}$$

while the hard constraints are invoked via

$$z^{\delta}_{k+i|k} \in \mathscr{Z}^{(r)}_{i|k} \implies F_H x_{k+i|k} + G_H u_{k+i|k} \leq h_H \quad i=0,\ldots,N-1 \tag{29a}$$

$$z_{k+i|k} \in \mathscr{Z}^{(r)}_{t} \implies F_H x_{k+i|k} + G_H u_{k+i|k} \leq h_H \quad i=N,N+1,\ldots \tag{29b}$$

Lemma 3.2. *If p_j and $\tilde{p}_{jm}$ satisfy*

$$p_j \le p_{j+1}, \qquad j = 1,\dots,r-1 \tag{30a}$$
$$\tilde{p}_{jm} \ge \tilde{p}_{jm+1}, \quad j = 1,\dots,r-1. \tag{30b}$$

then constraints (26), (27), (28) and (29), together with the terminal constraints that $z^0_{k+N|k} + z^\delta_{k+N|k} \in \mathscr{Z}_t^{(j)}$ for all $z^\delta_{k+N|k} \in \mathscr{Z}^{(j)}_{N|k}$, $j = 1,\dots,r$, are sufficient to ensure that (2) and (4) hold along predicted trajectories of (6)-(7).

Proof. Satisfaction of the hard constraint (2) is trivially ensured by (29a,b) and (27a,b) due to the nested property (24). On the other hand, satisfaction of (21), and hence also the probabilistic constraint (4), can be shown using (30a,b). For $i = 0$ this is obvious from (28), whereas for $i = 1$ we have from (28):

$$\begin{aligned}
&\Pr(F_S x_{k+2|k} + G_S u_{k+2|k} > h_S) \\
&\quad \le \begin{bmatrix} p_1 & \cdots & p_r \end{bmatrix} \left[\Pr\left(x_{k+1|k} \in \mathscr{S}^{(1)}_{1|k}\right) \ \cdots \ \Pr\left(x_{k+1|k} \in \mathscr{S}^{(r)}_{1|k}\right) \right]^T \\
&\quad = \begin{bmatrix} p_1 - p_2 & \cdots & p_r \end{bmatrix} \left[\Pr\left(x_{k+1|k} \in \mathscr{Z}^{(1)}_{1|k}\right) \ \cdots \ \Pr\left(x_{k+1|k} \in \mathscr{Z}^{(r)}_{1|k}\right) \right]^T \\
&\quad \le \begin{bmatrix} p_1 - p_2 & \cdots & p_r \end{bmatrix} \tilde{\Pi}\, e_1 \\
&\quad = \begin{bmatrix} p_1 & \cdots & p_r \end{bmatrix} \Pi\, e_1
\end{aligned}$$

where the last inequality follows from (30a) and (26a). Similarly, for $i = 2$:

$$\begin{aligned}
&\Pr(F_S x_{k+3|k} + G_S u_{k+3|k} > h_S) \\
&\quad \le \begin{bmatrix} p_1 & \cdots & p_r \end{bmatrix} \left[\Pr\left(x_{k+2|k} \in \mathscr{S}^{(1)}_{2|k}\right) \ \cdots \ \Pr\left(x_{k+2|k} \in \mathscr{S}^{(r)}_{2|k}\right) \right]^T \\
&\quad = \begin{bmatrix} p_1 - p_2 & \cdots & p_r \end{bmatrix} \left[\Pr\left(x_{k+2|k} \in \mathscr{Z}^{(1)}_{2|k}\right) \ \cdots \ \Pr\left(x_{k+2|k} \in \mathscr{Z}^{(r)}_{2|k}\right) \right]^T \\
&\quad \le \begin{bmatrix} p_1 - p_2 & \cdots & p_r \end{bmatrix} \tilde{\Pi} \left[\Pr\left(x_{k+1|k} \in \mathscr{S}^{(1)}_{1|k}\right) \ \cdots \ \Pr\left(x_{k+1|k} \in \mathscr{S}^{(r)}_{1|k}\right) \right]^T \\
&\quad = \begin{bmatrix} p_1 - p_2 & \cdots & p_r \end{bmatrix} \tilde{\Pi} T^{-1} \left[\Pr\left(x_{k+1|k} \in \mathscr{Z}^{(1)}_{1|k}\right) \ \cdots \ \Pr\left(x_{k+1|k} \in \mathscr{Z}^{(r)}_{1|k}\right) \right]^T \\
&\quad \le \begin{bmatrix} p_1 - p_2 & \cdots & p_r \end{bmatrix} \tilde{\Pi} T^{-1} \tilde{\Pi}\, e_1 \\
&\quad = \begin{bmatrix} p_1 & \cdots & p_r \end{bmatrix} \Pi^2 e_1
\end{aligned}$$

where the last inequality follows from (30b) (which implies the matrix $\tilde{\Pi} T^{-1}$ has non-negative elements in the first $r-1$ rows and $\begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix}$ in the last row) and (26a). The same arguments show that (21) also holds for all $i > 2$. ∎

Remark 2. *The condition (30a) is equivalent to requiring that the probability of soft constraint violation should decrease towards the centre of the tube. This is necessarily true for the linear soft constraints of (3) due to the nested property (24) and the convexity of $\mathscr{Z}_t^{(j)}$ and $\mathscr{Z}^{(j)}_{i|k}$. Furthermore, because of the linearity of (10)*

and (12), the nestedness and convexity of $\mathscr{Z}_t^{(j)}$ and $\mathscr{Z}_{i|k}^{(j)}$ imply that condition (30b) can also be assumed to hold without loss of generality.

To invoke (26)-(29) we use confidence intervals for the elements of $\varepsilon = Vd$ in (10) inferred from the distribution for d:

$$\Pr(|\varepsilon| \le \xi_j) = \tilde{p}_j, \quad \Pr(|\varepsilon| \le \xi_{jm}) = \tilde{p}_{jm}, \quad j,m = 1,\dots,r \tag{31a}$$

$$\Pr(|\varepsilon| \le \bar{\xi}) = 1. \tag{31b}$$

From (10) and (13) and (25) we obtain the bounds

$$|z^{\delta}_{k+i+1|k} - \hat{z}_{i+1|k}| \le |\Phi_{k+i|k}(z^{\delta}_{k+i|k} - \hat{z}_{i|k})| + \Gamma_z|z^{\delta}_{k+i|k}| + \Gamma_c|c^{\delta}_{i|k}| + |\varepsilon_{k+i}|$$

and, since $\mathscr{Z}_{i|k}^{(m)}$ has vertices $\hat{z}_{i|k} + D_p\bar{z}_{i|k}^{(m)}$ (where D_p, $p = 1,\dots,2^{n_x}$ are appropriate diagonal matrices), (26a) is therefore implied by the condition

$$\bar{z}^{(j)}_{i+1|k} \ge |\Phi_{k+i|k}D_p\bar{z}^{(m)}_{i|k}| + \Gamma_z|\hat{z}_{i|k} + D_p\bar{z}^{(m)}_{i|k}| + \Gamma_c|c^{\delta}_{i|k}| + \xi_{jm} \tag{32}$$

for $p = 1,\dots,2^{n_x}$, while (27a) is implied by

$$\bar{z}^{(r)}_{i+1|k} \ge |\Phi_{k+i|k}D_p\bar{z}^{(m)}_{i|k}| + \Gamma_z|\hat{z}_{i|k} + D_p\bar{z}^{(m)}_{i|k}| + \Gamma_c|c^{\delta}_{i|k}| + \bar{\xi} \tag{33}$$

for $p = 1,\dots,2^{n_x}$. Similarly, from (10), (13) and (25) it follows that sufficient conditions for (28a) are given by

$$\begin{aligned}(F_S + G_SK)V^{-1}(z^0_{k+i+1|k} + \hat{z}_{i+1|k} + \Phi_{k+i|k}D_p\bar{z}^{(j)}_{i|k}) + G_S(c^0_{i+1|k} + c^{\delta}_{i+1|k}) \\ + |(F_S + G_SK)V^{-1}|(\Gamma_z|\hat{z}_{i|k} + D_p\bar{z}^{(j)}_{i|k}| + \Gamma_c|c^{\delta}_{i|k}| + \xi_j) \le h_S\end{aligned} \tag{34}$$

for $p = 1,\dots,2^{n_x}$, whereas (29a) is implied by

$$(F_H + G_HK)V^{-1}(z^0_{k+i|k} + \hat{z}_{i|k} + D_p\bar{z}^{(r)}_{i|k}) + G_H(c^0_{i|k} + c^{\delta}_{i|k}) \le h_H \tag{35}$$

for $p = 1,\dots,2^{n_x}$. Note that the conditions (32)-(35) are linear in $\bar{z}^{(j)}_{i|k}$ and $c_{i|k}$, which are retained as variables in the online optimization described in Section 4.

3.2 Terminal Sets and Terminal Cost

In the interests of optimizing predicted performance, K in (6) should be optimal for the cost (5) when constraints are inactive. However the constraint (27b) also requires that $\mathscr{Z}_t^{(r)}$ is robustly invariant under (12), and this may conflict with the requirement for unconstrained optimality. We therefore specify K as optimal for the linearized model $(\partial f/\partial x|_{(0,0)}, \partial f/\partial u|_{(0,0)})$ with a suitable quadratic cost, and define V in (12) as the transformation matrix such that $\Phi = V\partial\phi/\partial x|_{(0,0)}V^{-1}$ is in modal form (see [4] for more details of this approach).

To maximize the region of attraction of the resulting receding horizon control law, it is desirable to maximize the terminal sets $\mathscr{Z}_t^{(j)}$. This suggests the following offline optimization problem:

$$(\bar{z}_t^{(1)},\dots,\bar{z}_t^{(r)}) = \arg \max_{(\bar{z}_t^{(1)},\dots,\bar{z}_t^{(r)})} \prod_{j=1}^{r} \mathrm{vol}(\mathscr{Z}_t^{(j)}) \tag{36a}$$

$$\text{s.t.} \quad \bar{z}_t^{(r)} \geq \bar{z}_t^{(r-1)} \geq \cdots \geq \bar{z}_t^{(1)} > 0 \tag{36b}$$

$$\bar{z}_t^{(j)} \geq (|\Phi| + \Gamma_t)\bar{z}_t^{(m)} + \xi_{jm}, \quad m = 1,\dots,r,\ j = 1,\dots,r-1 \tag{36c}$$

$$\bar{z}_t^{(r)} \geq (|\Phi| + \Gamma_t)\bar{z}_t^{(m)} + \bar{\xi}, \quad m = 1,\dots,r \tag{36d}$$

$$|(F_S + G_S K)V^{-1}|\left\{(|\Phi| + \Gamma_t)\bar{z}_t^{(j)} + \xi_j\right\} \leq h_S, \quad j = 1,\dots,r \tag{36e}$$

$$|(F_H + G_H K)V^{-1}|\bar{z}_t^{(r)} \leq h_H \tag{36f}$$

where (36b) ensures (24), (36c) and (36d) are sufficient for (26b) and (27b) respectively, while (36e) and (36f) are sufficient for (28b) and (29b) respectively. The objective (36a) is chosen so that the optimization problem is convex, but could be modified by introducing weights in order to obtain a more favourable solution for $\mathscr{Z}_t^{(j)}$, $j = 1,\dots,r$.

To obtain a finite value for the infinite horizon predicted cost despite the presence of non-decaying disturbances, we subtract a bound on the steady-state value of the stage cost under (6), and hence redefine the performance index as

$$J(\mathbf{c}_k, x_k) = \sum_{i=0}^{\infty} \mathbb{E}_k\left(\mathbf{1}^T|V^{-1}z_{k+i|k}| + \lambda\mathbf{1}^T|KV^{-1}z_{k+i|k} + c_{i|k}| - l_{ss}\right) \tag{37a}$$

$$l_{ss} = \mathbf{1}^T\left(|V^{-1}| + \lambda|KV^{-1}|\right)\left(I - |\Phi| - \Gamma_t\right)^{-1}\zeta \tag{37b}$$

where $\zeta = \mathbb{E}(|\varepsilon|)$. The following result enables the cost over the prediction interval $i = N, N+1,\dots$ to be bounded in terms of a function of $z_{k+N|k}$.

Lemma 3.3. *If q satisfies*

$$q^T(|z| - |\Phi z| - \Gamma_t|z| - \zeta) \geq \mathbf{1}^T|V^{-1}z| + \lambda\mathbf{1}^T|KV^{-1}z| - l_{ss} \tag{38}$$

for all $z \in \mathscr{Z}_t^{(r)}$, then

$$q^T|z_{k+N|k}| \geq \sum_{i=N}^{\infty} \mathbb{E}_k\left(\mathbf{1}^T|V^{-1}z_{k+i|k}| + \lambda\mathbf{1}^T|KV^{-1}z_{k+i|k}| - l_{ss}\right). \tag{39}$$

Proof. From (12), (14) and (31b), the inequality (38) implies

$$q^T|z_{k+i|k}| - \mathbb{E}_{k+i}(q^T|z_{k+i+1|k}|) \geq \mathbf{1}^T|V^{-1}z| + \lambda\mathbf{1}^T|KV^{-1}z| - l_{ss}.$$

Taking expectations and summing over $i = N, N+1,\dots$ yields (39). ∎

Using Lemma 3.3 we determine an optimal bound on the cost-to-go for the case that $z_{k+N|k} \in \mathscr{Z}_t^{(j)}$ by solving the following LPs for $q^{(j)}$, $j = 1,\ldots,r$:

$$\begin{aligned} q^{(j)} = \arg\min_{q} \;& q^T \bar{z}_t^{(j)} \\ \text{s.t. } & q^T(\bar{z}_t^{(r)} - |\Phi D_p \bar{z}_t^{(r)}| - \Gamma_t \bar{z}_t^{(r)} - \zeta) \geq \\ & \mathbf{1}^T |V^{-1} D_p \bar{z}_t^{(r)}| + \lambda \mathbf{1}^T |KV^{-1} D_p \bar{z}_t^{(r)}| - l_{ss}, \quad p = 1,\ldots,2^{n_x} \end{aligned} \tag{40}$$

Given the distribution of predictions (19), this implies the following bound

$$\sum_{j=1}^{r} q^{(j)T} \bar{z}_{k+N|k}^{(j)} p_N^{(j)} \geq \sum_{i=N}^{\infty} \mathbb{E}_k\left(\mathbf{1}^T |V^{-1} z_{k+i|k}| + \lambda \mathbf{1}^T |KV^{-1} z_{k+i|k}| - l_{ss}\right).$$

4 Receding Horizon Control Law

Let V be the following bound on the cost J in (37a):

$$\begin{aligned} & V(\mathbf{c}_k^{\delta}, \{\bar{z}_{i|k}^{(j)}, i = 1,\ldots,N, j = 1,\ldots,r\}, \{x_{k+i|k}^0\}, \mathbf{c}_k^0) = \\ & \sum_{j=1}^{r} \Bigg\{ \sum_{i=0}^{N-1} \max_{z_{i|k}^{\delta} \in \mathscr{Z}_{i|k}^{(j)}} \left(\mathbf{1}^T |V^{-1} z_{k+i|k}| + \lambda \mathbf{1}^T |KV^{-1} z_{k+i|k} + c_{i|k}^0 + c_{i|k}^{\delta}| - l_{ss}\right) p_i^{(j)} \\ & \qquad + q^{(j)T} \bar{z}_{k+N|k}^{(j)} p_N^{(j)} \Bigg\} \end{aligned} \tag{41}$$

and consider the following receding horizon control strategy.

Algorithm 1. ***Offline:*** *Given $p_j, \tilde{p}_{jm}$ satisfying (23), (30a,b): compute K, V, terminal sets $\mathscr{Z}_t^{(j)}$ and terminal weights $q^{(j)}$ using the procedures of Section 3.2.* ***Online:*** *At times $k = 0, 1, \ldots$*
1. given $\mathbf{c}_k^0$, determine $x_{k+i|k}^0$, and $\Phi_{k+i|k}, B_{k+i|k}$, $i = 0,\ldots,N$ and solve:

$$\mathbf{c}_k^{\delta *} = \arg \min_{\mathbf{c}_k^{\delta}, \{\bar{z}_{i|k}^{(j)}\}} \quad V(\mathbf{c}_k^{\delta}, \{\bar{z}_{i|k}^{(j)}\}, \{x_{k+i|k}^0\}, \mathbf{c}_k^0) \tag{42a}$$

$$s.t. \quad (32), (33), (34), (35) \tag{42b}$$

$$\mathscr{Z}_{k+N|k}^{(j)} + z_{k+N|k}^0 \subseteq \mathscr{Z}_t^{(j)}, \quad j = 1,\ldots,r \tag{42c}$$

2. set $u_k = Kx_k + c_{0|k}^0 + c_{0|k}^{\delta *}$ and $\mathbf{c}_{k+1}^0 = \{c_{1|k}^0 + c_{1|k}^{\delta *}, \ldots, c_{N-1|k}^0 + c_{N-1|k}^{\delta *}, 0\}$.

Theorem 4.1. *In closed-loop operation, Algorithm 1 has the properties:*
(i). the optimization (42) is feasible for all $k > 0$ if feasible at $k = 0$
(ii). the optimal value $V^*(\{x_{k+i|k}^0\}, \mathbf{c}_k^0)$ of the objective (42a) satisfies

$$\mathbb{E}_k\left[V^*(\{x_{k+i+1|k+1}^0\}, \mathbf{c}_{k+1}^0)\right] - V^*(\{x_{k+i|k}^0\}, \mathbf{c}_k^0) \leq l_{ss} - \mathbf{1}^T |x_k| - \lambda \mathbf{1}^T |u_k| \tag{43}$$

(iii). constraints (2) and (4) are satisfied at all times k and

$$\lim_{n\to\infty}\frac{1}{n}\sum_{k=0}^{n}\mathbb{E}_0\left(\mathbf{1}^T|x_k|+\lambda\mathbf{1}^T|u_k|\right)\leq l_{ss}. \tag{44}$$

Proof. (i) and (ii) follow from feasibility of $\mathbf{c}_k^{\delta}=0$ in (42). Constraint satisfaction in (iii) follows from (i), and (44) results from summing (43) over $0\leq k\leq n$ and noting that $V^*(\{x_{i|0}^0\},\mathbf{c}_0^0)$ is finite. ■

Remark 3. *The optimization (42) to be solved online in Algorithm 1 can be formulated as a linear program. Note that the number of constraints in (42) depends linearly on the horizon N and the number of tubes r, but grows exponentially with the dimension of the model state n_x due to the exponential growth in the number of vertices of $\mathscr{Z}_{i|k}^{(j)}$ with n_x. The required online computation therefore grows rapidly with model size. Possible methods of mitigating this growth in computational load are the use of tubes with ellipsoidal cross-sections and the use of robust optimization methods [2] to invoke constraints on the probabilities of transition between tubes and of constraint satisfaction within tubes.*

Remark 4. *If the constraints on online computation allow for more than one optimization at each sample, then setting $\mathbf{c}_k^0\leftarrow\mathbf{c}_k^0+\mathbf{c}_k^{\delta *}$ and repeating step 1 results in non-increasing optimal cost values $V^*(\{x_{k+i|k}^0\},\mathbf{c}_k^0)$. This process generates a sequence of iterates $\mathbf{c}_k^{\delta *}$ that converges to an optimum point for the problem of minimizing (41) for the nonlinear dynamics (7) at time k.*

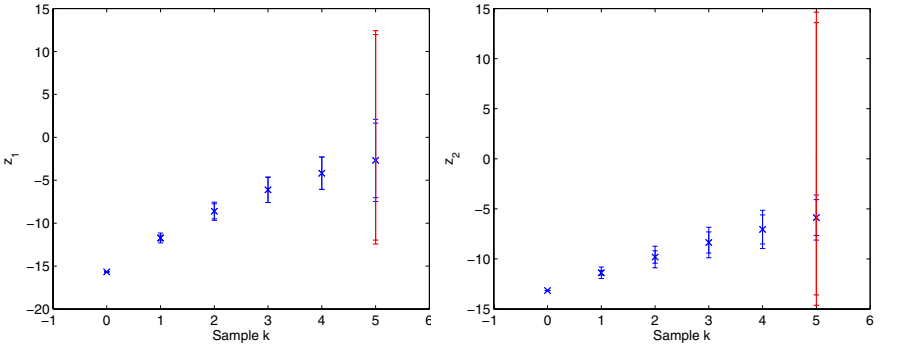


Fig. 1 Evolution of bounds on $e_1^T z$ (left) and $e_2^T z$ (right): $\mathscr{Z}_{i|k}^{(j)}$ (blue), $\mathscr{Z}_t^{(j)}$ (red), for $j=1,2$, and the nominal trajectory $z_{k+i|k}^0$ (blue x)

5 Example

The levels $h_1=x_1+x_1^r$ and $h_2=x_2+x_2^r$ of fluid in a pair of coupled tanks are related to the input flow-rate through the discrete-time system model:

$$\begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \end{bmatrix} = \begin{bmatrix} x_{1,k} - Ta_1\sqrt{h_{1,k} - h_{2,k}} \\ x_{2,k} + Ta_1\sqrt{h_{1,k} - h_{2,k}} - Ta_2\sqrt{h_{2,k}} \end{bmatrix} + \frac{T}{C_1}\begin{bmatrix} u_k + u^r \\ 0 \end{bmatrix} + \begin{bmatrix} d_{1,k} \\ d_{2,k} \end{bmatrix}$$

with $a_1 = 0.0690$, $a_2 = 0.0518$, $C_1 = 159.3\text{cm}^2$, sampling interval $T = 10\text{s}$, and where $x_1^r = 30.14\text{cm}$ and $x_2^r = 19.29\text{cm}$ are setpoints corresponding to flow-rate $u_r = 35\text{cm}^3/\text{s}$. The manipulated variable is the flow-rate u_k into tank 1, and d_{1k}, d_{2k} are zero-mean random disturbances with normal distributions truncated at the 95% confidence level. The system has probabilistic constraints: $\Pr(|x_{1k}| > 16) \leq 0.2$ and hard constraints: $|x_{1k}| < 16$, $0 \leq u_k \leq 70$. For the operating region: $|x_i| < 30$, $i = 1,2$, the Lipschitz constants were obtained as $\Gamma_z = \left[\begin{smallmatrix} 0.79 & 0.14 \\ 0.04 & 0.87 \end{smallmatrix}\right]$. Choosing $r = 2$ and $(p_{11}, p_{12}, p_{21}, p_{22}) = (0.8, 0.1, 0.2, 0.9)$, terminal sets $\mathscr{Z}_t^{(1)}, \mathscr{Z}_t^{(2)}$ and cost weights $q^{(1)}, q^{(2)}$ were computed offline. For a horizon $N = 5$, the sequence of sets $\mathscr{Z}_{i|k}^{(1)}, \mathscr{Z}_{i|k}^{(2)}$, $i = 0, \ldots, 5$, obtained with one iteration of Algorithm 1 are shown in Figure 1.

References

1. Batina, I., Stoorvogel, A.A., Weiland, S.: Optimal control of linear, stochastic systems with state and input constraints. In: Proc. IEEE Conf. Decision and Control, pp. 1564–1569 (2002)
2. Diehl, M., Bock, H.G., Kostina, E.: An approximation technique for robust nonlinear optimization. Math. Program. Ser. B 107, 213–230 (2006)
3. Langson, W., Chryssochoos, I., Rakovic, S.V., Mayne, D.Q.: Robust model predictive control using tubes. Automatica 40(1), 125–133 (2004)
4. Lee, Y.I., Kouvaritakis, B., Cannon, M.: Constrained receding horizon predictive control for nonlinear systems. Automatica 38(12), 2093–2102 (2002)
5. Mayne, D., Rawlings, J., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: Stability and optimality. Automatica 36(6), 789–814 (2000)
6. Primbs, J.A.: Stochastic receding horizon control of constrained linear systems with state and control multiplicative noise. In: Proc. American Control Conf., pp. 4470–4475 (2007)
7. van Hessem, D.H., Bosgra, O.H.: A conic reformulation of model predictive control including bounded and stochastic disturbances under state and input constraints. In: Proc. IEEE Conf. Decision and Control, pp. 4643–4648 (2002)
8. Yan, J., Bitmead, R.: Incorporating state estimation into model predictive control and its application to network traffic control. Automatica 41(4) (2005)

Sequential Monte Carlo for Model Predictive Control

N. Kantas, J.M. Maciejowski, and A. Lecchini-Visintini

Abstract. This paper proposes the use of Sequential Monte Carlo (SMC) as the computational engine for general (non-convex) stochastic Model Predictive Control (MPC) problems. It shows how SMC methods can be used to find global optimisers of non-convex problems, in particular for solving open-loop stochastic control problems that arise at the core of the usual receding-horizon implementation of MPC. This allows the MPC methodology to be extended to nonlinear non-Gaussian problems. We illustrate the effectiveness of the approach by means of numerical examples related to coordination of moving agents.

Keywords: Stochastic optimisation, Stochastic MPC, Sequential Monte Carlo.

1 Introduction

Nonlinear Model Predictive Control (MPC) usually involves non-convex optimisation problems, which in general suffer from the existence of several or even many local minima or maxima. This motivates the use of global optimisation algorithms, which guarantee asymptotic convergence to a global optimum. In most cases such algorithms employ a randomised search strategy to ensure that the search process is not trapped in some local mode. A popular example is Simulated Annealing (SA). Apart from the issue of multi-modalities of costs or rewards, solving such problems becomes even more complicated when stochastic processes are used to represent model uncertainties. In general, stochastic decision problems involve nonlinear dynamics with arbitrary distributions on general state spaces. In this paper we are mostly interested in continuous state spaces. Furthermore, the costs or rewards are

N. Kantas and J.M. Maciejowski
Cambridge University Engineering Dept., Cambridge, CB2 1PZ, UK
e-mail: {nk234,jmm}@cam.ac.uk

A. Lecchini-Visintini
Dept. of Engineering, University of Leicester, Leicester, LE1 7RH, UK
e-mail: alv1@leicester.ac.uk

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 263–273.
springerlink.com

usually expressed as expectations over relatively high-dimensional spaces. Monte Carlo methods are currently the most successful methods for evaluating such expectations under very weak assumptions, and have been widely applied in many areas such as finance, robotics, communications etc. An interesting point, which is overlooked often by the control community, is that Monte Carlo has also been applied for performing global optimisation, mainly in inference problems such as Maximum Likelihood or Maximum a Posteriori estimation, as presented recently in [1, 9, 13].

Still, solving stochastic optimal control problems on continuous state spaces for nonlinear non-Gaussian models is a formidable task. Solutions can be obtained by solving Dynamic Programming/ Bellman equations [3], but there is no analytical solution to this equation — except in very specific cases, such as finite state spaces or linear Gaussian state-space models with quadratic costs. In general, the value function takes as argument a probability distribution, and it is extremely difficult to come up with any sensible approximation to it. This is why, despite numerous potential applications, the literature on applications of Monte Carlo methods for control of non linear non Gaussian models is extremely limited [2].

MPC combined with Monte Carlo methods provides a natural approximation of solving the Bellman equation in the stochastic case, just as deterministic MPC can be viewed as a natural approximate method for solving deterministic optimal control problems [12]. For details of how MPC relates to dynamic programming and the Bellman equation, with emphasis on the stochastic case, see [4].

The most developed approaches for exploiting Monte Carlo methods for optimisation are based on either Markov Chain Monte Carlo (MCMC) methods [15], or Sequential Monte Carlo (SMC) methods [5, 7]. Considerable theoretical support exists for both MCMC and SMC under very weak assumptions, including general convergence results and central limit theorems [15, 5].

To date the control community has investigated the use of MCMC as a tool for evaluating approximate value functions, and SMC, in the guise of 'particle filters', for state estimation — see [14] for a setting closely related to MPC. Recently, in [10, 11] the authors proposed to use a MCMC algorithm similar to Simulated Annealing developed in [13], for sampling from a distribution of the maximisers of a finite-horizon open-loop problem, as the key component of an MPC-like receding-horizon strategy. As in any stochastic optimisation algorithm, the long execution times needed imply that these methods can be considered only for certain control problems, in which fast updates are not required. But even when restricted to such problems, the computational complexity of the algorithms can be very high. It is therefore important to take advantage of any structure that might be available in the problem. SMC seems to manage this better than MCMC in sequential problems. The computation can also be parallelised and requires less tuning than that required by standard MCMC algorithms.

In this paper we investigate the use of a Sequential Monte Carlo (SMC) approach, in contrast to the Markov chain Monte Carlo (MCMC) approach we proposed previously. This approach of using SMC methods for the sampling of global optimisers within MPC, is to the best of our knowledge novel. We propose some

specific algorithmic choices in order to accelerate convergence of Simulated Annealing methods when applied to stochastic MPC problems. We shall demonstrate the effectiveness of our approach by means of numerical examples inspired by Air Traffic Management.

2 Problem Formulation

In general control problems one focuses on dynamical models, in which a specified user or controller or decision maker influences the evolution of the state, $X_k \in \mathscr{X}$, and the corresponding observation, $Y_k \in \mathscr{Y}$, by means of an action or control input, $A_k \in \mathscr{A}$, at each time k. Consider the following nonlinear non-Gaussian state space model

$$X_{k+1} = \psi(X_k, A_{k+1}, V_{k+1}), \; Y_k = \phi(X_k, A_k, W_k),$$

where $\{V_k\}_{k\geq 1}$ and $\{W_k\}_{k\geq 0}$ are mutually independent sequences of independent random variables and ψ, ϕ are nonlinear measurable functions that determine the evolution of the state and observation processes. The decision maker tries to choose the sequence $\{A_k\}_{k\geq 0}$, so that it optimises some user specified sequence of criteria $\{J_k\}_{k\geq 0}$.

In this paper we shall restrict our attention to the fully observed case ($Y_k \equiv X_k$), although our results can be generalised for the partially observed case as well. Furthermore, as our goal is to develop an algorithm for use with MPC, we will focus only on finite horizon problems. We refer the interested reader to [2] for a treatment on how SMC has been used for the infinite horizon case using stochastic gradients instead.

Conditional upon $\{A_k\}_{k\geq 0}$, the process $\{X_k\}_{k\geq 0}$ is a Markov process with $X_0 \sim \mu$ and Markov transition density $f(x'|x,a)$, so that we can write

$$X_{k+1} | (X_k = x, A_{k+1} = a) \sim f(\cdot|x,a). \tag{1}$$

These models are also referred to as Markov Decision Processes (MDP) or controlled Markov Chains.

We will now formulate an open loop problem solved at each MPC iteration. Let us introduce a measurable reward function $h : \mathscr{X} \times \mathscr{A} \to \mathbb{R}_+$, for the following additive reward decision problem. At time $k-1$, the action A_{k-1} has been selected, the state X_{k-1} is measured and then at time k one wants to maximise the function J_k defined as

$$J_k(A_{k:k+H-1}) = \mathbb{E}\left[\sum_{n=k}^{k+H-1} h(X_n, A_n)\right], \tag{2}$$

where $A_{k:k+H-1}$ denotes the joint vector $A_{k:k+H-1} = [A_k, \ldots, A_{k+H-1}]$ and the expectations are with respect to the joint distribution of the states $p_{A_{k:k+H-1}}(x_{k:k+H-1})$, giving

$$J_k(A_{k:k+H-1}) = \int_{\mathscr{X}^H} \underbrace{\left(\sum_{n=k}^{k+H-1} h(x_n, A_n)\right)}_{u(A_{k:k+H-1}, x_{k:k+H-1})} \underbrace{\left(\prod_{n=k+1}^{k+H-1} f(x_n|x_{n-1}, A_n)\right) f(x_k|X_{k-1}, A_k)}_{p_{A_{k:k+H-1}}(x_{k:k+H-1})} dx_{k:k+H-1}, \quad (3)$$

where we define

$$u(A_{k:k+H-1}, x_{k:k+H-1}) = \sum_{n=k}^{k+H-1} h(x_n, A_n). \quad (4)$$

We aim to perform the following maximisation

$$A^*_{k:k+H-1} = \arg \max_{A_{k:k+H-1} \in \mathscr{A}^H} J_k(A_{k:k+H-1}),$$

in order to obtain a solution for the open loop problem.

Of course, this is not a trivial task. If the control input took its values in a finite set $\mathscr{A}$ of cardinality K, it would be possible to approximate numerically this cost using particle methods or MCMC for the K^H possible values of $A_{k:k+H-1}$ and then select the optimal value. In [2] the authors present in detail how to get a particle approximation of J_k using standard SMC results for filtering. Of course, in practice such an approach cannot handle large values of H or K. Moreover if A_k takes values in a continuous space $\mathscr{A}$ and $J_k(A_{k:k+H-1})$ is differentiable with respect to $A_{k:k+H-1}$, one can still resort to a gradient search in $\mathscr{A}^H$. This has been presented in [2]. Using gradients would imply, as in any local optimisation method, that multiple runs from different initial points are needed to get a better estimate of the global optimum, but still it is difficult to get any formal guarantees. This motivates the use of Monte Carlo optimisation.

3 Monte Carlo Optimisation

Maximising (3) falls into the broader class of problems of maximising

$$J(\theta) = \int_{\mathscr{Z}} u(\theta, z) p_\theta(z) dz, \quad (5)$$

where we define $\theta = A_{k:k+H-1}$ and $z = x_{k:k+H-1}$, while θ^* are the maximisers of J. In this section we show how Monte Carlo simulation can be used to maximise J. In [1, 13] MCMC algorithms have been proposed for this and in [10] the authors explained how they can be combined with MPC. More recently, in [9] SMC methods have been applied for solving a marginal Maximum Likelihood problem, whose expression is similar to (5). In the remainder of this paper we shall focus on deriving a similar algorithm to [9], intended to be used for MPC. The main difference of our approach is that our problem formulation exhibits a slightly different structure. In fact, we are using a dynamical model intended for control problems, and

therefore are doing inference to compute time varying optimal actions instead of static parameters, which is the purpose of parameter estimation. Although the difference seems subtle at first glance, it is important and leads to similar algorithms showing completely different behaviour.

The basic idea is the same as in [1, 9, 13]. First we assume $u(\theta, z)$ is nonnegative. Note that this might seem restrictive at the beginning but we remark that any maximisation remains unaffected with respect to shifting by some finite positive constant. As in the standard Bayesian interpretation of Simulated Annealing, we define a distribution $\widetilde{\pi}_\gamma$

$$\widetilde{\pi}_\gamma(\theta) \propto p(\theta) J(\theta)^\gamma,$$

where $p(\theta)$ is an arbitrary prior distribution, which contains the maximisers θ^* and encapsulates any prior information on θ not captured by the model. As such information is not likely to be available, uninformative priors might be used. Under weak assumptions, as $\gamma \to \infty$, $\widetilde{\pi}_\gamma(\theta)$ becomes concentrated on the set of maximisers of J [8].

We now introduce γ artificial replicates of z, all stacked into a joint variable $z_{1:\gamma}$ and define the distribution π_γ

$$\pi_\gamma(\theta, z_{1:\gamma}) \propto \prod_{i=1}^{\gamma} u(\theta, z_i) p_\theta(z_i).$$

It easy to show that the marginal of π_γ is indeed proportional $\widetilde{\pi}_\gamma$, i.e.

$$\widetilde{\pi}_\gamma(\theta) \propto \int \pi_\gamma(\theta, z_{1:\gamma}) dz_{1:\gamma}.$$

We now define a strictly increasing integer infinite sequence $\{\gamma_l\}_{l \geq 0}$, which will play the role of the inverse temperature (as in SA). For a logarithmic schedule one can obtain formal convergence results [13]. For a large l, the distribution $\pi_{\gamma_l}(\theta, z_{1:\gamma_l})$ converges to the uniform distribution of the maximisers of J, [8]. In practice logarithmic schedules lead to slow convergence; more quickly increasing rates and finite sequences $\{\gamma_l\}_{l \geq 0}$ are therefore used. In general it is impossible to sample directly from π_γ, hence various Monte Carlo schemes have been proposed. In [1, 13] this is achieved by MCMC, and in [9] an SMC sampling approach was proposed for a Maximum Likelihood problem, based on the generic SMC algorithm found in [6]. The SMC approach can achieve more efficient sampling from π_γ, and avoids some of the fundamental bottlenecks of MCMC-based optimisation.

4 Stochastic Control Using MPC Based on SMC

SMC is a popular technique, applied widely in sequential inference problems. The underlying idea is to approximate a sequence of distributions $\pi_l(x)$[1] of interest as a

[1] x is not meant to be confused with x_k. Later it will be apparent that we shall be using $(\theta_k, z_{k,1:\gamma_l})$ as the variable of interest.

collection of N discrete masses of the variables (also referred as particles $\{X_l^{(i)}\}_{i=1}^N$), properly weighted by a collection of weights $\{w_l^{(i)}\}_{i=1}^N$ to reflect the shape of the distribution π_l. As π_l can be time varying, the weights and the particles are propagated iteratively by using a sequential importance sampling and resampling mechanism, which uses the particles of iteration $l-1$ to obtain new particles at iteration l. We shall be referring to $\{X_l^{(i)}, w_l^{(i)}\}_{i=1}^N$ as the particle approximation $\widehat{\pi}_l$ of π_l and this should satisfy

$$\sum_{i=1}^{N} w_l^{(i)} \delta_{X_l^{(i)}}(dx) \overset{N\to\infty}{\underset{a.s.}{\to}} \pi_l(dx),$$

where δ is a Dirac delta mass. For more details, see [5, 6, 7]. In Figure 1, we set out an SMC algorithm which can be used for the MPC problem defined in Section 2.

At time k,

- I) For $l = 1,...,l_{\max}$:

1. **Sampling new particles:**
   - For each particle $i = 1,...,N$ sample: $A_{k:k+H-1,l}^{(i)} \sim q_l(\cdot|X_{k:k+H-1,1:\gamma_{l-1}}^{(i)}, A_{k:k+H-1,l-1}^{(i)})$
   - For each particle $i = 1,...,N$ sample replicas of the joint state trajectory, for $j = \gamma_{l-1}+1,...,\gamma_l$, $X_{k:k+H-1,j}^{(i)} \sim \prod_{n=k}^{k+H} f(x_n|x_{n-1}, A_{n,l}^{(i)})$.
2. **Weighting particles:** for each particle $i = 1,...,N$ assign weights $w_l^{(i)} = w_{l-1}^{(i)} \prod_{j=\gamma_{l-1}+1}^{\gamma_l} u(A_{k:k+H-1,l}^{(i)}, X_{k:k+H-1,j}^{(i)})$, normalise $w_l^{(i)} = \frac{w_l^{(i)}}{\sum_{j=1}^{N} w_l^{(j)}}$.
3. **Resample**, if necessary, to get new particle set $\{(A_{k:k+H-1,l}^{(i)}, X_{k:k+H-1,1:\gamma_l}^{(i)})\}_{i=1}^N$ with equal weights $w_l^{(i)} = \frac{1}{N}$.

- II) Compute the maximiser estimate $\widehat{A}_{k:k+H-1}$
- III) Apply $\widehat{A}_k$ as the action of time k.

Obtain measurement $Y_k = X_k$ and proceed to time $k+1$

Fig. 1 SMC Algorithm for MPC

Steps I) 1-3 of the algorithm are iterated recursively to obtain a particle approximation for the maximisers of J_k. Referring to the general description of SMC in the previous paragraph, one can associate π_{γ_l} with π_l. We shall be using iteration number l, to index the propagation of π_{γ_l}. As we cannot run an infinite number of iterations, we shall terminate the iteration at $l = l_{\max}$. Note that l should not be confused with the time index k of Section 2 regarding the real time evolution of the state. To avoid this, we define $\theta_k = A_{k:k+H-1}$ and $z_k = x_{k:k+H-1}$, and also

add a subscript k to π_γ to show the real time index. At each epoch k , we are interested in obtaining $l_{\max}$ consecutive particle approximations of $\pi_{k,\gamma_l}(\theta_k, z_{k,1:\gamma_l})$, where $z_{k,1:\gamma_l} = [z_{k,\gamma_1}, ..., z_{k,\gamma_l}]$[2]. At each iteration l, we obtain particle approximations $\widehat{\pi}_{\gamma_l}$, $\{(\Theta_{k,l}^{(i)}, Z_{k,1:\gamma_l}^{(i)}), w_l^{(i)}\}_{i=1}^N$, by propagating the particles of the previous approximation $\widehat{\pi}_{k,\gamma_{l-1}}$, $\{(\Theta_{k,l-1}^{(i)}, Z_{k,1:\gamma_{l-1}}^{(i)}), w_{l-1}^{(i)}\}_{i=1}^N$, weighting the new particles and then resampling.

We now explain briefly how steps 1 to 3 can be derived. Suppose we are at epoch k and iteration l. For the sampling step, we assume in this paper that we can sample from the model of the state, $p_{\theta_k}(z_k)$, by repeatedly sampling from each transition density f. This is not always possible, but for most practical control problems it is. If one cannot sample directly from f then importance sampling can be used. For every particle i, to get a sample $Z_{k,j}^{(i)} = X_{k:k+H-1,j}^{(i)}$, we use the previous measured state X_{k-1} and then repeatedly sample $X_{n,j}^{(i)} \sim f(\cdot|X_{n-1,j}^{(i)}, A_{n,l}^{(i)})$ for $n = k, ..., k+H-1$. For sampling new particles $\Theta_{k,l}^{(i)}$, an importance sampling approach has to be used at each l. We shall be using an importance distribution q_l to obtain $\Theta_{k,l}^{(i)} \sim q_l(\cdot|Z_{k,1:\gamma_{l-1}}^{(i)}, \Theta_{k,l-1}^{(i)})$ by simulation. We have intentionally chosen q_l to be varying with l and to depend on $Z_{k,1:\gamma_l}$ as this is more convenient for the general design setting. We shall not provide details on how to design q_l, as this depends on the problem specifics [7]. We shall refer the reader again to [5, 6] for a more general treatment.

For the weighting step we use

$$\frac{\pi_{k,\gamma_l}}{\pi_{k,\gamma_{l-1}}} \propto \prod_{i=\gamma_{l-1}+1}^{\gamma_l} u(\theta_k, z_{i,k}) p_{\theta_k}(z_{k,i})$$

to obtain $\frac{w_l^{(i)}}{w_{l-1}^{(i)}}$ as an importance ratio proportional to $\prod_{j=\gamma_{l-1}+1}^{\gamma_l} u(\Theta_{k,l}^{(i)}, Z_{k,j}^{(i)})$. To obtain $u(\Theta_{k,l}^{(i)}, Z_{k,j}^{(i)})$ – see (3) and (4) – one can evaluate $h\left(X_{n,j}^{(i)}, A_{n,l}^{(i)}\right)$ point-wise at each step n during the sampling stage, and then get the total value of $u(\Theta_{k,l}^{(i)}, Z_{k,j}^{(i)})$. After normalising the weights one can perform a resampling step according to the multinomial distribution $\{(\Theta_{k,l}^{(i)}, Z_{k,1:\gamma_l}^{(i)}), w_l^{(i)}\}_{i=1}^N$, if the variance of the weights is low; see [7] for details.

As as regards step II, after having obtained the particle approximation $\widehat{\pi}_{k,\gamma_{\max}}$,one could use in principle any sample $\Theta_{k,l_{\max}}^{(i)}$ from the final particle set as the estimator of θ_k^*, where θ_k^* is the maximiser of $J_k(\theta_k)$. Given that $\widehat{\pi}_{k,\gamma_{\max}}$ should converge close to a uniform distribution over the set of θ_k^*, then any sample $\Theta_{k,l_{\max}}^{(i)}$ should be sufficiently close to θ_k^*. To improve things one could either choose a sample randomly according to its weight as in resampling, or in a deterministic fashion, by

[2] Note that $z_{k,1:\gamma_l}$ is the stacked vector of the γ_l artificial replicates of $x_{k:k+H-1}$ and n is used as a sequence index for the interval $k : k+H-1$.

choosing the sample with the maximum weight. In many cases, such as if there is some symmetry in the location of the maximisers, this should be much better than using the empirical mean $\sum_{i=1}^{N} w_{l_{\max}}^{(i)} \Theta_{k,l_{\max}}^{(i)}$ to compute $\widehat{\theta}_k$.

We can use this open loop solution for performing an MPC step at step III. Once $\widehat{A}_{k:k+H-1}$ is computed, we then apply $\widehat{A}_k$. Then we proceed to time $k+1$ and repeat steps I-III for optimising J_{k+1}.

5 Numerical Examples

In this section we demonstrate how the proposed algorithm can be used in navigation examples, where it is required to coordinate objects flying at constant altitude, such as aircraft, UAVs, etc. We consider a two-dimensional constant speed model for the position of an object controlled by changing its bearing

$$X_{k+1} = X_k + v\tau[\sin\phi_{k+1}, \cos\phi_{k+1}]^T + b_{k+1} + V_{k+1}, \tag{6}$$

where v is the speed of the object, τ is a measuring period, ϕ is the bearing , b_n represents the predicted effect of the wind and $V_k \overset{iid}{\sim} \mathcal{N}(0,\Sigma)$. Although this is a linear kinematic model with Gaussian added noise, the algorithm in Figure 1 can handle nonlinear and non-Gaussian cases as it requires no assumptions on the dynamics or distributions.[3] We shall be using some way points α_n that the object is desired to pass through at each time n. We shall encode this in the following reward at time k,

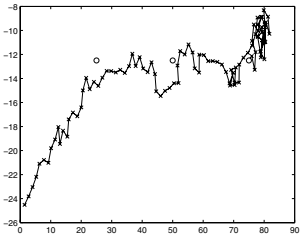
$$J_k(\phi_{k:k+H-1}) = \mathbb{E}[\sum_{n=k}^{k+H} (c - \|X_n - \alpha_n\|_Q^2 - \|\phi_n - \phi_{n-1}\|_R^2)],$$

where $c > 0$ is sufficiently large to ensure $c - \|X_n - \alpha_n\|_Q^2 - \|\phi_n - \phi_{n-1}\|_R^2 \geq 0$, and $Q, R \geq 0$ are matrices of appropriate sizes.

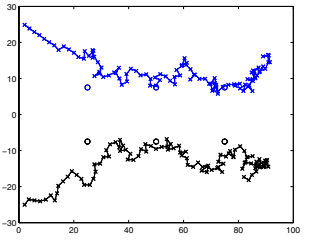
We shall be investigating a number of scenarios. Firstly assume there are three way-points to be cleared, such that $\alpha_1 = \alpha_2 = ... = \alpha_{H_1}$, $\alpha_{H_1+1} = ... = \alpha_{H_2}$ and $\alpha_{H_2+1} = ... = \alpha_H$. If a single object obeying (6) starts at some initial position, then choosing a maneuver to maximise J_k means that it should pass through the points and stay near the check points as long as possible. The result of applying the algorithm of Figure 1 is shown in Figure 2(a).

We proceed by adding additional objects that also obey the dynamics of (6). Suppose that safety requirements impose the constraint that objects should not come closer to each other than a distance d_{min}. This makes the problem much harder as one has to ensure that constraints are not violated and the constraints have a significant effect on the multi modality of the reward. Let X_k^j denote the position of the jth object. The feasible space $\mathcal{X}^H$ is modified so that $\mathcal{X}^H = \{X_n^j \in \mathbb{R}^2 : \left\|X_n^j - X_n^i\right\| \geq d_{min}, \forall i \neq j, n = k, ..., k+H-1\}$. Moreover, all expectations presented

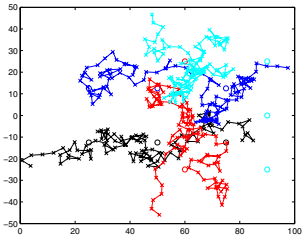
[3] Also non-convex constraints can be added.



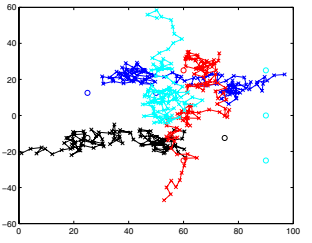
(a) Single-object example of MPC



(b) Two objects flying parallel using MPC



(c) Four-object example open loop solution at $k = 1$



(d) Four-object example using MPC

Fig. 2 Application of SMC optimisation for different scenarios. A linear inverse temperature schedule was used, $\gamma_l = 5+3l$ and $l_{max} = 100$. For the rest of the parameters we used $N = 300$, $T = 100$ $v\tau = 2$, $b_k = [0.2, 0.2]^T$, $\Sigma = I$, $d_{min} = .5$, $Q = .1$, and $R = .01$. The waypoints of each object are plotted with circles with the same colour as the track of the object. When implemented in Intel Core Duo T2250 at 1.73GHz processor without any parallelisation, the computational time for each k is .05 sec per H per N per plane

so far, for example equation (3), should be defined over the new feasible spaces for each object. To account for this we could modify the instantaneous reward to $h\left(x_n^j, A_n^j\right)\mathbf{1}_{X_{0:n}^j \in \mathscr{X}^H}$, where $\mathbf{1}_{x\in B}$ is an indicator function for the set B. Such a simple penalty approach means that no reward should be credited if a sampled trajectory $Z_{k,j}^{j,(i)}$ does not obey the constraint and its corresponding weight should be set to zero. This is a simulation based approach to deal with constraints, i.e. we propagate state samples only from the feasible region of the state space. We also assume that the initial condition of the system is not in violation of the constraint. Then, given that the SMC optimisation algorithm uses a large number of particles and samples many replicates of the state trajectories, this should allow safe decisions to be made. For a finite number of samples one could also obtain expressions for the probability of violating a constraints, e.g. using the Chebychev inequality. In practice, when using large number of particles the violation probability was observed to be very low (10^{-7}).

We have verified this using two different scenarios. In the first one seen in Figure 2(b), two objects flying in parallel towards the same direction, try to approach parallel way-points. MPC was used for repeated number of runs and no conflict between two objects took place. Further scenarios are depicted in Figures 2(c) and 2(d). These show a more complicated problem, in which four objects are directed towards each other and their way-points would lead them to a collision if constraints were not taken into account. In Figure 2(c) we plot the open loop solution of the problem at time $k = 1$ for a random disturbance sequence and in Figure 2(d) the MPC solution. We see that three objects try to cruise as closely as possible between their way-points and orbit each way point for some time, while one orbits waiting for the others. Again no conflict took place in multiple runs.

As a concluding remark, we would like to stress that little tuning was done to produce the results shown here. Also, we have not used any structural properties of Linear Gaussian systems with quadratic costs and just implemented the algorithm in Figure 1. The examples show early results from ongoing work, but they already demonstrate that the proposed SMC algorithm can be effective for non-convex decision problems.

Acknowledgements. This work was supported by EPSRC, Grant EP/C014006/1, and by the European Commission under project iFly FP6- TREN-037180. The authors are also grateful to Zhe (Anna) Yang for providing recommendations to improve the quality of the paper.

References

1. Amzal, B., Bois, F.Y., Parent, E., Robert, C.P.: Bayesian-Optimal Design via Interacting Particle systems. Journal of the American Statistical Association, Theory and Methods 101(474) (2006)
2. Andrieu, C., Doucet, A., Singh, S.S., Tadić, V.: Particle Methods for Change Detection, Identification and Control. Proceedings of the IEEE 92, 423–438 (2004)
3. Bertsekas, D.P.: Dynamic programming and optimal control. Athena Scientific, Belmont (1995)
4. Bertsekas, D.P.: Dynamic Programming and Suboptimal Control: A Survey from ADP to MPC. European J. of Control 11(4-5) (2005)
5. Del Moral, P.: Feynman-Kac formulae: genealogical and interacting particlesystems with applications. Springer, New York (2004)
6. Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo Samplers. J. Royal Statist. Soc. B 68(3), 411–436 (2006)
7. Doucet, A., de Freitas, J.F.G., Gordon, N.J.: Sequential Monte Carlo Methods in Practice. Springer, New York (2001)
8. Hwang, C.R.: Laplace's method revisited: weak convergence of probability measures. Ann. Probab. 8(6), 1177–1182 (1980)
9. Johansen, A.M., Doucet, A., Davy, M.: Particle methods for Maximum Likelihood Parameter Estimation in Latent Variable Models. Statistics and Computing 18(1), 47–57 (2008)
10. Lecchini Visintini, A., Glover, W., Lygeros, J., Maciejowski, J.M.: Monte carlo optimization for conflict resolution in air traffic control. IEEE Trans. Intell. Transp. Syst. 7(4), 470–482 (2006)

11. Lecchini Visintini, A., Lygeros, J., Maciejowski, J.M.: Simulated Annealing: Rigorous finite-time guarantees for optimization on continuous domains. In: Platt, J.C., Koller, D., Singer, Y., Roweis, S. (eds.) Advances in NIPS 20, pp. 865–872. MIT Press, Cambridge (2008)
12. Maciejowski, J.M.: Predictive control with Constraints. Prentice-Hall, Englewood Cliffs (2002)
13. Muller, P., Sanso, B.G., De Iorio, M.: Optimal Bayesian design by Inhomogeneous Markov Chain Simulation. Journal of the American Statistical Association, 788–798 (2004)
14. Rawlings, J.B., Bakshi, B.R.: Particle filtering and moving horizon estimation. Comput. Chem. Eng. 30, 1529–1541 (2006)
15. Robert, C.P., Casella, G.: Monte Carlo Statistical Methods, 2nd edn. Springer, Heidelberg (2004)

An NMPC Approach to Avoid Weakly Observable Trajectories

Christoph Böhm, Felix Heß, Rolf Findeisen, and Frank Allgöwer

Abstract. Nonlinear systems can be poorly or non-observable along specific state and output trajectories or in certain regions of the state space. Operating the system along such trajectories or in such regions can lead to poor state estimates being provided by an observer. Such trajectories should be avoided if used for state feedback control or monitoring purposes. In this paper, we outline a possible approach to avoid weakly observable trajectories in the frame of nonlinear model predictive control (NMPC). To illustrate the practical relevance and applicability, the proposed controller is used for an emergency collision avoidance maneuver for passenger cars.

Keywords: Nonlinear model predictive control; Observability; Autonomous steering; Vehicle dynamics.

1 Introduction

Nonlinear systems can be poorly or non-observable along specific state and output trajectories or in certain regions of the state space. Operating the system along such trajectories or in such regions can lead to poor state estimates being provided by an observer.

Observability for linear time-invariant systems is well understood and there exist several equivalent ways to define observability [9, 15]. Deriving measures for

Christoph Böhm, Felix Heß, and Frank Allgöwer
Institute for Systems Theory and Automatic Control,
University of Stuttgart, Germany
e-mail: {cboehm,allgower}@ist.uni-stuttgart.de

Rolf Findeisen
Institute for Automation Engineering,
Otto-von-Guericke-Universität Magdeburg, Germany
e-mail: rolf.findeisen@ovgu.de

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 275–284.
springerlink.com

observability has been the focus of many research activities in the past. The determinant of the observability matrix, the observability Gramian or e.g. the measures for observability derived in [10] may be used to determine the degree of observability of a system. In contrast to the linear case for nonlinear systems observability may depend on the states and inputs. Therefore, controllers might steer the system to regions where it is non-observable or weakly observable. The loss of observability can lead to deteriorating observer performance [13], and thus, in the case of output feedback control, to loss of performance or stability. Hence, it is desirable to avoid regions of weak observability in order to provide a satisfying estimation of the system states.

In this paper we recapitulate results presented in [2]. There, by imposing a constraint on the nonlinear local observability matrix, which is used as measure for observability, an NMPC based approach to avoid weakly observable trajectories is provided. The contribution of this paper is the illustration of the practical relevance of the results proposed in [2] by applying them to an emergency collision avoidance maneuver for passenger cars. Extending the NMPC setup for the collision avoidance maneuver by an observability constraint, the degree of observability as well as the performance of the lateral velocity observer is improved along the calculated trajectories.

The remainder of the paper is organized as follows. In Section 2 a motivation and the problem setup is given. Section 3 recapitulates the results presented in [2]. In Section 4 the collision avoidance maneuver is introduced and the obtained simulation results are discussed. The paper concludes with a short summary in Section 5.

2 Problem Setup and Motivation

We consider nonlinear systems of the form

$$\dot{x} = f(x,u),\ x(0) = x_0, \tag{1a}$$

$$y = h(x,u), \tag{1b}$$

with $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $y \in \mathbb{R}^p$. Additionally, the system might be subject to state and input constraints of the form $u(t) \in \mathscr{U}\ \forall\, t \geq 0$ and $x(t) \in \mathscr{X}\ \forall\, t \geq 0$. Here $\mathscr{X} \subseteq \mathbb{R}^n$ is the state constraint set and $\mathscr{U} \subset \mathbb{R}^m$ is the set of feasible inputs. The control task is to stabilize system (1) about the origin, while avoiding regions in which the system is weakly or non-observable. Since constraints are present, optimization based control methods, such as NMPC, are well suited. In NMPC the future behavior of the system is predicted. Therefore, we introduce predicted states and inputs, $\bar{x}$ and $\bar{u}$. The cost function J, that is minimized over the prediction horizon T_p, is

$$J\big(\bar{x}(\cdot),\bar{u}(\cdot)\big) = \int_{t_k}^{t_k+T_p} \bar{x}^T Q \bar{x} + \bar{u}^T R \bar{u}\, \mathrm{d}\tau + \bar{x}^T(t_k+T_p) P \bar{x}(t_k+T_p), \tag{2}$$

with $0 < Q = Q^T \in \mathbb{R}^{n\times n}$, $0 < R = R^T \in \mathbb{R}^{m\times m}$ and $0 < P = P^T \in \mathbb{R}^{n\times n}$. Hence, the open-loop optimal control problem that is solved repeatedly at the sampling instants t_k, where $t_k = \delta k$ with the fixed sampling time δ and $k \in \mathbb{N}$, is given by

$$\min_{\bar{u}(\cdot)} J\big(\bar{x}(\cdot),\bar{u}(\cdot)\big), \tag{3a}$$

subject to

$$\dot{\bar{x}}(\tau) = f\big(\bar{x}(\tau),\bar{u}(\tau)\big),\ \bar{x}(t_k) = x(t_k), \tag{3b}$$

$$\bar{x}(\tau) \in \mathscr{X},\ \bar{u}(\tau) \in \mathscr{U},\ \forall \tau \in \big[t_k, t_k + T_p\big], \tag{3c}$$

$$\bar{x}(t_k + T_p) \in \mathscr{E}. \tag{3d}$$

Note that the predicted states $\bar{x}$ are forced to lie in the so called terminal region $\mathscr{E}$ at the end of the prediction horizon to enforce stability.

The solution to the optimization problem leads to

$$\bar{u}^\star\big(t; x(t_k)\big) = arg\min_{\bar{u}(\cdot)} J\big(\bar{x}(\cdot),\bar{u}(\cdot)\big), \tag{4}$$

assuming that the minima is obtained. The control input applied to system (1a) is updated at each sampling instant t_k by the repeated solution of the open-loop optimal control problem (3), i.e. the applied control input is given by

$$u(t) = \bar{u}^\star(t; x(t_k)),\ t \in \big[t_k, t_k + \delta\big). \tag{5}$$

Since the solution to the optimization problem at each time instant t_k depends on the current system state $x(t_k)$, state feedback is provided. If certain well-known conditions on P and $\mathscr{E}$ are satisfied, the presented state feedback approach guarantees stability of the closed-loop system, see e.g. [6]. However, in most practical applications the system states have to be estimated by an observer which exploits the available measurement information $y = h(x,u)$ (1b). The approach presented does not guarantee that the system is observable along the obtained optimal trajectories. Loss of observability may lead to poor observer performance along such trajectories. Therefore, the prediction of the system behavior might be based on inaccurate estimates of the states. This may lead to poor closed-loop performance or even instability.

In the following section we outline an approach to guarantee observability along NMPC trajectories, see also [2]. The determinant of the local observability matrix is evaluated along the calculated trajectories and used as a measure for observability. Introducing an additional constraint on the observability matrix we can assure a certain degree of observability along the considered trajectories.

3 Avoiding Weakly Observable Trajectories

As discussed in the previous section, an NMPC controlled system may be steered to non-observable or poorly observable regions by the controller. This should in general be avoided. In [2] it is shown that by using the nonlinear local observability matrix in the NMPC setup, a certain degree of observability along the calculated trajectories can be guaranteed. The nonlinear local observability matrix is derived from the n-observability map (see e.g. [11, 13]), which is for system (1) defined as $q_n(x,u) = [y_1, \dot{y}_1, \ldots, y_1^{(n-1)}, \ldots, y_p, \dot{y}_p, \ldots, y_p^{(n-1)}]$, where $y = [y_1, \ldots, y_p]$. Basically,

the considered system is locally observable if the nonlinear local observability matrix $\mathscr{O}$, defined as the Jacobian of the n-observability map q_n,

$$\mathscr{O}(x,u) = \frac{\partial q_n(x,u)}{\partial x}, \tag{6}$$

has full rank n for all $x \in \mathscr{X}$ and all $u \in \mathscr{U}$. It is locally observable at some point x_s if $\mathscr{O}(x_s,u)$ has full rank at this point for all $u \in \mathscr{U}$. In the following the expression *observability* is used in the sense of *local observability*.

For simplicity we use the determinant of the observability matrix $\det(\mathscr{O})$ as measure for observability, which is possible if $\mathscr{O}$ is a square matrix. Clearly, system (1) is non-observable if $\det(\mathscr{O}) = 0$ holds. Furthermore, small values of $\det(\mathscr{O})$ imply weak observability since the observability matrix is close to singular. In this paper we consider observability along predicted trajectories $\bar{x}$ and $\bar{u}$. Thus, system (1) is observable along the predicted trajectory if $\mathscr{O}(\bar{x},\bar{u})$ has full rank along the complete trajectory.

In [2] two approaches are presented which guarantee observability along NMPC trajectories. The first one is based on penalizing weakly observable trajectories in a modified cost functional (2). In this paper we focus on the second approach presented in [2], which extends the optimization problem (3) by a further constraint on the observability matrix $\left|\det\left(\mathscr{O}(\bar{x},\bar{u})\right)\right| \geq \Omega_{min}$. Therefore, the modified optimal control problem is formulated as

$$\min_{\bar{u}(\cdot)} J\left(\bar{x}(\cdot),\bar{u}(\cdot)\right), \tag{7a}$$

subject to

$$\dot{\bar{x}}(\tau) = f\left(\bar{x}(\tau),\bar{u}(\tau)\right),\ \bar{x}(t_k) = x(t_k), \tag{7b}$$

$$\bar{x}(\tau) \in \mathscr{X},\ \bar{u}(\tau) \in \mathscr{U},\ \forall \tau \in \left[t_k, t_k + T_p\right], \tag{7c}$$

$$\bar{x}(t_k + T_p) \in \mathscr{E}, \tag{7d}$$

$$\left|\det\left(\mathscr{O}\left(\bar{x}(\tau),\bar{u}(\tau)\right)\right)\right| \geq \Omega_{min},\ \forall \tau \in \left[t_k, t_k + T_p\right]. \tag{7e}$$

The constraint (7e) assures that the determinant of the observability matrix is always larger than a minimum value Ω_{min}. Thus, with a suitable choice of the design parameter Ω_{min}, the obtained optimal input $\bar{u}^\star$ avoids steering the system to weakly observable regions.

4 Collision Avoidance Maneuver with Guaranteed Observability

In this section we illustrate the practical relevance of the approach presented above. We consider an autonomous collision avoidance maneuver for passenger cars. The vehicle is driving on a flat road towards an obstacle which is detected by a superior safety system. Since the driver, e.g. due to inattention, does not react to avoid a collision, the safety system takes over the control of the car and drives the

vehicle around the obstacle as fast as possible using an active steering system based on electric motors. The underlying NMPC controller calculates the required trajectories for this emergency maneuver. As will be shown, along the obtained NMPC trajectories observability of the lateral velocity is lost to a large extend. This causes a non-satisfying estimation of the lateral velocity, which might be crucial since safety systems such as e.g. ESP [14] require a precise estimation.

An approach providing a rather smooth lane change maneuver with predictive control has been proposed in the literature [5]. In this paper we consider an emergency collision avoidance maneuver with large lateral accelerations and velocities. Furthermore, we are interested in observability properties along the obtained trajectories.

In the following, we introduce the model of the considered vehicle and discuss its observability properties. Furthermore, we provide the NMPC setup for the collision avoidance maneuver and discuss the obtained simulation results. For a more detailed discussion of the considered problem we refer to [7].

4.1 Vehicle Dynamics

We consider the so called nonlinear one-track model describing the vehicle motion given by

$$\frac{d}{dt}\begin{bmatrix} v_y \\ \psi \\ \dot{\psi} \\ Y \\ X \end{bmatrix} = \begin{bmatrix} \frac{1}{m}(S^R + S^F\cos\delta) - \dot{\psi}v_x \\ \dot{\psi} \\ \frac{1}{J_z}(-l_R S^R + l_F S^F\cos\delta) \\ v_x\sin\psi + v_y\cos\psi \\ v_x\cos\psi - v_y\sin\psi \end{bmatrix}, \tag{8}$$

with the lateral velocity v_y, the yaw rate and yaw angle $\dot{\psi}$ and ψ, and the position of the vehicle X and Y. The parameters of the model are the vehicle mass m, the longitudinal velocity v_x (assumed to be constant), the distances of the front and rear axes to the center of gravity, l_F and l_R, and the moment of inertia J_z. The steering angle δ is available to control the motion of the car. The vehicle dynamics depend on the lateral forces acting on the front and rear tires, S^F and S^R. Both are nonlinear functions of the lateral slip angles α_F and α_R, which mainly depend on v_y, $\dot{\psi}$ and δ, see e.g. [12].

Measurable outputs are the lateral acceleration $a_y = \frac{S^F\cos\delta + S^R}{m}$ and the yaw rate $\dot{\psi}$. Furthermore, the states X and Y are available for measurement. The lateral velocity has to be estimated by an observer. In this paper we use the observer $\dot{\hat{v}}_y = -\dot{\psi} + a_y - K_{v_y}(a_y - \hat{a}_y)$ [8], with the observer gain K_{v_y} and with $\hat{a}_y = \frac{\hat{S}^F\cos\delta + \hat{S}^R}{m}$. In the following we analyze the observer for large lateral velocities. Consider the case where both v_y and $\hat{v}_y$ are large, but differ significantly, i.e. also the estimation error is large. Due to the nonlinear dependency of the forces S^F and S^R on the lateral slip angles α^F and α^R [12], which themselves are functions of v_y, this leads to $S^i \approx \hat{S}^i$, and therefore to $a_y \approx \hat{a}_y$. As a consequence, for large lateral velocities, which usually result in large slip angles α_F and α_R, the correction term

$K_{v_y}(a_y - \hat{a}_y)$ in the observer vanishes and the estimation error remains unchanged. This effect can also be explained by observability considerations. Using the measurements $y_1 = a_y$ and $y_2 = \dot{\psi}$ one obtains the observability map

$$q(v_y, \dot{\psi}) = \begin{bmatrix} y_1 \\ y_2 \\ \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{m}(S^R + S^F \cos\delta) \\ \dot{\psi} \\ \frac{d}{dt}\frac{1}{m}(S^R + S^F \cos\delta) \\ \frac{1}{J_z}(-l_R S^R + l_F S^F \cos\delta) \end{bmatrix}. \tag{9}$$

As defined in [4], the resulting nonlinear local observability matrix is

$$\mathcal{O}(v_y, \dot{\psi}) = \frac{\partial q(v_y, \dot{\psi})}{\partial (v_y, \dot{\psi})} = \begin{bmatrix} 0 & 1 \\ -\frac{C_F + C_R}{m v_x} & \frac{C_R l_R - C_F l_F}{m v_x} \\ g_1(v_y, \dot{\psi}) & g_2(v_y, \dot{\psi}) \\ g_4(v_y, \dot{\psi}) & g_3(v_y, \dot{\psi}) \end{bmatrix}, \tag{10}$$

where the functions g_i linearly depend on C_F and C_R defined as in [1] by

$$C_F = \frac{\partial S^F}{\partial \alpha^F}\bigg|_{\alpha_0} \frac{v_x^2 \cos\delta_0}{v_x^2 + (v_{y0} + \dot{\psi}_0 l_F)^2}, \; C_R = \frac{\partial S^R}{\partial \alpha^R}\bigg|_{\alpha_0} \frac{v_x^2 \cos\delta_0}{v_x^2 + (v_{y0} - \dot{\psi}_0 l_R)^2}. \tag{11}$$

Since $C_{F,R} = \frac{\partial S^{F,R}}{\partial \alpha^{F,R}} \to 0$ for large values of the slip angle $\alpha^{F,R}$, in the observability matrix (10) the functions g_i vanish due to their linear dependency on $C_{F,R}$. Thus, it is easy to see that $\mathcal{O}$ has a rank loss for large $\alpha^{F,R}$ and therefore, the system is not observable in these regions of the state space. Hence, it is reasonable to chose $\det(\tilde{\mathcal{O}})$ as measure for observability, where $\tilde{\mathcal{O}}$ consists of the first two rows of $\mathcal{O}$.

In the following section an NMPC based autonomous emergency collision avoidance maneuver is presented. Along the calculated trajectories, large slip angles occur which cause weak observability and poor observer performance for the lateral velocity. It is shown that this problem can be overcome applying the approach presented in Section 3.

4.2 Collision Avoidance Maneuver

In the following we present the setup of a state feedback NMPC controller which calculates the optimal input trajectories for the collision avoidance maneuver. The considered cost function is of the form (2), such that the input energy and the deviation from regular driving on the right lane is penalized. For this, we have chosen $Q = \mathrm{diag}(1, 1, 10, 10, 0)$ and $R = 50$ [7]. The corresponding optimization problem at each sampling instant, which is solved based on full state information, is subject to the one-track model (8) and subject to constraints that are derived in the following. The vehicle has to satisfy the constraints imposed by the road, Y_l and Y_r. Furthermore, it has to avoid a collision with the obstacle. Therefore, for all X-positions satisfying $X_{O1} \leq X \leq X_{O2}$ the lateral position of the car has to satisfy $Y \geq Y_O$, where X_{O1}, X_{O2} and Y_O are defined by the vehicle's geometry. Additionally, the steering

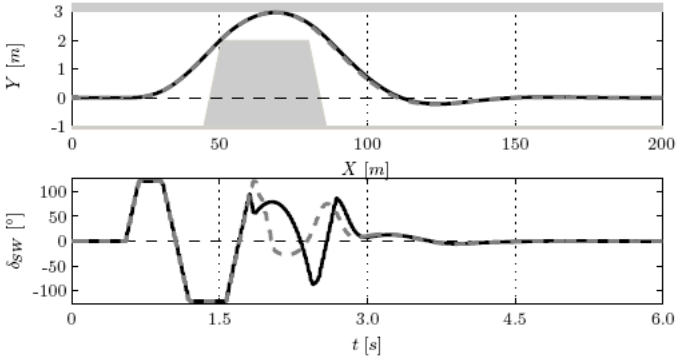


Fig. 1 $X-Y$-position and steering wheel angle. Comparison of classical (gray line) and modified (black line) NMPC approach

wheel angle δ_{SW}, which is linearly dependent on the front tires angle δ, and its derivative $\dot{\delta}_{SW}$ have to satisfy $\delta_{SW} \leq 120^\circ$ and $\dot{\delta}_{SW} \leq \frac{900^\circ}{s}$. If the optimization problem is subject to

$$c_1(X,Y) = -Y + Y_r \leq 0,\ c_2(X,Y) = Y - Y_l \leq 0, \tag{12a}$$

$$c_3(\delta_{SW}) = |\delta_{SW}| - 120^\circ \leq 0,\ c_4(\dot{\delta}_{SW}) = |\dot{\delta}_{SW}| - \frac{900^\circ}{s} \leq 0, \tag{12b}$$

$$c_5(X,Y) = \left\{ \begin{matrix} Y_O - Y & \text{if } \mathrm{d}X \geq 0 \\ Y_O - Y + dX & \text{if } \mathrm{d}X < 0 \end{matrix} \right\} \leq 0, \tag{12c}$$

in which $\mathrm{d}X = \frac{X_{O2}-X_{O1}}{2} - \left|X - \frac{X_{O2}+X_{O1}}{2}\right|$, then the obtained optimal trajectories satisfy input constraints and constraints imposed by the road and the obstacle. The terminal region $\mathscr{E}$ and the terminal penalty term to guarantee stability of the presented approach are calculated using the method proposed in [3] based on a linearized model which is valid in the terminal region.

The obtained input trajectories are applied to a two-track model of the vehicle which also includes the roll angle and the roll rate as states, in order to simulate a model-plant mismatch which has to be expected in practical realizations of the presented approach. Along the resulting trajectories we investigate the performance of the lateral velocity observer in order to illustrate the effect of steering the vehicle to weakly observable regions. However, we do not feed back the estimated lateral velocity to the controller, but provide state feedback in this paper. Future research is necessary to investigate the closed-loop performance when the controller relies on the estimated lateral velocity. The observer is designed for the one-track model and thus also faces a model-plant mismatch.

In the following we compare simulation results of a classical NMPC setup according to Section 2 to the modified approach derived in Section 3 where the additional observability constraint $1 - |\det(\tilde{\mathscr{O}})(v_y, \dot{\psi}))| \leq 0$ is imposed. The optimization problems inherent to both controllers are subject to the given state and input

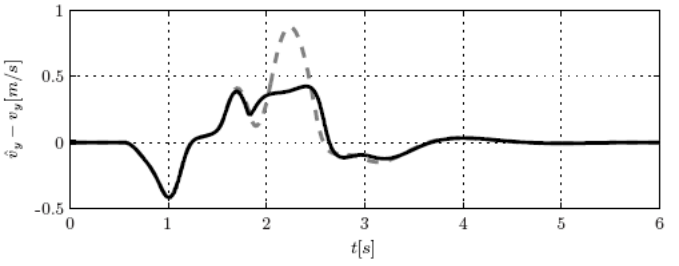


Fig. 2 Lateral velocity estimation error. Comparison of classical (gray line) and modified (black line) NMPC approach

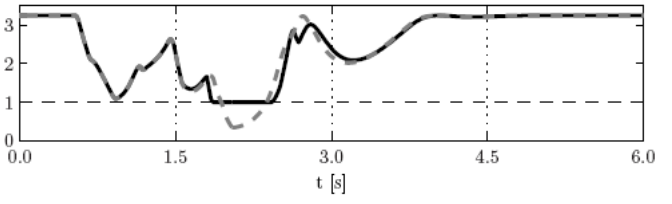


Fig. 3 Degree of observability. Comparison of classical (gray line) and modified (black line) NMPC approach

constraints (12). Figure 1 shows the position of the vehicle and the obstacle as well as the corresponding steering wheel angle during the collision avoidance maneuver. Although the steering angles in both scenarios differ significantly in the time interval $t = [1.8s, 3.0s]$, the position of the vehicle can hardly be distinguished. However, as illustrated in Figure 2, with the classical NMPC controller the estimation error of the lateral velocity observer in this interval becomes almost twice as large as when the modified controller is applied. The poor observer performance is a result of weak observability in the corresponding time interval. Figure 3 shows the determinant of the observability matrix during both scenarios. Adding the constraint on the observability matrix, in the modified approach the determinant is forced to be greater than $\Omega_{min} = 1$. The obtained higher degree of observability results in a better observer performance in the time interval where the observability constraint is active. As can be seen in Figure 2 and Figure 3, during the whole simulation the estimation performance deteriorates as soon as the determinant of the observability matrix decreases. The reason for this is the present model-plant mismatch, i.e. the simulation term in the observer does not represent the system dynamics exactly while at the same time the influence of the correction term decreases. Using a larger value for Ω_{min}, one could further increase the estimation performance, however at the cost of a slower collision avoidance maneuver.

Remark 1. *Alternatively to the determinant of the nonlinear local observability matrix, in principle any other observability measures could be used. Although we only use two rows of the observability matrix $\mathcal{O}$ to measure observability, any other*

observability measure would lead to similar effects, which can be explained by physical considerations.

Remark 2. *Due to the present model-plant mismatch, slightly "robust" constraints have to be introduced in order to guarantee satisfaction of the hard constraints (12). Although this issue was considered in the simulations, due to space limitations we do not provide a discussion here.*

Remark 3. *Simulations have shown that the introduction of the observability constraint in the NMPC setup is computationally not demanding in the case of the collision avoidance maneuver. The authors are aware of the fact that in general introducing the constraint (7e) might lead to computational problems.*

5 Conclusions

In this paper the results presented in [2], which guarantee a certain degree of observability along NMPC trajectories, have been recapitulated. The practical relevance of this modified NMPC approach has been illustrated via an emergency collision avoidance maneuver for passenger cars. It has been shown that, without loss of controller performance, the observability properties of the vehicle along the obtained trajectories could be improved as well as the performance of the lateral velocity observer when compared to a classical NMPC approach.

References

1. Antonov, S., Fehn, A., Kugi, A.: Ein neuartiger Ansatz zur Querdynamikregelung von Personenkraftwagen (A new approach to lateral dynamics control of passenger vehicles). at-Automatisierungstechnik 55(9), 488–496 (2007)
2. Böhm, C., Findeisen, R., Allgöwer, F.: Avoidance of poorly observable trajectories: A predictive control perspective. In: Proceedings of the 17th IFAC World Congress, Seoul, pp. 1952–1957 (2008)
3. Böhm, C., Raff, T., Findeisen, R., Allgöwer, F.: Calculating the terminal region for Lure systems via LMIs. In: Proceedings of the American Control Conference, Seattle, pp. 1127–1132 (2008)
4. Brandin, V.N., Kostyukovskii, Y.M.L., Razorenov, G.N.: Global observability conditions for nonlinear dynamic systems. Automation and Remote Control 36, 1585–1591 (1976)
5. Falcone, P., Borrelli, F., Asgari, J., Tseng, H.E., Hrovat, D.: Predictive active steering control for autonomous vehicle systems. IEEE Transactions on Control Systems Technology 15(3), 566–580 (2007)
6. Fontes, F.A.: A general framework to design stabilizing nonlinear model predictive controllers. System and Control Letters 42(2), 127–142 (2000)
7. Heß, F.: Autonomous vehicle collision avoidance with predictive control. Diploma thesis, University of Stuttgart (2008)
8. Imsland, L., Johansen, T.A., Fossen, T.I., Fjær Grip, H., Kalkkuhl, J.C., Suissa, A.: Vehicle velocity estimation using nonlinear observers. Automatica 42(12), 2091–2103 (2006)
9. Krener, A.J.: Nonlinear stabilizability and detectability. In: Helmke, U., Mennicken, R., Saurer, J. (eds.) Systems and Networks: Mathematical Theory and Applications, pp. 231–250 (1994)

10. Litz, L.: Modale Maße für Steuerbarkeit, Beobachtbarkeit, Regelbarkeit und Dominanz - Zusammenhänge, Schwachstellen, neue Wege. Regelungstechnik 31, 148–158 (1983)
11. Nijmeijer, H., van der Schaft, A.: Nonlinear Dynamical Control Systems. Springer, Heidelberg (1990)
12. Pacejka, H.B., Bakker, E.: The magic formula tyre model. Vehicle System Dynamics 21, 1–18 (2004)
13. Vargas, A.: Observer Design for Nonlinear Systemes with Reduced Observability Properties. PhD thesis, University of Stuttgart (2003)
14. Zanten, V.A.T.: Bosch ESP systems: 5 years of experience. SAE Technical Paper No. 2000-01-1633 (2000)
15. Zeitz, M., Xia, X.: On nonlinear continuous observers. International Journal of Control 66(6), 943–954 (1997)

State Estimation and Fault Tolerant Nonlinear Predictive Control of an Autonomous Hybrid System Using Unscented Kalman Filter

J. Prakash, Anjali P. Deshpande, and Sachin C. Patwardhan

Abstract. In this work, we propose a novel fault tolerant nonlinear model predictive control (FTNMPC) scheme for dealing with control problems associated with an autonomous nonlinear hybrid system (NHS). To begin with, we develop a scheme for state estimation of continuous as well as discrete states for autonomous NHS using unscented Kalman filter (UKF), a derivative free nonlinear state estimator, and further use it for formulating an NMPC scheme. The salient feature of the NMPC scheme is that the concept of sigma point propagation in UKF is extended to carry out the future trajectory predictions. We then proceed to develop a nonlinear version of generalized likelihood ratio (GLR) method that employs UKF for diagnosing sensor and/or actuator faults. The diagnostic information generated by the nonlinear GLR method is used for on-line correction of the measurement vector, the model used for state estimation/prediction and constraints in the NMPC formulation. The efficacy of the proposed state estimation, diagnosis and control schemes is demonstrated by conducting simulation studies on the benchmark three-tank hybrid system. Analysis of the simulation results reveals that the FTNMPC scheme facilitates significant recovery in the closed loop performance particularly on occurrence of sensor faults.

Keywords: Hybrid Systems, Unscented Kalman Filter, Fault Tolerant Control.

J. Prakash
MIT Campus, Anna University, Chennai, India
e-mail: prakaiit@rediffmail.com

Anjali P. Deshpande
Systems and Control Engineering, Indian Institute of Technology Bombay, Mumbai, India
e-mail: apdesh@iitb.ac.in

Sachin C. Patwardhan
Dept. of Chemical Engineering, Indian Institute of Technology Bombay, Mumbai, India
e-mail: sachinp@iitb.ac.in

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 285–293.
springerlink.com

1 Introduction

Dynamic systems that involve continuous and discrete states, broadly classified as hybrid systems, are often encountered in engineering applications. State estimation, fault diagnosis and estimator based predictive control of nonlinear hybrid systems poses a challenging problem as these systems involve discontinuities that are introduced by switching of the discrete variables. State estimation and model predictive control of hybrid system has gained increased attention in the recent years [1, 2]. However, many approaches available in the literature for predictive control of hybrid systems are based on linear or piecewise linear hybrid models and not much work has been reported on state estimation and control of nonlinear hybrid systems (NHS). Extended Kalman filter (EKF) is a popular choice for state estimation while formulating estimation and predictive control schemes for nonlinear processes. However, the state covariance propagation step in EKF requires computation of Jacobian of the nonlinear state transition operator at each sampling instant. This implies that the nonlinear state transition operator and the state-output map should be smooth and at least once differentiable. However, the dynamic models for hybrid systems involve discontinuities, which are introduced by switching of the discrete variables. Therefore, EKF cannot be used for state estimation of nonlinear hybrid systems particularly in the operating regimes where discrete variables undergo frequent transitions. In recent years, a number of derivative free nonlinear filtering techniques have been proposed in the literature. For example, the unscented Kalman filter (UKF) [3] has been proposed as an alternative to EKF where the above limitations can be overcome using the concept of (deterministic) sample statistics. Thus, UKF appears to be an appropriate choice for developing state estimation based predictive control scheme for NHS.

While the controller design and implementation forms an important component the operating strategy for an NHS, detection and isolation of faults (abrupt changes / drifts in models parameters / sensors / actuators) and (sensor / actuator) failures during operation is equally important aspect from the viewpoint of maintaining smooth operation over a long period of time. The area of diagnosis of malfunctioning in hybrid systems, however, has not yet been adequately addressed in the literature. Recently, Deshpande at al. [4] have developed an intelligent EKF based state estimation scheme that uses generalized likelihood ratio (GLR) based fault diagnosis strategy to isolate and identify faults / failures on-line. The fault / failure information is used to modify the state estimator and develop a fault tolerant NMPC scheme that can accommodate the faults and failures. In this work, we extend this approach to NHS using UKF based estimation and diagnosis scheme. The efficacy of the proposed state estimation, diagnosis and control schemes is demonstrated by conducting simulation studies on the benchmark three-tank hybrid system.

2 State Estimation Using UKF

Consider an autonomous hybrid system that can be represented by the following set of differential algebraic equations

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \int_{kT}^{(k+1)T} \mathbf{F}\left[\mathbf{x}(\tau),\, \mathbf{u}(k),\, \mathbf{z}(\tau)\right] d\tau + \mathbf{w}(k) \tag{1}$$

$$\mathbf{z}(\tau) = \mathbf{G}[\mathbf{x}(\tau)] \tag{2}$$

$$\mathbf{y}(k) = \mathbf{H}[\mathbf{x}(k)] + \mathbf{v}(k) \tag{3}$$

In the above model, $\mathbf{x} \in R^n$ represents the system state vector , $\mathbf{u} \in R^m$ represents manipulated input vector, $\mathbf{w} \in R^d$ represents random state disturbances with known distribution, $\mathbf{y} \in R^r$ represents the measured outputs and $\mathbf{v} \in R^r$ represents the measurement noise with known distribution. In equation (2), $\mathbf{z} \in R^h$ represent discrete variables such that it can take only finite integer values, such as {-1, 0, 1} depending on some events, which are functions of states in the case of autonomous hybrid systems. The function vector $\mathbf{G}[.]$ can be expressed using a combination of Dirac delta functions and logical operators, such as AND, OR, XOR etc.

The UKF formulation uses a deterministic sampling technique to select a minimal finite set of sample points $\{\widehat{\mathbf{x}}^{(j)}(k-1|k-1) : j = 0, 1, ...2n\}$ around $\{\widehat{\mathbf{x}}(k-1|k-1)\}$ as follows

$$\left\{\widehat{\mathbf{x}}(k-1|k-1), \widehat{\mathbf{x}}(k-1|k-1) + \boldsymbol{\eta}^{(j)}(k-1), \widehat{\mathbf{x}}(k-1|k-1) - \boldsymbol{\eta}^{(j)}(k-1)\right\}$$

where n represents the dimension of the state vector and the vector $\boldsymbol{\eta}^{(j)}(k-1)$ is the j^{th} column of matrix $\sqrt{(n+\kappa)\widehat{\mathbf{P}}(k-1|k-1)}$, where κ is a tuning parameter [3]. These sigma points are then propagated through the system dynamics to compute a cloud of transformed points as follows:

$$\widehat{\mathbf{x}}^{(j)}(k|k-1) = \widehat{\mathbf{x}}^{(j)}(k-1|k-1) + \int_{(k-1)T}^{kT} \mathbf{F}\left[\mathbf{x}(\tau), \mathbf{u}(k-1), \mathbf{G}[\mathbf{x}(\tau)], \theta\right] d\tau \tag{4}$$

where $j = 0$ to $2n$. The statistics of the nonlinearly transformed points are then computed as follows:

$$\widehat{\mathbf{x}}(k|k-1) = \sum_{j=0}^{2n} \omega_j \widehat{\mathbf{x}}^{(j)}(k|k-1) \;\; ; \;\; \widehat{\mathbf{y}}(k|k-1) = \sum_{j=0}^{2n} \omega_j \mathbf{H}[\widehat{\mathbf{x}}^{(j)}(k|k-1)\,] \tag{5}$$

$$\mathbf{P}_{\varepsilon,e}(k) = \sum_{j=0}^{2n} \omega_j \boldsymbol{\varepsilon}^{(j)}(k)[\mathbf{e}^{(j)}(k)]^T \;\; ; \;\; \mathbf{P}_{e,e}(k) = \sum_{j=0}^{2n} \omega_j \mathbf{e}^{(j)}(k)[\mathbf{e}^{(j)}(k)]^T \tag{6}$$

$$\boldsymbol{\varepsilon}^{(j)}(k) = \widehat{\mathbf{x}}^{(j)}(k|k-1) - \widehat{\mathbf{x}}(k|k-1); \; \mathbf{e}^{(j)}(k) = \mathbf{H}[\widehat{\mathbf{x}}^{(j)}(k|k-1)\,] - \widehat{\mathbf{y}}(k|k-1)$$

Here, ω_j represents fixed weights associated with sigma points, which are chosen as follows:

$$\left\{\omega_0 = \frac{\kappa}{n+\kappa}, \omega_j = \frac{1}{2(n+\kappa)} : j = 1, 2, ...n)\right\} \tag{7}$$

These estimated covariances are then used for updating Kalman gain matrix, $\mathbf{L}(k) = \mathbf{P}_{\varepsilon,e}(k)[\mathbf{P}_{e,e}(k)]^{-1}$, and the estimates of continuous and discrete states are generated as follows:

$$\widehat{\mathbf{x}}(k|k) = \widehat{\mathbf{x}}(k|k-1) + \mathbf{L}(k)\boldsymbol{\gamma}(k) \;\; ; \;\; \boldsymbol{\gamma}(k) = \mathbf{y}(k) - \widehat{\mathbf{y}}(k|k-1) \tag{8}$$

$$\widehat{\mathbf{z}}(k|k) = \mathbf{G}[\widehat{\mathbf{x}}(k|k)] \tag{9}$$

The posterior covariance necessary for generation of sigma points at the subsequent time instant is estimated as

$$\mathbf{P}(k|k-1) = \sum\nolimits_{j=0}^{2n} \omega_j \boldsymbol{\varepsilon}^{(j)}(k)[\boldsymbol{\varepsilon}^{(j)}(k)]^T \;\; ; \;\; \mathbf{P}(k|k) = \mathbf{P}(k|k-1) - \mathbf{L}(k)\mathbf{P}_{e,e}(k)\mathbf{L}(k)^T \tag{10}$$

In remainder of the text, we refer to this UKF developed under fault free conditions as *normal UKF*.

3 UKF Based NMPC Formulation

A distinguishing feature of UKF is that it explicitly accounts for the uncertainty in the initial state through propagation of sigma points in the prediction step. We propose to use this feature for capturing the propagated effects of the uncertainty in the initial state on the future trajectory predictions. Thus, given a set of future manipulated input moves, $\mathbf{U}_k \equiv \{\mathbf{u}(k+l|k) : l = 0, 1, ..p-1\}$, new sigma points are generated at each prediction step along the future trajectory as follows $\left\{\widehat{\mathbf{x}}(k+l|k), \widehat{\mathbf{x}}(k+l|k) + \boldsymbol{\eta}^{(j)}(k+l|k), \widehat{\mathbf{x}}(k+l|k) - \boldsymbol{\eta}^{(j)}(k+l|k)\right\}$ for $l = 0, 1, ..., p-1$ where $\boldsymbol{\eta}^{(j)}(k+l|k)$ represents the j^{th} column of matrix $\sqrt{(n+\kappa)\mathbf{P}(k+l|k)}$ and p represents prediction horizon. The predicted mean is computed as follows

$$\widehat{\mathbf{x}}^{(j)}(k+l+1|k) = \widehat{\mathbf{x}}^{(j)}(k+l|k) + \int_{(k+l)T}^{(k+l+1)T} F\left[\mathbf{x}(\tau), \mathbf{u}(k+l|k), G[\mathbf{x}(\tau)]\right] d\tau$$

$$\widehat{\mathbf{x}}(k+l+1|k) = \sum\nolimits_{j=0}^{2n} \omega_j \widehat{\mathbf{x}}^{(j)}(k+l+1|k) \tag{11}$$

for $j = 0, 1, ...2n$ and $l = 0, 1, ...p-1$. The prediction error covariance matrix required in the above step is updated as follows

$$\mathbf{P}(k+l|k) = \sum\nolimits_{j=0}^{2n} \omega_j \left[\boldsymbol{\varepsilon}^{(j)}(k+l|k)\right] \left[\boldsymbol{\varepsilon}^{(j)}(k+l|k)\right]^T \tag{12}$$

To account for the model plant mismatch, we propose to correct the predictions as follows

$$\widehat{\mathbf{x}}(k+l+1|k) = \widehat{\mathbf{x}}(k+l+1|k) + \mathbf{L}(k)\boldsymbol{\beta}(k) \tag{13}$$
$$\widehat{\mathbf{z}}(k+l+1|k) = \mathbf{G}[\widehat{\mathbf{x}}(k+l+1|k)] \tag{14}$$

$$\boldsymbol{\beta}(k) = \boldsymbol{\Phi}_e\boldsymbol{\beta}(k-1) + [\mathbf{I} - \boldsymbol{\Phi}_e]\,[\mathbf{y}(k) - \widehat{\mathbf{y}}(k|k-1)] \tag{15a}$$

$$\widehat{\mathbf{y}}(k+l+1|k) = \mathbf{H}[\widehat{\mathbf{x}}(k+l+1|k)] + \mathbf{d}(k) \tag{16}$$

$$\mathbf{d}(k) = \boldsymbol{\Phi}_d\mathbf{d}(k-1) + [\mathbf{I} - \boldsymbol{\Phi}_d]\,[\mathbf{y}(k) - \widehat{\mathbf{y}}(k|k)] \tag{17}$$

Here, the unit gain filter matrices $\boldsymbol{\Phi}_e$ and $\boldsymbol{\Phi}_d$ are treated as tuning parameters. At any sampling instant k, the NMPC problem is formulated as follows

$$\min_{\mathbf{U}_k} \sum_{l=1}^{p} \mathbf{e}_f(k+l|k)^T\mathbf{W}_E\mathbf{e}_f(k+l|k) + \sum_{l=0}^{q-1} \Delta\mathbf{u}(k+l|k)^T\mathbf{W}_U\Delta\mathbf{u}(k+l|k) \tag{18}$$
$$\mathbf{e}_f(k+l|k) = \mathbf{r} - \widehat{\mathbf{y}}(k+l|k) \quad ; \quad \Delta\mathbf{u}(k+l|k) = \mathbf{u}(k+l|k) - \mathbf{u}(k+l-1|k)$$

subject to constraints

$$\mathbf{u}^L \leq \mathbf{u}(k+l|k) \leq \mathbf{u}^U \quad ; \quad \Delta\mathbf{u}^L \leq \Delta\mathbf{u}(k+l|k) \leq \Delta\mathbf{u}^U$$

where $l = 0, 1, ..., q-1$. Here, q represents control horizon such that $\Delta\mathbf{u}(k+l|k) = \overline{0}$ for $l > q$ and $\mathbf{r} \in R^r$ represents the setpoint.

4 Fault Diagnosis

When the process starts behaving abnormally, the first task is to detect the deviations from the normal operating conditions. To simplify the task of fault detection, it is further assumed that, under normal operating conditions, the innovation sequence $\{\boldsymbol{\gamma}(k)\}$ generated by *normal UKF* is a zero mean Gaussian white noise sequence with covariance $\mathbf{P}_{e,e}(k)$. A significant and sustained departure from this behavior is assumed to result due to occurrence of a fault. To detect such departures systematically, simple statistical tests, namely fault detection test and fault conformation test, which are based on comparison of the log likelihood function of innovations with thresholds from standard chi-square distribution, are applied on the innovations obtained from the normal UKF [4].

To identify the fault(s) that might have occurred, it is necessary to develop a model for each hypothesized fault that describes its effect on the evolution of the process variables. It is assumed that a bias in sensor measurement develops as an abrupt (step-like) change. Suppose a bias develops in i^{th} sensor at instant t, then it is modeled as follows:

$$\mathbf{y}_{y_i}(k) = \mathbf{H}\left[\mathbf{x}(k)\right] + b_{y_i}\mathbf{e}_{y_i}\sigma(k-t) + \mathbf{v}(k) \tag{19}$$

where b_{y_i} represents bias magnitude, $\mathbf{e}_{y_i}$ represents a unit vector with i^{th} element equal to unity and all other elements equal to zero and $\sigma(k-t)$ represents a unit step function. Similarly, if a bias of magnitude b_{u_i} occurs in actuator i at time t, then

$$\mathbf{m}(k) = \mathbf{u}(k) + b_{u_i}\mathbf{e}_{u_i}\sigma(k-t) \tag{20}$$

where represents $\mathbf{u}(k)$ controller output and $\mathbf{m}(k)$ represents input to plant and $\mathbf{e}_{u_i}$ is the corresponding fault direction vector. It is possible to develop models for abrupt changes in model parameters and unmeasured disturbances in a similar manner (ref. [4]).

Once occurrence of a fault is detected at instant t, we collect data over time window $[t, t+N]$. Following the nonlinear GLR approach developed by Deshpande et al. [4], a separate UKF is formulated for each hypothesized fault model (referred to as *fault mode UKF*) over the time window $[t, t+N]$ with the assumption that the fault has occurred at time instant t. For example, assuming that a bias in i^{th} sensor has occurred at instant t, the process behavior over window $[t, t+N]$ can be described by equations (1), (2) and (19). These equations are used to formulate the corresponding fault mode UKF and the innovation sequence is computed as $\boldsymbol{\gamma}_{y_i}(k) = \mathbf{y}(k) - \widehat{\mathbf{y}}_{y_i}(k|k-1)$ where $\widehat{\mathbf{y}}_{y_j}$ represents predicted mean output under the fault hypothesis that i^{th} sensor has developed a bias. For each hypothesized fault, the the fault magnitude is estimated by formulating and solving a nonlinear optimization problems over the time window $[t, t+N]$ as follows

$$\min_{\widehat{b}_{f_i}} \left(\mathbf{J}_{f_i}\right) = \sum\nolimits_{k=t}^{t+N} \boldsymbol{\gamma}_{f_i}(k)[\mathbf{P}_{e_{f_i},e_{f_i}}(k)]^{-1}\boldsymbol{\gamma}_{f_i}(k) \tag{21}$$

where, $\boldsymbol{\gamma}_{f_i}(k)$ represent innovations and $[\mathbf{P}_{e_{f_i},e_{f_i}}(k)]^{-1}$ the innovations covariance matrices, respectively, computed using the *fault mode UKF* corresponding to fault f_i. The *fault mode UKF* that best explains the measurement sequence $\{\mathbf{y}(t).....\mathbf{y}(t+N)\}$ is one for which the value of $(\mathbf{J}_{f_i})$ is minimum. Thus, the fault f_i that corresponds to minimum value of $(\mathbf{J}_{f_i})$ is isolated as the fault that has occurred at time t with $\widehat{b}_{f_i}$ as its corresponding magnitude estimate.

5 NMPC with Fault Accommodation

Consider a situation when a fault is detected for the first time at instant t and assume that at instant $t+N$ a fault f_i has been identified using the UKF based NLGLR method. During the interval $[t, t+N]$, the NMPC formulation is based on the normal behavior prediction model. However, after the identification of the fault at instant $t+N$ for $k \geq t+N$, the model used for estimation and control is modified as follows:

- **Compensation for Sensor bias:** If i^{th} sensor bias is isolated, the measured output is compensated as $\mathbf{y}_c(k) = \mathbf{y}(k) - \widehat{b}_{y_i}\mathbf{e}_{y_i}$ and used in FDI as well as NMPC formulation for computing innovation sequence.
- **Compensation for Actuator bias:** If i^{th} actuator bias is isolated, the model used for state estimation and NMPC is modified as follows

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \int_{kT}^{(k+1)T} \mathbf{F}\left[\mathbf{x}(\tau), \mathbf{u}(k) + b_{u_i}\mathbf{e}_{u_i}, \mathbf{z}(\tau)\right] d\tau + \mathbf{w}(k) \quad (22)$$

The main concern with the above approach is that the magnitude and the position of the fault may not be accurately estimated. Thus, there is a need to introduce integral action in such a way that the errors in estimation of fault magnitude or position can be corrected in the course of time. Furthermore, other faults may occur at subsequent times. Thus, in the on-line implementation of NMPC, application of FDI method resumes at $t + N + 1$. The FDI method may identify a fault in the previously identified location or a new fault may be identified. In either case, the fault compensation can be modified using cumulative estimates [4], which are computed as

$$\widetilde{b}_{f_i} = \sum_{s=1}^{n_f} \widehat{b}_{fi}(s) \text{ with initial value } \widehat{b}_{f_i}(0) = 0 \quad (23)$$

where $f \in [u, y]$ denotes the fault type occurring at i^{th} position and n_f represents the number of times a fault of type f was confirmed and isolated in the i^{th} position. Thus, the measurement vector used in equations 8, 15a and 17 is replaced by corrected measurement, $\mathbf{y}_c(k) = \mathbf{y}(k) - \sum_{i=1}^{m} \widetilde{b}_{y_i}\mathbf{e}_{y_i}$, compensated using cumulative bias estimates. To accommodate biases in manipulated inputs, the model predictions (11) are generated using bias compensated future inputs, $\mathbf{u}(k+l|k) + \sum_{i=1}^{m} \widetilde{b}_{u_i}\mathbf{e}_{u_i}$, and the constraints on the future manipulated input in NMPC formulation are modified as follows

$$\mathbf{u}^L \leq \mathbf{u}(k+l|k) + \sum_{i=1}^{m} \widetilde{b}_{u_i}\mathbf{e}_{u_i} \leq \mathbf{u}^U$$

6 Simulation Study

The system considered to verify the proposed FTNMPC strategy is the benchmark three-tank hybrid system, (Prakash et. al. [5]). This system consists of three interconnected tanks and has three continuous states (level h_1, h_2 and h_3 in each tank). The three tanks are connected at the bottom and in the middle through pipes. Inflows to Tank 1 ($\mathbf{u}_1$) and to Tank 3 ($\mathbf{u}_5$) are treated as manipulated inputs. The governing equations for the three-tank hybrid system can be found in [5]. In this model, the dynamics due to the middle level connections between the tanks is captured through two discrete states, $\mathbf{z}_1$ and $\mathbf{z}_2$, which can take only finite integer values, such as {-1,0,1} depending on the level of fluids in the tanks. The control problem is to track level in the first and second tanks (h_1 and h_2) by manipulating valves $\mathbf{u}_1$ and

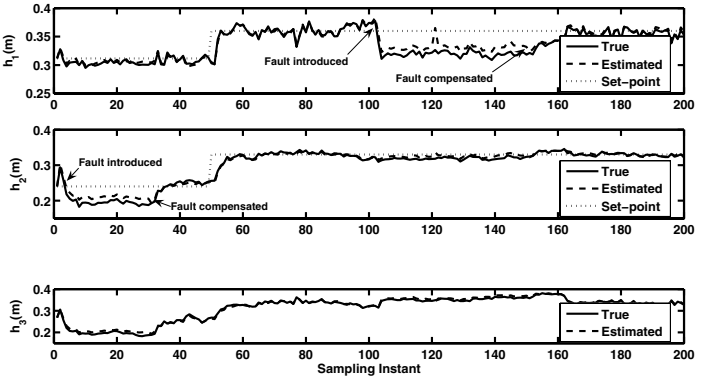


Fig. 1 Closed loop response of FTNMPC in presence of sensor biases

$\mathbf{u}_5$. The inputs are constrained as $0 < \mathbf{u}_1 < 1$ and $0 < \mathbf{u}_5 < 1$. The online FDI and compensation scheme has been integrated with state space based NMPC. The controller tuning parameters and the NLGLR tuning parameters are as follows

$$N = 30 \;\; ; \;\; \alpha_{FDT} = 0.75 \;\; ; \;\; \alpha_{FCT} = 0.01$$
$$p = 5, \;\; q = 1, \; \mathbf{W}_E = \mathbf{I} \;\; ; \; \mathbf{W}_{\Delta U} = [0] \; ; \; \mathbf{r} = \begin{bmatrix} 0.3117 \; 0.24 \end{bmatrix}$$

The covariance matrices of measurement noise and state noise are assumed as

$$\mathbf{R} = diag \begin{bmatrix} 6.2 \times 10^{-5} \; 4.8 \times 10^{-5} \; 5.4 \times 10^{-5} \end{bmatrix} \;\; ; \;\; \mathbf{Q} = 0.1\mathbf{R}$$

To test the efficacy of the proposed on-line FTC scheme, five simulation trials were performed with each trial lasting for 300 sampling instants. Simulation runs were carried out in the presence of multiple faults that occur sequentially in time. Four faults were hypothesized, namely biases in two sensors and two actuators. We have simulated the case in which the 0.05m ($\approx 5\sigma$) bias in level sensor of Tank 2 occur at 1^{st} sampling instant, which is followed by a 0.05m ($\approx 5\sigma$) bias in the level sensor of Tank 1 at 101^{th} sampling instant. The proposed FDI scheme successfully isolated the multiple sequential faults in the simulation trail. The cumulative bias estimates of fault magnitudes at the end of a simulation trial averaged over all trials are found to be 0.0487 m and 0.0494 m, respectively. These bias magnitudes have been used for on-line correction of measurements that are used in controller calculations. The response is shown in Figure 1. Figure 2 shows the evolution of true and estimated values of discrete states ($\mathbf{z}_1$ and $\mathbf{z}_2$). It can be observed from Figure 1 that, in the presence of bias in the sensor, the estimated states deviate from their true values and this results in offset in the outputs. However, when the measurements of the faulty sensors are compensated for sensor faults, state estimation improves and the outputs follow the given trajectory.

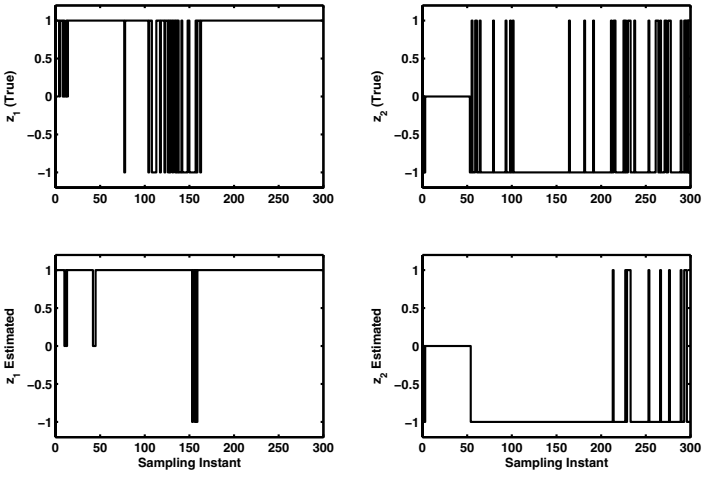


Fig. 2 Evolution of Discrete States

7 Conclusions

Analysis of the simulation results indicates that the proposed UKF based version of nonlinear GLR method accurately identifies the multiple sequential sensor faults. The proposed FTNMPC approach facilitates recovery of closed loop performance after the faults have been isolated and results in offset free behavior in the presence of sensor biases. The future work involves isolation of abrupt changes in model parameters and unmeasured disturbances.

References

1. Ferrari-Trecate, G., Mignone, D., Morari, M.: Moving Horizon Estimation for Hybrid Systems. IEEE Trans. on Automatic Control 47(10), 1663–1676 (2002)
2. Borrelli, F., Bemporad, A., Fodor, M., Hrovat, D.: An MPC/Hybrid System Approach to Traction Control. IEEE Trans. on Contol Systems Technology 14(3), 541–552 (2006)
3. Julier, S.J., Uhlmann, J.K.: Unscented Filtering and Nonlinear Estimation. Proceedings of the IEEE 92(3), 401–422 (2004)
4. Deshpande, A.P., Patwardhan, S.C., Narasimhan, S.: Intelligent State Estimation for Fault Tolerant Nonlinear Predictive Control. Journal of Process Control 19(2), 187–204 (2009)
5. Prakash, J., Patwardhan, S.C., Shah, S.L.: Control of an Autonomous Hybrid System using a Nonlinear Model Predictive Controller. In: Proc. of for IFAC World Congress, Korea (July 2008)

Design of a Robust Nonlinear Receding-Horizon Observer - First-Order and Second-Order Approximations

G. Goffaux and A. Vande Wouwer

Abstract. The objective of this study is to design a robust receding-horizon observer for systems described by nonlinear models with uncertain parameters. Robustification in the presence of model uncertainties naturally leads to the formulation of a nonlinear min-max optimization problem, which can be converted to a simpler minimization problem using approximation along a nominal trajectory. In this study, the suitability of first-order and second-order approximations is investigated. These methods are evaluated in simulation and with experimental data from continuous cultures of phytoplankton.

Keywords: State Estimation, Receding-horizon, Uncertain Systems, Bioprocesses.

1 Introduction

Software sensors play an increasingly important role in bioprocess monitoring. Facing uncertainties associated to the underlying bioprocess model, robust state estimation techniques are a solution of choice.

In this study, our objective is to design a receding-horizon state estimation method, which would be robust to parameter uncertainties. Receding-horizon estimation is a popular method and several applications can be found in, e.g., [1], [5].

A robust formulation of the problem naturally leads to a nonlinear min-max optimization problem. The latter can be converted to a simpler minimization problem by using a model linearization along a nominal trajectory (defined by nominal parameter values and the most likely initial conditions) and recent results in linear robust receding-horizon estimation developed in [2] can be used. However, this method is suboptimal as it deals with a first-order approximation of nonlinear functions. In this paper, a second-order error term is taken into account in the optimization problem.

G. Goffaux and A. Vande Wouwer
Service d'Automatique, Faculté Polytechnique de Mons, Boulevard Dolez 31,
B-7000 Mons, Belgium
e-mail: Guillaume.Goffaux,Alain.VandeWouwer@fpms.ac.be

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 295–304.
springerlink.com

The performance of this method is assessed with the estimation of the internal nitrogen quota and nitrate concentration in continuous cultures of phytoplankton, based on on-line measurements of the biovolume and an uncertain Droop model [6]. Both simulation and experimental data are considered.

This paper is organized as follows. Section 2 describes the class of nonlinear systems that we are considering and formulates the optimization problems underlying receding-horizon estimation. Section 3 describes the suboptimal method which makes uses of a model linearization and of recent results in linear robust estimation theory. The next section extends to second-order approximation and gives an upper bound of the corresponding error. In Section 5, the continuous culture of phytoplankton and Droop model are presented and the algorithm performance is demonstrated. Section 6 draws some conclusions.

2 Problem Statement

Let us first assume that the system can be modelled by the following equations for $t \geq t_0$:

$$(\Sigma): \quad \begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\theta}) & \mathbf{x}(t_0) = \mathbf{x}_0 \\ \mathbf{y}_k = \mathbf{y}(t_k) = C\mathbf{x}(t_k) + \boldsymbol{\varepsilon}(t_k) \end{cases} \tag{1}$$

The first part in (Σ) is the evolution equation and is represented by continuous-time differential equations where $\mathbf{f}$ is a vector of nonlinear functions, $\mathbf{x}(t) \in \Re^{n_x}$ is the vector of state variables, $\mathbf{u}(t) \in \mathcal{U} \subset \Re^{n_u}$ is the vector of inputs with $\mathcal{U}$ the set of admissible controls, a subset of the space of measurable bounded functions and $\boldsymbol{\theta} \in \Re^{n_\theta}$ is a parameter vector. The second part is the observation equation and is modelled by linear discrete-time equations where C is the measurement matrix, $\mathbf{y}(t_k) \in \Re^{n_y}$ is the vector of sampled measurements and $\boldsymbol{\varepsilon}(t_k) \in \Re^{n_y}$ is the measurement noise vector described by a Gaussian white noise sequence with a zero mean and a covariance matrix $Q(t_k)$. $\mathbf{x}_0$ is the initial state vector containing values of the state at the initial time t_0. As usual in practice, the parameter vector has been estimated using some identification procedure based on experimental data, and parameter intervals $[\boldsymbol{\theta}^- \boldsymbol{\theta}^+]$ have been evaluated (for instance using the knowledge about the error covariance matrix) which are likely to enclose the unknown vector $\boldsymbol{\theta}$. Finally, let us define $\mathbf{x}(t) = \mathbf{g}_{t_0}(t, \mathbf{x}_0, \boldsymbol{\theta})$ the state vector at t computed thanks to (1) from the initial state $\mathbf{x}_0$ at t_0 and corresponding to the input trajectory from $\mathbf{u}(t_0)$ to $\mathbf{u}(t)$.

2.1 Receding-Horizon Estimation

The basic principle of the receding-horizon observer is to compute estimates of the state trajectories using the process model and the best knowledge of the initial state vector resulting from an optimization procedure. Typically, the initial condition of a moving time frame is computed by minimizing the distance between the measurement data collected in the considered time frame and model prediction.

Let us consider a time interval containing $N+1$ measurement instants $\{t_{n-N},\ldots,t_n\}$ with measurements $\{\mathbf{y}_{n-N},\ldots,\mathbf{y}_n\}$. The typical receding-horizon optimization problem computes an estimation $\hat{\mathbf{x}}^{\circ}_{n-N,n}$ of the initial state $\mathbf{x}(t_{n-N})$:

$$\hat{\mathbf{x}}^{\circ}_{n-N,n} = \arg \min_{\hat{\mathbf{x}}_{n-N,n}} J_{n,N}(\hat{\mathbf{x}}_{n-N,n}) \quad \text{with} \tag{2}$$

$$J_{n,N}(\hat{\mathbf{x}}_{n-N,n}) = \left\|\hat{\mathbf{x}}_{n-N,n} - \bar{\mathbf{x}}^{\circ}_{n-N}\right\|^2_M + \sum_{k=n-N}^{n} \left\| C\mathbf{g}_{t_{n-N}}(t_k, \hat{\mathbf{x}}_{n-N,n}, \boldsymbol{\theta}) - \mathbf{y}_k \right\|^2_{Q^{-1}(t_k)} \tag{3}$$

where $\|v\|_P = \left(v^T P v\right)^{1/2}$ is a weighted norm. Knowing $\hat{\mathbf{x}}^{\circ}_{n-N,n}$, estimations until the next measurement time t_{n+1} are given by : $\hat{\mathbf{x}}(t) = \mathbf{g}_{t_{n-N}}\left(t, \hat{\mathbf{x}}^{\circ}_{n-N,n}, \boldsymbol{\theta}\right) \quad \forall t \in [t_n, t_{n+1}]$. The first term in (3), weighted by the matrix M, expresses the belief in the estimation of the initial conditions obtained from all the information collected up to n. Matrix M is assumed to be positive definite and can be considered as a design parameter. Depending on the value of n, three situations can be distinguished [5]:

- $0 \le n \le N_{min} - 1$ $\left(\text{s.t. } \mathrm{N_{min} n_y < n_x}\right)$: $\hat{\mathbf{x}}^{\circ}_{0,n} = \bar{\mathbf{x}}_0$ i.e. the best a priori available initial guess, in the absence of more on-line information.
- $N_{min} \le n \le N$: A full-horizon estimation scheme is then applied (computing $\hat{\mathbf{x}}^{\circ}_{0,n}$). $\bar{\mathbf{x}}^{\circ}_0 = \hat{\mathbf{x}}^{\circ}_{0,n-1}$ is the solution of the optimization problem (2) on a time window of $n \le N$ time instants.
- $n > N$: In this case, enough information is available and a receding-horizon can be used. The "reference" initial conditions of the time window $\bar{\mathbf{x}}^{\circ}_{n-N}$ are based on the optimal estimation computed in the previous step (t_{n-N-1}) and a one-step prediction computed with the system model : $\bar{\mathbf{x}}^{\circ}_{n-N} = \mathbf{g}_{t_{n-N-1}}\left(t_{n-N}, \hat{\mathbf{x}}^{\circ}_{n-N-1,n-1}, \boldsymbol{\theta}\right)$.

2.2 *Robust Receding-Horizon Estimation*

The previous procedure assumes a perfect knowledge of the parameter vector $\boldsymbol{\theta}$. In the case of uncertain parameters (a parameter interval $[\boldsymbol{\theta}^-, \boldsymbol{\theta}^+]$ is nonetheless available), an alternative procedure can be proposed implying a min-max optimization (see [2] for the linear formulation):

$$\hat{\mathbf{x}}^{\circ}_{n-N,n} = \arg \min_{\hat{\mathbf{x}}_{n-N,n}} \max_{\hat{\boldsymbol{\theta}} \in [\boldsymbol{\theta}^-, \boldsymbol{\theta}^+]} \bar{J}_{n,N}\left(\hat{\mathbf{x}}_{n-N,n}, \hat{\boldsymbol{\theta}}\right) \quad \text{with} \tag{4}$$

$$\bar{J}_{n,N}() = \left\|\hat{\mathbf{x}}_{n-N,n} - \bar{\mathbf{x}}^{\circ}_{n-N}\right\|^2_M + \sum_{k=n-N}^{n} \left\| C\mathbf{g}_{t_{n-N}}\left(t_k, \hat{\mathbf{x}}_{n-N,n}, \hat{\boldsymbol{\theta}}\right) - \mathbf{y}_k \right\|^2_{Q^{-1}(t_k)}$$

- If $N_{min} \le n \le N$: $\bar{\mathbf{x}}^{\circ}_0 = \hat{\mathbf{x}}^{\circ}_{0,n-1}$
- If $n > N$: $\bar{\mathbf{x}}^{\circ}_{n-N} = \mathbf{g}_{t_{n-N-1}}\left(t_{n-N}, \hat{\mathbf{x}}^{\circ}_{n-N-1,n-1}, \boldsymbol{\theta}_{nom}\right)$ with $\boldsymbol{\theta}_{nom} = \frac{\boldsymbol{\theta}^- + \boldsymbol{\theta}^+}{2}$.

When $n > N$, the "reference" initial conditions of the time window $\bar{\mathbf{x}}^{\circ}_{n-N}$ are based on the optimal estimation computed in the previous step (t_{n-N-1}) and a one-step prediction computed with a nominal model. Indeed, as the time window is shifted

of one step ahead, the previous knowledge on the initial conditions is "extrapolated" using an average process model.

3 On the Use of Linearization Techniques

To reduce the computational demand, we will make use of linearization techniques. Indeed, in the linear case, a theorem is available [9] for converting a min-max optimization problem into a standard minimization problem (computing $\lambda^\circ \in \Re$):

Theorem 3.1. *Consider a regularized robust least-squares problem of the form :*

$$\min_{z} \max_{\|S\| \leq 1} \|z\|_V^2 + \|[D + \delta D(S)]z - [e + \delta e(S)]\|_W^2 \tag{5}$$

$\delta D(S)$ *and* $\delta e(S)$ *are additive perturbations modelled by*

$$\delta D(S) = HSE_d, \quad V > 0, \quad \delta e(S) = HSE_e, \quad W \geq 0$$

H, E_d *and* E_e *are known matrices of appropriate dimensions.* S *is an arbitrary contraction,* $\|S\| \leq 1$.

Problem (5) *has a unique global minimum* z° *given by:*

$$z^\circ = \left(\hat{V} + D^T \hat{W} D\right)^{-1} \left(D^T \hat{W} e + \lambda^\circ E_d^T E_e\right)$$

where $\hat{V} = V + \lambda^\circ E_d^T E_d$, $\hat{W} = W + WH\left(\lambda^\circ I - H^T W H\right)^\dagger H^T W$ *and the scalar parameter* λ° *is determined as*

$$\lambda^\circ = \arg \min_{\lambda \geq \|H^T W H\|} \|z(\lambda)\|_V^2 + \lambda \|E_d z(\lambda) - E_e\|^2 + \|Dz(\lambda) - e\|_{\hat{W}(\lambda)}^2$$

$$z(\lambda) = \left(\hat{V}(\lambda) + D^T \hat{W}(\lambda) D\right)^{-1} \left(D^T \hat{W}(\lambda) e + \lambda E_d^T E_e\right)$$
$$\hat{V}(\lambda) = V + \lambda E_d^T E_d$$
$$\hat{W}(\lambda) = W + WH\left(\lambda I - H^T W H\right)^\dagger H^T W$$

Matrix norm, like e.g. $\|P\|$ is related to the maximum singular value of the corresponding matrix i.e. $\|P\| = \left(\bar{\sigma}\left(P^T P\right)\right)^{1/2}$ with $\bar{\sigma}(P)$ the maximum eigenvalue of P. † denotes the left pseudoinverse.

Proof. A demonstration can be found in [9].

In order to apply this theorem to the min-max optimization problem (4), we have to linearize the nonlinear function $\mathbf{g}_{t_{n-N}}(t_k, \hat{\mathbf{x}}_{n-N,n}, \boldsymbol{\theta})$ with respect to $\bar{\mathbf{x}}_{n-N}^\circ$ and $\boldsymbol{\theta}_{nom}$ [4]:

$$\mathbf{g}_{t_{n-N}}(t_k, \hat{\mathbf{x}}_{n-N,n}, \boldsymbol{\theta}) \approx$$
$$\mathbf{g}_{nom}(n-N, t_k) + G_x(n-N, t_k)\left(\hat{\mathbf{x}}_{n-N,n} - \bar{\mathbf{x}}_{n-N}^\circ\right) + G_\theta(n-N, t_k)\left(\boldsymbol{\theta} - \boldsymbol{\theta}_{nom}\right)$$

$$with \quad \mathbf{g}_{nom}(n-N,t_k) = \mathbf{g}_{t_{n-N}}\left(t_k, \bar{\mathbf{x}}^{\circ}_{n-N}, \boldsymbol{\theta}_{nom}\right) \tag{6}$$

$$G_x(n-N,t_k) = \left.\frac{\partial \mathbf{g}_{t_{n-N}}(t_k,\mathbf{x}_{n-N},\boldsymbol{\theta})}{\partial \mathbf{x}_{n-N}}\right|_{\substack{\mathbf{x}_{n-N}=\bar{\mathbf{x}}^{\circ}_{n-N} \\ \boldsymbol{\theta}=\boldsymbol{\theta}_{nom}}} \tag{7}$$

$$G_\theta(n-N,t_k) = \left.\frac{\partial \mathbf{g}_{t_{n-N}}(t_k,\mathbf{x}_{n-N},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\substack{\mathbf{x}_{n-N}=\bar{\mathbf{x}}^{\circ}_{n-N} \\ \boldsymbol{\theta}=\boldsymbol{\theta}_{nom}}} \tag{8}$$

Jacobian matrices defined in (7) and (8) can be computed for time $t \in [t_{n-N}, t_n]$ by solving numerically the following differential equations together with (1) (for $\mathbf{x}(t_{n-N}) = \bar{\mathbf{x}}^{\circ}_{n-N}$ and $\boldsymbol{\theta} = \boldsymbol{\theta}_{nom}$):

$$\dot{G}_x(n-N,t) = \frac{\partial \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t),\boldsymbol{\theta}_{nom})}{\partial \mathbf{x}} G_x(n-N,t) \tag{9}$$

$$\dot{G}_\theta(n-N,t) = \frac{\partial \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t),\boldsymbol{\theta}_{nom})}{\partial \mathbf{x}} G_\theta(n-N,t) + \left.\frac{\partial \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t),\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{nom}} \tag{10}$$

The initial conditions of (9) and (10) are respectively given by :

$$G_x(n-N,t_{n-N}) = \frac{\partial \mathbf{g}_{t_{n-N}}(t_{n-N},\mathbf{x}_{n-N},\theta)}{\partial \mathbf{x}_{n-N}} = \frac{\partial \mathbf{x}_{n-N}}{\partial \mathbf{x}_{n-N}} = I_{n_x}$$

$$G_\theta(n-N,t_{n-N}) = 0_{n_x}$$

$I_{n_x} \in \Re^{n_x \times n_x}$ is the identity matrix and $0_{n_x} \in \Re^{n_x \times n_x}$ is the zero matrix. A new cost function can now be given by the following expression :

$$\begin{aligned} \check{J}_{n,N}\left(\hat{\mathbf{x}}_{n-N,n}, \hat{\boldsymbol{\theta}}\right) &= \left\|\hat{\mathbf{x}}_{n-N,n} - \bar{\mathbf{x}}^{\circ}_{n-N}\right\|^2_M \\ &+ \left\|\bar{G}^n_{x,n-N}\left(\hat{\mathbf{x}}_{n-N,n} - \bar{\mathbf{x}}^{\circ}_{n-N}\right) - \left(Y^n_{n-N} - \bar{\mathbf{g}}^n_{n-N} - \bar{G}^n_{\theta,n-N} S \Delta\boldsymbol{\theta}_{max}\right)\right\|^2_W \end{aligned} \tag{11}$$

with $\Delta\boldsymbol{\theta}_{max} = \frac{\boldsymbol{\theta}^+ - \boldsymbol{\theta}^-}{2}$ and $\|S\| \leq 1$.

$W = diag\left(Q^{-1}(t_{n-N}), \ldots, Q^{-1}(t_n)\right)$ is a bloc diagonal matrix, $Y^n_{n-N} = \left[y^T(t_{n-N}), \ldots, y^T(t_n)\right]^T$ the column vector containing all the measurements, $\bar{\mathbf{g}}^n_{n-N} = \left[\mathbf{g}^T_{nom}(n-N, t_{n-N})C^T, \ldots, \mathbf{g}^T_{nom}(n-N,t_n)C^T\right]^T$ the column vector containing the measured state vectors for the "nominal" case $\left(\bar{\mathbf{x}}^{\circ}_{n-N}, \boldsymbol{\theta}_{nom}\right)$, $\bar{G}^n_{x,n-N} = \left[G^T_x(n-N,t_{n-N})C^T, \ldots, G^T_x(n-N,t_n)C^T\right]^T$ re-grouping the Jacobian matrices related to the initial state variables and $\bar{G}^n_{\theta,n-N} = \left[G^T_\theta(n-N,t_{n-N})C^T, \ldots, G^T_\theta(n-N,t_n)C^T\right]^T$ regrouping the Jacobian matrices related to the parameters. Finally, the solution of the robust receding-horizon problem is summarized as :

Theorem 3.2. *The problem defined by (4) with (11) as cost function has a unique solution given by*

$$\hat{\boldsymbol{x}}^{\circ}_{n-N,n} = \bar{\boldsymbol{x}}^{\circ}_{n-N} + \left(M + \bar{G}^{n,T}_{x,n-N}\hat{W}\bar{G}^{n}_{x,n-N}\right)^{-1}\bar{G}^{n,T}_{x,n-N}\hat{W}\left(Y^{n}_{n-N} - \bar{\boldsymbol{g}}^{n}_{n-N}\right)$$

$$with \quad \hat{W}(\lambda^{\circ}) = W + W\bar{G}^{n}_{\theta,n-N}\left(\lambda^{\circ}I - \bar{G}^{n,T}_{\theta,n-N}W\bar{G}^{n}_{\theta,n-N}\right)^{\dagger}\bar{G}^{n,T}_{\theta,n-N}W$$

λ° *is computed from the following minimization*

$$\lambda^{\circ} = \arg\min_{\lambda \geq \left\|G^{n,T}_{\theta,n-N}WG^{n}_{\theta,n-N}\right\|} \|\bar{z}(\lambda)\|^2_M + \lambda\|\Delta\boldsymbol{\theta}_{max}\|^2 + \left\|G^{n}_{x,n-N}\bar{z}(\lambda) - \left(Y^{n}_{n-N} - \bar{\boldsymbol{g}}^{n}_{n-N}\right)\right\|^2_{\hat{W}(\lambda)}$$

$$with \quad \bar{z}(\lambda) = \left(M + \bar{G}^{n,T}_{x,n-N}\hat{W}(\lambda)G^{n}_{x,n-N}\right)^{-1}\bar{G}^{n,T}_{x,n-N}\hat{W}(\lambda)\left(Y^{n}_{n-N} - \bar{\boldsymbol{g}}^{n}_{n-N}\right)$$

$$\hat{W}(\lambda) = W + W\bar{G}^{n}_{\theta,n-N}\left(\lambda^{\circ}I - \bar{G}^{n,T}_{\theta,n-N}W\bar{G}^{n}_{\theta,n-N}\right)^{\dagger}\bar{G}^{n,T}_{\theta,n-N}W$$

Proof: *a trivial application of Theorem 3.1.*

The linearized model differs from the nonlinear one, and results in an approximation of the original problem. The quality of this approximation will mostly depend on the fact that the worst linearized model is as worse as the worst nonlinear one. In order to better handle this approximation, a second-order extension is considered.

4 Extension to Second-Order

Applying the mean value theorem, a remainder term to the first-order approximation (6)-(8) is given in (12). A more compact notation is used : $\boldsymbol{\Delta}_{\theta} = \boldsymbol{\theta} - \boldsymbol{\theta}_{nom}$, $\boldsymbol{\Delta}_{x_0} = \hat{\mathbf{x}}_{n-N,n} - \bar{\mathbf{x}}^{\circ}_{n-N}$.

$$R\left(t_k, \bar{\mathbf{x}}, \bar{\boldsymbol{\theta}}\right) = \frac{1}{2}\begin{bmatrix} \left[\boldsymbol{\Delta}^T_{x_0}, \boldsymbol{\Delta}^T_{\theta}\right] H_1\left(t_k, \bar{\mathbf{x}}, \bar{\boldsymbol{\theta}}\right) \\ \vdots \\ \left[\boldsymbol{\Delta}^T_{x_0}, \boldsymbol{\Delta}^T_{\theta}\right] H_{n_x}\left(t_k, \bar{\mathbf{x}}, \bar{\boldsymbol{\theta}}\right) \end{bmatrix}\begin{bmatrix} \boldsymbol{\Delta}_{x_0} \\ \boldsymbol{\Delta}_{\theta} \end{bmatrix} \tag{12}$$

$\bar{\mathbf{x}} \in \Re^{n_x}$ and $\bar{\boldsymbol{\theta}} \in \Re^{n_\theta}$ are two vectors between $\left\{\hat{\mathbf{x}}_{n-N,n}, \bar{\mathbf{x}}^{\circ}_{n-N}\right\}$ and $\{\boldsymbol{\theta}, \boldsymbol{\theta}_{nom}\}$ respectively. These conditions can be expressed using the Euclidean distance:

$$\begin{array}{rl} \left\|\hat{\mathbf{x}}_{n-N,n} - \bar{\mathbf{x}}^{\circ}_{n-N}\right\| & \geq \left\|\bar{\mathbf{x}} - \bar{\mathbf{x}}^{\circ}_{n-N}\right\| \\ & \geq \left\|\hat{\mathbf{x}}_{n-N,n} - \bar{\mathbf{x}}\right\| \end{array} \qquad \begin{array}{rl} \|\boldsymbol{\theta} - \boldsymbol{\theta}_{nom}\| & \geq \left\|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_{nom}\right\| \\ & \geq \left\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\right\| \end{array}$$

$H_i\left(t_k, \bar{\mathbf{x}}, \bar{\boldsymbol{\theta}}\right)$ is the Hessian matrix related to the i^{th} state variable and can be partitioned w.r.t. $\bar{\mathbf{x}}$ and $\bar{\boldsymbol{\theta}}$. To simplify notations, the nonlinear function $\mathbf{g}_{t_{n-N}}\left(t_k, \hat{\mathbf{x}}_{n-N,n}, \boldsymbol{\theta}\right)$ is noted : $\mathbf{g}^T = [g_1, \ldots, g_{n_x}]$ and one has the following partition.

$$H_i\left(t_k, \bar{\mathbf{x}}, \bar{\boldsymbol{\theta}}\right) = \begin{bmatrix} H^i_{xx}(t_k) & H^i_{x\theta}(t_k) \\ H^i_{\theta x}(t_k) & H^i_{\theta\theta}(t_k) \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 g_i}{\partial \mathbf{x}\partial \mathbf{x}} & \frac{\partial^2 g_i}{\partial \mathbf{x}\partial \boldsymbol{\theta}} \\ \frac{\partial^2 g_i}{\partial \boldsymbol{\theta}\partial \mathbf{x}} & \frac{\partial^2 g_i}{\partial \boldsymbol{\theta}\partial \boldsymbol{\theta}} \end{bmatrix} \text{ with } \begin{array}{l} \mathbf{x} = \bar{\mathbf{x}} \\ \boldsymbol{\theta} = \bar{\boldsymbol{\theta}} \end{array} \tag{13}$$

Moreover, Jacobian and Hessian matrices related to the evolution equations $\mathbf{f}^T = [f_1, \ldots, f_{n_x}]$ with respect to the state vector $\mathbf{x}$ and the parameter vector $\boldsymbol{\theta}$ are written $F_x = \frac{\partial \mathbf{f}()}{\partial \mathbf{x}}$, $F_\theta = \frac{\partial \mathbf{f}()}{\partial \boldsymbol{\theta}}$, $F^i_{xx} = \frac{\partial^2 f_i()}{\partial \mathbf{x}^2}$, $F^i_{\theta\theta} = \frac{\partial^2 f_i()}{\partial \boldsymbol{\theta}^2}$, $F^i_{x\theta} = \frac{\partial^2 f_i()}{\partial \mathbf{x}\partial \boldsymbol{\theta}}$ and $F^i_{\theta x} = \frac{\partial^2 f_i()}{\partial \boldsymbol{\theta}\partial \mathbf{x}}$. Hessian matrices (13) are computed by solving the following differential equations given in a matrix form:

$$\begin{aligned}
\dot{H}^i_{xx} &= G_x^T F^i_{xx} G_x + \sum_{j=1}^{n_x} \frac{\partial f_i}{\partial x_j} H^j_{xx} \\
\dot{H}^i_{x\theta} &= G_x^T F^i_{x\theta} + G_x^T F^i_{xx} G_\theta + \sum_{j=1}^{n_x} \frac{\partial f_i}{\partial x_j} H^j_{x\theta} \\
\dot{H}^i_{\theta x} &= F^i_{\theta x} G_x + G_\theta^T F^i_{xx} G_x + \sum_{j=1}^{n_x} \frac{\partial f_i}{\partial x_j} H^j_{\theta x} \\
\dot{H}^i_{\theta\theta} &= F^i_{\theta\theta} + F^i_{\theta x} G_\theta + G_\theta^T F^i_{x\theta} + G_\theta^T F^i_{xx} G_\theta + \sum_{j=1}^{n_x} \frac{\partial f_i}{\partial x_j} H^j_{\theta\theta}
\end{aligned} \tag{14}$$

Their initial conditions are given by zero matrices with appropriate dimensions. As $\bar{\mathbf{x}}$ and $\bar{\boldsymbol{\theta}}$ are unknown, the remainder term (12) has to be upper bounded. Consequently, cost function (11) is modified so as to take this additional term into account and Theorem 3.2 can be applied again.

$$\begin{aligned}
\check{J}_{n,N}\left(\hat{\mathbf{x}}_{n-N,n}, \hat{\boldsymbol{\theta}}\right) &= \left\|\hat{\mathbf{x}}_{n-N,n} - \bar{\mathbf{x}}^\circ_{n-N}\right\|^2_M + \left\| \left(\bar{G}^n_{x,n-N} + \bar{G}^n_{xx,n-N}\right)\left(\hat{\mathbf{x}}_{n-N,n} - \bar{\mathbf{x}}^\circ_{n-N}\right) \right. \\
&\quad \left. - \left(Y^n_{n-N} - \bar{\mathbf{g}}^n_{n-N} - \left(\bar{G}^n_{\theta,n-N} + \bar{G}^n_{\theta\theta,n-N}\right) S\Delta\boldsymbol{\theta}_{max}\right)\right\|^2_W
\end{aligned} \tag{15}$$

$$\begin{aligned}
\bar{G}^n_{xx,n-N} &= [|CR^+_x(t_{n-N})|^T, \ldots, |CR^+_x(t_n)|^T]^T \quad R^+_x(t) = [R^{1,+,T}_x(t), \ldots, R^{n_x,+,T}_x(t)]^T \\
\bar{G}^n_{\theta\theta,n-N} &= [|CR^+_\theta(t_{n-N})|^T, \ldots, |CR^+_\theta(t_n)|^T]^T \quad R^+_\theta(t) = [R^{1,+,T}_\theta(t), \ldots, R^{n_x,+,T}_\theta(t)]^T
\end{aligned}$$

$R^{j,+}_x$ and $R^{j,+}_\theta$ are worst-case terms related to R^j_x and R^j_θ deduced from (12)-(13):

$$R^j_x(t) = \tfrac{1}{2}\left(\boldsymbol{\Delta}^T_{x_0} H^j_{xx}(t) + \boldsymbol{\Delta}^T_\theta H^j_{\theta x}(t)\right) \quad R^j_\theta(t) = \tfrac{1}{2}\left(\boldsymbol{\Delta}^T_{x_0} H^j_{x\theta}(t) + \boldsymbol{\Delta}^T_\theta H^j_{\theta\theta}(t)\right)$$

Then, $R^{j,+}_x$ and $R^{j,+}_\theta$ are given by

$$\begin{aligned}
R^{j,+}_x(t) &= \tfrac{1}{2}\left(\boldsymbol{\Delta}^{+,T}_{x_0} H^{+,j}_{xx}(t) + \Delta\boldsymbol{\theta}^T_{max} H^{+,j}_{\theta x}(t)\right) \\
R^{j,+}_\theta(t) &= \tfrac{1}{2}\left(\boldsymbol{\Delta}^{+,T}_{x_0} H^{+,j}_{x\theta}(t) + \Delta\boldsymbol{\theta}^T_{max} H^{+,j}_{\theta\theta}(t)\right)
\end{aligned}$$

where $\boldsymbol{\Delta}^{+,T}_{x_0} = \left|\hat{\mathbf{x}}^+_{n-N,n} - \bar{\mathbf{x}}^\circ_{n-N}\right| \geq \left|\hat{\mathbf{x}}_{n-N,n} - \bar{\mathbf{x}}^\circ_{n-N}\right|$. $\hat{\mathbf{x}}^+_{n-N,n}$ matches to the maximum distance with respect to $\bar{\mathbf{x}}^\circ_{n-N}$.

Remark 4.1. *$\hat{\boldsymbol{x}}^+_{n-N,n}$ is not a priori known and is derived by applying Theorem 3.2 twice. At first, cost function (11) is minimized using an initial estimation of $\hat{\boldsymbol{x}}^+_{n-N,n}$. Then, cost function (15) is minimized starting from the previous results. This procedure can be iterated.*

$H_{xx}^{+,j}(t)$, $H_{\theta x}^{+,j}(t)$, $H_{x\theta}^{+,j}(t)$ and $H_{\theta\theta}^{+,j}(t)$ are computed from (14). $\{\bar{\mathbf{x}}, \bar{\boldsymbol{\theta}}\}$ corresponds to the worst-case combination of $\hat{\mathbf{x}}_{n-N,n}^{+}$, $\bar{\mathbf{x}}_{n-N}^{\circ}$, $\boldsymbol{\theta}^{-}$ and $\boldsymbol{\theta}^{+}$.

5 Continuous Cultures of Phytoplankton

The application under consideration is the estimation of biological variables describing the behaviour of the green algae *Dunaliella tertiolecta* in the chemostat. Three state variables are considered: the biovolume X which is the total amount of biomass per unit of volume ($\mu m^3/L$), the internal nitrogen quota Q which is defined as the quantity of nitrogen per unit of biomass ($\mu mol/\mu m^3$) and the substrate (nitrate) concentration S ($\mu mol/L$). Droop model is used to predict the temporal evolution of the three above-mentioned state variables.

$$\begin{cases} \dot{X}(t) = -D(t)X(t) + \mu(Q)X(t) \\ \dot{Q}(t) = \rho(S) - \mu(Q)Q(t) \\ \dot{S}(t) = D(t)(S_{in}(t) - S(t)) - \rho(S)X(t) \end{cases} \tag{16}$$

with $\rho(S) = \rho_m \frac{S(t)}{S(t)+k_S}$ the uptake rate ($\mu mol/(\mu m^3 d)$), and $\mu(Q) = \bar{\mu}\left(1 - \frac{k_Q}{Q(t)}\right)$, the growth rate ($1/d$). In these equations, D represents the dilution rate (1/d, d : day), S_{in} the input substrate concentration ($\mu mol/L$). Moreover, to complete the model description, an observation equation is defined by measurements of biovolume X at measurement time t_k: $y_k = X(t_k)$ (see [3] for further details).

Concerning the simulation data, the reference model parameters are the following : $\rho_m = 9.3 \times 10^{-9}$ $\mu mol \mu m^{-3} d^{-1}$, $k_S = 0.105$ $\mu mol L^{-1}$, $\bar{\mu} = 2$ d^{-1} and $k_Q = 1.8 \times 10^{-9}$ $\mu mol \mu m^{-3}$. The uncertain parameter subspace $[\boldsymbol{\theta}^{-}, \boldsymbol{\theta}^{+}]$ is given by $[0.8\boldsymbol{\theta}, 1.5\boldsymbol{\theta}]$. The input parameters (S_{in}, D) are assumed perfectly known. The initial estimation $\bar{\mathbf{x}}_0$ is randomly chosen in the interval $[0.7\mathbf{x}_0, 3\mathbf{x}_0]$ with $\mathbf{x}_0$, the reference initial conditions given by : $X_0 = 0.1 \times 10^9$ $\mu m^3/L$, $Q_0 = 4.5 \times 10^{-9}$ $\mu mol/\mu m^3$ and $S_0 = 50$ $\mu mol/L$. The measurements of biovolume y_k are corrupted by a Gaussian white noise (with a relative standard deviation of 0.08 and a minimum s.d. of 1.5×10^9 $\mu m^3/L$). Figure 1 (up) shows some estimation results with the linearized robust receding-horizon algorithm (Section 3) and the second-order algorithm (Section 4). The comparison shows better properties for the second-order version which is less sensitive than the linear approximation. However, the computation time is heavier ($\sim 3\times$) mainly due to the iterative procedure (Remark 4.1) but is still quite competitive as compared to the original nonlinear problem.

Concerning the experimental data, parameters are identified and their corresponding intervals are (units are the same as previously): $\rho_m \in [9.25, 9.55]$, $k_S \in [0.01, 0.20]$, $\bar{\mu} \in [1.70, 2.30]$ and $k_Q \in [1.60, 2.00]$. The initial intervals are : $X_0 \in [10^{-6}, 0.20] \times 10^9 \mu m^3/L$, $Q_0 \in [1, 8] \times 10^{-9}$ $\mu mol/\mu m^3$ and $S_0 \in [40, 60] \mu mol/L$ (see [8] for further details). Figure 1 (bottom) shows estimation with experimental biovolume measurements using the second-order algorithm. An interval observer [8] is used for comparison and shows that the estimation is in agreement with the available knowledge on the model.

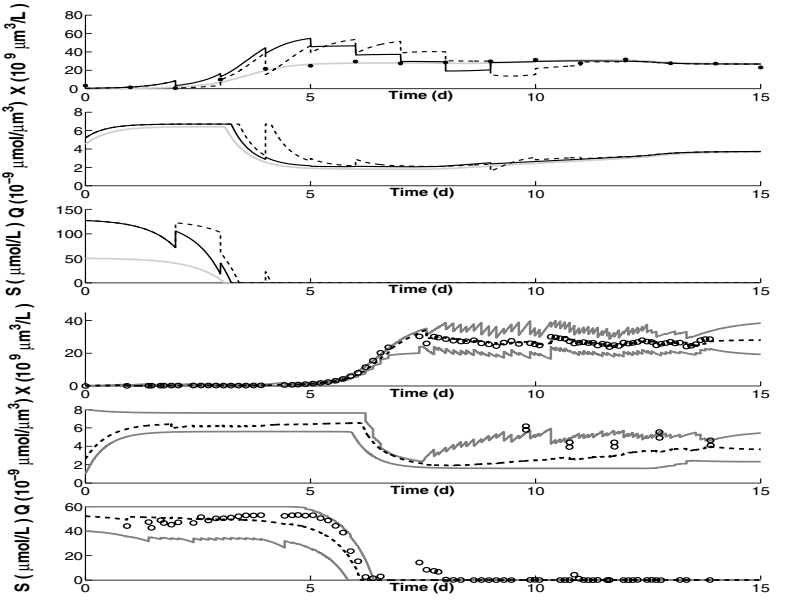


Fig. 1 Up : comparison of first (dashed) and second (solid black line) - order versions in a simulated example ($N = 5$ and $M = 0.1I$). Real trajectories are in grey. Bottom : second-order version ($N = 5$ and $M = 10I$) with experimental data. Biovolume is the only available measurement to the observer and the other measurements are used for cross-validation (circles). The results are enclosed by an interval observer [8]

6 Conclusions

Tests in simulation and real life applications show good performance of robust state estimation methods with respect to model uncertainties. Approximations (first and second order) allow the computational load to be significantly reduced. Tuning is easy because it is only related to the length of the time window and to the weighting matrix of the a priori initial estimation.

Acknowledgements. This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors.

References

1. Alamir, M., Corriou, J.P.: Nonlinear receding-horizon state estimation for dispersive adsorption columns with nonlinear isotherm. Journal of process control 13(6), 517–523 (2003)
2. Alessandri, A., Baglietto, M., Battistelli, G.: Robust receding-horizon state estimation for uncertain discrete-time linear systems. Systems & Control Letters 54, 627–643 (2005)

3. Bernard, O., Sallet, G., Sciandra, A.: Nonlinear observers for a class of biological systems. Application to validation of a phytoplanktonic growth model. IEEE Transaction on Automatic Control 43, 1056–1065 (1998)
4. Bogaerts, P., Hanus, R.: Nonlinear and linearized full horizon state observers - application to bioprocesses. In: Proceedings of the IFAC Symposium on Advanced Control Of Chemical Processes (2000)
5. Bogaerts, P., Hanus, R.: On-line state estimation of bioprocesses with full horizon observers. Mathematics and Computers in Simulation 56, 425–441 (2001)
6. Droop, M.: Vitamin B12 and marine ecology IV: The kinetics of uptake growth and inhibition in monochrisis lutheri. Journal of the Marine Biological Association 48(3), 689–733 (1968)
7. Goffaux, G., Vande Wouwer, A.: Design of a robust nonlinear receding-horizon observer - application to a biological system. In: Proceeding of the 17th IFAC World Congress, Seoul (2008)
8. Goffaux, G., Vande Wouwer, A., Bernard, O.: Continuous - Discrete Interval Observers for Monitoring Microalgae Cultures. Biotechnology Progress (to appear, 2009)
9. Sayed, A.H., Nascimento, V.H., Cipparrone, F.A.M.: A regularized robust design criterion for uncertain data. SIAM Journal Matrix Analysis and Application 23(4), 1120–1142 (2002)

State Estimation in Nonlinear Model Predictive Control, Unscented Kalman Filter Advantages

Giancarlo Marafioti, Sorin Olaru, and Morten Hovd

Abstract. Model predictive control (NMPC) proves to be a suitable technique for controlling nonlinear systems, moreover the simplicity of including constraints in its formulation makes it very attractive for a large class of applications. Due to heavy online computational requirements, NMPC has traditionally been applied mostly to systems with slow dynamics. Recent developments is likely to expand NMPCs applicability to systems with faster dynamics.

Most NMPC formulations are based on the availability of the present state. However, in many applications not all states are directly measurable. Although there is no Separation Theorem which is generally applicable for nonlinear systems, the control and state estimation problems are usually handled separately.

State estimation introduces extra computational load which can be relevant in case of systems with relatively fast dynamics. In this case accurate estimation methods with low computational cost are desired, for example the Extended Kalman Filter (EKF). Clearly, the EKF does not perform well with all nonlinear systems, but its straightforwardness is the main reason of its popularity.

In this work, a type of locally weakly unobservable system is studied. For this type of system, we find that the EKF drifts because the system is unobservable at the desired operation point. The Unscented Kalman Filter (UKF) is described, and similarities with the EKF are discussed. Finally, it is shown how the UKF is used for state estimation in this type of nonlinear systems, and that it provides a stable state estimate, despite the fact that the system is locally unobservable.

Keywords: state estimation, nonlinear model predictive control, unscented Kalman filter.

Giancarlo Marafioti and Morten Hovd
Department of Engineering Cybernetics, Norwegian University of Science and Technology, N-7491 Trondheim, Norway
e-mail: giancarlo.marafioti@itk.ntnu.no, morten.hovd@itk.ntnu.no

Sorin Olaru
Automatic Control Department, SUPELEC, Gif sur Yvette, 91192, France
e-mail: sorin.olaru@supelec.fr

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 305–313.
springerlink.com

1 Introduction

Model Predictive Control (MPC) is one of the most attractive advanced control technique. The principal reasons of its success are the capability to directly handle the control problem in the time domain, and the simplicity of considering physical constraints. Essentially, this is done by using a model for predicting the future process output. An objective function is optimized, yielding an input sequence which is applied to the process using a receding horizon strategy.

The development of NMPC is not straightforward, remarkable complications are required to verify nominal stability and optimization problem solution. A comprehensive analysis of linear MPC can be found in the literature, (e.g. [1]). However, most real world systems are nonlinear, and a continuous search for better performance shows the limitations of linear MPC. Thus, using a nonlinear model in the MPC framework seems to be a practical solution. An introduction to nonlinear model predictive control can be found in [2], and more details on stability are in [3].

A detailed survey of industrial MPC applications can be found in [4]. A limitation for the application of NMPC is that at every time step the state of the system is needed for prediction. However, it is not always possible to measure all the states, thus a filter or observer may be used. For nonlinear systems a Separation Theorem does not exist, then even if the state feedback controller and the observer are both stable, there is no guarantee of closed loop nominal stability. Findeisen et al. [5] give an interesting overview on both state and output feedback NMPC.

In Section 2 the nonlinear model predictive control formulation adopted in this work is described. More precisely, an output feedback NMPC is used where both the EKF and the UKF are implemented for state estimation. The EKF is a well known filter, it is an extension of the Kalman Filter [6] to nonlinear systems. In Section 3 the UKF is introduced as a valid alternative to the EKF. In Section 4 a locally weakly unobservable system is described and used to show, by simulations, how the UKF is able to give a stable state estimate where the EKF fails.

2 Nonlinear Model Predictive Control

Consider the discrete nonlinear state space model

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) \\ y_k &= h(x_k, u_{k-1}) \end{aligned} \tag{1}$$

where x_k is the state vector, u_k is the control input vector, y_k is the measurement vector. The subscript k represents the sampling time index.

The NMPC optimization problem for regulation can be stated as follows:

$$\min_{u_0, u_1, \ldots, u_{np-1}} \sum_{k=0}^{n_p-1} \mathrm{G}(x_{k+1}, u_k) \tag{2}$$

subject to

$$x_0 = \text{given} \tag{3a}$$
$$x_{k+1} - f(x_k, u_k) = 0 \tag{3b}$$
$$c_x(x_k) \leq 0 \tag{3c}$$
$$c_u(u_k) \leq 0 \tag{3d}$$

The optimization problem is solved at each sample instant, obtaining the optimal solution $[u_0^*, u_1^*, ..., u_{n_p-1}^*]$. However, only the first element, u_0^*, of this sequence is applied, before the optimization problem is solved again at the next timestep.

The stage cost $\mathrm{G}(\cdot)$ is typically quadratic, that is $\mathrm{G}(x,u) = x'Qx + u'Ru$, where $Q = Q' \succeq 0$ and $R = R' \succ 0$. Equations (3c) and (3d) can be either linear or nonlinear, and are used to introduce state and input constraints, respectively.

The nonlinear model (1) and the inequality constraints (3c), (3d) make the problem (2-3) a non convex nonlinear program. The Sequential Quadratic Programming technique (SQP) may be used to solve it, but compared to the convex QP case obtained in linear MPC, the solution is more difficult. For instance, there is no guarantee of the existence of unique optimal solution and that the global optimum can always be found. The prediction horizon length n_p affects the computational complexity, and influences the global stability at the same time. However, since this paper is not intended as a comprehensive analysis on NMPC, the reader is referred to the literature, some of which can be found in the references. Thus, in the next section we focus on the state estimation problem and in particularly on the Unscented Kalman Filter.

3 State Estimation: Unscented Kalman Filter

From a wide point of view, the state estimation problem can be seen as a probabilistic inference problem, which is the problem of estimating hidden variables (state) using a noisy set of observations (measurement), in an optimal way. In probability theory, this solution is obtained using the recursive Bayesian estimation algorithm. For linear Gaussian systems, it can be shown that the closed form of the optimal recursive solution is given by the well known Kalman Filter.

For real world applications, which are neither linear nor Gaussian, the Bayesian approach is intractable. Thus, a sub-optimal approximated solution has to be found. For instance, in case of nonlinear system with non Gaussian probability density function (pdf), the EKF algorithm approximates the nonlinear system with its truncated Taylor series expansion around the current estimate. Moreover, the non Gaussian pdf is approximated with its first two moments, mean and covariance. A more elaborate approach is to use a Sequential Monte-Carlo method, where instead of approximating the system or the probability distribution, the integrals in the Bayesian solution are approximated with finite sums. One famous Monte-Carlo filter is the Particle Filter, but in this work we focus on Kalman filtering for two simple reasons. Firstly, Kalman algorithms are easy to understand and to implement. Secondly, they do not have a high computational complexity, making them suitable to be coupled

with a NMPC. This will allow the controller to obtain more resource which can be used to solve the optimization problem.

A more recent approach than the EKF, to the state estimation problem, is the Unscented Kalman Filter. The UKF uses the Unscented Transformation (UT), which is based on the idea that is easier to approximate a probability distribution than an arbitrary nonlinear function or transformation [7]. This approximation is done using a finite set of points, called sigma points.

An important feature of the UKF, with respect to the EKF, is that no Jacobians need to be computed. This is relevant especially in the case of strong nonlinearities, as long as the introduction of linearization errors is avoided. In general both filters have the same computational complexity [8], and the UKF implementation is straightforward, similarly to the EKF one.

In the present work only the UKF formulation is detailed, more precisely the one used in [9], while the EKF can be easily found in the vast literature. Moreover, it is assumed that the process and the measurement disturbances are modeled as additive noise with zero means. As it can be seen in the next section, the UKF algorithm has a structure similar to that of the EKF. There is an initialization part (7-8), and a set of equations executed recursively: the time update equations (10-14) and the measurement update equations (15-19). In addition, at every time step the sigma points have to be computed by (9).

3.1 UKF Formulation

Before describing the particular UKF implementation used in the present framework, we introduce some notations. $\mathrm{E}[\cdot]$ is the expectation operator, $\hat{x}_k = \mathrm{E}[x_k]$ is the mean value of x_k, $\mathrm{P_k} = \mathrm{E}[(x_k - \hat{x}_k)(x_k - \hat{x}_k)']$ is the error covariance matrix. $\mathrm{P^W}$ and $\mathrm{P^V}$ are the process noise covariance and the measurement noise covariance, respectively. $\mathrm{Ch_k} = \mathrm{chol}(P_k)$ is the Cholesky factorization of P_k, L is the system state dimension, $\mathrm{F}[\cdot]$ and $\mathrm{H}[\cdot]$ are the nonlinear dynamics and the measurement equation, respectively. W_i are scalar weights calculated as follow

$$W_0^{(m)} = \lambda/(L+\lambda) \tag{4}$$

$$W_0^{(c)} = \lambda/(L+\lambda) + (1-\alpha^2+\beta) \tag{5}$$

$$W_i^{(m)} = W_i^{(c)} = 1/[2(L+\lambda)], \quad i = 1,\ldots 2L \tag{6}$$

where $\lambda = \alpha^2(L+\kappa) - L$, and α, β, κ are parameters to be chosen as described in [9]. Finally we define $\gamma = \sqrt{L+\lambda}$, which can be interpreted as a scaling factor used to move the position of sigma points around the mean value $\hat{x}$ (Equation (9)). Indeed, there are three independent design parameters that have to be chosen (α, β, κ). In most cases typical values are $\beta = 2$, and $\kappa = 0$ or $\kappa = 3 - L$, leaving only the parameter α as design parameter. Moreover, considering that $1 \leq \alpha \leq 1e^{-5}$ the

tuning of the UKF becomes simpler. For a finer tuning and a better description of the UKF parameters we remand to [9]. As it is the case for the EKF, the covariance matrices can be also used for performance tuning.

We state here the main steps of the UKF construction which starts with a given

$$\hat{x}_0 = \mathrm{E}[x_0] \tag{7}$$
$$P_0 = \mathrm{E}[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)'] \tag{8}$$

Then for each sample $k = 1, \ldots, \infty$

- calculate the sigma points, represented as column vectors in the following matrix of dimension $2L+1$:

$$\mathscr{X}_{k-1} = \left[\hat{x}_{k-1}, \quad \hat{x}_{k-1} + \gamma \mathrm{Ch}_{k-1}, \quad \hat{x}_{k-1} - \gamma \mathrm{Ch}_{k-1}\right] \tag{9}$$

- propagate the sigma points through the nonlinear dynamics $\mathrm{F}[\cdot]$, and compute the predicted state estimate, where the index i is used to select the appropriate sigma point column:

$$\mathscr{X}_{k|k-1} = \mathrm{F}\left[\mathscr{X}_{k-1}, u_{k-1}\right] \tag{10}$$
$$\hat{x}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \mathscr{X}_{i,k|k-1} \tag{11}$$

- compute the predicted covariance, instantiate the prediction points through the observation mapping $\mathrm{H}[\cdot]$, and calculate the predicted measurement:

$$P_k^- = \sum_{i=0}^{2L} W_i^{(c)} \left[\mathscr{X}_{i,k|k-1} - \hat{x}_k^-\right] \left[\mathscr{X}_{i,k|k-1} - \hat{x}_k^-\right]' + \mathrm{P}^{\mathrm{w}} \tag{12}$$

$$\mathscr{Y}_{k|k-1} = \mathrm{H}\left[\mathscr{X}_{k|k-1}\right] \tag{13}$$
$$\hat{y}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \mathscr{Y}_{i,k|k-1} \tag{14}$$

- obtain the innovation covariance and the cross covariance matrices:

$$P_{\tilde{y}_k \tilde{y}_k} = \sum_{i=0}^{2L} W_i^{(c)} \left[\mathscr{Y}_{i,k|k-1} - \hat{y}_k^-\right] \left[\mathscr{Y}_{i,k|k-1} - \hat{y}_k^-\right]' + \mathrm{P}^{\mathrm{v}} \tag{15}$$

$$P_{y_k x_k} = \sum_{i=0}^{2L} W_i^{(c)} \left[\mathscr{X}_{i,k|k-1} - \hat{x}_k^-\right] \left[\mathscr{Y}_{i,k|k-1} - \hat{y}_k^-\right]' \tag{16}$$

- perform the measurement update using the normal Kalman filter equations:

$$\mathscr{K}_k = P_{y_k x_k} P_{\tilde{y}_k \tilde{y}_k}^{-1} \tag{17}$$

$$\hat{x}_k = \hat{x}_k^- + \mathscr{K}_k \left(y_k - \hat{y}_k^- \right) \tag{18}$$

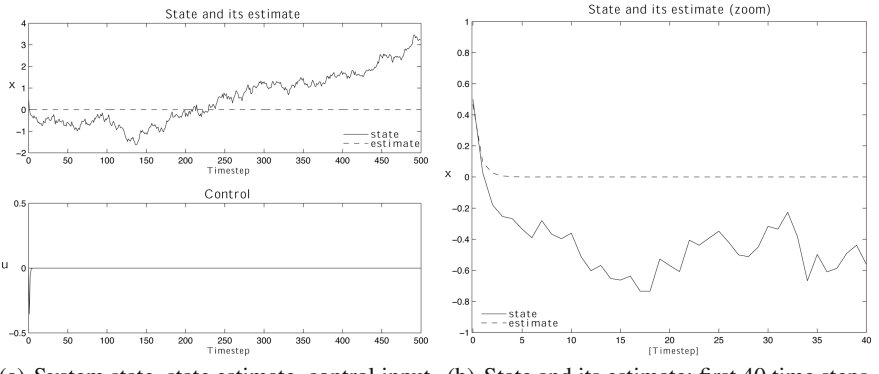
$$P_k = P_k^- - \mathscr{K}_k P_{\tilde{y}_k \tilde{y}_k} \mathscr{K}_k' \tag{19}$$

Note that no linearization procedure is required, and the most time demanding operation is the Cholesky factorization. The algorithm implementation is straightforward, since only simple operations need to be performed, e.g. weighted sums. There are several UKF formulations available in literature, see for instance [9], however the one chosen in this work has the benefit of a reduced number of sigma points. Since both process and measurement noise realizations are assumed additive, there is no need to associate sigma points to approximate their statistics. However, their effect is considered in (12) for the process noise, and in (15) for the measurement noise.

To understand how the UT is used in this framework let us consider (9). The mean $\hat{x}_{k-1}$ and covariance P_{k-1} are approximated with a set of $2L+1$ points (columns of (9)). The nonlinear function (10) is directly applied to each sigma point, yielding a cloud of transformed points. The statistics of the transformed points are then computed, by (11) and (12), to obtain an estimate of the transformed mean and covariance. This is done, in a similar way by (13-16), for the nonlinear function associated to the observation model .

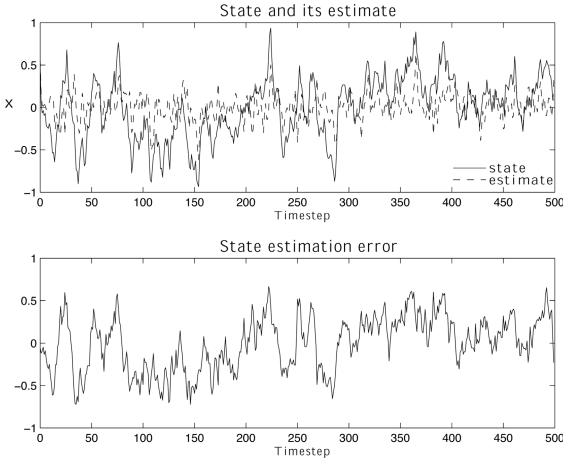
4 A Locally Weakly Unobservable System

It is well known, that the fundamental requirement for an observer to work properly is to be associated to an observable system. While for linear system this condition is trivial to check, using the Kalman rank condition of the observability matrix, for nonlinear systems things are more complicated. The principal motivation is that

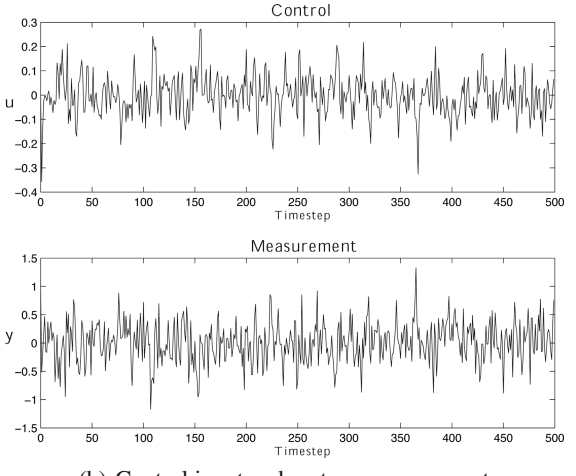


(a) System state, state estimate, control input (b) State and its estimate: first 40 time steps

Fig. 1 Simulation based results: EKF as state estimator



(a) System state and its estimate, state estimation error



(b) Control input and system measurement

Fig. 2 Simulation based results: UKF as state estimator

in general the observability of a nonlinear system depends on the system input itself. In [10] the authors show how to extend the Kalman rank condition for both controllability and observability of nonlinear systems using a differential-geometric approach. In [11] a viewpoint on observability and nonlinear observer design is given. In both works the Lie algebra is used, and for our purpose it is sufficient to know that a nonlinear system that does not satisfied the observability rank condition derived in [10] it is not locally weakly observable, meaning that the estimator may have problems once reaches an unobservability region.

As an illustrative example we consider the system

$$\begin{aligned} x_{k+1} &= x_k + (0.1x_k + 1)u_k + w_k \\ y_k &= x_k^3 + v_k \end{aligned} \tag{20}$$

where x is the state, u is the input, y is the measurement, w is the excitation state noise, v is the measurement noise, and the subscript k is the sampling time index. The noise sequences are both assumed to be Gaussian white noise.

System (20) is a locally weakly unobservable system, in the sense that the rank condition fails for the point $(\bar{x}, \bar{u}) = (0, 0)$. Simulation based results will show that while the EKF fails to estimate the state, the UKF is able to give a stable state estimate. This is due to the fact that the UKF does not use any linearization and in addition the pdf approximation obtained with the sigma points is more accurate than the EKF one.

Simulation based results: In the simulations a quadratic NMPC cost function is used, where $Q = 2$ and $R = 1$. The prediction horizon length is $n_p = 10$ and a sampling time of 1 second is used. The process and measurement noises have additive white Gaussian distributions with zero means and variances $\sigma_\omega^2 = 0.01$, $\sigma_v^2 = 0.1$, respectively. All the simulations are started from an initial condition $x_0 = 0.5$, with initial state variance $P_0 = 0.02$. The UKF tuning parameters are $L = 1$, $\alpha = 1$, $\beta = 2$, and $\kappa = 2$.

Figure 1 shows how the EKF fails. It is possible to notice how once both state and its estimate reach zero, the estimator is not able anymore to reconstruct the actual state (Figure 1(b)). This is due to the observability problem, in fact at zero the Kalman gain becomes zero and the filter is not able anymore to correct the estimate with the future measurements. The controller uses the estimate, yielding a zero control signal as soon as the state estimate is null. As a result the controller is not able to regulate the state anymore and the state itself drifts.

In Figure 2 the UKF is used instead. It is possible to observe how the filter is able to estimate the state, and in the meantime there is no drift in the state estimation. By consequence, the NMPC law can regulate the state, achieving its goal.

5 Conclusions

In this work an output feedback Nonlinear Model Predictive Control was implemented for a regulation problem of a locally weakly unobservable system. For this kind of systems the Extended Kalman Filter may fail to give a correct state estimate, leading to a drift in the state estimation. The Unscented Kalman Filter was introduced as an alternative estimation algorithm, and its main ingredients were presented. Using a simple but effective example, a set of simulations were carried out to show how the UKF gives a stable state estimate, even in presence of unobservability regions. Other estimation algorithms could be used, for instance Monte-Carlo based methods, but the choice of a Kalman based filter gives the advantage of a lower overall computational complexity.

References

1. Maciejowski, J.: Predictive control with constraints. Prentice-Hall, Englewood Cliffs (2002)
2. Findeisen, R., Allgöwer, F.: An introduction to nonlinear model predictive control. In: 21st Benelux Meeting on Systems and Control (2002)
3. Mayne, D., Rawlings, J., Rao, C., Scokaert, P.: Constrained model predictive control: Stability and optimality. Automatica 36(36), 789–814 (2000)
4. Qin, S., Badgwell, T.: A survey of industrial model predictive control technology. Control engineering practice 11, 733–764 (2003)
5. Findeisen, R., Imsland, L., Allgöwer, F., Foss, B.A.: State and output feedback nonlinear model predictive control: An overwiev. European Journal of Control 9(2-3), 921–936 (2003)
6. Kalman, R.E.: A new approach to linear filtering and prediction problems. Transactions of the ASME–Journal of Basic Engineering 82(D), 35–45 (1960)
7. Julier, S., Uhlmann, J.: A general method for approximating nonlinear transformations of probability distributions (1996)
8. Wan, E.A., Van Der Merwe, R.: The unscented kalman filter for nonlinear estimation. In: Adaptive Systems for Signal Processing, Communications, and Control Symposium, pp. 153–158 (2000)
9. Wan, E.A., Merwe, R.V.D.: Kalman Filtering and Neural Networks. The Unscented Kalman Filter, ch. 7, 50 pages. Wiley Publishing, Chichester (2001)
10. Hermann, R., Krener, A.: Nonlinear controllability and observability. IEEE Transactions on Automatic Control 22(5), 728–740 (1977)
11. Besançon, G.: A viewpoint on observability and observer design for nonlinear systems. New Directions in nonlinear observer design, 3–22 (1999)

MPC for Tracking of Constrained Nonlinear Systems

D. Limon, A. Ferramosca, I. Alvarado, T. Alamo, and E.F. Camacho

Abstract. This paper deals with the tracking problem for constrained nonlinear systems using a model predictive control (MPC) law. MPC provides a control law suitable for regulating constrained linear and nonlinear systems to a given target steady state. However, when the target operating point changes, the feasibility of the controller may be lost and the controller fails to track the reference. In this paper, a novel MPC for tracking changing constant references is presented. This controller extend a recently presented MPC for tracking for constrained linear systems to the nonlinear case. The main characteristics of this controller are: considering an artificial steady state as a decision variable, minimizing a cost that penalizes the error with the artificial steady state, adding to the cost function an additional term that penalizes the deviation between the artificial steady state and the target steady state (the so-called *offset cost function*) and considering an invariant set for tracking as extended terminal constraint. The properties of this controller has been tested on a constrained CSTR simulation model.

Keywords: Nonlinear systems; Nonlinear Model Predictive Control; Tracking.

1 Introduction

Model predictive control (MPC) is one of the most successful techniques of advanced control in the process industry. This is due to its control problem formulation, the natural usage of the model to predict the expected evolution of the plant, the optimal character of the solution and the explicit consideration of hard constraints in the optimization problem. Thanks to the recent developments of the underlying theoretical framework, MPC has become a mature control technique capable to

Departamento de Ingeniería de Sistemas y Automática.
Universidad de Sevilla, Escuela Superior de Ingenieros.
Camino de los Descubrimientos s/n. 41092 Sevilla, Spain
e-mail: {limon,ferramosca,alvarado,alamo,eduardo}@cartuja.us.es

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 315–323.
springerlink.com

provide controllers ensuring stability, robustness, constraint satisfaction and tractable computation for linear and for nonlinear systems [1].

The control law is calculated by predicting the evolution of the system and computing the admissible sequence of control inputs which makes the system evolves satisfying the constraints and minimizing the predicted cost. This problem can be posed as an optimization problem. To obtain a feedback policy, the obtained sequence of control inputs is applied in a receding horizon manner, solving the optimization problem at each sample time. Considering a suitable penalization of the terminal state and an additional terminal constraint, asymptotic stability and constraints satisfaction of the closed loop system can be proved [2].

Most of the results on MPC consider the regulation problem, that is steering the system to a fixed steady-state (typically the origin), but when the target operating point changes, the feasibility of the controller may be lost and the controller fails to track the reference [3, 4, 5, 6]. Tracking control of constrained nonlinear systems is an interesting problem due to the nonlinear nature of many processes in industry mainly when large transitions are required, as in the case of changing operating point.

In [7] a nonlinear predictive control for set point families is presented, which considers a pseudolinearization of the system and a parametrization of the set points. The stability is ensured thanks to a quasi-infinite nonlinear MPC strategy, but the solution of the tracking problem is not considered.

In [8] an output feedback receding horizon control algorithm for nonlinear discrete-time systems is presented, which solves the problem of tracking exogenous signals and asymptotically rejecting disturbances generated by a properly defined exosystem. In [9] an MPC algorithm for nonlinear systems is proposed, which guarantees local stability and asymptotic tracking of constant references. This algorithm need the presence of an integrator preliminarily plugged in front of the system to guarantee the solution of the asymptotic tracking problem.

Another approach to the tracking of nonlinear systems problem are the so-called reference governors [10, 4, 11]. A reference governor is a nonlinear device which manipulates on-line a command input to a suitable pre-compensated system so as to satisfy constraints. This can be seen as adding an artificial reference, computed at each sampling time to ensure the admissible evolution of the system, converging to the desired reference.

In [12] the tracking problem for constrained linear systems is solved by means of an approach called dual mode: the dual mode controller operates as a regulator in a neighborhood of the desired equilibrium wherein constraints are feasible, while it switches to a feasibility recovery mode, whenever this is lost due to a set point change, which steers the system to the feasibility region of the MPC as quickly as possible. In [13] this approach is extended to nonlinear systems, considering constraint-admissible invariant sets as terminal regions, obtained by means of a LPV model representation of the nonlinear plant.

In [14] an MPC for tracking of constrained linear systems is proposed, which is able to lead the system to any admissible set point in an admissible way. The main characteristics of this controller are: an artificial steady state is considered as a decision variable, a cost that penalizes the error with the artificial steady state

is minimized, an additional term that penalizes the deviation between the artificial steady state and the target steady state is added to the cost function (the so-called *offset cost function*) and an invariant set for tracking is considered as extended terminal constraint. This controller ensures that under any change of the target steady state, the closed loop system maintains the feasibility of the controller and ensures the convergence to the target if admissible. In this paper, this controller is extended to the case of nonlinear constrained systems.

The paper is organized as follows. In section 2 the constrained tracking problem is stated. In section 3 the new MPC for tracking is presented. In section 4 an illustrative example is shown. Finally, in section 5 some conclusions are drawn.

2 Problem Description

Consider a system described by a nonlinear invariant discrete time model

$$\begin{aligned} x^+ &= f(x,u) \\ y &= h(x,u) \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ is the system state, $u \in \mathbb{R}^m$ is the current control vector, $y \in \mathbb{R}^p$ is the controlled output and x^+ is the successor state. The function model $f(x,u)$ is assumed to be continuous. The solution of this system for a given sequence of control inputs $\mathbf{u}$ and initial state x is denoted as $x(j) = \phi(j,x,\mathbf{u})$ where $x = \phi(0,x,\mathbf{u})$. The state of the system and the control input applied at sampling time k are denoted as $x(k)$ and $u(k)$ respectively. The system is subject to hard constraints on state and control:

$$x(k) \in X, \quad u(k) \in U \tag{2}$$

for all $k \geq 0$. $X \subset \mathbb{R}^n$ and $U \subset \mathbb{R}^m$ are compact convex polyhedra containing the origin in its interior.

The steady state, input and output of the plant (x_s, u_s, y_s) are such that (1) is fulfilled, i.e. $x_s = f(x_s, u_s)$ and $y_s = h(x_s, u_s)$. Due to the relation derived from these equalities, it is possible to find a parameter vector $\theta \in \mathbb{R}^q$ which univocally defines each triplet (x_s, u_s, y_s), i.e., these can be posed as $x_s = g_x(\theta)$, $u_s = g_u(\theta)$ and $y_s = g_y(\theta)$. This parameter is typically the controlled output y_s although another parameter could be chosen for convenience.

For a (possible time-varying) target steady condition (x_t, u_t, y_t) given by θ_t, the problem we consider is the design of an MPC controller $\kappa_N^{MPC}(x, \theta_t)$ such that the system is steered as close as possible to the target while fulfilling the constraints.

3 MPC for Tracking

In this section, the proposed MPC for tracking is presented. This predictive controller is characterized by the addition of the steady state and input as decision variables, the use of a modified cost function and an extended terminal constraint.

The proposed cost function of the MPC is given by:

$$V_N(x,\theta_t;\mathbf{u},\theta)=\sum_{i=0}^{N-1}\ell((x(i)-x_s),(u(i)-u_s))+V_f(x(N)-x_s,\theta)+V_O(\theta-\theta_t)$$

where $x(j)=\phi(j,x,\mathbf{u})$, $x_s=g_x(\theta)$, $u_s=g_u(\theta)$ and $y_s=g_y(\theta)$. The controller is derived from the solution of the optimization problem $P_N(x,\theta_t)$ given by:

$$\begin{aligned} V_N^*(x,\theta_t) \;=\; & \min_{\mathbf{u},\theta} V_N(x,\theta_t;\mathbf{u},\theta) \\ & s.t.\ x(j)=\phi(j,x,\mathbf{u}), \quad j=0,\cdots,N \\ & \quad x(j)\in X, u(j)\in U, \quad j=0,\cdots,N-1 \\ & \quad x_s=g_x(\theta), u_s=g_u(\theta) \\ & \quad (x(N),\theta)\in\Gamma \end{aligned}$$

Considering the receding horizon policy, the control law is given by

$$\kappa_N^{MPC}(x,\theta_t)=u^*(0;x,\theta_t)$$

Since the set of constraints of $P_N(x,\theta_t)$ does not depend on θ_t, its feasibility region does not depend on the target operating point θ_t. Then there exists a region $\mathscr{X}_N\subseteq X$ such that for all $x\in\mathscr{X}_N$, $P_N(x,\theta_t)$ is feasible. This is the set of initial states that can be admissibly steered to the projection of Γ onto x in N steps or less.

Consider the following assumption on the controller parameters:

Assumption 1

1. *Let g_x, g_u and g_y be the functions of the parameter θ which define the steady state, input and output of system (1) respectively, i.e., $x_s=g_x(\theta)$, $u_s=g_u(\theta)$ and $y_s=g_y(\theta)$. Assume that g_x is a Lipschitz function.*
2. *Let Θ be a given convex set of admissible targets, i.e. $g_x(\theta)\in X$ and $g_u(\theta)\in U$ for all $\theta\in\Theta$. This set defines the set of potential target equilibrium points.*
3. *Let $k(x,\theta)$ be a continuous control law such that for all $\theta\in\Theta$, system $x^+=f(x,k(x,\theta))$ has $x_s=g_x(\theta)$ and $u_s=g_u(\theta)$ as steady state and input, and it is asymptotically stable.*
4. *Let $\Gamma\subset\mathbb{R}^{n+q}$ be a set such that for all $(x,\theta)\in\Gamma$, $x\in X$, $\theta\in\Theta$, $k(x,\theta)\in U$ and $(f(x,k(x,\theta)),\theta)\in\Gamma$.*
5. *Let $V_f(x-g_x(\theta),\theta)$ be a Lyapunov function for system $x^+=f(x,k(x,\theta))$:*

$$V_f(f(x,k(x,\theta))-g_x(\theta),\theta)-V_f(x-g_x(\theta),\theta)\le -l(x-g_x(\theta),k(x,\theta)-g_u(\theta))$$

   for all $(x,\theta)\in\Gamma$. Moreover, there exist $b>0$ and $\sigma>1$ which verify $V_f(x_1-x_2,\theta)\le b\|x_1-x_2\|^\sigma$ for all (x_1,θ) and (x_2,θ) contained in Γ.
6. *Let $l(x,u)$ be a positive definite function and let the offset cost function $V_O:\mathbb{R}^p\to\mathbb{R}$ be a convex, positive definite and subdifferentiable function.*

The following theorem proves asymptotic stability and constraints satisfaction of the controlled system.

Theorem 1 (Stability). *Consider that assumption 1 holds and consider a given target operation point parametrization θ_t, such that $x_t = g_x(\theta_t)$, $u_t = g_u(\theta_t)$ and $y_t = g_y(\theta_t)$. Then for any feasible initial state $x_0 \in \mathscr{X}_N = Proj_x(\Gamma)$, the system controlled by the proposed MPC controller $\kappa(x,\theta_t)$ is stable, fulfils the constraints along the time and, besides*

(i) If $\theta_t \in \Theta$ then the closed loop system asymptotically converges to a steady state, input and output (x_t, u_t, y_t), that means $\lim_{k\to\infty} \|x(k) - x_t\| = 0$, $\lim_{k\to\infty} \|u(k) - u_t\| = 0$ and $\lim_{k\to\infty} \|y(k) - y_t\| = 0$.

(ii)If $\theta_t \notin \Theta$, the closed loop system asymptotically converges to a steady state and input $(\tilde{x}_s, \tilde{u}_s)$, such that $\lim_{k\to\infty} \|x(k) - \tilde{x}_s\| = 0$ and $\lim_{k\to\infty} \|u(k) - \tilde{u}_s\| = 0$, where $\tilde{x}_s = g_x(\tilde{\theta}_s)$, $\tilde{u}_s = g_u(\tilde{\theta}_s)$ and

$$\tilde{\theta}_s = \arg\min_{\theta\in\Theta} V_O(\theta - \theta_t)$$

Proof

Feasibility. The first part of the proof is devoted to prove the feasibility of the controlled system, that is $x(k+1) \in \mathscr{X}_N$, for all $x(k) \in \mathscr{X}_N$ and θ_t. Assume that $x(k)$ is feasible and consider the optimal solution of $P_N(x(k),\theta_t)$, $\mathbf{u}^*(x(k),\theta_t)$, $\theta^*(x(k),\theta_t)$. Define the following sequences:

$$\begin{aligned}\mathbf{u}(x(k+1),\theta_t) &\stackrel{\Delta}{=} [u^*(1;x(k),\theta_t),\cdots,u^*(N-1;x(k),\theta_t),\\ &\qquad K(x^*(N;x(k),\theta_t),\theta^*(x(k),\theta_t))]\\ \bar{\theta}(x(k+1),\theta_t) &\stackrel{\Delta}{=} \theta^*(x(k),\theta_t)\end{aligned}$$

Then, due to the fact that $x(k+1) = f(x(k),u^*(0;x(k),\theta_t))$ and to condition 4 in assumption 1, it is easy to see that $\mathbf{u}(x(k+1),\theta_t)$ and $\bar{\theta}(x(k+1),\theta_t)$ are feasible solutions of $P_N(x(k+1),\theta_t)$. Consequently, $x(k+1) \in \mathscr{X}_N$.

Convergence. Consider the feasible solution at time $k+1$ previously presented. Following standard steps in the stability proofs of MPC [2], we get that

$$V_N^*(x(k+1),\theta_t) - V_N^*(x(k),\theta_t) \le -l(x(k) - g_x(\theta^*(x(k),\theta_t)), u(k) - g_u(\theta^*(x(k),\theta_t)))$$

Due to the definite positiveness of the optimal cost and its non-increasing evolution, we infer that $\lim_{k\to\infty} \|x(k) - g_x(\theta^*(x(k),\theta_t)\| = 0$ and $\lim_{k\to\infty} \|u(k) - g_u(\theta^*(x(k),\theta_t)\| = 0$.

Optimality. Define $x_s^*(x(k),\theta_t) = g_x(\theta^*(x(k),\theta_t)$ and $u_s^*(x(k),\theta_t) = g_u(\theta^*(x(k),\theta_t)$. Let $\tilde{\Theta}$ be the convex set such that $\tilde{\Theta} = \{\tilde{\theta} : \tilde{\theta} = \arg\min_{\theta\in\Theta} V_O(\theta - \theta_t)\}$.

We proceed by contradiction. Consider that $\theta^* \notin \tilde{\Theta}$ and take a $\tilde{\theta} \in \tilde{\Theta}$, then $V_O(\theta^* - \theta_t) > V_O(\tilde{\theta} - \theta_t)$. Due to continuity of the model and the control law, there exists a $\hat{\lambda} \in [0,1)$ such that, for every $\lambda \in [\hat{\lambda}, 1)$, the parameter $\bar{\theta} = \lambda\theta^* + (1-\lambda)\tilde{\theta}$ fulfils $(x_s^*, \bar{\theta}) \in \Gamma$.

Defining as $\mathbf{u}$ the sequence of control actions derived from the control law $k(x, \bar{\theta})$, it is inferred that $(\mathbf{u}, x_s^*, \bar{\theta})$ is a feasible solution for $P_N(x_s^*, \theta_t)$. Then from assumption 1 and using standard procedures in MPC, we have that

$$\begin{aligned} V_N^*(x_s^*, \theta_t) = V_O(\theta^* - \theta_t) &\leq V_N(x_s^*, \theta_t; \mathbf{u}, \bar{\theta}) \\ &= \sum_{i=0}^{N-1} \ell((x(i) - \bar{x}), (k(x(i), \theta) - \bar{u})) \\ &\quad + V_f(x(N) - \bar{x}, \theta) + V_O(\bar{\theta} - \theta_t) \\ &\leq V_f(x_s^* - \bar{x}, \theta) + V_O(\bar{\theta} - \theta_t) \\ &\leq L_{V_f} \|\theta^* - \bar{\theta}\|^\sigma + V_O(\bar{\theta} - \theta_t) \\ &= L_{V_f}(1-\lambda)^\sigma \|\theta^* - \tilde{\theta}\|^\sigma + V_O(\bar{\theta} - \theta_t) \end{aligned}$$

where $L_{V_f} = L_g^\sigma b$ and L_g is the Lipshitz constant of $g_x(\cdot)$.

Define $W(x_s^*, \theta_t, \lambda) \stackrel{\Delta}{=} L_{V_f}(1-\lambda)^\sigma \|\theta^* - \tilde{\theta}\|^\sigma + V_O(\bar{\theta} - \theta_t)$. Notice that $W(x_s^*, \theta_t, \lambda) = V_N^*(x_s^*, \theta_t)$ for $\lambda = 1$. Taking the partial of W about λ, we have that

$$\frac{\partial W}{\partial \lambda} = -L_{V_f}\sigma(1-\lambda)^{\sigma-1}\|\theta^* - \tilde{\theta}\|^\sigma + g^T(\theta^* - \tilde{\theta})$$

where $g^T \in \partial V_O(\bar{\theta} - \theta_t)$, defining $\partial V_O(\bar{\theta} - \theta_t)$ as the subdifferential of $V_O(\bar{\theta} - \theta_t)$. Evaluating this partial for $\lambda = 1$ we obtain that:

$$\left.\frac{\partial W}{\partial \lambda}\right|_{\lambda=1} = g^{*T}(\theta^* - \tilde{\theta})$$

where $g^{*T} \in \partial V_O(\theta^* - \theta_t)$, defining $\partial V_O(\theta^* - \theta_t)$ as the subdifferential of $V_O(\theta^* - \theta_t)$. Taking into account that V_O is a subdifferentiable function, from convexity [15] we can state that

$$g^{*T}(\theta^* - \tilde{\theta}) \geq V_O(\theta^* - \theta_t) - V_O(\tilde{\theta} - \theta_t)$$

Considering that $V_O(\theta^* - \theta_t) - V_O(\tilde{\theta} - \theta_t) > 0$, it can be derived that

$$\left.\frac{\partial W}{\partial \lambda}\right|_{\lambda=1} \geq V_O(\theta^* - \theta_t) - V_O(\tilde{\theta} - \theta_t) > 0$$

This means that there exists a $\lambda \in [\hat{\lambda}, 1)$ such that $W(x_s^*, \theta_t, \lambda)$ is smaller than the value of $W(x_s^*, \theta_t, \lambda)$ for $\lambda = 1$, which equals to $V_N^*(x_s^*, \theta_t)$.

This contradicts the optimality of the solution and hence the result is proved, finishing the proof.

Remark 1. The problem of computing the terminal conditions is not easy to solve. In literature, this problem is handled in many ways, such as LDI [16] or LPV [13, 12] model representations of the system. In [10] the authors state that the command governors strategy ensures the viability property, which implies the existence of such a not trivial invariant set.

Remark 2. The local nature of the terminal controller and the difficulty of computing set Γ makes this set potentially small. In fact, a sensible choice of Γ is as level sets of the local Lyapunov function. In order to minimize the effect of the conservative nature of the terminal ingredients, a formulation with a prediction horizon larger that the control horizon [17] can be used. This provides an enhanced closed loop performance and a larger domain of attraction maintaining the stabilizing properties.

4 Example

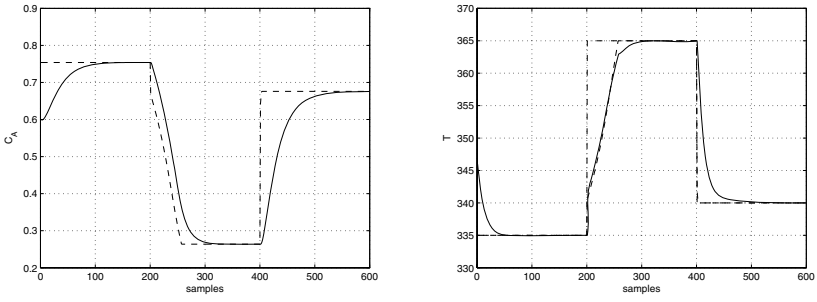
This section presents the application of the proposed controller to the highly nonlinear model of a continuous stirred tank reactor (CSTR), [17]. Assuming constant liquid volume, the CSTR for an exothermic, irreversible reaction, $A \rightarrow B$, is described by the following model:

$$\begin{aligned} \dot{C}_A &= \frac{q}{V}(C_{Af} - C_A) - k_o e^{(\frac{-E}{RT})} C_A \\ \dot{T} &= \frac{q}{V}(T_f - T) - \frac{\Delta H}{\rho C_p} k_o e^{(\frac{-E}{RT})} C_A + \frac{UA}{V \rho C_p}(T_c - T) \end{aligned} \tag{3}$$

where C_A is the concentration of A in the reactor, T is the reactor temperature and T_c is the temperature of the coolant stream. The objective is to regulate $y = x_2 = T$ and $x_1 = C_A$ by manipulating $u = T_c$. The constraints are $0 \leq C_A \leq 1$ mol/l, $280K \leq T \leq 370K$ and $280K \leq T_c \leq 370$. The nonlinear discrete time model of system (3) is obteined by defining the state vector $x = [C_A - C_A^{eq}, T - T^{eq}]^T$ and $u = T_c - T_c^{eq}$ and by discretizing equation (3) with $t = 0.03$ min as sampling time. We considered an MPC with $N_c = 3$ and $N_p = 10$ and with $Q = diag(1/0.5, 1/350)$ and $R = 1/300$ as weighting matrices.

The output $y = x_2$ has been chosen as the parameter θ. To illustrate the proposed controller, three references has been considered, $Ref_1 = 335$ K, $Ref_2 = 365$ K and $Ref_3 = 340$ K. In figures 1(a) and 1(b) the evolutions of the states (solid lines), the artificial references (dashed lines) and the real one (dashed-dotted line) are showed. See how the controller leads the system to track the artificial reference when the real one is unfeasible. The artificial reference represents the feasible trajectory determined by the value of $\tilde{\theta}_s$ that minimizes $V_O(\theta - \theta_t)$.

The terminal region and cost function have been computed in an explicit form, depending on θ. Linearizing the system around θ, the control gain has been



(a) Evolution of C_A. (b) Evolution of T.

Fig. 1 Evolutions of the states

computed explicitly, depending on θ, $K(\theta)$. Defining $A_K(\theta) = A + BK(\theta)$ and $Q(\theta)^* = Q + K(\theta)^T RK(\theta)$, $P(\theta)$ has been found as solution of $A_K(\theta)^T P(\theta) A_K(\theta) - P(\theta) = -Q(\theta)^*$. Then, $V_f(x - g_x(\theta), \theta) = (x - g_x(\theta))^T P(\theta)(x - g_x(\theta))$ and $\Gamma = \{(x, \theta) \in \mathbb{R}^{n+q} : V_f(x - g_x(\theta), \theta) \leq \alpha\}$, where $\alpha > 0$ is such that for all $(x, \theta) \in \Gamma$, $x \in X$, $u = K(\theta)(x - g_x(\theta)) + g_u(\theta) \in U$ and

$$V_f(f(x, k(x, \theta)) - g_x(\theta), \theta) - V_f(x - g_x(\theta), \theta) \leq -(x - g_x(\theta))^T Q(\theta)^* (x - g_x(\theta)).$$

5 Conclusion

In this paper a novel MPC controller for tracking changing references for constrained nonlinear systems has been presented, as extension, to the nonlinear case, of the one presented in [14]. This controller ensures feasibility by means of adding an artificial steady state and input as decision variable of the optimization problem. Convergence to an admissible target steady state is ensured by using a modified cost function and a stabilizing extended terminal constraint. Optimality is ensured by means of an offset cost function which penalizes the difference between the artificial reference and the real one. The proposed controller can be formulated with a prediction horizon larger than the control horizon. This formulation provides an enhanced closed loop performance and a larger domain of attraction maintaining the stabilizing properties. The properties of the controller have been illustrated in an example.

References

1. Camacho, E.F., Bordons, C.: Model Predictive Control, 2nd edn. Springer, Heidelberg (2004)
2. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: Stability and optimality. Automatica 36, 789–814 (2000)

3. Rossiter, J.A., Kouvaritakis, B., Gossner, J.R.: Guaranteeing feasibility in constrained stable generalized predictive control. IEEE Proc. Control theory Appl. 143, 463–469 (1996)
4. Bemporad, A., Casavola, A., Mosca, E.: Nonlinear control of constrained linear systems via predictive reference management. IEEE Transactions on Automatic Control 42, 340–349 (1997)
5. Pannocchia, G., Kerrigan, E.: Offset-free reciding horizon control of constrained linear systems. AIChE Journal 51, 3134–3146 (2005)
6. Alvarado, I.: Model Predictive Control for Tracking Constrained Linear Systems. PhD thesis, Univ. de Sevilla (2007)
7. Findeisen, R., Chen, H., Allgöwer, F.: Nonlinear predictive control for setpoint families. In: Proc. of the American Control Conference (2000)
8. Magni, L., De Nicolao, G., Scattolini, R.: Output feedback and tracking of nonlinear systems with model predictive control. Automatica 37, 1601–1607 (2001)
9. Magni, L., Scattolini, R.: On the solution of the tracking problem forn non-linear systems with MPC 36(8), 477–484 (2005)
10. Angeli, D., Mosca, E.: Command governors for constrained nonlinear systems. IEEE Transactions on Automatic Control 44, 816–820 (1999)
11. Gilbert, E., Kolmanovsky, I.: Nonlinear tracking control in the presence of state and control constraints: A generalized reference governor. Automatica 38, 2063–2073 (2002)
12. Chisci, L., Zappa, G.: Dual mode predictive tracking of piecewise constant references for constrained linear systems. Int. J. Control 76, 61–72 (2003)
13. Chisci, L., Falugi, P., Zappa, G.: Predictive tracking control of constrained nonlinear systems. IEE Proc.-Control Theory Appl. 152(3), 309–316 (2005)
14. Limon, D., Alvarado, I., Alamo, T., Camacho, E.F.: MPC for tracking of piece-wise constant references for constrained linear systems. Automatica (in press) (2008)
15. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2006)
16. Cannon, M., Deshmukh, V., Kouvaritakis, B.: Nonlinear model predictive control with polytopic invariant sets. Automatica 39, 1487–1494 (2003)
17. Magni, L., De Nicolao, G., Magnani, L., Scattolini, R.: A stabilizing model-based predictive control algorithm for nonlinear systems. Automatica 37, 1351–1362 (2001)

A Flatness-Based Iterative Method for Reference Trajectory Generation in Constrained NMPC

J.A. De Doná, F. Suryawan, M.M. Seron, and J. Lévine

Abstract. This paper proposes a novel methodology that combines the differential flatness formalism for *trajectory generation* of nonlinear systems, and the use of a model predictive control (MPC) strategy for *constraint handling*. The methodology consists of a *trajectory generator* that generates a reference trajectory parameterised by splines, and with the property that it satisfies performance objectives. The reference trajectory is generated *iteratively* in accordance with information received from the MPC formulation. This interplay with MPC guarantees that the trajectory generator receives *feedback from present and future constraints* for real-time trajectory generation.

Keywords: flatness, trajectory generation, B-splines, Nonlinear MPC.

1 Introduction

Differential flatness [1] is a property of some controlled (linear or nonlinear) dynamical systems, often encountered in applications, which allows for a complete parameterisation of all system variables (inputs and states) in terms of a finite number of independent variables, called *flat outputs*, and a finite number of their time derivatives. We consider a general system

$$\dot{x}(t) = f\big(x(t), u(t)\big), \tag{1}$$

J.A. De Doná, F. Suryawan, and M.M. Seron
CDSC, School of Electrical Engineering and Computer Science,
The University of Newcastle, Callaghan, NSW 2308, Australia
e-mail: {jose.dedona,fajar.suryawan,maria.seron}@newcastle.edu.au

J.A. De Doná and J. Lévine
CAS, Mathématiques et Systèmes, Mines-ParisTech, 35 rue Saint-Honoré.
77300 Fontainebleau, France
e-mail: jean.levine@ensmp.fr

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 325–333.
springerlink.com

where $x(t) \in \mathbb{R}^n$ is the state vector and $u(t) \in \mathbb{R}^m$ is the input vector. If the system is flat [1], we can write all trajectories $(x(t), u(t))$ satisfying the differential equation in terms of a finite set of variables, known as the flat output, $y(t) \in \mathbb{R}^m$ and a finite number of their derivatives:

$$\begin{aligned} x(t) &= \Upsilon\big(y(t), \dot{y}(t), \ddot{y}(t), \ldots, y^{(r)}(t)\big), \\ u(t) &= \Psi\big(y(t), \dot{y}(t), \ddot{y}(t), \ldots, y^{(r+1)}(t)\big). \end{aligned} \tag{2}$$

The parameterisation (2), afforded by the flatness property, allows to simplify (especially in the case of *nonlinear* flat systems) the generation of reference trajectories (trajectory planning). Typically, some 'desired' reference trajectory is prescribed for the flat output, y^{ref}, and the corresponding input and state trajectories for the system are obtained from (2); namely, $u^{\text{ref}}(t) = \Psi\big(y^{\text{ref}}(t), \dot{y}^{\text{ref}}(t), \ldots, (y^{\text{ref}}(t))^{(r+1)}\big)$, $x^{\text{ref}}(t) = \Upsilon\big(y^{\text{ref}}(t), \ldots, (y^{\text{ref}}(t))^{(r)}\big)$. However, a very common requirement in engineering applications is for some of the variables of the dynamical system to satisfy a number of constraints, usually expressed as *inequality constraints*. For example the input and state of the system can be required to satisfy $u \in \mathbb{U}$ and $x \in \mathbb{X}$, where $\mathbb{U} \subset \mathbb{R}^m$ and $\mathbb{X} \subset \mathbb{R}^n$ are specified *constraint sets*. The presence of such constraints makes trajectory generation for nonlinear systems (in general) a highly nontrivial task, due to the ensuing nonlinearity of the mappings $\Upsilon(\cdot)$ and $\Phi(\cdot)$ in (2). (In particular, it is typically very difficult, to specify constraint sets for the flat output variables y in terms of the constraint sets for u and x, respectively, $\mathbb{U}$ and $\mathbb{X}$.)

In this paper, we propose a novel methodology that exploits the flatness parameterisation (2) for *trajectory generation* and the use of the Model Predictive Control (MPC) strategy for *constraints handling*. The methodology consists of a *trajectory generator* module, that generates a reference trajectory $y^{\text{ref}}(t)$ with the property that it satisfies performance objectives (e.g., satisfies given *initial* and *final conditions*, passes through a given set of *way-points*, etc.). There are points of contact between some aspects of the approach advocated in this paper and, for example, the work in [3] where the problem of generation of a reference trajectory for a nonlinear flat system subject to constraints is formulated as a NonLinear Programming (NLP) problem. One of the main drawbacks of posing the problem as a NLP optimisation problem is that, in general, it is very difficult to prove convergence, or convergence to a global optimum. Hence, in this paper we explore an alternative algorithm for trajectory generation for nonlinear flat systems, in the presence of constraints, that is based on the information provided by a model predictive control (MPC) formulation. The approach is, to the best of the authors knowledge, the first attempt to combine the differential flatness formalism with model predictive control techniques in an iterative algorithm for constrained nonlinear trajectory generation. No proofs of convergence are available at present, due to the challenging nature of these problems, and this will be of concern in future work. However, simulation results, as the ones presented in this paper, are promissory and indicate that the effort of developing such algorithms and investigating formal proofs of convergence is worthwhile.

Thus, in the methodology investigated in this paper, the reference trajectory $y^{\text{ref}}(t)$ is generated *iteratively* in accordance with information (predicted in real time) received from an MPC formulation. That way, the trajectory generator receives "feedback from the (present and future) constraints" of the system while generating the desired trajectory. Thus, the proposed method unites two important properties. Firstly, since the trajectories are generated via the flatness parameterisation (2), with "feedback from the constraints," they constitute *natural* trajectories for the *nominal* model to follow. And, secondly, the information generated by an MPC formulation (via the solution of a Quadratic Programming optimisation, based on the linearised dynamics around the given reference trajectory) ensures that the system constraints are taken into account.

2 Flatness and Trajectory Parameterisation

We consider the problem of steering system (1) from an initial state at time t_0 to a final state at time t_f. Note that, in a Model Predictive Control context, this fixed interval problem is one window of a bigger scheme, implemented repeatedly in a receding horizon fashion. In order to generate a suitable reference trajectory, we will use a spline parameterisation, as explained in the following sections.

2.1 Parameterisation of Flat Outputs and Their Derivatives

We parameterise each of the flat outputs $y_j(t)$, $j = 1,\ldots,m$, as

$$y_j(t) = \sum_{i=1}^{N} \lambda_i(t) P_{ij}; \qquad t \in [t_0\ ,\ t_f], \tag{3}$$

where $\lambda_i, i = 1,\ldots,N$, is a set of basis functions, which is the same for each flat output y_j. The basis functions are assumed to be $\lambda_i \in \mathscr{C}^{r+1}[t_0,t_f], i = 1,\ldots,N$. This reduces the problem of characterising a function in an infinite dimensional space to finding a finite set of parameters P_{ij}. In a discrete set of $M+1$ sampling times, $t_0,t_1,\ldots,t_M = t_f$, this parameterisation becomes

$$Y_j = G_0 P_j, \tag{4}$$

where $Y_j \triangleq [y_j(t_0), y_j(t_1), \ldots, y_j(t_f)]^{\mathrm{T}}$, $P_j \triangleq [P_{1j},\ldots,P_{Nj}]^{\mathrm{T}}$ is a vector containing the parameters P_{ij}, $i = 1,\ldots,N$, defined in (3), and

$$G_0 \triangleq \begin{bmatrix} \lambda_1(t_0) & \ldots & \lambda_N(t_0) \\ \vdots & \ddots & \vdots \\ \lambda_1(t_f) & \ldots & \lambda_N(t_f) \end{bmatrix} \tag{5}$$

is the *basis function matrix* (also known as *blending matrix*). Collecting all the m flat outputs, we have

$$
\begin{aligned}
Y \triangleq \begin{bmatrix} Y_1 \; Y_2 \; \dots \; Y_m \end{bmatrix} &= \begin{bmatrix} y_1(t_0) & y_2(t_0) & \dots & y_m(t_0) \\ \vdots & \vdots & \ddots & \vdots \\ y_1(t_f) & y_2(t_f) & \dots & y_m(t_f) \end{bmatrix} \\
&= G_0 \cdot \begin{bmatrix} P_1 \; P_2 \; \dots P_m \end{bmatrix} = G_0 P = Y(P),
\end{aligned} \tag{6}
$$

where Y is an $(M+1)\times m$ output matrix, G_0 is the $(M+1)\times N$ blending matrix, and $P \triangleq \begin{bmatrix} P_1 \; P_2 \; \dots P_m \end{bmatrix}$ is an $N \times m$ matrix containing the coefficients P_{ij} of the parameterisation (3). The rows of P are m-dimensional vectors called *control points*.

Furthermore, we can also build the time-derivatives of y_i at discrete points in time, by successively differentiating (3) followed by time-discretisation. Doing this and using the notation as in (6), we obtain

$$
Y^{(1)} = G_1 P; \quad Y^{(2)} = G_2 P; \quad Y^{(3)} = G_3 P; \quad \dots \quad Y^{(r+1)} = G_{r+1} P; \tag{7}
$$

where $Y^{(q)} \triangleq [Y_1^{(q)} \; Y_2^{(q)} \; \dots \; Y_m^{(q)}]$, and

$$
Y_j^{(q)} \triangleq \begin{bmatrix} \frac{d^q}{dt^q} y_j(t)\big|_{t=t_0} \\ \vdots \\ \frac{d^q}{dt^q} y_j(t)\big|_{t=t_f} \end{bmatrix}; \qquad G_q \triangleq \begin{bmatrix} \frac{d^q}{dt^q} \lambda_1(t)\big|_{t=t_0} & \dots & \frac{d^q}{dt^q} \lambda_N(t)\big|_{t=t_0} \\ \vdots & \ddots & \vdots \\ \frac{d^q}{dt^q} \lambda_1(t)\big|_{t=t_f} & \dots & \frac{d^q}{dt^q} \lambda_N(t)\big|_{t=t_f} \end{bmatrix}, \tag{8}
$$

with $j = 1, \dots, m$ and $q = 1, \dots, r+1$.

2.2 Trajectory Parameterisation Using Splines

Given a reference trajectory parameterised as in (6), $Y^{\text{ref}} = G_0 P^{\text{ref}}$, with specified reference control points P^{ref}, in this section we will show how to parameterise variations around that reference trajectory using splines. In this paper, clamped B-splines [2] are chosen as basis functions which results in the blending matrix G_0 having a particular structure. Namely, G_0 has only one non-zero element in the first row (which lies in the first column) and only one non-zero element in the last row (which lies in the last column). The matrix G_1 has two non-zero elements in the first row (which lie in the first and second column) and two non-zero elements in the last row (which lie in the last and second-last column). The matrix G_2 has a similar property with three non-zero elements, etc. More properties of B-splines can be found in, e.g., [2].

Notice from (3) that,

$$
\frac{d^q}{dt^q} y_j(t)\big|_{t=t_0} = \sum_{i=1}^{N} \frac{d^q}{dt^q} \lambda_i(t)\big|_{t=t_0} P_{ij}, \tag{9}
$$

for $q = 0, 1, \dots, r+1$; $j = 1, \dots, m$. We can see from (9) and the structure of G_0 discussed above that $y_j(t_0) = \lambda_1(t_0) P_{1j}$, $j = 1, \dots, m$, and hence, by fixing the first row of P, $P_{1j} = P_{1j}^{\text{ref}}$, $j = 1, \dots, m$, the flat outputs at time t_0 are fixed and equal to the corresponding values of the reference trajectory. Fixing more rows of P (up to the

order of the B-spline) fixes the flat output derivatives at time t_0 (e.g. fixing two rows fixes the first derivatives, three rows fixes the second derivatives, etc.). This property (made possible by the structure of G_q, $q = 0, 1, \ldots$) can be used to maintain fixed end-points. For example, prescribed position and first and second order derivatives of the flat output at times t_0 and t_f, as in the *rest-to-rest* case, can be maintained by holding the 'external' control points (the three topmost and the three bottommost rows of P in Eq. (6)) fixed. This can be achieved by reparameterising P as:

$$P = P^{\text{ref}} + \rho \hat{P}; \qquad \rho = [0\, I\, 0]^{\text{T}}, \tag{10}$$

where matrix $\hat{P}$ is an $[N-(l_1+l_2)] \times m$ matrix that parameterises the deviation from the 'internal' control points of P^{ref} and ρ is an $N \times [N-(l_1+l_2)]$ matrix with the l_1 top rows set equal to zero, the l_2 bottom rows set equal to zero and the identity matrix of dimension $[N-(l_1+l_2)] \times [N-(l_1+l_2)]$ in the middle.

3 Using MPC to Shape the Reference Trajectory

In this section we will develop an iterative algorithm for trajectory generation for nonlinear systems, subject to constraints, that is based on information provided by model predictive control (MPC). The main motivation for resorting to MPC is to exploit the well-known capabilities for handling constraints of this control technique. The basic idea is to propose an initial reference trajectory based purely on performance considerations, parameterised as in (6), i.e. $Y^{\text{ref},0} = G_0 P^{\text{ref},0}$ (it is assumed here that an initial set of *reference* control points $P^{\text{ref},0}$ is specified), and to then use an MPC formulation to give information as to how well that trajectory can be followed in the presence of constraints and, moreover, which parts of the original trajectory are problematic and should be modified. Then a new reference trajectory is generated based on a trade-off between the information obtained from MPC (this information can be regarded as the *feedback from the constraints*) and the original performance specifications. This interplay between performance objectives and MPC (feedback from constraints) is then iterated, and the challenge is to devise an algorithm such that the iteration converges to a suitable reference trajectory.

3.1 MPC Formulation

We will assume, for simplicity, that the flat output is given by a (possibly nonlinear) combination of the states:

$$y(t) = h\big(x(t)\big). \tag{11}$$

Note that, although this is not the most general case for flat systems, many examples of practical interest satisfy this assumption (e.g., models of cranes, nonholonomic cars, etc.). Given a specified reference trajectory for the flat output, parameterised by control points P^{ref} as explained in the preceding section:

$$y_j^{\text{ref}}(t) = \sum_{i=1}^{N} \lambda_i(t) P_{ij}^{\text{ref}}; \quad t \in [t_0, t_f], \tag{12}$$

for $j = 1, \ldots, m$, we compute the corresponding state and input reference trajectories, $x^{\text{ref}}(t)$ and $u^{\text{ref}}(t)$, respectively, from (2). Note, in particular, that the flatness formulation implies that these trajectories satisfy the system's equation $\dot{x}^{\text{ref}}(t) = f\big(x^{\text{ref}}(t), u^{\text{ref}}(t)\big)$. Then, the dynamics of (1) together with the output equation (11) are linearised along the reference trajectory $\big(u^{\text{ref}}(t), x^{\text{ref}}(t), y^{\text{ref}}(t)\big)$ as follows: $\dot{\tilde{x}}(t) = A(t)\tilde{x}(t) + B(t)\tilde{u}(t)$, $\tilde{y}(t) = C(t)\tilde{x}(t)$, where:

$$\tilde{u}(t) \triangleq u(t) - u^{\text{ref}}(t), \quad \tilde{x}(t) \triangleq x(t) - x^{\text{ref}}(t), \quad \tilde{y}(t) \triangleq y(t) - y^{\text{ref}}(t), \tag{13}$$

and $A(t) = \big(\partial f/\partial x\big)\big|_{x^{\text{ref}}(t), u^{\text{ref}}(t)}$, $B(t) = \big(\partial f/\partial u\big)\big|_{x^{\text{ref}}(t), u^{\text{ref}}(t)}$ and $C(t) = \big(\partial h/\partial x\big)\big|_{x^{\text{ref}}(t)}$. The resulting linear time varying system is then discretised in time, so that the following time varying discrete time system is obtained:

$$\tilde{x}_{k+1} = A_k \tilde{x}_k + B_k \tilde{u}_k, \qquad \tilde{y}_k = C_k \tilde{x}_k. \tag{14}$$

In the discretisation (14) we consider a sampling interval $T_s \triangleq (t_f - t_0)/M$, so that exactly M sampling intervals fit in the interval of definition of the splines, $[t_0, t_f]$. Moreover, we define a grid of equally spaced sampling times, $t_k = t_0 + kT_s$, $k = 0, \ldots, M$. Note that the variables in (14) (cf. (13)) are measured with respect to the reference trajectory. Thus we will consider an MPC formulation for the time varying system (14) where the performance objective is regulation to the origin (this will ensure tracking of the respective reference trajectories).

Given the current state of the plant at time t, $x(t)$, we compute $\tilde{x}_0 \triangleq x(t) - x^{\text{ref}}(t_0)$ (where $x^{\text{ref}}(t_0)$ is obtained from (12) using (2)). The aim is to find the M-move control sequence $\{\tilde{u}_k\} \triangleq \{\tilde{u}_0, \ldots, \tilde{u}_{M-1}\}$ that minimises the finite horizon objective function:

$$V_M(\{\tilde{x}_k\}, \{\tilde{u}_k\}, \{\tilde{y}_k\}) \triangleq \frac{1}{2}\tilde{x}_M^{\mathrm{T}} P \tilde{x}_M + \frac{1}{2}\sum_{k=0}^{M-1} \tilde{y}_k^{\mathrm{T}} Q \tilde{y}_k + \frac{1}{2}\sum_{k=0}^{M-1} \tilde{u}_k^{\mathrm{T}} R \tilde{u}_k, \tag{15}$$

subject to the system equations (14) and $\tilde{x}_0 \triangleq x(t) - x^{\text{ref}}(t_0)$, and where $P \geq 0$, $Q \geq 0$, $R > 0$, and M is the prediction horizon. Using the standard vectorised notation $\tilde{\mathbf{x}} \triangleq [\tilde{x}_1^{\mathrm{T}} \ldots \tilde{x}_M^{\mathrm{T}}]^{\mathrm{T}}$, $\tilde{\mathbf{u}} \triangleq [\tilde{u}_0^{\mathrm{T}} \ldots \tilde{u}_{M-1}^{\mathrm{T}}]^{\mathrm{T}}$, the cost function (15) can be written in compact form as:

$$V_M = \frac{1}{2}\tilde{x}_0^{\mathrm{T}} C_0^{\mathrm{T}} Q C_0 \tilde{x}_0 + \frac{1}{2}\tilde{\mathbf{x}}^{\mathrm{T}} \mathbf{Q} \tilde{\mathbf{x}} + \frac{1}{2}\tilde{\mathbf{u}}^{\mathrm{T}} \mathbf{R} \tilde{\mathbf{u}}, \tag{16}$$

where $\mathbf{Q} \triangleq \operatorname{diag}\{C_1^{\mathrm{T}} Q C_1, \ldots, C_{M-1}^{\mathrm{T}} Q C_{M-1}, P\}$ and $\mathbf{R} \triangleq \operatorname{diag}\{R, \ldots, R\}$.

The system's state evolution from $k = 0$ to M can be expressed as $\tilde{\mathbf{x}} = \Gamma \tilde{\mathbf{u}} + \Omega \tilde{x}_0$, where Γ and Ω are formed from the system's A_k and B_k matrices (see, e.g., [4]). Substituting this expression for $\tilde{\mathbf{x}}$ into (16) yields: $V_M = \bar{V} + \frac{1}{2}\tilde{\mathbf{u}}^{\mathrm{T}} H \tilde{\mathbf{u}} + \tilde{\mathbf{u}}^{\mathrm{T}} F \tilde{x}_0$, where $\bar{V}$ is a constant term, $H \triangleq \Gamma^{\mathrm{T}} \mathbf{Q} \Gamma + \mathbf{R}$ and $F \triangleq \Gamma^{\mathrm{T}} \mathbf{Q} \Omega$.

If the problem is constrained, for example with input constraints $|u(t)| \leq u_{\max}$, then the solution is obtained from the following quadratic program:

$$\begin{aligned} \tilde{\mathbf{u}}^{\text{opt}} = [(\tilde{u}_0^{\text{opt}})^{\text{T}} \dots (\tilde{u}_{M-1}^{\text{opt}})^{\text{T}}]^{\text{T}} \triangleq \arg\min_{\tilde{\mathbf{u}}} \frac{1}{2}\tilde{\mathbf{u}}^{\text{T}} H \tilde{\mathbf{u}} + \tilde{\mathbf{u}}^{\text{T}} F \tilde{x}_0 \\ \text{subject to} \\ |\mathbf{u}^{\text{ref}} + \tilde{\mathbf{u}}| \leq U_{\max}, \end{aligned} \tag{17}$$

where $\mathbf{u}^{\text{ref}} \triangleq [(u^{\text{ref}}(t_0))^{\text{T}}\ (u^{\text{ref}}(t_1))^{\text{T}} \dots (u^{\text{ref}}(t_{M-1}))^{\text{T}}]^{\text{T}}$, $U_{\max} \triangleq [u_{\max}^{\text{T}} \dots u_{\max}^{\text{T}}]^{\text{T}}$, and the absolute value and the inequality are interpreted element-wise. (Other types of constraints, e.g., state and output constraints, can be incorporated in (17) in a straightforward manner.)

The corresponding j-th flat output trajectory, $j = 1, \dots, m$, obtained by MPC is computed from the result of (17), using (13) and (14). Using the expression $\tilde{\mathbf{x}} = \Gamma \tilde{\mathbf{u}} + \Omega \tilde{x}_0$, the MPC flat output trajectory can be expressed as:

$$Y_j^{\text{mpc}} \triangleq \mathbf{C}_j \begin{bmatrix} \tilde{x}_0 \\ \Gamma \tilde{\mathbf{u}}^{\text{opt}} + \Omega \tilde{x}_0 \end{bmatrix} + Y_j^{\text{ref}}, \tag{18}$$

where Y_j^{mpc} and Y_j^{ref} are the j-th flat output sequences stacked over time [defined similarly to Y_j in (4)], and $\mathbf{C}_j \triangleq \text{diag}\{C_{0,j}, \dots, C_{M,j}\}$, where $C_{k,j}$ is the j-th row of the time-varying matrix C_k, defined in (14) for $k = 0, \dots, M$. In an MPC implementation, one then applies the first control move obtained in (17), $\tilde{u}_0^{\text{opt}}$, and the process is repeated in a receding horizon fashion. However, in our proposed implementation (see next subsection) this process is iterated before the actual control input is applied.

3.2 *Iterative Method for Reference Trajectory Generation*

In this section we present the iterative algorithm that is proposed in this paper. The algorithm starts from a set of specified initial control points $P^{\text{ref},0}$ that parameterise an initial reference trajectory $Y^{\text{ref},0} = G_0 P^{\text{ref},0}$ which is generated based on performance considerations, and then it utilises the information about the effect of the constraints, provided by the MPC formulation, to update the reference trajectory through successive sets of control points, $P^{\text{ref},0}, P^{\text{ref},1}, \dots, P^{\text{ref,k}}, \dots$, etc.

Step 1. Given a set of control points $P^{\text{ref,k}}$;

Step 2. Compute, from (6), $Y^{\text{ref,k}} = G_0 P^{\text{ref,k}}$;

Step 3. Compute $Y_j^{\text{mpc,k}}$ from (12)–(18). Note that $Y^{\text{mpc,k}}$ so obtained is a (in general nonlinear) function of $P^{\text{ref,k}}$, that is, $Y^{\text{mpc,k}} = G\big(P^{\text{ref,k}}\big)$.

Step 4. Given $Y^{\text{mpc,k}}$, find the variation of the 'internal' control points in the parameterisation (10), denoted $\hat{P}^{\text{mpc,k}}$, that gives a reference trajectory that is closest in a least-squares sense to $Y^{\text{mpc,k}}$. Namely,

$$\hat{P}_j^{\text{mpc,k}} = \big((G_0\rho)^{\text{T}} G_0 \rho\big)^{-1} \big(G_0 \rho\big)^{\text{T}} \big(Y_j^{\text{mpc,k}} - G_0 P_j^{\text{ref,k}}\big). \tag{19}$$

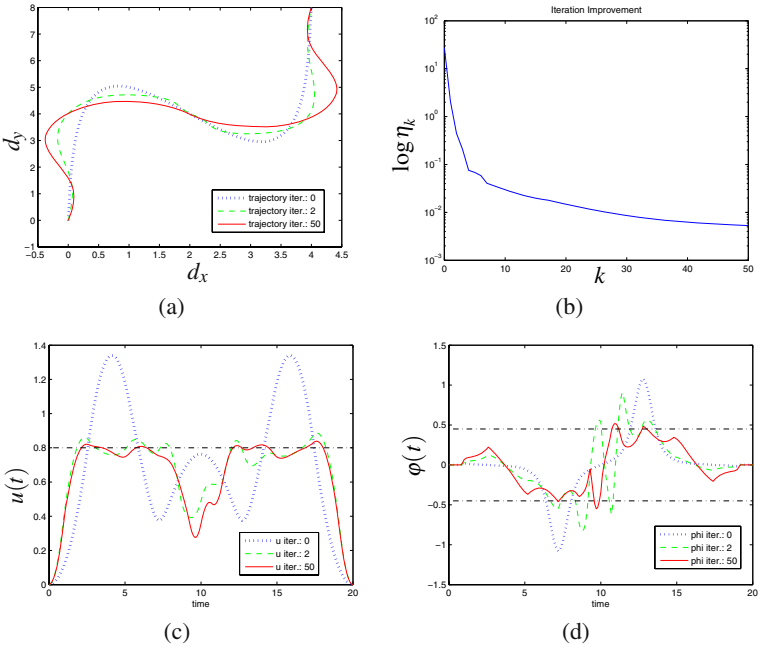


Fig. 1 Initial reference trajectory, 2nd and 50th iteration. (a) Flat output $y = \left(d_x, d_y\right)$; (b) Measure of convergence η_k; (c) Input $u(t)$; and, (d) Input $\varphi(t)$

Step 5. Update the control points according to: $P^{\text{ref,k+1}} = P^{\text{ref,k}} + \rho \hat{P}^{\text{mpc,k}}$.

Step 6. While (a weighted 2-norm of) the difference $\left(P^{\text{ref,k+1}} - P^{\text{ref,k}}\right)$ is larger than a prescribed tolerance level and the maximum number of iterations is not reached: assign $P^{\text{ref,k}} \leftarrow P^{\text{ref,k+1}}$ and go to Step 1.

Note from Steps 1–5 that the proposed algorithm implements a recursion $P^{\text{ref,k+1}} = F\left(P^{\text{ref,k}}\right)$, whose complexity depends predominantly on the (in general, nonlinear) mapping $Y^{\text{mpc,k}} = G\left(P^{\text{ref,k}}\right)$. The convergence properties of the recursive mapping, $P^{\text{ref,k+1}} = F\left(P^{\text{ref,k}}\right)$, will be investigated in future work.

4 Simulation Example

In this section we will test the previous algorithm on a classical example of a flat system, a nonholonomic car system. The system is modeled by the equations: $\dot{d}_x(t) = u(t)\cos\theta(t)$, $\dot{d}_y(t) = u(t)\sin\theta(t)$ and $\dot{\theta}(t) = (1/l)u(t)\tan\varphi(t)$, where the state $d_x(t)$ is the displacement in the "x-direction", the state $d_y(t)$ is the displacement in the "y-direction", the state $\theta(t)$ is the angle of the car with respect to the x-axis, the input $u(t)$ is the velocity of the car, the input $\varphi(t)$ is the angle of the steering wheels, and l is the distance between the front and the rear wheels. It is straightforward to determine that the flat output for this system is given by $y(t) = \left(d_x(t), d_y(t)\right)$.

A matrix of initial control points, $P^{\text{ref},0}$, is chosen so that, together with the parameterisation (12) using cubic B-splines $\lambda_i(t)$, gives the initial reference trajectory $y^{\text{ref},0}$ shown with a dotted line in Figure 1(a). The control inputs are assumed to be subject to the constraints $u(t) \leq 0.8$ and $|\varphi| \leq 0.45$. The inputs corresponding to the initial reference trajectory $y^{\text{ref},0}$ are shown with dotted lines in Figures 1(c) and (d), far exceeding the constraint limits. The result after 2, respectively 50, iterations of the algorithm is shown in Figures 1(a), 1(c), and 1(d) with dashed, respectively solid, lines. Notice that the algorithm produces a final reference trajectory which is close to the initial reference trajectory and with associated inputs only mildly exceeding the constraints. In addition, the initial and final end-point conditions are maintained. A measure of convergence of the algorithm, $\eta_k = \sum_{j=1}^{m} (P_j^{\text{ref,k}} - P_j^{\text{ref,k-1}})^{\text{T}} G_0^{\text{T}} G_0 (P_j^{\text{ref,k}} - P_j^{\text{ref,k-1}})$, is shown in Figure 1(b).

5 Conclusion

A novel methodology combining the differential flatness formalism for *trajectory generation* of nonlinear systems, and the use of a model predictive control strategy for *constraint handling* has been proposed. The methodology consists of a *trajectory generator* that generates a reference trajectory parameterised by splines, and with the property that it satisfies performance objectives. The reference trajectory is generated *iteratively* in accordance with information received from the MPC formulation. The performance of the iterative scheme has been illustrated with a simulation example. Future work will focus on investigating the conditions required to establish the convergence of the iterative algorithm, and on evaluating its computational performance for real-time applications.

References

1. Fliess, M., Lévine, J., Martin, P., Rouchon, P.: Flatness and defect of non-linear systems: Introductory theory and examples. International Journal of Control 61, 1327–1361 (1995)
2. Piegl, L., Tiller, W.: The NURBS Book. Springer, Heidelberg (1997)
3. Flores, M.E., Milam, M.B.: Trajectory generation for differentially flat systems via NURBS basis functions with obstacle avoidance. In: 2006 American Control Conference, Minneapolis, USA (2006)
4. Goodwin, G.C., Seron, M., De Doná, J.A.: Constrained Control and Estimation: An Optimisation Approach. Communications and Control Engineering Series. Springer, Heidelberg (2005)

Nonlinear Model Predictive Path-Following Control

Timm Faulwasser and Rolf Findeisen

Abstract. In the frame of this work, the problem of following parametrized reference paths via nonlinear model predictive control is considered. It is shown how the use of parametrized paths introduces new degrees of freedom into the controller design. Sufficient stability conditions for the proposed model predictive path-following control are presented. The method proposed is evaluated via simulations of an autonomous mobil robot.

Keywords: reference tracking, path-following, nonlinear systems, model predictive control, parametrized reference, stability.

1 Introduction

The design of feedback controllers for dynamical systems is usually subject to one of the following purposes: either suppress disturbances to stabilize a system around a fixed reference state via an appropriate input, or influence its dynamic behavior such that the system states or outputs converge to a time-varying reference signal. The presence of input and state constraints makes both problems considerably tougher.

Even in the absence of input and state constraints the controller design for tracking and tracking-related problems is a non-trivial task, especially for nonlinear systems. For example the design of output tracking controllers for non-minimum phase systems is subject to fundamental limits of achieveable tracking performance. These limits can arise from unstable zero-dynamics of non-minimum phase systems – see inter alia [12] for details on the linear case and [14] for the nonlinear case –. Recently, path-following approaches have shown their ability to circumvent these fundamental

Timm Faulwasser and Rolf Findeisen
Institute for Automation Engineering, Otto-von-Guericke
University Magdeburg, Universitätsplatz 2, 39104 Magdeburg, Germany
e-mail: {timm.faulwasser,rolf.findeisen}@ovgu.de

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 335–343.
springerlink.com

performance limits [1, 15]. Most of the existing methods are based on the idea of using parametrized reference signals instead of time-dependent reference trajectories. However, the aforementioned results on path-following are limited, since input and state constraints are not considered.

One control strategy that allows to take constraints on states and inputs into account is nonlinear model predictive control (NMPC). In [11] one of the first results on the application of NMPC to tracking problems is outlined. Robust output feedback tracking for time discrete systems is discussed in [8]. [9] presents results on the tracking of asymptotically constant references for the continuous case. Furthermore, NMPC can be applied to the tracking of non-holonomous wheeled robots as well, e.g. [6]. All these NMPC approaches towards the tracking problem rely on the definition of the tracking error as the difference between the output or state and a time-depending reference. Parametrized references in the context of (N)MPC have previously been discussed inter alia in [7]. There, tracking of piecewise constant reference signals is considered. As it will be shown, the use of parametrized reference signals leads to a control structure which affects both the system inputs and the evolution of reference signals. Feedback control of reference evolution can be also considered by applying reference governors. Reference governors, as presented inter alia in [2, 13], are usually hierarchically structured. An inner control loop stabilizes the system, while an outer loop controls the reference evolution such that input and state constraints are fulfilled. Unlike these approaches the results presented here only rely on one control loop.

The contribution of this paper is a scheme for model predictive path-following control (MPFC) which implements parametrized references into a NMPC setup while guaranteeing stability. Combining NMPC and the core idea of path-following leads to additional degrees of freedom in the controller design. These can be utilized to guarantee stability and to achieve better performance. In contrast to other works on path-following [1, 15], which apply back-stepping techniques to construct *output-feedback* controllers, here the results are based on *state-feedback*.

The remain of this work is structured as follows: Section 2 introduces the path-following problem and shows how parametrized references can be utilized in NMPC. Furthermore, results on the stability of the proposed model predictive path-following control are given. In Section 3 MPFC is applied to a model of a simple wheeled robot. Section 4 gives final conclusions and remarks.

2 Model Predictive Path-Following Control

Consider a continuous time nonlinear system, subject to input and state constraints:

$$\dot{x} = f(x,u), \quad x(0) = x_0, \tag{1}$$

where $x \in \mathscr{X} \subseteq \mathbb{R}^n, \quad u \in \mathscr{U} \subset \mathbb{R}^m$. Tracking of system (1) refers to the design of a controller such that the difference between the system state $x(t)$ and a time-varying reference signal $r(t)$ vanishes. Furthermore, it has to be guaranteed that the state $x(t)$ is in the state constraint set $\mathscr{X} \subseteq \mathbb{R}^n$ and that the inputs $u(t)$ are taken out of the

set of admissible inputs $\mathscr{U} \subset \mathbb{R}^m$. The tracking problem is often defined in terms of the time-dependent tracking error:

$$e_T(t) = x(t) - r(t). \tag{2}$$

Usually, the time-dependent reference signal $r(t)$ is assumed to be generated by an exo-system. Hence, the tracking problem can be reformulated as a stabilization problem. Typical applications of tracking are synchronization tasks, movement of robots or tracking of optimal state trajectories, which have been calculated previously.

Path-following refers to a different problem. Instead of a reference trajectory, a parametrized reference $r(\Theta)$ is considered. This reference is called *path* and it is often given by a regular curve in the state space $\mathbb{R}^n$:

$$r(\Theta): \quad [\hat{\Theta}, 0] \subset \mathbb{R} \mapsto r(\Theta) \in \mathbb{R}^n, \quad r(0) = 0. \tag{3}$$

It should be noted that in this paper the path as stated by (3) is negatively parametrized and ends in the origin. It is assumed that $r(\Theta)$ is sufficiently often continuously differentiable with respect to the parameter Θ. This path formulation does not distinguish between finitely and infinitely long paths, since the real interval $[\hat{\Theta}, 0]$ can be mapped onto both. The considered system (1) is subject to constraints, hence the path has to fulfill the state constraints $r(\Theta) \in \mathscr{X}$ for all $\Theta \in [\hat{\Theta}, 0]$. A path is denoted as *regular* if it is a non-singular curve and for each state $x \in \mathscr{X}$ there exists a unique path parameter $\tilde{\Theta}$, such that the distance between the path point $r(\tilde{\Theta})$ and x is minimal. Combining (2) and (3) yields to the path-following error:

$$e_P(t) = x(t) - r(\Theta). \tag{4}$$

Comparing the definition of $e_P(t)$ to (2) reveals important differences between tracking and path-following. Trajectory tracking implies that the reference signal inheres an explicit requirement *when* to be *where* in the state space. This arises from the fact that $r(t)$ is a reference *trajectory*. In path-following these requirements are relaxed. In general, the path parameter $\Theta = \Theta(t)$ is time dependent, but since its time evolution $\dot{\Theta}$ is not given a priori, it has to be obtained in the controller. Therefore, the timing law $\dot{\Theta}$ serves as an additional degree of freedom in the design of path-following controllers.

Typical practical path-following problems are the car-parking problem, autonomous ship control, the control of CNC-machines or the control of batch crystallisation processes. In these and many other applications it is desireable to stay very close to a given path even if this implies slower movement.

Considered Path-Following Problem

In the frame of this work, the subsequently stated path-following problem is considered. Find a controller, such that the following is satisfied:

P1 Path convergence: The path-following error vanishes, $\lim_{t \to \infty} e_P(t) = 0$.

P2 Forward motion: The system moves forward in path direction. $\dot{\Theta}(t) > 0$ holds for all $t > 0$ and all $\Theta \in [\hat{\Theta}, 0)$.

P3 Constraint satisfaction: The state and input constraints $x \in \mathscr{X} \subseteq \mathbb{R}^n, \quad u \in \mathscr{U} \subset \mathbb{R}^m$ are fulfilled.

Proposed Control Strategy

Since a predictive control strategy is considered, predicted states and inputs are referred as $\bar{x}$ and $\bar{u}$. At the sampling instants $t_k = k\delta$, where δ is the constant sampling time, the cost functional to be minimized over the prediction horizon T_P is given by:

$$J(\bar{x}, \bar{u}, \Theta, v) \quad = \quad \int_{t_k}^{t_k+T_P} F(\bar{x}, \bar{u}, \Theta, v)\, d\tau \quad + \quad E(\bar{x}(t_k+T_P), \Theta(t_k+T_P)). \quad (5)$$

In contrast to standard NMPC approaches, this cost functional does not only depend on the predicted states and inputs ($\bar{x}$ and $\bar{u}$) but also on the path parameter Θ and a to be defined *path parameter input* v. Θ is regarded as an internal or virtual state, hence the end penalty E is a function of $\bar{x}(t_k+T_P)$ and $\Theta(t_k+T_P)$. Similarly, the stage cost F depends on the variables $(\bar{x}, \bar{u}, \Theta, v)$. Nonlinear model predictive path-following control can then be stated as the repeated solution of the following open-loop optimal control problem:

$$\underset{\bar{u}(\cdot),\, v(\cdot)}{\textit{minimize}}\; J(\bar{x}, \bar{u}, \Theta, v) \quad (6)$$

subject to the usual constraints

$$\dot{\bar{x}} = f(\bar{x}(\tau), \bar{u}(\tau)), \quad \bar{x}(t_k) = x(t_k), \quad (7a)$$

$$\forall \tau \in [t_k, t_k+T_P]: \quad \bar{x}(\tau) \in \mathscr{X},\ \bar{u}(\tau) \in \mathscr{U}, \quad (7b)$$

$$\bar{x}(t_k+T_P) \in \mathscr{E} \subseteq \mathscr{X} \subseteq \mathbb{R}^n. \quad (7c)$$

The constraint (7c) indicates that at the end of each prediction the predicted state $\bar{x}(t_k+T_P)$ has to be in the terminal region $\mathscr{E}$ and $\mathscr{E} \subseteq \mathscr{X}$ is a closed subset of $\mathbb{R}^n$. Additonal path-following constraints must also be respected:

$$\dot{\Theta} = g(\Theta(\tau), v(\tau)), \quad \Theta(t_k) = \underset{\Theta}{argmin} \|x(t_k) - r(\Theta)\|, \quad (8a)$$

$$\forall \tau \in [t_k, t_k+T_P]: \quad \Theta(\tau) \in [\hat{\Theta}, 0] \subset \mathbb{R},\ v(\tau) \in \mathscr{V} \subseteq \mathbb{R}. \quad (8b)$$

These additional constraints state that the path parameter evolution is given by $\dot{\Theta} = g(\Theta, v)$, where g is called *timing law* and $v \in \mathscr{V}$ is a virtual input which influences the path parameter evolution. To solve the open-loop control problem at any sampling instance t_k, it is necessary to calculate the path point closest to the system state $x(t_k)$, since it serves as initial condition for $\dot{\Theta} = g(\Theta, v)$, see (8a). This initial condition is calculated via an extra minimisation. To ensure that the system moves forward along the path, this minimisation is subject to the constraint $\Theta(t_k) > \Theta(t_{k-1})$ for all $\Theta(t_k) \in [\hat{\Theta}, 0)$. The timing law $g(\Theta, v)$ in (8a) provides

an additional degree of freedom in the controller design. This function has to be chosen such that $\dot{\Theta} = g(\Theta, v) > 0$ holds for all $\Theta \in [\hat{\Theta}, 0)$ and all $v \in \mathscr{V}$ (compare with requirement P2).

The solution to the optimal control problem defined by (5) – (8) leads to the optimal input trajectory $\bar{u}^\star(t, x(t_k))$ which defines the input to be applied:

$$u(t) = \bar{u}^\star(t, x(t_k)),\ t \in [t_k, t_k + \delta]. \tag{9}$$

It should be noted that the additional virtual input v and the path parameter Θ are internal variables of the controller. The MPFC algorithm, as defined by (5) – (9), is a modified NMPC scheme, which chooses the velocity on the path such that the system stays close to it. Metaphorically speaking, if the system is far away from the path, first approach the path and then try to follow it along. Considering this MPFC scheme path convergence is more important than speed. In general, the considered approach does not lead to the fastest feasible path evolution. In contrast to the tracking of previously calculated optimal reference trajectories (which might be time-optimal), the MPFC scheme will iteratively adjust the reference evolution such that good path convergence is achieved. This online adjustment can be used to compensate disturbances or model-plant mismatch.

Stability

Sufficient conditions to prove the stability of NMPC schemes have been discussed widely throughout literature, e. g. in [4, 5, 10]. Since the proposed scheme is a modified NMPC scheme with *expanded* states and inputs, the following usual assumptions are made [3]:

A1 $\mathscr{X} \subseteq \mathbb{R}^n$ contains the origin in its interior. $\mathscr{X}$ is closed and connected.

A2 $\mathscr{U} \subset \mathbb{R}^m$ is compact and the origin is contained in the interior of $\mathscr{U}$.

A3 $f : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$ is a continuous and locally Lipschitz vector field. Furthermore, $f(0,0) = 0$.

A4 For all initial conditions in the region of interest and any piecewise continuous input function $u(\cdot) : [0, T_P] \mapsto \mathscr{U}$, (1) has a unique continuous solution.

A5 The cost function $F : \mathscr{X} \times \mathscr{U} \times [\hat{\Theta}, 0] \times \mathscr{V} \mapsto \mathbb{R}$ is continuous and positive definite in the domain $\mathscr{X} \times \mathscr{U} \times [\hat{\Theta}, 0] \times \mathscr{V}$.

Subsequently it is shown that the following additional assumptions are required to ensure stability of the proposed MPFC scheme:

A6 The path $r(\Theta)$ is regular and negatively parametrized, such that $\forall \Theta \in [\hat{\Theta}, 0] : r(\Theta) \in \mathscr{X} \subseteq \mathbb{R}^n$ and $r(0) = 0$ hold.

A7 The timing law $g(\Theta, v)$ has equivalent properties as required for $f(x,u)$ in assumptions A3 and A4. Furthermore, $\forall\ v \in \mathscr{V}$ and $\forall\ \Theta \in [\hat{\Theta}, 0) : g(\Theta, v) > 0$, where $\mathscr{V} \subseteq R$ is compact and $0 \in \mathscr{V}$.

If these conditions are fulfilled, then the following theorem holds.

Theorem 1 (Stability of Model Predictive Path-Following Control). *Consider the path-following problem for (11) as given by P1–P3 and assume that assumptions A1–A7 are fulfilled. Suppose that:*

(i) The terminal region $\mathscr{E} \times [\hat{\Theta},\ 0]$ is closed and $\forall \Theta \in [\hat{\Theta}, 0] : r(\Theta) \in \mathscr{E}$. The terminal penalty $E(x,\Theta)$ is continuously differentiable and positive semidefinite. Furthermore $E(0,0) = 0$ holds.

(ii) $\forall (x,\Theta) \in \mathscr{E} \times [\hat{\Theta},\ 0]$ there exists a pair of admissible inputs $(u_{\mathscr{E}}, v_{\mathscr{E}}) \in \mathscr{U} \times \mathscr{V}$ such that

$$\nabla E(x,\Theta) \cdot \begin{pmatrix} f(x,u_{\mathscr{E}}) \\ g(\Theta, v_{\mathscr{E}}) \end{pmatrix} + F\left(x,\Theta,u_{\mathscr{E}},v_{\mathscr{E}}\right) \leq 0 \tag{10}$$

and the solutions of $\dot{x} = f(x,u_{\mathscr{E}})$ and $\dot{\Theta} = g(\Theta, v_{\mathscr{E}})$ starting at $(x,\Theta) \in \mathscr{E} \times [\hat{\Theta},0]$ stay in $\mathscr{E} \times [\hat{\Theta},0]$ for all times.

(iii) The NMPC open-loop optimal control problem is feasible for t_0.

Then, for the closed-loop system defined by (11), (5)–(9), the path-following error $e_P(t) = x(t) - r(\Theta)$ converges to zero for $t \to \infty$. Furthermore the region of attraction is given by the set of states for which the open-loop optimal control problem (5)–(8) has a feasible solution.

The stability conditions contained in Theorem 1 are very similar to wellknown conditions for standard sampled data NMPC and the approach to establish stablity is very similar to [5, 10]. Hence, only a concise draft of the proof is provided. Mainly it is shown how the MPFC problem, as given by (5)–(9), can be reformulated such that it is equal to a standard NMPC problem.

Proof. Consider the following coordinate changes:

$$\begin{aligned} y &= (x - r(\Theta), \Theta)^T, & y &\in \mathbb{R}^{n+1} \\ w &= (u,v)^T, & w &\in \mathbb{R}^{m+1}. \end{aligned} \tag{11}$$

The problem specified by (5) – (8) can be reformulated for the expanded state and input y and w. In the new coordinates the state to be stabilized is the origin $0 \in \mathbb{R}^{n+1}$. Hence, the problem is equivalent to a stabilization NMPC problem. Start by applying the sufficient conditions for nominal stability of NMPC as given inter alia by [3, 5, 10] to the NMPC problem in y,w-coordinates. This leads to straightforward conditions on the feasibility of the open-loop optimal control problem at t_0 (compare with (iii)) and the invariance of x and Θ in the terminal region $\mathscr{E} \times [\hat{\Theta}, 0]$ in (ii). With respect to the coordinate transformation (11), the sufficient stability condition in y and w can be reformulated in the orginal coordinates x, u, Θ and v. This approach directly yields the condition as given in (ii) of Theorem 1. ■

Theorem 1 implies that the definition of suitable terminal penalties is relaxed in the MPFC setup compared to a straightforward tracking via NMPC. Starting with $E(x,\Theta) = E_1(x) + E_2(\Theta)$, the degrees of freedom to choose the timing law $g(\Theta, v)$ and $E_2(\Theta)$ can be utilized to assure that (10) holds.

3 Simulation Results

To illustrate the performance properties of the method proposed, an autonomous mobil robot in a fixed coordinate frame is considered:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} u_1 \cdot \cos(x_3) \\ u_1 \cdot \sin(x_3) \\ u_2 \end{pmatrix}. \tag{12}$$

x_1 and x_2 refer to the vehicle position in the x_1-x_2 plane and x_3 denotes the yaw angle. u_1 is the velocity of the vehicle and u_2 is the time derivative of the vehicles steering angle. The inputs u_1, u_2 are subject to the contraints $(u_1, u_2)^T \in \mathscr{U} = ([0, 12.5], [-\frac{\pi}{4}, \frac{\pi}{4}])^T$. For the path-following case the timing law is chosen to be: $\dot{\Theta} = -\lambda\Theta + 0.6 - v, \quad \lambda = 10^{-3}, \ v \in \mathscr{V} = [0, 0.6]$. The cost function is given by $F(\cdot) = y^T Q y + w^T R w$, where y and w are taken from (11) and

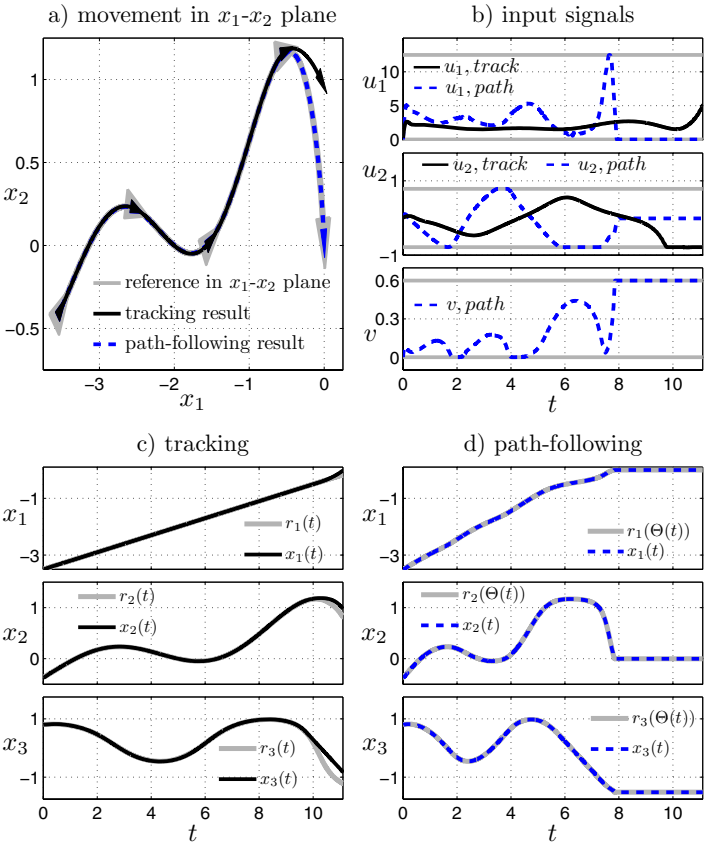


Fig. 1 Plot a) shows the motion of the vehicle in the x_1–x_2 plane for tracking and path-following. Plot b) depicts the corresponding input signals. Plots c) and d) show state and reference signals for tracking and path-following

$Q = diag(10^5, 10^5, 10^5, 10)$, $R = diag(10^{-1}, 10^{-2}, 10^3)$. In [6] a NMPC controller, which solves the tracking task for this system, was presented.

Plot a) of Figure 1 shows the movement of the vehicle (12) for tracking and path-following. While the projection of the reference trajectory into the x_1-x_2 plane is depicted by the thick grey line, the black line marks the vehicle movement when a NMPC tracking controller is applied. The dashed line shows the movement if the proposed scheme for predicitve path-following is considered. The triangles indicate the yaw angle at selected locations. In plot b) of Figure 1 the corresponding input signals for tracking (solid black lines), path-following (dashed lines) and the considered input constraints (solid grey lines) are shown.

The reference trajectory in the tracking case of Figure 1 is such that the approach used by [6] fails, since inputs constraints have to be violated to stay on the reference trajetory. Metaphorically speaking, the last turn of the reference, as depicted in plot a) of Figure 1, is too sharp to be realized by an admissible input in the tracking case. The time plots for all states of (12) are depicted in plots c) and d) of Figure 1. c) refers to the tracking case, d) shows the path-following results.

The MPFC results, as depicted by the dashed lines in plots a)-d) of Figure 1, show that the system under MPFC feedback is able to stay on the path. It needs to be pointed out that the MPFC controller requires less time to accomplished the whole path, therefore all states for path-following are stabilized at the final path point in plot d) of Figure 1. Nevertheless path-following slows down the path evolution in the turns to make the vehicle stay on the path.

The acceleration of the reference evolution in easy sections of the path – as it can be observed in plot d) of Figure 1 – is a direct consequence of the chosen reference value for the virtual input v. Not in every practical application this speed-up property might be desired. To avoide this, the reference evolution can be bounded by choosing the constant reference value for the path parameter input v such that the corresponding evolution $\dot{\Theta} = g(\Theta, v)$ matches a desired reference evolution.

4 Conclusions

It has been shown how the main idea of path-following can be implemented into a NMPC framework. Combining these approaches leads to a control scheme which computes the evolution of a reference signal and the input signals to follow this reference at the same time. Sufficient stability conditions for the proposed MPFC scheme have been presented. To investigate the performance of the method a simple vehicle has been considered as an example.

Future work will investigate the robustness of the proposed scheme as well as the robust design. In particular, the differences between the proposed approach and the tracking of offline calculated optimal trajectories will be discussed.

References

1. Aguiar, A.P., Hespanha, J.P., Kokotovic, P.V.: Path-following for nonminimum phase systems removes performance limitations. IEEE Trans. Automat. Contr. 50(2), 234–239 (2005)
2. Bemporad, A.: Reference governor for constrained nonlinear systems. IEEE Trans. Automat. Contr. 43(3), 415–419 (1998)
3. Findeisen, R.: Nonlinear Model Predictive Control: A Sampled-Data Feedback Perspective. Fortschr.-Ber. VDI Reihe 8 Nr. 1087, VDI Verlag (2006)
4. Findeisen, R., Allgöwer, F.: The quasi-infinite horizon approach to nonlinear model predictive control. In: Zinober, A., Owens, D. (eds.) Nonlinear and Adaptive Control. LNCIS series, pp. 89–105. Springer, Berlin (2001)
5. Fontes, F.: A general framework to design stabilizing nonlinear model predictive controllers. Sys. Contr. Lett. 42(2), 127–143 (2001)
6. Gu, D.B., Hu, H.S.: Receding horizon tracking control of wheeled mobile robots. IEEE Trans. Contr. Sys. Tech. 14(4), 743–749 (2006)
7. Limon, D., Alvarado, I., Alamo, T., Camacho, E.F.: MPC for tracking piecewise constant references for constrained linear systems. Automatica 44, 2382–2387 (2008)
8. Magni, L., de Nicolao, G., Scattolini, R.: Output feedback and tracking of nonlinear systems with model predictive control. Automatica 37(10), 1601–1607 (2001)
9. Magni, L., Scattolini, R.: Tracking of non-square nonlinear continuous time systems with piecewise constant model predictive control. J. Pro. Contr. 17(8), 631–640 (2007)
10. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.: Constrained model predictive control: Stability and optimality. Automatica 36(6), 789–814 (2000)
11. Michalska, H.: Trajectory tracking control using the receding horizon strategy. In: Symposium on Control, Optimization and Supervision: CESA 1996 IMACS (1996)
12. Qui, L., Davison, E.J.: Performance limitations of non-minimum phase systems in the servomechanism problem. Automatica 29(2), 337–349 (1993)
13. Rossiter, J.A., Kouvaritakis, B.: Reference governors and predictive control. In: Proc. American Control Conference 1998, vol. 6, pp. 3692–3693 (1998)
14. Seron, M., Braslavsky, J., Kokotovic, P.V., Mayne, D.Q.: Feedback limitations in nonlinear systems: from bode integrals to cheap control. IEEE Trans. Automat. Contr. 44(4), 829–833 (1999)
15. Skjetne, R., Fossen, T., Kokotovic, P.V.: Robust output maneuvering for a class of nonlinear systems. Automatica 40(3), 373–383 (2004)

A Survey on Explicit Model Predictive Control*

Alessandro Alessio and Alberto Bemporad

Abstract. Explicit model predictive control (MPC) addresses the problem of removing one of the main drawbacks of MPC, namely the need to solve a mathematical program on line to compute the control action. This computation prevents the application of MPC in several contexts, either because the computer technology needed to solve the optimization problem within the sampling time is too expensive or simply infeasible, or because the computer code implementing the numerical solver causes software certification concerns, especially in safety critical applications.

Explicit MPC allows one to solve the optimization problem off-line for a given range of operating conditions of interest. By exploiting multiparametric programming techniques, explicit MPC computes the optimal control action off line as an "explicit" function of the state and reference vectors, so that on-line operations reduce to a simple function evaluation. Such a function is piecewise affine in most cases, so that the MPC controller maps into a lookup table of linear gains.

In this paper we survey the main contributions on explicit MPC appeared in the scientific literature. After recalling the basic concepts and problem formulations of MPC, we review the main approaches to solve explicit MPC problems, including a novel and simple suboptimal practical approach to reduce the complexity of the explicit form. The paper concludes with some comments on future research directions.

Keywords: Model predictive control, explicit solutions, multiparametric programming, piecewise affine controllers, hybrid systems, min-max control.

1 Model Predictive Control

In Model Predictive Control (MPC) the control action is obtained by solving a finite horizon open-loop optimal control problem at each sampling instant. Each

Alessandro Alessio and Alberto Bemporad
Department of Information Engineering, University of Siena, Italy
e-mail: {alessio,bemporad}@dii.unisi.it

* This work was partially supported by the European Commission under the HYCON Network of Excellence, contract number FP6-IST-511368.

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 345–369.
springerlink.com

optimization yields a sequence of optimal control moves, but only the first move is applied to the process: At the next time step, the computation is repeated over a shifted time-horizon by taking the most recently available state information as the new initial condition of the optimal control problem. For this reason, MPC is also called *receding* or *rolling horizon control*.

The solution relies on a dynamic model of the process, respects all input and output (state) constraints, and optimizes a performance index. This is usually expressed as a quadratic or a linear criterion, so that, for linear prediction models, the resulting optimization problem can be cast as a quadratic program (QP) or linear program (LP), respectively, while for hybrid prediction models, the resulting optimization problem can be cast as a mixed integer quadratic or linear program (MIQP/MILP), as will be reviewed in the next sections. The main difference between MPC and conventional control is therefore that in the latter the control function is pre-computed off-line. The reason for the success of MPC in industrial applications is due to its ability to handle processes with many manipulated and controlled variables and constraints on them in a rather systematic manner.

The process to be controlled is usually modeled by the system of difference equations

$$x(t+1) = f(x(t), u(t)) \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, and $u(t) \in \mathbb{R}^m$ is the input vector. We assume for simplicity that $f(0,0) = 0$. The control and state sequences are requested to satisfy the constraints

$$x(t) \in \mathscr{X}, u(t) \in \mathscr{U} \tag{2}$$

where $\mathscr{U} \subseteq \mathbb{R}^m$ and $\mathscr{X} \subseteq \mathbb{R}^n$ are closed sets containing the origin in their interior[1]. Assuming that the control objective is to steer the state to the origin, MPC solves the constrained regulation problem as follows. Assume that a full measurement of the state $x(t)$ is available at the current time t. Then, the following finite-horizon optimal regulation problem is solved

$$\mathscr{P}_N(x(t)) : \min_z \quad \sum_{k=0}^{N-1} l(x_k, u_k) + F(x_N) \tag{3a}$$

$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k), \quad k = 0, \ldots, N-1 \tag{3b}$$

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

$$u_k \in \mathscr{U}, \quad k = 0, \ldots, N_u - 1 \tag{3d}$$

$$x_k \in \mathscr{X}, \quad k = 1, \ldots, N-1 \tag{3e}$$

$$x_N \in \mathscr{X}_N, \tag{3f}$$

$$u_k = \kappa(x_k), \quad k = N_u, \ldots, N-1 \tag{3g}$$

where $z \in \mathbb{R}^\ell$, is the vector of optimization variables, $z = [u_0' \ \ldots \ u_{N_u-1}']'$, $\ell = mN_u$ (more generally, z includes command inputs and additional optimization variables),

[1] Mixed constraints on (x,u) can be treated as well, for instance to handle constraints on outputs with direct feedthrough $y(t) = f_y(x(t), u(t))$.

and the choice of the closed set $\mathscr{X}_N \subseteq \mathscr{X}$, terminal cost F, and terminal gain κ ensure closed-loop stability of the MPC scheme [54]. At each time-step t, x_k denotes the predicted state vector at time $t+k$, obtained by applying the input sequence $u_0,..,u_{k-1}$ to model (1), starting from $x_0 = x(t)$. The number $N > 0$ is the prediction horizon, N_u is the input horizon ($1 \leq N_u \leq N$), and "$\leq$" denotes component-wise inequalities. Because N is finite, if f, l and F are continuous and $\mathscr{U}$ is also compact the minimum in (3a) exists. At each time-step t a solution to problem $\mathscr{P}_N(x(t))$ is found by solving the mathematical program

$$\begin{aligned} &\min_{z} \; h(z,x(t)) \\ &\text{s.t.} \;\; g(z,x(t)) \leq 0,\; g \in \mathbb{R}^q \end{aligned} \tag{4}$$

obtained from (3), yielding the optimal control sequence $z^*(x(t))$. Only the first input is applied to system (1)

$$u(t) = u_0^*(x(t)) \tag{5}$$

and the optimization problem (3) is repeated at time $t+1$, based on the new state $x(t+1)$.

The basic MPC setup (3) can be specialized to different cases, depending on the prediction model, performance index, and terminal conditions used.

1.1 Linear Model and Quadratic Cost

A finite-horizon optimal control problem (3) with quadratic stage costs is formulated by setting

$$l(x_k,u_k) = x_k'Qx_k + u_k'Ru_k,\; F(x_N) = x_N'Px_N \tag{6}$$

in (3a), where $Q = Q' \geq 0$, $R = R' > 0$, and $P = P' \geq 0$ are weight matrices of appropriate dimensions. Let (3b) be a deterministic linear discrete-time prediction model

$$f(x_k,u_k) = Ax_k + Bu_k \tag{7}$$

$\kappa(x) = Kx$ in (3g), $\mathscr{U}$, $\mathscr{X}$ be polyhedral sets, for example $\mathscr{U} = \{u \in \mathbb{R}^m :\; u_{\min} \leq u \leq u_{\max},\; \mathscr{X} = \{x \in \mathbb{R}^n :\; x_{\min} \leq x \leq x_{\max}\}$, and also $\mathscr{X}_N$ be polyhedral. Then, by substituting $x_k = A^k x(t) + \sum_{j=0}^{k-1} A^j B u_{k-1-j}$, problem (4) becomes a quadratic program (QP):

$$h(z,x(t)) = \frac{1}{2}z'Hz + x'(t)C'z + \frac{1}{2}x'(t)Yx(t) \tag{8a}$$

$$g(z,x(t)) = Gz - W - Sx(t), \tag{8b}$$

where $H = H' > 0$ and C, Y, G, W, S are matrices of appropriate dimensions [18]. Note that Y is not really needed to compute $z^*(x(t))$, it only affects the optimal value of (8a).

1.2 Linear Model and Linear Cost

Let ∞- or 1-norms be used to measure performance

$$l(x_k,u_k) = \|Qx_k\|_p + \|Ru_k\|_p,\ F(x_N) = \|Px_N\|_p,\ p = 1,\infty \tag{9}$$

where $R \in \mathbb{R}^{n_R \times m}$, $Q \in \mathbb{R}^{n_Q \times n}$, $P \in \mathbb{R}^{n_P \times n}$, and use the same setup as described in Section 1.1. In case of ∞-norms, by introducing auxiliary variables $\varepsilon_0^u, \ldots, \varepsilon_{N-1}^u, \varepsilon_1^x, \ldots, \varepsilon_N^x$ satisfying $\varepsilon_k^u \geq \|Ru_k\|_\infty$ $(k = 0,\ldots,N_u-1)$, $\varepsilon_k^x \geq \|Qx_k\|_\infty$ $(k = 1,\ldots,N-1)$, $\varepsilon_N^x \geq \|Px_N\|_\infty$, or, equivalently,

$$\begin{aligned} &\varepsilon_k^u \geq \pm R^i u_k,\ i = 1,\ldots,n_R,\ k = 0,\ldots,N_u-1 \\ &\varepsilon_k^x \geq \pm Q^i x_k,\ i = 1,\ldots,n_Q,\ k = 1,\ldots,N-1 \\ &\varepsilon_N^x \geq \pm P^i x_N,\ i = 1,\ldots,n_P \end{aligned} \tag{10}$$

where the superscript i in (10) denotes the ith row, problem (3) can be mapped into the linear program (LP) [12]

$$h(z,x(t)) = [\overbrace{1\ \ldots\ 1}^{N_u-1}\ \alpha\ \overbrace{1\ \ldots\ 1}^{N}\ \overbrace{0\ \ldots\ 0}^{mN_u}]z \tag{11a}$$

$$g(z,x(t)) = Gz - W - Sx(t), \tag{11b}$$

where $\alpha = N - N_u + 1$ is the number of times the last input move is repeated over the prediction horizon, $z \triangleq [\varepsilon_0^u\ \ldots\ \varepsilon_{N_u-1}^u\ \varepsilon_1^x\ \ldots\ \varepsilon_N^x\ u_0'\ \ldots\ u_{N_u-1}']'$ is the optimization vector, and G, W, S are obtained from weights Q, R, P, model matrices A, B, (10), constraint sets $\mathscr{U}$, $\mathscr{X}$, $\mathscr{X}_N$, and gain K. The case of 1-norms can be treated similarly by introducing slack variables $\varepsilon_{ik}^u \geq \pm R^i u_k$, $\varepsilon_{ik}^x \geq \pm Q^i x_k$, $\varepsilon_{iN}^x \geq \pm P^i x_N$.

Note that the above reformulation extends beyond 1/∞-norms to *any* convex piecewise affine cost l, F, that, thanks to the result of [62], can be rewritten as the max of a finite set of affine functions. The use of linear programming in optimization-based control dates back to the early sixties [60].

1.3 Linear Uncertain Model and Min-Max Costs

Robust MPC formulations [73] explicitly take into account uncertainties in the prediction model

$$f(x_k,u_k,w_k,v_k) = A(w_k)x_k + B(w_k)u_k + Ev_k \tag{12}$$

where

$$A(w) = A_0 + \sum_{i=1}^{q} A_i w_i, \quad B(w) = B_0 + \sum_{i=1}^{q} B_i w_i$$
$$w_k \in \mathscr{W} \in \mathbb{R}^{n_w}, \quad v_k \in \mathscr{V} \in \mathbb{R}^{n_v}$$

Let v_k, w_k be modeled as unknown but bounded exogenous disturbances and parametric uncertainties, respectively, and $\mathscr{W}$, $\mathscr{V}$ be polytopes. A robust MPC strategy often used consists of solving a *min-max* problem that minimize the worst-case

performance while enforcing input and state constraints for all possible disturbances. The following min-max control problem is referred to as *open-loop constrained robust optimal control problem* (OL-CROC) [13]

$$\begin{aligned} \min_{u_0,\ \ldots,\ u_{N-1}} & \left\{ \max_{\substack{v_0,\ldots,v_{N-1}\in\mathscr{V} \\ w_0,\ldots,w_{N-1}\in\mathscr{W}}} \sum_{k=0}^{N-1} l(x_k,u_k)+F(x_N) \right\} \\ \text{s.t.} \quad & \text{dynamics (12)} \\ & \text{(3d), (3e), (3f) satisfied } \forall v_k\in\mathscr{V},\ \forall w_k\in\mathscr{W} \end{aligned} \tag{13}$$

The min-max problem (12)–(13) can be solved via linear programming if 1- or ∞-norms are used [24, 4], or by quadratic programming if quadratic costs are used [56]. Being based on an *open-loop* prediction, in some cases the approach can be quite conservative. It is possible to reformulate the robust MPC problem using a *closed-loop* prediction scheme as described in [63], whose approach reminds the methods used in multi-stage stochastic optimization based on scenario trees. An alternative method in which the min and max problems are interleaved and dynamic programming is used is described in [13] to solve the (*closed-loop constrained robust optimal control problem*, CL-CROC):

For $j = N-1,\ldots,0$ solve

$$\begin{aligned} J_j^*(x_j) \triangleq & \min_{u_j\in\mathscr{U}} J_{\max,j}(x_j,u_j) \\ & \text{s.t. } A(w_j)x_j+B(w_j)u_j+Ev_j\in\mathscr{X}^{j+1},\ \forall v_j\in\mathscr{V},\ \forall w_j\in\mathscr{W} \end{aligned}$$

where

$$J_{\max,j}(x_j,u_j) \triangleq \max_{\substack{v_j\in\mathscr{V} \\ w_j\in\mathscr{W}}} \left\{ l(x_j,u_j)+J_{j+1}^*(A(w_j)x_j+B(w_j)u_j+Ev_j) \right\} \tag{14}$$

and

$$\mathscr{X}^j=\left\{x\in\mathscr{X}:\ \exists u\in\mathscr{U} \text{ such that } A(w)x+B(w)u+Ev\in\mathscr{X}^{j+1}, \forall v\in\mathscr{V}, w\in\mathscr{W}\right\} \tag{15}$$

with boundary condition

$$J_N^*(x_N)=\|Px_N\|_p. \tag{16}$$

1.4 Hybrid Model and Linear or Quadratic Costs

The MPC setup also extends to the case in which (1) is a hybrid dynamical model. When the hybrid dynamics and possible mixed linear/logical constraints on discrete and continuous input and state variables are modeled using the language HYSDEL [71], (3b) can be automatically transformed into the set of linear equalities and inequalities

$$f(x_k,u_k,\delta_k,\zeta_k) = Ax_k + B_1u_k + B_2\delta_k + B_3\zeta_k \tag{17a}$$
$$E_2\delta_k + E_3\zeta_k \leq E_1u_k + E_4x_k + E_5, \tag{17b}$$

involving both real and binary variables, denoted as the Mixed Logical Dynamical (MLD) model, where $x_k \in \mathbb{R}^{n_c} \times \{0,1\}^{n_b}$ is the state vector, $u_k \in \mathbb{R}^{m_c} \times \{0,1\}^{m_b}$ is the input vector, and $\zeta_k \in \mathbb{R}^{r_c}$, $\delta_k \in \{0,1\}^{r_b}$ are auxiliary variables implicitly defined by (17b) for any given pair (x_k,u_k). Matrices A, B_i, $(i=1,2,3)$, and E_i $(i=1,\ldots,5)$ denote real constant matrices, and inequalities (17b) must be interpreted component-wise.

The associated finite-horizon optimal control problem based on quadratic costs takes the form (8) with $z = [u_0' \cdots u_{N-1}'\ \delta_0' \cdots \delta_{N-1}'\ \zeta_0' \cdots \zeta_{N-1}']'$ and subject to the further restriction that some of the components of z must be binary. The hybrid MPC problem maps into a Mixed-Integer Quadratic Programming (MIQP) problem when the quadratic costs (6) are used in (3a) [17], or a Mixed-Integer Linear Programming (MILP) problem when ∞- or 1-norms are used as in (9) [11]. For both problems very efficient commercial and public domain solvers are available (see, e.g., http://plato.asu.edu/bench.html).

1.5 Extensions of the MPC Formulation

Tracking

The basic MPC regulation setup (3) can be extended to solve tracking problems. Given an output vector $y(t) = Cx(t) \in \mathbb{R}^p$ and a reference signal $r(t) \in \mathbb{R}^p$ to track under constraints (3d)–(3e), the cost function (3a) is replaced by

$$\sum_{k=0}^{N-1} (y_k - r(t))Q_y(y_k - r(t)) + \Delta u_k' R \Delta u_k \tag{18}$$

where $Q_y = Q_y' \geq 0 \in \mathbb{R}^{p\times p}$ is a matrix of output weights, and the increments of command variables $\Delta u_k \triangleq u_k - u_{k-1}$, $u_{-1} \triangleq u(t-1)$, are the new optimization variables, possibly further constrained by $\Delta u_{\min} \leq \Delta u_k \leq \Delta u_{\max}$. In the above tracking setup vector $[x'(t)\ r'(t)\ u'(t-1)]'$ replaces $x(t)$ in (4) and the control law becomes $u(t) = u(t-1) + \Delta u_0^*(x(t), r(t), u(t-1))$.

Rejection of Measured Disturbances

In order to take into account measured disturbances $v(t) \in \mathbb{R}^{n_v}$ entering the system, one can simply change (3b) into

$$f(x_k,u_k) = Ax_k + Bu_k + Ev(t) \tag{19}$$

where $v(t)$ is the most recent available measurement of the disturbance entering the process, and is supposed constant over the prediction horizon. In this case the extended vector $[x'(t)\ v'(t)]$ enters problem (4).

Soft Constraints and Time-Varying Constraints

Soft constraints $g(z,x(t)) \leq \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \varepsilon$, $\varepsilon \geq 0$, are also easily taken into account by adding a large penalty on ε or ε^2 in $h(z,x(t))$ in (4) and by including the new optimization variable ε in vector z.

Time-varying constraints $g(z,x(t)) \leq \gamma(t)$, $\gamma(t) \in \mathbb{R}^q$, $t = 0,1,\ldots$, can be also immediately taken into account, in this case the optimization problem (4) depends on both $x(t)$ and $\gamma(t)$.

2 Explicit Model Predictive Control

The MPC strategies described in Section 1 require running on-line optimization algorithms to solve the optimization problem (4) at each time-step t, based on the value of the current state vector $x(t)$ (or, more generally, based on $x(t)$, $r(t)$, $u(t-1)$, $v(t)$, $\gamma(t)$). For this reason, MPC has been traditionally labeled as a technology for slow processes. Advances in microcontroller and computer technology are progressively changing the concept of "slow", but still solving (4) on line prevents the application of MPC in many contexts, even for the simplest case of QP or LP. Computation speed is not the only limitation: the code implementing the solver might generate concerns due to software certification issues, a problem which is particularly acute in safety critical applications.

The idea of explicit MPC is to solve the optimization problem (4) off-line for all $x(t)$ within a given set X, that we assume here polytopic

$$X = \{x \in \mathbb{R}^n : \; S_1 x \leq S_2\} \subset \mathbb{R}^n \tag{20}$$

and to make the dependence of $u(t)$ on $x(t)$ *explicit*, rather than *implicitly* defined by the optimization procedure that solves problem (4). It turns out that, as intuited in [74], such a dependence is piecewise affine in most of the formulations seen in Section 1, so that the MPC controller defined by (3), (5) can be represented in a totally equivalent way as

$$u(x) = \begin{cases} F_1 x + g_1 \text{ if } H_1 x \leq k_1 \\ \quad \vdots \quad \vdots \\ F_M x + g_M \text{ if } H_M x \leq k_M. \end{cases} \tag{21}$$

Consequently, on-line computations are reduced to the simple evaluation of (21), which broadens the scope of applicability of MPC to fast-sampling applications. Explicit solutions of the form (21) to MPC problems can be obtained by solving *multiparametric programs*.

2.1 Multiparametric Programming: General Formulation

Consider the following mathematical program

$$\mathscr{NP}(x): \min_{z} \quad f(z,x) \tag{22a}$$

$$\text{s.t.} \quad g(z,x) \leq 0 \tag{22b}$$

$$Az + Bx + d = 0 \tag{22c}$$

where $z \in \mathbb{R}^{\ell}$ collects the decision variables, $x \in \mathbb{R}^{n}$ is a vector of parameters, $f : \mathbb{R}^{\ell} \times \mathbb{R}^{n} \to \mathbb{R}$ is the objective function, $g : \mathbb{R}^{\ell} \times \mathbb{R}^{n} \to \mathbb{R}^{q}$, A is a $q_e \times \ell$ real matrix, B is a $q_e \times n$ real matrix, and $d \in \mathbb{R}^{q_e}$. Problem (22) is referred as a *multiparametric programming problem*. We are interested in characterizing the solution of problem (22) for a given polytopic set X of parameters. The solution of a multiparametric problem is a triple (V^*, Z^*, X_f), where (i) the *set of feasible parameters* X_f is the set of all $x \in X$ for which problem (22) admits a solution, $X_f = \{x \in X : \ \exists z \in \mathbb{R}^{\ell}, \ g(z,x) \leq 0, \ Az + Bx + d = 0\}$; (ii) The *value function* $V^* : X_f \to \mathbb{R}$ associates with every $x \in X_f$ the corresponding optimal value of (22); (iii) The *optimal set* $Z^* : X_f \to 2^{\mathbb{R}^{\ell}}$ associates to each parameter $x \in X_f$ the corresponding set of optimizers $Z^*(x) = \{z \in \mathbb{R}^{\ell} : \ f(z,x) = V^*(x)\}$ of problem (22); (iv) An *optimizer function* $z^* : X_f \to \mathbb{R}^{\ell}$ associates to each parameter $x \in X_f$ an optimizer $z \in Z^*(x)$ ($Z^*(x)$ is just a singleton if $\mathscr{NP}(x)$ is strictly convex).

Let z be a feasible point of (22) for a given parameter x. The *active constraints* are the constraints that fulfill (22b) at equality, while the remaining constraints are called *inactive constraints*. The *active set* $\mathscr{A}(z,x)$ is the set of indices of the active constraints, that is,

$$\mathscr{A}(z,x) \triangleq \{i \in \{1,..,q\} | g_i(z,x) = 0\}.$$

The *optimal active set* $\mathscr{A}^*(x)$ is the set of indices of the constraints that are active for all $z \in Z^*(x)$, for a given $x \in X$,

$$\mathscr{A}^*(x) \triangleq \{i | i \in \mathscr{A}(z,x), \ \forall z \in Z^*(x)\}.$$

Given an index set $\mathscr{A} \subseteq \{1,..,q\}$, the *critical region* $CR_{\mathscr{A}}$ is the set of parameters for which the optimal active set is equal to $\mathscr{A}$, that is

$$CR_{\mathscr{A}} \triangleq \{x \in X | \mathscr{A}^*(x) = \mathscr{A}\}. \tag{23}$$

The following basic result for convex multiparametric programming was proved in [50] in the absence of equality constraints:

Lemma 1. *Consider the multiparametric problem (22) and let f, and the components g_i of g be jointly convex functions of (z,x), for all $i = 1,..,q$. Then, X_f is a convex set and V^* is a convex function of x.*

The result can be easily generalized to the presence of linear equality constraints. For generic nonlinear functions f, g_i, the critical regions $CR_{\mathscr{A}}$ and an optimizer

function z^* are not easily characterizable, and suboptimal methods have been proposed [38, 15]. Procedures for solving at optimality the multiparametric programs arising from the MPC problems introduced in Section 1 are described in the following sections.

2.2 Multiparametric Quadratic Programming

By treating $x(t)$ as the vector of parameters, the QP problem arising from the linear MPC formulation of Section 1.1 can be treated as the multiparametric QP (mpQP)

$$\mathcal{QP}(x): V^*(x) = \frac{1}{2}x'Yx + \min_z \quad \frac{1}{2}z'Hz + x'F'z \tag{24a}$$

$$\text{s.t.} \quad Gz \leq W + Sx. \tag{24b}$$

In [18], the authors investigated the analytical properties of the mpQP solution, that are summarized by the following theorem.

Theorem 1. *Consider a multiparametric quadratic program with* $H > 0$, $\left[\begin{smallmatrix} H & F \\ F' & Y \end{smallmatrix}\right] \geq 0$. *The set* X_f *of parameters* x *for which the problem is feasible is a polyhedral set, the value function* $V^* : X_f \mapsto \mathbb{R}$ *is continuous, convex, and piecewise quadratic, and the optimizer* $z^* : X_f \mapsto \mathbb{R}^\ell$ *is piecewise affine and continuous.*

The immediate corollary of the above theorem is that the linear MPC approach based on linear costs described in Section 1.2 admits a continuous piecewise-affine explicit solution of the form (21).

The arguments used to prove Theorem 1 rely on the first-order Karush-Kuhn-Tucker (KKT) optimality conditions for the mpQP problem (24)

$$Hz + Fx + G'\lambda = 0 \quad \lambda \in \mathbb{R}^q \tag{25a}$$

$$\lambda_i(G^i z - W^i - S^i x) = 0, \, i = 1, \ldots, q, \tag{25b}$$

$$\lambda \geq 0, \tag{25c}$$

$$Gz \leq W + Sx, \tag{25d}$$

where the superscript i denotes the ith row. Let us solve (25a) for z,

$$z = -H^{-1}(G'\lambda + Fx) \tag{26}$$

and substitute the result in (25b). Taken any point $x_0 \in X_f$ (easily determined by a linear feasibility test on $[G \; -S]\left[\begin{smallmatrix} z \\ x \end{smallmatrix}\right] \leq W$, for instance), solve $\mathcal{QP}(x_0)$ and determine the optimal active set $\mathcal{A}_0 = \mathcal{A}^*(x_0)$. Let $\hat{\lambda}$ and $\tilde{\lambda}$ denote the Lagrange multipliers corresponding to inactive and active constraints, respectively, and assume that the rows of $\tilde{G}$ are linearly independent. For inactive constraints, set $\hat{\lambda}^*(x) = 0$. For active constraints, $-\tilde{G}H^{-1}\tilde{G}'\tilde{\lambda} - \tilde{W} - \tilde{S}x = 0$, and therefore set

$$\tilde{\lambda}^*(x) = -(\tilde{G}H^{-1}\tilde{G}')^{-1}(\tilde{W} + (\tilde{S} + \tilde{G}H^{-1}F)x), \tag{27}$$

where $\tilde{G}$, $\tilde{W}$, $\tilde{S}$ correspond to the set of active constraints, and $(\tilde{G}H^{-1}\tilde{G}')^{-1}$ exists because the rows of $\tilde{G}$ are linearly independent. Thus, $\lambda^*(x)$ is an affine function of x. By simply substituting $\lambda^*(x)$ into (26) it is easy to obtain

$$z^*(x) = H^{-1}\tilde{G}'(\tilde{G}H^{-1}\tilde{G}')^{-1}(\tilde{W} + (\tilde{S} + \tilde{G}H^{-1}F)x) \tag{28}$$

and note that z^* is also an affine function of x. Vector z^* in (28) must satisfy the constraints in (25d), and by (25c), the Lagrange multipliers in (27) must remain non-negative. The set of inequalities defining the critical region $CR_{\mathscr{A}_0}$ in the x-space is hence given by the polyhedron

$$CR_{\mathscr{A}_0} = \{x \in \mathbb{R}^n : \ \hat{G}z^*(x) \leq \hat{W} + \hat{S}x, \ \tilde{\lambda}^*(x) \geq 0\}. \tag{29}$$

The critical region $CR_{\mathscr{A}_0}$ is the largest set of parameters for which the fixed combination of constraints $\mathscr{A}_0$ is the optimal active set and for which $z^*(x)$ is the optimizer function, because it satisfies all the KKT conditions (25) together with the dual solution $\lambda^*(x)$.

Different algorithms proposed to complete the characterization of the optimal solution $z^*(x)$ on the remaining set $CR^{rest} = X \setminus CR_{\mathscr{A}_0}$ are reviewed below.

In [18] the authors provided an algorithm for exploring CR^{rest} in order to generate all possible critical regions. The method proceeds recursively by (*i*) partitioning the set $X \subseteq \mathbb{R}^n$ into a finite number of polyhedra $\{CR_{\mathscr{A}_0}, R_1, \ldots, R_N\}$, where $R_i = \{x \in X : \ A^i x > B^i, \ A^j x \leq B^j, \ \forall j < i\}$ and A, B define a minimal representation $Ax \leq B$ of $CR_{\mathscr{A}_0}$, (*ii*) computing a new critical region CR_i in each region R_i through the KKT conditions (25), partitioning $R_i \setminus CR_i$, and so on. The procedure terminates when no more new optimal combinations of active constraints are found. Note that the number of critical regions for an mpQP problem is upper-bounded by the number 2^q of possible combinations of active constraints. Such an upper-bound is usually very conservative, as most of the combinations are never optimal for any $x \in X$.

A faster method is suggested in [68]. The active set of a neighboring region is found by using the active set of the current region and the knowledge of the *type* of the hyperplane that is crossed. That is, if an hyperplane of type $\hat{G}^i z^*(x) \leq \hat{W}^i + \hat{S}^i x$ is crossed, then the corresponding constraint is added to the active set, while if an hyperplane of type $\tilde{\lambda}_i^*(x) \geq 0$ is crossed, the corresponding constraint is withdrawn from the active set.

The approach described in [5] is based on the idea that neighboring regions of a given region CR can be determined by stepping out from each facet of CR by a very small distance[2], in order to determine new parameter vectors x around which to blow a new critical region.

The approaches of [68, 5] implicitly rely on the assumption that the intersection of the closures of two adjacent critical regions is a facet of both closures, a condition which is referred to as the *facet-to-facet* property. Conditions for the property to hold are provided in [68]. The property is usually satisfied, however an example of

[2] This is usually a fixed tolerance. To be exact, a simple mono-parametric QP should be partially solved over a line stemming out of the facet, to avoid missing narrow critical regions.

non-degenerate mpQP where the facet-to-facet property does not hold is shown in [65], where the authors suggest to combine the algorithms of [68, 18] to overcome the issue.

Once the optimizer function z^* and X_f are found, the explicit MPC controller is immediately obtained by saving the component u_0^* of z^*. The regions where $u_0^*(x)$ is the same can be combined, provided that their union is polyhedral [16].

An alternative approach was taken in [64], where the authors derive a closed-form expression for the global, analytical solution to the MPC problem for linear, time-invariant, discrete-time models with a quadratic performance index and magnitude constraints on the input. The approach exploits the geometric properties of constrained MPC to obtain global analytic solutions which can be precomputed off-line. A further approach to determine explicit MPC solutions was suggested independently in [42], based on the idea of approximating the constrained LQR problem.

A *reverse transformation* procedure was introduced in [52] and later refined in [51] to solve the finite-horizon problem (3) with linear prediction model and quadratic costs using dynamic programming, by treating each stage of the dynamic programming recursion as a parametric piecewise quadratic programming problem.

In [53, 55] also suggest reverse transformation to solve mpQP problems arising from MPC formulations. In particular in [55] the authors presented an algorithm based on dynamic programming, which explores all the possible solution conditions in such a way that the combinatorial explosion of enumeration of active sets is avoided.

From a practical viewpoint, the approaches mostly used for solving explicit MPC based on mpQP are those of [68, 5].

Degeneracy in Multiparametric QP

Definition 1. *Given an active set $\mathscr{A}$, the linear independence constraint qualification (LICQ) property is said to hold if the set of active constraint gradients are linearly independent, i.e., the associated matrix $\tilde{G}$ has full row rank.*

When LICQ is violated, we refer to as *primal degeneracy*. In this case the solution $\tilde{\lambda}^*(x)$ may not be uniquely defined (instead, $z^*(x)$ is always unique if $H > 0$). We refer to *dual degeneracy* if the dual problem of (24) is primal degenerate. In this case the solution $z^*(x)$ may not be unique, so clearly dual degeneracy can only happen if $H \geq 0$, $\det H = 0$. The absence of primal degeneracy ensures the following property of the value function V^* [6]:

Theorem 2. *Assume that the multiparametric QP (24) is not primal degenerate. Then the value function V^* is of class $\mathscr{C}^1$.*

In [18] the authors suggest two ways to handle primal degeneracy. The first one is to compute a QR decomposition of $\tilde{G}$, recognize whether the associated critical region is full-dimensional, and in this case project away the Lagrange multipliers to get the critical region. The second way consists simply of collecting ℓ linearly independent constraints arbitrarily chosen, and proceed with the new reduced set, therefore avoiding the computation of projections. Due to the recursive nature of

the algorithm, the remaining other possible subsets of combinations of constraints leading to full-dimensional critical regions will be automatically explored later.

A different approach is suggested in [70] for extending the algorithm of [68] to handle both primal and dual degeneracy. In [43] the authors propose an algorithm for solving multiparametric linear complementarity (mpLC) problems defined by positive semidefinite matrices. The approach is rather elegant, as it covers in a unified way both (degenerate) mpQP and the multiparametric LP case, which is described in the next Section 2.3.

2.3 Multiparametric Linear Programming

By treating $x(t)$ as the vector of parameters, the linear MPC formulation of Section 1.2 can be treated as the multiparametric LP (mpLP)

$$\mathscr{LP}(x): \min_{z} \quad c'z \tag{30a}$$

$$\text{s.t.} \quad Gz \leq W + Sx, \tag{30b}$$

where $z \in \mathbb{R}^{\ell}$ is the optimization vector, $x \in X \subset \mathbb{R}^n$ is the vector of parameters, c, G, W, S are suitable constant matrices and X is the set of parameters of interest defined in (20). The following result is proved in [29].

Theorem 3. *Consider the mpLP problem (30). Then, the set X_f is a convex polyhedral set, there exists an optimizer function $z^* : \mathbb{R}^n \to \mathbb{R}^l$ which is a continuous and piecewise affine function of x, and the value function $V^* : \mathbb{R}^n \to \mathbb{R}$ is a continuous, convex, and piecewise affine function of x.*

The first methods for solving parametric linear programs appeared in 1952 in the master thesis published in [58], and independently in [32]. Since then, extensive research has been devoted to sensitivity and (multi)parametric analysis, see the references in [29] and also [30] for advances in the field. Gal and Nedoma provided the first method for solving multiparametric linear programs in 1972 [31]. Subsequently, until the beginning of the new millennium only a few authors [29, 28, 62] have dealt with multiparametric linear programming solvers. Subsequently, a renewed interest has arisen, mainly pushed by the application of mpLP in explicit MPC. For a recent survey on multiparametric linear programming the reader is referred to [44].

The KKT conditions related to the mpLP (30) are

$$Gz \leq W + Sx, \tag{31a}$$

$$G'\lambda = c, \tag{31b}$$

$$\lambda \geq 0, \tag{31c}$$

$$(G_i z - W_i - S_i x)\lambda_i = 0, \ \forall i. \tag{31d}$$

For a given fixed parameter $x_0 \in X$, by solving $\mathscr{LP}(x_0)$ one finds a solution z_0^*, λ_0^* and defines an optimal set of active constraints $\mathscr{A}(x_0)$. By forming submatrices $\tilde{G}$, $\tilde{W}$, $\tilde{S}$ of active constraints and $\hat{G}$, $\hat{W}$, $\hat{S}$ of inactive constraints, as in the mpQP case, and by assuming $\tilde{G}$ square and full rank, one gets that

$$z^*(x) = (\tilde{G}^{-1}\tilde{S})x + (\tilde{G}^{-1}\tilde{W}), \tag{32}$$

is an affine function of x, $\lambda^*(x) \equiv \lambda^*(x_0)$ is constant, and the value function

$$V^*(x) = (W + Sx)'\lambda^*(x_0), \tag{33}$$

is also affine. Eqs. (32)–(33) provide the solution to (30) for all $x \in X$ such that

$$\hat{G}((\tilde{G}^{-1}\tilde{S})x + (\tilde{G}^{-1}\tilde{W})) < \hat{W} + \hat{S}x, \tag{34}$$

which defines the polyhedral *critical region* $CR_{\mathscr{A}_0}$ of all x for which the chosen combination of active constraints $\mathscr{A}(x_0)$ is the optimal one.

In [31] a critical region is defined as the set of all parameters x for which a certain basis is optimal for problem (30). The algorithm proposed in [31] for solving mpLPs generates non-overlapping critical regions by generating and exploring the graph of bases. In the graph of bases, the nodes represent optimal bases of the given multiparametric problem and two nodes are connected by an edge if it is possible to pass from one basis to another by one pivot step (in this case, the bases are called *neighbors*).

In [22] the authors proposed a geometric algorithm based on the direct exploration of the parameter space as in [18]. The definition of critical regions in [22] is not associated with the bases but with the set of active constraints and is related directly to the definition given in [28, 2].

The approach of [5] is also easily applicable to solve multiparametric linear programs. The *facet-to-facet* property however is not always guaranteed to hold also in mpLP problems. Conditions for ensuring that the property holds are reported in [46] and depend on degeneracy situations.

Degeneracy in Multiparametric LP

An LP problem (30) is said to be *primal degenerate* for some $x \in X$ if there exists a $z*(x) \in Z^*(x)$ such that the number of active constraints at the optimizer larger than the number ℓ of optimization variables. In this case more than one basis describes the optimal primal solution. *Dual degeneracy* occurs when the dual problem of (30) is primal degenerate. In this case more than one primal solution $z^*(x)$ is optimal.

Dual degeneracy is undesired in linear MPC formulations based on linear costs as, depending on the on-line LP solver used, the solution may chatter from one optimizer to another, with consequent stress of the actuators without a real performance benefit.

Ways to handle degeneracy in mpLP are reported in [22] and in [46].

2.4 Explicit Solution of Min-Max MPC

In [13] the authors proposed an approach based on a combination of multiparametric linear programming and dynamic programming to obtain solutions to constrained robust optimal control problems in state feedback form. In particular, the solution

of *closed-loop* constrained robust optimal control problems is a piecewise affine function of the state, obtained by solving N mpLPs, while the solution z^* to *open-loop* constrained robust optimal control problems with parametric uncertainties in the B matrix only can be found by solving an mpLP. A different approach was taken in [47] to solve the CL-CROC formulation of [63] through a single (but larger) mpLP.

For explicit MPC based on linear parameter-varying (LPV) models one could in principle embed the problem into the robust min-max formulation of [13, 47]. However, this would unavoidably lead to a loss of performance. The ability to measure the parameter information is exploited in [19] to derive an explicit LPV-MPC approach also based on dynamic programming and mpLP.

For min-max problems based on quadratic costs, the solution was proved to be piecewise affine in [61], where the authors also showed that the critical regions were defined not only by a set of active constraints, but also by a set of active vertices. In [56] the problem was reformulated as a single mpQP.

2.5 Explicit Solution of Hybrid MPC

As detailed in [11], the MPC formulation based on ∞- or 1-norms subject to the MLD dynamics (17) can be solved explicitly by treating the optimization problem associated with MPC as a multiparametric mixed integer linear programming (mpMILP) problem.

An algorithm for solving mpMILPs was proposed in [1], based on a branch and bound procedure in which an mpLP problem is solved at each node of the tree. In [27] an alternative algorithm was proposed, which only solves mpLPs where the integer variables are fixed to the optimal value determined by an MILP, instead of solving mpLP problems with relaxed integer variables. More in detail, the mpMILP problem is alternatively decomposed into an mpLP and an MILP subproblem. It is easy to show that the explicit hybrid MPC controller has the form (21), but the control law $u(x)$ may be discontinuous across neighboring polyhedra [20].

It turns out that mpMILP is not very efficient in practice for solving hybrid MPC problems based on linear costs explicitly. In addition, good algorithms for multiparametric mixed integer quadratic programming (mpMIQP) problems for dealing with quadratic costs have not yet been proposed in the literature. Better ways to handle hybrid MPC problems explicitly in the case of linear and quadratic costs use piecewise affine (PWA) hybrid models. PWA models are defined by partitioning the space of states and inputs into polyhedral regions and associating with each region a different linear state-update equation

$$f(x_k,u_k) = A_i x_k + B_i u_k + f_i \text{ if } \begin{bmatrix} x_k \\ u_k \end{bmatrix} \in \mathscr{X}_i, \tag{35}$$

where $\{\mathscr{X}_i\}_{i=1}^s$ is a polyhedral partition of the state+input set. Hybrid systems of the form (35) can be obtained for instance by converting HYSDEL/MLD models using the method of [8] or [34].

In [20] the authors propose an algorithm for computing the solution to problem (3) with linear or quadratic costs and PWA models. The procedure is based on dynamic programming (DP) iterations. Multiparametric QPs are solved at each iteration, and quadratic value functions are compared to possibly eliminate regions that are proved to never be optimal. In typical situations the total number of solved mpQPs (as well as of generated polyhedral regions) grows exponentially with the prediction horizon N, and may suffer the drawback of an excessive partitioning of the state space. Nonetheless, the approach is preferable to gridding approaches usually employed to solve DP problems, especially in the case of nontrivial state-space dimensions.

A different approach was proposed in [52, 53], where the authors propose a reverse transformation procedure to enumerate all possible switching sequences, and for each sequence convert the PWA dynamics into a time-varying system and solve an optimal control problem explicitly via mpQP. In [3] the authors propose a related approach that exploits dynamic programming ideas (more precisely, backwards reachability analysis) to obtain all the feasible mode sequences (therefore avoiding an explicit enumeration of all of them), and that, after solving an mpQP for each sequence, post-processes the resulting polyhedral partitions to eliminate all the regions (and their associated control gains) that never provide the lowest cost, using a DC (Difference of Convex functions) algorithm. A similar algorithm is also used for explicit hybrid MPC based linear costs in the Hybrid Toolbox for Matlab [9].

The partition associated with the fully explicit optimal solution to problem (3) in the hybrid system case may not be polyhedral [20]. In this case, most explicit hybrid MPC algorithms for quadratic costs keep possible overlapping critical regions (generated by different mpQPs) and also store either the value function V^* or the full optimizer z^*, so that the optimal MPC move can be determined on-line by comparing the optimal values of all regions (in most cases just one, or a only a few) where $x(t)$ belongs.

Finally, we mention that for the special class of hybrid models given by linear systems with quantized inputs $u(t)$ problem (4) maps into a multiparametric nonlinear integer programming problem, for which a solution algorithm was provided in [7].

3 Reducing Complexity of Explicit MPC

The complexity of the piecewise affine explicit MPC law can be easily assessed after the MPC optimization problem is pre-solved off-line via the multiparametric techniques described in the previous sections. For a fixed number of parameters (states and reference signals), the complexity of the solution is given by the number M of regions that form the explicit solution (21). The number M mainly depends (exponentially, in the worst case) on the number q of constraints (and also of binary variables/system modes in the hybrid case) included in the MPC problem formulation (3), and only mildly on the number n of states. It also depends on the number ℓ of optimization variables, but mainly because q depends on ℓ (consider for instance

input constraints). In [10, Table I], a table shows such dependencies on random linear MPC problems. In the multiparametric QP case, an upper-bound to M is 2^q, which is the number of all possible combination of active constraints at optimality.

At the price of a possible loss of closed-loop performance, one way of reducing complexity is to shorten the prediction horizons (and/or blocking input moves [67]) in order to decrease q and ℓ. In practice, explicit MPC is limited to relatively small problems (typically one/two inputs, up to five-ten states, up to three/four free control moves) but allows one to reach very high sampling frequencies (up to 1 MHz) and requires a very simple control code to be embedded in the system.

To address such limitations, several attempts have been made in different directions. Starting with a given piecewise affine solution, in [33] the authors provide an approach to reduce the number of partitions by optimally merging regions where the affine gain is the same, so that the original solution is maintained but equivalently expressed with a minimal number of partitions. However, techniques for achieving a more drastic reduction of complexity require changing the solution, by accepting a certain level of suboptimality with respect to the original problem formulation (3).

Suboptimal mpQP solutions were proposed in [14] by relaxing the KKT conditions, such as nonnegativity of optimal dual variables. For explicit MPC controllers based on linear costs, a quite efficient "beneath/beyond" approximate mpLP algorithm has been proposed in [44]. In [40, 39] the authors propose recursive rectangular partitions of the parameter space to determine a suboptimal solution to general classes of multiparametric programming problems. Simplicial recursive partitions are instead proposed in [15] for general classes of nonlinear multiparametric programs that are jointly convex with respect to optimization variables and parameters, used in [57] for solving robust MPC formulations explicitly in approximate form. Rather than looking directly at the multiparametric programming problem, in [36] the authors propose to change the MPC formulation by recursively solving a sequence of simpler explicit MPC problems with horizon $N = 1$, where the terminal set $\mathscr{X}_N$ is varied at each iteration and obtained from the previous iteration. A related approach is given in [35] for PWA systems, which leads to minimum-time controllers. Based on a dynamic programming formulation of the finite-horizon optimal control problem, in [49] the authors propose an approach to relax optimality within a prescribed bound in favor of the reduced complexity of the solution, which in the case of linear systems and piecewise affine convex costs leads to another form of computing approximate mpLP solutions. In the hybrid case, suboptimal solutions can be simply obtained by constraining the sequence of switching modes a priori, like for example has been proposed in [37], and then applying any of the solution methods proposed in Section 2.5.

More recently, authors have realized that a different way of obtaining suboptimal explicit MPC solutions is to avoid storing a full partition, by keeping only a subset of critical regions. For treating large MPC problems, in [59] the authors suggest the partial enumeration of the possible combinations of active constraints at optimality, so that the solution can be searched on line (explicitly or implicitly) within that small subset. An off-line training session of simulations is suggested to identify the subset of most important combinations of active constraints. In [26], the authors also

suggest to remove combinations of active constraints repeatedly, as long as closed-loop stability of the suboptimal solution is preserved. The drawback of the approach is that it need to start from the full exact solution, which may be very expensive to compute. A different novel approach which has analogies with both [59] and [26] is described below.

3.1 A Novel Practical Approach for Reducing Complexity of Explicit MPC

Under the assumption that only input constraints and soft state constraints can be enforced, given a maximum number L of regions accepted in the suboptimal solution, the proposed approach can be summarized as follows:

Off-line computations

1. Run extensive simulations of closed-loop MPC (using on-line QP evaluations) from several initial conditions and for several references and disturbances, as in [59]. Collect the L most recurrent combinations of active constraints $I_1, I_2, \ldots, I_L$ at optimality. Alternatively, if a full explicit solution is available, pick up the L regions with largest Chebychev radius.
2. Generate the corresponding critical regions $CR_1, CR_2, \ldots, CR_L$, where $CR_i = \{x : H_i x \leq K_i\}$.

On-line computations

Let $x = x(t)$ be the current state (or, more generally, the current parameter vector including also $r(t)$, $u(t-1)$, $v(t)$, $\gamma(t)$).

1. Compute $\beta_i(x) = \max_j \{H_i^j x - K_i^j\}$, that is the maximum violation of a facet inequality[3], for all regions $j = 1, \ldots, L$;
2. Let $h \in \{1, \ldots, L\}$ such that $\beta_h(x) = \min_{i=1,\ldots,L} \beta_i(x)$;
3. If $\beta_h(x) \leq 0$, then $x \in CR_h$ and $u(x) = F_h x + g_h$; stop;
4. Otherwise all $\beta_i(x) > 0$, and either compute the average

$$\bar{u}(x) = \left(\sum_{i=1,\ldots,L} \frac{1}{\beta_i(x)} \right)^{-1} \sum_{i=1,\ldots,L} \frac{1}{\beta_i(x)} (F_i x + g_i) \tag{36a}$$

or extrapolate the solution from the critical region CR_h corresponding to the least violation $\beta_h(x)$

$$\bar{u}(x) = F_h x + g_h \tag{36b}$$

[3] Alternatively, one can compute the largest distance of x from a violated facet half-space $\beta_i(x) = \max_j \{\frac{H_i^j x - K_i^j}{\sqrt{H_i^j (H_i^j)'}}\}$.

5. Create a suboptimal input move by saturating $u(x) = \text{sat}\,\bar{u}(x)$, where sat is a suitable saturation function (for instance, the standard saturation function in case $\mathscr{U}$ is the box $\{u \in \mathbb{R}^m : u_{\min} \le u \le u_{\max}\}$ and $\mathscr{X} = \mathbb{R}^n$).

Note that the procedure is very general, as it can be applied to any of the MPC formulations described in Section 1, and in particular to linear and hybrid MPC with linear or quadratic costs[4]. While stability properties of the suboptimal solution can only be checked a posteriori, contrary to [59] the number of regions is bounded a priori, and off-line computations are much simpler, as numerical manipulations over the exact full partition are not required. Note that for a given reduced partition and a set X of parameters of interest as defined in (20), the largest minimal violation $\beta_{\max} = \max_{x\in X} \beta_h(x)$ can be computed by solving the mixed-integer linear programming problem

$$\begin{aligned} \beta_{\max} = \max_{x,\varepsilon,\{\delta_i^j\}} \ & \varepsilon \\ \text{s.t.} \quad & [\delta_i^j = 1] \leftrightarrow [H_i^j x \le K_i^j + \mathbf{1}\varepsilon],\ \forall j = 1,\dots,n_i,\ \forall i = 1,\dots,L \\ & \textstyle\sum_{j=1}^{n_i} \delta_i^j \le n_i - 1 \\ & \delta_i \in \{0,1\}^{n_i},\ \forall i = 1,\dots,L \\ & S_1 x \le S_2 \end{aligned} \tag{37}$$

where $n_i = \dim(K_i)$ is the number of inequalities defining the ith critical region, superscript ^{j} denotes the jth row or component, and $\mathbf{1}$ is a vector of all ones. Note that, as the aim is to have L very small, the number $\sum_{i=1}^{L} n_i$ of binary variables in the MILP (37) is also small. The quantity $\beta_{\max}$ can be used as a tuning parameter to decide the number L of regions to include in the suboptimal explicit MPC controller.

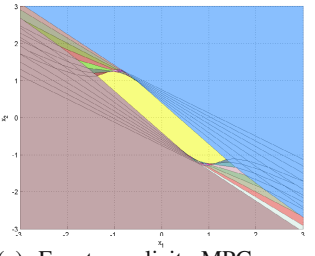
An Example

Consider the double integrator example described in [18, Example 7.3] with $N = N_u = 8$, but with the different sampling time $T_s = 0.1$. The exact explicit solution, computed in about 3 s on a Laptop PC Intel Core Duo 1.2 GHz using the Hybrid Toolbox for Matlab [9], consists of $M = 99$ regions and is depicted in Figure 1(a).

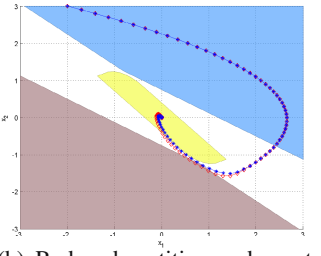
The solution with $L = 3$ regions reported in Figure 1(b), corresponds to $\varepsilon = 1.4661$ in (37) and leads to the closed loop trajectories of Figure 1(c), obtained by averaging as in (36a).

For the hybrid explicit MPC controller described in [23, Figure 6b], which consists of 671 regions, experiments have shown that, after simulation training, the suboptimal approach described above with $L = 60$ regions leads to almost indistinguishable closed-loop MPC trajectories.

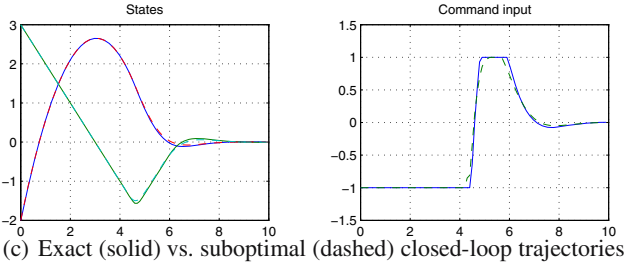
[4] In case of hybrid MPC based on quadratic costs, polyhedral critical regions may be overlapping. In this case, the explicit form of the cost function $V(x)$ on those regions must be also taken into account, in order to define a value for $\bar{u}(x)$ that penalizes both a large $\beta(x)$ and a large cost $V(x)$.



(a) Exact explicit MPC controller



(b) Reduced partition and exact ($\diamondsuit$) vs. suboptimal ($*$) closed-loop state trajectories



(c) Exact (solid) vs. suboptimal (dashed) closed-loop trajectories

Fig. 1 Exact and suboptimal explicit MPC controller ($L = 3$)

4 Implementation of Explicit MPC

The explicit MPC solution (21) pre-computed off-line is a lookup table of linear feedback gains. The right gain is selected on-line by finding the region $\{x : H_i x \leq k_i\}$ of the polyhedral partition where the current state $x(t)$ (or, more generally, the current vector of parameters) lies. This problem has been referred to as *point-location problem*. The most simple solution is to store all the M polyhedra of the partition and on-line search through them until the right one is found. While this procedure is extremely easy to implement in a computer code, more efficient ways have been proposed for evaluating explicit MPC controllers, which become appealing when the number M of regions is large.

By exploiting the properties of multiparametric linear and quadratic solutions, in [6] two new algorithms are proposed that avoid storing the polyhedral regions, significantly reducing the on-line storage demands and computational complexity of the evaluation of control. In [69] the authors suggest to organize the hyperplanes defining the regions on a binary search tree (possibly further subdividing some of the regions), so that the time to locate the state vector on-line within the partition becomes logarithmic in the number of stored cells and memory space is saved. For the multiparametric linear case, another method is given in [45], where additively weighted nearest neighbour search ideas are used to solve the point location problem logarithmically in the number N of regions. Assuming that a bound between the evolution of the real process and the prediction model is known, in [66] the authors

suggest to limit the point location problem only to those regions that are reachable in one time step from the currently active one. In [25] the authors propose a search tree algorithm based on bounding boxes and interval trees. A subdivision walking method is proposed in [72].

From the hardware synthesis viewpoint, [41] showed that explicit MPC solutions can be implemented in an application specific integrated circuit (ASIC) with about $20,000$ gates, leading to computation times in order of 1 μs.

Whether the explicit form (21) is preferable to the one based on on-line optimization depends on available CPU time, data memory, and program memory (see e.g. [10, Table II] for a comparison in the linear quadratic case).

5 Tools for Explicit MPC

The Hybrid Toolbox for Matlab [9] allows one to design explicit MPC control laws for linear and hybrid systems. Various functions for the design, visualization, simulation, and C-code generation of explicit MPC controllers are provided. Similar and other functionalities can be found in the Multi Parametric Toolbox for Matlab [48].

6 Explicit MPC in Applications

Industrial problems addressed through explicit MPC techniques have been reported in dozens of technical papers, starting from what is probably the first work in this domain [21]. The most suitable applications for explicit MPC are fast-sampling problems (in the order of 1-50 ms) and relatively small size (1-2 manipulated inputs, 5-10 parameters). Most of the applications of explicit MPC have been reported in the automotive domain and electrical power converters.

7 Conclusions

At the time of writing this survey, exact explicit characterizations of MPC for linear and hybrid systems have been quite well studied. Efficient algorithms exist to compute solutions to multiparametric linear and quadratic programming problems and explicit control laws, and Matlab toolboxes are available to let control engineers apply these ideas in practical applications. Nonetheless, the field is far from being mature. The clear limitation of exact explicit MPC solutions is that the number of regions composing the solution grows massively with the size of the problem (mainly with the number of constraints in the MPC problem).

Future research efforts should therefore pursue three main directions. First, new suboptimal methods that allow trading off between loss of closed-loop performance and number of partitions are needed; the whole concept of looking for optimal open-loop performance is actually weak, as the MPC cost function is usually chosen by trial and error. Second, PWA solutions are not necessarily the best way to define suboptimal MPC laws, other piecewise-nonlinear ways of defining the solution may

lead to more compact representations; in this respect, links to nonlinear approximation approaches could be sought to approximate samples of the MPC control profile computed through (4) within guaranteed error bounds. Third, semi-explicit methods should be also sought, in order to pre-process of line as much as possible of the MPC optimization problem without characterizing all possible optimization outcomes, but rather leaving some optimization operations on-line.

Acknowledgements. The authors thank Gabriele Pannocchia, Ilya Kolmanovsky, Davor Hrovat, and Stefano Di Cairano for discussing the ideas related to the novel practical approach for reducing complexity of explicit MPC described in this paper, and David Muñoz de la Peña, David Q. Mayne, and Ilya Kolmanovsky for suggesting improvements on the original manuscript.

References

1. Acevedo, J., Pistikopoulos, E.N.: A multiparametric programming approach for linear process engineering problems under uncertainty. Ind. Eng. Chem. Res. 36, 717–728 (1997)
2. Adler, I., Monteiro, R.D.C.: A geometric view of parametric linear programming. Algorithmica 8(2), 161–176 (1992)
3. Alessio, A., Bemporad, A.: Feasible mode enumeration and cost comparison for explicit quadratic model predictive control of hybrid systems. In: 2nd IFAC Conference on Analysis and Design of Hybrid Systems, Alghero, Italy, pp. 302–308 (2006)
4. Allwright, J.C., Papavasiliou, G.C.: On linear programming and robust model-predictive control using impulse-responses. Systems & Control Letters 18, 159–164 (1992)
5. Baotić, M.: An efficient algorithm for multi-parametric quadratic programming. Technical Report AUT02-05, Automatic Control Institute, ETH, Zurich, Switzerland (2002)
6. Baotić, M., Borrelli, F., Bemporad, A., Morari, M.: Efficient on-line computation of constrained optimal control. SIAM Journal on Control and Optimization 47(5), 2470–2489 (2008)
7. Bemporad, A.: Multiparametric nonlinear integer programming and explicit quantized optimal control. In: Proc. 42nd IEEE Conf. on Decision and Control, Maui, Hawaii, USA, pp. 3167–3172 (2003)
8. Bemporad, A.: Efficient conversion of mixed logical dynamical systems into an equivalent piecewise affine form. IEEE Trans. Automatic Control 49(5), 832–838 (2004)
9. Bemporad, A.: Hybrid Toolbox – User's Guide (January 2004), http://www.dii.unisi.it/hybrid/toolbox
10. Bemporad, A.: Model-based predictive control design: New trends and tools. In: Proc. 45th IEEE Conf. on Decision and Control, San Diego, CA, pp. 6678–6683 (2006)
11. Bemporad, A., Borrelli, F., Morari, M.: Piecewise linear optimal controllers for hybrid systems. In: American Control Conference, Chicago, IL, pp. 1190–1194 (June 2000)
12. Bemporad, A., Borrelli, F., Morari, M.: Model predictive control based on linear programming — The explicit solution. IEEE Trans. Automatic Control 47(12), 1974–1985 (2002)
13. Bemporad, A., Borrelli, F., Morari, M.: Min-max control of constrained uncertain discrete-time linear systems. IEEE Trans. Automatic Control 48(9), 1600–1606 (2003)

14. Bemporad, A., Filippi, C.: Suboptimal explicit receding horizon control via approximate multiparametric quadratic programming. Journal of Optimization Theory and Applications 117(1), 9–38 (2003)
15. Bemporad, A., Filippi, C.: An algorithm for approximate multiparametric convex programming. Computational Optimization and Applications 35(1), 87–108 (2006)
16. Bemporad, A., Fukuda, K., Torrisi, F.D.: Convexity recognition of the union of polyhedra. Computational Geometry: Theory and Applications 18, 141–154 (2001)
17. Bemporad, A., Morari, M.: Control of systems integrating logic, dynamics, and constraints. Automatica 35(3), 407–427 (1999)
18. Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N.: The explicit linear quadratic regulator for constrained systems. Automatica 38(1), 3–20 (2002)
19. Besselmann, T., Löfberg, J., Morari, M.: Explicit model predictive control for systems with linear parameter-varying state transition matrix. In: Proc. 17th IFAC World Congress, Seoul, Corea (2008)
20. Borrelli, F., Baotić, M., Bemporad, A., Morari, M.: Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. Automatica 41(10), 1709–1721 (2005)
21. Borrelli, F., Bemporad, A., Fodor, M., Hrovat, D.: A hybrid approach to traction control. In: Di Benedetto, M.D., Sangiovanni-Vincentelli, A.L. (eds.) HSCC 2001. LNCS, vol. 2034, pp. 162–174. Springer, Heidelberg (2001)
22. Borrelli, F., Bemporad, A., Morari, M.: A geometric algorithm for multi-parametric linear programming. Journal of Optimization Theory and Applications 118(3), 515–540 (2003)
23. Di Cairano, S., Bemporad, A., Kolmanovsky, I., Hrovat, D.: Model predictive control of magnetically actuated mass spring dampers for automotive applications. International Journal of Control 80(11), 1701–1716 (2007)
24. Campo, P.J., Morari, M.: Robust model predictive control. In: American Control Conference, vol. 2, pp. 1021–1026 (1987)
25. Christophersen, F.J., Kvasnica, M., Jones, C.N., Morari, M.: Efficient evaluation of piecewise control laws defined over a large number of polyhedra. In: Proc. European Control Conf., Kos, Greece, pp. 2360–2367 (2007)
26. Christophersen, F.J., Zeilinger, M.N., Jones, C.N., Morari, M.: Controller complexity reduction for piecewise affine systems through safe region elimination. In: IEEE Conference on Decision and Control, New Orleans, LA, pp. 4773–4778 (2007)
27. Dua, V., Pistikopoulos, E.N.: An algorithm for the solution of multiparametric mixed integer linear programming problems. Annals of Operations Research 1, 123–139 (2000)
28. Filippi, C.: On the geometry of optimal partition sets in multiparametric linear programming. Technical Report 12, Department of Pure and Applied Mathematics, University of Padova, Italy (June 1997)
29. Gal, T.: Postoptimal Analyses, Parametric Programming, and Related Topics, 2nd edn. de Gruyter, Berlin (1995)
30. Gal, T., Greenberg, H.J. (eds.): Advances in Sensitivity Analysis and Parametric Programming. International Series in Operations Research & Management Science, vol. 6. Kluwer Academic Publishers, Dordrecht (1997)
31. Gal, T., Nedoma, J.: Multiparametric linear programming. Management Science 18, 406–442 (1972)
32. Gass, S.I., Saaty, T.L.: The computational algorithm for the parametric objective function. Naval Research Logistics Quarterly 2, 39–45 (1955)

33. Geyer, T., Torrisi, F.D., Morari, M.: Optimal complexity reduction of polyhedral piecewise affine systems. Automatica 44, 1728–1740 (2008)
34. Geyer, T., Torrisi, F.D., Morari, M.: Efficient Mode Enumeration of Compositional Hybrid Models. In: Maler, O., Pnueli, A. (eds.) HSCC 2003. LNCS, vol. 2623, pp. 216–232. Springer, Heidelberg (2003)
35. Grieder, P., Kvasnica, M., Baotić, M., Morari, M.: Low complexity control of piecewise affine systems with stability guarantee. In: American Control Conference, Boston, MA (2004)
36. Grieder, P., Morari, M.: Complexity reduction of receding horizon control. In: Proc. 42th IEEE Conf. on Decision and Control, Maui, Hawaii, USA, pp. 3179–3184 (2003)
37. Ingimundarson, A., Ocampo-Martinez, C., Bemporad, A.: Model predictive control of hybrid systems based on mode-switching constraints. In: Proc. 46th IEEE Conf. on Decision and Control, New Orleans, LA, pp. 5265–5269 (2007)
38. Johansen, T.A.: On multi-parametric nonlinear programming and explicit nonlinear model predictive control. In: Proc. 41th IEEE Conf. on Decision and Control, Las Vegas, Nevada, USA, pp. 2768–2773 (December 2002)
39. Johansen, T.A.: Approximate explicit receding horizon control of constrained nonlinear systems. Automatica 40, 293–300 (2004)
40. Johansen, T.A., Grancharova, A.: Approximate explicit constrained linear model predictive control via orthogonal search tree. IEEE Trans. Automatic Control 58(5), 810–815 (2003)
41. Johansen, T.A., Jackson, W., Schrieber, R., Tøndel, P.: Hardware synthesis of explicit model predictive controllers. IEEE Trans. Contr. Systems Technology 15(1), 191–197 (2007)
42. Johansen, T.A., Petersen, I., Slupphaug, O.: On explicit suboptimal LQR with state and input constraints. In: Proc. 39th IEEE Conf. on Decision and Control, Sydney, Australia, pp. 662–667 (December 2000)
43. Jones, C., Morari, M.: Multiparametric linear complementarity problems. In: Proc. 45th IEEE Conf. on Decision and Control, San Diego, CA, pp. 5687–5692 (2006)
44. Jones, C.N., Baric, M., Morari, M.: Multiparametric linear programming with applications to control. European Journal of Control 13, 152–170 (2007)
45. Jones, C.N., Grieder, P., Raković, S.V.: A logarithmic-time solution to the point location problem for parametric linear programming. Automatica 42(12), 2215–2218 (2006)
46. Jones, C.N., Kerrigan, E.C., Maciejowski, J.M.: Lexicographic perturbation for multiparametric linear programming with applications to control. Automatica 43(10), 1808–1816 (2007)
47. Kerrigan, E.C., Maciejowski, J.M.: Feedback min-max model predictive control using a single linear program: Robust stability and the explicit solution. Int. J. Robust Nonlinear Control 14(4), 395–413 (2004)
48. Kvasnica, M., Grieder, P., Baotić, M.: Multi Parametric Toolbox, MPT (2006), http://control.ee.ethz.ch/~mpt/
49. Lincoln, B., Rantzer, A.: Relaxing dynamic programming. IEEE Trans. Automatic Control 51(8), 1249–1260 (2006)
50. Mangasarian, O.L., Rosen, J.B.: Inequalities for stochastic nonlinear programming problems. Operations Research 12, 143–154 (1964)
51. Mayne, D.Q., Rakovíc, S.V., Kerrigan, E.C.: Optimal control and piecewise parametric programming. In: Proc. European Control Conf., Kos, Greece, pp. 2762–2767 (2007)

52. Mayne, D.Q.: Control of constrained dynamic systems. Technical Report EEE/C&P/DQM/9/2001, Imperial College, London, U.K. (September 2001)
53. Mayne, D.Q., Rakovic, S.: Optimal control of constrained piecewise affine discrete time systems using reverse transformation. In: Proc. 41th IEEE Conf. on Decision and Control, Las Vegas, Nevada, USA, pp. 1546–1551 (December 2002)
54. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: Stability and optimality. Automatica 36(6), 789–814 (2000)
55. Muñoz de la Peña, D., Alamo, T., Bemporad, A., Camacho, E.F.: A dynamic programming approach for determining the explicit solution of MPC controllers. In: Proc. 43th IEEE Conf. on Decision and Control, Paradise Island, Bahamas, pp. 2479–2484 (2004)
56. Muñoz de la Peña, D., Alamo, T., Ramírez, D.R., Camacho, E.F.: Min-max model predictive control as a quadratic program. IET Control Theory & Applications 1(1), 328–333 (2007)
57. Muñoz de la Peña, D., Bemporad, A., Filippi, C.: Robust explicit MPC based on approximate multi-parametric convex programming. IEEE Trans. Automatic Control 51(8), 1399–1403 (2006)
58. Orchard-Hays, W.: Notes on linear programming (part 6): the Rand code for the simplex method (sx4). Technical Report 1440, Rand Corporation (1955)
59. Pannocchia, G., Rawlings, J.B., Wright, S.J.: Fast, large-scale model predictive control by partial enumeration. Automatica 43, 852–860 (2007)
60. Propoi, A.I.: Use of linear programming methods for synthesizing sampled-data automatic systems. Automation and Remote Control 24(7), 837–844 (1963)
61. Ramírez, D.R., Camacho, E.F.: Piecewise affinity of min-max mpc with bounded additive uncertainties and a quadratic criterion. Automatica 42(2), 295–302 (2006)
62. Schechter, M.: Polyhedral functions and multiparametric linear programming. Journal of Optimization Theory and Applications 53(2), 269–280 (1987)
63. Scokaert, P.O.M., Mayne, D.Q.: Min-max feedback model predictive control for constrained linear systems. IEEE Trans. Automatic Control 43(8), 1136–1142 (1998)
64. Seron, M.M., DeDoná, J.A., Goodwin, G.C.: Global analytical model predictive control with input constraints. In: Proc. 39th IEEE Conf. on Decision and Control, Sydney, Australia, pp. 154–159 (2000)
65. Spjøtvold, J., Kerrigan, E.C., Jones, C.N., Tøndel, P., Johansen, T.A.: On the facet-to-facet property of solutions to convex parametric quadratic programs. Automatica 42(12), 2209–2214 (2006)
66. Spjøtvold, J., Rakovic, S.V., Tøndel, P., Johansen, T.A.: Utilizing reachability analysis in point location problems. In: Proc. 45th IEEE Conf. on Decision and Control, San Diego, CA, pp. 4568–4569 (2006)
67. Tøndel, P., Johansen, T.A.: Complexity reduction in explicit linear model predictive control. In: Proc. 15th IFAC World Congress, Barcelona, Spain (2002)
68. Tøndel, P., Johansen, T.A., Bemporad, A.: An algorithm for multi-parametric quadratic programming and explicit MPC solutions. Automatica 39(3), 489–497 (2003)
69. Tøndel, P., Johansen, T.A., Bemporad, A.: Evaluation of piecewise affine control via binary search tree. Automatica 39(5), 945–950 (2003)
70. Tøndel, P., Johansen, T.A., Bemporad, A.: Further results on multiparametric quadratic programming. In: Proc. 42th IEEE Conf. on Decision and Control, Maui, Hawaii, USA, pp. 3173–3178 (2003)

71. Torrisi, F.D., Bemporad, A.: HYSDEL — A tool for generating computational hybrid models. IEEE Trans. Contr. Systems Technology 12(2), 235–249 (2004)
72. Wang, Y., Jones, C.N., Maciejowski, J.: Efficient point location via subdivision walking with application to explicit MPC. In: Proc. European Control Conf., Kos, Greece, pp. 447–453 (2007)
73. Witsenhausen, H.S.: A min-max control problem for sampled linear systems. IEEE Trans. Automatic Control 13(1), 5–21 (1968)
74. Zafiriou, E.: Robust model predictive control of processes with hard constraints. Computers & Chemical Engineering 14(4/5), 359–371 (1990)

Explicit Approximate Model Predictive Control of Constrained Nonlinear Systems with Quantized Input

Alexandra Grancharova and Tor A. Johansen

Abstract. In this paper, a Model Predictive Control problem for constrained nonlinear systems with *quantized* input is formulated and represented as a multi-parametric Nonlinear Integer Programming (mp-NIP) problem. Then, a computational method for *explicit* approximate solution of the resulting mp-NIP problem is suggested. The proposed approximate mp-NIP approach is applied to the design of an *explicit* approximate MPC controller for a clutch actuator with on/off valves.

Keywords: MPC, multi-parametric Nonlinear Integer Programming.

1 Introduction

In several control engineering problems, the system to be controlled is characterized by a finite set of possible control actions. Such systems are referred to as systems with *quantized* control input and the possible values of the input represent the levels of *quantization*. For example, hydraulic systems using on/off valves are systems with *quantized* input. In order to achieve a high quality of the control system performance it would be necessary to take into account the effect of the control input *quantization*. Thus, in [8] receding horizon optimal control ideas were proposed for synthesizing *quantized* control laws for *linear* systems with *quantized* inputs and quadratic optimality criteria. Further in [1], a method for *explicit* solution of optimal control problems with *quantized* control input was developed. It is based

Alexandra Grancharova
Institute of Control and System Research, Bulgarian Academy of Sciences,
P.O. Box 79, Sofia 1113, Bulgaria
e-mail: alexandra.grancharova@abv.bg

Tor A. Johansen
Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway
e-mail: Tor.Arne.Johansen@itk.ntnu.no

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 371–380.
springerlink.com

on solving multi-parametric Nonlinear Integer Programming (mp-NIP) problems, where the cost function and the constraints depend linearly on the vector of parameters. In this paper, a Model Predictive Control (MPC) problem for constrained *nonlinear* systems with *quantized* input is formulated and represented as an mp-NIP problem. Then, a computational method for *explicit* approximate solution of the resulting mp-NIP problem is suggested. The benefits of the *explicit* solution consist in efficient on-line computations using a binary search tree and verifiability of the design and implementation. The mp-NIP method proposed here is more general compared to the mp-NIP method in [1], since it allows the cost function and the constraints to depend nonlinearly on the vector of parameters.

In the paper, $A \succ 0$ means that the square matrix A is positive definite. For $x \in \mathbb{R}^n$, the Euclidean norm is $\|x\| = \sqrt{x^T x}$ and the weighted norm is defined for some symmetric matrix $A \succ 0$ as $\|x\|_A = \sqrt{x^T A x}$.

2 Formulation of Quantized Nonlinear Model Predictive Control Problem

Consider the discrete-time nonlinear system:

$$x(t+1) = f(x(t), u(t)) \tag{1}$$
$$y(t) = Cx(t), \tag{2}$$

where $x(t) \in \mathbb{R}^n$ is the state variable, $y(t) \in \mathbb{R}^p$ is the output variable, and $u(t) \in \mathbb{R}^m$ is the control input, which is constrained to belong to the finite set of values $U^A = \{\overline{u}_1, \overline{u}_2, \ldots, \overline{u}_L\}$, $\overline{u}_i \in \mathbb{R}^m$, $\forall i = 1, 2, \ldots, L$, i.e. $u \in U^A$. Here, $\overline{u}_1, \overline{u}_2, \ldots, \overline{u}_L$ represent the levels of *quantization* of the control input u. In (1), $f : \mathbb{R}^n \times U^A \longmapsto \mathbb{R}^n$ is a nonlinear function.

We consider a reference tracking problem where the goal is to have the output variable $y(t)$ track the reference signal $r(t) \in \mathbb{R}^p$. Suppose that a full measurement of the state $x(t)$ is available at the current time t. For the current $x(t)$, the reference tracking *quantized* NMPC solves the following optimization problem:

Problem P1

$$V^*(x(t), r(t)) = \min_{U \in U^B} J(U, x(t), r(t)) \tag{3}$$

subject to $x_{t|t} = x(t)$ and:

$$y_{\min} \leq y_{t+k|t} \leq y_{\max}, k = 1, \ldots, N \tag{4}$$
$$u_{t+k} \in U^A = \{\overline{u}_1, \overline{u}_2, \ldots, \overline{u}_L\}, k = 0, 1, \ldots, N-1 \tag{5}$$
$$\left\| y_{t+N|t} - r(t) \right\| \leq \delta \tag{6}$$
$$x_{t+k+1|t} = f(x_{t+k|t}, u_{t+k}), k \geq 0 \tag{7}$$
$$y_{t+k|t} = Cx_{t+k|t}, k \geq 0 \tag{8}$$

Here, $U = [u_t, u_{t+1}, \ldots, u_{t+N-1}] \in \mathbb{R}^{Nm}$ is the set of free control moves, $U^B = (U^A)^N = U^A \times \ldots \times U^A$ and the cost function is given by:

$$J(U, x(t), r(t)) = \sum_{k=0}^{N-1} \left[\left\| y_{t+k|t} - r(t) \right\|_Q^2 + \left\| h(x_{t+k|t}, u_{t+k}) \right\|_R^2 \right] + \left\| y_{t+N|t} - r(t) \right\|_P^2 \tag{9}$$

Here, N is a finite horizon and $h : \mathbb{R}^n \times U^A \longmapsto \mathbb{R}^s$ is a nonlinear function. It is assumed that $P, Q, R \succ 0$. From a stability point of view it is desirable to choose δ in (6) as small as possible. However, in the case of quantized input, the equilibrium point of the closed-loop system may either have an offset from the reference, or there may be a limit cycle about the reference. Therefore, the feasibility of (3)–(9) will rely on δ being sufficiently large. A part of the NMPC design will be to address this tradeoff. We introduce an extended state vector:

$$\tilde{x}(t) = [x(t), r(t)] \in \mathbb{R}^{\tilde{n}}, \ \tilde{n} = n + p \tag{10}$$

Let $\tilde{x}$ be the value of the extended state at the current time t. Then, the optimization problem P1 can be formulated in a compact form as follows:

Problem P2

$$V^*(\tilde{x}) = \min_{U \in U^B} J(U, \tilde{x}) \ \text{ subject to } \ G(U, \tilde{x}) \leq 0 \tag{11}$$

The *quantized* NMPC problem defines a multi-parametric Nonlinear Integer Programming problem (mp-NIP), since it is a Nonlinear Integer Programming problem in U parameterized by $\tilde{x}$. An optimal solution to this problem is denoted $U^* = [u_t^*, u_{t+1}^*, \ldots, u_{t+N-1}^*]$ and the control input is chosen according to the receding horizon policy $u(t) = u_t^*$. Define the set of N-step feasible initial states as follows:

$$X_f = \{\tilde{x} \in \mathbb{R}^{\tilde{n}} \,|\, G(U, \tilde{x}) \leq 0 \text{ for some } U \in U^B\} \tag{12}$$

If δ in (6) is chosen such that the problem P1 is feasible, then X_f is a non-empty set.

In parametric programming problems one seeks the solution $U^*(\tilde{x})$ as an *explicit* function of the parameters $\tilde{x}$ in some set $\underline{X} \subseteq X_f \subseteq \mathbb{R}^{\tilde{n}}$ [2]. In this paper we suggest a computational method for constructing an *explicit* piecewise constant (PWC) approximate solution of the reference tracking *quantized* NMPC problem.

3 Approximate mp-NIP Approach to Explicit Quantized NMPC

3.1 Computation of Feasible PWC Solution

Definition 1 (Feasibility on a discrete set)
Let $\bar{X} \subset \mathbb{R}^{\tilde{n}}$ be a hyper-rectangle and $V_{\bar{X}} = \{v_1, v_2, \ldots, v_Q\} \subset \bar{X}$ be a discrete set. A function $U(\tilde{x})$ is feasible on $V_{\bar{X}}$ if $G(U(v_i), v_i) \leq 0$, $i \in \{1, 2, \ldots, Q\}$.

We restrict our attention to a hyper-rectangle $X \subset \mathbb{R}^{\tilde{n}}$ where we seek to approximate the optimal solution $U^*(\tilde{x})$ to problem P2. We require that the state space partition is orthogonal and can be represented as a $k-d$ tree. The main idea of the approximate mp-NIP approach is to construct a feasible on a discrete set *piecewise constant* (PWC) approximation $\widehat{U}(\tilde{x})$ to $U^*(\tilde{x})$ on X, where the constituent constant functions are defined on hyper-rectangles covering X. The solution of problem P2 is computed at the $2^{\tilde{n}}$ vertices of a considered hyper-rectangle X_0, as well as at some interior points. These additional points represent the vertices and the facets centers of one or more hyper-rectangles contained in the interior of X_0. The following procedure is used to generate a set of points $V_0 = \{v_0, v_1, v_2, \ldots, v_{N_1}\}$ associated to a hyper-rectangle X_0:

Procedure 1 (Generation of set of points)
Consider any hyper-rectangle $X_0 \subseteq X$ with vertices $\Lambda^0 = \{\lambda_1^0, \lambda_2^0, \ldots, \lambda_{N_\lambda}^0\}$ and center point v_0. Consider also the hyper-rectangles $X_0^j \subset X_0$, $j = 1,2,\ldots,N_0$ with vertices respectively $\Lambda^j = \{\lambda_1^j, \lambda_2^j, \ldots, \lambda_{N_\lambda}^j\}$, $j = 1,2,\ldots,N_0$. Suppose $X_0^1 \subset X_0^2 \subset \ldots \subset X_0^{N_0}$. For each of the hyper-rectangles X_0 and $X_0^j \subset X_0$, $j = 1,2,\ldots,N_0$, denote the set of its facets centers with $\Phi^j = \{\phi_1^j, \phi_2^j, \ldots, \phi_{N_\phi}^j\}$, $j = 0,1,2,\ldots,N_0$. Define the set of all points $V_0 = \{v_0, v_1, v_2, \ldots, v_{N_1}\}$, where $v_i \in \left\{\bigcup_{j=0}^{N_0} \Lambda^j\right\} \bigcup \left\{\bigcup_{j=0}^{N_0} \Phi^j\right\}$, $i = 1,2,\ldots,N_1$.

A close-to-global solution $U^*(v_i)$ of problem P2 at a point $v_i \in V_0$ is computed by using the routine 'glcSolve' of the TOMLAB optimization environment in Matlab [4]. The routine 'glcSolve' implements an extended version of the DIRECT algorithm [5], that handles problems with both nonlinear and integer constraints. The DIRECT algorithm (DIviding RECTangles) [5] is a deterministic sampling algorithm for searching for the global minimum of a multivariate function subject to constraints, using no derivative information. It is a modification of the standard Lipschitzian approach that eliminates the need to specify a Lipschitz constant.

Based on the close-to-global solutions $U^*(v_i)$ at all points $v_i \in V_0$, a local constant approximation $\widehat{U}_0(\tilde{x}) = K_0$ to the optimal solution $U^*(\tilde{x})$, feasible on the set V_0 and valid in the whole hyper-rectangle X_0, is determined by applying the following procedure:

Procedure 2 (Computation of explicit approximate solution)
Consider any hyper-rectangle $X_0 \subseteq X$ with a set of points $V_0 = \{v_0, v_1, \ldots, v_{N_1}\}$ determined by applying Procedure 1. Compute K_0 by solving the following NIP:

$$\min_{K_0 \in U^B} \sum_{i=0}^{N_1} \left(J(K_0, v_i) - V^*(v_i)\right) \quad \text{subject to} \quad G(K_0, v_i) \leq 0, \forall v_i \in V_0 \tag{13}$$

3.2 Estimation of Error Bounds

Suppose that a constant function $\widehat{U}_0(\tilde{x}) = K_0$ that is feasible on $V_0 \subset X_0$ has been determined by applying Procedure 2. Then, for the cost function approximation error in X_0 we have:

$$\varepsilon(\tilde{x}) = \widehat{V}(\tilde{x}) - V^*(\tilde{x}) \leq \varepsilon_0 \,,\ \tilde{x} \in X_0 \tag{14}$$

where $\widehat{V}(\tilde{x}) = J(\widehat{U}_0(\tilde{x}), \tilde{x})$ is the sub-optimal cost and $V^*(\tilde{x})$ denotes the cost corresponding to the close-to-global solution $U^*(\tilde{x})$, i.e. $V^*(\tilde{x}) = J(U^*(\tilde{x}), \tilde{x})$. The following procedure can be used to obtain an estimate $\widehat{\varepsilon}_0$ of the maximal approximation error ε_0 in X_0.

Procedure 3 (Computation of error bound approximation)
Consider any hyper-rectangle $X_0 \subseteq X$ with a set of points $V_0 = \{v_0, v_1, \ldots, v_{N_1}\}$ determined by applying Procedure 1. Compute an estimate $\widehat{\varepsilon}_0$ of the error bound ε_0 through the following maximization:

$$\widehat{\varepsilon}_0 = \max_{i \in \{0,1,2,\ldots,N_1\}} (\widehat{V}(v_i) - V^*(v_i)) \tag{15}$$

3.3 Approximate mp-NIP Algorithm

Assume the tolerance $\bar{\varepsilon} > 0$ of the cost function approximation error is given. The following algorithm is proposed to design *explicit* reference tracking *quantized* NMPC:

Algorithm 1 (explicit reference tracking quantized NMPC)

1. Initialize the partition to the whole hyper-rectangle, i.e. $\Pi = \{X\}$. Mark the hyper-rectangle X as unexplored.
2. Select any unexplored hyper-rectangle $X_0 \in \Pi$. If no such hyper-rectangle exists, terminate.
3. Generate a set of points $V_0 = \{v_0, v_1, v_2, \ldots, v_{N_1}\}$ associated to X_0 by applying Procedure 1.
4. Compute a solution to problem P2 for $\tilde{x}$ fixed to each of the points v_i, $i = 0, 1, 2, \ldots, N_1$ by using routine 'glcSolve' of TOMLAB optimization environment. If problem P2 has a feasible solution at all these points, go to step 6. Otherwise, go to step 5.
5. Compute the size of X_0 using some metric. If it is smaller than some given tolerance, mark X_0 infeasible and explored and go to step 2. Otherwise, split X_0 into hyper-rectangles $X_1, X_2, \ldots, X_{N_s}$ by applying the heuristic rule 1 from [3]. Mark $X_1, X_2, \ldots, X_{N_s}$ unexplored, remove X_0 from Π, add $X_1, X_2, \ldots, X_{N_s}$ to Π, and go to step 2.
6. Compute a constant function $\widehat{U}_0(\tilde{x})$ using Procedure 2, as an approximation to be used in X_0. If no feasible solution was found, split X_0 into two hyper-rectangles X_1 and X_2 by applying the heuristic rule 3 from [3]. Mark X_1 and X_2 unexplored, remove X_0 from Π, add X_1 and X_2 to Π, and go to step 2.

7. Compute an estimate $\widehat{\varepsilon}_0$ of the error bound ε_0 in X_0 by applying Procedure 3. If $\widehat{\varepsilon}_0 \leq \bar{\varepsilon}$, mark X_0 as explored and feasible and go to step 2. Otherwise, split X_0 into two hyper-rectangles X_1 and X_2 by applying Procedure 4 from [3]. Mark X_1 and X_2 unexplored, remove X_0 from Π, add X_1 and X_2 to Π, and go to step 2.

4 Explicit Quantized NMPC of an Electropneumatic Clutch Actuator Using On/Off Valves

Here, a pneumatic actuator of an electropneumatic clutch system is considered. The pneumatic actuator acts on the clutch plates through the clutch spring, and the state of the clutch directly depends on the actuator position. The actuator is controlled by using on/off valves. In comparison to proportional valves, the on/off valves are smaller and cheaper. In [7], [9] the case when only fully open and closed are possible states of the valves is considered. Then, a controller is designed to govern switches between these states based on backstepping and Lyapunov theory. It should be noted however, that the methods in [7], [9] can not explicitly handle the constraints imposed on the clutch actuator position. On the other hand, Model Predictive Control (MPC) is an optimization based method for control which can explicitly handle both state and input constraints. This makes the MPC methodology very suitable to the optimal control of the clutch actuator. The fast dynamics of the clutch actuator, characterized with sampling time of about 0.01 [s] requires the design of an *explicit* MPC controller, where the only computation performed on-line would be a simple function evaluation.

4.1 Description of the Electropneumatic Clutch Actuator

The clutch actuator system is shown in Figure 1. To control both supply to and exhaust from the clutch actuator chamber, at least one pair of on/off valves are needed. As we only allow these to be fully open or closed, with two valves and under the assumption

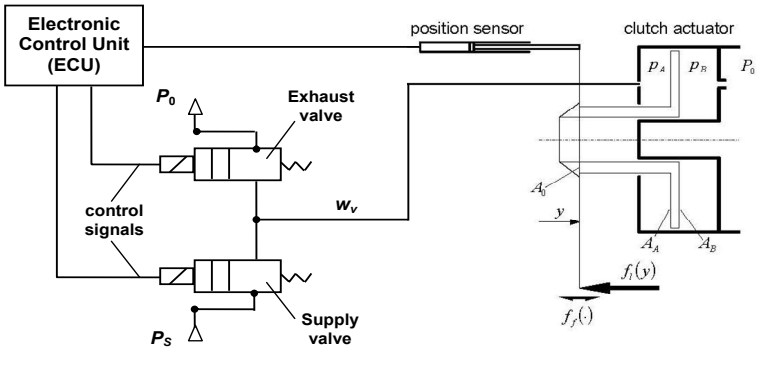


Fig. 1 Electropneumatic clutch actuator ([6], [9])

of choked flow, we restrict the flow of the clutch actuator to three possible values: maximum flow into the volume, maximum flow out of the volume, or no flow [7], [9]. The electronic control unit (ECU) calculates and sets voltage signals to control the on/off valves. These signals control whether the valve should open or close, and thus also the flow into the actuator. A position sensor measures position and feeds it back to the ECU. To calculate the control signals, knowledge of other states of the system are also needed, and these can be obtained either by sensors or by estimation. The full 5-th order model of the clutch actuator dynamics is the following [6]:

$$\dot{y} = v \tag{16}$$

$$\dot{v} = \frac{1}{M}(A_0 P_0 + A_A p_A - A_B p_B - f_f(v,z) - f_l(y)) \tag{17}$$

$$\dot{p}_A = -\frac{A_A}{V_A(y)} v p_A + \frac{RT_0}{V_A(y)} w_v(p_A, u) \tag{18}$$

$$\dot{p}_B = \frac{A_B}{V_B(y)} v p_B + \frac{RT_0}{V_B(y)} w_r(p_B) \tag{19}$$

$$\dot{z} = v - \frac{K_z}{F_C}|v|_q z \tag{20}$$

where y is the position, v is the velocity, p_A is the pressure in chamber A, p_B is the pressure in chamber B, z is the friction state, $w_v(p_A,u)$ is the flow to/from chamber A, $w_r(p_B)$ is the flow to/from chamber B, u is an integer control variable introduced below, and $V_A(y)$ and $V_B(y)$ are the volumes of chambers A and B. The meaning of the parameters is the following: A_A and A_B are the areas of chambers A and B, $A_0 = A_B - A_A$ is piston area, M is piston mass, P_0 is the ambient pressure, T_0 is the temperature, R is the gas constant of air, K_z is asperity stiffness, F_C is Coulomb friction. In (20), $|v|_q = \sqrt{v^2 + \sigma^2}$, where $\sigma > 0$ is an arbitrary small design parameter. In (17), $f_l(y)$ and $f_f(v,z)$ are the clutch load and the friction force, described by:

$$f_l(y) = K_l(1 - e^{-L_l y}) - M_l y\,,\ \ f_f(v,z) = D_v v + K_z z + D_{\dot{z}} \dot{z}(v,z) \tag{21}$$

An integer control variable $u \in U^A = \{1,2,3\}$ is introduced which is related to the flow $w_v(p_A,u)$ to/from chamber A in the following way:

$$u = 1 \Rightarrow w_v(p_A,1) = -\rho_0 C_{v,out}\, \psi(r, B_{v,out}) p_A\,,\ \ r = \frac{P_0}{p_A} \tag{22}$$

$$u = 2 \Rightarrow w_v(p_A,2) = 0 \tag{23}$$

$$u = 3 \Rightarrow w_v(p_A,3) = \rho_0 C_{v,in}\, \psi(r, B_{v,in}) P_S\,,\ \ r = \frac{p_A}{P_S} \tag{24}$$

In (24), P_S is the supply pressure. Therefore, $u = 1$ corresponds to maximal flow from chamber A, $u = 2$ means no flow, and $u = 3$ corresponds to maximal flow to chamber A. The expressions for the valve flow function $\psi(r, B_{v,in/out})$, as well as for the flow $w_r(p_B)$ to/from chamber B can be found in [6].

4.2 Design of Explicit Quantized NMPC

In order to reduce the computational burden, the design of the *explicit quantized* NMPC controller is based on a simplified 3-rd order model of the clutch actuator, where the states are the actuator position y^s, the velocity v^s and the pressure p_A^s in chamber A:

$$\dot{y}^s = v^s \tag{25}$$

$$\dot{v}^s = \frac{1}{M}(-A_A P_0 + A_A p_A^s - f_f^*(v^s) - f_l(y^s)) \tag{26}$$

$$\dot{p}_A^s = -\frac{A_A}{V_A(y^s)} v^s p_A^s + \frac{RT_0}{V_A(y^s)} w_{v^s}(p_A^s, u) \tag{27}$$

In (26), $f_f^*(v^s) = D_v v^s + F_C \frac{v^s}{\sqrt{v^{s2}+\sigma^2}}$ is a static sliding friction characteristic [6]. The system (25)–(27) is discretized using a sampling time $T_s = 0.01$ [s] and zero-order hold (the control input is assumed constant over the sampling interval). The forward Euler method with stepsize $T_E = 0.0001$ [s] is used to integrate the equations (25)–(27). The control objective is to have the actuator position y^s track a reference signal $r(t) > 0$, which is achieved by minimizing the following cost function:

$$J(U, y^s(t), r(t)) = \sum_{k=0}^{N-1} \left[Q \left(\frac{y_{t+k|t}^s - r(t)}{r(t)} \right)^2 + R \left(\frac{w_{v^s}(p_{A,t+k|t}^s, u_{t+k})}{w_{v^s,max} - w_{v^s,min}} \right)^2 \right] + P \left(\frac{y_{t+N|t}^s - r(t)}{r(t)} \right)^2 \tag{28}$$

where $N = 10$ is the horizon, $Q = 1$, $R = 0.1$, $P = 1$ are the weighting coefficients, and $w_{v^s,max}$ and $w_{v^s,min}$ are the maximal and the minimal flows to/from chamber A. The following constraints are imposed:

$$y_{\min} \leq y_{t+k|t}^s \leq y_{\max}, k = 1, ..., N; u_{t+k} \in U^A = \{1, 2, 3\}, k = 0, 1, ..., N-1 \tag{29}$$

where $y_{\min} = 0$, $y_{\max} = 0.025$ [m]. In (28), $U \in U^B = (U^A)^N$. The *quantized* NMPC minimizes the cost (28) subject to the system equations (25)–(27) and the constraints (29). The extended state vector is $\tilde{x}(t) = [e(t), v^s(t), p_A^s(t), r(t)] \in \mathbb{R}^4$, where the state $e(t)$ is the projected reference tracking error defined as:

$$e(t) = \begin{cases} r(t) - y^s(t), \text{ if } -0.005 \leq r(t) - y^s(t) \leq 0.005 \\ -0.005, \text{ if } r(t) - y^s(t) < -0.005 \\ 0.005, \text{ if } r(t) - y^s(t) > 0.005 \end{cases} \tag{30}$$

The state space to be partitioned is 4-dimensional and it is defined by $X = [-0.005; 0.005] \times [-0.05; 0.15] \times [P_0; P_S] \times [0.0001; 0.024]$. The cost function approximation tolerance is chosen as $\bar{\varepsilon}(X_0) = \max(\bar{\varepsilon}_a, \bar{\varepsilon}_r \min_{\tilde{x} \in X_0} V^*(\tilde{x}))$, where $\bar{\varepsilon}_a = 0.001$ and $\bar{\varepsilon}_r = 0.02$ are the absolute and the relative tolerances. The partition has 10871

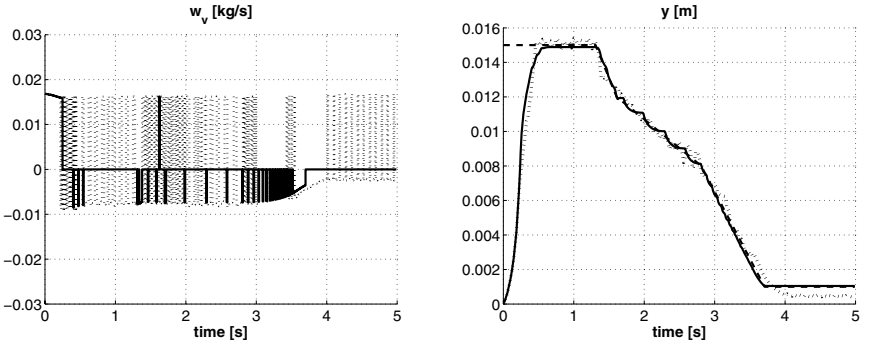


Fig. 2 Left: The valve flow $w_v(p_A, u)$. Right: The clutch actuator position y. The dotted curves are with the approximate explicit quantized NMPC, the solid curves are with the exact quantized NMPC and the dashed curve is the reference signal

regions and 17 levels of search. Thus, 17 arithmetic operations are needed in real-time to compute the control input (17 comparisons).

The performance of the *explicit quantized* NMPC controller was simulated for a typical clutch reference signal and the resulting response is depicted in Figure 2. The simulations are based on the full 5-th order model (16)–(20).

Acknowledgements. This work was sponsored by the *Research Council of Norway*, projects 168194/I40 and 177616/V30.

References

1. Bemporad, A.: Multiparametric nonlinear integer programming and explicit quantized optimal control. In: 42-nd IEEE Conference an Decision and Control, Maui, Hawaii, USA (December 2003)
2. Fiacco, A.V.: Introduction to sensitivity and stability analysis in nonlinear programming. Academic Press, Orlando (1983)
3. Grancharova, A., Johansen, T.A., Tøndel, P.: Computational aspects of approximate explicit nonlinear model predictive control. In: Findeisen, R., Allgöwer, F., Biegler, L. (eds.) Assessment and Future Directions of Nonlinear Model Predictive Control. LNCIS, vol. 358, pp. 181–192. Springer, Heidelberg (2007)
4. Holmström, K., Göran, A.O., Edvall, M.M.: User's guide for TOMLAB (April 2007)
5. Jones, D.R.: The DIRECT global optimization algorithm. In: Floudas, C.A., Pardalos, P.M. (eds.) Encyclopedia of optimization, vol. 1, pp. 431–440. Kluwer, Dordrecht (2001)
6. Kaasa, G.O.: Nonlinear output-feedback control applied to electro-pneumatic clutch actuation in heavy-duty trucks. Ph.D. Thesis, Norwegian University of Science and Technology (2006)
7. Langjord, H., Johansen, T.A., Hespanha, J.P.: Switched control of an electropneumatic clutch actuator using on/off valves. In: American Control Conference, Seattle (2008)

8. Picasso, B., Pancanti, S., Bemporad, A., Bicchi, A.: Receding-horizon control of LTI systems with quantized inputs. In: IFAC Conference on Analysis and Design of Hybrid Systems, Saint Malo, France (June 2002)
9. Sande, H., Johansen, T.A., Kaasa, G.O., Snare, S.R., Bratli, C.: Switched backstepping control of an electropneumatic clutch actuator using on/off valves. In: American Control Conference, New York (2007)

Parametric Approach to Nonlinear Model Predictive Control

M. Herceg, M. Kvasnica, and M. Fikar

Abstract. In model predictive control (MPC), a dynamic optimization problem (DOP) is solved at each sampling instance for a given value of the initial condition. In this work we show how the computational burden induced by the repetitive solving of the DOP for nonlinear systems can be reduced by transforming the unconstrained DOP to a suboptimal DOP with horizon one. The approach is based on solving the stationary Hamilton-Jacobi-Bellman (HJB) equation along a given path while constructing control Lyapunov function (CLF). It is illustrated that for particular cases the problem can be further simplified to a set of differential algebraic equations (DAE) for which an explicit solution can be found without performing optimization.

Keywords: Nonlinear Model Predictive Control, Control Lyapunov Function.

1 Introduction

Model predictive control [9] is a well established control strategy where the optimal control actions are obtained by optimizing the predicted plant behavior. The predictions are obtained by employing a model of the controlled plant. MPC is traditionally implemented in the so-called *Receding Horizon* fashion, which introduces feedback by repeating the optimization at each time instance based on the actual measurements of the state variables. If the model of the plant is nonlinear, a nonlinear optimization problem needs to be solved at each time step. Even though many efficient strategies for fulfilling this task have been proposed, see e.g. [1, 3, 5], they could be of prohibitive complexity when considering real-time implementation. If the model of the plant is linear or piecewise linear, it was shown e.g. by [2] that

M. Herceg, M. Kvasnica, and M. Fikar
Institute of Information Engineering, Automation and Mathematics,
Slovak University of Technology in Bratislava,
Radlinského 9, 81237 Bratislava, Slovakia
e-mail: martin.herceg,michal.kvasnica,miroslav.fikar@stuba.sk

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 381–389.
springerlink.com

the optimal solution to MPC problems can be obtained as a piecewise affine explicit function of the state. Hence, the on-line implementation can be reduced to a simple set-membership test, mitigating the implementation effort. This scheme is often referred to as the *explicit* or *parametric* approach to MPC.

If the model of the plant is nonlinear, however, little is known about the existence of explicit solutions to nonlinear MPC problems. The parametric approach of [8] uses approximations of the nonlinearities to come up with an explicit representation of the feedback law, leading to suboptimal solutions. It was shown in [4] that optimal MPC feedback laws could be obtained in an explicit form if the prediction model is assumed to be nonlinear and input-affine. The predictions are then based on Taylor series expansion. However, as the Taylor series are valid only locally, the MPC is formulated over a short finite horizon. As investigated by [10], exact solutions can be found even for an infinite horizon. The disadvantage is, though, that approach of [10] proceeds with the synthesis in a reversed order, i.e. starts with a known optimal solution and constructs the corresponding nonlinear model.

This contribution illustrates a way towards obtaining parametric solutions of MPC problems for unconstrained nonlinear systems. The idea is based on replacing the unconstrained DOP formulated over an infinite horizon by a different optimization problem, where optimization is performed along the shortest path from the origin to a given initial condition. The problem is parametrized by a set of parameters, which originate from the first order optimality conditions, expressed by the stationary HJB equation. The task is hence to find these parameters such that the resulting closed-loop system is stable. For this purpose the concept of CLF [12] is employed which imposes additional constraints for finding these parameters. The link between HJB equation and CLF has been already pointed out by [11, 13], but here the DOP problem is formulated in a parametrized form. However, CLF constraints do not uniquely determine the missing parameters and solution of the original problem can be suboptimal. Therefore, in this paper we concentrate on a two-dimensional case where additional constraints are provided and the shortest path DOP is further transformed to a set of DAE equations, which are parametrized in states. The DAE equations create an implicit solution of the original DOP and by integrating the DAE system along a given path, the optimal solution to the original problem can be recovered.

2 Problem Statement

Consider a non-linear time-invariant dynamical system in a state-space form:

$$\dot{x} = f(x,u), \quad x(0) = x_0, \tag{1a}$$

satisfying $f(0,0) = 0$ where $f(\cdot)$ is at least twice differentiable, $x(t) \in \mathbb{R}^n$ denotes a state vector and $u(t) \in \mathbb{R}^m$ is a vector of manipulated variables (inputs). Under the assumption that all states are measurable, and the system described by (1a) is controllable, the objective is to regulate the system (1a) from any admissible initial condition x_0 to its origin $(0,0)$ while minimizing a given performance index $F(x,u)$,

which is a positive semi-definite, at least twice differentiable function satisfying $F(0,0)=0$. The optimal solution to the regulation problem can be found by solving the following optimization problem on-line for a known value of x_0:

$$V(x) = \min_{u} \int_0^{\infty} F(x,u)\mathrm{d}t \tag{1b}$$
$$\text{s.t.} \quad (1a),$$

which leads to an infinite-dimensional problem. In order to solve such problems numerically, it is a usual practice to truncate the infinite time horizon to a finite horizon plus some terminal cost and terminal set constraint. For practical reasons the finite horizon is usually large, rendering the optimization a complex task. Therefore in the next section we show how to transform the DOP problem (1) to a simpler optimization problem which doesn't rely on terminal cost and terminal set to guarantee stability.

3 Transformation to Shortest Path DOP

3.1 Optimality Conditions

The necessary conditions of optimality stem from defining the Hamiltonian function $H(x,u,\lambda) = F(x,u) + \lambda^T f(x,u)$ by introducing adjoint variables $\lambda \in \mathbb{R}^n$, which are orthogonal to the system dynamics equation (1a) and represent partial derivatives of the value function $V(x)$ at the optimum $\lambda = \partial V(x)/\partial x$. By appending the constraints represented by (1a) to the functional (1b), the optimum can be found by expressing the first variation of the Hamiltonian and by setting it equal to zero. This operation yields a set of criteria differential equations [14] which are not mentioned here due to lack of space. Essentially, these equations define a two-point boundary value problem, because state equations are determined by the initial condition $x(0) = x_0$ and by the equations for adjoint variables with the terminal condition $\lambda(t \to \infty) = 0$. The latter condition is associated with the infinite-time horizon, and moreover, as shown by [14], in this case the Hamiltonian is equal to zero, i.e. $H(x,u,\lambda) = 0$.

Suppose that the adjoint variables λ are only functions of states. Thus, denoting $\lambda = \lambda(x)$, and $\dot{\lambda} = (\partial\lambda/\partial x)\dot{x}$ it is possible to rewrite the optimality conditions in the state space as

$$\frac{\partial F(x,u)}{\partial x} + \frac{\partial \lambda(x)}{\partial x} f(x,u) + \frac{\partial f(x,u)^T}{\partial x}\lambda(x) = 0, \tag{2a}$$

$$\frac{\partial F(x,u)}{\partial u} + \frac{\partial f(x,u)^T}{\partial u}\lambda(x) = 0. \tag{2b}$$

Equations (2a)–(2b) represent the HJB equations in the state-space domain with $\lambda(x)$ as the unknown variable. However, the explicit solution to this equation is not trivial, unless considering some special cases.

3.2 Parametrization of the HJB Equation

Since the HJB equation (2) is time-independent, it is possible to express the initial and terminal conditions by $\lambda(0) = 0$ and $u(0) = 0$. Clearly, these conditions express optimal values at the origin and unknown values are those corresponding to the point x_0, namely $\lambda(x_0)$ and $u(x_0)$. Recalling the principles of receding horizon control, only these values are of interest, because the rest of the computed trajectory is not used anyway. The motivation is thus to find these unknown values as quickly as possible without taking care of the rest of the trajectory. That is, not to solve the HJB equation (2) for the whole state space, but only for the point x_0. However, because the HJB equation is in differential form, the solution is given by initial conditions which are only known at the origin. Therefore, the only way to solve the HJB equation is to integrate it from the origin towards x_0. Intuitively, the shortest, and possibly fastest path in the state space how to reach the point $(x_0, u(x_0))$ is to integrate along a straight line. Consider that initial conditions for states x_0 can be treated as symbolic parameters. Then, the states can be parametrized with help of new variable $l \in \mathbb{R}$, which ranges from $l \in [0,\ 1]$ as follows:

$$x = l x_0. \tag{3}$$

Starting from (3), the total differential of $\lambda(x)$ reads

$$\mathrm{d}\lambda(x) = \frac{\partial \lambda(x)^T}{\partial x}\mathrm{d}x = \frac{\partial \lambda(x)^T}{\partial x} x_0 \mathrm{d}l. \tag{4}$$

Exploiting the fact that the matrix $\partial\lambda(x)^T/\partial x$ is symmetric, its non-diagonal elements can be replaced with new variables

$$v_k = \frac{\partial \lambda_i}{\partial x_j}, \tag{5}$$

where the number k depends on the dimension of the state. Substituting the diagonal elements of the matrix $\partial\lambda(x)^T/\partial x$ in (2a) by (4) one obtains a system of equations written in a condensed form by

$$0 = \mathrm{diag}(f)\frac{\mathrm{d}\lambda}{\mathrm{d}l} + \mathrm{diag}(x_0)\left[\left(\frac{\partial f}{\partial x}\right)^T \lambda + \frac{\partial F}{\partial x}\right] + R(f, x_0)v, \tag{6a}$$

$$0 = \frac{\partial F}{\partial u} + \frac{\partial f^T}{\partial u}\lambda. \tag{6b}$$

where the state is parametrized by (3), $v = [v_1, \ldots, v_k]^T$ is a vector of mixed partial derivatives (5) with dimension $k = 0.5n(n-1)$, and $R(f, x_0)$ is a matrix built by elements of vectors f, x_0, e.g. in the 2-dimensional case (i.e. $n = 2$ and $k = 1$), the matrix takes the form

$$R(f, x_0) = \mathrm{diag}(x_0)\bar{f} - \mathrm{diag}(f)\bar{x}_0,$$

where the bar denotes the reverted vectors $\bar{f} = [f_2,\ f_1]^T$, $\bar{x}_0 = [x_{02},\ x_{01}]^T$. Function arguments are omitted for brevity. The vector of initial conditions x_0 is left as a parameter while the unknown variables v are to be determined.

3.3 Shortest Path DOP

The parametrization (3) of the HJB equation (2) thus results in a set of DAE equations (6a) and (6b). These equations have been obtained by substituting the off-diagonal elements of $\partial\lambda(x)/\partial x$ by new unknown variables v. Obviously, such substitution is not complete, since the fully determined system of HJB equations is now undetermined, that is, contains more unknowns than equations. To be able to find the remaining unknowns, the concept of CLF is adopted [12]. As shown by [11, 13], the CLF approach imposes additional constraints to the DOP formulation, which help to specify the solution more closely. Here we pursue the same idea, but we formulate the DOP over a parameter domain.

Simply speaking, the function $V(x)$ is called CLF if it fulfills three fundamental conditions: (i) it is radially unbounded, (ii) the time derivative along the system trajectories is negative, and (iii) it is positive semi-definite. These conditions can be incorporated to our approach as follows. The third property is automatically satisfied by the selection of the cost function as $F(x,u) > 0$. Second property is fulfilled by the zero Hamiltonian, i.e. $F(x,u) + \lambda^T f(x,u) = 0$ and since $F(x,u) > 0$ then $\lambda^T f(x,u)$ (time derivative of CLF along system trajectory) is negative. The first property is enforced via additional constraint, which requires the value function to be increasing along the path l, i.e.

$$\frac{\mathrm{d}V}{\mathrm{d}l} = \frac{\partial V}{\partial x}^T \frac{\mathrm{d}x}{\mathrm{d}l} = \lambda^T x_0 > 0. \tag{7}$$

The transformed DOP can be formulated as follows:

$$\text{find } v(l)_{l\in[0,1]} \tag{8a}$$

$$\text{s.t.} \quad 0 = \operatorname{diag}(f)\frac{\mathrm{d}\lambda}{\mathrm{d}l} + \operatorname{diag}(x_0)\left[\left(\frac{\partial f}{\partial x}\right)^T \lambda + \frac{\partial F}{\partial x}\right] + R(f,x_0)v, \tag{8b}$$

$$0 = \frac{\partial f^T}{\partial u}\lambda + \frac{\partial F}{\partial u}, \tag{8c}$$

$$0 < \lambda^T x_0, \tag{8d}$$

$$0 = \lambda(0), \tag{8e}$$

$$0 = u(0), \tag{8f}$$

which constructs a CLF along the path $l \in [0,1]$ for a given initial condition x_0. This problem should be solved at each sampling instant, as a suboptimal replacement of the infinite horizon DOP (1). As our primal goal is to solve the optimization

problem as quickly as possible, in the next section we show how to transform the problem (8) to a set of DAE equations, solution to which can be found on-line by mere integration.

4 Parametric Solutions for 2D Cases

A two dimensional case with one input is of special interest here, because one equation can make the undetermined system (6) fully determined. By differentiating the equation (6b) with respect to l one obtains

$$0 = \left(\frac{\partial^2 F}{\partial u^2} + \lambda^T \frac{\partial^2 f}{\partial u^2}\right)\frac{\mathrm{d}u}{\mathrm{d}l} + \frac{\partial^2 F}{\partial u \partial l} + \frac{\partial f^T}{\partial u}\frac{\mathrm{d}\lambda}{\mathrm{d}l} + \lambda^T \frac{\partial^2 f}{\partial u \partial l}, \tag{9}$$

which creates an implicit constraint for v. Exact solution can be recovered by substituting $\mathrm{d}\lambda/\mathrm{d}l$ from (6a) to (9).

Equations (6a), (6b) and (9) now form 'parametric' solution to original DOP (1) in the sense that, for a given parameter x_0, the value $u(x_0)$ can be obtained by simple integration. Thus, the whole optimization problem with infinite-time horizon (1) is reduced to an initial value problem which can be solved very fast using standard integration algorithms. Overall procedure consists of two steps. First, the HJB equations (2a), (2b) are parametrized to obtain DAE equations (6a), (6b), and (9). This step can be performed off-line. Then, in the on-line implementation phase, the optimal control action is found by forward integration of (6a), (6b), (9) for a given x_0.

The benefit of this approach is that the computational burden involved in the on-line part relies only on **one integration** of the DAE system (6a), (6b) and (9). However, the extension to higher dimensional cases is not trivial as the number of additional equations determining v grow with the dimension of state quadratically. Furthermore, additional assumptions are required in order to guarantee existence of a solution and this is a subject of further research.

5 Illustrative Example

Consider the following example adopted from [13]:

$$\min_{u(t)} \int_0^\infty (x_2^2 + u^2)\mathrm{d}t \tag{10a}$$

$$\text{s.t. } \dot{x}_1 = x_2, \tag{10b}$$

$$\dot{x}_2 = -x_1 e^{x_1} + 0.5x_2^2 + e^{x_1} u \tag{10c}$$

where $x(0) = x_0$. For this setup, the analytic optimal solution exists and is given by

$$u = -x_2, \quad V(x) = x_1^2 + x_2^2 e^{-x_1}. \tag{11}$$

The aim of this section is to compare this explicit solution with the parametric solution, obtained by parametrizing the HJB equation. First, optimality conditions for

the unconstrained problem (10) are obtained. Then, the parametrization $x_1 = x_{01}l$ and $x_2 = x_{02}l$ gives

$$
\begin{aligned}
0 &= x_{02}l\lambda_1' + x_{01}(-e^{x_{01}l} - x_{01}le^{x_{01}l} + e^{x_{01}l}u)\lambda_2 - r(f,x_0)v,\\
0 &= (-x_{01}le^{x_{01}l} + 0.5x_{02}^2l^2 + e^{x_{01}l}u)\lambda_2' + x_{02}\lambda_1 + x_{02}^2l\lambda_2 + 2x_{02}^2l + r(f,x_0)v,\\
0 &= 2u + \lambda_2 e^{x_{01}l},
\end{aligned}
$$

where $r(f,x_0) = x_{02}^2l - x_{01}(-x_{01}le^{x_{01}l} + 0.5x_{02}^2l^2 + e^{x_{01}l}u)$. By applying (9), the expression for v can be found analytically, but is omitted due to space restrictions. Substituting for v back, the following DAE system is obtained:

$$
\begin{aligned}
0 &= x_{02}l\lambda_1' - \lambda_2 x_{01}e^{x_{01}l} + x_{02}\lambda_1 + x_{02}^2l\lambda_2 + 2x_{02}^2l\\
&\quad + 2u'x_{01}l - e^{-x_{01}l}u'x_{02}^2l^2 - 2u'u - 0.5\lambda_2 x_{01}x_{02}^2l^2, && \text{(12a)}\\
0 &= -\lambda_2'x_{01}le^{x_{01}l} + 0.5\lambda_2'x_{02}^2l^2 + \lambda_2'e^{x_{01}l}u - 2u'x_{01}l + e^{-x_{01}l}u'x_{02}^2l^2\\
&\quad + 2u'u - \lambda_2 x_{01}^2e^{x_{01}l}l + 0.5\lambda_2 x_{01}x_{02}^2l^2 + \lambda_2 x_{01}e^{x_{01}l}u, && \text{(12b)}\\
0 &= 2u + \lambda_2 e^{x_{01}l}. && \text{(12c)}
\end{aligned}
$$

Note that the DAE system (12) is of index 2. To remedy this, a new additional variable $z = u'$ is introduced, which leads (12) to

$$
\begin{aligned}
0 &= x_{02}l\lambda_1' - \lambda_2 x_{01}e^{x_{01}l} + x_{02}\lambda_1 + x_{02}^2l\lambda_2 + 2x_{02}^2l\\
&\quad + 2zx_{01}l - e^{-x_{01}l}zx_{02}^2l^2 - 2zu - 0.5\lambda_2 x_{01}x_{02}^2l^2, && \text{(13a)}\\
0 &= -\lambda_2'x_{01}le^{x_{01}l} + 0.5\lambda_2'x_{02}^2l^2 + \lambda_2'e^{x_{01}l}u - 2zx_{01}l + e^{-x_{01}l}zx_{02}^2l^2\\
&\quad + 2zu - \lambda_2 x_{01}^2e^{x_{01}l}l + 0.5\lambda_2 x_{01}x_{02}^2l^2 + \lambda_2 x_{01}e^{x_{01}l}u, && \text{(13b)}\\
0 &= 2u + \lambda_2 e^{x_{01}l}, && \text{(13c)}\\
0 &= 2z + \lambda_2'e^{x_{01}l} + \lambda_2 x_{01}e^{x_{01}l}. && \text{(13d)}
\end{aligned}
$$

which is now an index-1 DAE parametrized in the initial state x_0. Integrating (13) along the path $l = [0,1]$ yields the same solution as solving the DOP problem (10) for x_0. This equivalence can be proved by substituting the explicit solution (11) to (12). Using the SUNDIALS Toolbox [6], the DAE system was integrated on-line for a changing value of x_0 with sampling time 0.1 s and the closed loop control is depicted in Fig. 1. For this path, the HJB solution gives $V = 33.9283$, while the optimal controller (11) results in $V^* = 33.9166$. This shows very good performance of the proposed scheme. The minor difference is attributed to numerical issues of the integration routine. What is more interesting is the elapsed CPU time needed to perform the on-line DAE integration. In the worst case, the integration took as much as 0.7872 s for $x_0 = [-2,\ 2]^T$. This number was steadily decreasing as the states approached the origin. If, on the other hand, the original DOP (10a) would be solved by approximating the infinite-time horizon by a horizon of 5 s, using 50 equal discretizations steps, the optimization would take 27.27 s on the same computer using the DOTcvp Toolbox [7].

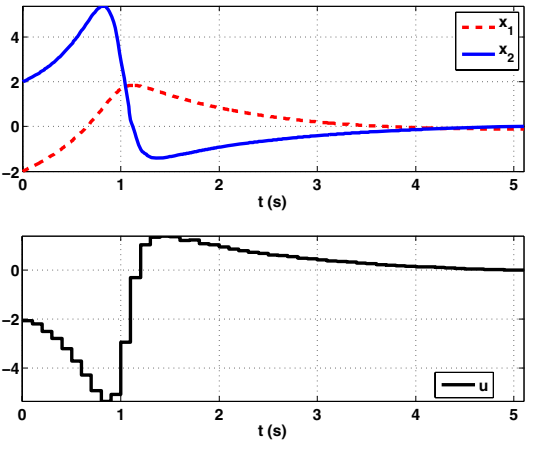


Fig. 1 NMPC control with online integration of DAE system (13)

6 Conclusion

In this work parametrization of HJB equations in the state-space domain along the shortest path was used to transform an infinite-time horizon dynamic optimization problem into a simpler problem. It was shown that for a two-dimensional case, this procedure leads to a set of DAE equations, parametrized in the initial conditions. The optimal solution can then be obtained by a mere integration of these differential equations. Issues of feasibility of the proposed scheme, as well as complexity issues, will be subject to future research.

Acknowledgements. The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under the grants 1/0071/09 and 1/4055/07. This work was supported by the Slovak Research and Development Agency under the contracts No. VV-0029-07 and No. LPP-0092-07.

References

[1] Biegler, L.: Efficient solution of dynamic optimization and NMPC problems. In: Allgöwer, F., Zheng, A. (eds.) Nonlinear Model Predictive Control, pp. 219–244. Birkhäuser, Basel (2000)
[2] Borrelli, F.: Constrained Optimal Control of Linear and Hybrid Systems. LNCIS, vol. 290. Springer, Heidelberg (2003)
[3] Cannon, M., Kouvaritakis, B., Lee, Y.I., Brooms, A.C.: Efficient non-linear model based predictive control. International Journal of Control 74(4), 361–372 (2001)
[4] Chen, W.-H., Ballance, D.J., Gawthrop, P.J.: Optimal control of nonlinear systems: a predictive control approach. Automatica 39, 633–641 (2003)
[5] Diehl, M., Findeisen, R., Nagy, Z., Bock, H.G., Schlöder, J.P., Allgöwer, F.: Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. Jour. of Process Control (2002)

[6] Hindmarsh, A.C., Brown, P.N., Grant, K.E., Lee, S.L., Serban, R., Shumaker, D.E., Woodward, C.S.: SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. ACM Transactions on Mathematical Software 31(3), 363–396 (2005), https://computation.llnl.gov/casc/sundials/main.html
[7] T. Hirmajer, E. Balsa-Canto, and J. R. Banga. DOTcvpSB: a matlab toolbox for dynamic optimization in systems biology. Technical report, Instituto de Investigaciones Marinas - CSIC, Vigo, Spain (October 2008), http://www.iim.csic.es/~dotcvpsb/
[8] Johansen, T.A.: Approximate explicit receding horizon control of constrained nonlinear systems. Automatica 40(2), 293–300 (2004)
[9] Maciejowski, J.M.: Predictive Control with Constraints. Prentice-Hall, Englewood Cliffs (2002)
[10] Nevistic, V., Primbs, J.A.: Constrained nonlinear optimal control: a converse HJB approach. Technical report, ETH Zürich (1996), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.2763
[11] Primbs, J.A., Nevistic, V., Doyle, J.: Nonlinear optimal control: A control lyapunov function and receding horizon perspective. Asian Journal of Control 1(1), 14–24 (1999)
[12] Sontag, E.D.: Mathematical Control Theory. Springer, Heidelberg (1998)
[13] Sznaier, M., Cloutier, J.: Model predictive control of nonlinear parameter varying systems via receding horizon control lyapunov functions. In: Kouvaritakis, B. (ed.) Nonlinear Model Based Predictive Control, London. IEE Control Engineering Series, vol. 61 (2002)
[14] Víteček, A., Vítečková, M.: Optimální Systémy Řízení (Optimal Control Systems). VŠT–Technická Univerzita Ostrava (2002)

Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation

Moritz Diehl, Hans Joachim Ferreau, and Niels Haverbeke

Abstract. This overview paper reviews numerical methods for solution of optimal control problems in real-time, as they arise in nonlinear model predictive control (NMPC) as well as in moving horizon estimation (MHE). In the first part, we review numerical optimal control solution methods, focussing exclusively on a discrete time setting. We discuss several algorithmic "building blocks" that can be combined to a multitude of algorithms. We start by discussing the sequential and simultaneous approaches, the first leading to smaller, the second to more structured optimization problems. The two big families of Newton type optimization methods, Sequential Quadratic Programming (SQP) and Interior Point (IP) methods, are presented, and we discuss how to exploit the optimal control structure in the solution of the linear-quadratic subproblems, where the two major alternatives are "condensing" and band structure exploiting approaches. The second part of the paper discusses how the algorithms can be adapted to the real-time challenge of NMPC and MHE. We recall an important sensitivity result from parametric optimization, and show that a tangential solution predictor for online data can easily be generated in Newton type algorithms. We point out one important difference between SQP and IP methods: while both methods are able to generate the tangential predictor for fixed active sets, the SQP predictor even works across active set changes. We then classify many proposed real-time optimization approaches from the literature into the developed categories.

Keywords: real-time optimization, numerical optimal control, Newton type methods, structure exploitation.

Moritz Diehl, Hans Joachim Ferreau, and Niels Haverbeke
Optimization in Engineering Center (OPTEC) and ESAT-SCD,
K.U. Leuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium
e-mail: {moritz.diehl,joachim.ferreau,niels.haverbeke}@esat.kuleuven.be

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 391–417.
springerlink.com

1 Introduction

Nonlinear optimal control algorithms are at the core of all nonlinear MPC or moving horizon estimation (MHE) schemes. In contrast to linear MPC, where convex quadratic programs are mostly solved exactly at each sampling time, nonlinear MPC faces a dilemma: either the nonlinear iteration procedure is performed until a pre-specified convergence criterion is met, which might introduce considerable feedback delays, or the procedure is stopped prematurely with only an approximate solution, so that a pre-specified computation time limit can be met. Fortunately, considerable progress has been achieved in the last decade that allows to reduce both, computational delays and approximation errors. This progress would not have been possible by using just off-the-shelf optimal control codes; it is the development of dedicated real-time optimization algorithms for NMPC and MHE that allows to nowadays apply NMPC to plants with tens of thousands of states or to mechatronic applications.

While several excellent numerical optimization textbooks exist [25, 28, 44], in the field of numerical *optimal control* there are only a few [2, 11], and when it comes to real-time optimal control algorithms there is even less overview material [5]. The aim of the present article is to help closing this gap and to summarize the state-of-the-art in this field by presenting those algorithmic ideas that appear to be crucial to the authors. We choose a rather simplified setting, leaving many important special cases aside, in order to present the major ideas as clearly as possible.

The article is organized as follows: In Section 2 the NMPC and MHE problems are stated, in Section 3 we review Newton type optimization methods of different flavor, and in Section 4 we discuss how to exploit the optimal control structure of the linear equation systems to be solved in each Newton type iteration. In Section 5 we present online initialization strategies for subsequent NMPC problems, and in Section 6 the online algorithms of different flavours are discussed, and we finally conclude the paper in Section 7.

2 Problem Formulation

Throughout this paper we regard discrete time dynamical systems augmented with algebraic equations, as follows:

$$x_{k+1} = f_k(x_k, z_k, u_k) \tag{1a}$$

$$g_k(x_k, z_k, u_k) = 0 \tag{1b}$$

Here, $x_k \in \mathbb{R}^{n_x}$ is the differential state, $z_k \in \mathbb{R}^{n_z}$ the algebraic state, and $u_k \in \mathbb{R}^{n_u}$ is the control. Functions f_k and g_k are assumed twice differentiable and map into $\mathbb{R}^{n_x}$ and $\mathbb{R}^{n_z}$, respectively. The algebraic state z_k is uniquely determined by (1b) when x_k and u_k are fixed, as we assume that $\frac{\partial g_k}{\partial z}$ is invertible everywhere.

We choose to regard this difference-algebraic system form because it covers several parametrization schemes for continuous time dynamic systems in differential algebraic equation (DAE) form, in particular direct multiple shooting with DAE relaxation [39] and direct collocation [3, 59]. Note that in the case of collocation, all collocation equations on a collocation interval would be collected within the function g_k and the collocation node values in the variables z_k.

2.1 NMPC Optimal Control Problem

Based on this dynamic system form, we regard the following simplified optimal control problem in discrete time:

$$\begin{aligned}
\underset{x,\,z,\,u}{\text{minimize}} \quad & \sum_{i=0}^{N-1} L_i(x_i, z_i, u_i) \quad + \quad E\left(x_N\right) && \text{(2a)}\\
\text{subject to} \quad & x_0 - \bar{x}_0 = 0, && \text{(2b)}\\
& x_{i+1} - f_i(x_i, z_i, u_i) = 0, \quad i = 0, \ldots, N-1, && \text{(2c)}\\
& g_i(x_i, z_i, u_i) = 0, \quad i = 0, \ldots, N-1, && \text{(2d)}\\
& h_i(x_i, z_i, u_i) \leq 0, \quad i = 0, \ldots, N-1, && \text{(2e)}\\
& r\left(x_N\right) \leq 0. && \text{(2f)}
\end{aligned}$$

Here, the free variables are the differential state vector $x = (x_0^T, x_1^T \ldots, x_{N-1}^T, x_N^T)^T$ at all considered time points and the algebraic and control vector on all but the last time points: $z = (z_0^T, z_1^T \ldots, z_{N-1}^T)^T$ and $u = (u_0^T, u_1^T \ldots, u_{N-1}^T)^T$.

Remark on fixed and free parameters: In most NMPC applications there are some *constant* parameters $\bar{p}$ that are assumed constant for the NMPC optimization, but that change for different problems, like $\bar{x}_0$. We do not regard them here for notational convenience, but note that they can be regarded as constant system states with fixed initial value $\bar{p}$. In some NMPC applications *free* parameters p exist that are part of the optimization variables, but that are – in contrast to the controls u_k – constant in time. Again, we disregard this case for notational simplicity.

2.2 Moving Horizon Estimation: Nearly a Dual Problem

For moving horizon estimation (MHE), see e.g. [21, 48, 65], we typically choose convex functions to penalize the mismatch between the real measurements y_k and the corresponding model predictions $m_k(x_k, z_k, u_k, w_k)$. For notational simplicity, we regard only weighted Euclidean norms here, $\|y_k - m_k(x_k, z_k, u_k, w_k)\|_Q^2$, but point out that it is often useful to regard

other penalty functions like e.g. the ℓ_1 penalty, which necessitate slight adaptations in the numerical solution algorithms presented later. The controls u_k are here regarded as fixed and known and enter the system dynamics only as constant but time varying parameters. However, time varying disturbances w_k are often introduced in the MHE problem to account for plant-model mismatch. They take the same role as the controls in the NMPC problem and are often ℓ_2 penalized.

$$\underset{x,\,z,\,w}{\text{minimize}} \quad \|x_0 - \bar{x}_0\|_P^2 \;+\; \sum_{i=0}^{N-1} \|y_i - m_i(x_i, z_i, u_i, w_i)\|_Q^2 + \|w_i\|_R^2 \tag{3a}$$

subject to

$$x_{i+1} - f_i(x_i, z_i, u_i, w_i) = 0, \quad i = 0, \ldots, N-1, \tag{3b}$$

$$g_i(x_i, z_i, u_i, w_i) = 0, \quad i = 0, \ldots, N-1, \tag{3c}$$

$$h_i(x_i, z_i, u_i, w_i) \leq 0, \quad i = 0, \ldots, N-1, \tag{3d}$$

Due to the fact that the MHE problem has the same optimal control structure as the NMPC problem, they are often called "dual" to each other, in a slight abuse of terminology. However, the starkest contrast to the NMPC problem is the fact that the MHE problem has a free initial value x_0 and often has a much higher dimensional "control vector" w_k. This necessitates possibly different linear algebra solvers in the solution procedures described below.

2.3 Sequential vs. Simultaneous Optimal Control

For simplicity of presentation, we will in this subsection only focus on the NMPC problem (2a)-(2f). Here, the equality constraints (2b)-(2d) uniquely determine the variables x and z if the vector u is fixed. Thus, they can be inverted to yield the implicit functions $\tilde{x}(u)$ and $\tilde{z}(u)$ that satisfy (2b)-(2d) for all u, by a system simulation. It allows to reduce the optimization problem to

$$\underset{u}{\text{minimize}} \quad \sum_{i=0}^{N-1} L_i(\tilde{x}_i(u), \tilde{z}_i(u), u_i) \;+\; E\left(\tilde{x}_N(u)\right) \tag{4a}$$

$$\text{subject to} \quad h_i(\tilde{x}_i(u), \tilde{z}_i(u), u_i) \leq 0, \quad i = 0, \ldots, N-1, \tag{4b}$$

$$r\left(\tilde{x}_N(u)\right) \leq 0. \tag{4c}$$

This problem has a strongly reduced variable space compared to the original problem, and it is thus an appealing idea to use the reduced problem within an optimization procedure. This gives rise to the so called "sequential" approach to optimal control problems, where in each optimization iteration the two steps, system simulation and optimization, are performed sequentially, one after the other. This approach emerged early in the nonlinear optimal control literature [50].

In contrast to the sequential approach, the so called "simultaneous" approach addresses the full nonlinear program as stated above in (2a)-(2f) directly by a Newton type optimization algorithm, i.e., optimization and simulation are performed simultaneously. It comes in the form of direct collocation methods [3, 59, 64] as well as in form of direct multiple shooting [9, 39].

The optimization problem of the sequential approach has much less variables, but also less structure in the linear subproblems than the simultaneous approach (an interesting structure preserving sequential algorithm was however presented in [58]). Even more important, the Newton type optimization procedure behaves quite differently for both approaches: typically, faster local convergence rates are observed for the simultaneous approach, in particular for unstable or highly nonlinear systems, because – intuitively speaking – the nonlinearity is equally distributed over the nodes.

3 Newton Type Optimization

Newton's method for solution of a nonlinear equation $R(W) = 0$ starts with an initial guess W^0 and generates a series of iterates W^k that each solves a linearization of the system at the previous iterate, i.e., for given W^k the next iterate W^{k+1} shall satisfy $R(W^k) + \nabla R(W^k)^T(W^{k+1} - W^k) = 0$. The hope is that the linearizations – that can be solved w.r.t. W^{k+1} by standard linear algebra tools – are sufficiently good approximations of the original nonlinear system and that the iterates converge towards a solution W^*. Newton's method has locally a quadratic convergence rate, which is as fast as making any numerical analyst happy. If the Jacobian $\nabla R(W^k)^T$ is not computed or inverted exactly, this leads to slower convergence rates, but cheaper iterations, and gives rise to the larger class of "Newton type methods". An excellent overview of the field is given in [13]. But how are these ideas generalized to nonlinear optimization?

The NMPC and MHE problems as stated above are specially structured cases of a generic nonlinear program (NLP) that has the form

$$\underset{X}{\text{minimize}}\, F(X) \quad \text{s.t.} \quad \begin{cases} G(X) = 0 \\ H(X) \leq 0 \end{cases} \tag{5}$$

Under mild assumptions, any locally optimal solution X^* of this problem has to satisfy the famous Karush-Kuhn-Tucker (KKT) conditions: there exist multiplier vectors λ^* and μ^* so that the following equations hold:

$$\nabla_X \mathcal{L}(X^*, \lambda^*, \mu^*) = 0 \tag{6a}$$

$$G(X^*) = 0 \tag{6b}$$

$$0 \;\geq\; H(X^*) \quad \perp \quad \mu^* \geq 0. \tag{6c}$$

Here we have used the definition of the Lagrange function

$$\mathcal{L}(X, \lambda, \mu) = F(X) + G(X)^T \lambda + H(X)^T \mu \tag{7}$$

and the symbol $\perp$ between the two vector valued inequalities in Eq. (6c) states that also the complementarity condition

$$H_i(X^*)\, \mu_i^* = 0, \quad i = 1, \ldots, n_H, \tag{8}$$

shall hold. All Newton type optimization methods try to find a point satisfying these conditions by using successive linearizations of the problem functions. Major differences exist, however, on how to treat the last condition (6c) that is due to the inequality constraints, and the two big families are Sequential Quadratic Programming (SQP) type methods and Interior Point (IP) methods.

3.1 Sequential Quadratic Programming

A first variant to iteratively solve the KKT system is to linearize all nonlinear functions appearing in Eqs. (6a)–(6c). It turns out that the resulting linear complementarity system can be interpreted as the KKT conditions of a quadratic program (QP)

$$\underset{X}{\text{minimize}}\, F_{\text{QP}}^k(X) \quad \text{s.t.} \quad \begin{cases} G(X^k) + \nabla G(X^k)^T (X - X^k) = 0 \\ H(X^k) + \nabla H(X^k)^T (X - X^k) \le 0 \end{cases} \tag{9}$$

with objective function

$$F_{\text{QP}}^k(X) = \nabla F(X^k)^T X + \frac{1}{2}(X - X^k)^T \nabla_X^2 \mathcal{L}(X^k, \lambda^k, \mu^k)(X - X^k). \tag{10}$$

In the case that the so called Hessian matrix $\nabla_X^2 \mathcal{L}(X^k, \lambda^k, \mu^k)$ is positive semidefinite, this QP is convex so that global solutions can be found reliably. This general approach to address the nonlinear optimization problem is called Sequential Quadratic Programming (SQP). Apart from the presented "exact Hessian" SQP variant presented above, several other – and much more widely used – SQP variants exist, that make use of inexact Hessian or Jacobian matrices.

3.1.1 Powell's Classical SQP Method

One of the most successfully used SQP variants is due to Powell [47]. It uses exact constraint Jacobians, but replaces the Hessian matrix $\nabla_X^2 \mathcal{L}(X^k, \lambda^k, \mu^k)$ by an approximation A_k. Each new Hessian approximation A_{k+1} is obtained from the previous approximation A_k by an update formula that uses the difference of the Lagrange gradients, $\gamma = \nabla_X \mathcal{L}(X^{k+1}, \lambda^{k+1}, \mu^{k+1}) - \nabla_X \mathcal{L}(X^k, \lambda^{k+1}, \mu^{k+1})$ and the step $\sigma = X^{k+1} - X^k$. Aim of these

"Quasi-Newton" or "Variable-Metric" methods is to collect second order information in A_{k+1} by satisfying the secant equation $A_{k+1}\sigma = \gamma$. The most widely used update formula is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update $A_{k+1} = A_k + \gamma\gamma^T/(\gamma^T\sigma) - A_k\sigma\sigma^T A_k/(\sigma^T A_k\sigma)$, see e.g. [44]. Quasi-Newton methods can be shown to converge superlinearly under mild conditions, and had a tremendous impact in the field of nonlinear optimization. Successful implementations are the packages NPSOL and SNOPT for general NLPs [27], and MUSCOD-II [39] for optimal control. Note that in this paper we omit all discussion on the usually crucial issue of globalisation strategies, because these are less important in online optimization.

3.1.2 Constrained Gauss-Newton Method

Another particularly successful SQP variant – the Constrained (or Generalized) Gauss-Newton method – is also based on approximations of the Hessian. It is applicable when the objective function is a sum of squares:

$$F(X) = \frac{1}{2}\|R(X)\|_2^2. \tag{11}$$

In this case, the Hessian can be approximated by

$$A_k = \nabla R(X^k)\nabla R(X^k)^T \tag{12}$$

and the corresponding QP objective is easily seen to be

$$F_{\mathrm{QP}}^k(X) = \frac{1}{2}\|R(X^k) + \nabla R(X^k)^T(X - X^k)\|_2^2 \tag{13}$$

The constrained Gauss-Newton method has only linear convergence but often with a surprisingly fast contraction rate. The contraction rate is fast when the residual norm $\|R(X^*)\|$ is small or the problem functions R, G, H have small second derivatives. It has been developed and extensively investigated by Bock and coworkers, see e.g. [6, 53]. The constrained Gauss-Newton method is implemented in the packages PARFIT [6], FIXFIT [53], and also as one variant within MUSCOD-II [14, 39].

Remark on adjoint based SQP variants: Newton type SQP methods may not only use an approximation of the Hessian, but also of the constraint Jacobians. The most general formulation including inexact inequalities, which is originally due to [7] and was analysed in [61], uses approximations A_k, B_k, C_k of the matrices $\nabla_X^2\mathcal{L}(\cdot), \nabla G(X^k), \nabla H(X^k)$, and a so called "modified gradient"

$$a_k = \nabla_X\mathcal{L}(X^k, \lambda^k, \mu^k) - B_k\lambda^k - C_k\mu^k \tag{14}$$

in the QP objective

$$F^k_{\text{adjQP}}(X) = a_k^T X + \frac{1}{2}(X - X^k)^T A_k (X - X^k). \tag{15}$$

The following QP is solved in each iteration:

$$\underset{X}{\text{minimize}}\, F^k_{\text{adjQP}}(X) \quad \text{s.t.} \quad \begin{cases} G(X^k) + B_k^T (X - X^k) = 0 \\ H(X^k) + C_k^T (X - X^k) \leq 0. \end{cases} \tag{16}$$

It can be shown that using a modified gradient a_k allows to locally converge to solutions of the original nonlinear NLP even in the presence of inexact inequality constraint Jacobians [7, 20, 61]. A crucial ingredient of the adjoint based SQP scheme is the fact that the Lagrange gradient needed for a_k in (14) can be evaluated efficiently by adjoint based techniques or, equivalently, by the reverse mode of automatic differentiation [30]. Adjoint based SQP schemes are at the core of the multi-level real-time iterations described in Section 6.1. Even quasi Newton update schemes can be used in order to approximate the Jacobians [32].

3.2 Interior Point Methods

In contrast to SQP methods, an alternative way to address the solution of the KKT system is to replace the last nonsmooth KKT condition in Eq. (6c) by a smooth nonlinear approximation, with $\tau > 0$:

$$\nabla_X \mathcal{L}(X^*, \lambda^*, \mu^*) = 0 \tag{17a}$$

$$G(X^*) = 0 \tag{17b}$$

$$H_i(X^*)\, \mu_i^* = \tau, \quad i = 1, \ldots, n_H. \tag{17c}$$

This system is then solved with Newton's method. The obtained solution is not a solution to the original problem, but to the problem

$$\underset{X}{\text{minimize}}\, F(X) - \tau \sum_{i=1}^{n_H} \log(-H_i(X)) \quad \text{s.t.} \quad G(X) = 0. \tag{18}$$

Thus, the solution is in the interior of the set described by the inequality constraints, and closer to the true solution the smaller τ gets. The crucial feature of the family of "interior point methods" is the fact that, once a solution for a given τ is found, the parameter τ can be reduced by a constant factor without jeopardising convergence of Newton's method. After only a limited number of Newton iterations a quite accurate solution of the original NLP is obtained. We refer to the excellent textbooks [10, 63] for details.

A widely used implementation of nonlinear Interior Point methods is the open source code IPOPT [60].

Remark on the structure of the linear subproblems: It is interesting to note that the linearization of the smoothed KKT system (17a)-(17c) is a linear system that is equivalent – after elimination of the variable μ^{k+1} – to the KKT conditions of an equality constrained quadratic program. It is important to remark that most structure exploiting features of SQP type methods also have an equivalent in IP methods, like globalisation strategies, use of adjoints, structure preserving linear algebra, etc., and we will mention them when applicable.

Remark on Ohtsuka's inequality treatment: An interesting treatment of inequality constraints that is similar to interior point methods was proposed and successfully used in the context of NMPC by Ohtsuka [45]. He proposes to approximate the inequality constrained NLP (5) by a formulation

$$\underset{X,\,Y}{\text{minimize}}\; F(X) - \tau \sum_{i=1}^{n_H} Y_i \quad \text{s.t.} \quad \begin{cases} G(X)=0 \\ H_i(X) + Y_i^2 = 0, \; i = 1, \ldots, n_H. \end{cases} \tag{19}$$

which is equivalent to

$$\underset{X}{\text{minimize}}\; F(X) - \tau \sum_{i=1}^{n_H} \sqrt{-H_i(X)} \quad \text{s.t.} \quad G(X) = 0. \tag{20}$$

This barrier is not self-concordant and does not connect easily to the dualtiy theory of interior-point methods, but we will nevertheless call this approach a variant of IP methods in this paper.

4 Numerical Optimal Control

When Newton type optimization strategies are applied to the optimal control problem (2a)-(2f), the first question is, if a simultaneous or a sequential approach is used. In the case of a sequential approach, where all state variables x, z are eliminated and the optimization routine only sees the control variables u, the specific optimal control problem structure plays a minor role. Thus, often an off-the-shelf code for nonlinear optimization can be used. This makes practical implementation very easy and is a major reason why the sequential approach is used by many practitioners. It is in strong contrast to the simultaneous approach, that addresses the optimal control problem (2a)-(2f) in the full variable space x, z, u, and thus allows – and necessitates – to exploit the specific problem structure. In all Newton type optimization routines there are two crucial and often costly computational steps, namely

(a) Derivative Computation and (b) Solution of the Quadratic Subproblems. In both areas, specific structures can be exploited. In this paper we will focus on (b), the solution of the QPs, but add suitable remarks on how to treat derivative computations when necessary.

4.1 The Linearized Optimal Control Problem

Let us regard the linearization of the optimal control problem (2a)-(2f) within an SQP method, which is a structured QP. It turns out that due to the dynamic system structure the Hessian of the Lagrangian function has the same separable structure as the Hessian of the original objective function (2a), so that the quadratic QP objective is still representable as a sum of linear-quadratic stage costs, which was first observed by Bock and Plitt [9]. Thus, the QP subproblem has the following form, where we left out the SQP iteration index k for notational simplicity, and where the summands of the objective each are linear-quadratic.

$$\underset{x,\,z,\,u}{\text{minimize}} \quad \sum_{i=0}^{N-1} L_{\text{QP},i}(x_i, z_i, u_i) \quad + \quad E_{\text{QP}}\left(x_N\right) \tag{21a}$$

$$\text{subject to} \qquad x_0 - \bar{x}_0 = 0, \tag{21b}$$

$$x_{i+1} - f_i' - F_i^x x_i - F_i^z z_i - F_i^u u_i = 0, \quad i = 0, \ldots, N-1, \tag{21c}$$

$$g_i' + G_i^x x_i + G_i^z z_i + G_i^u u_i = 0, \quad i = 0, \ldots, N-1, \tag{21d}$$

$$h_i' + H_i^x x_i + H_i^z z_i + H_i^u u_i \leq 0, \quad i = 0, \ldots, N-1, \tag{21e}$$

$$r' + R x_N \leq 0. \tag{21f}$$

When the linear algebra within the QP solution is concerned, the dynamic system structure can be exploited in different ways.

Remark on high rank Hessian updates: The fact that the Hessian matrix of the optimal control problem is block diagonal does not only allow to write down the objective (21a) in a separable form and exploit this sparsity in the linear algebra; when quasi Newton Hessian update methods are used, it also allows to perform "partitioned variable metric" or "high rank updates" of the Hessian, by updating all Hessian blocks separately [9, 31].

4.2 Elimination of Algebraic Variables

We consider now several algorithmic building blocks helping to solve the QP problem (21a)-(21f). Let us first regard Eq. (21d). Due to our assumptions in the problem statement of (2a)-(2f), we know that the Jacobian matrix G_i^z is invertible. Thus, Eq. (21d) can directly be inverted by a factorization of the matrix G_i^z, yielding an explicit expression for z_i:

$$z_i = -\left(G_i^z\right)^{-1} \left[g_i' + G_i^x x_i + G_i^u u_i\right] \tag{22}$$

Note that the matrix G_i^z is often sparse and might best be factorized by a direct sparse solver. Once this factorization is performed, it is possible to reduce problem (21a)-(21f) to a smaller scale QP in the variables x and u only, which has the following form:

$$\underset{x,\,u}{\text{minimize}} \quad \sum_{i=0}^{N-1} L_{\text{redQP},i}(x_i, u_i) \quad + \quad E_{\text{QP}}\left(x_N\right) \tag{23a}$$

$$\text{subject to} \qquad x_0 - \bar{x}_0 = 0, \tag{23b}$$

$$x_{i+1} - c_i - A_i x_i - B_i u_i = 0, \quad i = 0, \ldots, N-1, \tag{23c}$$

$$\bar{h}_i + \bar{H}_i^x x_i + \bar{H}_i^u u_i \leq 0, \quad i = 0, \ldots, N-1, \tag{23d}$$

$$r' + R x_N \leq 0. \tag{23e}$$

This partially reduced QP can be post-processed either by a condensing or a band structure exploiting strategy.

Remark on Leineweber's Partially Reduced SQP Method: In the context of a direct multiple shooting method, the evaluation of the Jacobian matrices F_i^x, F_i^z, F_i^u in (21c) is a very CPU intensive step. Given the fact that finally only the reduced matrices A_i and B_i are needed in the reduced QP, Leineweber [39] proposed a partially reduced SQP method that never computes the matrices needed in the QP (21a)-(21f). Instead, it first performs the sparse matrix factorization of G_i^z needed for elimination of the variables z_i via Eq. (22), and then it computes the matrices A_i and B_i directly as directional derivatives of $f_i(x_i, z_i, u_i)$:

$$A_i = \frac{\partial f_i(\cdot)}{\partial (x,z,u)} \begin{bmatrix} \mathbb{I} \\ -(G_i^z)^{-1} G_i^x \\ 0 \end{bmatrix} \quad \text{and} \quad B_i = \frac{\partial f_i(\cdot)}{\partial (x,z,u)} \begin{bmatrix} 0 \\ -(G_i^z)^{-1} G_i^u \\ \mathbb{I} \end{bmatrix}. \tag{24}$$

This allows to reduce the computational burden significantly in case of many algebraic variables z_i and expensive evaluation of f_i.

4.3 Condensing

In order to see how the variable space of a QP can be reduced further in a very simple way, let us recall that it was possible to reduce the large scale NLP via a nonlinear system simulation in the sequential approach. The basic idea of the "condensing" approach that was first proposed by Bock and Plitt [9] is to use the same fact, but apply it only to the linearized dynamic system. For this aim let us note that Eqs. (23b) and (23c) describe nothing else than a linear time varying discrete time system, and that for fixed u the values for x can easily be obtained by a forward simulation of the linear dynamics. Hence, the vector x is completely determined by the vector u and the given

initial state $\bar{x}_0$. Therefore, the states can be eliminated from the linearized problem resulting in a smaller, but dense quadratic program of the form

$$\underset{u}{\text{minimize}} \quad f_{\text{condQP},i}(\bar{x}_0, u) \tag{25a}$$

$$\text{subject to} \quad \bar{r} + \bar{R}^{x_0}\bar{x}_0 + \bar{R}^{u}u \leq 0. \tag{25b}$$

Here, the inequalities (25b) contain both types of the original inequalities, (23d) and (23e), in "condensed" form. If the dimension of the vector $u = (u_0^T, u_1^T, \ldots, u_{N-1}^T)^T$ is not too large, this QP can be solved fast using dense general purpose QP solvers. By doing so, the cost of solving one QP subproblem grows with $O(N^3 n_u^3)$, i.e. cubically with the horizon length N.

Remark on Schlöder's Reduction Trick: In the context of direct multiple shooting methods, the generation of the matrices A_i and B_i in (23c) is expensive if the differential state dimension n_x is large. It needs $O(N(n_x + n_u))$ stage wise directional derivatives. We might instead, as in Leineweber's partially reduced SQP method, directly compute the quantities needed in the objective and the constraints of the condensed QP (25a)-(25b). This idea was first proposed by Schlöder [53], in the context of the Generalized Gauss-Newton method. The method is implemented in the codes FIXFIT [53] and MSOPT [52]. It is only advantageous for large state but small control dimensions ($n_u \ll n_x$), and it exploits the fact that the initial value x_0 is fixed in the NMPC problem. Thus, it offers no advantages in the MHE problem where the initial value is free for optimization.

4.4 Band Structure Exploiting Riccati Based Solutions

Instead of condensing the linearized problem, one can opt to keep the constraints (23b) and (23c) and the variables x as unknowns in the QP. To sketch the idea, let us regard a QP without the inequalities (23d) and (23e). The KKT conditions of this equality constrained QP (23a)-(23c) in the primal and dual variables $w = (\lambda_0^T, x_0^T, u_0^T, \lambda_1^T, x_1^T, u_1^T, \ldots, \lambda_N^T, x_N^T)^T$ are a symmetric linear system $Mw = b$ with KKT matrix

$$M = \begin{bmatrix} \mathbb{I} & & & & & \\ \mathbb{I} & Q_0 & S_0 & -A_0^T & & \\ & S_0^T & R_0 & -B_0^T & & \\ & -A_0 & -B_0 & \ddots & \mathbb{I} & \\ & & & \mathbb{I} & Q_N & \end{bmatrix} \tag{26}$$

The almost block diagonal structure of this linear system allows it to be efficiently factorized by a (discrete time) Riccati recursion. This was shown for

optimal control problems within an active set framework in [29] and within an interior point framework in [56]. For linear model predictive control, Riccati based solutions are described in [35, 49]. The cost of this factorization, which is usually dominating the cost for solving the QP, is $O(N(n_x + n_u)^3)$. The cost grows only linearly with the horizon length N, in contrast to condensing with its cubic growth $O(N^3 n_u^3)$. This makes the Riccati method favorable for larger horizon lengths N and when $n_x \approx n_u$. A Riccati based factorization is particularly advantageous for the MHE problem where the dimension of the "controls" w is typically as big as the state dimension.

Remark on direct or iterative sparse solvers: Note that it is not necessary to use a Riccati based solution in order to obtain the complexity $O(N(n_x + n_u)^3)$, but that this can also be achieved by using a direct sparse solver, as e.g. done in the general purpose and open-source NLP package IPOPT [60]. Also, iterative linear solvers might be used.

Remark on Tenny's Feasibility Perturbed SQP Method: An interesting method for optimal control and NMPC was proposed by Tenny, Wright and Rawlings [58], who regard a simultaneous formulation within an SQP type framework, but "perturb" the result of each SQP iteration in order to make the state trajectory consistent, i.e., they close all nonlinear continuity conditions (2c). This can be done by a simple "open loop" forward simulation of the system given the new controls, or by more complex "closed loop" simulations. In the open loop variant, this is nearly a sequential approach and performs, if exact Hessians are used, even the same SQP iterations. But it differs in one important aspect: it allows to exploit the same sparsity structure as a simultaneous approach, e.g. full space derivative computation, Riccati based linear algebra, or high rank updates for the block structured Hessian [9]. This makes it an interesting cross-over between typical features of sequential and simultaneous methods.

4.5 A Classification of Optimal Control Methods

It is an interesting exercise to try to classify Newton type optimal control algorithms. Let us have a look at how nonlinear optimal control algorithms perform their major algorithmic components, each of which comes in several variants:

(a) Treatment of Inequalities: Nonlinear IP vs. SQP
(b) Nonlinear Iterations: Simultaneous vs. Sequential
(c) Derivative Computations: Full vs. Reduced
(d) Linear Algebra: Banded vs. Condensing

In the last two of these categories, we observe that the first variants each exploit the specific structures of the simultaneous approach, while the second variant reduces the variable space to the one of the sequential approach. Note that reduced derivatives imply condensed linear algebra, so the combination

[Reduced,Banded] is excluded. In the first category, we might sometimes distinguish two variants of SQP methods, depending on how they solve their underlying QP problems, via active set QP solvers (SQP-AS) or via interior point methods (SQP-IP).

Based on these four categories, each with two alternatives, and one combination excluded, we obtain seven possible combinations. In these categories, the classical single shooting method [50] could be classified as [SQP,Sequential,Reduced] or as [SQP,Sequential,Full,Condensing] because some variants compute directly the reduced derivatives $\bar{R}^u$ in (25b), while others compute first the matrices A_i and B_i in (23c) and condense then. Tenny's feasibility perturbed SQP method [58] could be classified as [SQP,Sequential,Full,Banded], and Bock's multiple shooting [9] as well as the classical reduced SQP collocation methods [2, 3, 59] as [SQP,Simultaneous,Full,Condensing]. The band structure exploiting SQP variants from Steinbach [56] and Franke [26] are classified as [SQP-IP,Simultaneous,Full,Banded], while the widely used interior point direct collocation method in conjunction with IPOPT by Biegler and Wächter [60] as [IP,Simultaneous,Full,Banded]. The reduced Gauss-Newton method of Schlöder [53] would here be classified as [SQP,Simultaneous,Reduced].

5 Online Initialization and NLP Sensitivities

For exploiting the fact that NMPC requires the solution of a whole sequence of "neighboring" NLPs and not just a number of stand-alone problems, we have first the possibility to *initialize* subsequent problems efficiently based on previous information. In this section we introduce several concepts for such initializations, in particular the important concept of NLP sensitivities. On the other hand, in Section 6 we will give an overview of specially tailored *online algorithms* for approximately solving each NLP, that deliver on purpose inaccurate solutions and postpone calculations from one problem to the next.

5.1 Shift Initialization

A first and obvious way to transfer solution information from one solved NMPC problem to the initialization of the next one is based on the principle of optimality of subarcs, also called the dynamic programming principle. It states the following: Let us assume we have computed an optimal solution $(x_0^*, z_0^*, u_0^*, x_1^*, z_1^*, u_1^*, \ldots, x_N^*)$ of the NMPC problem (2a)-(2f) starting with initial value $\bar{x}_0$. If we regard a shortened NMPC problem without the first interval, which starts with the initial value $\bar{x}_1$ chosen to be x_1^*, then for this shortened problem the vector $(x_1^*, z_1^*, u_1^*, \ldots, x_N^*)$ is the optimal solution.

Based on the expectation that the measured or observed true initial value for the shortened NMPC problem differs not much from x_1^* – i.e. we believe

our prediction model and expect no disturbances – this "shrinking" horizon initialization is canonical, and it is used in MPC of batch or finite time processes, see e.g. [15, 34].

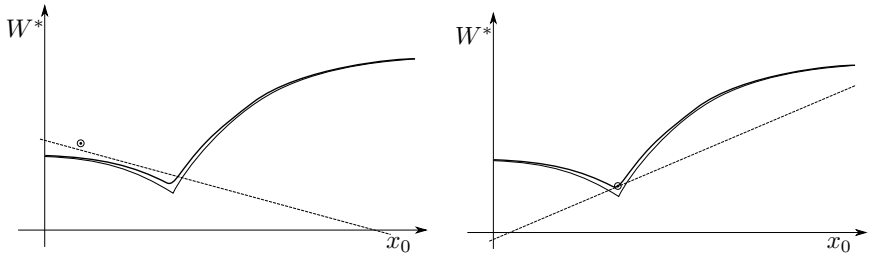
However, in the case of moving (finite) horizon problems, the horizon is not only shortened by removing the first interval, but also prolonged at the end by appending a new terminal interval – i.e. the horizon is moved forward in time. In the moving horizon case, the principle of optimality is thus not strictly applicable, and we have to think about how to initialize the appended new variables z_N, u_N, x_{N+1}. Often, they are obtained by setting $u_N := u_{N-1}$ or setting u_N as the steady state control. The states z_N and x_{N+1} are then obtained by forward simulation. This transformation of the variables from one problem to the next is called "shift initialization". It is not as canonical as the "shrinking horizon" case, because the shifted solution is not optimal for the new undisturbed problem. However, in the case of long horizon lengths N we can expect the shifted solution to be a good initial guess for the new solution. Moreover, for most NMPC schemes with stability guarantee (for an overview see e.g. [42]) there exists a canonical choice of u_N that implies feasibility (but not optimality) of the shifted solution for the new, undisturbed problem. The shift initialization is very often used e.g. in [4, 19, 41, 43].

A comparison of shifted vs. non-shifted initializations was performed in [8] with the result that for autonomous NMPC problems that shall regulate a system to steady state, there is usually no advantage of a shift initialization compared to the "primitive" warm start initialization that leaves the variables at the previous solution. In the extreme case of short horizon lengths, it turns out to be even advantageous NOT to shift the previous solution, as subsequent solutions are less dominated by the initial values than by the terminal conditions. On the other hand, shift initialization are a crucial prerequisite in periodic tracking applications [19] and whenever the system or cost function are not autonomous.

5.2 Parametric Sensitivities

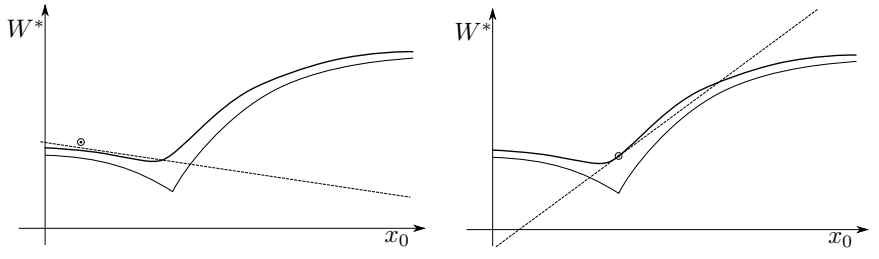
In the shift initialization discussed above we did assume that the new initial value corresponds to the model prediction. This is of course never the case, because exactly the fact that the initial state is subject to disturbances motivates the use of MPC. By far the most important change from one optimization problem to the next one are thus the unpredictable changes in the initial value. Is there anything we can do about this in the initialization of a new problem?

It turns out that we can, if we use the concept of parametric NLP sensitivities to construct a new initial guess. To illustrate the idea, let us first regard the parametric root finding problem $R(\bar{x}_0, W) = 0$ that results from the necessary optimality conditions of an IP method, i.e. the system (17a)–(17c) in variables $W = (X, \lambda, \mu)$. In the NMPC context, this system depends on the uncertain initial value $\bar{x}_0$. We denote the solution manifold by $W^*(\bar{x}_0)$. When



(a) Linearizing at approximate solution (b) Linearizing at active set change

Fig. 1 Tangential predictors for interior point method using a small τ



(a) Linearizing at approximate solution (b) Linearizing at active set change

Fig. 2 Tangential predictors for interior point method using a larger τ

we know the solution $W = W^*(\bar{x}_0)$ for a previous initial value $\bar{x}_0$ and want to compute the solution for a new initial value $\bar{x}_0'$, then a good solution predictor for $W^*(\bar{x}_0')$ is provided by $W' = W + \frac{dW^*}{d\bar{x}_0}(\bar{x}_0)(\bar{x}_0' - \bar{x}_0)$ where $\frac{dW^*}{d\bar{x}_0}(\bar{x}_0)$ is given by the implicit function theorem. An important practical observation is that an approximate tangential predictor can also be obtained when it is computed at a point W that does not exactly lie on the solution manifold. This more general predictor is given by the formula

$$W' = W - \left(\frac{\partial R}{\partial W}(\bar{x}_0, W)\right)^{-1} \left[\frac{\partial R}{\partial \bar{x}_0}(\bar{x}_0, W)\ \left(\bar{x}_0' - \bar{x}_0\right) + R(\bar{x}_0, W)\right]. \quad (27)$$

This fact, that is illustrated in Fig. 1(a), and that leads to a combination of a predictor and corrector step in one linear system, is exploited in the continuation method by Ohtsuka [45] and in a generalized form in the real-time iteration scheme [16], both described below. When $R(\bar{x}_0, W) = 0$ the formula simplifies to the tangential predictor of the implicit function theorem, which is e.g. employed in the advanced step controller [64].

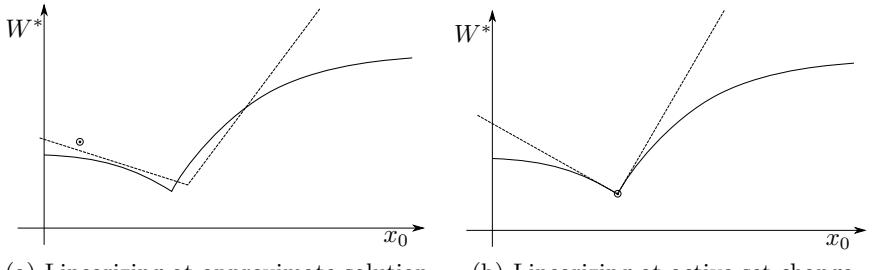
Remark on IP Sensitivities at Active Set Changes: Unfortunately, the interior point solution manifold is strongly nonlinear at points where the active set changes, and the tangential predictor is not a good approximation

when we linearize at such points, as visualized in Fig. 1(b). One remedy would be to increase the path parameter τ, which decreases the nonlinearity, but comes at the expense of generally less accurate IP solutions. This is illustrated in Figs. 2(a) and 2(b) for the same two linearization points as before. In Fig. 2(b) we see that the tangent is approximating the IP solution manifold well in a larger area around the linearization point, but that the IP solution itself is more distant to the true NLP solution.

5.3 Generalized Tangential Predictors via SQP Methods

In fact, the true NLP solution is not determined by a smooth root finding problem (17a)–(17c), but by the KKT conditions (6a)–(6c). It is a well-known fact from parametric optimization, cf. [33], that the solution manifold has smooth parts when the active set does not change (and bifurcations are excluded), but that non-differentiable points occur whenever the active set changes. Is there anything we can do in order to "jump" over these non-smooth points in a way that delivers better predictors than the IP predictors discussed before?

In fact, at points with weakly active constraints, we have to regard *directional* derivatives of the solution manifold, or "generalized tangential predictors". These can be computed by suitable quadratic programs [33, Thm 3.3.4] and are visualized in Fig. 3(b). The theoretical results can be made a practical algorithm by the following procedure proposed in [14]: first, we have to make sure that the parameter $\bar{x}_0$ enters the NLP linearly, which is automatically the case for simultaneous optimal control formulations, cf. Eq. (2b). Second, we address the problem with an exact Hessian SQP method. Third, we just take our current solution guess W for a problem $\bar{x}_0$, and then solve the QP subproblem (21a)–(21f) for the new parameter value $\bar{x}_0'$, but initialized at W. It can be shown [14, Thm. 3.6] that this "initial value embedding" procedure delivers exactly the generalized tangential predictor when started



(a) Linearizing at approximate solution (b) Linearizing at active set change

Fig. 3 Generalized tangential predictors for SQP method

at a solution $W = W^*(\bar{x}_0)$, as in Fig. 3(b). It is important to remark that the predictor becomes approximately tangential when (a) we do not start on the solution manifold, see Fig. 3(a), or (b) we do not use an exact Hessian or Jacobian matrix or (c) we do not evaluate the Lagrange gradient or constraint residuals exactly.

6 Online Algorithms

In NMPC and MHE we would dream to have the solution to a new optimal control problem instantly, which is impossible due to computational delays. Several ideas help to deal with this issue, which we discuss before explaining in detail several of the existing online algorithms. We focus on the NMPC problem but remark that all ideas are also transferable to the MHE problem, which we sometimes mention explicitly.

Offline precomputations: As consecutive NMPC problems are similar, some computations can be done once and for all before the controller starts. In the extreme case, this leads to an explict precomputation of the NMPC control law that has raised much interest in the linear MPC community [1], or a solution of the Hamilton-Jacobi-Bellman Equation, both of which are prohibitive for state and parameter dimensions above ten. But also when online optimization is used, code optimization for the model routines is often essential, and it is in some cases even possible to precompute and factorize Hessians or even Jacobians in Newton type Optimization routines, in particular in the case of neighboring feedback control along reference trajectories [12, 37].

Delay compensation by prediction: When we know how long our computations for solving an NMPC problem will take, it is a good idea *not* to address a problem starting at the current state but to simulate at which state the system will be when we will have solved the problem. This can be done using the NMPC system model and the open-loop control inputs that we will apply in the meantime [24]. This feature is used in many practical NMPC schemes with non-negligible computation time.

Division into preparation and feedback phase: A third ingredient of several NMPC algorithms is to divide the computations in each sampling time into a preparation phase and a feedback phase [16]. The more CPU intensive preparation phase (a) is performed with an old predicted state $\bar{x}_0$ before the new state estimate, say $\bar{x}'_0$, is available, while the feedback phase (b) then delivers quickly an *approximate* solution to the optimization problem for $\bar{x}'_0$. Often, this approximation is based on one of the tangential predictors discussed in the previous section.

Iterating while the problem changes: A fourth important ingredient of some NMPC algorithms is the idea to work on the optimization problem while it changes, i.e., to never iterate the Newton type procedure to convergence

for an NMPC problem getting older and older during the iterations, but to rather work with the most current information in each new iteration. This idea is used in [16, 41, 45].

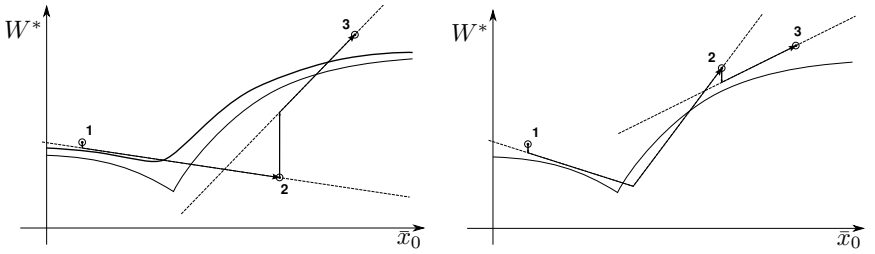
6.1 A Survey of Online Optimization for NMPC

We will in the following review several of the approaches suggested in the literature, in a personal and surely incomplete selection, and try to classify them along the algorithmic lines discussed in this paper.

The Newton-Type Controller of Li and Biegler [40]: This was probably one of the first true online algorithms for NMPC. It is based on a sequential optimal control formulation, thus it iterated in the space of controls $u = (u_0, u_1, \ldots, u_{N-1})$ only. It uses an SQP type procedure with Gauss-Newton Hessian and line search, and in each sampling time, only one SQP iteration is performed. The transition from one problem to the next uses a shift of the controls $u^{\text{new}} = (u_1, \ldots, u_{N-1}, u_N^{\text{new}})$. The result of each SQP iterate is used to give an approximate feedback to the plant. As a sequential scheme without tangential predictor, it seems to have had sometimes problems with nonlinear convergence, though closed-loop stability was proven for open-loop stable processes [41], and in principle, the theoretical NMPC stability analysis from [18] is applicable.

The Continuation/GMRES Method of Ohtsuka [45]: Similar to the Newton-Type controller, the Continuation/GMRES method performs only one Newton type iteration in each sampling time, and is based on a sequential formulation. It is different in that it is based on an IP treatment of the inequalities with fixed path parameter $\tau > 0$, see Section 3.2, that it uses an exact Hessian, and that it uses the iterative GMRES method for linear system solution in each Newton step. Most important, it makes no use of a shift, but instead use of the tangential predictor described in Eq. (27). This features seems to allow it to follow the nonlinear IP solution manifold well – which is strongly curved at active set changes. For a visualization, see Fig. 4(a). In each sampling time, only one linear system is built and solved by the GMRES method, leading to a predictor-corrector pathfollowing method. The closed-loop stability of the method is in principle covered by the stability analysis for the real-time iterations without shift given in [17]. A variant of the method is given in [54], which uses a simultanous approach and condensing and leads to improved accuracy and lower computational cost in each Newton type iteration.

The Real-Time Iteration Scheme [16]: Similar to the Newton-Type controller, the real-time iteration scheme presented in [14, 16] performs one SQP type iteration with Gauss-Newton Hessian per sampling time. However, it employs a simultaneous NLP parameterization, Bock's direct multiple shooting method, with full derivatives and condensing. Moreover, it uses the



(a) Ohtsuka's C/GMRES method (b) Real-Time Iteration scheme

Fig. 4 Subsequent solution approximations

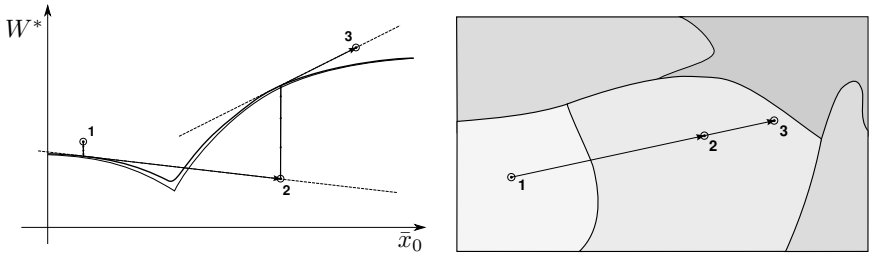
generalized tangential predictor of the "initial value embedding" discussed in Section 5.3 to correct for the mismatch between the expected state $\bar{x}_0$ and the actual state $\bar{x}_0'$. In contrast to the C/GMRES method, where the predictor is based on one linear system solve from Eq. (27), here an inequality constrained QP is solved. The computations in each iteration are divided into a long "preparation phase" (a), in which the system linearization, elimination of algebraic variables and condensing are performed, as described in Sections 4.1–4.3, and a much shorter "feedback phase" (b). The feedback phase solves just one condensed QP (25a)–(25b), more precisely, an "embedded" variant of it, where the expected state $\bar{x}_0$ is replaced by the actual one, $\bar{x}_0'$. Depending on the application, the feedback phase can be several orders of magnitude shorter than the feedback phase. The iterates of the scheme are visualized in Fig. 4(b). The same iterates are obtained with a variant of the scheme that uses Schlöder's trick for reducing the costs of the preparation phase in the case of large state dimensions [51]. Note that only one system linearization and one QP solution are performed in each sampling time, and that the QP corresponds to a linear MPC feedback along a time varying trajectory. In contrast to IP formulations, the real-time iteration scheme gives priority to active set changes and works well when the active set changes faster than the linearized system matrices. In the limiting case of a linear system model it gives the same feedback as linear MPC. Error bounds and closed loop stability of the scheme have been established for shrinking horizon problems in [15] and for NMPC with shifted and non-shifted initializations in [18] and [17].

Advanced Step Controller by Zavala and Biegler [64]: In order to avoid the convergence issues of predictor-corrector pathfollowing methods, in the "advanced step controller" of Zavala and Biegler a more conservative choice is made: in each sampling time, a complete Newton type IP procedure is iterated to convergence (with $\tau \to 0$). In this respect, it is just like offline optimal control – IP, simultaneous, full derivatives with exact Hessian, structure exploiting linear algebra. However, two features qualify it as an online algorithm: first, it takes computational delay into account by solving

an "advanced" problem with the expected state $\bar{x}_0$ as initial value – similar as in the real-time iterations with shift – and (b), it applies the obtained solution not directly, but computes first the tangential predictor which is correcting for the differences between expected state $\bar{x}_0$ and the actual state $\bar{x}_0'$, as described in Eq. (27) with $R(W, \bar{x}_0) = 0$. Note that in contrast to the other online algorithms, several Newton iterations are performed in part (a) of each sampling time, the "preparation phase". The tangential predictor (b) is computed in the "feedback phase" by only one linear system solve based on the last Newton iteration's matrix factorization. As in the C/GMRES method, the IP predictor cannot "jump over" active set changes as easily as the SQP based predictor of the real-time iteration. Roughly speaking, the advanced step controller gives lower priority to sudden active set changes than to system nonlinearity. As the advanced step controller solves each expected problem exactly, classical NMPC stability theory [42] can relatively easily be extended to this scheme [64].

Multi-Level Real-Time Iterations [7]: While the advanced step controller deviates from the other online NMPC schemes in that it performs many Newton iterations per sampling time, the opposite choice is made in the multi-level real-time iterations presented in [7], where even cheaper calculations are performed in each sampling time than one Newton step usually requires. At the lowest level (A), only one condensed QP (25a)–(25b) is solved, for the most current initial value $\bar{x}_0$. This provides a form of linear MPC at the base level, taking at least active set changes into account with a very high sampling frequency. On the next two intermediate levels, that are performed less often than every sampling time, only the nonlinear constraint residuals are evaluated (B), allowing for feasibility improvement, cf. also [12], or the Lagrange gradient is evaluated (C), allowing for optimality improvement, based on the adjoint based SQP presented in Section 3.1.2. Note that in all three levels A, B, and C mentioned so far, no new QP matrices are computed and that even system factorizations can be reused again and again. Level C iterations are still considerably cheaper than one full SQP iteration [61], but also for them optimality and NMPC closed-loop stability can be guaranteed by the results in [17] – as long as the system matrices are accurate enough to guarantee Newton type contraction. Only when this is not the case anymore, an iteration on the highest level, D, has to be performed, which includes a full system linearization and is as costly as a usual Newton type iteration.

Remark on Critical Regions and Online Active Set Strategies: It is interesting to have a look at the parameter space $\bar{x}_0$ visualized in Fig. 5(b). The picture shows the "critical regions" on each of which the active set in the solution is stable. It also shows three consecutive problems on a line that correspond to the scenario used in Figures 4(a), 4(b), and 5(a). Between problem 1 and 2 there is one active set change, while problems 2 and 3 have the same active set, i.e., are in the same critical region. The C/GMRES



(a) Solutions of Advanced Step Controller (b) Critical regions of a parametric NLP

Fig. 5 Subsequent solution approximations (left), and critical regions (right)

method and Advanced Step Controller exploit the smoothness on each critical region in order to obtain the conventional Newton predictor that, however, looses validity when a region boundary is crossed. The real-time iteration basically "linearizes" the critical regions which then become polytopic, by using the more accurate, but also more expensive QP predictor.

As the QP cost can become non-negligible for fast MPC applications, a so-called online active set strategy was proposed in [23]. This strategy goes on a straight line in the space of linearized regions from the old to the new QP problem. As long as one stays within one critical region, the QP solution depends affinely on $\bar{x}_0$ – exactly as the conventional Newton predictor. Only if the homotopy crosses boundaries of critical regions, the active set is updated accordingly. The online active set strategy is available in the open-source QP package qpOASES [22], and is particularly suitable in combination with real-time iterations of level A, B, and C, where the QP matrices do not change, see [62].

Remark on Online MHE Algorithms: Many algorithmic NMPC ideas have been generalized to MHE problems. For example, a Newton-type control framework was used for MHE in [43], the C/GMRES method in [55], cf. also [46], the real-time iteration in [21] and [38], and the advanced step framework in [65]. A somewhat interesting online MHE approach related to the Newton-type control framework was presented in [36], which uses *backwards* single shooting making it not suitable for stiff systems. Other numerical MHE schemes were presented in [35] and [57].

7 Conclusions

In this paper we have tried to give a self-contained overview of Newton type methods for online solution of nonlinear optimal control problems. We first reviewed several categories in which offline algorithms differ, such as simultanous vs. sequential approaches, Interior Point (IP) vs. Sequential Quadratic Programming (SQP) methods, band structure exploiting linear algebra

vs. condensing, and different ways to compute the derivatives needed in Newton type iterations. We then categorized several offline approaches along these lines. The second part started by a discussion of online initializations. We stressed the importance of sensitivity results from parametric optimization, which in SQP type frameworks even allow to obtain cheaply a solution predictor across active set changes. We then classified many proposed real-time optimization approaches from the literature into the developed categories, starting with the "Newton-type controller" [40] and the related "continuation method" [45], both based on sequential approaches, and then went over to the "real-time iteration scheme" [16], a simultaneous approach characterized by an SQP type solution predictor and iterations that perform only one system linearization at each sampling time. We also discussed the recently proposed simultaneous "advanced step controller" [64] and "multi-level real-time iterations" [7], as well as fast online QP solutions [23].

Acknowledgements. M. Diehl is a professor at the Katholieke Universiteit Leuven, H. J. Ferreau and N. Haverbeke are research assistants there. Research supported by Research Council KUL: CoE EF/05/006 Optimization in Engineering(OPTEC), GOA AMBioRICS, IOF-SCORES4CHEM, PhD/postdoc and fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects G.0452.04, G.0499.04, G.0211.05, G.0226.06, G.0321.06, G.0302.07, G.0320.08, G.0558.08, G.0557.08, research communities (ICCoS, ANMMM, MLDM); IWT: PhD Grants, McKnow-E, Eureka-Flite; Helmholtz: viCERP; EU: ERNSI,HD-MPC; Contracts: AMINAL; Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007-2011).

References

[1] Bemporad, A., Borrelli, F., Morari, M.: Model predictive control based on linear programming - the explicit solution. IEEE Transactions on Automatic Control 47(12), 1974–1985 (2002)

[2] Betts, J.T.: Practical Methods for Optimal Control Using Nonlinear Programming. SIAM, Philadelphia (2001)

[3] Biegler, L.T.: Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. Computers and Chemical Engineering 8, 243–248 (1984)

[4] Biegler, L.T., Rawlings, J.B.: Optimization approaches to nonlinear model predictive control. In: Ray, W.H., Arkun, Y. (eds.) Proc. 4th International Conference on Chemical Process Control - CPC IV, pp. 543–571. AIChE, CACHE (1991)

[5] Binder, T., Blank, L., Bock, H.G., Bulirsch, R., Dahmen, W., Diehl, M., Kronseder, T., Marquardt, W., Schlöder, J.P., Stryk, O.V.: Introduction to model based optimization of chemical processes on moving horizons. In: Grötschel, M., Krumke, S.O., Rambau, J. (eds.) Online Optimization of Large Scale Systems: State of the Art, pp. 295–340. Springer, Heidelberg (2001)

[6] Bock, H.G.: Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen. Bonner Mathematische Schriften, vol. 183. Universität Bonn, Bonn (1987)
[7] Bock, H.G., Diehl, M., Kostina, E.A., Schlöder, J.P.: Constrained optimal feedback control of systems governed by large differential algebraic equations. In: Biegler, L., Ghattas, O., Heinkenschloss, M., Keyes, D., van Bloemen Waanders, S.B. (eds.) Real-Time and Online PDE-Constrained Optimization, pp. 3–22. SIAM, Philadelphia (2007)
[8] Bock, H.G., Diehl, M., Leineweber, D.B., Schlöder, J.P.: Efficient direct multiple shooting in nonlinear model predictive control. In: Keil, F., Mackens, W., Voß, H., Werther, J. (eds.) Scientific Computing in Chemical Engineering II, vol. 2, pp. 218–227. Springer, Berlin (1999)
[9] Bock, H.G., Plitt, K.J.: A multiple shooting algorithm for direct solution of optimal control problems. In: Proceedings 9th IFAC World Congress Budapest, pp. 243–247. Pergamon Press, Oxford (1984)
[10] Boyd, S., Vandenberghe, L.: Convex Optimization. University Press, Cambridge (2004)
[11] Bryson, A.E., Ho, Y.-C.: Applied Optimal Control. Wiley, New York (1975)
[12] Büskens, C., Maurer, H.: Sqp-methods for solving optimal control problems with control and state constraints: adjoint variables, sensitivity analysis and real-time control. Journal of Computational and Applied Mathematics 120, 85–108 (2000)
[13] Deuflhard, P.: Newton Methods for Nonlinear Problems. Springer, New York (2004)
[14] Diehl, M.: Real-Time Optimization for Large Scale Nonlinear Processes, VDI Verlag, Düsseldorf. Fortschr.-Ber. VDI Reihe 8, Meß-, Steuerungs- und Regelungstechnik, vol. 920 (2002), http://www.ub.uni-heidelberg.de/archiv/1659/
[15] Diehl, M., Bock, H.G., Schlöder, J.P.: A real-time iteration scheme for nonlinear optimization in optimal feedback control. SIAM Journal on Control and Optimization 43(5), 1714–1736 (2005)
[16] Diehl, M., Bock, H.G., Schlöder, J.P., Findeisen, R., Nagy, Z., Allgöwer, F.: Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. J. Proc. Contr. 12(4), 577–585 (2002)
[17] Diehl, M., Findeisen, R., Allgöwer, F.: A stabilizing real-time implementation of nonlinear model predictive control. In: Biegler, L., Ghattas, O., Heinkenschloss, M., Keyes, D., van Bloemen Waanders, S.B. (eds.) Real-Time and Online PDE-Constrained Optimization, pp. 23–52. SIAM, Philadelphia (2007)
[18] Diehl, M., Findeisen, R., Allgöwer, F., Bock, H.G., Schlöder, J.P.: Nominal stability of the real-time iteration scheme for nonlinear model predictive control. IEE Proc.-Control Theory Appl. 152(3), 296–308 (2005)
[19] Diehl, M., Magni, L., De Nicolao, G.: Online NMPC of unstable periodic systems using approximate infinite horizon closed loop costing. Annual Reviews in Control 28, 37–45 (2004)
[20] Diehl, M., Walther, A., Bock, H.G., Kostina, E.: An adjoint-based SQP algorithm with quasi-newton jacobian updates for inequality constrained optimization. Technical Report Preprint MATH-WR-02-2005, TU Dresden (2005)

[21] Diehl, M., Kühl, P., Bock, H.G., Schlöder, J.P.: Schnelle algorithmen für die zustands- und parameterschätzung auf bewegten horizonten. Automatisierungstechnik 54(12), 602–613 (2006)
[22] Ferreau, H.J.: qpOASES User's Manual (2007–2008), http://homes.esat.kuleuven.be/~optec/software/qpOASES/
[23] Ferreau, H.J., Bock, H.G., Diehl, M.: An online active set strategy to overcome the limitations of explicit MPC. International Journal of Robust and Nonlinear Control 18(8), 816–830 (2008)
[24] Findeisen, R., Allgöwer, F.: Computational delay in nonlinear model predictive control. In: Proc. Int. Symp. Adv. Control of Chemical Processes, ADCHEM (2003)
[25] Fletcher, R.: Practical Methods of Optimization, 2nd edn. Wiley, Chichester (1987)
[26] Franke, R.: Integrierte dynamische Modellierung und Optimierung von Systemen mit saisonaler Wärmespeicherung. PhD thesis, Technische Universität Ilmenau, Germany (1998)
[27] Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: An SQP algorithm for large-scale constrained optimization. Technical report, Numerical Analysis Report 97-2, Department of Mathematics, University of California, San Diego, La Jolla, CA (1997)
[28] Gill, P.E., Murray, W., Wright, M.H.: Practical optimization. Academic Press, London (1999)
[29] Glad, T., Johnson, H.: A method for state and control constrained linear-quadratic control problems. In: Proceedings of the 9th IFAC World Congress, Budapest, Hungary, pp. 1583–1587 (1984)
[30] Griewank, A.: Evaluating Derivatives, Principles and Techniques of Algorithmic Differentiation. Frontiers in Appl. Math, vol. 19. SIAM, Philadelphia (2000)
[31] Griewank, A., Toint, P.L.: Partitioned variable metric updates for large structured optimization problems. Numerische Mathematik 39, 119–137 (1982)
[32] Griewank, A., Walther, A.: On constrained optimization by adjoint based quasi-Newton methods. Optimization Methods and Software 17, 869–889 (2002)
[33] Guddat, J., Guerra Vasquez, F., Jongen, H.T.: Parametric Optimization: Singularities, Pathfollowing and Jumps. Teubner, Stuttgart (1990)
[34] Helbig, A., Abel, O., Marquardt, W.: Model predictive control for on-line optimization of semi-batch reactors. In: Proc. Amer. Contr. Conf, Philadelphia, pp. 1695–1699 (1998)
[35] Jorgensen, J.B., Rawlings, J.B., Jorgensen, S.B.: Numerical methods for large-scale moving horizon estimation and control. In: Proceedings of Int. Symposium on Dynamics and Control Process Systems (DYCOPS) (2004)
[36] Kang, W.: Moving horizon numerical observers of nonlinear control systems. IEEE Transactions on Automatic Control 51(2), 344–350 (2006)
[37] Krämer-Eis, P., Bock, H.G.: Numerical treatment of state and control constraints in the computation of feedback laws for nonlinear control problems. In: Deuflhard, P., et al. (eds.) Large Scale Scientific Computing, pp. 287–306. Birkhäuser, Basel (1987)

[38] Kraus, T., Kühl, P., Wirsching, L., Bock, H.G., Diehl, M.: A moving horizon state estimation algorithm applied to the tennessee eastman benchmark process. In: Proc. of IEEE Robotics and Automation Society conference on Multisensor Fusion and Integration for Intelligent Systems (2006)
[39] Leineweber, D.B., Bauer, I., Schäfer, A.A.S., Bock, H.G., Schlöder, J.P.: An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization (Parts I and II). Computers and Chemical Engineering 27, 157–174 (2003)
[40] Li, W.C., Biegler, L.T.: Multistep, Newton-type control strategies for constrained nonlinear processes. Chem. Eng. Res. Des. 67, 562–577 (1989)
[41] Li, W.C., Biegler, L.T.: Newton-type controllers for constrained nonlinear processes with uncertainty. Industrial and Engineering Chemistry Research 29, 1647–1657 (1990)
[42] Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: stability and optimality. Automatica 26(6), 789–814 (2000)
[43] M'hamdi, A., Helbig, A., Abel, O., Marquardt, W.: Newton-type receding horizon control and state estimation. In: Proc. 13rd IFAC World Congress, San Francisco, pp. 121–126 (1996)
[44] Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, Heidelberg (1999)
[45] Ohtsuka, T.: A continuation/gmres method for fast computation of nonlinear receding horizon control. Automatica 40(4), 563–574 (2004)
[46] Ohtsuka, T., Fujii, H.A.: Nonlinear receding-horizon state estimation by real-time optimization technique. Journal of Guidance, Control, and Dynamics 19(4) (1996)
[47] Powell, M.J.D.: A fast algorithm for nonlinearly constrained optimization calculations. In: Watson, G.A. (ed.) Numerical Analysis, Dundee 1977. LNM, vol. 630, Springer, Berlin (1978)
[48] Rao, C.V., Rawlings, J.B., Mayne, D.Q.: Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. IEEE Transactions on Automatic Control 48(2), 246–258 (2003)
[49] Rao, C.V., Wright, S.J., Rawlings, J.B.: Application of interior-point methods to model predictive control. Journal of Optimization Theory and Applications 99, 723–757 (1998)
[50] Sargent, R.W.H., Sullivan, G.R.: The development of an efficient optimal control package. In: Stoer, J. (ed.) Proceedings of the 8th IFIP Conference on Optimization Techniques, Springer, Heidelberg (1977) (Part 2, 1978)
[51] Schäfer, A., Kühl, P., Diehl, M., Schlöder, J.P., Bock, H.G.: Fast reduced multiple shooting methods for nonlinear model predictive control. Chemical Engineering and Processing 46(11), 1200–1214 (2007)
[52] Schäfer, A.A.S.: Efficient reduced Newton-type methods for solution of large-scale structured optimization problems with application to biological and chemical processes. PhD thesis, Universität Heidelberg (2005)
[53] Schlöder, J.P.: Numerische Methoden zur Behandlung hochdimensionaler Aufgaben der Parameteridentifizierung. Bonner Mathematische Schriften, vol. 187. Universität Bonn, Bonn (1988)
[54] Shimizu, Y., Ohtsuka, T., Diehl, M.: A real-time algorithm for nonlinear receding horizon control using multiple shooting and continuation/krylov method. International Journal of Robust and Nonlinear Control (accepted) (2008)

[55] Soneda, Y., Ohtsuka, T.: Nonlinear moving horizon state estimation for a hovercraft with continuation/gmres method. In: Proc. of IEEE Conf. on Control Applications (2002)
[56] Steinbach, M.C.: A structured interior point sqp method for nonlinear optimal control problems. In: Bulirsch, R., Kraft, D. (eds.) Computation Optimal Control, pp. 213–222. Birkhäuser, Basel (1994)
[57] Tenny, M.J., Rawlings, J.B.: Efficient moving horizon estimation and nonlinear model predictive control. In: Proceedings of the American Control Conference, Anchorage, AK (2002)
[58] Tenny, M.J., Wright, S.J., Rawlings, J.B.: Nonlinear model predictive control via feasibility-perturbed sequential quadratic programming. Computational Optimization and Applications 28(1), 87–121 (2004)
[59] Tsang, T.H., Himmelblau, D.M., Edgar, T.F.: Optimal control via collocation and non-linear programming. International Journal on Control 21, 763–768 (1975)
[60] Wächter, A.: An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering. PhD thesis, Carnegie Mellon University (2002)
[61] Wirsching, L.: An SQP algorithm with inexact derivatives for a direct multiple shooting method for optimal control problems. Master's thesis, University of Heidelberg (2006)
[62] Wirsching, L., Ferreau, H.J., Bock, H.G., Diehl, M.: An online active set strategy for fast adjoint based nonlinear model predictive control. In: Preprints of the 7th Symposium on Nonlinear Control Systems (NOLCOS), Pretoria (2007)
[63] Wright, S.J.: Primal-Dual Interior-Point Methods. SIAM Publications, Philadelphia (1997)
[64] Zavala, V.M., Biegler, L.T.: The advanced step nmpc controller: Optimality, stability and robustness. Automatica (accepted for publication) (2008)
[65] Zavala, V.M., Biegler, L.T.: Nonlinear programming sensitivity for nonlinear state estimation and model predictive control. In: International Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control (2008)

Nonlinear Programming Strategies for State Estimation and Model Predictive Control

Victor M. Zavala and Lorenz T. Biegler

Abstract. Sensitivity-based strategies for on-line moving horizon estimation (MHE) and nonlinear model predictive control (NMPC) are presented both from a stability and computational perspective. These strategies make use of full-space interior-point nonlinear programming (NLP) algorithms and NLP sensitivity concepts. In particular, NLP sensitivity allows us to partition the solution of the optimization problems into background and negligible on-line computations, thus avoiding the problem of computational delay even with large dynamic models. We demonstrate these developments through a distributed polymerization reactor model containing around 10,000 differential and algebraic equations (DAEs).

Keywords: large-scale, MHE, NMPC , nonlinear programming, sensitivity, interior-point methods, sparse linear algebra.

1 Introduction

General model-based control frameworks based on MHE and NMPC represent an attractive alternative for the operation of complex processes. These frameworks allow the incorporation of highly sophisticated dynamic process models and the direct handling of multivariable interactions and operational constraints. In addition, the potential of incorporating detailed first-principles models allows a closer interaction of the controller with traditional economic optimization layers such as real-time optimization (RTO). Crucial enabling developments for this include: a) increased process understanding leading to highly-detailed first-principles dynamic process models, b) enhanced formulations with stability and robustness guarantees, c) advances in numerical

Victor M. Zavala and Lorenz T. Biegler
Department of Chemical Engineering, Carnegie Mellon University, USA
e-mail: vzavala@mcs.anl.gov,lb01@andrew.cmu.edu

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 419–432.
springerlink.com

strategies for DAE-constrained optimization and NLP algorithms, and d) advances in computational resources including the availability of parallel and multi-core technology.

In this work, special emphasis is made on the numerical solution aspects and performance of combined MHE and NMPC strategies. In particular, a general solution framework based on interior-point NLP solvers and sensitivity concepts is considered. In the following section, we introduce some basic concepts and notation and describe specific formulations of the MHE and NMPC nonlinear programming problems. In Section 3 we discuss advantages of interior-point NLP solvers and present some basic NLP sensitivity results. In Section 4 we derive advanced-step approximation strategies for MHE and NMPC, based on NLP sensitivity to reduce on-line computational time. We also discuss their general stability and performance properties, especially when both are applied together. In Section 5, the potential of the combined MHE and NMPC solution framework is demonstrated on a large-scale case study involving the simultaneous monitoring and control of a distributed low-density polyethylene tubular reactor. The paper then closes with general conclusions and recommendations.

2 MHE and NMPC Formulations

We begin with a discrete-time dynamic model of an *uncertain* plant of the form,

$$x_{k+1} = f(x_k, u_k) + \xi_k, \qquad y_{k+1} = \chi(x_{k+1}) + v_{k+1} \tag{1a}$$

where $x_k \in \Re^{n_x}$ is the *true* plant state at time instant t_k and $u_k \in \Re^{n_u}$ is the implemented control action. The nonlinear dynamic model $f(\cdot,\cdot) : \Re^{n_x+n_u} \to \Re^{n_x}$ is the nominal model and satisfies $f(0,0) = 0$. The observed output $y_k \in \Re^{n_y}$ with $n_y \leq n_x$ is related to the state-space x_k through the nonlinear mapping $\chi(\cdot) : \Re^{n_x} \to \Re^{n_y}$. The true plant deviates from the nominal prediction due to the process disturbance $\xi_k \in \Re^{n_x}$ and measurement noise $v_k \in \Re^{n_y}$.

Assume that the plant is currently located at sampling time t_k with the output and input measurements $\eta_k^{mhe} := \{y_{k-N}, ..., y_k, u_{k-N}, ..., u_{k-1}\}$ distributed over a horizon containing N steps. The output measurement covariance is given by $\mathbf{R} \in \Re^{n_y \times n_y}$. The *a priori* estimate of the past state of the plant is denoted as $\bar{x}_{k-N}$ and has an associated covariance $\mathbf{\Pi_{0,k}} \in \Re^{n_x \times n_x}$. Using this information, we would like to compute an estimate $\tilde{x}_k$ of the current state x_k. In order to do this, we solve the MHE problem,

$$\mathcal{M}(\eta_k^{mhe}) \quad \min_{z_0} \ \|z_0 - \bar{x}_{k-N}\|^2_{\mathbf{\Pi}^{-1}_{\mathbf{0,k}}} + \sum_{l=0}^{N} \|y_{k+l-N} - \chi(z_l)\|^2_{\mathbf{R}^{-1}} \tag{2a}$$

$$\text{s.t.} \ \ z_{l+1} = f(z_l, u_{k+l-N}), \ \ l = 0, ..., N-1 \tag{2b}$$

$$z_l \in \mathbb{X} \tag{2c}$$

All the MHE problem data can be summarized in the vector η_k^{mhe}. Symbols $z_l \in \Re^{n_x}$ are *internal* decision variables of the optimization problem. This problem has n_x degrees of freedom corresponding to z_0. From the solution trajectory, $\{z_0^*, ..., z_N^*\}$, we obtain the optimal estimate $\tilde{x}_k = z_N^*$ with associated estimation error $e_k := \tilde{x}_k - x_k$. Using this estimate, we define the problem data $\eta_k^{mpc} := \tilde{x}_k$ for the NMPC problem,

$$\mathcal{P}(\eta_k^{mpc}) \quad \min_{v_l} \quad \Psi(z_N) + \sum_{l=0}^{N-1} \psi(z_l, v_l) \tag{3a}$$

$$\text{s.t. } z_{l+1} = f(z_l, v_l) \quad l = 0, \dots N-1 \tag{3b}$$

$$z_0 = \tilde{x}_k \tag{3c}$$

$$z_l \in \mathbb{X}, v_l \in \mathbb{U} \tag{3d}$$

where $v_l \in \Re^{n_u}$ are internal decision variables. This problem has $(N-1) \times n_u$ degrees of freedom corresponding to $v_l, l = 0, ..., N-1$. Here, we assume that the states and controls are restricted to the domains $\mathbb{X}$ and $\mathbb{U}$, respectively. The stage cost is defined by $\psi(\cdot, \cdot) : \Re^{n_x+n_u} \to \Re$, while the terminal cost is denoted by $\Psi(\cdot) : \Re^{n_x+n_u} \to \Re$. The control action is extracted from the trajectory optimal trajectory $\{z_0^* ... z_N^* \, v_0^*, ..., v_{N-1}^*\}$ as $u_k = v_0^* := h(\tilde{x}_k)$, and $h(\cdot)$ denotes the feedback law. Note that this control action is *inaccurate* because the true state of the plant is x_k and not the estimate $\tilde{x}_k$. That is, the estimation error acts as an additional disturbance. At the next time, the plant will evolve as,

$$x_{k+1} = f(x_k, h(\tilde{x}_k)) + \xi_k, \qquad y_{k+1} = h(x_{k+1}) + v_{k+1} \tag{4}$$

With this, we shift the measurement sequence one step forward to obtain $\eta_{k+1}^{mhe} := \{y_{k-N+1}, ..., y_{k+1}, u_{k-N+1}, ..., u_k\}$, and we solve the new MHE problem. Having the new state estimate $\tilde{x}_{k+1}$ we solve the next NMPC problem.

Note that the above formulations are rather simplified. This makes them convenient for the conceptual analysis in subsequent sections. In practical applications, both NMPC and MHE problems are solved as *general* continuous-time DAE-constrained optimization problems. In this work, we assume that a full discretization approach is used to derive the discrete-time NMPC and MHE formulations. In this case, these NLP problems will be *sparse*. This is a crucial property to be exploited in the following sections.

A problem that is normally encountered in model-based control frameworks is that there exists a computational feedback delay equal to the solution time of the MHE and NMPC problems. In large-scale applications (say $n_x \approx 100 - 10,000$), this computational delay might dominate the time constant of the plant and destabilize the process. Therefore, we seek to derive strategies to reduce the on-line computational time. The first crucial component of these strategies is a fast NLP algorithm. In the next section, we discuss some of the advantages that interior-point NLP solvers offer for the solution of very large problems.

3 Full-Space Interior-Point NLP Solvers

The NLP problems (2) and (3) can be posed in the general form,

$$\mathcal{N}(\eta) \ \min_{\mathbf{x}} F(\mathbf{x},\eta) \tag{5a}$$

$$s.t.\ \mathbf{c}(\mathbf{x},\eta) = 0 \tag{5b}$$

$$\mathbf{x} \geq 0 \tag{5c}$$

where $\mathbf{x} \in \Re^{n_\mathbf{x}}$ is variable vector containing all the states and controls and η is the data vector.

Full-space interior-point solvers have become a popular choice for the solution of large-scale and sparse NLPs. In particular, the solvers LOQO, KNITRO and IPOPT are widely used. In this work, we use IPOPT, an open-source NLP solver originally developed in our research group [1]. In interior-point solvers, the inequality constraints of problem (5) are handled *implicitly* by adding barrier terms to the objective function,

$$\min_{\mathbf{x}} \ F(\mathbf{x},\eta) - \mu_\ell \sum_{j=1}^{n_\mathbf{x}} \ln(\mathbf{x}^{(j)}), \quad \text{s.t.} \quad \mathbf{c}(\mathbf{x},\eta) = 0 \tag{6}$$

where $\mathbf{x}^{(j)}$ denotes the jth component of vector $\mathbf{x}$. Solving (6) for a decaying sequence of $\mu_\ell \to 0, \ell \to \infty$ results in an efficient strategy to solve the original NLP (5). IPOPT solves the Karush-Kuhn-Tucker (KKT) conditions of this sequence of barrier problems (6),

$$\nabla_\mathbf{x} F(\mathbf{x},\eta) + \nabla_\mathbf{x} \mathbf{c}(\mathbf{x},\eta)\lambda - \nu = 0 \tag{7a}$$

$$\mathbf{c}(\mathbf{x},\eta) = 0 \tag{7b}$$

$$\mathbf{X} \cdot \mathbf{V}\, e = \mu_\ell e \tag{7c}$$

where $\mathbf{X} = \text{diag}(\mathbf{x}), \mathbf{V} = \text{diag}(\nu)$ and $e \in \Re^{n_\mathbf{x}}$ is a vector of ones. Symbols $\lambda \in \Re^{n_\lambda}$ and $\nu \in \Re^{n_\mathbf{x}}$ are Lagrange multipliers for the equality constraints and bounds, respectively. To solve this system of nonlinear equations we apply an exact Newton method with the iteration sequence initialized at $s_o^T := [\mathbf{x}_o^T \ \lambda_o^T \ \nu_o^T]$. At the ith iteration, the search direction $\Delta s_i = s_{i+1} - s_i$ is computed by linearization of the KKT conditions (7),

$$\begin{bmatrix} \mathbf{H}_i & \mathbf{A}_i & -\mathbb{I}_{n_\mathbf{x}} \\ \mathbf{A}_i{}^T & 0 & 0 \\ \mathbf{V}_i & 0 & \mathbf{X}_i \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}_i \\ \Delta\lambda_i \\ \Delta\nu_i \end{bmatrix} = -\begin{bmatrix} \nabla_\mathbf{x} F(\mathbf{x}_i) + \mathbf{A}_i\lambda_i - \nu_i \\ \mathbf{c}(\mathbf{x}_i) \\ \mathbf{X}_i\mathbf{V}_i e - \mu_\ell e \end{bmatrix} \tag{8}$$

where $\mathbf{A}_i := \nabla_\mathbf{x} \mathbf{c}(\mathbf{x}_i,\eta)$, $\mathbf{H}_i \in \Re^{n_\mathbf{x} \times n_\mathbf{x}}$ is the Hessian of the Lagrange function $\mathcal{L}F(\mathbf{x}_i,\eta) + \lambda_i^T \mathbf{c}(\mathbf{x}_i,\eta) - \nu_i{}^T \mathbf{x}_i$ and $\mathbb{I}_{n_\mathbf{x}}$ denotes the identity matrix.

We provide exact Hessian and Jacobian information through the modeling platform AMPL. With this, Newton's method guarantees fast local convergence and is able to handle problems with many degrees of freedom

without altering these convergence properties. After solving a sequence of barrier problems for $\mu_\ell \to 0$, the solver returns the optimal solution triplet $s_*^T = [\mathbf{x}_*^T \ \lambda_*^T \ \nu_*^T]$ which implicitly defines the *active-set* (set of variables satisfying $\mathbf{x}^{(j)} = 0$).

3.1 Computational Issues

Solving the KKT system (8) is the most computationally intensive step in the solution of the NLP. A crucial advantage that interior-point solvers offer over active-set solvers is that the structure of the KKT matrix in (8) *does not change* between iterations. This facilitates the design of tailored linear algebra strategies to exploit special structures. For instance, the KKT matrix arising from DAE-constrained optimization problems has a natural forward structure (almost-block-diagonal) in time and classical Riccati-like recursions and condensing techniques are often applied, where the complexity of these solution strategies scales linearly with the horizon length N, but cubically with the number of states n_x and controls n_u. On the other hand, specialized strategies have been developed that reduce the cubic computational complexity and also preserve numerical stability in the face of unstable dynamics [3, 4].

In IPOPT, we use a symmetric indefinite factorization of the KKT matrix (with $\Delta\nu_i$ eliminated). With this, we exploit only the *sparsity pattern* of the KKT matrix. The computational complexity of this strategy is in general very favorable, scaling nearly linearly and at most quadratically with the overall dimensions of the NLP (e.g. length of prediction horizon, number of states and number of degrees of freedom). This general approach also remains stable in the face of unstable dynamics. However, significant fill-in and computer memory bottlenecks might arise during the factorization step if the sparsity pattern is not properly exploited. In order to factorize the KKT matrix, we use the linear solver MA57 from the Harwell library [5]. Since the structure of the KKT matrix does not change between iterations, the linear solver needs to analyze the sparsity pattern *only once*. During this analysis phase, the linear solver permutes the matrix to reduce fill-in and computer memory requirements in the factorization phase. Two reordering strategies are normally used in MA57. The first is an approximate minimum degree (AMD) ordering algorithm while the second is a nested dissection algorithm based on the multi-level graph partitioning strategy, implemented in Metis [6]. For very large-scale problems, these nested dissection techniques excel at identifying high-level (coarse-grained) structures and thus play a crucial role in the factorization time and reliability of the linear solver. These notable advances in numerical linear algebra can dramatically expand the application scope of NMPC and MHE.

IPOPT also applies a regularization scheme to the KKT matrix in order to account for directions of negative curvature and rank-deficient Jacobians which are commonly encountered in highly nonlinear NLPs and/or ill-posed

formulations. Directions of negative curvature are detected implicitly through the linear solver, which returns the so-called inertia of the KKT matrix (number of positive, negative and zero eigenvalues). If the inertia is *correct* at the solution, no regularization is necessary and we can guarantee that the optimal point is a well-defined minimum satisfying strong second order conditions (SSOC) and the linear independence qualification of the constraints (LICQ) [7]. In the context of NMPC and MHE, checking for SSOC is important since this is directly related to properties of the dynamic system such as controllability and observability. Consequently, checking for SSOC through the inertial properties of the KKT matrix is another important advantage of using a general factorization strategy, as opposed to other tailored linear algebra strategies.

3.2 NLP Sensitivity and Warm-Starts

Problem (5) is parametric in the data η and the optimal primal and dual variables can be treated as implicit functions of η. For a *sufficiently small* μ_ℓ, the KKT conditions (7) of the barrier problem (6) can be expressed as $\varphi(s(\eta), \eta) = 0$ and we define $\mathbf{K}_*(\eta_0)$ as the KKT matrix in (8).

We are interested in computing fast approximate solutions for neighboring problems around an already available nominal solution $s_*(\eta_0)$. In order to do this, we make use of the following classical results,

Theorem 1. *(NLP Sensitivity) [7, 8]. If $F(\cdot)$ and $c(\cdot)$ of the parametric problem $\mathcal{N}(\eta)$ are twice continuously differentiable in a neighborhood of the nominal solution $s_*(\eta_0)$ and this solution satisfies LICQ and SSOC, then $s_*(\eta_0)$ is an isolated local minimizer of $\mathcal{N}(\eta_0)$ and the associated Lagrange multipliers are unique. Moreover, for η in a neighborhood of η_0 there exists a unique, continuous and differentiable vector function $s_*(\eta, N)$ which is a local minimizer satisfying SSOC and LICQ for $\mathcal{N}(\eta)$. Finally, there exists a positive Lipschitz constant L such that $\|s_*(\eta, N) - s_*(\eta_0, N)\| \leq L\|\eta - \eta_0\|$ along with a positive Lipschitz constant L_F such that the optimal values $F(\eta)$ and $F(\eta_0)$ satisfy $\|F(\eta) - F(\eta_0)\| \leq L_F\|\eta - \eta_0\|$.*

Under these results, a step $\Delta s(\eta)$ computed from,

$$\begin{aligned} \mathbf{K}_*(\eta_0)\Delta s(\eta) &= -\left(\varphi(s_*(\eta_0), \eta) - \varphi(s_*(\eta_0), \eta_0)\right) \\ &= -\varphi(s_*(\eta_0), \eta). \end{aligned} \tag{9}$$

with $\Delta s(\eta) = \tilde{s}(\eta) - s_*(\eta_0)$, is a Newton step taken from $s_*(\eta_0)$ towards the solution of a neighboring problem $\mathcal{N}(\eta)$. Consequently, $\tilde{s}(\eta)$ satisfies,

$$\|\tilde{s}(\eta) - s_*(\eta)\| \leq L_s\|\eta - \eta_0\|^2 \tag{10}$$

with $L_s > 0$. Furthermore, since the KKT matrix $\mathbf{K}_*(\eta_0)$ is already available from the solution of the nominal problem $\mathcal{N}(\eta_0)$, computing this step requires

only a *single backsolve* which can be performed orders of magnitude faster than the factorization of the KKT matrix.

Since the approximate solution $\tilde{s}(\eta)$ is accurate to first order, we can use it as the initial guess $s_o(\eta)$ to warm-start the NLP $\mathcal{N}(\eta)$. For instance, if the perturbation $(\eta - \eta_0)$ does not induce an active-set change, we can fix μ to a small value (e.g. say 1×10^{-6}) and reuse the KKT matrix $\mathbf{K}_*(\eta_0)$ to perform fast fixed-point iterations on the system,

$$\mathbf{K}_*(\eta_0)\Delta s_i(\eta) = -\varphi(s_i(\eta), \eta) \tag{11}$$

with $s_o = s_*(\eta_0)$. With this, we can reduce the primal and dual infeasibility of the perturbed problem $\mathcal{N}(\eta)$ until no further progress can be made with the fixed KKT matrix. For sufficiently small perturbations, these fast fixed-point iterations can converge to the solution of the perturbed problem $s_*(\eta)$. However, for large perturbations, the KKT matrix needs to be reevaluated and refactorized.

When the perturbation $\eta - \eta_0$ induces an active-set change, the linearization of the complementarity relaxation (7c) contained in the nominal KKT matrix $\mathbf{K}_*(\eta_0)$ will drive the first Newton iteration *outside* of the feasible region and the sensitivity approximation is inconsistent. To compute a fast sensitivity approximation, one could reuse the factorization of the KKT matrix through a Schur complement scheme to correct the active-set (e.g. add slack variables and constraints to drop and fix variables and bound multipliers) [9]. This is equivalent to an active-set sequential quadratic programming (SQP) iteration. Fixed-point iterations can also be performed in this way.

In the context of the proposed MHE and NMPC formulations, we define the optimal solutions,

$$s^*_{MHE} := \{z^*_0, ..., z^*_{N-1}, z^*_N, \lambda^*_1, ..., \lambda^*_{N-1}, \lambda^*_N\} \tag{12a}$$

$$s^*_{MPC} := \{z^*_0, ..., z^*_{N-1}, z^*_N, v^*_0, ..., v^*_{N-2}, v^*_{N-1}, \lambda^*_0, ..., \lambda^*_{N-1}, \lambda^*_N\}. \tag{12b}$$

The associated sensitivity approximations are denoted as $\tilde{s}_{MHE}$ and $\tilde{s}_{MPC}$, respectively, and the corresponding *warm-start* vectors as s^o_{MHE} and s^o_{MPC}. Notice that we have not included the bound multipliers in order to simplify the presentation.

4 Advanced-Step MHE and NMPC Strategies

It is possible to minimize the on-line time required to solve the MHE problem and then the NMPC problem to two fast backsolves using an advanced-step framework [2, 10]. Imagine that at time t_k we know the control action u_k and we would like to obtain an estimate of the future state x_{k+1} but we don't know the future measurement y_{k+1}. Nevertheless, we can use the current estimate $\tilde{x}_k$ and control u_k to predict the future state and associated measurement,

$$\bar{x}_{k+1} = f(\tilde{x}_k, u_k), \qquad \bar{y}_{k+1} = \chi(\bar{x}_{k+1}) \tag{13}$$

to complete the problem data $\bar{\eta}^{mhe}_{k+1} := \{y_{k+1-N}, ..., \bar{y}_{k+1}, u_{k-N}, ..., u_k\}$ and start the solution of the predicted problem $\mathcal{M}(\bar{\eta}^{mhe}_{k+1})$. Simultaneously, we can use the predicted state to define $\bar{\eta}^{mpc}_{k+1} := \bar{x}_{k+1}$ and start the solution of the predicted problem $\mathcal{P}(\bar{\eta}^{mpc}_{k+1})$. Note that both problems are decoupled so this can be done simultaneously and thus reduce the sampling time. At the solution of these problems, we hold the corresponding KKT matrices $\mathbf{K}^{mhe}_*$ and $\mathbf{K}^{mpc}_*$.

Once the true measurement y_{k+1} becomes available, we compute a fast backsolve with $\mathbf{K}^{mhe}_*$ to obtain an *approximate* state estimate $\tilde{x}^{as}_{k+1}$ which differs from the optimal state estimate $\tilde{x}_{k+1}$ and the true state x_{k+1}. Using the approximate state estimate we perform a fast backsolve with $\mathbf{K}^{mpc}_*$ to obtain the approximate control action $u_{k+1} = h^{as}(\tilde{x}^{as}_{k+1})$. Of course, this also differs from the ideal NMPC control $h(\tilde{x}_{k+1})$.

To warm-start the background problems at the next sampling time, we use the approximate solutions $\tilde{s}_{MHE}$ and $\tilde{s}_{MPC}$ to generate the shifted warm-start sequences for the next problems $\mathcal{M}(\bar{\eta}^{mhe}_{k+2})$ and $\mathcal{P}(\bar{\eta}^{mpc}_{k+2})$ [11],

$$s^o_{MHE} := \{\tilde{z}_1, ..., \tilde{z}_N, f(\tilde{x}^{as}_{k+1}, u_{k+1}), \tilde{\lambda}_2, ..., \tilde{\lambda}_N, 0\} \tag{14a}$$

$$s^o_{MPC} := \{\tilde{z}_1, ..., \tilde{z}_N, \tilde{z}_N, \tilde{v}_1, ..., \tilde{v}_{N-1}, \tilde{\lambda}_1, ..., \tilde{\lambda}_N, \tilde{\lambda}_N\}. \tag{14b}$$

from which we update the KKT matrices in between sampling times. Note that the approximate solutions $\tilde{s}_{MHE}$ and $\tilde{s}_{MPC}$ can also be refined in background using fixed-point iterations with $\mathbf{K}^{mhe}_*$ and $\mathbf{K}^{mpc}_*$ *before* using them to generate the warm-start sequences. We summarize the proposed framework for the advanced-step MHE and NMPC strategies, *asMHE* and *asNMPC*, respectively, as follows:

In background, between t_k and t_{k+1}:

1. Use *current* estimate $\tilde{x}^{as}_k$ and control u_k to predict the future state $\bar{x}_{k+1} = f(\tilde{x}^{as}_k, u_k)$ and corresponding output measurement $\bar{y}_{k+1} = \chi(\bar{x}_{k+1})$.
2. Define the data $\bar{\eta}^{mhe}_{k+1} = \{y_{k+1-N}...y_k, \bar{y}_{k+1}, u_{k+1-N}, ..., u_k\}$ and $\bar{\eta}^{mpc}_{k+1} = \bar{x}_{k+1}$. Use the available warm-start points s^o_{MHE} and s^o_{MPC} to solve the predicted problems $\mathcal{M}_N(\bar{\eta}^{mhe}_{k+1})$ and $\mathcal{P}_N(\bar{\eta}^{mhe}_{k+1})$.
3. Hold the KKT matrices $\mathbf{K}^{mhe}_*$ and $\mathbf{K}^{mpc}_*$.

On-line, at t_{k+1}:

1. Obtain the true measurement y_{k+1} and define the *true* MHE data η^{mhe}_{k+1}. Reuse factorization of $\mathbf{K}^{mhe}_*$ to quickly compute $\tilde{s}_{MHE}$ from (9) and extract $\tilde{x}^{as}_{k+1}$.
2. Use $\tilde{x}^{as}_{k+1}$ to define the *true* NMPC problem data η^{mpc}_{k+1}. Reuse factorization of $\mathbf{K}^{mpc}_*$ to quickly compute $\tilde{s}_{MPC}$ from (9) and extract $u_{k+1} = h^{as}(\tilde{x}^{as}_{k+1})$.
3. If necessary, refine $\tilde{s}_{MHE}$ and $\tilde{s}_{MPC}$. Generate the warm-starts s^o_{MHE} and s^o_{MPC}, set $k := k + 1$, and return to background.

4.1 Stability Issues

It is clear that both the state estimate and the associated control action are *suboptimal* due to the presence of NLP approximation errors. Here, we are interested in assessing the impact of these errors in the stability of the closed-loop system. From the controller point of view, we are interested in finding sufficient conditions under which the closed-loop remains stable in the face of disturbances and NLP sensitivity errors. Due to space limitations we outline the main results here and refer the interested reader to [2] for more details.

To start the discussion, we first note that solving the predicted problem $\mathcal{P}(\bar{x}_{k+1})$ in the *asNMPC* controller is equivalent to solving the *extended* problem,

$$\mathcal{P}_{N+1}(\eta_k^{mpc}) \quad \min_{v_l} \quad \Psi(z_N) + \psi(x_k, u_k) + \sum_{l=0}^{N-1} \psi(z_l, v_l) \tag{15a}$$

$$\text{s.t. } z_{l+1} = f(z_l, v_l) \quad l = 0, \dots N-1 \tag{15b}$$

$$z_0 = f(x_k, u_k) \tag{15c}$$

$$z_l \in \mathbb{X}, v_l \in \mathbb{U} \tag{15d}$$

with fixed x_k, $u_k = h(x_k)$ and $\eta_k^{mpc} = \{x_k, h(x_k)\}$. For the optimal or ideal NMPC controller (instantaneous optimal solutions), we consider the neighboring costs of the extended problems with *perfect* state information $J_{x_k}^{h(x_k)} := J_{N+1}(x_k, h(x_k))$ and $J_{x_{k+1}}^{h(x_{k+1})} := J_{N+1}(x_{k+1}, h(x_{k+1}))$ as reference points. As observed by Muske and Rawlings [12], since the *implemented* control action is based on the state *estimate* $\tilde{x}_k$ coming from MHE and not on the true state x_k, we consider this as an additional disturbance to the closed-loop system through the cost $J_{\hat{x}_{k+1}}^{h(\hat{x}_{k+1})}$ where $\hat{x}_{k+1} = f(x_k, h(\tilde{x}_k)) + \xi_k$. From Lipschitz continuity of the cost function we have,

$$|J_{\hat{x}_{k+1}}^{h(\hat{x}_{k+1})} - J_{x_{k+1}}^{h(x_{k+1})}| \leq L_J L_f L_h \|x_k - \tilde{x}_k\|.$$

Explicit bounds and convergence properties on the estimator error $\|x_k - \tilde{x}_k\|$ can be established for the MHE formulation (2) [15]. Moreover, we can also treat this error as another disturbance ξ_k and define $\tilde{x}_k := x_k + \xi_k$. This allows us to restate the following robustness result for the combined *asMHE* and *asNMPC* strategies.

Theorem 2 (*Theorem 6 in [2]*)**.** *Assume that the NLPs for (2) and (3) can be solved within one sampling time. Assume also that nominal and robust stability assumptions for ideal NMPC hold (see [2]), then there exist bounds on the noise ξ and v for which the cost function $J_{N+1}(x)$, obtained from the combined asMHE-asNMPC strategy, is an input to state stable (ISS) Lyapunov function, and the resulting closed-loop system is ISS stable.*

5 Case Study

We demonstrate the performance of the proposed advanced-step framework on a low-density polyethylene (LDPE) tubular reactor process. A schematic representation of a typical multi-zone LDPE reactor is presented in Figure 1. In these reactors, high-pressure (2000-3000 atm) ethylene polymerizes through a free-radical mechanism in the presence of peroxide initiators, which are fed at multiple zones in order to start and stop the polymerization. The large amounts of heat produced by polymerization are removed at each zone using cooling water, along with multiple feeds of ethylene that cool the ethylene-polymer reacting mixture flowing inside the reactor core. Initiator flow rates, ethylene side-streams flow rates and temperatures, and the cooling water inlet temperatures and flow rates can be manipulated to achieve an axial reactor temperature profile that produces a desired polymer grade. A common problem in these reactors is that polymer accumulates (i.e., fouls) on the reactor walls. The resulting fouling layer blocks heat flow to the jacket cooling water and can be seen as a persistent dynamic disturbance. In the absence of a suitable control system, this fouling layer will eventually lead to thermal runaway. A centralized model-based control strategy based on a first-principles reactor model can deal effectively with fouling monitoring, zone control decoupling and direct optimization of the overall process economics (e.g. maximize production, minimize energy consumption). Nevertheless, LDPE reactor models consist of very large sets of PDAEs that describe the evolution of the reactor mixture and of the cooling water temperature along the axial and time dimension. After axial discretization, a typical LDPE reactor model can easily contain more than 10,000 DAEs.

An MHE estimator and an NMPC controller based on first-principles LDPE reactor models have been reported in [13, 14]. While these reports stress the benefits of these strategies for the LDPE process, little emphasis has been placed on the computational limitations associated to their on-line solution. Here, we consider these issues through the proposed advanced-step control framework where we effectively minimize the on-line computation with negligible approximation errors. We simulate the scenario in which the reactor is fouled and cleaned over time, by ramping the reactor heat-transfer coefficients (HTCs) down and up. Because this effect is directly reflected through HTCs in the LDPE reactor model, we do not estimate the process disturbance ξ_k, and instead use the MHE estimator to estimate the HTCs

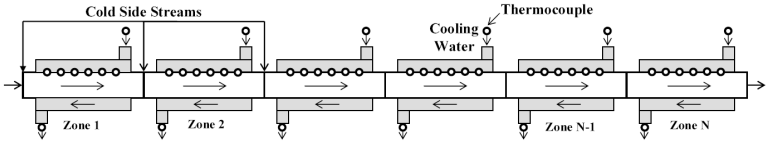


Fig. 1 Schematic representation of multi-zone LDPE tubular reactor

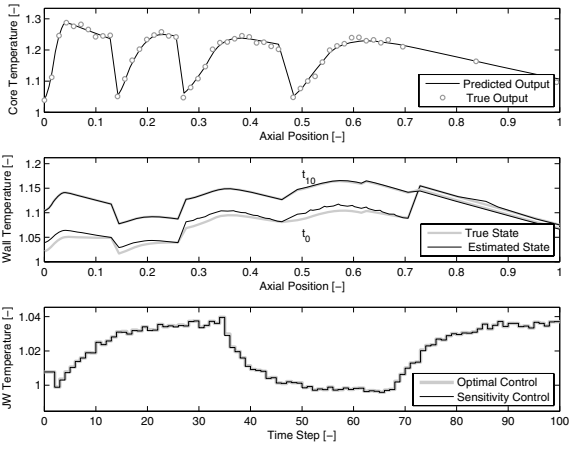


Fig. 2 Performance of advanced-step MHE and NMPC in LDPE case study

and the unmeasured model states (e.g. wall temperature profile) at each time step. For the MHE estimator, y_k consists of multiple measurements of the reactor core temperature and the output jacket temperatures in each zone. The objective of the NMPC controller is to use the estimated reactor state $\tilde{x}_k^{as}$ to drive the axial reactor temperature profile to the specified target profile. In order to do this, the NMPC controller uses the multiple inputs distributed along the reactor to obtain $u_k = h^{as}(\tilde{x}_k^{as})$. In this simulated scenario, we generate the plant response x_k from the model with the *true* HTCs. In addition, the plant is initialized at a different state from that of the NMPC controller. Finally, we corrupt the output measurements with Gaussian noise.

Since the plant response differs from that of the NMPC controller prediction and we introduce noise, the *asMHE* estimator will see a difference between the measured and the predicted outputs (see top graph of Figure 2) and will correct on-line using NLP sensitivity. We have found that the approximation errors are negligible and the *asMHE* estimator has almost identical convergence properties to that of the ideal MHE estimator. In the middle graph of Figure 2, we see that while the estimate of the reactor wall profile is inaccurate at t_0, the dashed and solid lines coincide by t_{10}, and the *asMHE* estimator converges to the *true* reactor wall profile (and the one obtained from ideal MHE) using reactor core measurements in about 10 time steps. Using the estimated states and HTCs, the *asNMPC* controller then updates the predicted state on-line. In the bottom graph of Figure 2 we present the closed-loop response of one of the jacket water inlet temperatures for the *asNMPC* controller and its ideal NMPC counterpart. As can be seen, both control actions are identical. In this graph we can also appreciate how the HTC cycles influence the controller response.

In the top graph of Figure 3 we present the total wall-clock time required to refine the perturbed solution, generate the warm-start point and solve the

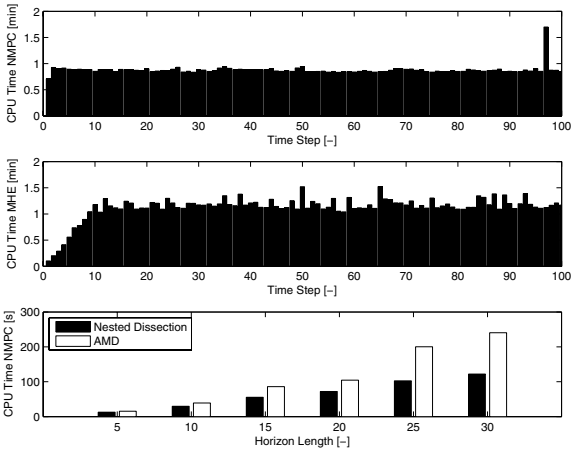


Fig. 3 Computational results. Background tasks NMPC (top). Background tasks MHE (middle). Scale-up of NMPC problem (bottom)

background NMPC problem. This time also includes some overhead coming from I/O communication tasks and from AMPL, which requires some time to generate the derivative information before calling the NLP solver. A prediction horizon of $N = 10$ time steps (20 minutes) and sampling times of 2 minutes have been used. The NMPC problem consists of an NLP with 80,950 constraints and 370 degrees of freedom. As can be seen, the overall background time is around 60 seconds and is well below the specified sampling time. A single factorization of the KKT matrix takes 15.34 seconds, a single fixed-point iteration requires 0.1 seconds, and an average of 5 fixed point iterations are required to solve the NLP. In the middle graph of Figure 3, we present total background times for the MHE estimator. The estimator is initialized in batch mode (accumulate measurements until an estimator horizon of N time steps is filled). Once the estimation horizon is complete, the background tasks take around 70 seconds to be completed. The MHE problem consists of an NLP with 80,300 constraints and 648 degrees of freedom. One fixed-point iteration requires 0.12 seconds and an average of 10 fixed point iterations solve the NLP. In the bottom graph of Figure 3, we present scale-up results of the solution time for the NMPC problem with increasing horizon length. We compare the impact of AMD and nested dissection sparse matrix reordering on the solution time of the background NLP problem (without refinement or overhead). The multi-level nested dissection strategy is more efficient here and achieves a linear scale-up. Using this strategy, a $N = 30$ NMPC problem with 242,850 constraints and 1,110 degrees of freedom is solved in around 2 minutes, the factorization of the KKT matrix takes 32.31 seconds and a fixed-point iteration requires 0.33 seconds. The AMD strategy shows quadratic scale-up and the largest problem requires 4 minutes. This difference can be attributed to the fact that the Metis nested dissection

algorithm is much more efficient in identifying coarse-grained structures in the NMPC problem (LDPE multi-zone model, DAE forward structure, etc.), while AMD tends to focus on fine-grained structures. All calculations were obtained using a quad-core Intel processor running Linux at 2.4 GHz.

6 Conclusions

In this work, we present computational strategies for MHE and NMPC problems. In particular, a general solution framework based on interior-point NLP solvers and sensitivity concepts is considered. We emphasize that exploiting the overall sparsity pattern of the KKT matrix arising in NMPC and MHE problems leads to a computationally efficient and stable strategy to compute the Newton step. We analyze the impact of different reordering techniques of the KKT matrix on the factorization time and computer memory limitations. In particular, we present NLP sensitivity-based strategies for MHE and NMPC that reduce the on-line computation time to only two fast backsolves. This negligible computation effectively removes the problem of computational delay even for very large NLP models. Finally, we discuss stability issues of the NMPC controller in the face of sensitivity errors and demonstrate the developments in a distributed polymerization reactor process, where highly accurate solutions can be obtained in a negligible amount of time.

References

1. Wächter, A., Biegler, L.T.: On The Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming. Math. Programm. 106, 25–57 (2006)
2. Zavala, V.M., Biegler, L.T.: The Advanced Step NMPC Controller: Stability, Optimality and Robustness. Automatica 45, 86–93 (2009)
3. Zavala, V.M., Laird, C.D., Biegler, L.T.: Fast Implementations and Rigorous Models: Can Both be Accommodated in NMPC? Int. Journal of Robust and Nonlinear Control 18, 800–815 (2008)
4. Schäfer, A., Kühl, P., Diehl, M., Schlöder, J., Bock, H.G.: Fast reduced multiple shooting methods for nonlinear model predictive control. Chemical Engineering and Processing 46, 1200–1214 (2007)
5. Duff, I.S.: MA57 - A Code for the Solution of Sparse Symmetric Definite and Indefinite Systems. ACM Transactions on Mathematical Software 30, 118–144 (2004)
6. Karypis, G., Kumar, V.: A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. SIAM J. Sci. Comput. 20, 359–392 (1999)
7. Fiacco, A.V.: Introduction to Sensitivity and Stability Analysis in Nonlinear Programming. Academic Press, New York (1983)
8. Büskens, C., Maurer, H.: Sensitivity Analysis and Real-Time Control of Parametric Control Problems Using Nonlinear Programming Methods. In: Grötschel, M., et al. (eds.) On-line Optimization of Large-scale Systems. Springer, Berlin (2001)

9. Bartlett, R.A., Biegler, L.T., Backstrom, J., Gopal, V.: Quadratic Programming Algorithms for Large-Scale Model Predictive Control. Journal of Process Control 12, 775–795 (2002)
10. Zavala, V.M., Laird, C.D., Biegler, L.T.: A Fast Moving Horizon Estimation Algorithm Based on Nonlinear Programming Sensitivity. Journal of Process Control 18, 876–884 (2008)
11. Findeisen, R., Diehl, M., Burner, T., Allgöwer, F., Bock, H.G., Schlöder, J.P.: Efficient Output Feedback Nonlinear Model Predictive Control. In: Proceedings of American Control Conference, vol. 6, pp. 4752–4757 (2002)
12. Muske, K.R., Meadows, E.S., Rawlings, J.B.: The Stability of Constrained Receding Horizon Control with State Estimation. In: Proceedings of American Control Conference, vol. 3, pp. 2837–2841 (1994)
13. Zavala, V.M., Biegler, L.T.: Large-Scale Nonlinear Programming Strategies for the Operation of Low-Density Polyethylene Tubular Reactors. In: Proceedings of ESCAPE 18, Lyon (2008)
14. Zavala, V.M., Biegler, L.T.: Optimization-Based Strategies for the Operation of Low-Density Polyethylene Tubular Reactors: Moving Horizon Estimation. Comp. and Chem. Eng. 33, 379–390 (2009)
15. Alessandri, A., Baglietto, M., Battistelli, G.: Moving-Horizon State Estimation for Nonlinear Discrete-Time Systems: New Stability Results and Approximation Schemes. Automatica 44, 1753–1765 (2008)

A Framework for Monitoring Control Updating Period in Real-Time NMPC Schemes

Mazen Alamir

Abstract. In this contribution, a general scheme for on-line monitoring of the control updating period in real-time Nonlinear Model Predictive Control (NMPC) schemes is proposed. Such a scheme can be of a great interest when applying NMPC to systems with fast dynamics. The updating scheme is based on the on-line identification of generic models for both solver efficiency and disturbance effects on the optimal cost behavior. A simple example is used to illustrate the efficiency of the proposed methodology.

Keywords: Nonlinear Model Predictive Control; Real-Time Implementation; Fast Systems; Updating Period Monitoring.

1 Introduction

The classical stability theory of NMPC schemes relies on the availability of an optimal solution in within a fraction of the basic sampling period [7]. When applying NMPC to fast systems, this assumption can be questionable and many realistic approaches have been proposed during the last decades to come closer to concrete implementation schemes. The common feature in these new approaches is to *distribute* the iterations of some optimization scheme over the real lifetime of the system. Various optimization schemes have been investigated such as multiple shooting [5, 6], continuation [8] or gradient-based descent scheme [4]. Although the very idea of *distributing* the iteration was early *in the air* in an intuitive way [2], the recent works mentioned above address its implications on the stability and the robustness of the resulting closed-loop for each particular scheme. All these schemes however may not

Mazen Alamir
CNRS-Gipsalab, Control Systems Department. University of Grenoble
BP 46, Domaine Universitaire, Saint Martin d'Hères, France
e-mail: mazen.alamir@gipsa-lab.inpg.fr

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 433–445.
springerlink.com

be suitable when an implicit parametrization of the control profile is used [1] since theoretical and/or computational differentiability issues may arise. Moreover, even in the very precise frameworks of these approaches, there is no evidence that the optimization time and the real system's lifetime have to be taken strictly equal. Some trade-off may be needed especially in the presence of model mismatch and/or unpredictable external disturbance.

The aim of the present paper is to provide a general framework addressing the following issue: Given an optimization subroutine $\mathcal{S}$, a control profile bandwidth leading to a basic sampling period $\tau > 0$ and a computation time τ_c needed to perform one basic iteration, how to monitor the control updating period to be used? Namely, how many iterations one has to perform in order between two successive update of the future control sequence. The existence of an optimal trade-off is clearly shown and a general updating framework is suggested that can be used as a monitoring layer in any existing real-time NMPC scheme.

The paper is organized as follows: First, the problem is stated in section 2. The theoretical framework is described in section 3. Some computational issues related to on-line identification of unknown maps are discussed in section 4. Finally, an illustrative example is proposed in section 5.

2 Problem Statement

2.1 Recall on Parameterized NMPC

Let us consider nonlinear systems that admit the following implicit model:

$$x(t) = X(t, x_0, \mathbf{u}) \quad ; \quad t \leq T \tag{1}$$

where $x(\cdot) \in \mathbb{R}^n$ is the state trajectory that starts at the initial value $x(0) = x_0$ under the control profile $\mathbf{u} \in \mathbb{U}^{[0,T]}$ for some subset $\mathbb{U} \subset \mathbb{R}^m$ of admissible control inputs. The implicit map X is obtained by any suitable integrating a suitable process model. T is some prediction horizon over which the model (1) is meaningful.

In what follows, the evolution of the real system is clearly distinguished from that of the model (1) by adopting the following notation:

$$x(t) = X^r(t, x_0, \mathbf{u}, \mathbf{w}) \quad ; \quad t \leq T \tag{2}$$

where $\mathbf{w}$ represents the unknown uncertainty/disturbance profile that may affect the system during the time interval.

When the model (1) is used in a parameterized NMPC scheme [1], it is quite common to use some fixed sampling period $\tau > 0$ that reflects the characteristic time of the system together with some piecewise-constant open-loop control parametrization:

$$\mathcal{U}_{pwc}(p) := \big(u^{(1)}(p), \ldots, u^{(N)}(p)\big) \in \mathbb{U}^N \quad ; \quad N\tau = T \tag{3}$$

where $u^{(k)}(p)$ defines the open-loop control value to be applied during the future interval $[(k-1)\tau, k\tau]$. As soon as such a parametrization is fixed, the implicit model can be written in terms of the parameter vector p with an obvious overloaded notations:

$$x(t) = X(t, x_0, p) \quad ; \quad t \leq T \tag{4}$$

By doing so, NMPC related cost functions can be viewed as functions of the initial state and the parameter vector p, namely $J(p, x_0)$ that is assumed to be positive. This enables to define a state dependent optimal parameter $\hat{p}(x)$ by:

$$\hat{p}(x) := \arg \min_{p \in \mathbb{P}(x) \subset \mathbb{P}} J(p, x) \tag{5}$$

where $\mathbb{P}(x) \subset \mathbb{P} \subset \mathbb{R}^{n_p}$ is some set of admissible parameter values that may depend on the state. In what follows, the above optimization problem is denoted by $\mathcal{P}(x)$.

According to (3), the optimal value $\hat{p}(x)$ defines a sequence of control inputs by:

$$\mathcal{U}_{pwc}(\hat{p}(x)) := \left(u^{(1)}(\hat{p}(x)), \ldots, u^{(N)}(\hat{p}(x))\right) \in \mathbb{U}^N \quad ; \quad N\tau = T \tag{6}$$

Classically, NMPC schemes use τ as updating period with the following time sampled state feedback law:

$$K = u^{(1)} \circ \hat{p} \tag{7}$$

that is, only the first input in the optimal sequence $\mathcal{U}_{pwc}(\hat{p}(x(t_k))$ is applied during the time interval $[t_k, t_k + \tau = t_{k+1}]$ before a new optimal sequence is computed by solving the optimization problem associated to the next state $x(t_{k+1})$ and so on.

Such scheme assumes that the optimization problem can be solved, or at least a sufficient number of iterations towards its solution can be done in less than τ time units. For fast systems where τ needs to be very small, this condition may become hard to satisfy and some dedicated scheme has to be adopted as shown in the following section. Note that throughout the paper, the *number of iterations* refers to the *number of functions evaluations* as it is the key operation in NMPC schemes.

2.2 Implementation Scheme for Fast Systems

As the basic sampling period τ may be too small for control updating, the control updating period can be taken equal to some multiple $\tau_u = N_u\tau$ of the basic sampling period τ. So let us denote by $t_i^u = i\tau_u$ the so-called updating instants (see later for a precise definition of the updating process). At instant $t = 0$, some initial parameter vector $p(t_0^u = 0) \in \mathbb{P}(x_0)$ is chosen (arbitrarily

or according to some ad-hoc initialization rule). The corresponding sequence of control inputs

$$\mathcal{U}_{pwc}(p(t_0^u)) = \big(u^{(1)}(p(t_0^u)) \ \dots \ u^{(N_u)}(p(t_0^u)) \ \dots \ u^{(N)}(p(t_0^u))\big)$$

is computed. During the time interval $[t_0^u, t_1^u]$, the first N_u control inputs

$$\big(u^{(1)}(p(t_0^u)) \ \dots \ u^{(N_u)}(p(t_0^u))\big)$$

are applied. In parallel, the computation unit performs successively the following two tasks:

1. Compute the model based prediction of the state at the future instant t_1^u according to:
$$\hat{x}(t_1^u) = X(\tau_u, x(t_0^u), p(t_0^u))$$
This prediction involves only the first N_u elements of $\mathcal{U}_{pwc}(p(t_0^u))$.
2. Try to solve the optimization problem $\mathcal{P}(\hat{x}(t_1^u))$ by performing q steps of some optimization process $\mathcal{S}$ with an initial guess $p^+(t_0^u)$ such that:
$$J\big(p^+(t_0^u), \hat{x}(t_1^u)\big) \leq J\big(p(t_0^u), x(t_0^u)\big) \tag{8}$$
The later may be obtained from $p(t_0^u)$ by some appropriate transformation. For instance, the translatability property invoked in [1] can be used. However, the precise choice of the transformation $p^+(t_1^u)$ is meaningless for the remainder of the sequel. This is shortly denoted as follows:
$$p(t_1^u) = \mathcal{S}^q\big(p^+(t_0^u), \hat{x}(t_1^u)\big) \tag{9}$$

Note that $p(t_1^u)$ is generally different from the optimal value $\hat{p}(\hat{x}(t_1^u))$ that may need much more iterations to be obtained. During the time interval $[t_1^u, t_2^u]$, the N_u control inputs

$$\big(u^{(1)}(p(t_1^u)) \ \dots \ u^{(N_u)}(p(t_1^u))\big)$$

are applied and so on.

To summarize

Under the above implementation rules, the state of the real system at the updating instants $\{t_i^u\}_{i\geq 0}$ is governed by the following coupled dynamic equations:

$$x(t_i^u) = X^r\big(\tau_u, x(t_{i-1}^u), p(t_{i-1}^u), \mathbf{w}\big) \tag{10}$$

$$p(t_i^u) = \mathcal{S}^q\Big(p^+(t_{i-1}^u), \underbrace{X(\tau_u, x(t_{i-1}^u), p(t_{i-1}^u))}_{\hat{x}(t_i^u)}\Big) \tag{11}$$

that clearly involve an extended dynamic state $z = \big(p^T \ x^T\big)^T \in \mathbb{R}^{n+n_p}$ and that heavily depend on the design parameters τ_u, $\mathcal{S}$ and q.

2.3 The Scope of the Present Contribution

In the present contribution, the interest is focused on the following issue:

> *Given some optimization process $\mathcal{S}$, propose a concrete updating rules for both the control updating period τ_u and the number of iterations q in order to* improve *the closed-loop behavior under the real-time NMPC implementation framework proposed in section 2.2.*

In particular, it is worth emphasizing here that the choice of the optimization process $\mathcal{S}$ that may be used in order to come closer to the optimal value $\hat{p}(x(t_i^u))$ is not the issue of this paper. The updating rule proposed hereafter can be used as an additional *layer* to any specific choice of the optimization process $\mathcal{S}$.

3 Theoretical Framework

Although we seek a general framework, we still need some assumptions on the optimization process $\mathcal{S}$ being used. In particular, it is assumed that the process is *passive* in the following trivial sense

Assumption 1 [The optimizer is passive]
For all $q \in \mathbb{N}$ and all $z := (p, x) \in \mathbb{R}^{n_p} \times \mathbb{R}^n$, the following inequality holds:

$$J(\mathcal{S}^q(p, x), x) \leq J(p, x) \tag{12}$$

which simply means that at worst, the optimize returns the initial guess p.

A key property in the success of the real-time NMPC scheme is the efficiency of the optimizer, namely, its ability to lead to a best value $\mathcal{S}^q(p, x)$ of the parameter vector starting from the initial guess p. This efficiency may heavily depend on the pair (p, x) and can be quantified through the following definition:

Definition 1 [Efficiency of the optimizer]
For all $(p, x) \in \mathbb{R}^{n_p} \times \mathbb{R}^n$, the map defined by:

$$E^f_{(p,x)}(q) := \frac{J\big(\mathcal{S}^q(p, x), x\big)}{J(p, x)} \tag{13}$$

is called the efficiency map at (p, x).

Note that by virtue of assumption 1, the efficiency map satisfies the inequality $E^f_{(p,x)}(q) \leq 1$ for all q and all (p, x).

The last element that plays a crucial role in determining what would be an optimal choice is the level of model discrepancy. More precisely, how this discrepancy degrades the value of the resulting cost function at updating instants. This may be handled by the following definition:

Definition 2 [Model Mismatch Indicator]
For all pair (p, x), the map defined by:

$$D_{(p,x)}(\tau_u) := \sup_{(\bar{p},\mathbf{w})\in\mathbb{P}\times\mathbb{W}} \Big[\frac{J(\bar{p}, X^r(\tau_u, x, p, \mathbf{w}))}{J(\bar{p}, X(\tau_u, x, p))}\Big] \tag{14}$$

is called the model mismatch indicator at (p, x).

Note that by definition, $D_{(p,x)}(0) = 1$ for all (p, x) since one clearly has $X(0, x, p) = X^r(0, x, p, \mathbf{w}) = x$. As a matter of fact, $D_{(p,x)}(\tau_u)$ represents a worst case degradation of the cost function that is due to the bad prediction $\hat{x}(t^u_{i+1})$ of the future state $x(t^u_{i+1})$ at the next decision instant.

Based on the above definitions, the following result is straightforward:

Lemma 1 [A Small Gain Result]
Under the real-time implementation scheme given by (10)-(11), the dynamic of the cost function satisfies the following inequality:

$$J(p(t^u_{i+1}), x(t^u_{i+1})) \le \big[E^f_{(i)}(q)\big]\big[D_{(i)}(\tau_u)\big] \cdot J(x(t^u_i), p(t^u_i)) \tag{15}$$

where the following short notations are used:

$$E^f_{(i)}(q) := E^f_{(p(t^u_i),x(t^u_i))}(q) \quad ; \quad D_{(i)}(\tau_u) := D_{(p(t^u_i),x(t^u_i))}(\tau_u)$$

Consequently, if the following small gain condition is satisfied for all i:

$$K_{(i)}(q, \tau_u) := \big[E^f_{(i)}(q)\big]\big[D_{(i)}(\tau_u)\big] \le \gamma < 1 \tag{16}$$

then the closed loop evolution of (p, x) is such that $J(\cdot, x(t^u_i))$ asymptotically converges to 0. ♡

Recall that our aim is to define a rationale in order to choose the parameters of the real-time scheme, namely q and τ_u. The *small gain* condition (16) of lemma 1 provides a constraint guiding this choice (provided that estimations of the maps $E^f_{(i)}$ and $D_{(i)}$ are available). Another trivial constraint expresses the fact that the time needed to perform q iterations is lower than the control updating period τ_u, this constraint can be written as follows:

$$q = \Big[\frac{\tau_u}{\tau_c}\Big] \tag{17}$$

where τ_c is the time needed to perform a single iteration (function evaluation) while for all $\rho \in \mathbb{R}$, $[\rho]$ stands for the greatest integer lower than ρ.

Note that inequalities (16)-(17) are constraints that take account for convergence and feasibility issues respectively. The quality of the closed loop may be expressed in term of the settling time, that is the predicted time necessary to contract the cost function by a given ratio. Using (17), this can be expressed by the following cost function:

$$t_r(\tau_u) := \frac{\tau_u}{|\log(K_{(i)}(\lceil \frac{\tau_u}{\tau_c} \rceil, \tau_u))|} \tag{18}$$

Therefore, provided that an approximation of the map $K_{(i)}(q, \tau)$ is obtained (via on-line identification), the following constrained optimization problem may be used to compute at each updating instant t_i^u the value of the next *optimal* control updating period $\tau_u(t_i^u)$ and the number of iterations $q(t_i^u)$ [by virtue of (17)]:

$$\tau_u(t_i^u) := \begin{cases} arg \min\limits_{\tau_u \in \mathbb{N}} \left[t_r(\tau_u) \right] & \text{under } K_{(i)}(\lceil \frac{\tau_u}{\tau_c} \rceil, \tau_u) < 1 \text{ when feasible} \\ arg \min\limits_{\tau_u \in \mathbb{N}} \left[K_{(i)}(\lceil \frac{\tau_u}{\tau_c} \rceil, \tau_u) \right] & \text{otherwise} \end{cases} \tag{19}$$

In the remainder of this paper, a scheme is proposed to identify the key function map $K_{(i)}(\cdot, \cdot)$ based on on-line available measurements.

4 On-Line Identification of the Key Maps

The identification of the map

$$K_{(i)}(q, \tau) = E_{(i)}^f(q) \cdot D_{(i)}(\tau_u) \tag{20}$$

can be done by identifying both the efficiency map $E_{(i)}^f$ and the model uncertainty related map $D_{(i)}$ using the definitions (13) and (14) respectively. In order to do this, some notations are introduced hereafter in order to simplify the expressions. For instance, the following short expressions related to the cost function are used:

$J_{(i)}^+ = J(p^+(t_{i-1}^u), \hat{x}(t_i^u))$ The value when the iterations start
$\hat{J}_{(i)} = J(p(t_i^u), \hat{x}(t_i^u))$ The value when the iterations stop
$J_{(i)} = J(p(t_i^u), x(t_i^u))$ The effective value given the true state $x(t_i^u)$

More precisely, at instant t_{i-1}^u, the iterations start with the initial guess $J_{(i)}^+ = J(p^+(t_{i-1}^u), \hat{x}(t_i^u))$. During the interval $[t_{i-1}^u, t_i^u]$, $q(t_{i-1}^u)$ iterations are performed and the predicted sub-optimal value

$$\hat{J}_{(i)} = J(p(t_i^u), \hat{x}(t_i^u)) \quad ; \quad p(t_i^u) := \mathcal{S}^{q(t_{i-1}^u)}\Big(p^+(t_{i-1}^u), \hat{x}(t_i^u)\Big) \tag{21}$$

is achieved. However, the true value of the cost function at instant t_i^u is given by

$$J_{(i)} = J(p(t_i^u), x(t_i^u))$$

which is generally different from the predicted value $\hat{J}_{(i)}$ because of model mismatches. Now, the next optimization process (to be performed over $[t_i^u, t_{i+1}^u]$) is initialized using the value $p^+(t_i^u)$ and the predicted future state

$\hat{x}(t^u_{i+1})$ leading to the initial guess $J^+_{(i+1)}$ that may be slightly different from the present value $J_{(i)}$ and so on.

Recall that at each instant t^u_i, the computation of the next updating instant $t^u_{i+1} = t^u_i + \tau_u(t^u_i)$ and the corresponding number of iterations $q(t^u_i)$ is based on the solution of the constrained optimization problem (19). This needs the maps $E^f_{(i)}$ and $D_{(i)}$ to be identified based on the past measurements. However, the identification step has to rely on some beforehand given structure of the functions to be identified. The following structures may be used among many others for $E^f_{(i)}$ and $D_{(i)}$:

$$E^f(q) := \frac{1}{\alpha^f \cdot \max\{0, q - q^f\} + 1} \quad ; \quad D(\tau_u) := 1 + \alpha^D \cdot \tau_u^d \tag{22}$$

where α^f, q^f and α^D are the parameters to be identified while $d \in \mathbb{N}$ is chosen in accordance with the definition of the cost function. Note that α^f monitors the speed of convergence of the optimizer while q^f stands for the minimum number of function evaluations before a decrease in the cost function may be obtained. This *dead-zone* like property is observed in many optimization algorithms. Finally, the coefficient α^D reflects the effect of the model mismatch on the evolution of the cost function.

Consequently, by the very definition of the map $D_{(i)}$, the following straightforward identification rule for α^D can be adopted:

$$\alpha^D_{(i)} := \frac{J_{(i)} - \hat{J}_{(i)}}{\tau_u^d(t^u_{i-1}) \cdot \hat{J}_{(i)}}$$

The estimation of the efficiency map's parameters $q^f_{(i)}$ and $\alpha^f_{(i)}$ is obtained based on the behavior of the iterations that are performed during the last updating period $[t^u_{i-1}, t^u_i]$. This behavior is described by the following sequence:

$$\left\{ d_{(i,j)} := \frac{\hat{J}_{(i,j)}}{J^+_{(i)}} \right\}_{j=0}^{q(t^u_{i-1})} := \left\{ \frac{\mathcal{S}^j(p^+(t^u_i), \hat{x}(t^u_i))}{\mathcal{S}^0(p^+(t^u_i), \hat{x}(t^u_i))} \right\}_{j=0}^{q(t^u_{i-1})} \tag{23}$$

where the notation $\hat{J}_{(i,j)} := \mathcal{S}^j(p^+(t^u_i), \hat{x}(t^u_i))$ is used to refer to the value of the estimated cost after j function evaluations.

Indeed, based on the computed sequence (23), the following estimations of $q^f_{(i)}$ and $\alpha^f_{(i)}$ can be obtained:

$$q^f_{(i)} = \max\left\{ j \in \{1, \ldots, q(t^u_{i-1})\} \quad | \quad d_{(i,j)} = 1 \right\} \tag{24}$$

$\alpha^f_{(i)}$ is the least squares solution of the following system [see (22)]

$$\left[d_{(i,j)} \cdot \max(0, j - q^f_{(i)}) \right] \cdot \alpha^f_{(i)} = 1 - d_{(i,j)} \quad ; \quad j = 1, \ldots, q(t^u_{i-1}) \tag{25}$$

where the least squares problem is obtained by putting together all the linear equations (25) (in the unknown $\alpha^f_{(i)}$) corresponding to the different values of j. Note that the computations in (24) and (25) are straightforward and can therefore be done without significant computational burden when compared to NMPC-related computations.

5 Numerical Investigations

In this section, numerical examples are given in order to illustrate the concepts presented in the preceding sections. First, the form of the cost function used in (19) to determine the optimal updating period is first illustrated under several configurations of the problem's parameters q^f, α^f and α^D. Then a concrete and simple example of closed-loop behavior under model mismatches is proposed to illustrate the efficiency of the proposed scheme.

5.1 Qualitative Analysis

Let us consider an efficiency map E^f and an uncertainty map D that have the structure given by (22). Other structures may be used, our conjecture is that the qualitative conclusions of this paper would remain unchanged. The aim of this section is to show how different sets of parameters involved in (22), namely q^f, α^f and α^D influence the resulting optimal updating period τ_u. More precisely, Figures 1 and 2 underline the influence of the efficiency parameter α^f (Figure 1) and the uncertainty parameter α^D (Figure 2).

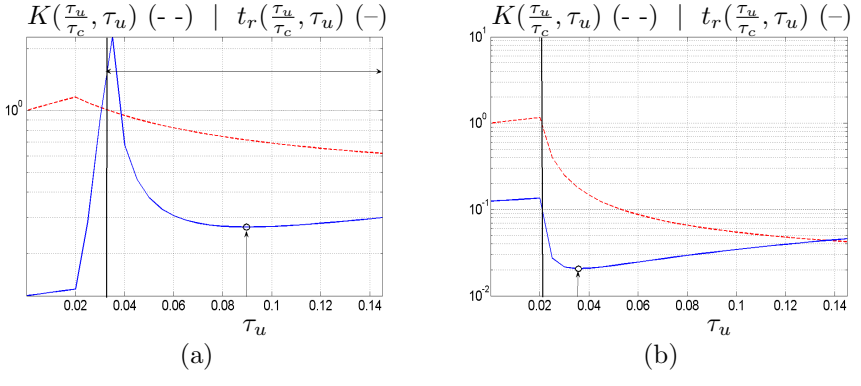


Fig. 1 Variations of the stability indicator $K(\tau_u/\tau_c, \tau_u)$ [dotted line] and the settling time $t_r(\tau_u/\tau_c, \tau_u)$ [solid line] as functions of the updating period τ_u. Figures (a) and (b) correspond to two different values of the efficiency parameter α^f respectively equal to 0.1 (a) and 2 (b). The remaining parameters are identical [$q^f = 4$, $\tau_c = 0.005$, $\alpha^D = 8$ and $d = 1$]. Note the two different corresponding optimal updating periods (minimum of the curve $t_r(\tau_u/\tau_c, \tau_u)$) respectively equal to 90 ms (a) and 35 ms (b)

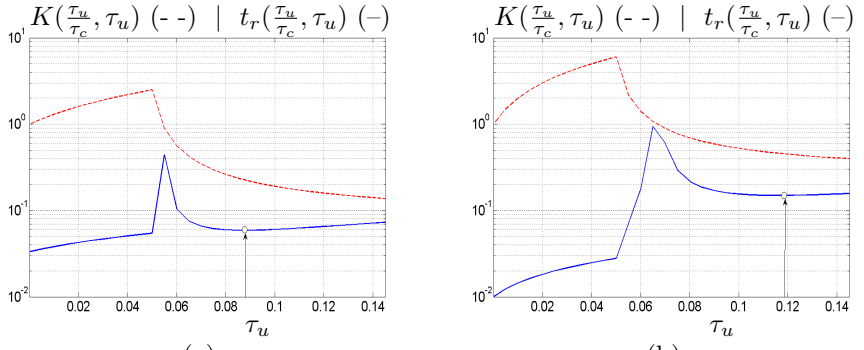


Fig. 2 Variations of the stability indicator $K(\tau_u/\tau_c, \tau_u)$ [dotted line] and the settling time $t_r(\tau_u/\tau_c, \tau_u)$ [solid line] as functions of the updating period τ_u. Figures (a) and (b) correspond to two different values of the uncertainty parameter α^D respectively equal to 8 (a) and 30 (b). The remaining parameters are identical [$q^f = 10$, $\tau_c = 0.005$, $\alpha^f = 2$ and $d = 1$]. Note the two different corresponding optimal updating periods (minimum of the curve $t_r(\tau_u/\tau_c, \tau_u)$) respectively equal to 88 ms (a) and 120 ms (b). Moreover, the use of an updating period $\tau_u \leq 0.052$ for the case (a) or $\tau_u \leq 0.065$ for case (b) may lead to instability

5.2 An Illustrative Example

Let us consider the discrete version of the nonholonomic system in power form

$$x_1^+ = x_1 + u_1 \quad ; \quad x_{j+1}^+ = x_{j+1} + x_1^j u_2 \quad ; \quad j = 1, \ldots, n-1$$

that one aims to stabilize at $x = x^d$ using receding-horizon scheme. In [3], a scalar parametrization of open loop control has been suggested. According to this parametrization, the discrete time open-loop trajectory of x_1 is given by:

$$X_1(0, x^0, p) = x_1^0 \tag{26}$$

$$X_1(k, x^0, p) = (x_1^0 + p) \cdot (1 - \frac{k-1}{N_p - 1}) + \frac{k-1}{N_p - 1}(x_1^d - (x_1^0 + p)) \tag{27}$$

and a corresponding piecewise constant control sequence $\mathbf{u}_1(\cdot) = \mathcal{U}_{pwc,1}(p)$ can be computed. Note that the trajectory defined by (27) is nothing but a piecewise-affine trajectory that passes through p at the next sampling instant and then goes following a straight line to some desired value x_1^d. Injecting this candidate trajectory of x_1 in the equations of the sub-state $z = (x_2, \ldots, x_{n-1})$ results in the following time varying p-dependent linear system:

$$z_{k+1} = [A_k(x_1^0, p)]z_k + [B_k(x_1^0, p)]u_2(k)$$

which can be used to to compute the sequence $\mathbf{u}_2(\cdot) = \mathcal{U}_{pwc,2}(p)$ that minimizes the quadratic cost $\sum_{i=0}^{N_p} \left[\|z(i) - z^d\|^2 + \|u_2(i)\|_R^2\right]$ while meeting the final constraint $z(N_p) = z^d$. Consequently, for each choice of the scalar parameter p, a control profile is obtained and the framework of the paper can be applied with the following cost function $J(p, x^0) = \sum_{i=1}^{N_p} \|X(i, x_0, p)\|^2 + \beta \cdot |p|$. For more details on the parametrization, the reader may refer to [3].

Uncertainty on the system comes form the fact that x^d is not constant but varies according to some unknown dynamics. Moreover, two drift terms η_1 and η_2 are added so that the equations of the real system are given by:

$$x_1^+ = x_1 + u_1 + \eta_1 \quad ; \quad x_{j+1}^+ = x_{j+1} + x_1^j u_2 + \eta_2 \quad ; \quad j = 1, \ldots, n-1$$

The following parameters have been used in the simulations:

$$N_p = 100 \; ; \; \tau_c = 4\tau \; ; \; R = 1 \; ; \; \eta = (-0.01, 0.02) \; \tau = 0.01 \; ; \; \tau_u/\tau \in [5 \; , \; 40]$$
$$n = 2 \; ; \; \beta = 0.1$$

The main MATLAB constrained optimizer FMINCON has been used. The admissible set $\mathbb{P} = [-10, +10]$ has been used for the control parameter p. Finally, in order to meet the translatability property [1] the updating value $p^+ = X_1(2, x^0, p)$ has been used in accordance with the very definition of p.

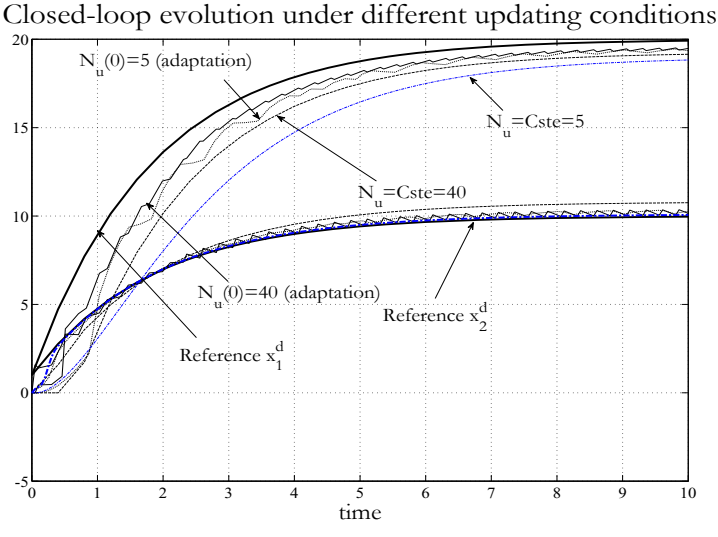


Fig. 3 Closed-loop trajectories under different updating conditions. Namely: two constant updating periods $N_u = 40$ (dashed) and $N_u = 5$ (dotted-dashed) and two different initialization values for the closed-loop updating using the proposed scheme: $N_u(0) = 40$ (solid line) and $N_u(0) = 5$ (dotted)

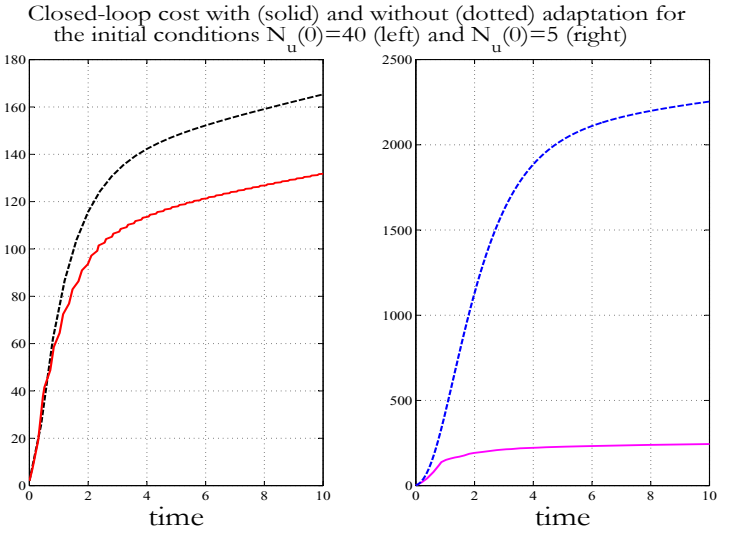


Fig. 4 Evolution of the closed-loop cost (28) with and without updating mechanism for two different initial values $N_u(0) = 40$ (left) and $N_u(0) = 5$ (right)

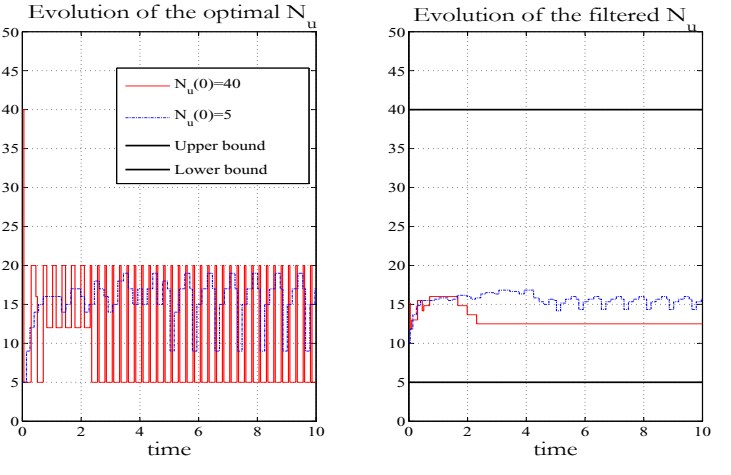


Fig. 5 Evolution of the optimal value $N_u(t_i^u) = \tau_u(t_i^u)/\tau$ for two different initialisation of $N_u(0) = 40$ and $N_u(0) = 5$ (left) and their filtered values (right)

Figure 3 shows the closed-loop trajectory with a fixed or adaptive updating period and for different initial values. The tracking performance is clearly improved by the adaptation mechanism.

This improvement can also be observed on Figure 4 where the evolutions of the closed-loop cost function given by:

$$J_{cl}(k) := \sum_{i=0}^{i=k} \left[\|x(i\tau) - x^d(i\tau)\|^2 + \beta |p(i)| \right] \tag{28}$$

is depicted.

Finally, the evolution of the closed-loop evolution of the updating period $\tau_u(t_i^u) = N_u(t_i^u) \cdot \tau$ during the simulation corresponding to two different initial values $N_u(0) = 40$ and $N_u(0) = 5$ are shown on Figure 5 together with their filtered value. The later is obtained by moving horizon mean computation over 5 successive values. The fact that despite the different initial values, the mean values quickly reach the same region suggests that the latter corresponds to some near optimal behavior.

6 Conclusion

In this paper, a general framework has been proposed for on-line monitoring of control updating period in fast NMPC schemes. The proposed framework can be *superposed* to any existing framework in which iterations are *distributed* over time. Further investigations are needed to understand how this scheme behave when used for different solvers. Deeper insight on the choice of the mathematical structure of the efficiency function is also to be carefully studied.

References

1. Alamir, M.: Stabilization of Nonlinear Systems Using Receding-Horizon Control Schemes: A Parametrized Approach for Fast Systems. LNCIS. Springer, London (2006)
2. Alamir, M.: Nonlinear Receding Horizon sub-optimal Guidance Law for minimum interception time problem. Control Engineering Practice 9(1), 107–116 (2001)
3. Alamir, M., Marchand, N.: Constrained Minimum-Time Oriented Feedback Control for the Stabilization of Nonholonomic Systems in Chained Form. Journal of Optimization Theory and Applications 118(2), 229–244 (2003)
4. DeHaan, D., Guay, M.: Non-Convex optimization and robustness in realtime model predictive control. Internationl J. of Robust and Nonlinear Control 17, 1634–1650 (2007)
5. Diehl, M., Findeisein, R., Bock, H.G., Schlöder, J.P., Allgöwer, F.: Nominal stability of the real-time iteration scheme for nonlinear model predictive control. IEE Control Theory Appl. 152(3), 296–308 (2005)
6. Diehl, M., Bock, H.G., Schlöder, J.P.: A Real-Time Iteration Scheme for Nonlinear Optimization in Optimal Feedback Control. SIAM Journal on Control and Optimization 43(5), 1714–1736 (2005)
7. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: stability and optimality. Automatica 36, 789–814 (2000)
8. Ohtsuka, T.: A Continuation/GMRES Method for Fast Computation of Nonlinear Receding Horizon Control. Automatica 40(4), 563–574 (2004)

Practical Issues in Nonlinear Model Predictive Control: Real-Time Optimization and Systematic Tuning

Toshiyuki Ohtsuka and Kohei Ozaki

Abstract. In this paper, we discuss two important practical issues in nonlinear model predictive control (NMPC): real-time optimization and systematic tuning. First, we present a couple of efficient algorithms based on the assumption that the sampling period is sufficiently short. Real-time algorithms are obtained in a unified manner as initial-value problems of ordinary differential equations (ODEs) for unknown quantities. A brief survey is given on applications of such ODE-type real-time algorithms in mechanical systems. Furthermore, as a first step toward systematic tuning of a performance index, we propose combining feedback linearization with NMPC. The proposed performance index can be tuned with only *one* parameter to adjust the output response and the magnitude of the control input. The effectiveness of the proposed method is demonstrated in numerical examples.

Keywords: real-time algorithm; continuation method; reference input.

1 Introduction

Real-time computation of nonlinear model predictive control (NMPC) poses a challenging problem with regard to its practical implementation and as an area of active research [1, 2, 3, 4]. From a computational viewpoint, the longer the sampling period available for real-time computation, the easier it is to

Toshiyuki Ohtsuka
Department of Systems Innovation, Graduate School of Engineering Science, Osaka University
e-mail: ohtsuka@sys.es.osaka-u.ac.jp

Kohei Ozaki
Department of Mechanical Engineering, Graduate School of Engineering, Osaka University
e-mail: ozaki@newton.mech.eng.osaka-u.ac.jp

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 447–460.
springerlink.com

implement NMPC. In this paper, on the contrary, we review a couple of real-time algorithms [5, 6, 7] that involve no iterative search and are efficient when the sampling period is sufficiently short. The key idea behind them is the representation of a real-time optimization algorithm as a differential equation rather than iterative processes at each sampling time, which is justified by the short sampling period. Then, real-time algorithms can be obtained as initial-value problems of ordinary differential equations (ODEs) for unknown quantities, which are derived by differentiating optimality conditions with respect to time and can be solved without any iterative search. This paper also gives a brief survey on applications of ODE-type real-time algorithms in mechanical systems including a hardware experiment with a sampling period of the order of milliseconds.

Even if real-time optimization becomes a realistic option in implementing NMPC, another fundamental problem for practical application remains—how to choose an NMPC performance index. Since an explicit solution is not available for a nonlinear optimal control problem, it is difficult to clearly relate the closed-loop response to free parameters in a performance index such as weighting matrices and horizon length. Therefore, tuning the performance index to achieve a desired closed-loop performance is laborious. As a first step toward systematic tuning of a performance index, we propose combining feedback linearization with NMPC. If the system output is governed by a linear system with a sufficient number of free parameters, the output response can be specified arbitrarily using linear control theory, though it may require an unrealistic control effort to cancel nonlinearities. To achieve a reasonable trade-off between output response and control effort, we use a performance index to penalize both deviation of the actual control input from the linearizing input and the magnitude of the actual control input. Then, the performance index can be tuned with a single parameter to adjust the characteristic to be emphasized—output response or the magnitude of the control input. The effectiveness and weakness of the proposed method are assessed through numerical simulations.

2 Problem Formulation

We consider a continuous-time system in this paper and assume that every function is differentiable as many times as necessary. The state vector is denoted by $x(t) \in \mathbf{R}^n$, and the control input vector is denoted by $u(t) \in \mathbf{R}^{m_u}$. The state equation and an m_c-dimensional equality constraint are given respectively by

$$\dot{x} = f(x(t), u(t)), \quad C(x(t), u(t)) = 0.$$

An inequality constraint can be converted into an equality constraint by introducing a dummy input [7]. In NMPC, an optimal control problem is

solved at each time t to minimize the following performance index with the initial state given by the actual state $x(t)$:

$$J = \varphi(x(t+T)) + \int_t^{t+T} L(x(t'), u(t'))\,\mathrm{d}t'. \tag{1}$$

The horizon is defined over $[t, t+T]$, i.e., from the current time t to a finite future $t+T$. The optimal control $u^*(t'; t, x(t))$ minimizing J is computed over $t' \in [t, t+T]$, and only its initial value at t, $u^*(t; t, x(t))$ is used as an actual system control input $u(t)$. Then, NMPC results in a kind of state feedback control.

The NMPC problem is essentially a family of finite-horizon optimal control problems along a fictitious time τ as follows:

$$\begin{aligned}
&\text{Minimize}: \quad J = \varphi(x^*(T,t)) + \int_0^T L(x^*(\tau,t), u^*(\tau,t))\,\mathrm{d}\tau,\\
&\text{Subject to}: \begin{cases} x^*_\tau(\tau,t) = f(x^*(\tau,t), u^*(\tau,t)), \quad x^*(0,t) = x(t),\\ C(x^*(\tau,t), u^*(\tau,t)) = 0, \end{cases}
\end{aligned}$$

where subscript τ denotes partial differentiation with respect to τ. The new state vector $x^*(\tau, t)$ represents a trajectory along the τ axis starting from $x(t)$ at $\tau = 0$. The optimal control input $u^*(\tau, t)$ is determined on the τ axis as the solution of the finite-horizon optimal control problem for each t, and the actual control input is given by $u(t) = u^*(0, t)$. The horizon T is a function of time, $T = T(t)$ in general.

Let H denote the Hamiltonian defined by

$$H(x, \lambda, u, \mu) := L(x, u) + \lambda^{\mathrm{T}} f(x, u) + \mu^{\mathrm{T}} C(x, u),$$

where $\lambda \in \mathbf{R}^n$ denotes the costate, and $\mu \in \mathbf{R}^{m_c}$ denotes the Lagrange multiplier associated with the equality constraint. The first-order conditions necessary for the optimal control are obtained by the calculus of variation as the Euler–Lagrange equations [8]:

$$\begin{cases} x^*_\tau = f(x^*, u^*), & x^*(0,t) = x(t),\\ \lambda^*_\tau = -H_x^{\mathrm{T}}(x^*, \lambda^*, u^*, \mu^*), & \lambda^*(T,t) = \varphi_x^{\mathrm{T}}(x^*(T,t)),\\ H_u(x^*, \lambda^*, u^*, \mu^*) = 0, & \\ C(x^*, u^*) = 0. & \end{cases}$$

The Euler–Lagrange equations define a two-point boundary-value problem (TPBVP), in which the initial state is given while the terminal costate is a function of the terminal state. The control input u^* and the Lagrange multiplier μ^* at each time τ on the horizon are determined from x^* and λ^* by algebraic equations $H_u = 0$ and $C = 0$. The nonlinear TPBVP has to be solved within the sampling period for the measured state $x(t)$ at each sampling time, which is one of the major difficulties in NMPC.

3 Real-Time Algorithms

In this section, we derive two real-time NMPC algorithms assuming a sufficiently short sampling time. That is, we regard the update of the optimal solution (more precisely, a stationary solution) as a continuous-time dynamic process described as a differential equation. In the first algorithm, the differential equation for the unknown initial costate $\lambda^*(0,t)$ is derived in a closed form without discretization. The differential equation can be solved as an initial-value problem, which can be viewed as a real-time algorithm without an iterative search for the optimal solution. In the second algorithm, the fictitious time τ on the horizon is discretized, while the real time t remains continuous. Then, the discretized sequence of the control input over the horizon is updated by integrating a differential equation in real time. In the sense that the variation of a solution is traced according to the variation in time t as a parameter, both algorithms can be viewed as a type of continuation method [9].

3.1 Real-Time Costate Equation

If a certain initial value for the costate $\lambda^*(0,t)$ is assumed, the state and costate over the horizon are determined by the initial value problem, although the terminal condition is not necessarily satisfied. To find the initial costate $\lambda^*(0,t)$ without an iterative search, its differential equation along real time t is derived. We denote the unknown quantity $\lambda^*(0,t)$ by $\lambda(t)$. Note that $\dot{\lambda}(t) = \lambda_t^*(0,t)$ holds and λ_t^* is unknown while λ_τ^* is given as $\lambda_\tau^* = -H_x^{\mathrm{T}}$. In this subsection, we regard μ as part of u to simplify expressions, because $C = 0$ is equivalent to $H_\mu = 0$.

To express λ_t^* in terms of the perturbation in $\lambda_\tau^* = -H_x^{\mathrm{T}}$, we derive the following linear TPBVP for $(x_t^* - x_\tau^*, \lambda_t^* - \lambda_\tau^*)$ along the τ axis by differentiating the Euler–Lagrange equations with respect to either t or τ:

$$\begin{cases} \dfrac{\partial}{\partial \tau}\begin{bmatrix} x_t^* - x_\tau^* \\ \lambda_t^* - \lambda_\tau^* \end{bmatrix} = \begin{bmatrix} A & -B \\ -C & -A^{\mathrm{T}} \end{bmatrix}\begin{bmatrix} x_t^* - x_\tau^* \\ \lambda_t^* - \lambda_\tau^* \end{bmatrix}, \\ x_t^*(0,t) - x_\tau^*(0,t) = 0, \\ \lambda_t^*(T,t) - \lambda_\tau^*(T,t) = \varphi_{xx}(x_t^*(T,t) - x_\tau^*(T,t)) + (H_x^{\mathrm{T}} + \varphi_{xx} f)|_{\tau=T}(1+\dot{T}), \end{cases}$$

where matrices A, B, and C are defined along the trajectory (x^*, λ^*) as

$$A = f_x - f_u H_{uu}^{-1} H_{ux}, \quad B = f_u H_{uu}^{-1} f_u^{\mathrm{T}}, \quad C = H_{xx} - H_{xu} H_{uu}^{-1} H_{ux}.$$

It is well known that a linear TPBVP can be solved with backward sweep [8]. In this case, we assume the following relationship and derive the conditions for S^* and c^*.

$$\lambda_t^*(\tau,t) - \lambda_\tau^*(\tau,t) = S^*(\tau,t)(x_t^*(\tau,t) - x_\tau^*(\tau,t)) + c^*(\tau,t), \tag{2}$$

where the matrix $S^*(\tau, t)$ represents the sensitivity of the costate with respect to the state or, equivalently, the Hessian of the optimal cost function along the optimal trajectory [8].

Since $\lambda_t^*(0,t) = \dot{\lambda}(t)$ and $x_t^*(0,t) = x_\tau^*(0,t)$ hold, the differential equation of the unknown quantity $\lambda(t) = \lambda^*(0,t)$ with respect to the real time t is obtained from (2) for $\tau = 0$ as follows [6]:

$$\dot{\lambda}(t) = -H_x^{\mathrm{T}}(x(t), \lambda(t), u(t)) + c^*(0,t), \tag{3}$$

where $u(t)$ is determined from $x(t)$ and $\lambda(t)$ by $H_u(x(t), \lambda(t), u(t)) = 0$, and $c^*(0,t)$ is determined as follows:

$$\begin{cases} x_\tau^*(\tau,t) = f, \quad \lambda_\tau^*(\tau,t) = -H_x^{\mathrm{T}}, \, H_u = 0, \\ x^*(0,t) = x(t), \, \lambda^*(0,t) = \lambda(t). \end{cases} \tag{4}$$

$$\begin{cases} S_\tau^*(\tau,t) = -A^{\mathrm{T}}S^* - S^*A + S^*BS^* - C, \quad S^*(T,t) = \varphi_{xx}|_{\tau=T}, \\ c_\tau^*(\tau,t) = -(A^{\mathrm{T}} - S^*B)c^*, \quad c^*(T,t) = (H_x^{\mathrm{T}} + \varphi_{xx}f)|_{\tau=T}(1+\dot{T}). \end{cases} \tag{5}$$

The differential equation (3) has a form similar to costate equation in the Euler–Lagrange equations. However, (3) can be integrated in real time as an initial value problem, while the Euler–Lagrange equations define a TPBVP. At each time t, (4) is integrated from $\tau = 0$ to $\tau = T$ as an initial value problem, and (5) is integrated backward from $\tau = T$ to $\tau = 0$. Then, $\lambda(t)$ is updated by integrating (3) along the real time t. It is also possible to explicitly derive an equivalent but different form of the costate equation depending on $\dot{x}(t)$ as $\dot{\lambda}(t) = S^*(0,t)\dot{x}(t) + d^*(0,t)$, where $d^* = c^* + \lambda_\tau^* - S^*x_\tau^*$ is determined by the same differential equation as c^* with a different boundary condition [5].

In practice, the initial value $\lambda(0) = \lambda^*(0,0)$ is given by the trivial solution to the TPBVP for $T = 0$, and the horizon length T is smoothly increased to some constant value. Moreover, an error feedback term in the terminal condition is also added in $c^*(T,t)$ to attenuate the error as time increases [6].

One drawback of the algorithm based on the real-time costate equation (3) is that the Riccati-type differential equation for S^* in (5) often involves complicated nonlinear functions due to the second-order partial derivative H_{xx} in the matrix C. Moreover, the number of unknown elements in the symmetric matrix S^* is $n(n-1)/2$, which grows quadratically in the system dimension n. Therefore, even if an iterative search is not needed, it is computationally demanding to solve the differential equations over the horizon at each sampling time.

Another drawback is that the forward integration of the Euler–Lagrange equations is often numerically unstable; this is the same drawback as the classical shooting method for solving an optimal control problem. To avoid the forward integration of the Euler–Lagrange equations, it is common to regard the control input function as the unknown quantity instead of the initial costate. This motivates us to develop the real-time algorithm based on

a differential equation for the input sequence, which is described in the next subsection.

3.2 Continuation/GMRES Method

In this subsection, we regard the control input function over the horizon as the unknown quantity in the TPBVP. To represent the unknown control input function with a finite number of parameters, we discretize the horizon of the optimal control problem into N steps. Then, the discretized conditions for optimality are given as follows:

$$x^*_{i+1}(t) = x^*_i(t) + f(x^*_i(t), u^*_i(t))\Delta\tau, \quad x^*_0(t) = x(t), \tag{6}$$

$$\lambda^*_i = \lambda^*_{i+1} + H^{\mathrm{T}}_x(x^*_i(t), \lambda^*_{i+1}(t), u^*_i(t), \mu^*_i(t))\Delta\tau, \quad \lambda^*_N(t) = \varphi^{\mathrm{T}}_x(x^*_N(t)), \tag{7}$$

$$H^{\mathrm{T}}_u(x^*_i(t), \lambda^*_{i+1}(t), u^*_i(t), \mu^*_i(t)) = 0, \tag{8}$$

$$C(x^*_i(t), u^*_i(t)) = 0, \tag{9}$$

where $\Delta\tau := T/N$. Note that the real time t remains continuous while the fictitious time τ is discretized. On the discretized horizon, sequences of the state, costate, input, and Lagrange multiplier associated with the equality constraint are denoted by $\{x^*_i(t)\}^N_{i=0}$, $\{\lambda^*_i(t)\}^N_{i=0}$, $\{u^*_i(t)\}^{N-1}_{i=0}$, and $\{\mu^*_i(t)\}^{N-1}_{i=0}$, respectively. As a result, NMPC is formulated as a discrete-time TPBVP (6)–(9) for a measured state $x(t)$ at time t.

Let us define vector $U(t) \in \mathbf{R}^{mN}$ $(m := m_u + m_c)$ composed of the sequences of the input vectors and multipliers as

$$U(t) := \left[u^{*\mathrm{T}}_0(t)\ \mu^{*\mathrm{T}}_0(t)\ \cdots\ u^{*\mathrm{T}}_{N-1}(t)\ \mu^{*\mathrm{T}}_{N-1}(t)\right]^{\mathrm{T}}. \tag{10}$$

The sequences of $\{x^*_i(t)\}^N_{i=0}$ and $\{\lambda^*_i(t)\}^N_{i=1}$ can be regarded as functions of $U(t)$ and $x(t)$ because $\{x^*_i(t)\}^N_{i=0}$ and $\{\lambda^*_i(t)\}^N_{i=1}$ can be calculated by (6) and (7) for given $U(t)$ and $x(t)$. Namely, the states $\{x^*_i(t)\}^N_{i=0}$ are calculated recursively by (6), $\lambda^*_N(t)$ is given by (7), and the costates $\{\lambda^*_i(t)\}^{N-1}_{i=1}$ are also calculated recursively by (7). Note that the costate equation is integrated backward, as opposed to the algorithm in the previous subsection.

Then, the optimality conditions (8) and (9) can be regarded as an mN-dimensional equation system given by

$$F(U(t), x(t), t) := \begin{bmatrix} H^{\mathrm{T}}_u(x^*_0, \lambda^*_1, u^*_0, \mu^*_0) \\ C(x^*_0, u^*_0) \\ \vdots \\ H^{\mathrm{T}}_u(x^*_{N-1}, \lambda^*_N, u^*_{N-1}, \mu^*_{N-1}) \\ C(x^*_{N-1}, u^*_{N-1}) \end{bmatrix} = 0, \tag{11}$$

where F depends on t when the horizon length T is time dependent.

Now, let us define a projection $P_0 : \mathbf{R}^{mN} \to \mathbf{R}^{m_u}$ as $P_0(U(t)) := u_0^*(t)$. If (11) is solved with respect to $U(t)$ for the measured $x(t)$ at each sampling time, the control input is determined as $u(t) = P_0(U(t))$, which results in state feedback because the solution $U(t)$ implicitly depends on the current state $x(t)$.

Solving (11) at each time by an iterative method such as Newton's method is computationally expensive and thus inefficient. Instead, we apply the continuation method [9], considering the real time t as the continuation parameter. That is, the time derivative of U is obtained so that (11) is satisfied identically. If the initial solution $U(0)$ of the problem is determined so as to satisfy $F(U(0), x(0), 0) = 0$, then we can trace $U(t)$ by integrating $\dot{U}(t)$ fulfilling the condition:

$$\dot{F}(U(t), x(t), t) = -\xi F(U(t), x(t), t), \tag{12}$$

where ξ is a positive real number. The right-hand side of (12) stabilizes $F = 0$. Equation (12) is equivalent to a linear equation with respect to $\dot{U}$ given by

$$F_U \dot{U} = -\xi F - F_x \dot{x} - F_t. \tag{13}$$

If the matrix F_U is nonsingular, (13) is solved efficiently by GMRES [10], one of the Krylov subspace methods for linear equations. We can update the unknown quantity U by integrating the obtained $\dot{U}$ by, for example, the Euler method in real time. In the case of the explicit Euler method, the computational cost for updating U corresponds to only one iteration in Newton's method but achieves higher accuracy by taking the time dependency of the equation into account. It is worth noting that the optimal control input can be updated by the trail of its derivative with respect to time without using any iterative optimization methods. Since the continuation method is combined with GMRES, this algorithm is called C/GMRES [7]. It is also worth noting that $\dot{U}$ can be integrated with more sophisticated schemes at the expense of simplicity and computational burden. For example, the linearly implicit Euler method [11] could be used for a stiff problem at the expense of approximating the Jacobian of the right-hand side of (13). Developing efficient and reliable integration schemes tailored for NMPC is a topic of future research.

3.3 Applications

Both algorithms in the previous subsection have been implemented in various applications. The real-time costate equation in the form of $\dot{\lambda}(t) = S^*(0,t)\dot{x}(t) + d^*(0,t)$ has been applied to a hardware experiment with a two-wheeled mobile robot [5]. The algorithm was successfully implemented for a three-dimensional state space model with a sampling period of 1/30 s on a 16 MHz CPU. The same idea has also been applied to derive an algorithm for moving horizon state estimation [12, 13], which is a dual of NMPC.

In the state estimation problem, the unknown state is updated by a differential equation, which can be viewed as a generalization of the Kalman filter.

C/GMRES has been applied to a hardware experiment of position control for an underactuated hovercraft [14, 15]. The algorithm was implemented for a six-dimensional state space model with a sampling period of 1/120 s on a 900 MHz CPU. The sampling period was specified by the frame rate of the CCD camera used in the experiment, and the actual computational time for the algorithm was 1.5 ms. Since the laboratory experiment, C/GMRES has been employed in a commercial autopilot system for ships [16] and in an autonomous area avoidance experiment of an experimental airplane [17]. Other applications of C/GMRES include path generation for automobiles [18], a tethered satellite system [19], and other mechanical systems [20, 21, 22]. C/GMRES has also been applied to nonlinear moving horizon state estimation [23]. To avoid laborious calculation and coding of complicated partial derivatives, an automatic code generation program was developed in Mathematica. Given the state equation and the performance index, the Mathematica program generates C codes for simulation of NMPC [24]. Further attempts to improve C/GMRES includes multiple shooting to distribute numerical errors not only in F but also in the state and costate equations [25].

4 Tuning of Performance Index

In the NMPC problem, the performance index is not necessarily restricted to quadratic form for the state and control input, and the characteristics of NMPC depend on the performance index. Therefore, selection of the performance index is as important a problem as real-time algorithms. From a practical viewpoint, it is desirable to choose a performance index consisting of a few parameters that are linked directly with the closed-loop characteristics. As a first step toward developing a systematically tunable performance index, we focus on the transient responses of the controlled output and the control input. The NMPC problem is used for a reasonable trade-off between a desired output response and the necessary control input.

4.1 Reference Model

We consider an n-dimensional single-input single-output nonlinear system:

$$\begin{cases} \dot{x} = f(x) + g(x)u, \\ y = h(x). \end{cases} \tag{14}$$

The present discussion can be readily extended to a multiple-output multiple-input system as long as the number of the outputs is same as that of the inputs.

To construct a reference model for specifying the closed-loop response systematically, we first apply input–output linearization [26]. We assume the system has the relative degree r. That is, there exists a coordinate transformation $z = T(x)$ to transform the system (14) to the following normal form:

$$\frac{d}{dt}\begin{bmatrix} \xi_1 \\ \vdots \\ \xi_{r-1} \\ \xi_r \\ \eta_1 \\ \vdots \\ \eta_{n-r} \end{bmatrix} = \begin{bmatrix} \xi_2 \\ \vdots \\ \xi_r \\ L_f^r h(x) \\ q_1(\xi,\eta) \\ \vdots \\ q_{n-r}(\xi,\eta) \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ L_g L_f^{r-1} h(x) \\ 0 \\ \vdots \\ 0 \end{bmatrix} u, \tag{15}$$

where $L_f h$ denotes the Lie derivative $(\partial h/\partial x) f$ and $L_f^i h := L_f(L_f^{i-1} h)$ $(i \geq 2)$. The output is given by $y = \xi_1$. The new state vector z and the coordinate transformation $T\colon \mathbf{R}^n \to \mathbf{R}^n$ are divided accordingly as $z = [\xi^{\mathrm{T}}, \eta^{\mathrm{T}}]^{\mathrm{T}}$ $(\xi \in \mathbf{R}^r, \eta \in \mathbf{R}^{n-r})$ and $T(x) = [\psi^{\mathrm{T}}(x), \mu^{\mathrm{T}}(x)]^{\mathrm{T}}$ $(\psi : \mathbf{R}^n \to \mathbf{R}^r, \mu : \mathbf{R}^n \to \mathbf{R}^{n-r})$, respectively. Then, a feedback law

$$u_{\mathrm{ref}} = \frac{-L_f^r h(x) + v}{L_g L_f^{r-1} h(x)} \tag{16}$$

linearizes the input–output relation as $y^{(r)} = v$, where v is the new input to specify the closed-loop response. If v is given as

$$v = -(k_1 y^{(r-1)} + \cdots + k_r y), \tag{17}$$

then the output response is governed by the following linear differential equation:

$$y^{(r)} + k_1 y^{(r-1)} + \cdots + k_r y = 0,$$

where $k_1, \ldots, k_r$ are real numbers and are specified so that all characteristic roots are stable and a desired response is achieved. Since there are a sufficient number of free parameters, the characteristic roots are freely assignable. For example, all characteristic roots are set at $s = -1/T_s$ by imposing $s^r + k_1 s^{r-1} + \cdots + k_r = (s + 1/T_s)^r$, where T_s denotes the time constant. The design of the desired response depends on the specific control application and is beyond the scope of this paper. By substituting (17) into (16) and using $y^{(i)} = L_f^i h$ $(0 \leq i \leq r-1)$, the reference control input can be expressed as a function of state x and is denoted by $u = u_{\mathrm{ref}}(x)$.

In the case of $r < n$, even if ξ converges to zero, the dynamics of η remain:

$$\dot{\eta} = q(0, \eta).$$

This is not manipulated directly by input u and is called a zero dynamics. The zero dynamics can be ignored as long as it is stable because it does not affect the output response.

4.2 Performance Index

Since the desired response defined by input–output linearization may require an excessive magnitude of control input, a trade-off between output response and control input should be made. To this end, we define a performance index as follows.

$$J := \int_t^{t+T} [\gamma(u - u_{\mathrm{ref}}(x))^2 + (1-\gamma)u^2]\mathrm{d}t'. \tag{18}$$

The first term in the integrand penalizes the deviation of the actual input u from the reference input $u_{\mathrm{ref}}(x)$, and the second term penalizes the magnitude of the input. The parameter γ adjusts the balance between the first term and the second term over the range $0 \le \gamma \le 1$.

If the system has an unstable zero dynamics, the performance index should also penalize the magnitude of the zero dynamics so that the zero dynamics is stabilized by NMPC. In this case, the performance index is modified as

$$J := \int_t^{t+T} [\gamma(u - u_{\mathrm{ref}}(x))^2 + (1-\gamma)u^2 + \beta\eta^{\mathrm{T}}\eta]\,\mathrm{d}t'. \tag{19}$$

This performance index penalizes the magnitude of the state $\eta = \mu(x)$ of the unstable zero dynamics. The additional parameter $\beta > 0$ is expected to affect the stability of the zero dynamics, although there is no theoretical guarantee.

It should be noted that performance indices (18) and (19) involve, at most, two parameters, γ and β, and their tuning is much simpler than tuning weighting matrices in a widely used quadratic performance index. Although performance indices can also be defined in more general forms, there is a trade-off between the degrees of freedom in a performance index and the complexity in its tuning. For simple tuning, it is necessary to restrict the degrees of freedom to a certain extent.

4.3 Numerical Examples

System without zero dynamics

Consider the following system with the relative degree $r = n$:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_1 + x_1^3 + x_3 \\ \dot{x}_3 = u \end{cases}, \quad y = x_1.$$

Since this system does not have a zero dynamics, the performance index in (18) is used. The horizon length is given by $T = T_f(1 - e^{-\alpha t})$ with $T_f = 1$ and $\alpha = 0.5$. The reference model is expressed as a linear system with a triple root at -3. Figure 1 shows several simulation results for various values of the tuning parameter γ, with the initial state fixed at $x(0) = [1\ 0\ 0]^{\mathrm{T}}$. When γ is decreased, the magnitude of the control input decreases, especially at the initial time, while the output response $y = x_1$ does not change significantly. Therefore, it is possible to make a reasonable trade-off with the proposed performance index for a system without zero dynamics. Note that the performance index (18) is valid even if the system has a stable zero dynamics, because the state η of the stable zero dynamics also converges to zero as the state ξ converges to zero (the simulation results are omitted).

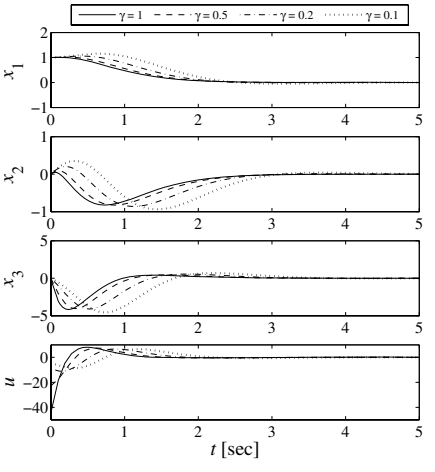


Fig. 1 Time history of states and input ($T_f = 1$)

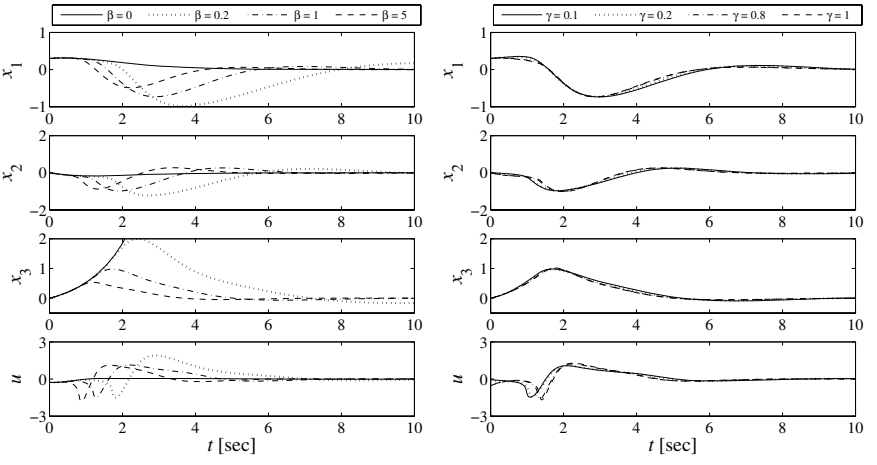
System with unstable zero dynamics

Next, consider the following system:

$$\begin{cases} \dot{x}_1 = x_2 + x_1^2 \\ \dot{x}_2 = x_2^3 + u \\ \dot{x}_3 = x_1 + x_2^3 + x_3 \end{cases}, \quad y = x_1.$$

This system has the relative degree $r = 2$ and an unstable zero dynamics: $\dot{x}_3 = x_3$. Therefore, the performance index (19) with the additional parameter β is used. The poles of the reference model are chosen as multiple roots at -1. The initial state is given as $x(0) = [0.3\ 0\ 0]^{\mathrm{T}}$ and the final horizon length is $T_f = 3$.

Figure 2(a) shows the simulation results for various values of β with $\gamma = 0.5$, and Fig. 2(b) indicates the simulation results for various values of γ with $\beta = 1$. Although the zero dynamics of x_3 diverges when $\beta = 0$, the



(a) $T_f = 3$, $\gamma = 0.5$ (b) $T_f = 3$, $\beta = 1$

Fig. 2 Time history of states and input using performance index (19)

zero dynamics is stabilized by choosing a positive value of β. This demonstrates the effectiveness of the performance index (19), which places an additional penalty on the zero dynamics. However, parameter γ does not affect the closed-loop response in Fig. 2(b). Therefore, parameter β should be determined first to balance responses of the output and zero dynamics, and γ can then be used for fine-tuning the magnitude of the input, which imposes additional tuning steps on a designer.

5 Conclusions

In this paper, two fundamental problems in practical NMPC, real-time algorithms, and tuning of the performance index are discussed. It was shown that algorithms for updating the optimal solution without an iterative search can be derived as ordinary differential equations by assuming a sufficiently short sampling period. Such algorithms have been in fact implemented successfully in hardware experiments with short sampling periods of the order of milliseconds. Ongoing work includes improvement of the algorithms in terms of accuracy and numerical stability.

As a first step toward systematic tuning of the performance index, a particular form of the performance index was proposed for a trade-off between transient responses of the controlled output and the control input. Input–output linearization was used to specify a desired output response, and a single parameter was used for tuning the magnitude of the control input with an acceptable deterioration in the output response. If the zero dynamics is unstable, an additional parameter is introduced to penalize the zero dynamics as well. Numerical examples show that the proposed method is

useful for both unstable and stable zero dynamics, although the magnitude of the control input is not tuned effectively in the case of unstable zero dynamics. Although performance indices in more general forms can also be defined, there is a trade-off between the degrees of freedom in a performance index and the complexity of the tuning. Development of a performance index with a minimal number of design parameters, transparent relationships between parameters and various closed-loop characteristics, and theoretical justification is an open problem for future research.

Acknowledgement. This work was supported by Grant-in-Aid for Scientific Research (C) No. 19560442 from the Japan Society for the Promotion of Science.

References

1. Cannon, M.: Efficient nonlinear model predictive control algorithms. Annual Reviews in Control 28(2), 229–237 (2004)
2. DeHaan, D., Guay, M.: A real-time framework for model-predictive control of continuous-time nonlinear systems. IEEE Transactions on Automatic Control 52(11), 2047–2057 (2007)
3. Diehl, M., Ferreau, H.J., Haverbeke, N.: Efficient numerical methods for nonlinear MPC and moving horizon estimation. In: Magni, L., et al. (eds.): Nonlinear Model Predictive Control. LNCIS, vol. 384, pp. 391–417. Springer Heidelberg (2009)
4. Zavala, V.M., Biegler, L.T.: Nonlinear programming strategies for state estimation and model predictive control. In: Magni, L., et al. (eds.): Nonlinear Model Predictive Control. LNCIS, vol. 384, pp. 419–432. Springer Heidelberg (2009)
5. Ohtsuka, T., Fujii, H.A.: Real-time optimization algorithm for nonlinear receding-horizon control. Automatica 33(6), 1147–1154 (1997)
6. Ohtsuka, T.: Time-variant receding-horizon control of nonlinear systems. Journal of Guidance, Control, and Dynamics 21(1), 174–176 (1998)
7. Ohtsuka, T.: A continuation/GMRES method for fast computation of nonlinear receding horizon control. Automatica 40(4), 563–574 (2004)
8. Bryson Jr., A.E., Ho, Y.C.: Applied Optimal Control. Hemisphere (1975)
9. Richter, S.L., DeCarlo, R.A.: Continuation methods: Theory and applications. IEEE Transactions on Automatic Control AC-28(6), 660–665 (1983)
10. Kelley, C.T.: Iterative Methods for Linear and Nonlinear Equations. SIAM, Philadelphia (1995)
11. Hairer, E., Wanner, G.: Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, 2nd edn. Springer, Heidelberg (1996)
12. Ohtsuka, T., Fujii, H.A.: Nonlinear receding-horizon state estimation by real-time optimization technique. Journal of Guidance, Control, and Dynamics 19(4), 863–870 (1996)
13. Ohtsuka, T.: Nonlinear receding-horizon state estimation with unknown disturbances. Transactions of the Society of Instrument and Control Engineers 35(10), 1253–1260 (1999)
14. Seguchi, H., Ohtsuka, T.: Nonlinear receding horizon control of an underactuated hovercraft. International Journal of Robust and Nonlinear Control 13(3–4), 381–398 (2003)

15. Videos of the experiment,
http://www-sc.sys.es.osaka-u.ac.jp/~ohtsuka/reseach.htm
16. Kohno, Y., Hamamatsu, M., Nakashima, K., Fujimoto, H., Saito, Y., Ikeda, H., Ohnishi, H.: Development of ship maneuvering control system. In: Proceedings of the 2nd SICE Annual Conference on Control Systems, pp. 441–444 (2002) (in Japanese)
17. Nagatsuka, M., Shishido, N., Sudo, I., Yokoyama, A., Masui, K., Tomita, H.: Flight demonstration for autonomous area avoidance algorithm, using receding horizon control. In: Proceedings of the Aircraft Symposium, pp. 201–205 (2006) (in Japanese)
18. Kawabe, T., Nishira, H., Ohtsuka, T.: An optimal path generator using a receding horizon control scheme for intelligent automobiles. In: Proceedings of the 2004 IEEE International Conference on Control Applications, Taipei, pp. 1597–1602 (2004)
19. Okazaki, M., Ohtsuka, T.: Switching control for guaranteeing the safety of a tethered satellite. Journal of Guidance, Control, and Dynamics 29(4), 822–830 (2006)
20. Kawai, Y., Hirano, H., Azuma, T., Fujita, M.: Visual feedback control of an unmanned planar blimp system with self-scheduling parameter via receding horizon control. In: Proceedings of the 2004 IEEE International Conference on Control Applications, Taipei, pp. 1603–1608 (2004)
21. Lee, J., Yamakita, M.: Nonlinear model predictive control for constrained mechanical systems with state jump. In: Proceedings of the 2006 IEEE International Conference on Control Applications, Munich, pp. 585–590 (2006)
22. Saffarian, M., Fahimi, F.: Control of helicopters' formation using non-iterative nonlinear model predictive approach. In: Proceedings of the 2008 American Control Conference, Seattle, pp. 3707–3712 (2008)
23. Soneda, Y., Ohtsuka, T.: Nonlinear moving horizon state estimation with continuation/generalized minimum residual method. Journal of Guidance, Control, and Dynamics 28(5), 878–884 (2005)
24. The Mathematica program,
http://www-sc.sys.es.osaka-u.ac.jp/~ohtsuka/code/index.htm
25. Shimizu, Y., Ohtsuka, T., Diehl, M.: A real-time algorithm for nonlinear receding horizon control using multiple shooting and continuation/Krylov method. In: International Journal of Robust and Nonlinear Control (to appear, 2009)
26. Marquez, H.J.: Nonlinear Control Systems. Wiley Interscience, Hoboken (2003)

Fast Nonlinear Model Predictive Control via Set Membership Approximation: An Overview

Massimo Canale, Lorenzo Fagiano, and Mario Milanese

Abstract. The use of Set Membership (SM) function approximation techniques is described, in order to compute off–line a control law κ^{SM} which approximates a given Nonlinear Model Predictive Control (NMPC) law. The on–line evaluation time of κ^{SM} is faster than the optimization required by the NMPC receding horizon strategy, thus allowing application of NMPC also on processes with "fast" dynamics. Moreover, SM methodology allows to derive approximated control laws with guaranteed worst–case accuracy, which can be suitably tuned to achieve closed loop stability and performance properties that are arbitrarily close to those of the exact NMPC controller. In particular, the properties of three different SM techniques are reviewed here, namely the "optimal", "nearest point" and the "local" approximations, and their performances are compared on a numerical example.

Keywords: MPC approximation, Fast NMPC implementation.

1 Introduction

In Nonlinear Model Predictive Control (NMPC, see e.g. [1]) the control move u_t at time t, for time invariant systems, results to be a nonlinear static function of the system state x_t, i.e. $u_t = \kappa^0(x_t)$, implicitly evaluated on–line by solving a constrained optimization problem. A serious limitation in using NMPC is the presence of fast plant dynamics, which require small sampling periods that do not allow to perform the optimization problem online. In order to solve this problem, a viable solution is the use of function approximation techniques,

Massimo Canale, Lorenzo Fagiano, and Mario Milanese
Politecnico di Torino, Dipartimento di Automatica e Informatica,
Corso Duca degli Abruzzi 24, 10129 Torino, Italy
e-mail: massimo.canale@polito.it,
lorenzo.fagiano@polito.it, mario.milanese@polito.it

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 461–470.
springerlink.com

to compute an approximated NMPC control law requiring lower on–line computational load. Contributions in this field have been given in [2], using neural networks, and in [3], using piecewise affine approximations. The approximation of κ^0 is carried using a finite number ν of exact control moves computed off–line. Indeed, a crucial issue, which arises when an approximated control law is employed, is to obtain guaranteed closed loop stability, performance and constraint satisfaction properties. To overcome these problems, approaches based on Set Membership (SM) approximation techniques have been introduced in [4]–[6]. The effectiveness of these techniques has been also shown in complex applications, like control of tethered airfoils for wind energy generation ([7]) and semi–active suspension control ([8]). SM methodology allows to obtain approximating functions κ^{SM} with guaranteed accuracy, computable in terms of a bound on the error $\kappa^0(x) - \kappa^{\mathrm{SM}}(x)$. Such a bound converges to zero as ν increases, thus with the proper value of ν any desired level of approximation accuracy can be achieved. This way, it is possible to introduce conditions under which guaranteed closed loop stability, performance and constraint satisfaction are guaranteed. Contrary to other existing techniques for NMPC approximation (see e.g. [3]), the only considered assumption with the SM approaches is the continuity of the stabilizing exact NMPC control law, over the set considered for the approximation. In this paper, the properties of three different SM techniques, namely the "optimal" and "nearest point" approximation (see [4]) as well as the "local" approximation (see [6]), are reviewed and their performances are compared on a numerical example. Given the same value of ν, such approaches have different approximation accuracy but also different evaluation times.

2 Problem Settings

Nonlinear Model Predictive Control is a model based control technique where the control action is computed using a receding horizon (RH) strategy, solving at each sampling time t a constrained finite horizon optimal control problem, in which the cost function is evaluated on the basis of the predicted future behaviour of the controlled nonlinear system. The prediction is computed by means of a nonlinear state–space system model, i.e. $x_{t+1} = f(x_t, u_t)$, where $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^m$ are the system state and control input respectively and function f is assumed to be continuous over $\mathbb{R}^n \times \mathbb{R}^m$. The prediction is carried out using the actual measured (or estimated) state x_t as initial condition. It is assumed that the optimization problem considered in the RH strategy is feasible over a set $\mathcal{F} \subseteq \mathbb{R}^n$, which will be referred to as the "feasibility set". The application of the RH strategy gives rise to a nonlinear state feedback configuration, i.e. $x_{t+1} = f(x_t, \kappa^0(x_t)) = F^0(x_t)$, where $\kappa^0(x)$ results to be a nonlinear time invariant control law, i.e. $u_t = [u_{t,1} \dots u_{t,m}]^T = [\kappa_1^0(x_t) \dots \kappa_m^0(x_t)]^T = \kappa^0(x_t)$, $\kappa^0 : \mathcal{F} \to \mathbb{U}$, where $\mathbb{U}$. It is supposed that κ^0 stabilizes the closed loop system and that it is continuous over $\mathcal{F}$.

Such property depends on the considered optimization problem (see [9] and the references therein for further details). In the following, function κ^0 will be denoted as "nominal" or "exact" NMPC law.

A limitation in the practical use of NMPC is the presence of fast plant dynamics, for which the required sampling time may be too low for real-time optimization. A viable solution to this problem is the use of an approximated control function $\kappa^{\mathrm{SM}} \approx \kappa^0$, computed off-line, whose on-line evaluation time is smaller. However, when using an approximated NMPC law, closed loop stability and constraint satisfaction properties are critical issues that have to be treated. In [4] and [5], sufficient conditions for κ^{SM} to guarantee closed loop stability for linear and nonlinear systems have been derived. Such results are now briefly resumed.

2.1 Approximated NMPC Laws: Stability Results

It is considered that κ^{SM} is defined over a compact set $\mathcal{X} \subseteq \mathcal{F}$, containing the origin in its interior. Moreover, κ^{SM} is computed on the basis of the knowledge of a finite number ν of exact control moves, which are computed off–line and stored, i.e.:

$$\tilde{u}^k = \kappa^0(\tilde{x}^k), k = 1, \ldots, \nu \tag{1}$$

where the state values $\tilde{x}^k$ define the set $\mathcal{X}_\nu = \{\tilde{x}^k,\, k = 1, \ldots, \nu\} \subseteq \mathcal{F}$. It is assumed that $\mathcal{X}_\nu$ is chosen such that $\lim_{\nu \to \infty} d_H(\mathcal{X}, \mathcal{X}_\nu) = 0$, where $d_H(X, X_\nu)$ is the Hausdorff distance between $\mathcal{X}$ and $\mathcal{X}_\nu$ (see e.g. [10]). Note that uniform gridding over $\mathcal{X}$ satisfies such condition.

According to the stability results derived in [4], if κ^{SM} has the following *key properties*:

i) Input constraint satisfaction:

$$\kappa^{\mathrm{SM}}(x) \in \mathbb{U}\ \forall x \in \mathcal{X} \tag{2}$$

ii) Bounded pointwise approximation error $\Delta_{\kappa^{\mathrm{SM}}}(x) = \kappa^0(x) - \kappa^{\mathrm{SM}}(x)$:

$$\|\Delta_{\kappa^{\mathrm{SM}}}(x)\| \leq \zeta,\ \forall x \in \mathcal{X} \tag{3}$$

where $\|\cdot\|$ is a suitable norm, e.g. Euclidean.

iii) Convergence of $\zeta(\nu)$ to zero:

$$\lim_{\nu \to \infty} \zeta(\nu) = 0 \tag{4}$$

then it is always possible to explicitly compute a suitable finite value of ν, such that there exists a finite value $\Delta \in \mathbb{R}^+$ with the following characteristics (see [4] for details):

i) the distance between the closed loop state trajectories obtained with the nominal and the approximated control laws is bounded by $\Delta(\nu)$, which can be explicitly computed
ii) $\lim_{\nu\to\infty} \Delta(\nu) = 0$
iii) the closed loop state trajectories, obtained when the approximated control law is used, are kept inside the compact set $\mathcal{X}$, and asymptotically converge to an arbitrarily small neighborhood of the origin

Thus, the aim is to find approximated NMPC laws enjoying properties (2)–(4). Indeed, this can be done in the framework of set membership (SM) approximation ([4], [5]).

3 Set Membership Approximation Techniques for NMPC

In this Section, three Set Membership (SM) techniques will be surveyed. In particular, each approach gives a different compromise between accuracy, on–line computational load and memory usage, while all of them provide approximated control laws satisfying properties (2)–(4). The obtained SM approximations κ^{SM} will be also denoted as Fast Model Predictive Control (FMPC) laws, since their application can significantly reduce the computational load of NMPC.

In the remaining of the paper, for the sake of simplicity it will be assumed that $\mathbb{U} = \{u \in \mathbb{R}^m : \underline{u}_i \leq u_i \leq \overline{u}_i, \imath = 1, \ldots, m\}$, where $\underline{u}_i, \overline{u}_i \in \mathbb{R}$, $i = 1, \ldots, m$.

3.1 Prior Information

As already pointed out, the approximation of function κ^0 is performed on a compact set $\mathcal{X} \subseteq \mathcal{F}$. Since $\mathcal{X}$ and the image set $\mathbb{U}$ of κ^0 are compact, continuity of κ^0 over $\mathcal{F}$ implies that its components κ_i^0, $i = 1, \ldots, m$ are Lipschitz continuous functions over $\mathcal{X}$, i.e. there exist finite constants γ_i, $i = 1, \ldots, m$ such that $\forall x^1, x^2 \in \mathcal{X}$, $\forall i \in [1, m], |\kappa_i^0(x^1) - \kappa_i^0(x^2)| \leq \gamma_i \|x^1 - x^2\|_2$, and $\forall x^1, x^2 \in \mathcal{X}, \|\kappa^0(x^1) - \kappa^0(x^2)\|_2 \leq \|\gamma\|_2 \|x^1 - x^2\|_2$, where $\gamma = [\gamma_1, \ldots, \gamma_m]^T$.

This prior information, together with the knowledge of the off–line computed values (1), can be summarized by concluding that

$$\kappa^0 \in FFS, \tag{5}$$

where the set FFS (Feasible Functions Set) is defined as $FFS = \{\kappa : \mathcal{X} \to \mathbb{U}, \kappa = [\kappa_1, \ldots, \kappa_m]^T : \kappa_i \in FFS_i, \forall i \in [1, m]\}$, with $FFS_i = \{\kappa_i \in \mathcal{A}_{\gamma_i} : \kappa_i(\tilde{x}^k) = \tilde{u}_i^k, \ k = 1, \ldots, \nu\}$ and $\mathcal{A}_{\gamma_i}$ being the set of all continuous Lipschitz functions κ_i on $\mathcal{X}$, with constant γ_i, such that $\underline{u}_i \leq \kappa_i(x) \leq \overline{u}_i$, $\forall x \in \mathcal{X}$. As regards the computation of the Lipschitz constants $\gamma = [\gamma_1, \ldots \gamma_m]$, which

are needed to apply the presented SM techniques, estimates $\hat{\gamma}_i, i = 1, \ldots, m$ can be derived off-line as follows:

$$\hat{\gamma}_i = \inf \left(\tilde{\gamma}_i : \tilde{u}_i^h + \tilde{\gamma}_i \|\tilde{x}^h - \tilde{x}^k\|_2 \geq \tilde{u}_i^k, \ \forall k, h = 1, \ldots, \nu \right) \tag{6}$$

Convergence of $\hat{\gamma}_i$ to $\gamma_i, i = 1, \ldots, m$ has been showed e.g. in [4], i.e. $\lim_{\nu \to \infty} \hat{\gamma}_i = \gamma_i, \ \forall i = 1, \ldots, m$.

3.2 "Optimal" Approximation

The "optimal" (OPT) SM approximation technique and its accuracy and stabilizing properties have been described in [4], for the case of linear systems. With the OPT approach, for a given value of ν the obtained approximating function κ^{OPT} gives the minimal worst–case approximation error.

For each function $\kappa_i^0, \ i \in [1, m]$ and for a given value $x \in \mathcal{X}$, an approximation $\kappa_i^{\text{OPT}}(x) \approx \kappa_i^0(x)$ is computed as (see [4]):

$$\kappa_i^{\text{OPT}}(x) = \tfrac{1}{2}[\overline{\kappa}_i(x) + \underline{\kappa}_i(x)] \in FFS_i \tag{7}$$

where $\overline{\kappa}_i(x) = \sup_{\tilde{\kappa}_i \in FFS_i} \tilde{\kappa}_i(x)$ and $\underline{\kappa}_i(x) = \inf_{\tilde{\kappa}_i \in FFS_i} \tilde{\kappa}_i(x)$, called *optimal bounds*, are the tightest upper and lower bounds of $\kappa_i^0(x)$. As shown in [4], the optimal bounds can be computed as $\overline{\kappa}_i(x) \doteq \min[\overline{u}_i, \min_{k=1,\ldots,\nu} \left(\tilde{u}_i^k + \gamma_i \|x - \tilde{x}^k\|_2 \right)] \in FFS_i$ and $\underline{\kappa}_i(x) \doteq \max[\underline{u}_i, \max_{k=1,\ldots,\nu} \left(\tilde{u}_i^k - \gamma_i \|x - \tilde{x}^k\|_2 \right)] \in FFS_i$. Function κ_i^{OPT} (7) is such that the related *guaranteed approximation error* $E(\kappa_i^{\text{OPT}}) \doteq \sup_{\tilde{\kappa}_i \in FSS_i} \|\tilde{\kappa}_i - \kappa_i^{\text{OPT}}\|_p$ is minimal for any L_p–norm, with $p \in [1, \infty]$, and it is therefore equal to the *radius of information* $r_{p,i}$ (see e.g. [11]).

Then, the optimal SM approximation is defined as $\kappa^{\text{OPT}} = \left[\kappa_1^{\text{OPT}}, \ldots, \kappa_m^{\text{OPT}}\right]^T$. It can be proved that ([4]):

$$\kappa^{\text{OPT}} : \mathcal{X} \to \mathbb{U} \tag{8a}$$

$$\|\Delta_{\kappa^{\text{OPT}}}(x)\|_2 = \|\kappa^0(x) - \kappa^{\text{OPT}}(x)\|_2 \leq \|r_\infty\|_2 = \zeta^{\text{OPT}}, \ \forall x \in \mathcal{X} \tag{8b}$$

$$\lim_{\nu \to \infty} \zeta^{\text{OPT}}(\nu) = 0 \tag{8c}$$

with $r_\infty = [r_{\infty,1}, \ldots, r_{\infty,m}]$. Thus, function κ^{OPT} satisfies the key properties (2)–(4). As regards the computation of $r_{\infty,i}, i = 1, \ldots, m$, the results presented in [12] can be employed.

3.3 "Nearest Point" Approximation

Function κ^{OPT} provides an approximation with minimal guaranteed error, however its on–line computational load may result too high for the considered application. The "Nearest Point" (NP) approach, originally introduced

in [5], is a very simple example of a technique that may overcome this problem, giving an approximating function whose accuracy is not the optimal one, but whose computation is simpler. In particular, for a given value of ν, the NP approximation leads in general to a higher approximation error bound ζ^{NP} than OPT approximation, but to lower on-line computation times, whose growth as a function of ν is much slower than that of OPT approximation. Thus, the NP approximation required to guarantee given stability and performance properties may need much lower on-line computation times with respect to OPT approximation, at the expenses of longer off-line computation time and higher memory usage.

For any $x \in \mathcal{X}$, denote with $\tilde{x}^{\mathrm{NP}} \in \mathcal{X}_\nu$ a state value such that $\|\tilde{x}^{\mathrm{NP}} - x\|_2 = \min_{\tilde{x} \in \mathcal{X}_\nu} \|\tilde{x} - x\|_2$. Then, the NP approximation $\kappa^{\mathrm{NP}}(x)$ is computed as $\kappa^{\mathrm{NP}}(x) = \kappa^0(\tilde{x}^{\mathrm{NP}})$. Such approximation trivially satisfies condition (2). Moreover, it can be shown that also the key properties (3), (4) are satisfied (see [5]):

$$\|\Delta_{\kappa^{\mathrm{NP}}}(x)\|_2 = \|\kappa^0(x) - \kappa^{\mathrm{NP}}(x)\|_2 \leq \zeta^{\mathrm{NP}} = \|\gamma\|_2 \, d_H(\mathcal{X}, \mathcal{X}_\nu), \ \forall x \in \mathcal{X} \quad (9a)$$

$$\lim_{\nu \to \infty} \zeta^{\mathrm{NP}}(\nu) = 0 \quad (9b)$$

3.4 "Local" Approximation

A limitation of the optimal SM approach is that the considered prior information (5) and, consequently, the obtained guaranteed approximation error ζ^{OPT}, may prove to be too conservative, since global Lipschitz constants γ_i, $i = 1, \ldots, m$ over $\mathcal{X}$ are considered. Such conservativeness may be reduced if local information on κ^0 could be taken into account, e.g. by partitioning the set $\mathcal{X}$ into a finite number p of subregions and computing for each one the related local Lipschitz constants $\gamma_{i,j}$, $i = 1, \ldots, m$, $j = 1, \ldots, p$. However, with such procedure problems would arise both in the off–line computation, e.g. regarding the choice of the subregions, and in the on–line implementation, to search for the active subregion (i.e. the subregion in which the actual state x_t lies in). A much simpler approach, to achieve the same goal, is the "Local" SM technique (LOC), which allows to improve the performance of the OPT technique, by reducing its conservativeness (see [6]). The key idea, derived from [13], is to compute an optimal SM approximation of κ^0, satisfying properties (2)–(4), starting from a preliminary approximating function $\hat{\kappa}$. The latter has to be a continuous function which can be computed with any approximation method (e.g. neural networks, polynomial series, PWL approximation).

For each function κ_i^0, $i = 1, \ldots, m$, consider the related residue function $\Delta_{\hat{\kappa},i} = \kappa_i^0 - \hat{\kappa}_i$, which results to be Lipschitz continuous over $\mathcal{X}$ with constant $\gamma_{\Delta_{\hat{\kappa},i}}$, computable numerically using a procedure similar to (6). Then, the information available on κ_i^0 is summarized by the set $FFS_{\Delta,i} = \{\kappa_i : \mathcal{X} \to \mathbb{U}, \ \kappa_i - \hat{\kappa}_i \in \mathcal{A}_{\gamma_{\Delta_{\hat{\kappa},i}}}, \ \kappa_i(\tilde{x}) = \tilde{u}_i, \ \forall \tilde{x} \in \mathcal{X}_\nu\}$, where $\mathcal{A}_{\gamma_{\Delta_{\hat{\kappa},i}}} = \{\Delta_i : \mathcal{X} \to \mathbb{R},$

$|\Delta_i(x^1) - \Delta_i(x^2)| \leq \gamma_{\Delta_{\hat{\kappa},i}} \|x^1 - x^2\|_2,\ \forall x^1, x^2 \in \mathcal{X}\}$. Note that such prior information may prove to be less conservative than (5), because it also takes into account the "information" provided by the preliminary approximation $\hat{\kappa}$. Then, define the following function:

$$\kappa_i^{\mathrm{LOC}} = \hat{\kappa}_i + \frac{1}{2}[\overline{\Delta}_{\hat{\kappa},i}(x) + \underline{\Delta}_{\hat{\kappa},i}(x)] \tag{10}$$

with $\overline{\Delta}_{\hat{\kappa},i}$, $\underline{\Delta}_{\hat{\kappa},i}$ being the optimal upper and lower bounds of $\Delta_{\hat{\kappa},i}$:

$$\begin{aligned} \overline{\Delta}_{\hat{\kappa},i}(x) &\doteq \min[\overline{u}_i - \hat{\kappa}_i(x), \min_{\tilde{x}\in\mathcal{X}_\nu} \left(\Delta_{\hat{\kappa},i}(\tilde{x}) + \gamma_{\Delta_{\hat{\kappa},i}}\|x - \tilde{x}\|_2\right)] \\ \underline{\Delta}_{\hat{\kappa},i}(x) &\doteq \max[\underline{u}_i - \hat{\kappa}_i(x), \max_{\tilde{x}\in\mathcal{X}_\nu} \left(\Delta_{\hat{\kappa},i}(\tilde{x}) - \gamma_{\Delta_{\hat{\kappa},i}}\|x - \tilde{x}\|_2\right)] \end{aligned} \tag{11}$$

Similarly to the OPT approach, function κ_i^{LOC} is such that the related guaranteed approximation error $E(\kappa_i^{\mathrm{LOC}}) \doteq \sup_{\tilde{\kappa}_i \in FSS_{\Delta,i}} \|\tilde{\kappa}_i - \kappa_i^{\mathrm{LOC}}\|_p$ is minimal for any L_p–norm, with $p \in [1, \infty]$, and it is equal to the radius of information $r_{p,\Delta,i}$ of $FFS_{\Delta,i}$. The local SM approximation $\kappa^{\mathrm{LOC}} \approx \kappa^0$ is then computed as $\kappa^{\mathrm{LOC}} = \left[\kappa_1^{\mathrm{LOC}}, \ldots, \kappa_m^{\mathrm{LOC}}\right]^T$. It can be proved that (see [6]):

$$\kappa^{\mathrm{LOC}} : \mathcal{X} \to \mathbb{U} \tag{12a}$$

$$\|\Delta_{\kappa^{\mathrm{LOC}}}(x)\|_2 = \|\kappa^0(x) - \kappa^{\mathrm{LOC}}(x)\|_2 \leq \|r_{\infty,\Delta}\|_2 = \zeta^{\mathrm{LOC}},\ \forall x \in \mathcal{X} \tag{12b}$$

$$\lim_{\nu\to\infty} \zeta^{\mathrm{LOC}}(\nu) = 0 \tag{12c}$$

with $r_{\infty,\Delta} = [r_{\infty,\Delta,1}, \ldots, r_{\infty,\Delta,m}]$. Thus, function κ^{LOC} satisfies the key properties (2)–(4). Note that the OPT approach can be seen as a particular case of the LOC technique, i.e. using $\hat{\kappa} = 0$. Moreover, it can be noted that if $\zeta^{\mathrm{LOC}} < \zeta^{\mathrm{OPT}}$, then the guaranteed accuracy obtained with κ^{LOC} is higher than the one given by κ^{OPT}. As a consequence, a lower number ν of off-line computed values are sufficient for κ^{LOC} to achieve given guaranteed stability and performance properties according to [4]. Lower ν numbers may lead to lower function evaluation times, depending also on the computational burden of the preliminary approximation $\hat{\kappa}$.

4 Numerical Example

Consider the nonlinear Duffing oscillator (see e.g. [14]):

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = u(t) - 0.6\, x_2(t) - x_1(t)^3 - x_1(t) \end{cases}$$

with input constraint set $\mathbb{U} = \{u \in \mathbb{R} : |u| \leq 5\}$. A discrete time model, obtained by forward difference approximation with sampling time $T_s = 0.05\,\mathrm{s}$,

Table 1 Example: mean evaluation times and maximum trajectory distances

	κ^{OPT}		κ^{NP}		κ^{LOC}	
	$\overline{t}$	$\overline{d}$	$\overline{t}$	$\overline{d}$	$\overline{t}$	$\overline{d}$
$\nu = 3.5\,10^5$	$6\,10^{-2}$ s	$1\,10^{-3}$	$1\,10^{-5}$ s	$3\,10^{-3}$	$6\,10^{-2}$ s	$2\,10^{-4}$
$\nu = 1.4\,10^4$	$2\,10^{-3}$ s	$4\,10^{-3}$	$1\,10^{-5}$ s	$1.5\,10^{-2}$	$2\,10^{-3}$ s	$6\,10^{-4}$
$\nu = 3.5\,10^3$	$6\,10^{-4}$ s	$8\,10^{-3}$	$1\,10^{-5}$ s	$3\,10^{-2}$	$6\,10^{-4}$ s	$6\,10^{-3}$

has been used in the nominal NMPC design. In particular, the NMPC controller κ^0 has been designed using prediction horizon $N_p = 100$, control horizon $N_c = 5$ and per-stage cost $L(x,u) = x^T x$ (see [15]). The considered state constraint set is $\mathbb{X} = \{x \in \mathbb{R}^2 : \|x\|_\infty \leq 3\}$. The state prediction has been performed setting $u_{t+j|t} = u_{t+N_c-1|t}$, $j = N_c, ..., N_p - 1$. The optimization problem employed to compute $\kappa^0(x)$ has been solved using a sequential constrained Gauss-Newton quadratic programming algorithm (see e.g. [16]), where the underlying quadratic programs have been solved using the Matlab® function `quadprog`. The maximum and mean computational times of the on-line optimization were $6\,10^{-1}$ s and $4.3\,10^{-2}$ s respectively, using Matlab® 7 with an AMD Athlon(tm) 64 3200+ with 1 GB RAM. The set $\mathcal{X}_\nu$ has been chosen with uniform gridding over $\mathcal{X}$ and different values of ν. On the basis of such off–line computed data, the approximating functions κ^{OPT}, κ^{NP} and κ^{LOC} have been computed. A single layer neural network with 7 nodes and sigmoidal activating function has been used as preliminary approximation to compute κ^{LOC}. Extensive Monte Carlo simulations have been performed with each control law, derived with each ν value. The maximum computational times with $\nu = 3.5\,10^3$ were $1.0\,10^{-3}$s with OPT, $1.1\,10^{-3}$s with LOC and $1.1\,10^{-5}$s with NP approximation. The mean computational time $\overline{t}$, over all time steps of all simulations, and the maximum Euclidean distance $\overline{d}$, between the closed loop state trajectories obtained with κ^0 and with any of the approximated laws, are reported in Table 1. Indeed, the presented example has the aim of comparing the computational efficiency of the various methods in relative terms only. To this end, it can be noted that the mean computational times of the approximated controllers may be up to 4000 times lower than on–line optimization. The NP approximation κ^{NP} achieves the lowest value of $\overline{t}$, which is also independent on ν: this can be obtained with a suitable data arrangement. Functions κ^{OPT} and κ^{LOC} have better precision than κ^{NP} with the same ν value, but also higher computational times, which grow linearly with ν. The approximation κ^{LOC} is able to improve the precision with respect to κ^{OPT}, with the same mean computational time. As regards the memory usage required by the SM approximations, about 90 KBytes, 340 KBytes and 8.4 MBytes were needed with $\nu = 3.5\,10^3$, $\nu = 1.4\,10^4$ and $\nu = 3.5\,10^5$ respectively, without any effort to improve the storage efficiency and using 8 Bytes for all the values contained in the off–line computed data.

5 Conclusions

The SM approaches OPT, NP and LOC to compute an approximation of a NMPC law have been surveyed. Such techniques guarantee closed loop stability, performance and state constraint satisfaction properties. The only required assumption is the continuity of the stabilizing nominal NMPC law. As it has been also showed through a nonlinear oscillator example, a compromise between accuracy, off–line computation, memory usage and on–line evaluation time can be obtained, by suitably choosing and designing one of the reviewed techniques. In particular, given the same off-line computed data, the NP approach has the fastest evaluation time, but worse performance than the other two techniques. The OPT approach has higher on–line computational complexity but also better accuracy, which may be further improved with the LOC technique, provided that a continuous preliminary approximating function is computed. Computational times of OPT and LOC approaches are similar and they grow linearly with ν, while the evaluation time of NP technique can be made practically constant with ν, using a suitable arrangement of the data. Thus, a desired accuracy level can be obtained either with OPT/LOC techniques, with lower memory usage and higher computational time, or with the NP approximation, with higher memory usage but much faster on-line computation.

References

1. Allgöwer, F., Zheng, A.: Nonlinear model predictive control. Wiley, New York (2000)
2. Parisini, T., Zoppoli, R.: A receding-horizon regulator for nonlinear systems and a neural approximation. Automatica 31, 1443–1451 (1995)
3. Johansen, T.: Approximate explicit receding horizon control of constrained nonlinear systems. Automatica 40, 293–300 (2004)
4. Canale, M., Fagiano, L., Milanese, M.: Set membership approximation theory for fast implementation of model predictive control laws. Automatica 45, 45–54 (2009)
5. Canale, M., Fagiano, L., Milanese, M.: Fast nonlinear model predictive control using set membership approximation. In: 17^{th} IFAC World Congress, Seoul, Korea (2008)
6. Canale, M., Fagiano, L., Milanese, M.: On the Use of Approximated Predictive Control Laws for Nonlinear Systems. In: 47^{th} IEEE Conference On Decision and Control, Cancun, Mexico (2008)
7. Canale, M., Fagiano, L., Milanese, M.: Power kites for wind energy generation. IEEE Control Systems Magazine 27(6), 25–38 (2007)
8. Canale, M., Milanese, M., Novara, C.: Semi-active suspension control using "fast" model-predictive techniques. IEEE Transactions on Control System Technology 14, 1034–1046 (2006)
9. Spjøtvold, J., Tøndel, P., Johansen, T.A.: Continuous selection and unique polyhedral representation of solutions to convex parametric quadratic programs. Journal of Optimization Theory and Applications 134, 177–189 (2007)

10. Blagovest, C.S.: Hausdorff Approximations. Springer, Heidelberg (1990)
11. Traub, J.F., Woźniakowski, H.: A General Theory of Optimal Algorithms. Academic Press, New York (1980)
12. Milanese, M., Novara, C.: Computation of local radius of information in SM-IBC-identification of nonlinear systems. Journal of Complexity 23, 937–951 (2007)
13. Milanese, M., Novara, C.: Set membership identification of nonlinear systems. Automatica 40, 957–975 (2004)
14. Jordan, D.W., Smith, P.: Nonlinear ordinary differential equations. Oxford University Press, Oxford (1987)
15. Goodwin, G.C., Seron, M.M., De Doná, J.A.: Constrained Control and Estimation - An Optimisation Approach. Springer, London (2005)
16. Nocedal, J., Wright, S.: Numerical Optimization. Springer, Heidelberg (2006)

Fast Nonlinear Model Predictive Control with an Application in Automotive Engineering

Jan Albersmeyer, Dörte Beigel, Christian Kirches, Leonard Wirsching, Hans Georg Bock, and Johannes P. Schlöder

Abstract. Although nonlinear model predictive control has become a well-established control approach, its application to time-critical systems requiring fast feedback is still a major computational challenge. In this article we investigate a new multi-level iteration scheme based on theory and algorithmic ideas from [2], and extending the idea of real-time iterations as presented in [4]. This novel approach takes into account the natural hierarchy of different time scales inherent in the dynamic model. Applications from aerodynamics and chemical engineering have been successfully treated. In this contribution we apply the investigated multi-level iteration scheme to fast optimal control of a vehicle and discuss the computational performance of the scheme.

Keywords: nonlinear model predictive control, direct multiple shooting, realtime optimal control, multi-level iteration scheme, computational results.

1 Introduction

Given a system state x_0, a nonlinear model predictive control (NMPC) scheme obtains a feedback control $u^*(t; x_0)$ by solving an open-loop optimal control problem (OCP) on a prediction horizon $[t_0, t_f]$. Based on this optimal solution, the associated nonlinear control law $u^*(t; x_0)$ is fed back to the system. At the next sampling point, the procedure is repeated for the new system state. The feedback control is obtained as a minimizer of an objective function that is evaluated based on the trajectories $x(t)$ of the dynamic

Jan Albersmeyer, Dörte Beigel, Christian Kirches, Leonard Wirsching,
Hans Georg Bock, and Johannes P. Schlöder
Interdisciplinary Center for Scientific Computing (IWR)
Ruprecht-Karls-Universität Heidelberg
Im Neuenheimer Feld 368, D-69120 Heidelberg, Germany
e-mail: christian.kirches@iwr.uni-heidelberg.de

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 471–480.
springerlink.com

process under consideration, modelled by an ODE system, and subject to inequality path constraints h,

$$\min_{x(\cdot),u(\cdot)} \quad \int_{t_0}^{t_\mathrm{f}} L\left(x(t),u(t)\right)\ \mathrm{d}t + E\left(x(t_\mathrm{f})\right) \tag{1a}$$

$$\text{s.t.} \quad x(t_0) = x_0, \tag{1b}$$

$$\dot{x}(t) = f\left(t,x(t),u(t)\right), \tag{1c}$$

$$0 \le h\left(x(t),u(t)\right). \tag{1d}$$

The present contribution is concerned with efficient ways to solve the OCP under real time conditions for processes with dynamics on multiple time scales. Exemplarily, a disturbance rejection scenario for a nonlinear vehicle system is considered.

Vehicle Model

A nonlinear single-track car model featuring a Pacejka type tire model is used for the vehicle dynamics [7]. It is described by an ODE system in seven states $x = (c_\mathrm{x}, c_\mathrm{y}, v, \delta, \beta, \psi, w_\mathrm{z})$ and three controls $u = (w_\delta, F_\mathrm{B}, \phi)$

$$\begin{aligned}
\dot{c}_\mathrm{x}(t) &= v(t)\ \cos\big(\psi(t)-\beta(t)\big),\\
\dot{c}_\mathrm{y}(t) &= v(t)\ \sin\big(\psi(t)-\beta(t)\big),\\
\dot{v}(t) &= \frac{1}{m}\Big((F_\mathrm{lr}(F_\mathrm{B},\phi) - F_\mathrm{Ax})\ \cos\beta(t) + F_\mathrm{lf}(F_\mathrm{B})\ \cos\big(\delta(t)+\beta(t)\big)\\
&\qquad - (F_\mathrm{sr} - F_\mathrm{Ay})\ \sin\beta(t) - F_\mathrm{sf}\ \sin\big(\delta(t)+\beta(t)\big)\Big),\\
\dot{\delta}(t) &= w_\delta,\\
\dot{\beta}(t) &= w_\mathrm{z}(t) - \frac{1}{m\,v(t)}\Big((F_\mathrm{lr}(F_\mathrm{B},\phi) - F_\mathrm{Ax})\ \sin\beta(t) + F_\mathrm{lf}(F_\mathrm{B})\ \sin\big(\delta(t)+\beta(t)\big)\\
&\qquad + (F_\mathrm{sr} - F_\mathrm{Ay})\ \cos\beta(t) + F_\mathrm{sf}\ \cos\big(\delta(t)+\beta(t)\big)\Big),\\
\dot{\psi}(t) &= w_\mathrm{z}(t),\\
\dot{w}_\mathrm{z}(t) &= \frac{1}{I_\mathrm{zz}}\big(F_\mathrm{sf}\ l_\mathrm{f}\ \cos\delta(t) - F_\mathrm{sr}\ l_\mathrm{sr} - F_\mathrm{Ay}\ e_\mathrm{SP} + F_\mathrm{lf}(F_\mathrm{B})\ l_\mathrm{f}\ \sin\delta(t)\big).
\end{aligned}$$

where the coordinates, angles, and forces are visualized in Figure 1. Details on the moment of inertia I_zz and the mass m can be found in [7] as well. The model is nonlinear in the states as well as in the control ϕ.

Disturbance Rejection Scenario

A vehicle of 1.300 kg mass is driving on a straight lane at a speed of 30 m/s. After 2 seconds, an impulse of magnitude $2 \cdot 10^4$ N is acting on the rear axle perpendicular to the driving direction for 0.1 seconds. Applying the optimal offline control for driving on the straight lane, the impact is strong enough

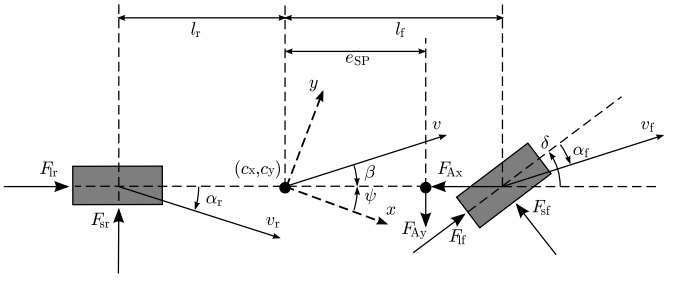


Fig. 1 Coordinates, forces, and angles of the single-track vehicle model. The figure is aligned with the vehicle's local coordinate system, while the dashed pair of vectors (x, y) depicts the global coordinate system used for computations

to make the car spin multiple times and force it off the lane. The model thus is not open-loop stable for this scenario.

Aim of the controller is to keep the vehicle on the lane while retaining a speed of 30 m/s. The full system state information is available at a resolution of 0.05 seconds.

The NMPC formulation of the scenario contains a least-square objective to minimize the deviation from the straight lane as well as the prescribed velocity. Further, the controls w_δ, F_B, ϕ are regularized over the prediction horizon.

Two-sided simple bounds on all states and controls are formulated, while no nonlinear constraints are present.

2 Direct Multiple Shooting for NMPC

This section briefly sketches the direct multiple shooting method, described in [3, 8]. The purpose of the method is to transform the infinite-dimensional OCP (1) into a finite-dimensional nonlinear program (NLP) and then solve it with a structure-exploiting sequential quadratic programming (SQP) approach.

2.1 Parameterization of the Infinite OCP

For a suitable partition of the horizon $[t_0, t_f]$ into N subintervals $[t_i, t_{i+1}]$, $0 \leq i < N$, we discretize the control function u as a piecewise constant vector function

$$\hat{u}_i(t, q_i) = q_i, \ t \in [t_i, t_{i+1}], \ 0 \leq i < N.$$

Furthermore, we introduce additional variables s_i that serve as initial values for computing the state trajectories independently on the subintervals

$$\dot{x}_i(t) = f(t, x_i(t), \hat{u}_i(t, q_i)),\ x_i(t_i) = s_i,\ t \in [t_i, t_{i+1}],\ 0 \le i < N.$$

To ensure continuity of the optimal trajectory on the whole interval $[t_0, t_\mathrm{f}]$ we add matching conditions to the optimization problem

$$s_{i+1} = x_i(t_{i+1}; t_i, s_i, q_i),\ 0 \le i < N$$

where $x_i(t; t_i, s_i, q_i)$ denotes the solution of the IVP on $[t_i, t_{i+1}]$, depending on s_i and q_i. This method allows using state-of-the-art adaptive integrators, cf. [1]. The path constraints (1d) are enforced in the shooting nodes t_i.

2.2 Structure-Exploiting Nonlinear Programming

From the multiple shooting discretization we obtain the structured NLP

$$\min_{s,q} \quad \sum_{i=0}^{N-1} L_i(s_i, q_i) + E(s_N) \tag{3a}$$

$$\text{s.t.} \quad 0 = s_0 - x_0, \tag{3b}$$

$$0 = s_{i+1} - x_i(t_{i+1}; t_i, s_i, q_i), \quad 0 \le i < N \tag{3c}$$

$$0 \le h(s_i, q_i), \quad 0 \le i \le N. \tag{3d}$$

Note that this NLP depends parametrically on x_0. It can be written in the generic form $\min_w \phi(w)$ s.t. $c(w) + Lx_0 = 0$, $d(w) \ge 0$, where $L = (-I_{n_x}, 0, 0, \dots)$ and $w = (s_0, q_0, \dots, s_{N-1}, q_{N-1}, s_N)$ the vector of all unknowns.

Sequential Quadratic Programming

The NLP is solved by a Newton-type framework. Various structural features are exploited by tailored linear algebra as detailed in [3, 8]. Starting with an initial guess (w^0, λ^0, μ^0), a full step SQP iteration is performed as follows

$$w^{k+1} = w^k + \Delta w^k, \quad \lambda^{k+1} = \lambda^k_{\mathrm{QP}}, \quad \mu^{k+1} = \mu^k_{\mathrm{QP}} \tag{4}$$

where $(\Delta w^k, \lambda^k_{\mathrm{QP}}, \mu^k_{\mathrm{QP}})$ is the solution of the QP subproblem

$$\min_{\Delta w} \quad \tfrac{1}{2} \Delta w^\intercal B^k \Delta w + b^{k\intercal} \Delta w \tag{5a}$$

$$\text{s.t.} \quad 0 = C^k \Delta w + c(w^k) + Lx_0, \tag{5b}$$

$$0 \le D^k \Delta w + d(w^k), \tag{5c}$$

Here, B^k denotes an approximation of the Hessian of the Lagrangian of (3), and b^k, C^k and D^k are the objective gradient and the Jacobians of the constraints c and d.

3 The Multi-level Iteration Scheme

In NMPC, the optimal control problem (3) has to be treated for varying system states x_0 obtained online.

3.1 Initial Value Embedding and Real-Time Iterations

In [4] and related works it has been proposed to initialize the current problem with the full solution of the previous optimization run, i.e., control *and* state variables. The arising violation of the initial value constraint (3b) is cured after the first full Newton-type step due to its linearity in x_0. Furthermore, for each x_0 only one SQP iteration is done providing already a first-order approximation of the solution. The classical real-time iteration (RTI) scheme separates each iteration into the following three phases.

Preparation

Using the iterate of the previous step (w^k, λ^k, μ^k) for function and derivative evaluation it is possible to assemble QP (5) almost completely without a measurement of the current system state $x_0(t^k)$. Due to its special structure, the variables $(\Delta s_1, \ldots, \Delta s_N)$ can be eliminated from QP (5). This is usually referred to as the *condensing* step.

Feedback

As soon as $x_0(t^k)$ is available, Δs_0 can be eliminated as well and a small QP only in the variables $(\Delta q_0, \ldots, \Delta q_{N-1})$ is solved. The feedback $u(t^k) := q_0^k + \Delta q_0^k$ is then given to the real system. Thus, the actual feedback delay reduces to the solution time of the condensed QP. The affine-linear dependence of the QP on $x_0(t^k)$ could be further exploited as described in [6].

Transition

Finally, the eliminated variables are recovered and step (4) is performed to obtain the new set of NLP variables $(w^{k+1}, \lambda^{k+1}, \mu^{k+1})$.

3.2 Multi-level Extensions

In [2], the RTI scheme is extended to new multi-level iteration schemes. The main idea is to deal with different time scales inherent in the system, e.g. validity of linearizations, on different levels in the algorithm. High-frequency feedback is provided by a fast basic mode, called mode A. Depending on the characteristics of the process, a selected subset of data used in mode A is updated by several higher-level modes. Mode B would provide feasibility improvements [2], while mode C provides adjoint sensitivity information for systems with many differential states [9]. Mode D computes a new full linearization of the system. In the presented example neither nonlinear

constraints nor a large number of system states are present. We therefore consider a scheme assembled from modes A and D, in which mode D is used each n_{D} iterations.

Mode A: Linear MPC on a Reference Trajectory

The basic idea is to regard QP (5) as a linear MPC controller that is set up with a Hessian approximation $\overline{B}$, objective gradient $\overline{b}$, constraint values $\overline{c}$, $\overline{d}$, and Jacobians $\overline{C}$, $\overline{D}$, and working on a reference solution $(\overline{w}, \overline{\lambda}, \overline{\mu})$. This LMPC controller supplies the process with control feedback $u(t^k) := \overline{q}_0 + \Delta q_0^k$.

Mode D: Full RTI Preparation Phase

This variant updates the data of mode A each n_{D} steps by a single preparation phase, i.e. calculation of a new Hessian approximation, objective gradient as well as constraint residuals and Jacobians. Furthermore, the updated QP (5) is recondensed.

Using the terminology of modes, the classical real-time iteration scheme is denoted by A^1D^1. In [5] nominal stability of the RTI scheme has been proven. However, stability of schemes $A^{n_{\mathrm{D}}}D^1$ for $n_{\mathrm{D}} \geq 2$ is still an open issue.

Applying mode D often, i.e. n_{D} close to 1, is costly and may become real-time infeasible, but tracks fast changes in the system linearization. On the other hand, applying D scarcely, i.e. n_{D} high, is cheap but comes close to LMPC which shows inferior performance for nonlinear models.

4 Computational Results

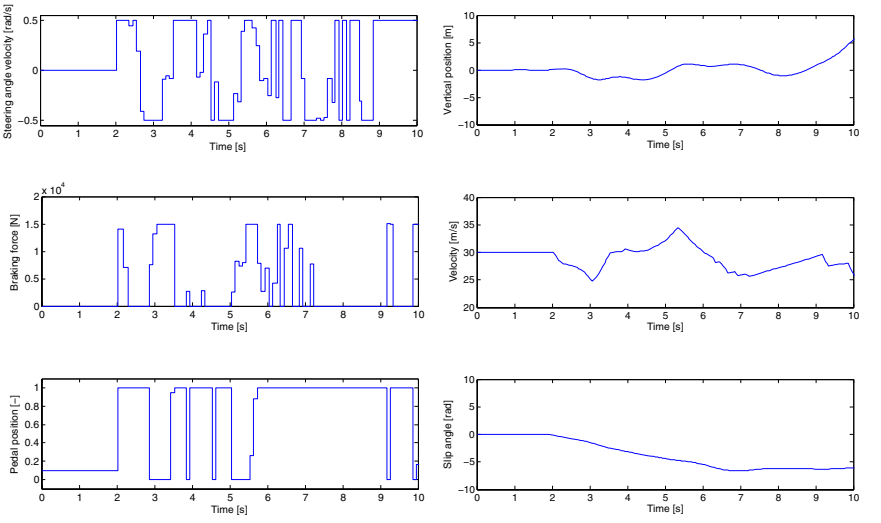
The delay due to computation time spent in the feedback phase is properly taken into account in the vehicle system simulation. All computation times are given for an *Intel Pentium 4* machine with *2.8 GHz* and *3 GB RAM*, running *SuSE Linux 10.1*.

4.1 Classical Real-Time Iterations

From Figure 2 we can see that a sampling rate of 0.1 s is insufficient. The vehicle spins uncontrollably, the final deviation from the straight lane exceeds 5 meters, and the velocity of 30 m/s is not maintained. Figure 3 shows the controller performance for a sampling rate of 0.05 s. Obviously, the shorter sampling intervals allow the RTI scheme to reject the disturbance.

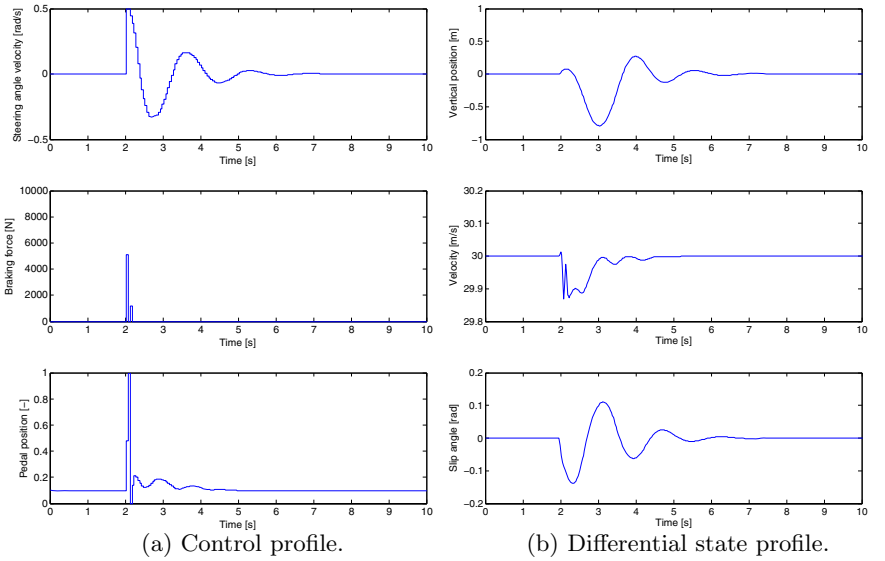
4.2 Multi-level Iteration Schemes

All controller schemes A^2D^1 to A^6D^1 are able to reject the disturbance, while those which apply mode D more often are able to catch nonlinear effects better and generate "cleaner" control profiles, although at higher



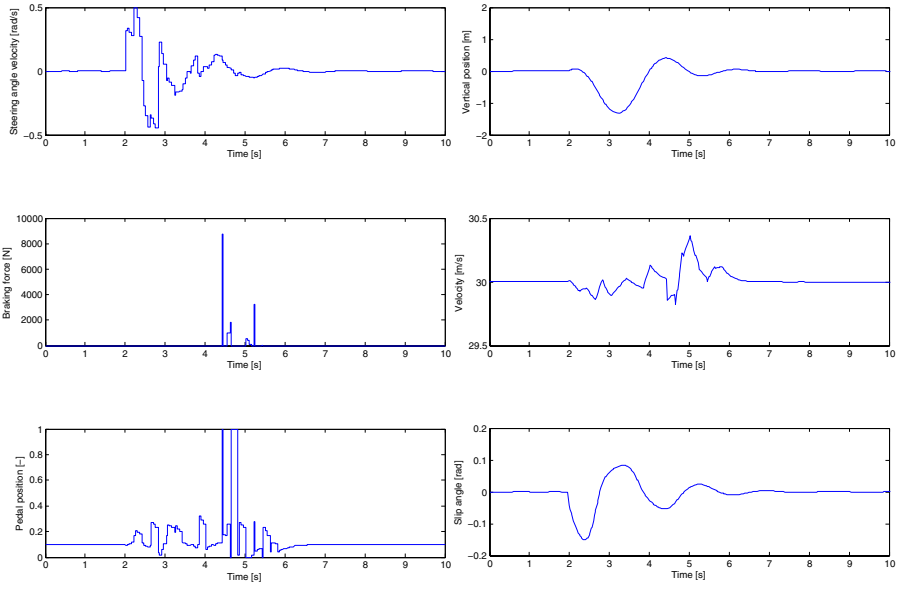
(a) Control profile. (b) Differential state profile.

Fig. 2 Unsatisfactory performance of the A^1D^1 controller scheme for a sampling time of 0.1 seconds. The controller is unable to reject the impact. From the right column it can be seen that the vehicle leaves the straight lane (top), the velocity v drops below 25 m/s (center), while the vehicle spins



(a) Control profile. (b) Differential state profile.

Fig. 3 Performance of the A^1D^1 controller scheme for a sampling time of 0.05 seconds. Within less than 5 seconds, the controller recovers from the impact suffered at $t = 2$ s. The left column depicts the optimal controls (from top to bottom: steering angle velocity w_δ, braking force F_B, pedal position ϕ). The right column shows a selection of resulting differential states (deviation from the straight lane c_y, velocity v, slip angle β)



(a) Control profile. (b) Differential state profile.

Fig. 4 Performance of the A^4D^1 controller scheme for a sampling time of 0.05 seconds. Even doing a mode D step only every 0.2 seconds is still sufficient. Note the impact of the nonlinear updates of mode D on the control trajectories generated by the mode A controller. Every 4 control interval (0.2 seconds time), a new piecewise constant approximation to a control arc starts

Table 1 Computation times for the results presented in Figures 3, and 4

Controller	Mode A Calls	Avg. [s/call]	Mode D Calls	Avg. [s/call]	Total [s]
A^1D^1	200	–	200	0.019	3.76
A^2D^1	200	0.005	101	0.015	2.43
A^4D^1	200	0.005	51	0.015	1.77
A^6D^1	200	0.006	34	0.016	1.86

computational cost. The results depicted in Figure 4 show the vehicle behavior under control profiles generated by an A^4D^1 multi-level iteration scheme.

4.3 Computation Times

In Table 1 computation times for the multi-level NMPC implementation within the optimal control software package `MUSCOD-II` [8] are shown. All controllers are faster than the classical real-time iterations, while the most performant controller is A^4D^1.

Notably, when the nonlinear updates performed by mode D are applied scarcely to save on computation time (A^6D^1 scheme), the mode A workload starts to increase again due to more active set changes occurring during the dense QP solution. It is important to recognize that the mode A computation times are independent of the differential state dimension, whereas that dimension dominates the mode D computation times. This decoupling of computation time dependencies easily allows the multi-level algorithm to scale to larger models without impacting the LMPC feedback rate.

5 Conclusions

In this contribution, we presented a new multi-level NMPC controller scheme based on theoretical concepts from [2]. As discussed, the controller easily scales with the ODE model dimension while maintaining constant fast LMPC feedback rates independent of the model size. It reflects dynamics on different time scales inherent in the model by providing several independent options to update the core LMPC feedback law using feasibility, optimality, and nonlinearity improvements. Finally, the design allows to distribute available processing power among multiple modes to be executed in parallel, according to scenario requirements.

Using the described NMPC controller we studied a disturbance rejection scenario from automotive engineering. Compared to classical real-time iterations [4] we were able to reduce the feedback delay by a factor of three. We have shown that for the scenario at hand, the nonlinearity update frequency of mode D can be reduced to every sixth control feedback with only very limited loss of performance of the resulting controller. In a practical application, this rate could be chosen according to the embedded control system's capabilities.

References

1. Albersmeyer, J., Bock, H.G.: Sensitivity Generation in an Adaptive BDF-Method. In: Bock, H.G., Kostina, E., Phu, X.H., Rannacher, R. (eds.) Modeling, Simulation and Optimization of Complex Processes: Proc. Int. Conf. on High Performance Scientific Computing, Hanoi, Vietnam, March 6-10, 2006, pp. 15–24. Springer, Heidelberg (2008)
2. Bock, H.G., Diehl, M., Kostina, E.A., Schlöder, J.P.: Constrained Optimal Feedback Control for DAE. In: Biegler, L., Ghattas, O., Heinkenschloss, M., Keyes, D., van Bloemen Waanders, B. (eds.) Real-Time PDE-Constrained Optimization, pp. 3–24. SIAM, Philadelphia (2007)
3. Bock, H.G., Plitt, K.J.: A multiple shooting algorithm for direct solution of optimal control problems. In: Proc. 9th IFAC World Congress Budapest, pp. 243–247. Pergamon Press, Oxford (1984)

4. Diehl, M., Bock, H.G., Schlöder, J.P., Findeisen, R., Nagy, Z., Allgöwer, F.: Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. J. Proc. Contr. 12(4), 577–585 (2002)
5. Diehl, M., Findeisen, R., Allgöwer, F., Bock, H.G., Schlöder, J.P.: Nominal Stability of the Real-Time Iteration Scheme for Nonlinear Model Predictive Control. IEE Proc.-Control Theory Appl. 3, 296–308 (2005)
6. Ferreau, H.J., Bock, H.G., Diehl, M.: An online active set strategy to overcome the limitations of explicit mpc. Int. J. of Robust and Nonlinear Control 18(8), 816–830 (2008)
7. Kirches, C., Sager, S., Bock, H.G., Schlöder, J.P.: Time-optimal control of automobile test drives with gear shifts. Optimal Control Applications and Methods (submitted, 2009)
8. Leineweber, D.B., Bauer, I., Schäfer, A.A.S., Bock, H.G., Schlöder, J.P.: An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization (Parts I and II). Computers and Chemical Engineering 27, 157–174 (2003)
9. Wirsching, L., Bock, H.G., Diehl, M.: Fast NMPC of a chain of masses connected by springs. In: Proc. IEEE Int. Conf. on Control Applications, Munich, pp. 591–596 (2006)

Unconstrained NMPC Based on a Class of Wiener Models: A Closed Form Solution

Shraddha Deshpande, V. Vishnu, and Sachin C. Patwardhan

Abstract. In this work, we obtain a closed form solution to a nonlinear model predictive control problem when the controller is developed using a class of black-box Wiener type model. The linear dynamic component of the Wiener model parameterized using orthonormal basis filters. The nonlinear state to output map is constructed using quadratic polynomials. The resulting nonlinear state space model is then used to formulate an unconstrained NMPC problem. A closed form solution to this problem is constructed analytically using the theory of solutions of quadratic operator polynomials. The efficacy of the proposed quadratic control law is demonstrated by conducting servo control studies at a singular operating point on a benchmark experimental setup.

Keywords: Wiener Model, Fast NMPC, Multidimensional Quadratic Polynomials, Input Multiplicities, Control at Optimum.

1 Introduction

Nonlinear model predictive control is increasingly being used for controlling systems, such as fuel cells, which exhibit strongly nonlinear and significantly fast dynamics. To control such systems efficiently, it is necessary to develop fast solution approaches that can facilitate real-time implementation of NMPC. In this work, we obtain a closed form solution to a nonlinear

Shraddha Deshpande and V. Vishnu
Systems and Control Engineering

Sachin C. Patwardhan
Dept. of Chemical Engineering, Indian Institute of Technology,
Bombay, Mumbai, India
e-mail: sachinp@iitb.ac.in

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 481–490.
springerlink.com

model predictive control problem when the controller is developed using a class of black-box Wiener type model. Dynamics of a multivariable nonlinear system with fading memory is captured by employing MISO Wiener type discrete state space models with nonlinear output error structure [2]. The linear dynamic part of the Wiener model is parameterized using generalized ortho-normal basis filters. The nonlinear state to output map is constructed using quadratic polynomials. It is shown that the multi-step unconstrained NMPC formulation can be rearranged as a single multi-dimensional quadratic equation. Using the theory of solutions of quadratic operator polynomials in Banach spaces [1], we construct an analytical solution to this multi-step predictive control problem. The solution procedure involves computations of matrix square roots and this can give rise to multiple and /or complex solutions. The control law is constructed by making a qualified choice of the matrix square root and choosing to implement real part of the complex solutions. The efficacy of the proposed closed form control law is demonstrated by conducting experimental studies on the benchmark Heater-Mixer set up [5].We demonstrate that the proposed quadratic control law is able to move the system to an optimum (singular) operating point, where the steady state gain is reduced to zero and the gain changes its sign across the optimum, from a sub-optimal point and maintain the operation at the optimum.

2 OBF-Wiener Model

The nonlinear dynamic systems under consideration are with fading memory and can be represented as a set of nonlinear ODEs of the form

$$\frac{d\mathbf{z}}{dt} = \mathbf{F}\left[\mathbf{z}, \mathbf{u}\right] \quad ; \quad \mathbf{y}(t) = \mathbf{G}\left[\mathbf{z}\right] + \mathbf{v}(k) \tag{1}$$

where $\mathbf{u} \in R^m$ represent the vector of manipulated inputs, $\mathbf{y} \in R^r$ represent the vector of measured outputs and $\mathbf{v} \in R^r$ represent the vector of unknown additive disturbances. In practice, the operators $\mathbf{F}[.]$ and $\mathbf{G}\,[.]$ are seldom known exactly and are too complex to be used for developing a closed form control law. The information available from the plant for model development is the sampled sequence of input and output vectors $\Sigma_{Ny} = \{(\mathbf{y}(k), \mathbf{u}(k)) : k = 1, 2, ...Ny\}$. Given input and output data set Σ_{Ny} collected from a plant, we propose to identify a MIMO Wiener model of the form

$$\mathbf{x}(k+1) = \Phi\,\mathbf{x}(k) + \Gamma\,\mathbf{u}(k) \tag{2}$$

$$\mathbf{y}(k) = \ \Omega\left[\mathbf{x}(k)\right] + \varepsilon(k) = \mathbf{C}\,\mathbf{x}(k) + \{\,\mathbf{D}\,\}\left[\mathbf{x}(k), \mathbf{x}(k)\right] + \varepsilon(k) \tag{3}$$

where $\mathbf{x}(k) \in R^n$ represents model state vector . The matrices $(\mathbf{\Phi}, \mathbf{\Gamma})$ are parameterized using Generalized Orthonormal Basis Filters [2], which represent an orthonormal basis for the set of strictly proper stable transfer functions.

A nested optimization strategy for identification of GOBF poles and matrices [**C**, **D**] is discussed in Srinivasrao et al.[2]. Note that $\{\,\mathbf{D}\,\}$ is a $(r \times N \times N)$ bilinear matrix representation of a three dimensional array (see Appendix and [3, 4] for more details).

3 Multistep Quadratic Control Law

In the development that follows, it is assumed that the system under consideration is square, i.e. $m = r$. The model equation (2) can be used to formulate an state estimator as follows

$$\widehat{\mathbf{x}}(k) = \Phi\,\widehat{\mathbf{x}}(k-1) + \Gamma\,\mathbf{u}(k-1) \tag{4}$$
$$\boldsymbol{\varepsilon}(k) = \mathbf{y}(k) - \Omega\,[\widehat{\mathbf{x}}(k)] \tag{5}$$

At sampling instant k, given p future input moves

$$\mathbf{U}(k) = \left[\,\mathbf{u}(k|k)^T \;...\; \mathbf{u}(k+p-1|k)^T\;\right]^T$$

$(p+j)$-step ahead predictions are generated as follows

$$\widehat{\mathbf{x}}(k+p+j|k) = \boldsymbol{\Phi}^{p+j}\widehat{\mathbf{x}}(k) + \mathbf{S}^{(j)}\,\mathbf{U}(k) \tag{6}$$
$$\widetilde{\mathbf{y}}(k+p+j|k) = \Omega\,[\widehat{\mathbf{x}}(k+p+j|k)] + \mathbf{d}(k) \tag{7}$$

$$\mathbf{S}^{(j)} = \left[\,\boldsymbol{\Phi}^{p+j-1}\;\boldsymbol{\Phi}^{p+j-2}\;....\;\boldsymbol{\Phi}\;\mathbf{I}\,\right]\boldsymbol{\Gamma}$$

where $\mathbf{d}(k)$ accounts for the plant-model mismatch / unmeasured disturbances. The signal $\mathbf{d}(k)$ is estimated by filtering the prediction error, $\boldsymbol{\varepsilon}(k)$, through a unity gain filter as follows

$$\mathbf{d}(k) = \boldsymbol{\Phi}_d\mathbf{d}(k-1) + [\mathbf{I} - \boldsymbol{\Phi}_d]\,\boldsymbol{\varepsilon}(k) \tag{8}$$
$$\boldsymbol{\Phi}_d = diag\left[\,\alpha_1\;\alpha_2\;...\;\alpha_r\,\right] \tag{9}$$

where $0 \le \alpha_i < 1$ are treated as tuning parameters. The above prediction equation can be rearranged as follows

$$\begin{aligned}\widehat{\mathbf{y}}(k+p+j|k) &= \overline{\mathbf{y}}(k+p+j|k) + [\boldsymbol{\Lambda}^{(j)}(k)]\mathbf{U}(k)\\ &\quad + \left\{\boldsymbol{\Psi}^{(j)}\right\}(\mathbf{U}(k),\mathbf{U}(k))\end{aligned} \tag{10}$$

$$\begin{aligned}\overline{\mathbf{y}}(k+p+j|k) &= \mathbf{d}(k) + \mathbf{C}\boldsymbol{\Phi}^{p+j}\widehat{\mathbf{x}}(k)\\ &\quad + \left\{\{\mathbf{D}\}\circ\boldsymbol{\Phi}^{p+j}\bullet\boldsymbol{\Phi}^{p+j}\right\}(\widehat{\mathbf{x}}(k),\widehat{\mathbf{x}}(k))\end{aligned} \tag{11}$$

$$\boldsymbol{\Lambda}^{(j)}(k) = \mathbf{C}\mathbf{S}^{(j)} + 2\left\{\{\mathbf{D}\}\circ\boldsymbol{\Phi}^{p+j}\bullet\mathbf{S}^{(j)}\right\}\widehat{\mathbf{x}}(k) \tag{12}$$

$$\left\{\boldsymbol{\Psi}^{(j)}\right\} = \left\{\{\mathbf{D}\}\circ\mathbf{S}^{(j)}\bullet\mathbf{S}^{(j)}\right\} \tag{13}$$

The symbols ($\circ$) and ($\bullet$) appearing in above equation represent *circle product* and *right dot product* of a bilinear matrix with a matrix, respectively (ref. Appendix for definitions). The prediction equation developed above assumes that p future input moves are available for manipulation. However, in a typical NMPC formulation, the degrees of freedom (q) for future trajectory manipulation are typically chosen fewer than the prediction horizon ($q << p$) and are spread across the horizon through input blocking. The computation of matrices $\mathbf{S}^{(j)}$ has to be modified accordingly to accommodate these constraints.

Now, consider an unconstrained NMPC formulation formulated over prediction horizon $[p, p+q-1]$

$$\min_{\mathbf{U}(k)} \sum_{j=p}^{p+q-1} \left[\mathbf{e}_f(k+j|k)\right]^T \mathbf{e}_f(k+j|k) \tag{14}$$

where $\mathbf{e}_f(k+j|k) = \mathbf{r} - \widehat{\mathbf{y}}(k+j|k)$ and $\mathbf{r} \in R^r$ represent the desired setpoint. The global minimum of the above optimization problem, if it exists, can be obtained by setting

$$\mathbf{e}_f(k+j|k) = \overline{\mathbf{0}} \quad for \quad j = p, ..., p+q-1 \tag{15}$$

where $\overline{\mathbf{0}}$ represents the zero vector. To develop a closed form solution to above control problem, the above set of q vector equations are arranged in a single vector equation as follows

$$\widehat{\mathbf{Y}}(k) = \overline{\mathbf{Y}}(k) + [\mathbf{\Lambda}(k)]\mathbf{U}(k) + \{\Psi\}\left(\mathbf{U}(k), \mathbf{U}(k)\right) \tag{16}$$

$$\widehat{\mathbf{Y}}(k) = \left[\, \widehat{\mathbf{y}}(k+p|k)^T \; ... \; \widehat{\mathbf{y}}(k+p+q-1|k)^T \,\right]^T$$

$$[\mathbf{\Lambda}(k)] = \left[\, \left(\mathbf{\Lambda}^{(1)}(k)\right)^T \; ... \; \left(\mathbf{\Lambda}^{(q)}(k)\right)^T \,\right]^T \quad ; \quad \{\mathbf{\Psi}\} = \left[\begin{array}{c} \{\mathbf{\Psi}^{(1)}\} \\ \hline \{.....\} \\ \hline \{\mathbf{\Psi}^{(q)}\} \end{array}\right]$$

With these definitions, the controller design equation reduces to a multidimensional quadratic equation of the form

$$\mathbf{Q}\left(\mathbf{U}(k)\right) = \{\Psi\}\left(\mathbf{U}(k), \mathbf{U}(k)\right) + [\mathbf{\Lambda}(k)]\mathbf{U}(k) + \left[\overline{\mathbf{Y}}(k) - \mathbf{R}\right] = \overline{0} \tag{17}$$

where $\mathbf{R} = \left[\, \mathbf{r}^T \; .... \; \mathbf{r}^T \,\right]^T$ .The above equation is multi dimensional quadratic operator polynomial in $R^{m \times q}$. An analytical approach for solving such quadratic operator polynomial equations in Banach spaces equations has been developed by Rall [1]. Patwardhan and Madhavan [4] have adopted this approach to develop one step ahead control law under nonlinear IMC [3] framework. Following the controller synthesis approach suggested by

Patwardhan and Madhavan [4], a **multistep quadratic control law** can be derived as follows:

Let $\mathbf{U}_0$ denote some arbitrary input vector such that the gradient matrix

$$\nabla_U \left[Q\left(\mathbf{U}_0\right)\right] = 2\left[\{\Psi\}\left(\mathbf{U}_0\right)\right] + \Lambda(k) \tag{18}$$

is nonsingular. Then, the analytical solution of the above multi-dimensional quadratic equation can be written as

$$\begin{aligned}
\mathbf{U}(k) - \mathbf{U}_0 &= -\left(\frac{1}{2}\left[I + \left(\Delta\left(k\right)\right)^{\frac{1}{2}}\right]\right)^{-1} \left(\nabla_U \left[Q\left(\mathbf{U}_0\right)\right]\right)^{-1} \left[Q\left(\mathbf{U}_0\right)\right] \quad (19)\\
\Delta\left(k\right) &= \left[I - 4\left\{\widetilde{\Psi}\left(k\right)\right\}\left(\nabla_U \left[Q\left(\mathbf{U}_0\right)\right]\right)^{-1}\left[Q\left(\mathbf{U}_0\right)\right]\right]\\
\left\{\widetilde{\Psi}(k)\right\} &= \left(\nabla_U \left[Q\left(\mathbf{U}_0\right)\right]\right)^{-1} \bullet \{\Psi\}
\end{aligned}$$

Here, symbol $(\bullet)$ denotes *left dot product* between matrix $\left(\nabla_U \left[Q\left(\mathbf{U}_0\right)\right]\right)^{-1}$and the bilinear matrix $\{\Psi\}$ (see Appendix). Patwardhan and Madhavan [4] have proved following results, which help in understanding properties of the above solution.

Theorem 1. *[4]: If square roots of matrix $\Delta\left(k\right)$ exist, then it is always possible to select $\Delta\left(k\right)^{1/2}$ such that $(\mathbf{I}+\Delta\left(k\right)^{1/2})$ is nonsingular.*

Lemma 1. *[4]: For the system of linear algebraic equation given by $(\mathbf{I} + \mathbf{A}^{1/2})\mathbf{v} = \mathbf{b}$ where $\mathbf{A}$ is diagonalizable matrix, consider the set*

$$\Xi = \left\{\mathbf{v} : \mathbf{v} = (\mathbf{I} + \mathbf{A}^{1/2})^{-1}\mathbf{b}, \|\mathbf{b}\|_2 \leq 1\right\}$$

The smallest bound for the diameter of set Ξ is obtained by choosing $\mathbf{A}^{1/2}$ such that its eigenvalues have non-negative real parts.

In general, a matrix has multiple square roots and consequently different values of $\mathbf{U}(k)$ will be obtained for every choice of the square root of matrix $\Delta\left(k\right)$. Based on Lemma 2, Patwardhan and Madhavan [4] have suggested that the matrix square root $\left(\Delta\left(k\right)\right)^{\frac{1}{2}}$ should be selected such that all its eigen values have non-negative real part. It may be noted that the matrix square root can have complex elements and consequently the resulting $\mathbf{U}(k)$ can be complex. To alleviate this difficulty, Patwardhan and Madhavan [4] have suggested that the real part of the complex solution vector can be used for manipulation. Incorporating these suggestions, a modified solution to the control problem can be stated as follows

$$\mathbf{U}\left(k\right) = \mathbf{U}_0 - REAL\left\{\left[\frac{1}{2}\left\{I + \left(\Delta\left(k\right)\right)^{\frac{1}{2}}\right\}\right]^{-1} \mathbf{E}_0\left(k\right)\right\} \tag{20}$$

This controller is implemented in a moving horizon framework and only first element of $\mathbf{U}(k)$, i.e. $\mathbf{u}(k|k)$, is implemented on the system. In practice, the unconstrained control law developed above requires additional measures to avoid integral windup, such as incorporating input bounds $\mathbf{u}_L \leq \mathbf{u}(k) \leq \mathbf{u}_H$.The stability of the resulting closed loop can be analyzed using nonlinear internal model control framework [3]. Patwardhan and Madhavan [4] have shown that the situation where the solution becomes complex arises when the specified setpoint is unattainable due to system nonlinearity. However, if the specified setpoint is attainable at the steady state and the prediction horizon is selected sufficiently large, then the complex solutions are not expected to arise during control law implementation.

4 Experimental Studies

In this section we present results on a experimental case study involving a benchmark heater-mixer setup [5, 2]. The heater mixer setup consists of two stirred tanks in series as shown in Figure 1. The cold water inlet flow to both the tanks can be manipulated using pneumatic control valves. The temperatures in the first tank (T_1), in the second tank (T_2) and the liquid level in the second tank (h_2) are measured (controlled) variables while the heat inputs to first and second tank ($\mathbf{m}_1$ and $\mathbf{m}_2$) and cold water flow to

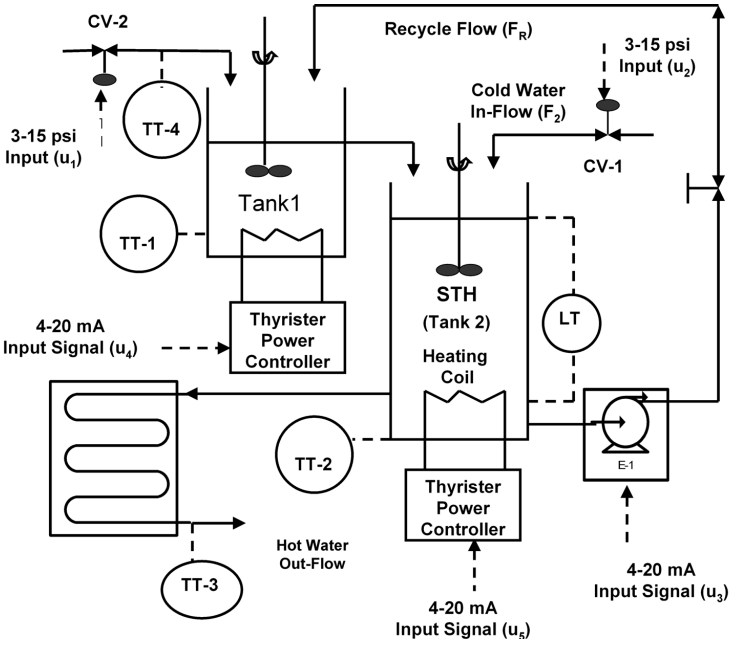


Fig. 1 Heater Mixer Setup: Schematic Diagram

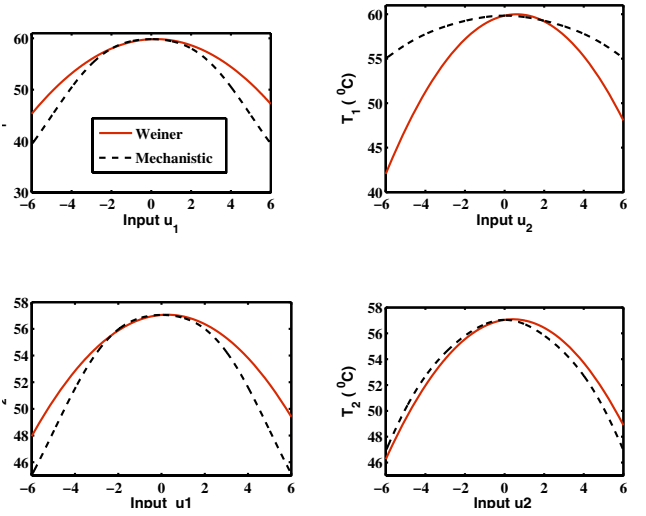


Fig. 2 Heater Mixer Setup: Model Validation - Stedy State Behavior

the second tank ($\mathbf{m}_3$) are used as manipulated inputs. The cold water flow to the first tank and the cold water temperature are treated as unmeasured disturbances and recycle flow is kept constant.

To make the problem difficult and strongly nonlinear, bell shaped nonlinearities of the form

$$\mathbf{m}_1 = 16.3 * exp(-(\mathbf{u}_1/6)^2 \quad ; \quad \mathbf{m}_2 = 18 * exp(-(\mathbf{u}_2/6)^2)$$

are deliberately introduced between the controller outputs ($\mathbf{u}$) and manipulated inputs ($\mathbf{m}$)**.** These nonlinearities gives rise to input multiplicity behavior as shown in Figure 2. The resulting system has an unconstrained optimum operating point at $\mathbf{u} = \overline{\mathbf{0}}$. This is a singular point where the steady state gains with respect to controller outputs ($\mathbf{u}$) are reduced to zero and change their respective signs across the optimum.

The steady state optimum point is chosen as the desired operating point. OBF-Wiener type models with NOE structure were identified from input-output data as proposed by Srinivasrao et al [2]. Figure 2 presents comparison of steady state behavior of the OBF-Wiener model with that of a mechanistic model [2] for Heter-Mixer setup. The identified OBF-Wiener model was able to capture the dynamic as well as the steady state behavior of the system with a reasonable accuracy.

The tuning parameters of quadratic control law are chosen as follows

$$p = 60, \quad \mathbf{\Phi}_d = diag\left[0.9\ 0.9\ 0.92\right]$$
$$-6 \leq u_1 \leq 6 \ ; -6 \leq u_2 \leq 6 \ ; \quad 8\ mA \leq u_3 \leq 20\ mA$$

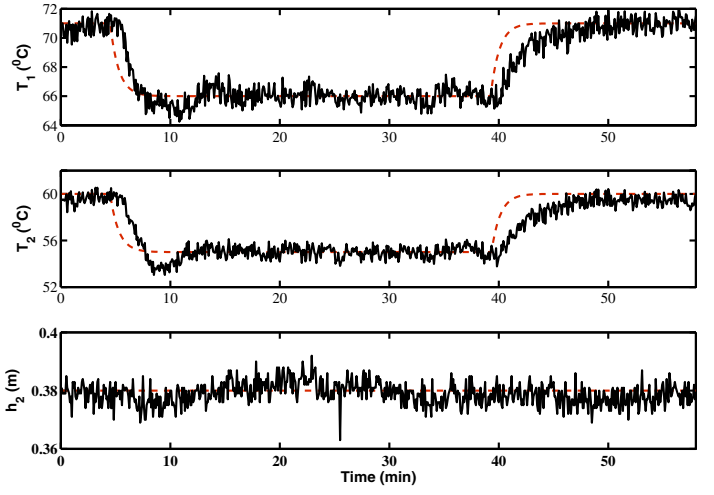


Fig. 3 Quadratic Controller (q =1, C++ DLL) : Behavior of controlled outputs

The controller is required to shift the operation from the optimum point ($T_1 = 71$, $T_2 = 60$ and $h_2 = 0.38$) to a sub-optimal operating point ($T_1 = 66$, $T_2 = 55$ and $h_2 = 0.38$) and then shift the operating point back to the optimum operating point. The closed loop response of the Heater-Mixer system is evaluated for control horizon $q = 1$ and $q = 10$. The closed loop responses obtained for $q = 1$ are presented in Figure 3. As evident from this figure, the quadratic controller smoothly conducts the desired setpoint transition. Moreover, the process is tightly controlled at the singular optimum operating point in spite of fluctuations in unmeasured disturbances.

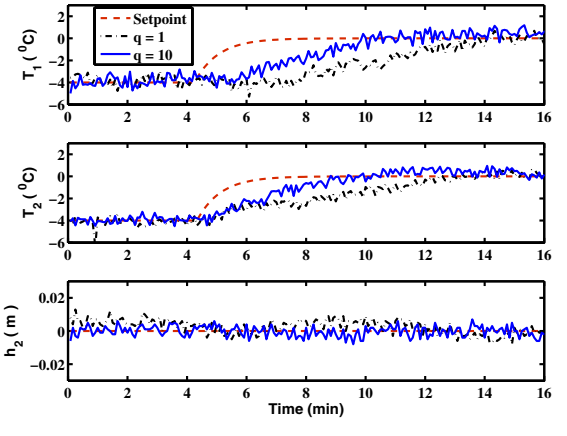


Fig. 4 Comparison of Closed Loop Performances for q = 10 and q = 1

Figure 4 compares response of quadratic control law with ($q = 1$) and ($q = 10$). As can be expected, the controlled output attains the optimum point faster when degrees of freedom are increased to 10. However, if we compare average computation time for quadratic control law implementation, then the controller with $q = 10$ required 311.6 ms while the controller with $q = 1$ required only 9.5 ms. It may be noted that the controller was implemented through a MATLAB window created in LABVIEW. We also implemented the quadratic control law ($q = 1$) by converting MATLAB code to a C++ code and using resulting controller DLL file through LABVIEW. This resulted in a drastic reduction in the mean computation time required for control law computations (0.184 ms). Thus, it is possible to employ the proposed quadratic control law for systems with fast dynamics if $q << p$.

5 Conclusion

In this work, we obtain a closed form solution to a nonlinear model predictive control problem when the controller is developed using a class of black-box Wiener type model. The efficacy of the proposed closed form control law is demonstrated by conducting experimental studies on the benchmark Heater-Mixer set up. We demonstrate that the proposed quadratic control law is able to operate the system at a singular operating point and establish the feasibility of employing the proposed control law for systems with fast dynamics.

Appendix: Bilinear Matrices

A $(r \times n \times m)$ *bilinear matrix* $\{B\}$ is ordered collection of numbers $b_{\alpha\beta\gamma}, \alpha = 1, 2, ...r,\ \beta = 1, 2, ...n,\ \gamma = 1, 2, ...m$. Various operation between a matrix and a bilinear matrix are defined in Table (1).

Table 1 Bilinear Matrix operations

Operation	Representation	Definition
Left Dot Product	$\{D\} = A \bullet \{B\}$ $\{k \times n \times m\}$ $= (k \times r) \bullet \{r \times n \times m\}$	$d_{\alpha\beta\gamma} = \sum_{\eta=1}^{r} a_{\alpha\eta} b_{\eta\beta\gamma}$
Right Dot Product	$\{D\} = \{B\} \bullet A$ $\{r \times n \times k\}$ $= (r \times n \times m) \bullet (m \times k)$	$d_{\alpha\beta\gamma} = \sum_{\eta=1}^{m} b_{\alpha\beta\eta} a_{\eta\gamma}$
Circle Product	$\{D\} = \{B\} \circ A$ $\{r \times m \times k\}$ $= (r \times n \times m) \circ (n \times k)$	$d_{\alpha\beta\gamma} = \sum_{\eta=1}^{n} b_{\alpha\eta\gamma} a_{\eta\beta}$

References

1. Rall, L.B.: Quadratic Equations in Banach Spaces. Rend. Circ. Matem. Palermo-serie II - Tomo X, 314–332 (1961)
2. Srinivasarao, M., Patwardhan, S.C., Gudi, R.D.: From Data to Non-linear Predictive Control: Part I. Identification of Multivariable Nonlinear State Observers. Ind. Eng. Chem. Res. 45, 1989–2001 (2006)
3. Economou, C.G.: An Operator Theory Approach to Nonlinear Controller Design, Ph.D. Dissertation, California Institute of Technology (1985)
4. Patwardhan, S.C., Madhavan, K.P.: Nonlinear Internal Model Control using Quadratic Prediction Models. Comp. Chem. Engng. 22(4/5), 587–601 (1998)
5. Thornhill, N.H., Patwardhan, S.C., Shah, S.L.: A Continuous Stirred Tank Heater Simulation Model with Applications. Journal of Process Control 18, 347–360 (2008)

An Off-Line MPC Strategy for Nonlinear Systems Based on SOS Programming

Giuseppe Franzè, Alessandro Casavola, Domenico Famularo, and Emanuele Garone

Abstract. A novel moving horizon control strategy for input-saturated nonlinear polynomial systems is proposed. The control strategy makes use of the so called sum-of-squares (SOS) decomposition, i.e. a convexification procedure able to give sufficient conditions on the positiveness of polynomials. The complexity of SOS-based numerical methods is polynomial in the problem size and, as a consequence, computationally attractive. SOS programming is used here to derive an "off-line" model predictive control (MPC) scheme and analyze in depth its relevant properties. The main contribution here is to show that such an approach may lead to less conservative MPC strategies than most existing methods based on global linearization approaches. An illustrative example is provided to show the effectiveness of the proposed SOS-based algorithm.

Keywords: Sum of Squares, Nonlinear Systems, Predictive Control, Convex Relaxations, Constrained Systems.

1 Introduction

Model Predictive Control (MPC) is an optimization based control strategy able to efficiently deal with plant constraints. At each time interval, the MPC algorithm computes an open-loop sequence of inputs by minimizing, compatibly with prescribed constraints, a cost index based on future plant predictions. The first input of

Giuseppe Franzè, Alessandro Casavola, and Emanuele Garone
DEIS - Università degli studi della Calabria, Rende (CS), 87036, Italy
e-mail: {casavola,franze,egarone}@deis.unical.it

Domenico Famularo
DIMET - Università degli studi di Reggio Calabria, Via Graziella, Loc. Feo di Vito, Reggio Calabria (RC), 89100, Italy
e-mail: domenico.famularo@unirc.it

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 491–499.
springerlink.com

the optimal sequence is applied to the plant and the entire optimization procedure is repeated at future time instants.

Though almost all processes are inherently nonlinear, the vast majority of MPC applications and results are based on linear or uncertain linear dynamic models (see [1, 2] and references therein). One of the main reasons for this choice is probably related to the huge on-line computational burdens typically resulting from direct nonlinear programming techniques which are, in some cases, non-convex programming algorithms [3, 4].

Nevertheless, there are cases when nonlinear effects are significant enough to justify the use of direct nonlinear MPC (NMPC) technologies, contrasted to linearized MPC approaches (see [5]). These include at least two broad categories of applications: regulation problems, where the plant is highly nonlinear and subject to large frequent disturbances, and servo problems, where the set point changes frequently and spans a sufficiently wide range of nonlinear process dynamics.

The purpose of this paper is to consider a particular class of nonlinear plants and constraints described by means of polynomials. The formulation of the MPC problem in such a case gives rise to polynomial optimization problems solvable by using efficient numerical methods exploiting Gröbner bases, cylindrical algebraic decomposition etc., which have been recently proposed in the literature (see [6, 7, 8, 9]).

In particular, SOS decomposition and semidefinite programming [10, 11, 12, 13, 14] techniques will be used here, whose computational complexity is polynomial in the problem size. Strictly speaking, the SOS-based approach is a powerful convexification method which generalizes the well-known S-procedure [7] by searching for polynomial multipliers. As one of its major merits, the SOS-based approach provides less conservative results than most available methods. Preliminary results along this research line have been achieved in [12] where an on-line constrained MPC strategy for open-loop stable polynomial systems has been developed.

Here we propose an off-line formulation of a Receding Horizon Control (RHC) problem for polynomial systems based on the computation of a nested sequence of positive invariant sets (see [15] where a similar algorithm is detailed for uncertain linear plants). With this off-line approach, the SOS computation time is not a limiting factor and increased control performance can be achieved also for fast processes and large scale nonlinear systems.

2 Problem Formulation

Consider the following nonlinear system with polynomial vector field

$$x(t+1) = f(x(t)) + g(x(t))\,u(t) \tag{1}$$

where $f \in \mathbb{R}^n[x]$, $g \in \mathbb{R}^{n\times m}[x]$, with $x \in \mathbb{R}^n$, denoting the state and $u \in \mathbb{R}^m$ the control input which is subject to the following component-wise saturation constraints

$$u(t) \in \mathcal{U},\ \forall t \geq 0: \quad \mathcal{U} \triangleq \{u \in \mathbb{R}^m \mid |u_i| \leq \bar{u}_i,\ i = 1,\ldots,m,\}. \tag{2}$$

here $\mathcal{U}$ is a compact subset of $\mathbb{R}^m$ containing the origin as an interior point. It is assumed in this paper that f and g are continuous and $0_x \in \mathbb{R}^n$ is an equilibrium point for (1) with $u = 0$, i.e. $f(0) = 0$ [3].

The aim is to find, given a certain initial state $x(0)$, a state feedback regulation strategy $u(t) = g(x(t))$ which, under the prescribed constraints (2), asymptotically stabilizes (1) and minimizes the infinite horizon quadratic cost

$$J(u,x(0)) \triangleq \sum_{t=0}^{\infty} \left(\|x(t)\|^2_{\Psi_x} + \|u(t)\|^2_{\Psi_u} \right) \tag{3}$$

where (3) $\Psi_x = \Psi_x^T$, $\Psi_u = \Psi_u^T$ are positive definite weighting matrices.

In what follows, a Receding Horizon Control (RHC) scheme for the proposed regulation problem will be introduced and outlined. Therefore, by resorting to well-established ideas (see [1] for a comprehensive and detailed discussion), we look for a guaranteed cost, input constrained state feedback regulation strategy.

We recall that sufficient conditions guaranteeing the feasibility and closed-loop stability of the RHC paradigm for nonlinear discrete-time systems have been presented in [16]. There, under mild assumptions, it has been proved that the RHC optimization problem has a solution if a non-increasing Lyapunov function $V(x(\cdot))$, can be found (Fundamental Theorem, pag. 293, [16]) and the regional stability of the feedback control system is achieved. Therefore, under a state-dependent control law $u(t) = K(x(t))$, with $x(t)$ the initial state, an upper bound to the quadratic performance index (3) of the form

$$J(K(x(t)),x(t)) \leq V(x(t)) \tag{4}$$

can be found if $V(x)$ is a nonincreasing Lyapunov function. Moreover, $\mathcal{E} := \{x \in \mathbb{R}^n \mid\ V(x) \leq \gamma,\ \ \gamma \geq 0\}$, is a positive invariant region for the regulated input constrained system. In the presence of input constraints $u \in \mathcal{U}$, all of the above results continue to be true provided that the pair $(V(x),K(x))$ is chosen so that $\forall x \in \mathcal{E} \Rightarrow K(x) \subset \mathcal{U}$.

3 Main Result

In order to develop a constrained RHC strategy for the nonlinear system (1), the following state-dependent feedback law will be adopted hereafter

$$u(t) = K(x(t)),\ \forall t. \tag{5}$$

where $K(x)$ denotes a multivariate polynomial in the unkwnown x. Moreover it is supposed that the upper-bound $V(x(t))$ to the cost (3), introduced in (4), will be a SOS. Using this assumption and exploiting standard Hamilton-Jacobi-Bellman (HJB) inequality arguments [16], it is possible to derive conditions under which

there exists a closed-loop stabilizing control law (5) which achieves a guaranteed cost (4) and is compatible with the input constraints (2):

Proposition 1. *Let $x(t)$ be the current state of the polynomial nonlinear system (1) subject to (2). Then, there exists a state feedback control law of the form (5) ensuring asymptotical stability and constraints fulfilment from t onward if a SOS $V \in \Sigma[x]$, a polynomial $K \in R[x]$ and a scalar $\gamma \geq 0$ are found such that*

- $V(x) > 0 \; \forall x \in \mathcal{R}^n \setminus \{0\}$ *and* $V(0) = 0$;
- *the following Hamilton-Jacobi-Bellman inequality holds true*

$$\{x \in \mathcal{R}^n | V(x) \leq \gamma\} \subseteq \{x \in \mathcal{R}^n | V(x(t+1)) - V(x(t)) + x^T \Psi_x x + K(x)^T \Psi_u K(x) < 0\}$$

- *the saturation constraints are fulfilled*

$$\{x \in \mathcal{R}^n | V(x) \leq \gamma\} \subseteq \{x \in \mathcal{R}^n | K_i(x) \leq \bar{u}_i\}$$

$$\{x \in \mathcal{R}^n | V(x) \leq \gamma\} \subseteq \{x \in \mathcal{R}^n | K_i(x) \geq -\bar{u}_i\}$$

- *the initial state belongs to the positive invariant set $\mathcal{E}$ with margin* ε

$$\{x \in \mathcal{R}^n | (x - x(0))^T (x - x(0)) \leq \varepsilon\} \subseteq \{x \in \mathcal{R}^n | V(x) \leq \gamma\}, \; \varepsilon > 0$$

Proof. See [16]. □

By resorting to a "Positivstellensatz" (P-satz) [7] argument, the conditions stated in Proposition 1 can be recast as the ones of finding a SOS $V \in \Sigma[x]$, a polynomial $K \in R[x]$ and a scalar $\gamma \geq 0$ such that the following set, achieved as the intersection of each single region in the following list, is empty

$$\begin{cases} \{x \in \mathcal{R}^n | V(x) \leq 0, l_1 \neq 0\} \\ \{x \in \mathcal{R}^n | V(x) \leq \gamma, l_2 \neq 0, V(x(t+1)) - V(x(t)) + x^T \Psi_x x + K(x)^T \Psi_u K(x) \geq 0\} \\ \{x \in \mathcal{R}^n | K_i(x) > \bar{u}_i, V(x) \leq \gamma\} \\ \{x \in \mathcal{R}^n | K_i(x) < -\bar{u}_i, V(x) \leq \gamma\} \\ \{x \in \mathcal{R}^n | (x - x(0))^T (x - x(0)) \geq \varepsilon, V(x) > \gamma\} \end{cases} \tag{6}$$

where $l_1, \; l_2 \in R[x]$ are appropriate positive definite polynomials such that $l_1(0_x) = 0, l_2(0_x) = 0$. Finally, the following result can be stated

Proposition 2. *Let $x(0) \in \mathbb{R}^n$ be given. Then, a pair $(V(x), K(x))$ compatible with the conditions of Proposition 1 and minimizing the upper-bound (4) can be found by solving the following minimization problem hereafter named* **SOS-V-K**(x),

$$\min_{V, s_1, s_2, s_{3,i}, s_{4,i}, s_5 \in \Sigma[x], \; K \in \mathcal{R}[x], \; \gamma \geq 0} \gamma$$

subject to

$$V - l_1 \in \Sigma[x] \quad (7)$$

$$-((\gamma - V)s_1 + (V(f(x,K(x))) - V(x) + x^T \Psi_x x + K(x)^T \Psi_u K(x))s_2 + l_2) \in \Sigma[x] \quad (8)$$

$$(\bar{u}_i - K_i) - (\gamma - V)s_{3,i} \in \Sigma[x],\ i = 1,\ldots,m \quad (9)$$

$$(\bar{u}_i + K_i) - (\gamma - V)s_{4,i} \in \Sigma[x],\ i = 1,\ldots,m \quad (10)$$

$$-\varepsilon + (x - x(0))^T (x - x(0))s_5 + (\gamma - V) \in \Sigma[x] \quad (11)$$

under the following conditions on the degrees of the involved polynomials necessary for problem solvability

$$\begin{cases} \max\left(\partial\left(V s_1\right), \partial\left(V s_2\right)\right) \geq \\ \geq \max\left(\partial\left(V(f(x),K(x))s_2\right), \partial\left(x^T \Psi_x x s_2\right), \partial\left(K(x)^T \Psi_u K(x) s_2\right)\right), \\ \partial(V) = \partial(l_1),\ \partial\left(V s_{3,i}\right) \geq \partial(K_i),\ i = 1,\ldots,m,\ \ \partial(s_5) + 2 \geq \partial(V) \end{cases} \quad (12)$$

Proof. A detailed proof can be found in [17]. □

Remark 1. Note that the decision polynomials $s_1,\ s_2,\ s_{3,i},\ s_{4,i},\ s_5 \in \Sigma[x],\ \ i = 1,\ldots,m$ do not enter linearly in the constraints. It can be proved that the proposed optimization problem **SOS-V-K**(x) is a BMI and can be solved using a bisection procedure. Then, each bisection step involves three sequential linear SOS problems (see [17] for details), solvable by means of standard semidefinite programming packages [18]. Nonetheless, it is obvious that the obtained optimal solution is local and its behavior depends on the point from which the optimization starts. □

Remark 2. The numerical solution of the optimization $\mathbf{SOS-V-K}(x)$ procedure could be affected by different assumptions of the involved set of parameters, i.e. the degrees of optimization variables V and K and the degrees of all multipliers s_i deriving by the S-procedure application. A convenient choice of them (e.g. linearizations), satisfying the degree conditions (12), could improve the numerical efficiency of the overall scheme. In fact, in principle the more the degrees of $\{p, V, K, s_i\}$ increases the more possibly better numerical solutions result. □

4 A Low-Demanding Receding Horizon Control Algorithm

This section is devoted to show how the proposed procedure **SOS-V-K**(x) is capable to achieve satisfactory level of control performance within a Receding Horizon Control (RHC) framework. The idea is to resort to a computationally low demanding RHC scheme where most of the computations are carried out off-line. It is generally recognized that the evaluation of the on-line parts of traditional robust MPC schemes is computationally prohibitive in many practical situations. The problem is especially severe for nonlinear schemes and most of current research on MPC is in fact devoted to reduce such a high computational burden while still ensuring the same level of control performance of the traditional schemes (see [19], [15] and references therein). Here, the procedure **SOS-V-K**(x) will be exploited within the Robust RHC scheme proposed in [15] for the uncertain linear time invariant systems.

All the arguments developed in the previous sections allows one to write down a computable RHC scheme, hereafter denoted as **WK-SOS**, which consists of the following algorithm:

Algorithm-WK-SOS
Off-line

0.0 *Derive* V_0, *a Control Lyapunov Function (CLF) for the linearized system [9] around the desired set point and a control law* $K_0(x)$ *that asymptotically stabilizes (1).*
0.1 *Pick an initial feasible state* $x_1 \in \mathcal{E}_0 \triangleq \{x \in \mathbb{R}^n \mid V_0(x) \leq 1\}$, *generate a control law* $K_1(\cdot)$ *and an invariant region* $\mathcal{E}_1$, *by solving the SOS program* **SOS-V-K**(x_r). *Choose an integer* N *and put* $r = 2$. *Store* $K_1(\cdot),\ \mathcal{E}_1(\cdot)$ *in a lookup table;*
0.2 *Generate a control law* $K_r(\cdot)$ *and an invariant region* $\mathcal{E}_r$, *by solving the SOS program* **SOS-V-K**(x_r) *with the additional constraint* $\mathcal{E}_r \subset \mathcal{E}_{r-1}$ *translated as an extra SOS condition*

$$-((\alpha - V)\, s_{16} + (V_{k-1} - 1)) \in \Sigma[x] \text{ with } s_{16} \in \Sigma[x],\ \partial(V\, s_{16}) \geq \partial(V_{k-1}) \quad (13)$$

0.3 *Store* $K_r(\cdot),\ \mathcal{E}_r(\cdot)$;
0.4 *If* $r < N$, *choose a new state* x_{r+1} *s.t.* $x_{r+1} \in \mathcal{E}_r$, *Let* $r = r+1$ *and go to step* 0.2

On-line

1.1 *Given an initial feasible state* $x(0)$ *s.t.* $x(0) \in \mathcal{E}_1$, *put* $t = 0$;
1.2 *Given* $x(t)$, *perform a bisection search over the sets* $\mathcal{E}_r$ *in the look-up table to find* $\hat{r} := \arg\min_r$ *s.t.* $x(t) \in \mathcal{E}_r$
1.3 *Feed the plant by the input* $K_{\hat{r}}(x(t))$
1.4 $t \leftarrow t+1$ *and go to step* 1.2

Remark 3. Note that Step [0.2] of the proposed scheme involves a suboptimal search of a solution for **SOS-V-K**(x_r) meaning that a stopping criterion [17] is used, and in the worst case, for each value of the index $r = 2, \ldots, N$, the algorithm achieves the previously computed couple $K_{r-1}(\cdot),\ \mathcal{E}_{r-1}(\cdot)$. □

Next lemma ensures that the proposed MPC scheme admits a feasible solution at each time t and the SOS-based input strategy $K_{\hat{r}}(x(t))$ is a stabilizing control law for (1) under (2).

Lemma 1. *Given the system (1), let the off-line steps of proposed scheme have solution at time* $t = 0$. *Then, the on-line part of the* **WK-SOS** *algorithm has solution at each future time instant, satisfies the input constraints and yields an asymptotically stable closed-loop system.*

Proof. It follows by the existence of the sequences $K_r, \mathcal{E}_r$ which ensure that any initial state $x(0) \in \mathcal{E}_1$ can be steered to the origin without constraints violation. In particular, because of the additional constraint $\mathcal{E}_r \subseteq \mathcal{E}_{r-1}$, the regulated state trajectory emanating from the initial state satisfies

$$x(t+1) = \begin{cases} f(x(t)) + g(x(t))\, K_r(x(t)) & \text{if } x(t) \in \mathcal{E}_r,\ x(t) \notin \mathcal{E}_{r+1},\ r \neq N \\ f(x(t)) + g(x(t))\, K_N(x(t)) & \text{if } x(t) \in \mathcal{E}_N \end{cases} \quad (14)$$

Then, under both conditions $x(t) \in \mathcal{E}_r$ and $x(t) \notin \mathcal{E}_{r+1}$, $r = 1, \ldots, N-1$, the control law $K_r(\cdot)$ is guaranteed to ultimately drive the state from $\mathcal{E}_r$ into the ellipsoid $\mathcal{E}_{r+1}$ because the Lyapunov difference $V(f(x, \hat{K}_r(x)) - V(x)$ is strictly negative. Finally, the positive invariance of $\mathcal{E}_N$ and the contraction provided by K_N, guarantee that the state remains within $\mathcal{E}_N$ and converges to 0_x. □

5 Illustrative Example

The aim of this section is to give a measure of the improvements achievable by exploiting the SOS programming framework within a RHC framework. To this end, the **Algorithm-WK**, (see [15]) will be contrasted with the proposed **Algorithm-WK-SOS**. The simulations are instrumental to show especially the reduction of conservativeness in terms of achievable basins of attraction, when compared with its linear counterpart (viz. **Algorithm-WK**). All the computations have been carried out on a PC Pentium 4.

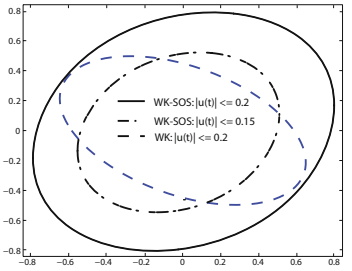
A controlled Van Der Pol nonlinear equation is taken into consideration [13]

$$\dot{x}_1(t) = x_2(t),\ \dot{x}_2(t) = -x_1(t) - (1 - x_1^2(t))x_2(t) + u(t) \tag{15}$$

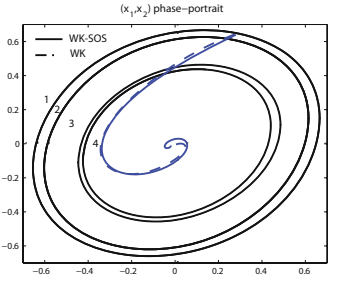
It has an unstable limit cycle around 0_x, which is a local asymptotically stable equilibrium point. The problem of computing inner approximations of the attraction basin with SOS machinery has been extensively studied (see [14] and references therein). The system (15) has been discretized by using forward Euler differences with a sampling time $T_c = 0.1$ sec. It has been further assumed: weighting matrices $\Psi_x = diag([0.01\ 0.01])$, and $\Psi_u = 1$ and input saturation constraint $|u(t)| \leq 0.2,\ \forall t$.

The design knobs are here summarized: Candidate Lyapunov function degree: $\partial(V(x)) = 6$; Candidate stabilizing controller degree: $\partial(K(x)) = 4$. The following degrees have been chosen $\partial(s_6) = 2$, $\partial(s_8) = 4$, $\partial(s_9) = 0$, $\partial(s_{11}) = 2$, $\partial(s_{12}) = 2$, $\partial(s_{14}) = 2$, $\partial(s_{16}) = 2$, for the free polynomials in the SOS formulation in order to satisfy the solvability conditions (12). Finally, the quantity ε in (32) has been chosen equal to 10^{-8}. Fig. (a) reports the basins of attraction for the two control schemes. As expected, **WK-SOS** (continuous line) enjoys an enlarged region of feasible initial states w.r.t. **WK** (dashed). Moreover, when one computes the basins of attraction under the more stringent saturation constraint $|u(t)| \leq 0.15$ it results that no solutions exist for the **WK** algorithm whereas a restricted region (dotted in Fig. (a)) is found for **WK-SOS** The results reported in next Figs (b)-(d) have been achieved with the input constraint $|u(t)| \leq 0.2$. Four pairs $(K_i, \mathcal{E}_i)$ have been determined with the **SOS-V-K(x)** algorithm initialized with a sequence of four states $x^{set} = \begin{bmatrix} 0.4 & 0.3 & 0.15 & 0.1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$. The initial state has been set to $x(0) = \begin{bmatrix} 0.25 & 0.68 \end{bmatrix}^T$.

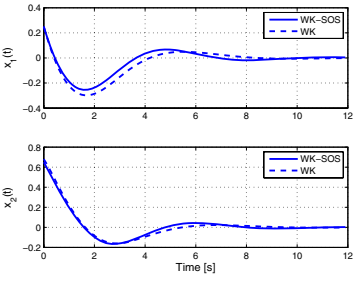
Finally, Figs. (b)-(d) depict respectively: the regulated phase portraits (four invariant regions $\mathcal{E}_i$ are graphically represented), the state regulated response and the control input for the two schemes.



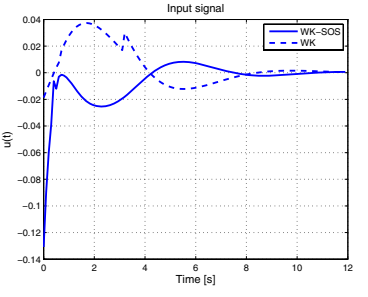
(a) State Attraction Region with input bound constraints - **WK-SOS**, $|u(t)| \leq 0.2$ (Continuous line), $|u(t)| \leq 0.15$ (Dash-dotted line); **WK**, $|u(t)| \leq 0.2$ (Dashed line)



(b) Phase portrait with input bound constraints $|u(t)| \leq 0.2$.



(c) State evolutions with input bound constraints $|u(t)| \leq 0.2$



(d) Input signal with input bound constraints $|u(t)| \leq 0.2$

6 Conclusions

In this paper, we have developed an off-line RHC algorithm for constrained polynomial nonlinear systems by means of SOS programming. The advantage of this algorithm is that it provides a set of stabilizing polynomial control laws, corresponding to a nested set of positive invariant regions. Up to our knowledge this is a first attempt in literature to formulate a RHC problem using SOS machinery. Numerical experiments have shown the benefits of the proposed RHC strategy w.r.t. linear embedding MPC schemes and makes SOS based MPC schemes potentially attractive.

References

1. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: stability and optimality. Automatica 36, 789–814 (2000)
2. Angeli, D., Casavola, A., Franzè, G., Mosca, E.: An Ellipsoidal Off-line MPC Scheme for Uncertain Polytopic Discrete-time Systems. Automatica 44(12), 3113–3119 (2009)

3. Allgöwer, F., Badgwell, T.A., Qin, J.S., Rawlings, J.B., Wright, S.J.: Nonlinear predictive control and moving horizon estimation – An introductory overview. In: Frank, P.M. (ed.) Advances in Control, Highlights of ECC 1999, pp. 391–449. Springer, Berlin (1999)
4. Magni, L., De Nicolao, G., Magnani, L., Scattolini, R.: A stabilizing model-based predictive control for nonlinear systems. Automatica 37, 1351–1362 (2001)
5. Björnberg, J., Diehl, M.: Approximate robust dynamic programming and robustly stable MPC. Automatica 42, 777–782 (2006)
6. Chesi, G.: Estimating the domain of attraction for uncertain polynomial systems. Automatica 40, 1981–1986 (2004)
7. Parrillo, P.: Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization. Ph.D. Thesis, California Institute of Technology (2000)
8. Scherer, C.W., Hol, C.W.J.: Matrix Sum-of-Squares Relaxations for Robust Semi-Definite Programs. Mathematical Programmming, Ser. B 107, 189–211 (2006)
9. Henrion, D., Peaucelle, D., Arzelier, D., Šebek, M.: Ellipsoidal Approximation of the Stability Domain of a Polynomial. IEEE Transactions on Automatic Control 48, 2255–2259 (2003)
10. Prajna, S., Parrilo, P.A., Rantzer, A.: Nonlinear control synthesis by convex optimization. IEEE Transactions on Automatic Control 49, 310–314 (2004)
11. Fotiou, I.A., Rostalski, P., Parrilo, P.A., Morari, M.: Parametric optimization and optimal control using algebraic geometry methods. International Journal of Control 79, 1340–1350 (2006)
12. Raff, T., Ebenbauer, C., Findeisen, R., Allgöwer, F.: Nonlinear Model Predictive Control and Sum of Squares Techniques. In: Fast Motions in Biomechanics and Robotics, vol. 340, pp. 325–344. Springer, Heidelberg (2006)
13. Tan, W., Packard, A.: Stability region Analysis using polynomila and composite polynomial Lyapunov functions and Sun-of-Squares programming. IEEE Transactions on Automatic Control 53, 565–571 (2008)
14. Topcu, U., Packard, A., Seiler, P.: Local stability analysis using simulations and sum-of-squares programming. Automatica 44, 2669–2675 (2008)
15. Wan, Z., Kothare, M.V.: An efficient off-line formulation of robust model predictive control using linear matrix inequalities. Automatica 39, 837–846 (2003)
16. Alamir, M., Bornard, G.: On the stability of receding horizon control of nonlinear discrete-time systems. System & Control Letters 23, 291–296 (1994)
17. Franzè, G., Casavola, A., Famularo, D., Garone, E.: An off-line MPC strategy for nonlinear systems based on SOS programming. Technical Report (2008), http://staff.icar.cnr.it/franze/tech_rep_08_11_01.pdf
18. Löfberg, J.: A Toolbox for Modeling and Optimization in MATLAB. Proceedings of the (CACSD) Conference. Taipei, Taiwan (2004), http://control.ee.ethz.ch/~joloef/yalmip.php
19. Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N.: The explicit linear quadratic regulator for constrained systems. Automatica 38, 3–20 (2002)

NMPC for Propofol Drug Dosing during Anesthesia Induction

S. Syafiie, J. Niño, C. Ionescu, and R. De Keyser

Abstract. This paper presents the application of nonlinear predictive control to drug dosing during anesthesia in prospective patients for undergoing surgery. A single-input (propofol) single output (Bispectral index (BIS)) patient model has been employed. The pharmacokinetic-pharmacodynamic model, which is in fact a Wiener-type model, has been used for prediction. A set of 12 patient models were studied while controlling BIS at 50 by applying our in-house nonlinear extended-prediction self-adaptive control strategy (NEPSAC). The results of this simulation study show that NEPSAC outperforms EPSAC, using a nominal patient model for prediction.

Keywords: Predictive Control, Nonlinear, Anesthesia.

1 Introduction

Anesthesia regarding from the aspect of drug dosing has a number of particular interests for control engineer to give contributions for designing automatic controllers. The prospective contributions are mainly: *a*) reduce the workload of the anesthetist; *b*) optimize the performance (minimize drug consumption and ensure the patient being in acceptable clinical state); *c*) avoid dangerous undershoot (during the induction phase); *d*) reduce variability for all recovery parameters.

General anesthesia of patients undergoing surgery is defined by a sequence of clinical procedures to induce a specific physiological state, and comprises three main components: Muscle relaxation, Analgesia and Hypnosis. Muscle relaxation (paralysis) can occur by interrupting function at several sites in the central nervous system. Muscle relaxation is measured by an index between

S. Syafiie, J. Niño, C. Ionescu, and R. De Keyser
Department of Electrical energy, Systems and Automation, Ghent University, Technologiepark 913, 9052 GENT, Belgium
e-mail: {syafiie,jorge,clara,rdk}@autoctrl.UGent.be

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 501–509.
springerlink.com

0 (full paralysis) and 100 (normal state), using available sensors (such as electromyographic response [1]). Analgesia (pain relief) is achieved through the administration of opioid drugs such as remifentanil. Hitherto, there is no direct sensor able to deliver an unique index of the analgesia level. Heart rate variability has been suggested to indicate the analgesia level [2], while research for delivering other indices is going on.

Several sensor systems including evoked auditory potential, spectral entropy of the electroencephalography (EEG) and bispectral index of the EEG (BIS) can be used to measure the hypnosis [3]. The BIS index is widely used in clinical practice and most adequate for controller design. The BIS index is ranging from 0 to 100 [4, 5], with zero denoting flat EEG line (i.e. isoelectric EEG) and 100 denoting a fully awake state and consciousness.

Most of the reported studies of anesthesia level regulation using anestetic propofol are based on using effect site concentration, while only few report using BIS as a controlled variable. For example, Lazy Learning [6] has been used for a simulated patient model to regulate propofol by controlling BIS. However, due to large inter-patient variability in the model parameters, a direct application of an agent based controller to clinical practice is difficult and requires apriori data for training.

During the last decades, target controlled infusion (TCI) introduced in 1968 by Kruger and Thiemer [7] has been succefully applied in clinical practice since the launch of comercial devices such as Disprifusor^{TM} in the year 1996 and Rugloop^{TM} in the year 1998. The basic theory of the TCI dosing is based on the pharmacokinetic (PK) model of the propofol hypnotic drug. Since TCI is essentially an open loop control strategy, the researchers argue that to correct the errors from desired BIS values, it is to mandatory to apply closed loop control strategy [5]. Most of the applications of advanced model-based controllers are based on pharmacokinetic-pharmacodynamic (PKPD) models and examples of using linear MPC in simulation studies are available [8, 9].

However, PKPD models contain linear dynamic and nonlinear static models, which are known as Wiener models. In this paper, PKPD-like Wiener model is used for simulation of 12 patient models during the induction phase using nonlinear MPC (NMPC). The paper is organized as follows: patient model is given in section 2. The nonlinear extended prediction self-adaptive control (NEPSAC) algorithm is briefly discussed in section 3 and the simulation results are discussed in section 4., while final remarks are given in the conclusion section.

2 Patient Model

The effect in a patient from the infused anesthetic can be represented by PKPD models. Such models may be used to predict the future output

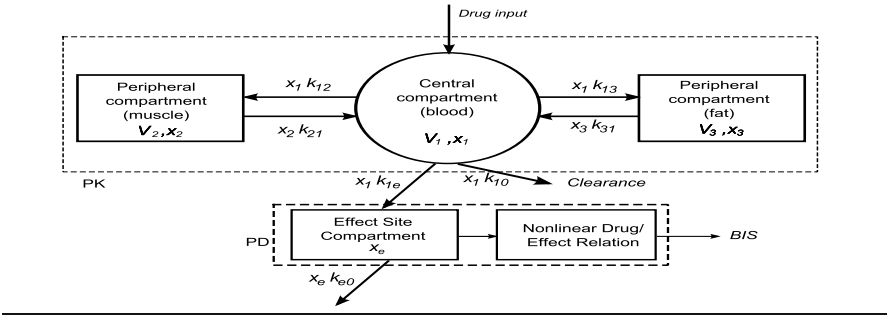


Fig. 1 Propofol distribution of a 3 compartment patient model

and calculate an optimal control action by using a MPC strategy, such as NEPSAC.

In this study, a 3-compartment PKPD patient model (figure 1) is employed [10]:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -k_{10} - k_{12} - k_{13} & k_{21} & k_{31} \\ k_{12} & -k_{21} & 0 \\ k_{13} & 0 & -k_{31} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u \tag{1}$$

where x_1 denotes the amount of drug in the central compartment (blood) and x_2 and x_3 represent the amount of the drug in compartments 2 and 3 respectively; while u denotes the infused propofol in the central compartment. The constant k_{ij} represents the transfer rate of the drug from i^{th} compartment to j^{th} compartment, while k_{10} is the rate of the drug metabolism. The constants k_{ij} and k_{10} of the PK model (1) depend on the weight, height, age, gender and lean body mass of the patient, as well as on the 3-compartment volumes (blood V_1, muscle V_2, and fat V_3).

The pharmacodynamics (PD) are characterized by a first-order relation to the central compartment concentration C_p (represented by x_1) in blood. The apparent concentration in the effect compartment can be calculated, since k_{e0} will precisely characterize the temporal effects of equilibration between the plasma concentration and the corresponding drug effect. Consequently, the equation is often used as in:

$$\dot{C}_e = k_{e0}(C_p - C_e) \tag{2}$$

where C_e is called the *effect-site compartment concentration*. This concentration is strongly related to the level of unconsciousness of a patient, consequently used to define the depth of anesthesia (hypnosis). The BIS variable can be related to the drug effect concentration C_e by the empirical static nonlinear relationship, called also the *Sigmoid Hill Model* (SHM):

$$E = E_0 - E_{max} \frac{C_e^{\gamma}}{C_e^{\gamma} + C_{50}^{\gamma}} \tag{3}$$

where E is BIS measure, E_0 denotes the baseline value (awake state - without propofol), which typically set to 100; E_{max} denotes the maximum effect achieved by the drug infusion; C_{50} is the drug concentration at half maximal effect and represents the patient's sensitivity to the drug; and γ determines the steepness of the static nonlinearity in (3). The dynamics of effect site concentration and plasma concentration are calculated by using the linear relation (1) and (2). The effect site concentration is then introduced with (3), which denote a static nonlinear model. Therefore, the overall PKPD model is a Wiener model, which typically consists of a dynamic linear model and a static nonlinear equation.

3 The NEPSAC Approach to MPC

The Extended Prediction Self Adaptive Controller (EPSAC) [11] is our in-house developed MPC algorithm. The extended algorithm for nonlinear case (NEPSAC) has been previously explained and employed in industrial applications (e.g. [12]). The optimal control signals are calculated based on: i) the prediction model output $y(t+k)$ at time t, indicated by $y(t+k|t), k = 1, \ldots, N_2$ over the prediction horizon N_2; ii) measurements until time t: $\{y(t), y(t-1), \ldots, u(t-1), u(t-2), \ldots\}$ and iii) future inputs: $\{u(t|t), u(t+1|t), \ldots\}$ (postulated at time t). The core of the EPSAC strategy is that the future response is the result of two effects:

$$y(t+k|t) = y_{\text{base}}(t+k|t) + y_{\text{optimize}}(t+k|t), \tag{4}$$

where $y_{\text{base}}(t+k|t)$ is effects of past control signal, a base for future output scenario and future (predicted) disturbance. $y_{\text{optimize}}(t+k|t)$ is the effect of the optimizing future control actions. The component $y_{\text{optimize}}(t+k|t)$ is the cumulative effect of a series of impulse inputs and a step input that is:

$$\begin{aligned} y_{\text{optimize}}(t+k|t) = h_k \delta u(t|t) + h_{k-1} \delta u(t+1|t) + \ldots \\ + g_{k-Nu+1} \delta u(t+N_u-1|t), \end{aligned} \tag{5}$$

where $g_1, g_2, \ldots, g_k$ are the coefficients of the unit step response of the system; $h_1, h_2, \ldots, h_k$ are the coefficients of the unit impulse response of the system; and δu is the optimizing future control actions. From (4) and (5), the EPSAC-MPC equation in matrix form can be written as following by taking $\mathbf{G}$ that contains $\mathbf{h}$ and $\mathbf{g}$:

$$\mathbf{Y} = \bar{\mathbf{Y}} + \mathbf{G}\mathbf{U}. \tag{6}$$

where $\bar{\mathbf{Y}}$ is the vector of $y_{\text{base}}(t+k|t)$. The optimal control signal is calculated by minimizing a quadratic cost function. However, the concept of the EPSAC strategy (4) is only valid for linear systems. For nonlinear systems, by appropriate selection of the *base* control $u_{\text{base}}(t+k|t)$, the second term in the right hand side of (4) can be gradually made small enough compared

to $y_{\text{base}}(t+k|t)$. Therefore, $y_{\text{optimize}}(t+k|t)$ is small if $\delta u(t+k|t)$ is small. To have $\delta u(t+k|t)$ small, it is necessary that $u_{\text{base}}(t+k|t)$ is *close* to the optimal $u^*(t+k|t)$ and let $\mathbf{R}$ be a reference vector. In NEPSAC, an iterative procedure is employed, involving no linearisation, which consists of 2 major steps:

- Step 1:
  - measure the output of the process, i.e. $y(t) = y(t|t)$
  - select the vector of control $\mathbf{U}_{\text{base}} = [u_{\text{base}}(t|t), u_{\text{base}}(t+1|t), \ldots, u_{\text{base}}(t+N_2-1|t)]$ defined the previous control action.
  - calculate the vector $\bar{\mathbf{Y}}$ using $\mathbf{U}_{\text{base}}$, via the process model and using the measured outputs as initial conditions,
  - calculate the matrix $\mathbf{G}$ by entering a step or impulse in the prediction model with the measured outputs as initial conditions
- Step 2:
  - calculate the optimal $\mathbf{U}^* = [\mathbf{G}^T\mathbf{G}]^{-1}[\mathbf{G}^{\mathbf{T}}(\mathbf{R} - \bar{\mathbf{Y}})]$
  - if $|\mathbf{U}^*| \leq \epsilon$, apply $u(t) = u_{\text{base}}(t|t) + \delta u(t|t)$ to the plant and go to step 1 in the next sample.
  - else, set $\mathbf{U}_{\text{base}} = \mathbf{U}_{\text{base}} + \mathbf{U}^*$, calculate the new $\mathbf{Y}_{\text{base}}$ based on $\mathbf{U}_{\text{base}}$ and return to the beginning of step 2.

For further detail on the NEPSAC strategy, see [11].

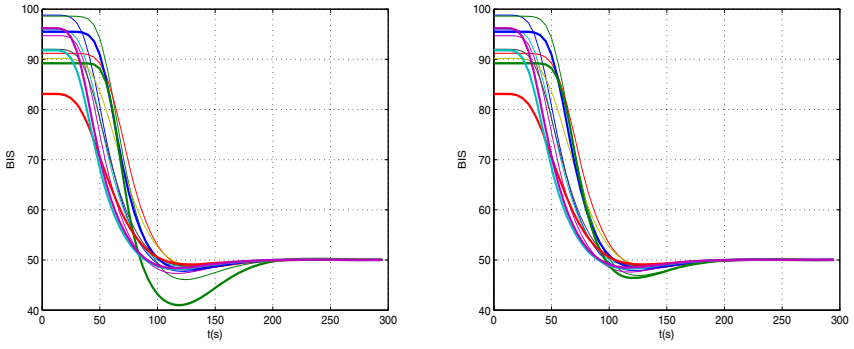
4 Simulation Results during Induction Phase

For the NEPSAC study in this paper we employed the patient PKPD model parameters from past clinical studies [8]. For this simulation study, a sample period of 5 seconds [8] was used, with a control horizon of N_u=1, and a prediction horizon N_2 which corresponds to 50 seconds (10 samples) starting after the estimated system delay of 10 seconds (given $N_1 = 2$ samples).

4.1 Ideal Case: No Modelling Errors

The ideal case is defined as the scenario in which the patient model is perfectly known and both EPSAC and NEPSAC approaches will be compared. The (N)EPSAC controller is applied for regulating BIS during the induction phase as in Figure 2, with the controller effort as shown in Figure 3 (3.3 mg/s is a upper bound limited syringe pump). As expected, the NEPSAC controller outperforms the EPSAC controller in terms of undershoot.

The inter-patient variability as denoted by 3 is depicted in Figure 4-a, while the corresponding first derivative is given in Figure 4-b. Although there are no modelling errors, in NEPSAC the nonlinear equation that relates BIS to the effect-site concentration C_e (Equation 3) is used for prediction, while its first derivative is used to calculate the matrix $\mathbf{G}$ at each iteration. This



(a) BIS response EPSAC case (b) BIS response NEPSAC case

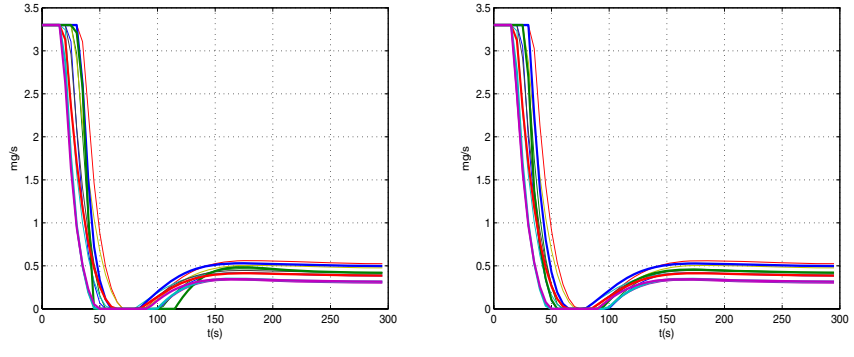
Fig. 2 BIS-response comparison in the ideal case of no modelling errors

means that the calculation of the control action takes into account the patient sensitivity.

As observed, the static nonlinear models have the lowest values of $|dBIS/dCE|$ (see Figure 4 b) for the zones with low hypnotic effect and the biggest ones for the region near to the reference (50), which result in fast responses of NEPSAC strategy.

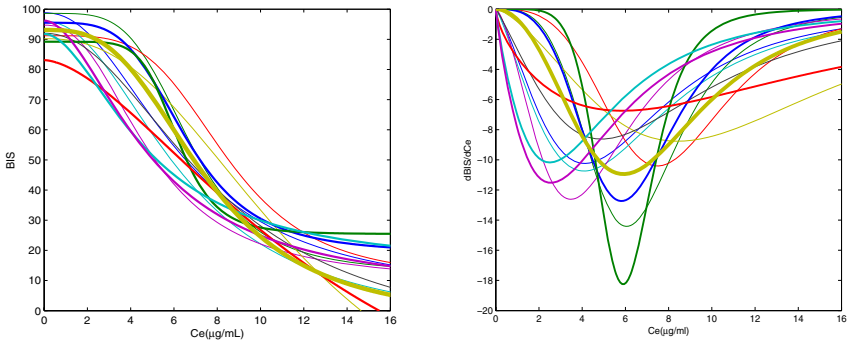
4.2 Real Case: Modelling Errors

The patient model from (1) can be adapted with the patient biometric data, while (3) can never be a priori known. Consequently, an average population model for (3) can be used for prediction, as calculated from the 12 patient models (denoted by the bolded line in Figure 4-a). This situation is closer



(a) Propofol input EPSAC case (b) Propofol input NEPSAC case

Fig. 3 Propofol input profile in the ideal case of no modelling errors

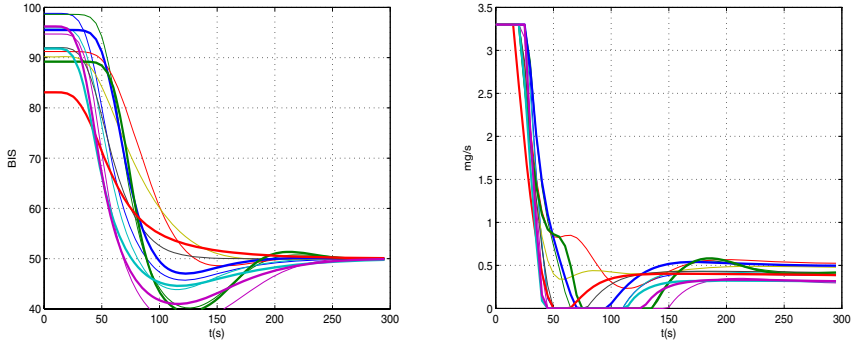


(a) BIS/C_e relation (b) $dBIS/dC_e$ sensitivity

Fig. 4 BIS-Ce characteristic of the study population

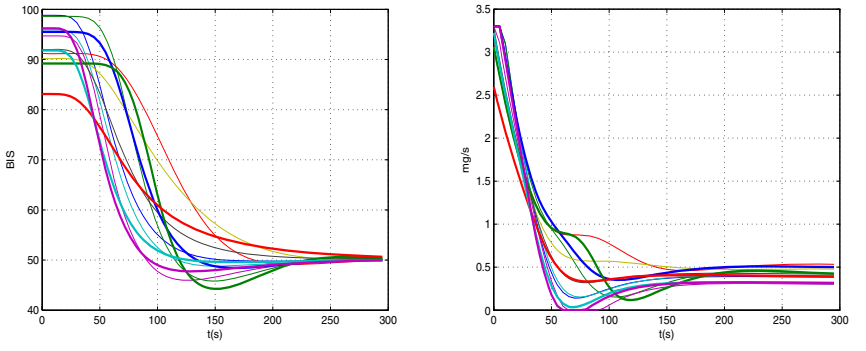
to the clinical practice, when the relation BIS/C_e is totally unknown before staring the induction phase for anesthesia. The simulation results for NEPSAC strategies are depicted in Figure 5. The nominal model produces errors in prediction and consequently in the computation of the control signal.

Initially, both controllers (linear and nonlinear EPSAC) have similar performance due to the fact that the upper bound is reached for the manipulated variable. In this case, both algorithms drive the system to the same sub-optimal control action. As a consequence, both input-output profiles are calculated based on the same history, thus leading to the same performance result. The reason why NEPSAC is not able to iterate to the optimal solution is that constraints are not taken into account in the solution of the cost function $\mathbf{U}^*$.



(a) BIS response NEPSAC case (b) Propofol input NEPSAC case

Fig. 5 BIS response in a more realistic scenario



(a) BIS responses (b) Propofol input

Fig. 6 NEPSAC results for increased prediction horizon N_2=16

If one takes a look at the initial phase in Figure 5, one may realize that the controller's reaction during these first moments is crucial. Since the controller's reaction is based on the BIS/C_e relation (3) and its derivative (gain)(see Figure 4), it is mandatory to implement a feasible prediction sequence. Due to the fact that the input to the patient is constrained, the optimal solution is clipped, leading to sub-optimal results. Increasing the prediction horizon allows the controller to take into account the changes in gain occurring after these crucial first moments. Figure 6 shows the results for the case where the prediction horizon is increased to N_2=16.

As observed, an overall improvement is obtained if N_2 is increased, by avoiding the saturation of the input during the first part of the induction phase for most patients. Compared to the ideal case in Figure 2-b, the effect of the increased prediction horizon is visible; an increased settling time (from 150 seconds to 200 seconds) and the patient BIS responses are not homogeneous due to the high inter-patient variability and modelling errors. The patients with the highest sensitivity (Figure 4-b), i.e. highest gain, present higher values of undershoot. On the other hand, the patient with highest drug resistance (e.g. red bolded line in the same figure) have the slowest response.

5 Conclusion

The objective of this paper was to present a simulation study of NEPSAC-MPC applied to anesthesia during the induction phase for patients undergoing surgery. The patient models of the study containing a linear dynamic part (pharmacokinetics) and a static nonlinearity (pharmacodynamics) were considered as Wiener-type models. Both linear and nonlinear versions of EPSAC were applied. Simulation results show that NEPSAC controller outperformes EPSAC if no modelling error is present, and has similar results with average model for prediction.

In order to implement the NEPSAC algorithm in clinical trials, the intra- and inter-patient variability problem is ongoing research. Performance improvement can be achieved if the prediction model is adapted to the specific patient under study.

References

1. Brown, B.H., Asbury, J., Linkens, D.A., Perks, R., Anthony, M.: Closed-loop control of muscle relaxation during surgery. Clinical Physics and Physiological Measurement 1(3), 203–210 (1980)
2. Nunes, C.S., Mendonca, T., Bras, S., Ferreira, D.A., Amorin, P.: Modeling Anesthetic Drugs' Pharmacodynamic Interaction on the Bispectral Index of the EEG: the Influence of Heart Rate. In: Proceeding of the 29th Annual International Conference of IEEE EMBS, Lyon, France, August 23-26 (2007)
3. Vanluchene, A.L., Vereecke, H., Thas, O., Mortier, E.P., Shafer, S.L., Struys, M.: Spectral entropy as an electroencephalographic measure of anesthetic drug effect: a comparison with bispectral index and processed midlatency auditory evoked response. Anesthesiology 101(1), 34–42 (2004)
4. Bailey, J.M., Haddad, W.M.: Drug dosing control in clinical pharmacology. IEEE Control Systems Magazine 25, 35–51 (2005)
5. Struys, M.R.F., De Smet, T., Mortier, E.P.: Closed-loop control of anaesthesia. Current Opinion in Anesthesiology 15, 421–425 (2002)
6. Caelen, O., Bontempi, G., Coussaert, E., Barvais, L., Clement, F.: Machine Learning Technique to Enable Closed-loop Control in Anesthesia. In: Proceeding of the 19th IEEE symposium on Computer Based Medical Systems (2006)
7. Kruger-Thiemer: Continuous intravenous infusion and multicomparment accumulation. European Journal of Pharmacology 4, 317–324 (1968)
8. Ionescu, C., De Keyser, R., Torrico, B.C., De Smet, T., Struys, M., Normey-Rico, J.E.: Robust Predictive Control Strategy Applied for Propofol Dosing using BIS as a Controlled Variable during Anesthesia. IEEE Transactions on Biomedical Engineering 55(9), 2161–2170 (2008)
9. Sawaguchi, Y., Furutani, E., Shirakami, G., Araki, M., Fukuda, K.: A Model-Predictive Hypnosis Control System Under Total Intravenous Anesthesia. IEEE Transaction on Biomedical Engineering 55(3), 874–887 (2008)
10. Schnider, T.W., Minto, C.F., Gambus, P.L., Andresen, C., Goodale, D.B., Youngs, E.J.: The influence of method of administration and covariates on the pharmacokinetics of propofol in adult volunteers. Anesthesiology 88, 1170–1182 (1998)
11. De Keyser, R.M.C.: Model Based Predictive Control. In: Invited Chapter in UNESCO Encyclopedia of Life Support Systems (EoLSS), vol. 83. Eolss Publishers Co Ltd., Oxford (2003) (invited chapter) article 6.43.16.1, ISBN 0 9542 989 18-26-34
12. De Keyser, R., Donald III, J.: Application of the NEPSAC Nonlinear Predictive Control Strategy to a Semiconductor Reactor. In: Findeisen, R. (ed.) Assessment and Future Directions of Nonlinear Model Predictive Control. LNCIS, vol. 385, pp. 503–512. Springer, Heidelberg (2007)

Spacecraft Rate Damping with Predictive Control Using Magnetic Actuators Only

Christoph Böhm, Moritz Merk, Walter Fichter, and Frank Allgöwer

Abstract. A nonlinear model predictive control (NMPC) approach for rate damping control of a low Earth orbit satellite in the initial acquisition phase is proposed. The only available actuators are magnetic coils which impose control torques on the satellite in interaction with the Earth's magnetic field. In the initial acquisition phase large rotations and high angular rates, and therefore strong nonlinearities must be dealt with. The proposed NMPC method, which is shown to guarantee closed-loop stability, efficiently reduces the kinetic energy of the satellite while satisfying the constraints on the magnetic actuators. Furthermore, due to the prediction of future trajectories, the negative effect of the well-known controllability restriction in magnetic spacecraft control is minimized. It is shown via a simulation example that the obtained closed-loop performance is improved when compared to a classical P-controller.

Keywords: Magnetic spacecraft control; Rate damping; Nonlinear model predictive control.

1 Introduction

Magnetic coils are one opportunity to control the attitude and angular rate dynamics of low Earth orbit satellites. Their high reliability and durability and further advantageous features led to a high interest in magnetic spacecraft control in the literature in the past.

Christoph Böhm, Moritz Merk, and Frank Allgöwer
Institute for Systems Theory and Automatic Control,
University of Stuttgart, Germany
e-mail: {cboehm,merk,allgower}@ist.uni-stuttgart.de

Walter Fichter
Institute for Flight Mechanics and Control,
University of Stuttgart, Germany
e-mail: fichter@ifr.uni-stuttgart.de

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 511–520.
springerlink.com

Controlling the attitude of a satellite, which is necessary e.g. in pointing scenarios, is one important task in spacecraft control that can be solved with approaches using magnetic coils. Most often those control methods are based on a linear description of the satellite dynamics. In [8] an overview of existing methods for the attitude control problem is given. Additional to the well-known PD-controller, see [8] and references therein, there exist approaches exploiting the periodic behavior of the Earth's magnetic field [7, 12] and several predictive control based methods [5, 8, 15]. Further approaches are discussed in [6], [12] and [13].

Another important scenario in spacecraft control is the rate damping problem in the initial acquisition phase of the satellite. Here, in contrast to many attitude control problems, large rotations and angular velocities, and therefore strong nonlinearities, occur. The most famous approach in rate damping control is the P-controller, see e.g. [6, 8]. Further methods are discussed in [10, 14]. To the author's best knowledge no results exist using NMPC to tackle the rate damping problem.

In the frame of this paper we propose a nonlinear model predictive controller for the rate damping problem of a circular low Earth orbit satellite. Since the considered system is highly nonlinear and subject to hard input constraints, NMPC is a suitable control method for this kind of problem. It is shown that, due to the inherent optimization of a cost functional, NMPC provides a noticeable performance improvement when compared to the classical P-controller [8]. Furthermore, the prediction of future trajectories allows to minimize the negative effect of the well-known controllability restriction in magnetic spacecraft control [8, 12]. Note that in this paper we do not focus on computational issues, although we are aware of the restrictions on available computation time in spacecraft control. We are rather interested, as a first step, in improving the rate damping performance with NMPC approaches in simulations. Further research is necessary to investigate the applicability for real space missions.

The paper is organized as follows. In Section 2 a brief review on NMPC is given. Section 3 introduces the dynamics of the considered spacecraft. In Section 4 the main result is proposed, namely a stabilizing nonlinear model predictive control approach for rate damping using magnetic actuators. Simulation results of the proposed controller with a comparison to the classical P-controller are presented in Section 5. A brief summary concludes the paper in Section 6.

2 Nonlinear Model Predictive Control

We consider nonlinear systems of the form

$$\dot{x} = f(x, u), \;\; x(0) = x_0, \tag{1}$$

with $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$. The system might be subject to state and input constraints of the form $u(t) \in \mathcal{U} \;\forall\; t \geq 0$ and $x(t) \in \mathcal{X} \;\forall\; t \geq 0$. Here $\mathcal{X} \subseteq \mathbb{R}^n$

is the state constraint set and $\mathcal{U} \subset \mathbb{R}^m$ is the set of feasible inputs. In NMPC the open-loop optimal control problem

$$\min_{\bar{u}(\cdot)} J_c\big(\bar{x}(\cdot), \bar{u}(\cdot)\big), \tag{2a}$$

subject to

$$\dot{\bar{x}}(\tau) = f\big(\bar{x}(\tau), \bar{u}(\tau)\big),\ \bar{x}(t_k) = x(t_k), \tag{2b}$$

$$\bar{x}(\tau) \in \mathcal{X},\ \bar{u}(\tau) \in \mathcal{U},\ \forall \tau \in \big[t_k, t_k + T_p\big], \tag{2c}$$

$$\bar{x}(t_k + T_p) \in \mathcal{E}, \tag{2d}$$

is solved repeatedly at each sampling instant t_k with the cost functional

$$J_c\big(\bar{x}(\cdot), \bar{u}(\cdot)\big) = \int_{t_k}^{t_k+T_p} F(\bar{x}, \bar{u})\ \mathrm{d}\tau + E\big(\bar{x}(t_k + T_p)\big), \tag{3}$$

where $F(\bar{x}, \bar{u}) > 0$ and $E\big(\bar{x}(t_k+T_p)\big) > 0$ and with the prediction horizon T_p. The solution to the optimization problem leads to

$$\bar{u}^\star\big(t; x(t_k)\big) = arg \min_{\bar{u}(\cdot)} J\big(\bar{x}(\cdot), \bar{u}(\cdot)\big). \tag{4}$$

The control input applied to system (1) is updated at each sampling instant t_k by the repeated solution of the open-loop optimal control problem (2), i.e. the applied control input is given by

$$u(t) = \bar{u}^\star(t; x(t_k)),\ t \in \big[t_k, t_k + \delta\big), \tag{5}$$

where δ is the sampling time between each optimization (assumed to be fixed). Since the solution to the optimization problem at each time instant t_k depends on the current system state $x(t_k)$, state feedback is provided. If certain well-known conditions on the terminal penalty term E and the terminal region $\mathcal{E}$ are satisfied, the presented NMPC approach guarantees stability of the closed-loop system, see e.g. [1, 2, 3].

3 Spacecraft Dynamics

The control task considered in this paper is to stabilize the rate dynamics of a rigid spacecraft in the initial acquisition phase where large angular velocities and rotations occur. Therefore, motion and attitude of the spacecraft have to be described by nonlinear differential equations, for which it is necessary to consider two coordinate frames. The body frame has its origin in the satellite's center of gravity and its axes are given by the spacecraft geometry [12]. The inertial coordinate frame has its origin in the center of the Earth and its axes are not moving with time. According to [8, 11, 12] the dynamics of the angular rates of a rigid spacecraft can be described by Euler's equations

$$J\dot{\omega} = -\omega \times J\omega + \tau. \tag{6}$$

Here $\omega = [\omega_x\ \omega_y\ \omega_z]^T \in \mathbb{R}^3$ represents the angular rate of the spacecraft expressed in body frame. $J \in \mathbb{R}^{3\times3}$ is the inertia matrix assumed to be diagonal and $\tau \in \mathbb{R}^3$ is the vector of magnetic control torques.

There exist several ways to describe the spacecraft attitude kinematics [8, 11, 12]. In the frame of this paper we use the parameterization given by the four quaternions (also called Euler parameters), which leads to

$$\dot{q} = \frac{1}{2} W(\omega) q, \tag{7}$$

where $q \in \mathbb{R}^4$ is the unit norm vector of quaternions and where

$$W(\omega) = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix}. \tag{8}$$

To impose external control torques on a low Earth orbit satellite one has in principle three possibilities: momentum wheels, thrusters and electromagnetic coils. Magnetic coils use solar energy and thus, energy consumption is of minor importance when using them as control inputs. Furthermore, coils have a high reliability and durability. Motivated by these advantageous properties, in this paper we consider spacecrafts which only possess magnetic actuators. In interaction with the Earth's magnetic field, the three magnetic coils, which are aligned with the spacecraft principal axes, generate torques according to

$$\tau = m \times b = B(b)m. \tag{9}$$

Here $m = [m_1 \; m_2 \; m_3]^T \in \mathbb{R}^3$ represents the control input variables, namely the vector of magnetic dipoles for the three coils, which are constrained by

$$|m_i| \leq m_{max}, \;\; i = 1, 2, 3. \tag{10}$$

The vector $b = [b_x \; b_y \; b_z]^T \in \mathbb{R}^3$ describes the Earth's magnetic field in body frame and delivers the cross product matrix

$$B(b) = \begin{bmatrix} 0 & b_z & -b_y \\ -b_z & 0 & b_x \\ b_y & -b_x & 0 \end{bmatrix}. \tag{11}$$

The Earth's magnetic field in body frame is obtained via the coordinate transformation $b = \Omega(q) b_I$, in which b_I is the Earth's magnetic field in inertial coordinates and with the direction cosine matrix $\Omega(q)$ defined as

$$\Omega(q) = \begin{bmatrix} 2(q_1^2 + q_4^2) - 1 & 2(q_1 q_2 + q_3 q_4) & 2(q_1 q_3 - q_2 q_4) \\ 2(q_1 q_2 - q_3 q_4) & 2(q_2^2 + q_4^2) - 1 & 2(q_2 q_3 + q_1 q_4) \\ 2(q_1 q_3 + q_2 q_4) & 2(q_2 q_3 - q_1 q_4) & 2(q_3^2 + q_4^2) - 1 \end{bmatrix}. \tag{12}$$

To calculate the time-varying vector b_I we use the approximation of the Earth's magnetic field presented in [11]. Due to space limitations we do not discuss this approximation in this paper.

Clearly, in (11) the matrix B is always of rank two. Therefore, it is not possible to apply independent control torques to all three satellite axes. The reason for this is that the mechanical torque, generated by the

interaction of the Earth's magnetic field with the magnetic field induced by the coils, is always perpendicular to the Earth's magnetic field. Loosely speaking, at each sampling instant the satellite can only be steered in two directions by the controller. This is a serious and well-known limitation in magnetic spacecraft control, although the directions in which torques can be imposed change when the spacecraft moves in orbit [8, 12]. In contrast to classical feedback controllers $u = k(x)$, where the control action at each sampling instant only depends on the current system state at this sampling instant, in NMPC the controller predicts the future behavior of the system states. Therefore, it can be expected that the negative effect of the controllability restriction can be reduced by applying NMPC to control the satellite. In the following section we propose an NMPC scheme for rate damping in the initial acquisition phase of the satellite. In this phase large rotations and angular velocities, and therefore strong nonlinearities, must be dealt with. Considering the nonlinear dynamics (6),(7) with multiple inputs which are subject to the constraints (10), NMPC is a suitable control method to tackle the rate damping control problem.

4 Rate Damping with NMPC

The control task is to withdraw the kinetic energy of the spacecraft using magnetic coils as actuators. It is desirable to achieve the control task as fast as possible, since the satellite in the initial acquisition phase runs on batteries. Therefore, consider the cost functional with the kinetic energy $E_{kin} = \frac{1}{2}\omega^T J\omega$

$$J_c\big(\bar{\omega}(\cdot), \bar{m}(\cdot)\big) = \int_{t_k}^{t_k+T_p} g^T(\bar{\omega}, \bar{b}) I_\alpha g(\bar{\omega}, \bar{b}) + \bar{m}^T R\bar{m} \, \mathrm{d}\tau + E_{kin}(t_k + T_p), \quad (13)$$

which is minimized over the prediction horizon T_p. Here $J \in \mathbb{R}^3$ is the inertia matrix and $R \in \mathbb{R}^3$ is a positive definite diagonal matrix penalizing the inputs. The matrix $I_\alpha = \alpha I$ defined as the identity matrix $I \in \mathbb{R}^3$ multiplied with a positive constant $\alpha > 0$ penalizes ω in the cost functional via the term $g(\bar{\omega}, \bar{b}) = [g_1 \;\; g_2 \;\; g_3]$ with $g_i = \frac{2m_{max}}{\pi} \arctan\big(K_i(\bar{b} \times \bar{\omega})_i\big)$, $i = 1, 2, 3$. The choice of this particular term is motivated by the fact that a terminal controller, similar to the "b-dot" control law [8], can be found to guarantee stability. Some conditions on the constants K_i and α and on the weighting matrix R will be given in Theorem 1. The open-loop optimal control problem based on the cost functional (13) that is solved repeatedly at the sampling instants t_k is formulated as

$$\min_{\bar{m}(\cdot)} J_c\big(\bar{\omega}(\cdot), \bar{m}(\cdot)\big), \quad (14a)$$

subject to

$$\dot{\bar{\omega}} = J^{-1}(-\bar{\omega} \times J\bar{\omega}) + J^{-1}B(\bar{b})\bar{m},\ \bar{\omega}(t_k) = \omega(t_k), \tag{14b}$$
$$\dot{\bar{q}} = W(\bar{\omega}),\ \bar{q}(t_k) = q(t_k), \tag{14c}$$
$$\dot{\bar{b}} = \Omega(\bar{q})\bar{b}_I, \tag{14d}$$
$$|\bar{m}_i| \leq m_{max}. \tag{14e}$$

Choosing the matrix R and the constant α such that the term E_{kin} dominates in the cost functional results in those trajectories leading to minimal kinetic energy at the end of the prediction horizon, which is desirable. However, it is necessary that R and α appear in the cost functional to guarantee closed-loop stability. Furthermore, the following assumption is required to proof stability.

Assumption 1. *If* $(b \times \omega) = 0$ *holds, then* $\frac{d}{dt}(b \times \omega) \neq 0$.

Remark 1. *Assumption 1 assures that if the vector of the magnetic field in body coordinates is parallel to the rate vector, then the motion of the satellite is such that this situation only holds for an infinitesimal short time. This assumption is not of practical relevance, however it is required in the proof of Theorem 1.*

The NMPC controller defined in the following theorem based on the open-loop optimal control problem (14), guarantees closed-loop rate dynamics stability.

Theorem 1. *The nonlinear model predictive controller*

$$m(t) = \bar{m}^{\star}\big(t; \omega(t_k), q(t_k)\big),\ t \in [t_k, t_{k+1}) \tag{15}$$

where

$$\bar{m}^{\star}\big(t; \omega(t_k), q(t_k)\big) = arg\min_{\bar{m}(\cdot)} J_c\big(\bar{\omega}(\cdot), \bar{m}(\cdot)\big) \tag{16}$$

is the optimal solution to the open-loop optimal control problem (14) which is solved repeatedly at the sampling instants t_k *based on the corresponding system states* $\omega(t_k)$ *and* $q(t_k)$*, asymptotically stabilizes the rate dynamics (6) of the considered spacecraft, if the condition* $0 < K_i < \frac{\pi}{2(R_{ii}+\alpha)m_{max}}$ *is satisfied and if the optimization problem (14) is initially feasible.*

Proof. The proof is based on the proof provided in [2]. Since we do not consider a terminal constraint on the state ω in the NMPC setup, the condition for closed-loop stability is that there exists an input m which satisfies the constraints (10) such that

$$\int_t^{t+\epsilon} \frac{1}{2}\frac{\partial(\omega^T J\omega)}{\partial\omega}\dot{\omega} + g^T(\omega, b)I_\alpha g(\omega, b) + m^T Rm\ \mathrm{d}\tau < 0 \tag{17}$$

holds for all $t \geq 0$, all $\epsilon > 0$ and all $\omega \in \mathbb{R}^3$. If the term under the integral $\frac{1}{2}\frac{\partial(\omega^T J\omega)}{\partial\omega}\dot{\omega} + g^T(\omega, b)I_\alpha g(\omega, b) + m^T Rm$ can be shown to be negative in the whole integration interval except at countable many time instants where it is zero, then clearly (17) is satisfied. With the dynamics (6) the obtained condition is

$$\omega^T(m \times b) + g^T(\omega, b) I_\alpha g(\omega, b) + m^T R m < 0, \tag{18}$$

which is identical to

$$(b \times \omega)^T m + g^T(\omega, b) I_\alpha g(\omega, b) + m^T R m < 0. \tag{19}$$

This inequality is clearly satisfied if for each component of the vectors $(b \times \omega)$ and m and for the diagonal entries of the matrices I_α and R

$$(b \times \omega)_i^T m_i + g_i^T(\omega, b) \alpha g_i(\omega, b) + m_i^T R_{ii} m_i < 0, \;\; i = 1, 2, 3 \tag{20}$$

holds. The control input

$$m_i = -\frac{2m_{max}}{\pi} \arctan\left(K_i (b \times \omega)_i\right) \tag{21}$$

with $K_i > 0$, $i = 1, 2, 3$, satisfies the constraint $|m_i| \leq m_{max}$ for all $\omega \in \mathbb{R}^3$ and all $b \in \mathbb{R}^3$. Plugging (21) into inequality (20) one obtains

$$\frac{2(R_{ii} + \alpha) m_{max}}{\pi} \arctan\left(K_i (b \times \omega)_i^T\right) \arctan\left(K_i (b \times \omega)_i\right) - (b \times \omega)_i^T \arctan\left(K_i (b \times \omega)_i\right) < 0. \tag{22}$$

In the case of $(b \times \omega)_i > 0$ this is equivalent to

$$\frac{\pi}{2(R_{ii} + \alpha) m_{max}} (b \times \omega)_i^T > \arctan\left(K_i (b \times \omega)_i\right). \tag{23}$$

Using the monotonicity of the arctan one can show that (23) holds if

$$0 < K_i < \frac{\pi}{2(R_{ii} + \alpha) m_{max}}. \tag{24}$$

For the case $(b \times \omega)_i < 0$ we obtain the same condition on K_i. Summarizing, if K_i in the control law (21) satisfies (24), $i = 1, 2, 3$, then inequality (18) holds for all ω satisfying $(b \times \omega) \neq 0$. However, as follows from (22), the expression $\frac{1}{2} \frac{\partial \omega^T J \omega}{\partial \omega} \dot{\omega} + g^T(\omega, b) I_\alpha g(\omega, b) + m^T R m$ is zero for $(b \times \omega) = 0$ when the controller (21) is applied, i.e. condition (18) is violated for all $(b \times \omega) = 0$. Following from Assumption 1, this violation only occurs at countable many time instants and therefore the integral condition (17) is satisfied for all t, ω and $\epsilon > 0$. According to [2], this guarantees closed-loop stability of the NMPC scheme defined in Theorem 1 for all $\alpha > 0$. ■

Remark 2. *Since the proposed NMPC controller guarantees stability without a terminal constraint, the prediction horizon can be chosen arbitrarily small. Although large prediction horizons are recommendable to obtain good controller performance, small horizons are interesting from a computational point of view.*

5 Simulation Results

In this section we provide preliminary simulation results of the NMPC approach introduced in Section 4. We point out that further investigations are necessary to finally rate the presented controller. However, without providing an extensive simulation study, several conclusions can be drawn. The

simulation has been carried out with the NMPC environment *OptCon* [9], which uses a large-scale nonlinear programming solver (HQP, [4]). We used a prediction horizon $T_p = 8\ min$, the sampling rate was $\delta = 4.8\ s$, the inertia matrix was chosen to $J = diag(211, 2730, 2650)$ and the input constraint was $m_{max} = 400\ Am^2$. The weightings R and α were chosen such that they only had vanishing influence on the overall cost functional, i.e. mainly the kinetic energy at the end of the prediction horizon was minimized. Choosing a cost functional penalizing the kinetic energy in the integral term over the whole prediction horizon would certainly lead to a larger kinetic energy at the end of the horizon. Furthermore, non-trivial modifications in the stability proof would be required. As shown in Figure 1, the NMPC controller withdraws the kinetic energy significantly faster than the standard P-controller. Figures 3 and 4 show the corresponding angular rates of the NMPC controlled satellite and the magnetic dipole m_1, respectively. In Figure 2 one of the advantageous properties of the NMPC approach is illustrated. After 117.8 s the NMPC controller first increases the kinetic energy before withdrawing it after reaching a peak much faster than the P-controller (with the same initial condition at 117.8 s) does. As can be seen from the weak energy decrease obtained by the P-controller, withdrawing the energy is hardly possible at the beginning of the considered time interval. This is caused by the controllability restriction of the satellite. Based on the prediction of future trajectories the NMPC controller first increases the energy in order to obtain an attitude of the satellite which allows a faster withdrawing of the energy afterwards. A similar behavior can be observed at further time instants. However, due to the relative small peaks this effect is not visible in Figure 1. Some obviously numerical problems occur after 150 s. Here, the kinetic energy increases slightly, and especially the angular rate ω_x increases significantly. This is caused by too aggressive control actions in the corresponding time interval. Probably this follows from penalizing the control actions not strong enough. Thus, further investigations are necessary to analyze and overcome this problem. To obtain a satisfying solution to the optimization problem via numerical solvers, a

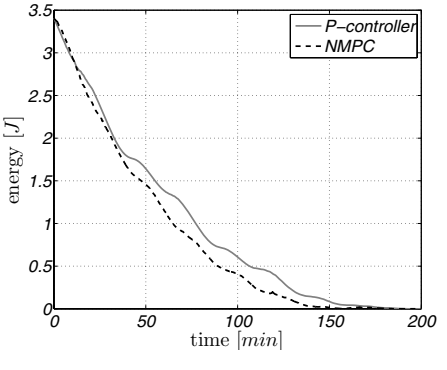


Fig. 1 Kinetic energy

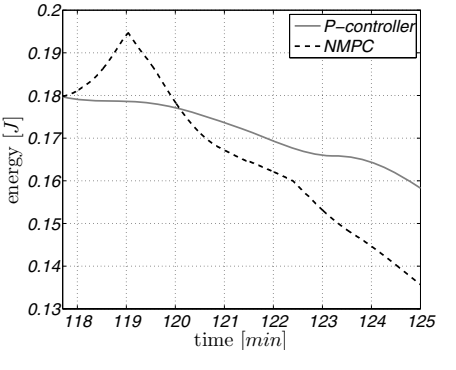


Fig. 2 Energy in short time interval

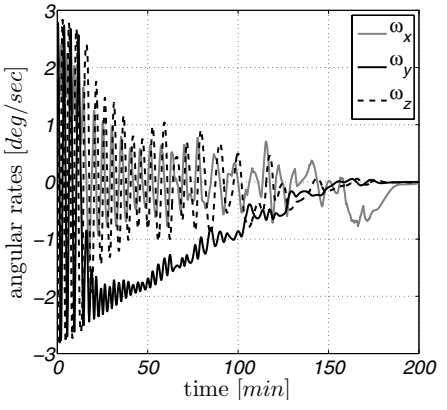


Fig. 3 Angular rates

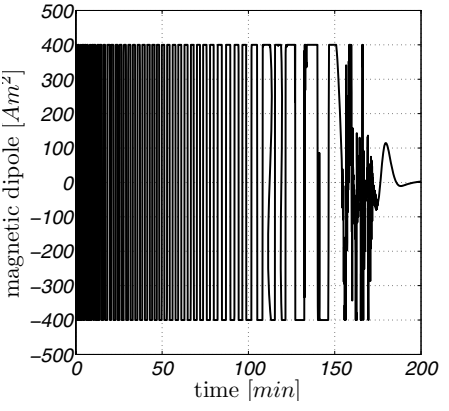


Fig. 4 Magnetic dipole m_1

discretization of the input m is provided. The discretization step length (here $\delta = 4.8s$) has to be short to obtain the shattering behavior in Figure 4 (which also occurs with standard controllers). Thus, large prediction horizons lead to large optimization problems. One has to find a suitable tradeoff between a satisfying controller performance and relatively low computational demand.

Summarizing, the simulation results provided are promising and it can be expected that the performance of existent controllers for the rate damping problem can be improved by the novel NMPC approach, although further research is necessary to finally rate the proposed controller.

6 Conclusions

We presented an NMPC approach for the spacecraft rate damping problem in the initial acquisition phase using only magnetic actuators. The controller satisfies the given input constraints and has been shown to guarantee closed-loop stability. First simulation results show that the controller reduces the negative effect of the controllability restriction in magnetic spacecraft control and improves the performance of a standard P-controller.

References

1. Camacho, E.F., Bordons, C.: Nonlinear model predictive control: An introductory review. In: Findeisen, R., Allgöwer, F., Biegler, L.T. (eds.) Assessment and Future Directions of Nonlinear Model Predictive Control, pp. 1–16. Springer, Heidelberg (2007)
2. Findeisen, R., Imsland, L., Allgöwer, F., Foss, B.: State and output feedback nonlinear model predictive control: An overview. European Journal of Control 9, 190–206 (2003)

3. Fontes, F.A.: A general framework to design stabilizing nonlinear model predictive controllers. System and Control Letters 42(2), 127–142 (2000)
4. Franke, R., Arnold, E., Linke, H.: HQP: A solver for nonlinearly constrained large-scale optimization, http://hqp.sourceforge.net
5. Hegrenæs, Ø., Gravdahl, J.T., Tøndel, P.: Spacecraft attitude control using explicit model predictive control. Automatica 41(12), 2107–2114 (2005)
6. Lovera, M., Astolfi, A.: Global magnetic attitude control of spacecraft in the presence of gravity gradient. IEEE Transactions on Aerospace and Electronic Systems 42(3), 796–895 (2006)
7. Psiaki, M.L.: Magnetic torquer attitude control via asymptotic periodic linear quadratic regulation. Journal of Guidance, Control, and Dynamics 24(2), 386–394 (2001)
8. Silani, E., Lovera, M.: Magnetic spacecraft attitude control: a survey and some new results. Control Engineering Practice 13(3), 357–371 (2005)
9. Simon, L.L., Nagy, Z.K., Hungerbuehler, K.: Swelling constrained control of an industrial batch reactor using a dedicated NMPC environment: OptCon. In: Proceedings of the International Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control, Pavia, Italy (2008)
10. Wang, P., Shtessel, Y.: Satellite attitude control via magnetorquers using switching control laws. In: Proceedings of the 14th IFAC world congress, Beijing, China (1999)
11. Wertz, J.R.: Spacecraft Attitude Determination and Control. Kluwer Academic Pub., Dordrecht (1978)
12. Wisniewski, R.: Satellite Attitude Control Using Only Electromagnetic Actuation. PhD thesis, Department of Control Engineering, Aalborg University (1996)
13. Wisniewski, R.: Linear time-varying approach to satellite attitude control using only electromagnetic actuation. Journal of Guidance, Control, and Dynamics 23(4), 640–647 (2000)
14. Wisniewski, R., Blanke, M.: Fully magnetic attitude control for spacecraft subject to gravity gradient. Automatica 35(7), 1201–1214 (1999)
15. Wood, M., Chen, W.-H.: Regulation of magnetically actuated satellites using model predictive control with disturbance modelling. In: IEEE International Conference on Networking, Sensing and Control, ICNSC, Sanya, China, pp. 692–697 (2008)

Nonlinear Model Predictive Control of a Water Distribution Canal Pool

J.M. Igreja and J.M. Lemos

Abstract. Water distribution canals provide interesting examples of distributed parameter plants for which nonlinear model predictive control may be applied. Canals are formed by a sequence of pools separated by gates. Output variables are the pool level at certain points, manipulated variables are the position of the gates and disturbances are the outlet water flows for agricultural use. The operation of this system is subject to a number of constraints. These are the minimum and maximum positions of the gates, gate slew-rate and the minimum and maximum water level. The objective considered in this paper is to drive the canal level to track a reference in the presence of the disturbances. The pool level is a function of both time and space that satisfies the Saint-Venant equations. These are a set of hyperbolic partial differential equations that embody mass and momentum conservation. In order to develop a NMPC algorithm, the Saint-Venant equations are approximated by a set of ordinary differential equations corresponding to the variables at the so called collocation points. Together with the boundary conditions, this forms a nonlinear reduced predictive model. In this way, a nonlinear prediction of future canal levels is obtained. The paper details the control general formulation along with a computationally efficient algorithm as well as the results obtained from its application.

Keywords: Saint-Venant equations, open-channel hydraulic systems, level nonlinear preditive control.

J.M. Igreja
INESC-ID and ISEL, Rua Alves Redol, 9, 1000-029 Lisboa, Portugal
e-mail: jmigreja@deq.isel.ipl.pt

J.M. Lemos
INESC-ID/IST, Rua Alves Redol, 9, 1000-029 Lisboa, Portugal
e-mail: jlml@inesc-id.pt

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 521–529.
springerlink.com

1 Introduction

This work proposes an approach to the multivariable control of nonlinear hyperbolic systems based on nonlinear receding horizon control (RHC) also known as Nonlinear Model Predictive Control (NMPC) [1]. It has important advantages for advanced process control. This is due to the fact that it allows an easy incorporation of constraints, that is a decisive advantage for industrial applications when compared to other methods. However, from a theoretical standpoint, the major issue consists in ensuring stability for a finite horizon.

Predictive control of hyperbolic PDE systems, namely transport-reaction processes, were studied in [2] and [3] for SISO cases. In the former the controller is based in a predictive model developed using the method of characteristics and does not consider constraints. In the latter finite differences for space discretization and a space distributed actuator were used with success. In [4] and [5] adaptive predictive control was obtained via Orthogonal Collocation (OC) reduced modelling, also for SISO hyperbolic tubular transport-reaction processes, that demonstrated to achieved the control objectives. Stability conditions are also derived and included.

In this paper a NMPC formulation for multivariable hyperbolic systems is considered along with a computationally efficient control algorithm using the same techniques. In particular the algorithm is applied to a water distribution canal pool prototype. The pool level is a function of both time and space that satisfies the Saint-Venant equations. These are a set of hyperbolic partial differential equations that embody mass and momentum conservation. In order to develop the control algorithm, the Saint-Venant equations are approximated by a set of difference equations corresponding to the variables at the collocation points. Together with the boundary conditions, this forms a nonlinear set of differential equations. In this way, a nonlinear prediction of future canal levels is obtained. The paper details the algorithm as well as the results obtained from its application.

A wide bibliography on water distribution open-canals is available. In [6] a selection of the controller structure is combined with robust design methods in order to achieve a compromise between water resources management and disturbance rejection. Predictive control is considered in [7] and in [8] with adaptation.

The rest of the paper is organized as follows. Sections 2 and 3 respectively introduces the canal pool dynamical model and the model reduction approach. Section 4 is dedicated to a general formulation of multivariable NMPC for distributed hyperbolic dynamical systems along with the control algorithm. The study case is developed in section 5. Section 6 draws conclusions.

2 Water Distribution Canal Pool Prototype

Saint-Venant equations for a single pool model without infiltration are given by:

$$\frac{\partial h}{\partial t} + \frac{v}{L}\frac{\partial h}{\partial x} + \left(\frac{da}{dh}\right)^{-1}\frac{a(h)}{L}\frac{\partial v}{\partial x} = 0$$
$$\frac{\partial v}{\partial t} + \frac{v}{L}\frac{\partial h}{\partial x} + \frac{g}{L}\frac{\partial v}{\partial x} + g(\mathcal{I}(h,v) - \mathcal{J}) = 0 \quad (1)$$

where $h(x,t)$ and $v(x,t)$ are respectively level and water velocity distributions along space ($x \in [0.1]$) and time, wet surface $a(h)$ and friction $\mathcal{I}(h,v)$ are nonlinear functions, g is the gravity acceleration, J is the constant canal slope and L is the pool length. Boundary conditions are given by flow at upstream and downstream gates:

$$v(0,t) = c_d A_d(u)\sqrt{(2g(H_u(t) - h(0,t)))}/a(h(0,t)) \quad (2)$$

$$v(L,t) = c_d A_d(u)\sqrt{(2g(h(L,t) - H_d(t)))}/a(h(L,t)) \quad (3)$$

In this paper a single trapezoidal reach with two pools, two moving gates (upstream ends) and a fixed gate at the downstream end is considered. The water elevation immediately before and after the reach are assumed known, $H_u(t)$ and $H_d(t)$ respectively.

3 Orthogonal Collocation

In order to design the controller, the distributed model (1) is first approximated by a lumped parameter model via the OC method [9]. The most common approximations are based in finite differences or in OCM. The latter method has the advantage of drastically reducing the number of ODEs needed to yield a solution for a given level approximation. As a drawback, it implies the choice of several parameters, such as the number of collocation points or the characteristic parameter (α, β) of Jacobi polynomials. Note that this choice can be critical and time consuming for finding good results before an adequate selection to the problem at hand is made [10]. For this sake, it is assumed that the level and water velocity along the pool are approximated by the weighted sum:

$$h(x,t) \cong \sum_{i=0}^{N+1} \varphi_i(x)\, h_i(t) \qquad v(x,t) \cong \sum_{i=0}^{N+1} \varphi_i(x)\, v_i(t) \quad (4)$$

where the functions $\varphi_i(z)$ are Lagrange interpolation polynomials, orthogonal at the so called interior collocation points z_i for i=1,..., N and at the boundary collocation points z_0 and z_{N+1}:

$$\varphi_i(x_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad (5)$$

Inserting (4) into (1) results in the ordinary differential equation verified by the time weights $h_i(t)$ and $v_i(t)$:

$$\sum_{i=0}^{N+1} \varphi_i(x)\, \frac{dh_i(t)}{dt} + \frac{v}{L} \sum_{i=0}^{N+1} \frac{d\varphi_i(x)}{dx}\, h_i(t) + \left(\frac{da}{dh}\right)^{-1} \frac{a(h)}{L} \sum_{i=0}^{N+1} \frac{d\varphi_i(x)}{dx}\, v_i(t) = 0$$

$$\sum_{i=0}^{N+1} \varphi_i(x)\, \frac{dv_i(t)}{dt} + \frac{v}{L} \sum_{i=0}^{N+1} \frac{d\varphi_i(x)}{dx}\, v_i(t) + \frac{g}{L} \sum_{i=0}^{N+1} \frac{d\varphi_i(x)}{dx}\, v_i(t) + g(\mathcal{I}(h,v) - \mathcal{J}) = 0 \tag{6}$$

Compute now (6) at each of the collocation points $z = z_j$, it follows that:

$$\frac{dh_j(t)}{dt} = -\frac{v_j}{L} \sum_{i=0}^{N+1} \frac{d\varphi_i(x_j)}{dx}\, h_i(t) - \left(\frac{da}{dh}\right)^{-1} |_{z=z_j}\, a(h_j) \sum_{i=0}^{N+1} \frac{d\varphi_i(x_j)}{dx}\, v_i(t)$$

$$\frac{dv_j(t)}{dt} = -\frac{v_j}{L} \sum_{i=0}^{N+1} \frac{d\varphi_i(x_j)}{dx}\, v_i(t) - \frac{g}{L} \sum_{i=0}^{N+1} \frac{d\varphi_i(x_j)}{dx}\, v_i(t) - g(\mathcal{I}(h_j,v_j) - \mathcal{J}) \tag{7}$$

The PDE (1) is therefore approximated by $n = N + 1$ ordinary ODEs, reading in matrix form:

$$\begin{aligned} \dot{h} &= -\frac{1}{L}\, (V_e A_h h + F(h)\, A_h v_e) \\ \dot{v} &= -\frac{1}{L}\, (g A_v h + V A_v v_e) - g I(h,v) \end{aligned} \tag{8}$$

where $h = \left[\, h_0 \; h_1 \cdots h_{N+1} \,\right]^T$ and $v = \left[\, v_1 \cdots v_N \,\right]^T$ are the states at collocation points, $v_e = \left[\, v_0 \; v^T \; v_{N+1} \,\right]^T$, A_h and A_v matrices are given by:

$$A_h = \begin{bmatrix} \varphi_1'(z_0) & \varphi_2'(z_0) & \cdots & \varphi_{N+1}'(z_0) \\ \varphi_1'(z_1) & \varphi_2'(z_1) & \cdots & \varphi_{N+1}'(z_1) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1'(z_{N+1}) & \varphi_2'(z_{N+1}) & \cdots & \varphi_{N+1}'(z_{N+1}) \end{bmatrix} \tag{9}$$

$$A_v = \begin{bmatrix} \varphi_1'(z_1) & \varphi_2'(z_1) & \cdots & \varphi_{N+1}'(z_1) \\ \varphi_1'(z_2) & \varphi_2'(z_2) & \cdots & \varphi_{N+1}'(z_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1'(z_N) & \varphi_2'(z_N) & \cdots & \varphi_{N+1}'(z_N) \end{bmatrix} \tag{10}$$

($\varphi_j'(z_i) \equiv \frac{d\varphi_j(z)}{dz}\, |_{z=z_i}$), $V = diag(v_1, ..., v_N)$, $V_h = diag(v_0, ..., v_{N+1})$, $I(h,v)$ is column vector ($I_j = (\mathcal{I}(h_j, v_j) - \mathcal{J})$) and $v_0(t)$ and $v_{N+1}(t)$ are boundary conditions obtained by (4).

Water open-channel dynamics reduced modelling based on the same approach for the Saint-Venant equations can also be found in [11]. OCM is the best systematic available method for model reduction after the Galerkin's method not suitable for hyperbolic equations [2].

4 NMPC for Hyperbolic Systems

Consider the following general formulation of multivariable receding horizon control for distributed hyperbolic dynamical systems. The aim is to control the output $y(t)$, a state nonlinear function weighted in the space domain, by manipulating the input $u(t)$, solving an open loop optimization problem and applying a receding horizon strategy according to the NMPC approach. Therefore, define the cost functional:

$$\min_u J = \int_t^{t+T} \left(l(\tilde{y}(\tau)) + u^T(\tau) \Gamma \tilde{u}(\tau) \right) d\tau \tag{11}$$

where $l(\tilde{y}) \geq 0$ is a penalty function and $\Gamma > 0$, subject to the model and input, output, state operational constraints:

$$\mathcal{C}(\dot{\eta}(t), \dot{y}(t), \dot{u}(t), \eta(t), y(t), u(t), t) \leq 0 \tag{12}$$

and stability constraints:

$$\mathcal{S}(\eta(t), u(t), t) \leq 0 \tag{13}$$

with:

$$y(t) = \int_0^1 a(z) h(x(z,t)) \, dz \tag{14}$$

$$\tilde{y} = y_r - y \quad \tilde{u} = u_r - u \tag{15}$$

$$\eta_k(t) = \int_0^1 b(z) x_k(z,t) \, dz \tag{16}$$

and:

$$\int_0^1 a(z) \, dz = \int_0^1 b(z) \, dz = 1 \tag{17}$$

where y_r and u_r define the reference trajectory to track.

One computationally efficient procedure for solving the enunciated nonlinear, infinite dimension, programming problem is to use a finite parametrization for the control signal $u(t) \in [t, t+T[$, where N_u segments of constant value $u_1, \ldots, u_{N_u}$ and duration $\frac{T}{N_u}$ are considered. Thus, the nonlinear, finite dimension, programming problem amounts to solve:

$$\min_{u_1, \ldots, u_{N_u}} J = \int_t^{t+T} \left(l(\tilde{y}(\tau)) + u^T(\tau) \Gamma \tilde{u}(\tau) \right) d\tau \tag{18}$$

subject to the model and to:

$$\mathcal{C}(\dot{\eta}(\bar{t}), \dot{y}(\bar{t}), \dot{u}(\bar{t}), \eta(\bar{t}), y(\bar{t}), u(\bar{t}), \bar{t}) \leq 0 \tag{19}$$

$$\mathcal{S}(\eta(\bar{t}), u(\bar{t}), \bar{t}) \leq 0 \tag{20}$$

$$u(\bar{t}) = seq\{u_1, \ldots, u_{N_u}\} \tag{21}$$

where, $u(\bar{t})$ is a sequence of steps of amplitude u_i $(i = 1, \ldots, N_u)$ and duration $\frac{T}{N_u}$. The variable $\bar{t}$ represents time during the minimization process $\bar{t} \in [0, T[$. Once the minimization result $u(\bar{t})$ obtained, the first sample u_1 is applied at $t + \delta$ and the whole procedure is repeated. The interval δ corresponds to the time needed to obtain a solution, being assumed that δ is much smaller than the sampling interval.

5 Results

Consider now the application of the above techniques to the specific case of the prototype canal (1) and the controller is based in the reduced model (8) with $N = 3$, $\alpha = 0.5$ and $\beta = 0.5$. Assuming that the state is accessible at the collocation points, the NMPC optimizer is define by:

$$\min_{u_1 \ldots u_3} J = \int_t^{t+T} \left(\tilde{y}^T \tilde{y} + 2\ u^T \tilde{u}\right) d\tau \tag{22}$$

subject to (8) and:

$$0 \leq u(\bar{t}) \leq 0.5$$

$$u(\bar{t}) = seq\{u_1, \ldots, u_3\} \tag{23}$$

As mentioned before the manipulated input are the gates openings and the outputs are the water elevation at half pool length: $y = [y_1(t, L/2)\ y_2(t, L/2)]^T$, using a $5s$ sampling period, N_u=3 and $T = 15s$ for the horizon.

Table 1 Pool physical parameters

Parameter		Value	Units
Gravitational constant	g	9.8	ms^{-1}
Manning coefficient	n	1.0	$m^{-1}s^{-3}$
Discharge coefficient	c_d	0.6	–
Discharge area	$A_d(u)$	0.49 u	m^2
Bottom width	b	0.15	m
Trapezoid slope	d	0.15	–
Canal slope	$\mathcal{J}$	2×10^{-3}	–
Upstream elevation	H_u	2.0	m
Downstream elevation	H_d	1.0	m

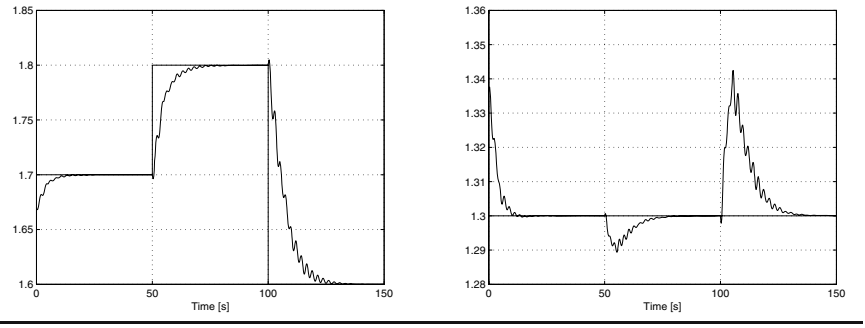


Fig. 1 Water elevation tracking the setpoint at $L/2$, first pool: $y_1(t)$ $[m]$ (left). Water elevation regulation around setpoint at $L/2$, second pool: $y_2(t)$ $[m]$ (right)

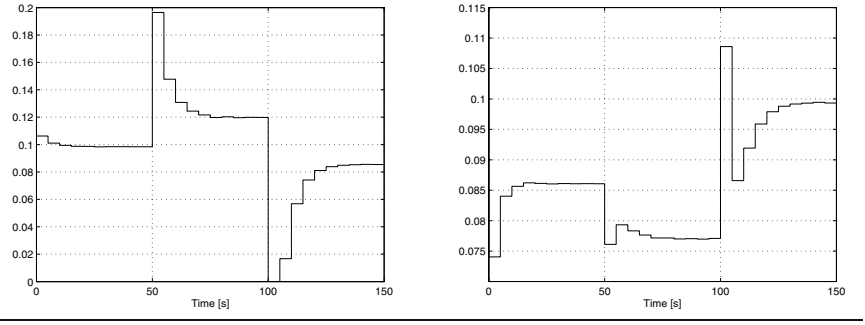


Fig. 2 Gate opening upstream, first pool: $u_1(t)$ $[m]$ (left). Gate opening upstream, second pool: $u_2(t)$ $[m]$ (right)

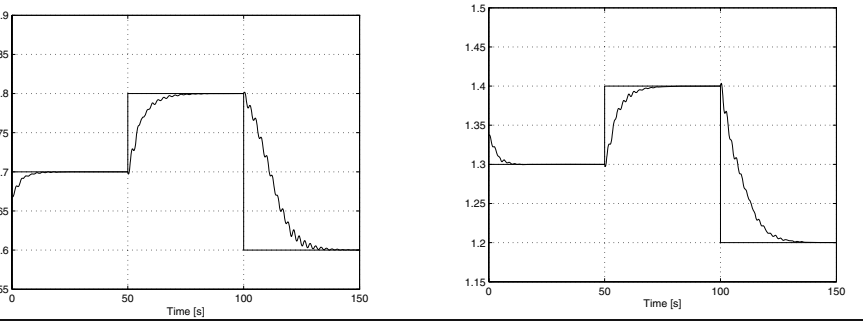


Fig. 3 Water elevation setpoint tracking at $L/2$, first pool: $y_1(t)$ $[m]$ (left). Water elevation setpoint tracking at $L/2$ second pool: $y_2(t)$ $[m]$ (right)

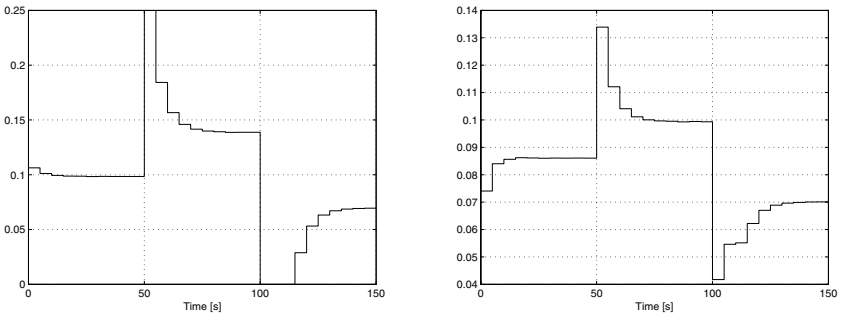


Fig. 4 Gate opening upstream, first pool: $u_1(t)$ $[m]$(left). Gate opening upstream, second pool: $u_2(t)$ $[m]$ (right)

Table (1) shows the pool parameters and equations (24) and (25) the wet surface for trapezoidal section and friction equation, respectively:

$$a(h) = bh + dh^2 \tag{24}$$

$$\mathcal{I}(h, v) = \frac{v|v|n^2}{R(h)^{(4/3)}} \qquad R(h) = \frac{a(h)}{b + 2\sqrt{1 + d^2}} \tag{25}$$

Figures 1 and 2 shows results for y_1 setpoint tracking and y_2 regulation around setpoint. Figures 3 and 4 shows results for y_1 and y_2 both for setpoint tracking. Note that the constraints are active after $t = 100s$ for a short period.

6 Conclusions

A multivariable NMPC for a set of hyperbolic partial differential equations that embody mass and momentum conservation has been developed. A general formulation along with a computational efficient algorithm is presented. Model reduction is obtained via the Orthogonal Collocation Method. Two numerical examples based in a water distribution canal pool prototype illustrates the application with success.

References

1. Findeisen, R., Allgower, F.: An Introduction to Nonlinear Model Predictive Control. In: 21st Benelux Meeting on Systems and Control, Veidhoven (2002)
2. Dubljevic, S., Mhaskar, P., El-Farra, N.H., Cristofides, P.: Predictive control of transport-reaction processes. Computer and Chemical Engineering 29, 2335–2345 (2005)
3. Shang, H., Forbes, J.F., Guay, M.: Model preditive control for quasilinear hyperbolic distributed parameters systems. Industrial and Enginnering Chemical Research 43, 2140–2149 (2004)

4. Igreja, J.M., Lemos, J.M., Silva, R.N.: Adaptive Receding Horizon Control of Tubular Reactors. In: Proceedings of 44th IEEE CDC-ECC (2005)
5. Igreja, J.M., Lemos, J.M., Silva, R.N.: Adaptive Receding Horizon Control of a Distributed Collector Solar Field. In: Proceedings of 44th IEEE CDC-ECC (2005)
6. Litrico, X., Fromion, V.: H_∞ Control of an Irrigation canal Pool With a Mixed Control Politics. IEEE Trans. Control Syst. Tech. 14(1), 99–111 (2006)
7. Gomez, M., Rodellar, J., Mantecon, J.: Predictive control method for decentralised operation of irrigation canals. Applied Mathematical Modelling 26, 1039–1056 (2002)
8. Lemos, J.M., Rato, L.M., Machado, F., Nogueira, N., Salgueiro, P., Silva, R.N., Rijo, M.: Predictive adaptive control of water level in canal pools. In: Proc. 16th Int. Conf. Systems Science, Wroclaw, Poland, pp. 139–148 (2007)
9. Rice, G.R., Do, D.D.: Applied Mathematics and Modeling for Chemical Engineers. John Wiley & Sons, Chichester (1995)
10. Dochain, D., Babary, J.P., Tali-Maamar: Modeling and adaptive control of nonlinear distributed parameter bioreactors via orthogonal collocation. Automatica 28, 873–883 (1992)
11. Dulhoste, J.F., Georges, D., Besançon, G.: Nonlinear control of open-channel water flow based on a collocation control model. ASCE Journal of Hydraulic Eng. 130, 254–266 (2004)

Swelling Constrained Control of an Industrial Batch Reactor Using a Dedicated NMPC Environment: *OptCon*

Levente L. Simon, Zoltan K. Nagy, and Konrad Hungerbuehler

Abstract. This work presents the application of nonlinear model predictive control (NMPC) to a simulated industrial batch reactor subject to safety constraint due to reactor level swelling. The reactions are equilibrium limited, one of the products is in vapor phase, and the catalyst decomposes in the reactor. The catalyst is fed in discrete time steps during the batch, and the end-point objective is the maximum conversion in a fixed time. The reaction kinetics is determined by the temperature profile and catalyst shots, while the chemical equilibrium is shifted by operating at low pressure and removing one of the products. However, the formed vapor causes liquid swelling, due to the gas or vapor stream resulted from the reaction. As a result reaction mass may enter in the pipes and condenser, creating productivity losses and safety hazard. The end-point objective function (maximum conversion) of this problem can be converted into a level set point tracking problem. The control method is based on the moving horizon NMPC methodology and a detailed first-principles model of reaction kinetics and fluid hydrodynamics is used in the controller. The NMPC approach is based on the sequential quadratic programming (SQP) algorithm implemented in a user-friendly software environment, *OptCon*. The application of the fast real-time iteration scheme in the NMPC allows the use of small sampling period minimizing this way the violation of the maximum level constraints, due to disturbances within sampling period.

Keywords: swelling control, nonlinear model predictive control, *OptCon*.

Levente L. Simon and Konrad Hungerbuehler
ETH Zurich, Institute of Chemical and Bioengineering, Switzerland
e-mail: levente.simon@chem.ethz.ch, konrad.hungerbuehler@chem.ethz.ch

Zoltan K. Nagy
Chemical Engineering Department, Loughborough University, Loughborough, LE11 3TU, United Kingdom
e-mail: Z.K.Nagy@lboro.ac.uk

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 531–539.
springerlink.com

1 Introduction

Since the advent of dynamic matrix control (DMC), model predictive control (MPC) has been the most popular advanced control strategy in the chemical industries [1]. Linear MPC has been heralded as a major advance in industrial control [2]. However, due to their nonstationary and highly nonlinear nature, linear model based control usually cannot provide satisfactory performance in the case of complex batch processes [3]. Nonlinear model predictive control (NMPC) reformulates the MPC problem based on nonlinear process models, providing the advantage to cope inherently with process nonlinearities [4] characteristic to batch systems. Robust formulations are also available which incorporate the effect of parameter uncertainties [5]. The presented paper illustrates the benefits of an efficient on-line optimizing non-linear model based control to a simulated industrial batch reactor subject to the level constraint from safety and productivity considerations which arise from the industrially relevant problem of potential swelling. Reactor content swelling occurs when the vessel content level rises due to a gas or vapor stream that passes through the liquid. The vapor or gas stream can have different sources: gas is injected in liquid phase reactors where a reaction has to be carried out; vapor flow occurs in a reactor when the reaction produces a gas phase product which travels to the reaction mass surface; another reactor level rise is due to direct steam heating when some of the steam does not condense and disengages to the top of the vessel. As a result of the swelling phenomena reaction mass enters the pipes and the condensers connected to the reactor. As a consequence of such undesired events reactor shut-down is mandatory and production time is lost for cleaning operations. The pipe and condenser cleaning is carried out by charging solvent which is evaporated and condensed for a certain time (refluxing conditions). Reactor or evaporator content swelling phenomena can lead to significant productivity losses if it is not considered during process operation and is regarded as a reactor productivity and safety problem; the off-line optimal temperature control of batch reactors with regard to swelling was subject of investigation [6]. This work aims to implement a model based level control strategy which is required to accommodate the reaction rate disturbances which arise due to catalyst dosing uncertainties (catalyst mass and feed time).

2 Motivation and Process Model

The system considered in this study is based on a proprietary industrial batch process, for which the model has been tuned based on plant data. The catalyst used in the chemical reaction decomposes in the reaction mixture; therefore it is fed several times during the process operation. The first feeding takes place at the beginning of the operation, later on the catalyst shots are added as the reaction rate decreases. This type of process operation is often used in the industrial practice. The process is characterized by significant

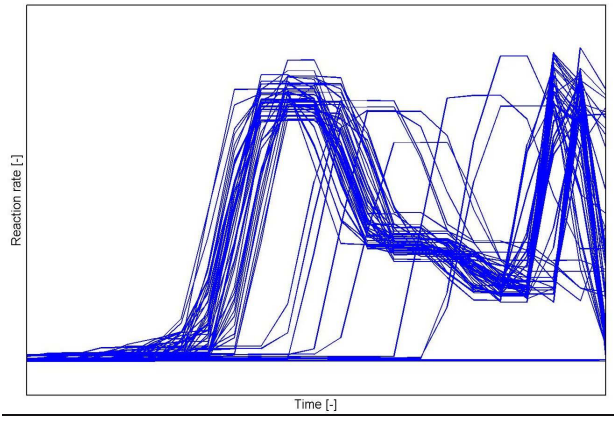


Fig. 1 Change of the reaction rate in time for the industrial batch process; the reaction rate and time values have been removed due to confidentiality

uncertainties in the addition time and mass of the catalyst. Figure 1 shows the experimental reaction rate measurements (normalized data) from the real industrial plant, in the case of repeated application of the same recipe with two consecutive catalyst dosings. The significant bath-to-batch variation of the reaction rate can be observed, which can be countered by the design of a suitable control strategy.

The reactor level is controlled using the pressure as the manipulated variable in order to compensate for the change in the reaction rate. The process operation can be optimized off-line by calculating an optimal pressure profile in function of the catalyst dosage time, dosed mass and purity. However the off-line calculated optimal pressure profile does not ensure safe operation in the case of disturbances in the catalyst feeding policy, Figure 2.

Hence, a control strategy is needed to recalculate the pressure profile during the operation considering the unknown disturbances. The used strategy is based on the nonlinear model predictive framework. During the process operation the reactor system consists of two phases: liquid and gas. Four equilibrium reactions in series take place in the liquid phase and a catalyst is used in solubilized form. The reaction scheme is as follows:

$$A + 4B \overset{Cat.}{\leftrightarrow} P + 4D \tag{1}$$

Raw materials are components A and B, and P is the desired product. The basic assumption of the kinetic model is that the reactions take place in liquid phase. In order to model the forward reactions the Arrhenius formulation is implemented, using a reference reaction constant determined at a reference temperature. The reaction volume is not constant due to two factors: on one hand there is the removal of component D and on the other hand the density

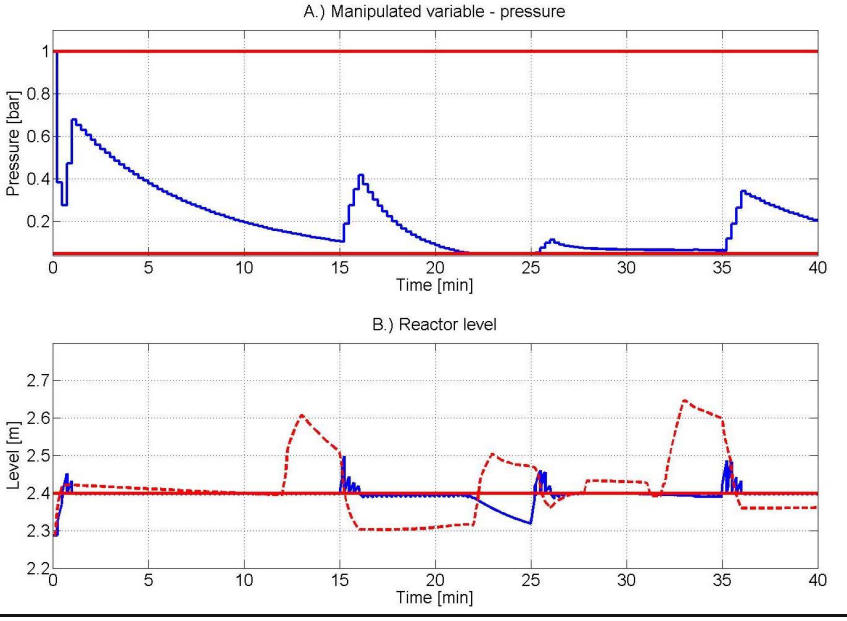


Fig. 2 Open-loop pressure profile (A) and the corresponding reactor level (B). Dashed line shows the uncertainty effect of the catalyst feeding on the reactor level (feeding takes place 3 min earlier and 20% more catalyst mass)

of the mixture changes. Although the reactions are equilibrium reactions, the system is modeled using forward reactions only. This way we calculate the safest pressure profile, since the reaction rates will never be faster than the forward reactions.

Product D is in vapor phase at the temperature and pressure conditions in the reactor, and the production of the co-product D creates a vapor flow that travels to the reaction mass surface and produces a certain void fraction in the liquid mass. The true level depends on the reaction rate, temperature, pressure, physical properties and filling degree (resting liquid). The extent of the void fraction is dependent on the liquid properties and vapor hold-up in liquid phase which in turn are dependent on the vapor flow rate, thus on the reaction rate of gas product D. In order to describe the effect of liquid swelling the pool void fraction, is used. The swelled height H [m] in terms of the average pool void fraction and the height of the resting liquid H_0 [m] is given by:

$$H = \frac{H_0}{1 - \alpha} \tag{2}$$

In order to calculate the void fraction the Wilson model is used [7]. Wilson and co-workers determined the void fraction by bubbling steam through water in a pressurized vessel in the 20-40 bar pressure range. This model has proven

suitable for void fraction calculations around 1 bar pressure as well [8]. The Wilson empirical correlation is presented in Equation 3:

$$\alpha = K \left[\frac{\rho_v}{\rho_l - \rho_v}\right]^{0.17} \left[\sqrt{\frac{\sigma}{g(\rho_l - \rho_v)}}/D_r\right]^{0.1} \left[\frac{j_v}{\left(g\sqrt{\frac{\sigma}{g(\rho_l-\rho_v)}}\right)^{0.5}}\right]^{f} \tag{3}$$

where K=0.68, f=0.62 for

$$\left[j_v/\left(g\sqrt{\frac{\sigma}{g(\rho_l - \rho_v)}}\right)^{0.5}\right] < 2 \tag{4}$$

K=0.88, f=0.40 for

$$\left[j_v/\left(g\sqrt{\frac{\sigma}{g(\rho_l - \rho_v)}}\right)^{0.5}\right] \geq 2 \tag{5}$$

where ρ_l and ρ_v [kg/m^3] are the liquid and vapor densities, σ [N/m] is the surface tension, g [m/s^2] is the gravitational acceleration, D_r [m] is the reactor vessel diameter and j_v [m/s] is the superficial vapor velocity. The connection between the chemical reactor model (ten mass and one heat balances - differential equations) and the hydrodynamic model (algebraic equations) is made by the formation rate of co-product D and the ideal gas law. The formation rate is converted into volumetric flow rate and by division with the reactor area is converted into gas velocity, j_v. Using the hydrodynamic model and the calculated gas velocity the swelled reactor level H [m] is calculated.

3 The NMPC Strategy and the *OptCon* Environment

3.1 The Control Problem

The end-point goal of the operation is the maximum conversion. Since the reactions are chemical equilibrium limited it is important that the pressure in the reactor is the lowest possible. However, minimum reactor pressure yields maximum gas volume and eventually maximum liquid level. Thus, the maximum conversion goal can be achieved by tracking the maximum reactor level. The control variable is the pressure and the batch time is fixed. The constraints on the manipulated variable P are: 0.01 mbar $<$ P $<$ 1 bar. The used process model contains information about the recipe based catalyst feeding procedure (anticipation). The presented NMPC strategy relies on state feedback. In practice, these states are measurable (concentrations) with spectroscopy based techniques (IR, UV). The measured spectra are previously calibrated to known concentration samples using partial-least squares (PLS)

models. However, the low catalyst concentration (about 1:100 to the batch size) may show a weak spectroscopic signal; the peak position and intensity is chemistry dependent.

3.2 The NMPC Formulation

The optimal control problem to be solved on-line in every sampling time in the NMPC algorithm can be formulated as:

$$\min_{u(t)\in\mathcal{U}} \{\mathcal{H}(x(t),u(t);\theta) = \mathcal{M}(x(t_F);\,\theta) + \int_{t_k}^{t_F} \mathcal{L}(x(t),u(t);\theta)dt\} \tag{6}$$

$$s.t.\;\; \dot{x}(t) = f(x(t),\; u(t);\; \theta), \qquad x(t_k) = \hat{x}(t_k), \quad x(t_0) = \hat{x}_0 \tag{7}$$

$$h(x(t),\; u(t);\; \theta) \leq 0, \qquad t \in [t_k,\; t_F] \tag{8}$$

where $\mathcal{H}$ is the performance objective, t is the time, t_k is the time at sampling instance k, t_F is the final time at the end of prediction, x is the n_x vector of states, $u(t) \in \mathcal{U}$ is the n_u set of input vectors, y is the n_y vector of measured variables used to compute the estimated states $\hat{x}(t_k)$, and $\theta \in \Theta \subset \mathcal{R}^{n_\theta}$ is the n_θ vector of possible uncertain parameters, where the set Θ can be either defined by hard bounds or probabilistic, characterized by a multivariate probability density function. The function $f : \mathcal{R}^{n_x} \times \mathcal{U} \times \Theta \rightarrow \mathcal{R}^{n_x}$ is the twice continuously differentiable vector function of the dynamic equations of the system, and $h : \mathcal{R}^{n_x} \times \mathcal{U} \times \Theta \rightarrow \mathcal{R}^{c}$ is the vector of functions that describe all linear and nonlinear, time-varying or end-time algebraic constraints for the system, where c denotes the number of these constraints. We assume that $\mathcal{H} : \mathcal{R}^{n_x} \times \mathcal{U} \times \Theta \rightarrow \mathcal{R}$ is twice continuously differentiable, thus fast optimization algorithms, based on first and second order derivatives may be exploited in the solution of (6). The form of $\mathcal{H}$ is general enough to express a wide range of objectives encountered in NMPC applications. In NMPC the optimization problem (6)-(8) is solved iteratively on-line, in a moving (receding) horizon ($t_F < t_f$) or shrinking horizon ($t_F = t_f$) approach, where t_f is the batch time.

3.3 Solution Strategy and Software Tool: OptCon

The NMPC [9] approach exploits the advantages of an efficient optimization algorithm based on the multiple shooting technique [10], [11] to achieve real-time feasibility of the on-line optimization problem, even in the case of the large control and prediction horizons. Considering the discrete nature of the on-line control problem, the continuous time optimization problem involved in the NMPC formulation is solved by formulating a discrete approximation to it, that can be handled by conventional nonlinear programming

(NLP) solvers [12]. The time horizon $t \in [t_0,\ t_f]$ is divided into N equally spaced time intervals Δt (stages), with discrete time steps $t_k = t_0 + k\Delta t$, and $k = 0, 1, ..., N$. Model equations are discretized, $x_{k+1} = f_k(x_k, u_k; \theta)$, and added to the optimization problem as constraints. For the solution of the optimization problem a specially tailored NMPC tool - *OptCon* [9]- was developed that includes a number of desirable features. In particular, the NMPC is based on first-principles or gray box models, and the problem setup can be done in Matlab. The NMPC approach is based on a large-scale NLP solver (HQP) [10], [13] which offers an efficient optimization environment, based on multiple shooting algorithm, that divides the optimization horizon into a number of subintervals (stages) with local control parameterizations. The differential equations and cost on these intervals are integrated independently during each optimization iteration, based on the current guess of the control. The continuity/consistency of the final state trajectory at the end of the optimization is enforced by adding consistency constraints to the nonlinear programming problem.

3.4 Real-Time Implementation

In NMPC simulation studies usually immediate feedback is considered, i.e. the optimal feedback control corresponding to the information available up to the moment t_k, is computed, $u^\star(t_k) = [u_{0|t_k}, u_{1|t_k}, \ldots, u_{N|t_k}]$, and the first value $(u_{0|t_k})$ is introduced into the process considering no delay. However, the solution of the NLP problem requires a certain, usually not negligible, amount of computation time δ_k, while the system will evolve to a different state, where the solution $u^\star(t_k)$ will no longer be optimal. Computational delay δ_k has to be taken into consideration in real-time applications. In the approach used here, in moment t_k, first the control input from the second stage of the previous optimization problem $u_{1|t_{k-1}}$ is injected into the process, and then the solution of the current optimization problem is started, with fixed $u_{0|t_k} = u_{1|t_{k-1}}$. After completion, the optimization idles for the remaining period of $t \in (t_k + \delta_k, t_{k+1})$, and then at the beginning of the next stage, at moment $t_{k+1} = t_k + \Delta t$, $u_{1|t_k}$ is introduced into the process, and the algorithm is repeated. This approach requires real-time feasibility for the solution of each open-loop optimization problems ($\delta_k \leq \Delta t$).

4 Results and Discussion

The simulation results using the process model with anticipation are presented in Figure 3. In the case of the early catalyst feed the level is the highest within the first sampling interval after the disturbance has been detected. We can conclude that in this scenario the anticipation information is not useful. In the second case, when the catalyst feed is delayed, the anticipation information is somewhat useful in the sense that the controller expects

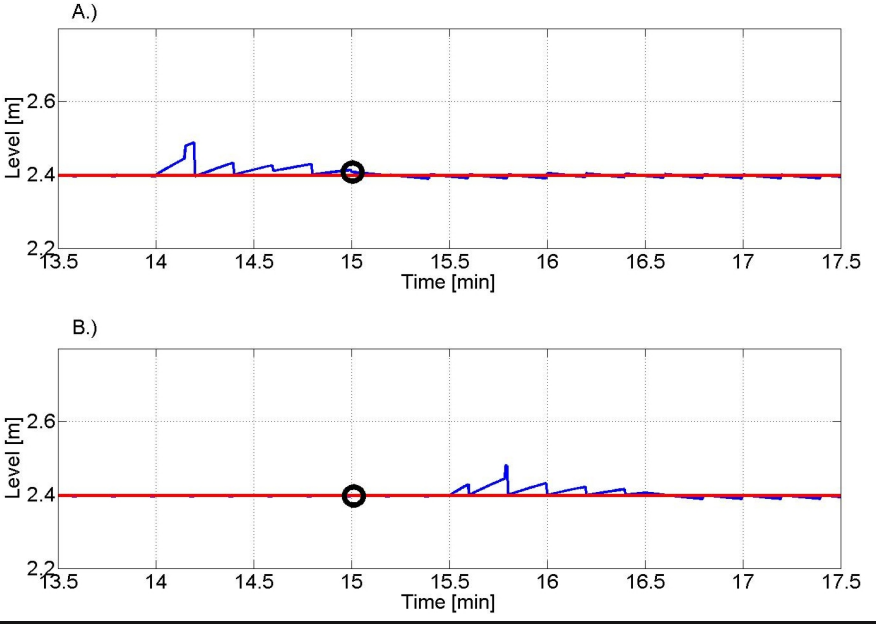


Fig. 3 Level control under early (A) and delayed (B) catalyst feeding. The circles represent the catalyst dosing time according to the recipe

that some catalyst will be fed. We can notice that the level within the first interval is lower than the second interval. Based on presented results we conclude that the level can be controlled the best if the disturbance is after the recipe based feeding time, however not too far. In the case the disturbance is before, only a small sampling time based strategy can safely control the level.

The sum of absolute deviation from the 2.4 m set point (SAD), calculated for a sampling point, for the off-line optimized profile without the control loop is 0.0783. In comparison to this, by using an NMPC strategy the index value drops to 0.0109 (early feed) and to 0.0117 (late feed) by this justifying the use of an NMPC strategy. The level tracking accuracy is directly related to the minimum batch time and productivity.

The sampling time was set to 12 seconds and the optimization calculations are completed in 1.5-2 seconds with 6-8 QP iterations.

5 Conclusions

This work presents the non-linear model based level control of a batch reactor. The control strategy is required to accommodate the reaction rate disturbances which arise due to catalyst dosing uncertainties (catalyst mass

and feeding time). It is concluded that the efficient NMPC strategy allows the implementation of small sampling interval control actions. Thus, the disturbances are quickly and efficiently rejected. Finally, we conclude that the NMPC strategy ensures safe level control during the catalyst feeding period and it drives the process along the optimal pressure profile.

Acknowledgements. The authors highly appreciate the financial support from the industrial partner.

References

1. Morari, M., Lee, J.H.: Model Predictive Control: Past, Present and Future. In: PSE 1997-ESCAPE-7 Symposium, Trondheim (1997)
2. Richalet, J., Rault, A., Testud, J.L., Papon, J.: Model Predictive Heuristic Control - Applications to Industrial Processes. Automatica 14, 413–428 (1978)
3. Qin, S.J., Badgwell, T.A.: A Survey of Industrial Model Predictive Control Technology. Control Engineering Practice 11, 733–764 (2003)
4. Allgoewer, F., Findeisen, R., Nagy, Z.K.: Nonlinear Model Predictive Control: From Theory to Application. Journal of the Chinese Institute of Chemical Engineers 35, 299–315 (2004)
5. Nagy, Z.K., Braatz, R.D.: Robust Nonlinear Model Predictive Control of Batch Processes. AIChE Journal 49, 1776–1786 (2003)
6. Simon, L.L., Introvigne, M., Fischer, U., Hungerbuehler, K.: Batch Reactor Optimization under Liquid Swelling Safety Constraint. Chemical Engineering Science 63, 770–781 (2008)
7. Wilson, J.F., Grenda, R.J., Patterson, J.F.: Steam volume fraction in bubbling two-phase mixture. Transactions of the American Nuclear Society 4, 356 (1961)
8. Wiss, J., Stoessel, F., Kille, G.: A Systematic Procedure for the Assessment of the Thermal Safety and for the Design of Chemical Processes at the Boiling-Point. Chimia 47, 417 (1993)
9. Nagy, Z.K., Allgoewer, F., Franke, R., Frick, A., Mahn, B.: Efficient Tool for Nonlinear Model Predictive Control of Batch Processes. In: 12th Mediterranean Conference on Control and Automation MED 2004, Kusadasi, Turkey (2004)
10. Franke, R., Arnold, E., Linke, H.: HQP: A solver for nonlinearly constrained large-scale optimization, http://hqp.sourceforge.net
11. Diehl, M.: Real-Time Optimization for Large Scale Nonlinear Processes. PhD Thesis, University of Heidelberg, Heidelberg (2001)
12. Biegler, L., Rawlings, J.: Optimization Approaches to Nonlinear Model Predictive Control. Chemical Process Control, South Padre Island, TX (1991)
13. Blaszczyk, J., Karbowski, A., Malinowski, K.: Object library of algorithms for dynamic optimization problems: benchmarking SQP and nonlinear interior point methods. Int. J. Appl. Math.Comput. Sci 17, 515–537 (2007)

An Application of Receding-Horizon Neural Control in Humanoid Robotics

Serena Ivaldi, Marco Baglietto, Giorgio Metta, and Riccardo Zoppoli

Abstract. Optimal trajectory planning of a humanoid arm is addressed. The reference setup is the humanoid robot James [1]. The goal is to make the end effector reach a desired target or track it when it moves in the arm's workspace unpredictably. Physical constraints and setup capabilities prevent us to compute the optimal control online, so an off-line explicit control is required. Following previous studies [2], a receding-horizon method is proposed that consists in assigning the control function a fixed structure (e.g., a feedforward neural network) where a fixed number of parameters have to be tuned. More specifically a set of neural networks (corresponding to the control functions over a finite horizon) is optimized using the Extended Ritz Method. The expected value of a suitable cost is minimized with respect to the free parameters in the neural networks. Therefore, a nonlinear programming problem is addressed that can be solved by means of a stochastic gradient technique. The resulting approximate control functions are sub-optimal solutions, but (thanks to the well-established approximation properties of the neural networks) one can achieve any desired degree of accuracy [3]. Once the off-line finite-horizon problem is solved, only the first control function is retained in the on-line phase: at any sample time t, given the system's state and the target's position and velocity, the control action is generated with a very small computational effort.

Keywords: ERIM, robotics, receding horizon.

Serena Ivaldi and Giorgio Metta
Robotics, Brain and Cognitive Science Department, Italian Institute of Technology
e-mail: {serena.ivaldi,giorgio.metta}@iit.it

Serena Ivaldi, Marco Baglietto, Giorgio Metta, and Riccardo Zoppoli
Department of Communications, Computer and System Sciences,
University of Genoa, Italy
e-mail: {mbaglietto,rzop}@dist.unige.it

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 541–550.
springerlink.com

1 Introduction

In robotics, the task of positioning the end effectors is fundamental: whenever a robot has to move its arm in order to grasp an object, track a moving target, avoid collision with the environment or just explore it, reaching is involved. Given the target position, estimated for example by a vision system, it is common practice to plan a suitable trajectory in the cartesian space and then to find the corresponding joint and torque commands. In industrial robotics, trajectories usually have a parametrized but fixed structure, e.g. splines or polynomials, or motor commands can be found analytically after the minimization of some Lyapunov function describing the reaching goal. In humanoid robotics, the focus is not only on reaching the target, but on how the target is reached, that is the criterion which a certain limb accomplishes while performing a movement or acting on the environment. One of the main goals of humanoid robotics is indeed to exploit redundancy and constraints of the humanoid shape to achieve behaviors that are approximately efficient as human movements. It is common belief that the human body moves "optimally" with respect to different cost functions, depending on action, limbs, task. In order to give a humanoid robot the chance to implement different motion criteria, it is necessary to provide a technique which allows finding optimal control commands for any given cost function. To this end, a Finite Horizon (FH) optimal control problem can be considered, but it is scarcely useful as generally the duration of the movements cannot be predicted *a priori*. Moreover, moving through a fixed horizon strategy could lead to a lack of responsiveness, whenever the target dynamics is too fast and no previous information is available to predict the target behavior. A Receding Horizon (RH) approach is suggested. Within the classical RH approach, at each time instant t, when the system state is $\mathbf{x}_t$, a FH optimal control problem is solved and a sequence of N optimal control actions is computed, $\mathbf{u}^{FH}_{0|t}, \mathbf{u}^{FH}_{1|t}, \ldots, \mathbf{u}^{FH}_{N-1|t}$ (corresponding to velocity, acceleration or torque commands, depending on the controller design), which minimize a suitable cost function affecting the motion performance; then only the first control vector is applied: $\mathbf{u}^{RH}_t = \mathbf{u}^{FH}_{0|t}$. This procedure is repeated at each instant t, thus yielding a feedback control law. Stabilizing properties of RH control have been shown for both linear and nonlinear systems, in continuous and discrete time, using the terminal equality constraints $\mathbf{x}_{t+N} = \mathbf{0}$ [4], relaxing it [5] and just imposing the attractiveness of the origin by means of penalty functions [2]. The classical RH technique assumes the control vectors to be generated after the solution of a nonlinear programming problem at each time instant: this assumption is generally unrealistic in the case of humanoid robotics, as the robot's and the target's dynamics are fast and the complexity of the problem increase with the number of DOF to control. In order to solve the optimization problem on-line, with the guarantee of respecting the temporal constraint, a suitable hardware and software are required, usually a real-time processing unit supporting fast and highly precise computations,

directly connected to the robot' sensing and actuation devices, which is utterly complicated for complex kinematic structures. Unfortunately, different multi-level control architectures often do not support this control scheme. This is the case of our humanoid robot, James [1]. James is a humanoid torso, consisting of a head and a left arm , with the overall size of a 10 years old boy. Of the 7 DOF of the arm (3-shoulder, 1-elbow, 3-wrist), only 4 have been used in this paper (the wrist is considered as the end-effector), while numerical results are shown for the 2 DOF case. Torque is transmitted to the joints by rubber toothed belts, pulleys and stainless-steel tendons, actuated by rotary DC motors. The robot's motion can be controlled by sending position and velocity commands from a remote PC to 12 Digital Signal Processing (DSP) boards (Freescale DSP56F807, 80MHz, fixed point 16 bits), via CAN bus. DSP boards have limited memory and computation capability and cannot support more than simple operations, namely low level motor control (mostly PID controllers, 1KHz rate), signal acquisition and pre-filtering from the encoders. For this reason, implementing an on-line controller is impossible in the current setup: an explicit off-line RH controller is considered. The goal of this work is to design a feedback RH regulator for reaching tasks, with the requirement of being quick and reactive to changes, in particular to track a target moving unpredictably in the robot's workspace. We will also describe a technique which concentrates the computation of a time-invariant feedback optimal control law in an off-line phase , for every possible system and target states belonging to an opportune set of admissible states. The proposed algorithm consists of two steps. In the first step, a suitable sequence of neural networks is trained off line, so that they can approximate the optimal solutions of a stochastic FH control problem, which is generalized for every possible state configuration. In the second (online phase), only the first control law is applied, at each time instant. The Extended RItz Method (ERIM) [6] is chosen as a functional approximation technique. The use of feedforward neural networks (thanks to their well known approximation capabilities [7]) guarantees that the optimal solutions can be approximated at any desired degree of accuracy. We would like to remark that the computation demand is concentrated in the off-line phase, while in the on-line phase only the computation of a single control law is performed, thus yielding a fast response to unpredictable changes in the target's state, since we can do the computations quickly. The feasibility of this approach has already been tested on the control of a thrusts-actuated nonholonomic robot [8]. James can be modeled as an open kinematic chain. In the following we shall only focus on the arm motion control, in particular from the shoulder up to the wrist, which will be considered as the end effector of the kinematic chain, and neglect the rotation of the hand. Let us denote by $\mathbf{x}_c^r$ the cartesian coordinates of the end effector with respect to a base frame fixed to the robot waist, and by $\mathbf{q}$ the vector of the joint coordinates of the arm. Then the forward kinematics $\mathbf{x}_c^r = f_{\text{arm}}(\mathbf{q})$, $f_{\text{arm}} : \mathbb{R}^{n_q} \rightarrow \mathbb{R}^{n_c}$, can be easily found by measuring the length of the robot links and represent it with the Denavit-Hartenberg

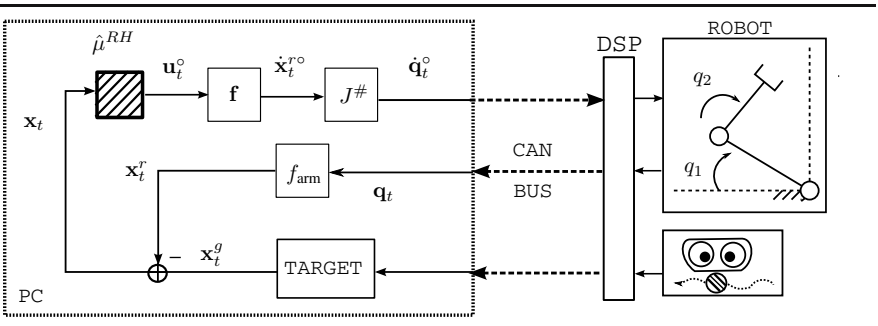


Fig. 1 James's arm control scheme. Velocity commands are sent through a CAN bus, while direct motor control is performed by DSP cards. The retrieving of the target's cartesian coordinates is not modeled, as it would require to discuss the robotic visual system. The arm kinematic model is reported. We indicated two arm joints (q_1, q_2), corresponding to the case of a 2 DOF arm $(n_q = 2)$

convention [9]. We shall denote by $\mathbf{x}_t^r$ and $\mathbf{x}_t^g$ the robot's end effector state vector and the target's one at time instant t. We remark that once the optimal control $\mathbf{u}_t^\circ$ is found, then the optimal velocity controls in the joint space can be easily computed with standard formulations, i.e., $\dot{\mathbf{q}}_t^\circ = J^\#(\mathbf{q}_t)\dot{\mathbf{x}}_t^{r\circ}$, where $J^\#$ denotes the Moore-Penrose pseudo-inverse of the jacobian matrix $J(\mathbf{q}) = \partial f_{\text{arm}}(\mathbf{q})/\partial \mathbf{q}$, being $\dot{\mathbf{x}}_c^r = J(\mathbf{q})\dot{\mathbf{q}}$. In particular, as explained in the previous section, they are computed by a standard Pentium based PC, then sent through the CAN bus to the DSP cards, where the low level control loop is performed. The control scheme is shown in Figure 1.

2 Receding Horizon Regulator: A Neural Approach

The goal of the reaching control problem is to find, at any time instant t, the optimal control $\mathbf{u}_t^\circ$ minimizing a suitable cost function, which is chosen so as to characterize the trajectories of the end effector reaching or tracking a target moving unpredictably in the robot workspace. We denote by $\mathbf{x}_t$, at time instant t, the difference between the end effector and the target cartesian coordinates and velocities ($\mathbf{x}_t \triangleq col(\mathbf{x}_t^g - \mathbf{x}_t^r, \dot{\mathbf{x}}_t^g - \dot{\mathbf{x}}_t^r)$). Let us represent the previous equations in the more general and compact form

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) \ , \ t = 0, 1, \ldots$$

where at the time instant t, $\mathbf{x}_t$ is the state vector, taking values from a finite set $X \subseteq \mathbb{R}^n$, and $\mathbf{u}_t$ is the control vector, constrained to take values from a finite set $U \subseteq \mathbb{R}^m$. At any time instant t, the desired state is $\mathbf{x}_t^* = 0$, meaning that the goal is to bring the difference between the end effector and

the target to zero. By making this assumption, we implicitly apply a certainty equivalence principle: at time instant t, the target vector $\mathbf{x}_t^g$ is supposed to remain constant for N time instants, that is: $\mathbf{x}_{t+i+1}^g = \mathbf{x}_{t+i}^g, i = 0, \ldots, N-1$. We can now state a RH control problem.

Problem 1. At every time instant $t \geq 0$, find the RH optimal controls $\mathbf{u}_t^\circ \in U$, where $\mathbf{u}_t^\circ$ is the first vector of the control sequence $\mathbf{u}_{0|t}^\circ, \ldots, \mathbf{u}_{N-1|t}^\circ$ that minimize the FH cost functional

$$\mathcal{J}(\mathbf{x}_t) = \left\{ \sum_{i=0}^{N-1} h_i(\mathbf{x}_{t+i}, \mathbf{u}_{i|t}) + h_N(\mathbf{x}_{t+N}) \right\} .$$

The classical RH control assumes that at each time instant of control a FH control problem is solved, and a sequence of N optimal controls is found. As we previously discussed, this approach is not suitable in our case, for the hardware limitations imposed by the DSP cards. Therefore we will change the problem's formulation so as to be able to compute the control laws in an off-line phase.

Problem 2 (RH). For every time instant $t \geq 0$, find the RH optimal control law $\mathbf{u}_t^\circ = \mu_t^\circ(\mathbf{x}_t) \in U$, where μ_t° is the first control function of the sequence $\mu_{0|t}^\circ, \ldots, \mu_{N-1|t}^\circ$ that minimize the FH cost functional [1]

$$\bar{\mathcal{J}}_t = \underset{\mathbf{x}_t \in X}{E} \left\{ \sum_{i=0}^{N-1} h_i(\mathbf{x}_{t+i}, \mu_{i|t}(\mathbf{x}_{t+i})) + h_N(\mathbf{x}_{t+N}) \right\} .$$

Thanks to the time invariance of the system dynamics and of the cost function, $t = 0$ can be considered as a generic time instant. Then, a single (functional) FH optimization problem is addressed.

Problem 3 (FH). Find a sequence of optimal control functions $\mu_0^\circ, \ldots, \mu_{N-1}^\circ$, that minimize the cost functional

$$\bar{\mathcal{J}} = \underset{\mathbf{x}_0 \in X}{E} \left\{ \sum_{i=0}^{N-1} h_i(\mathbf{x}_i, \mu_i(\mathbf{x}_i)) + h_N(\mathbf{x}_N) \right\} \tag{1}$$

subject to the constraints $\mu_i^\circ \in \mathbf{U} \subseteq \mathbb{R}^m$ and $\mathbf{x}_{i+1} = \mathbf{f}(\mathbf{x}_i, \mu_i(\mathbf{x}_i))$.

The RH control strategy will correspond to use μ_0° as a time invariant control function, i.e., to apply $\mathbf{u}_t^{RH} = \mu^{RH}(\mathbf{x}_t) = \mu_0^\circ(\mathbf{x}_t)$.

[1] Hereinafter, the notation $\underset{\xi}{E}\{g(\xi)\}$ means the expectation of function g with respect to the stochastic variable ξ. It is important to notice that in Problem 1 the expectation is not necessary, because it is a deterministic problem.

2.1 From a Functional Optimization Problem to a Nonlinear Programming One

In order to solve Problem FH we shall apply the ERIM [6], by which the functional optimization problem is transformed into a nonlinear programming one. More specifically, we constrain the admissible control functions $\mu_0, \mu_1, \ldots, \mu_{N-1}$ to take on a fixed parametrized structure, in the form of one-hidden-layer (OHL) neural networks:

$$\hat{\mu}_i(\mathbf{x}_i, \omega_i) = col\left[\sum_{h=1}^{\nu} c_{hj}\varphi_h(\mathbf{x}_i, \kappa_h) + b_j\right] \tag{2}$$

where $\hat{\mu}_i(\cdot, \omega_i) : \mathbb{R}^n \times \mathbb{R}^{(n+1)\nu+(\nu+1)m} \mapsto \mathbb{R}^m$, $c_{hj}, b_j \in \mathbb{R}, \kappa_h \in \mathbb{R}^k, j = 1, \ldots, m$, being ν the number of *neurons* constituting the network. By substituting (2) into (1), calling ω_i the parameters of the i-th OHL network $\hat{\mu}_i(\mathbf{x}_i, \omega_i)$, the general functional cost $\bar{\mathcal{J}}(\mu_0, \mu_1, \ldots, \mu_{N-1})$ is turned into a function $\hat{\mathcal{J}}_\nu(\omega)$ which is only dependent on a finite number of real parameters, $\omega = col(\omega_i, i = 0, 1, \ldots, N-1)$. We can now restate Problem 3 as:

Problem 4 (FH_ν). *Find the optimal vectors of parameters $\omega_0^\circ, \ldots, \omega_{N-1}^\circ$ that minimize the cost function*

$$\hat{\mathcal{J}}_\nu = \underset{\mathbf{x}_0 \in X}{E}\left\{\sum_{i=0}^{N-1} h_i(\mathbf{x}_i, \hat{\mu}_i(\mathbf{x}_i, \omega_i)) + h_N(\mathbf{x}_N)\right\}$$

subject to the constraints $\hat{\mu}_i(\mathbf{x}_i, \omega_i) \in \mathbf{U} \subseteq \mathbb{R}^m$ and $\mathbf{x}_{i+1} = \mathbf{f}(\mathbf{x}_i, \hat{\mu}_i(\mathbf{x}_i, \omega_i))$.

Then, for every time instant t, the time-invariant RH control law corresponds to $\mathbf{u}_t^{RH} = \hat{\mu}^{RH}(\mathbf{x}_t, \omega_0^\circ) = \hat{\mu}_0^\circ(\mathbf{x}_t, \omega_0^\circ)$.

2.2 Solution of the Nonlinear Programming Problem by Stochastic Gradient

The optimal parameters in the OHL control functions can be found by a usual gradient algorithm, i.e.

$$\omega_i(k+1) = \omega_i(k) - \alpha(k)\nabla_{\omega_i} \underset{\{\mathbf{x}_0\}}{E}\left\{\hat{\mathcal{J}}_\nu\left[\omega(k), \mathbf{x}_0\right]\right\}, \; k = 0, 1, \ldots \; .$$

Within this context, it is impossible to calculate exactly all the gradient components, because of the stochastic nature of $\mathbf{x}_0$; then, instead of the gradient $\nabla_\omega E\left[\hat{\mathcal{J}}_\nu(\omega, \mathbf{x}_0)\right]$ a single "realization" $\nabla_\omega \hat{\mathcal{J}}_\nu(\omega, \mathbf{x}_0(k))$ is computed, where the stochastic variable $\mathbf{x}_0$ is generated randomly according to its known probability density function. Then a simple gradient steepest descent algorithm can be applied:

$$\omega_i(k+1) = \omega_i(k) - \alpha(k)\nabla_{\omega_i}\hat{\mathcal{J}}_\nu\left[\omega(k), \mathbf{x}_0(k)\right] + \eta(\omega_i(k) - \omega_i(k-1))$$

for $k = 0, 1, \ldots$, where we added a regularization term, weighted by $\eta \in [0, 1]$, as it is usually done when training neural networks. The convergence of the method, which is known as *stochastic gradient*, is assured by a particular choice of the step size $\alpha(k)$, that must fulfill a set of conditions [10]. Of course, one has to compute the partial derivatives of the cost $\hat{\mathcal{J}}_\nu$ with respect to the parameters to be optimized, ω_i:

$$\frac{\partial \hat{\mathcal{J}}_\nu}{\partial \omega_i} = \frac{\partial \hat{\mathcal{J}}_\nu}{\partial \mathbf{u}_i} \frac{\partial \hat{\mu}_i(\mathbf{x}_i, \omega_i)}{\partial \omega_i} \ .$$

The proposed algorithm for the computation of the optimal parameters consists in two phases, a forward and a backward one, and in a back-propagation technique. In the *forward phase* we "unroll" the system and the neural controllers in time, making the feedback explicit. At iteration step k, given the initial state $\mathbf{x}_0$, we compute all the state and controls generated by the sequence of OHL networks that is $\mathbf{u}_i = \hat{\mu}_i(\mathbf{x}_i, \omega_i(k))$, given $\mathbf{x}_0, \mathbf{x}_i = f(\mathbf{x}_{i-1}, \mathbf{u}_{i-1})$, $i = 1, \ldots, N$. Then we can compute all the partial costs $h_i(\mathbf{x}_i)$, $h_N(\mathbf{x}_N)$. In the *backward phase*, we compute all the gradient components and "back-propagate" them through the networks' chain. The recursive propagation is described by the following equations, for $i = N-1, N-2, \ldots, 0$:

$$\frac{\partial \hat{\mathcal{J}}_\nu}{\partial \mathbf{u}_i} = \frac{\partial h_i(\mathbf{x}_i, \mathbf{u}_i)}{\partial \mathbf{u}_i} + \frac{\partial \hat{\mathcal{J}}_\nu}{\partial \mathbf{x}_{i+1}} \frac{\partial f(\mathbf{x}_i, \mathbf{u}_i)}{\partial \mathbf{u}_i}$$

$$\frac{\partial \hat{\mathcal{J}}_\nu}{\partial \mathbf{x}_i} = \frac{\partial h_i(\mathbf{x}_i, \mathbf{u}_i)}{\partial \mathbf{x}_i} + \frac{\partial \hat{\mathcal{J}}_\nu}{\partial \mathbf{x}_{i+1}} \frac{\partial f(\mathbf{x}_i, \mathbf{u}_i)}{\partial \mathbf{x}_i} + \frac{\partial \hat{\mathcal{J}}_\nu}{\partial \mathbf{u}_i} \frac{\partial \hat{\mu}_i(\mathbf{x}_i, \omega_i)}{\partial \mathbf{x}_i}$$

initialized by $\partial \hat{\mathcal{J}}_\nu / \partial \mathbf{x}_N = \partial h_N(\mathbf{x}_N) / \partial \mathbf{x}_N$.

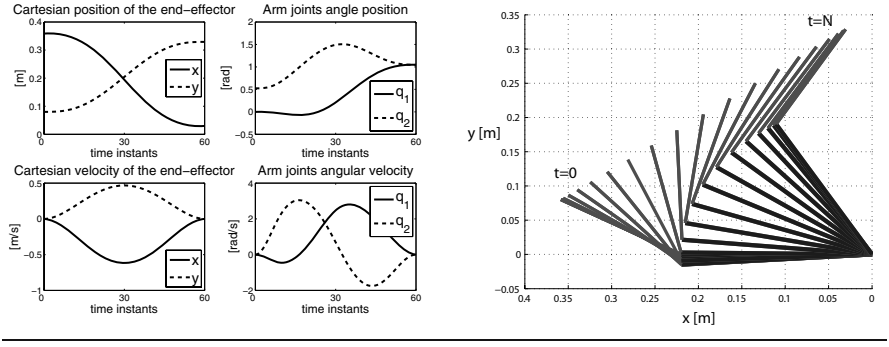


Fig. 2 A minimum jerk movement of James'arm: cartesian and joints position and velocity are shown, as well as samples of the planar trajectory. The neural approximation and the analytical solution [11] coincide (m.s.e. $\cong 10^{-7}$)

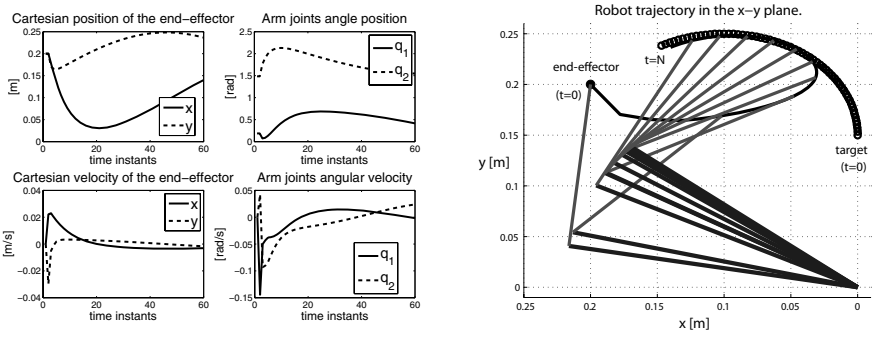


Fig. 3 James' left end effector tracking a target moving in an unpredictable way, according to cost function (3), where $V_i = diag(1.0, 80.0, 5.0, 10.0)$, $V_N = 40I$. Moreover, $N = 30, \nu = 40$

3 Results

Many neuro-computational studies investigate the arm motion on a plane, considering the arm as a two-rotative joints limb. In this case, it has been shown that the human arm movement can be approximated by the function optimizing the following cost function (*minimum jerk* principle) [11]:

$$\mathcal{J} = \int_0^T \left[\left(\frac{d^3 x^r}{dt^3} \right)^2 + \left(\frac{d^3 y^r}{dt^3} \right)^2 \right] dt \ .$$

This criterion has been chosen to verify the effectiveness of the proposed method. We set $n_q = n_c = 2$ to consider James' arm as a two-link rigid body, moving on a planar surface, $T = 60$, $\nu = 40$, and used approximatively 10^9 samples for the off-line training of the neural networks. Results are shown in Figure 2. The method has been also tested with a different cost function:

$$\mathcal{J} = \sum_{i=t}^{t+N-1} \mathbf{c}(\mathbf{u}_i) + \mathbf{x}_{i+1}^T V_{i+1} \mathbf{x}_{i+1} \tag{3}$$

where the criterion for the task accomplishment is a tradeoff between the minimization of the energy consumption (for physical limits, it is important not to exceed in the maximum rated current consumption) and the "best" end-effector proximity to the target during and at the end of the manoeuvre (it could not be able to reach it perfectly, as a consequence of the unpredictable behavior of the target or the robot's intrinsic physical limits). Weight matrices V_i are chosen such as to obtain reasonable compromise between the attractiveness of the target and the energy consumption, whereas $c(u_t^j), j = x, y$ is a nonlinear but convex function (the same reported in [8]). An example of a

RH trajectory during a tracking task are shown in Figure 3. We remark that the constraints on the admissible values of $\mathbf{x}_t$ and $\mathbf{u}_t$ are always verified. To be more precise, the classical OHL networks were slightly modified, specifically by adding two bounded sigmoidal functions $\sigma(z) = U \tanh(z)$ to the final output layer: with this choice, the constraints on the control values can be removed from the problem formulation since the neural networks already embed them.

4 Conclusion

This paper focused on the computation of a neural time invariant feedback control law optimized off-line. The on-line computation of the control action is efficient, as it consists only of few mathematical operations. We point out that the requirement of computing control values in real-time as fast as possible is strict. Given that this method has been designed to be applied to a full body humanoid robot, we concentrated in making the computation of the control law as efficient as possible. We have presented simulations to clarify the problem. Early experiments on James, controlling 2 DOF, have confirmed the effectiveness of the proposed approach. Simulations for the control of the 4 DOF arm are currently ongoing. In the future, the control scheme will take into account singularities, redundancies of the kinematic chain, and delays which have been neglected for the moment.

Acknowledgement. This work is supported by the RobotCub project (IST-2004-004370), funded by the European Commission through the Unit E5 "Cognitive Systems, Robotics and Interfaces".

References

1. Jamone, L., Metta, G., Nori, F., Sandini, G.: James: a humanoid robot acting over an unstructured world. In: Int. Conf. on Humanoids Robots, Italy (2006)
2. Parisini, T., Zoppoli, R.: A receding horizon regulator for nonlinear systems and a neural approximation. Automatica 31, 1443–1451 (1995)
3. Kurkova, V., Sanguineti, M.: Error estimates for approximate optimization by the Extended Ritz method. SIAM J. on Optimization 15, 461–487 (2005)
4. Keerthi, S., Gilbert, E.: Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: stability and moving-horizon approximations. Journ. of Optim. Theory and Applications 57, 265–293 (1988)
5. Michalska, H., Mayne, D.: Robust receding horizon control of constrained nonlinear systems. IEEE Trans. on Automatic Control 38, 1623–1633 (1993)
6. Zoppoli, R., Sanguineti, M., Parisini, T.: Approximating networks and Extended Ritz Method for the solution of functional optimization problem. Journ. of Optim. Theory and Applications 112, 403–439 (2002)
7. Barron, A.: Universal approximation bounds for superposition of a sigmoidal function. IEEE Trans. on Information Theory 39, 930–945 (1993)

8. Ivaldi, S., Baglietto, M., Zoppoli, R.: Finite and receding horizon regulation of a space robot. In: Int. Conf. on Mathem. Probl. in Engin., Aerospace and Sciences, Genoa, Italy (2008)
9. Sciavicco, L., Siciliano, B.: Modeling and Control of Robot Manipulators, 2nd edn. Springer, London (2000)
10. Kushner, H.J., Yang, J.: Stochastic approximation with averaging and feedback: rapidly convergent "on-line" algorithms. IEEE Trans. on Automatic Control 40, 24–34 (1995)
11. Richardson, M.J., Flash, T.: On the Emulation of Natural Movements by Humanoid Robots. In: Int. Conf. on Humanoids Robots, Boston, USA (2000)

Particle Swarm Optimization Based NMPC: An Application to District Heating Networks

Guillaume Sandou and Sorin Olaru

Abstract. Predictive control is concerned with the on-line solution of successive optimization problems. As systems are more and more complex, one of the limiting points in the application of optimal receding horizon strategy is the tractability of these optimization problems. Stochastic optimization methods appear as good candidates to overcome some of the difficulties. Indeed, these methods are not dependent on the structure of costs and constraints (linear, convex...), can escape from local minima and do not require the computation of local informations (gradient, hessian). In this paper, a Particle Swarm Optimization (PSO) is proposed to solve the receding horizon principle with an application to district heating networks. Tests of the approach are given for a network benchmark, showing that more than satisfactory results are achieved, compared with classical control laws for such systems.

Keywords: Particle swarm optimization (PSO), NMPC, Energy savings.

1 Introduction

Receding horizon based methods are efficient tools to control industrial systems based on the introduction of on-line optimisation problems in the feedback loop. The systems to control being more and more complex, the corresponding optimization problems continuously challenge the limitation of classical deterministic methods such as Successive Quadratic Programming. These optimization problems can often be solved by stochastic optimization algorithms which are able to find good suboptimal solutions to hard

Guillaume Sandou and Sorin Olaru
SUPELEC, Automatic Control Department, 3 rue Joliot Curie
91192 Gif-sur-Yvette, France
e-mail: guillaume.sandou,sorin.olaru@supelec.fr

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 551–559.
springerlink.com

optimization problems. Of course, a severe attention has to be paid to the stability of the corresponding closed loop control law. In this way of thinking, ant colony was used in previous studies to compute a receding horizon control law for the case of constrained hybrid systems [1]. In this paper, the focus is on the use of an other metaheuristic optimization method, Particle Swarm Optimization (PSO) to compute a closed loop law for the control of district heating networks.

In a competitive technological, economical and environmental context, *Energy Savings* has emerged as a crucial point. Due to the development of cogeneration systems or heat storage tanks, the use of district heating networks appears as an interesting way to achieve high global efficiencies of energy networks. However, the modeling of these systems are concerned with partial differential equations (for the computation of thermal energy propagation) and non linear algebraic and implicit equations (for the computation of mass flows and pressures in the whole system). Finally, costs and constraints of the corresponding continuous optimization problems can only be computed in a simulation environment. The problem could be solved by deterministic algorithms. As no analytic expressions are available, the computation of descent directions (gradient or subgradients) imply numerous evaluations of costs and constraints functions. Further, numerous local minima do exist, and near optimal initial points have to be known to get satisfactory results. Thus, this kind of system appears to be a good benchmark for the study of stochastic optimization algorithms for the computation of closed loop control laws.

The paper is organized as follows. PSO is firstly described in section 2. District heating networks are presented and modeled in section 3 together with the use of PSO for the receding horizon based control law. Finally, conclusions and forthcoming works are drawn in section 4.

2 Particle Swarm Optimization Based NMPC

2.1 Classical PSO Algorithm

Particle swarm Optimization (PSO) was firstly introduced by Russel and Eberhart [2]. This optimization method is inspired by the social behavior of bird flocking or fish schooling. Consider the following optimization problem:

$$\min_{x \in \chi} f(x) \tag{1}$$

P particles are moving in the search space. Each of them has its own velocity, and is able to remember where it has found its best performance. Each particle has some "friends". The following notations are used:

- x_k^p (resp. v_k^p): position (resp. velocity) of particle p at iteration k;
- $b_k^p = \text{argmin}(f(x_{k-1}^p, b_{k-1}^p))$: best position found by particle p until iteration k;

- $V(x_k^p) \subset \{1, \ldots, P\}$: set of "friend particles" of particle p at iteration k;
- $g_k^p = \text{argmin}(f(x_k^j), j \in V(x_k^p))$: best position found by the friend particles of particle p until iteration k.

The particles move in the search space according to the transition rule:

$$\begin{aligned} v_{k+1}^p &= w \times v_k^p + c_1 \otimes (b_k^p - x_k^p) + c_2 \otimes (g_k^p - x_k^p) \\ x_{k+1}^p &= x_k^p + v_{k+1}^p \end{aligned} \tag{2}$$

where w is the inertia factor, $\otimes$ denotes the element wise multiplication of vectors and c_1 (resp. c_2) is a random vector whose length is the number of optimization variables, and whose components are in the range $[0, \bar{c_1}]$ (resp. $[0, \bar{c_2}]$).

The construction of the transition rule 2 is represented in figure 1.

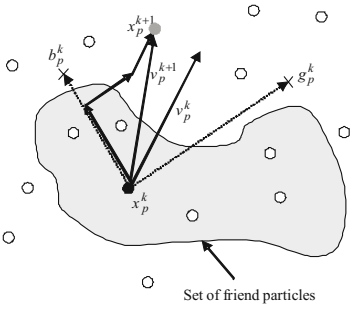


Fig. 1 Geometric representation of the transition rule

The choice of parameters is very important to ensure the satisfying convergence of the algorithm. Important results have been reported on the topic; see for instance [3], [4]. Is beyond the scope of the present study to present the exhaustive description of tuning strategies (the Automatic Control community being less enthusiastic about metaheuristics details). Standard values, which are given in [5] will be used: swarm size $P = 10 + \sqrt{n}$, where n is the number of optimization variables, $w = \frac{1}{2 \cdot \ln(2)}$, $\bar{c_1} = \bar{c_2} = 0.5 + \ln(2)$

Several topologies exist for the design of the set of *friend* particles. For a comprehensive study of this topic, see [6]. In particular, if these sets do not depend on k, neighborhoods are said to be "social". This choice is the simplest for the implementation of the algorithm and so a social neighborhood will be used in this paper.

2.2 Application to NMPC

NMPC is based on the on-line solution of optimization problems. Due to the increasing complexity of the systems to control, these optimization problems become harder and harder to solve, especially in a real time framework. Thus, the use of PSO as a solver for these problems appear as an interesting trend

to overcome some of the difficulties, as it is able to find good near optimal solutions with relatively low computation times.

Of course, the main drawback of this optimization method is the fact that no guarantee can be given on the actual optimality of the solution. Thus, stability and robustness has to be carefully studied before the effective implementation of PSO for NMPC.

3 District Heating Networks Control

3.1 District Heating Networks Modeling

A district heating network is depicted in figure 2. It is a part of a more general district heating network which has been reported in [7]. It is made of two main subnetworks which are interconnected with the help of two valves. The main components which have to be modeled are the producers, the energy supply network made of pipes, pumps and nodes, and consumers.

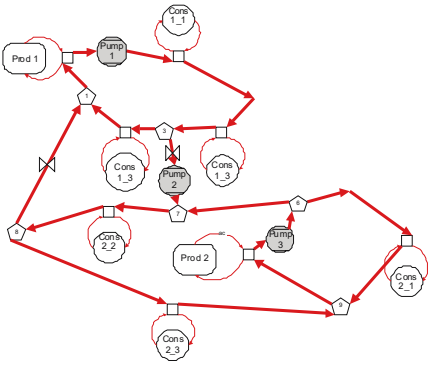


Fig. 2 District heating network benchmark

Producers

A production site s is modeled by a characteristic, identified from technical data. For hour n, production costs can be derived from produced thermal power Q_n^s:

$$c_{prod}^s(Q_n^s, Q_{n-1}^s) = a_2^s(Q_n^s)^2 + a_1^s Q_n^s + a_0 + \lambda(Q_n^s - Q_{n-1}^s)^2 \tag{3}$$

where λ is a weighting factor penalizing the control increments, and modeling dynamics of production units. The thermal power given to primary network is related to network temperatures by:

$$Q_n^s = c_p \cdot m \cdot (T_s - T_r) \tag{4}$$

where m[kg.s^{-1}] is the mass flow in the energy supply network, T_s(K) the supply temperature, T_r(K) the return temperature in primary network and c_p[J.kg^{-1}.K^{-1}] the specific heat of water.

Energy supply network

The energy supply network is concerned with pipes, valves, nodes and pumps. Notations for pipe modeling are given in figure 3.

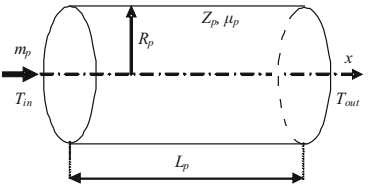


Fig. 3 Notations for pipes modeling

Mechanical losses in pipes can be expressed by:

$$H_{out} = H_{in} - Z_p \cdot m_p^2 \tag{5}$$

with m_p[kg.s^{-1}] the mass flow in pipe, H_{in} (resp. H_{out}) [m] the pressure at the beginning (resp. the end) of the pipe, and Z_p(m.kg^{-2}.s^2) the friction coefficient. For a valve, this coefficient becomes Z_p/d, where d is the opening degree of the valve (from 0 for a closed valve to 1 for an open one) which is a control input of the system.

The thermal energy propagation in pipes can then be modeled by a partial differential equation:

$$\frac{\partial T}{\partial t}(x,t) + \frac{m_p(t)}{\pi \rho R_p^2}\frac{\partial T}{\partial x}(x,t) + \frac{2\mu_p}{c_p \rho R_p}(T(x,t) - T_0) = 0 \tag{6}$$

To counterbalance mechanical losses in pipes, pumps are installed in the network leading to an increase of pressure:

$$\Delta H = b_2 (m\frac{\omega_0}{\omega})^2 + b_1 m \frac{\omega_0}{\omega} + b_0 \tag{7}$$

with m[kg.s^{-1}] is the mass flow through the pump, ω[rad.s^{-1}] its rotation speed and ω_0 its nominal rotation speed.

Finally, nodes in the network are easily modeled using mass flow balance equations and energy balance equations.

Consumers

Secondary networks of consumers are connected to the primary network by way of heat exchangers. Notations are those of figure 4.

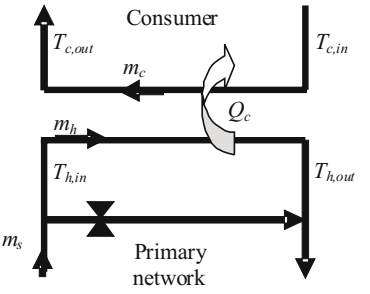


Fig. 4 Notations for consumers modeling

The following equation is the classical equation for a counter flow heat exchanger with $S[\mathrm{m}^2]$ the surface of the heat exchanger, and $e[\mathrm{W.K}^{-1}.\mathrm{m}^{-2}]$ its efficiency:

$$Q_c = eS\frac{(T_{h,in} - T_{c,out}) - (T_{h,out} - T_{c,in})}{\ln(T_{h,in} - T_{c,out}) - \ln(T_{h,out} - T_{c,in})} \tag{8}$$

Assuming no thermal energy loss between primary and secondary networks, the thermal power given by the primary network can be also expressed by:

$$Q_c = c_p m_h (T_{h,in} - T_{h,out}) \tag{9}$$

$$Q_c = c_p m_c (T_{c,out} - T_{c,in}) \tag{10}$$

Assuming that m_c and $T_{c,out}$ are given, and that mass flow m_h is determined by the opening degree of the valve, then $T_{c,in}$, Q_c and $T_{h,out}$ can be computed from $T_{h,in}$. Q_c is an increasing function of m_h: the maximal thermal power which can be given to a consumer is obtained for $m_h = m_s$. Consequently, the given power is finally expressed by:

$$Q_c = \min(Q_{dem}, Q\max) \tag{11}$$

where Q_{dem} is the heat demand of the consumer, and Q_{max} is the maximum power that can be given by the primary network. Q_{max} is computed by solving the system made of 8, 9 and 10, in the particular case $m_h = m_s$.

3.2 Receding Horizon Based Control of District Heating Networks

Open loop and optimization

Consider a district heating network, with S production sites, V valves and C consumers. For simplicity, rotation speeds of pumps are supposed to be constant. The open loop control law of the whole system can be computed from the solution of the optimization problem:

$$\begin{array}{c} \min \\ \{Q_n^s, d_n^v\}, n \in \{1, \ldots, N\} \\ s \in \{1, \ldots, S\}, v \in \{1, \ldots, V\} \end{array} \quad \sum_{n=1}^{N} \sum_{k=1}^{K} c_{prod}^{s}(Q_n^s, Q_{n-1}^s) \tag{12}$$

where, Q_n^s is the thermal power produced by site s during time interval n and d_n^v is the opening degree of valve v during time interval n.

Constraints are the satisfaction of technical constraints (pressures and mass flows in the energy supply network) and the fulfilling of consumers demands $Q_{dem,n}^c, c \in \{1, \ldots, C\}$. To compute these constraints, one has to simulate the whole network. From the modeling details presented in the previous section, this implies the numerical solution of non linear algebraic systems of equations for the mass flow and pressure computation and the simulation of systems of partial differential equations for the thermal energy propagation part.

Finally, the solution of this problem is hard to be solved with a classical deterministic method. A PSO method is then chosen as a solution algorithm.

Closed loop control

The open loop computed by the solution of (12) can not be directly applied to the real system. Indeed, consumers demands $Q_{dem,n}^c$ are not known in advance, but only predicted. To get a robust behavior of the system, one has to control the system in a closed loop framework. The real control inputs are the supply temperatures of producers. These values are bounded due to physical limitations of steam boilers. Further, consumers take power from the energy supply network if temperatures are sufficiently high (if not, the consumer demand is not fulfilled, but the behavior of the energy supply network remains correct). An important remark is that whatever the control strategy is employed, due to these physical limitations, all temperatures in the network remain in the acceptable range. In conclusion there is no instability danger for the control law, and the receding horizon strategy can be applied, even if a stochastic optimization problem is used without global optimality guarantee.

3.3 Numerical Results

The receding horizon based control law has been applied for the control of the district heating benchmark depicted in figure 2. Tests have been performed for a total time horizon of 24 or 48 hours, with a sampling time of one hour. The prediction horizon for the optimization problem is 12 hours. Thus, as the benchmark represented in figure 2 has 2 producers and 2 valves, the optimization problem is made of $12 \times (2+2) = 48$ optimization variables. The solution of the optimization problem is performed in 120 seconds on a Pentium IV, 2.5 GHz with Matlab 2007, for 50 iterations of the PSO algorithm.

Robustness of the closed loop structure

To validate the control law, a worst case experiment has been performed. It is assumed that all consumer demands are always underestimated by a factor of 10%. This represent a worst case experiment as long as in the real world, load error predictions can partially compensate each other. Tests of the proposed approach have shown that consumers demands are always fulfilled, by using the receding horizon control structure.

Economical benefit of the receding horizon strategy

In the district heating network (figure 2), producer 1 is a cogeneration site. Cogeneration refers to the simultaneous production of electric and thermal powers, leading to high global efficiencies. Briefly speaking, the main goal of the producer is to satisfy the thermal power demand. But he has the opportunity to use the exhaust fumes to produce and to sold electric power. Finally, for the thermal power point of view, the higher the price of sold electricity, the lower the thermal power production costs. The simulation has been performed for different electricity prices, and corresponding total productions over the whole horizon (24 or 48 hours) are given in table 1.

The price $40E/MWh$ corresponds approximately to the price in France from November 1^{st} to March 31^{st}, whereas the null price corresponds to the price from April 1^{st} to October 31^{st}.

Table 1 Numerical Results: total production of producers for different configurations

	Production 1 over		**Production 2 over**	
Electricity price	**24 hours**	**48 hours**	**24 hours**	**48 hours**
$40E/MWh$	$535MWh$	$947MWh$	$537MWh$	$1016MWh$
$0E/MWh$	$541MWh$	$963MWh$	$492MWh$	$950MWh$

Results show that the higher the price, the higher the production of the cogeneration site. The control law uses the interconnection valves to make the extra amount of power to pass from subnetwork 1 to subnetwork 2. Although obvious, the possibility is not used in classical district heating networks: controls laws only use local information, and the interconnections are often viewed as safety means, and are rarely used. The receding horizon law is able to take into account the whole technological string "production - distribution - consumption" and the whole system through the solution of the optimization problem 12. The solution of this problem is made tractable by the use of a stochastic approximated optimization method.

Note that in the future, the price of sold electricity may depend on the electricity market. In such a situation, production costs would be predicted,

and the closed loop structure is also a good trend to get a robust behavior against cost uncertainties.

4 Conclusions and Discussion

In this paper, Particle Swarm Optimization has been used as the core of receding horizon control method. The interest of such methods is that it can solve, or more precisely find a near optimal solution of, hard optimization problems. The drawback of these methods is the lack of guarantee of the actual optimality of the solution. As a results a special attention has to be paid on the stability and the performances of the closed loop.

The use of such methods is of great interest for dictrict heating networks. To decrease global costs, one has to capture the whole system in a close loop framework. The use of stochastic algorithms is tractable for that purpose. Furthermore, in the case of the control of district heating networks, the system is always stable, whatever the control strategy. Thus the use of a stochastic optimization algorithm for receding horizon control can only enhance the performance that could be achieved by a classical control law.

Further studies have to be done so as to deeply understand the pros and cons of the use of such methods and to establish some robustness properties of the corresponding control laws.

References

1. Sandou, G., Olaru, S.: Ant colony and genetic algorithm for constrained predictive control of power systems. In: Bemporad, A., Bicchi, A., Buttazzo, G. (eds.) HSCC 2007. LNCS, vol. 4416, pp. 501–514. Springer, Heidelberg (2007)
2. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
3. Shi, Y., Eberhart, R.C.: Parameter selection in particle swarm optimization. In: Proceedings of the Seventh Annual Conference on Evolutionary Programming, pp. 591–600 (1998)
4. Eberhart, R.C., Shi, Y.: Comparing inertia weigthts and constriction factors in particle swarm optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation (2000)
5. Kennedy, J., Clerc, M.: Standard PSO (2006), http://www.particleswarm.info/Standard-PSO-2006.c
6. Kennedy, J.: Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC 1999), pp. 1931–1938 (1999)
7. Sandou, G., Font, S., Tebbani, S., Hiret, A., Mondon, C.: Global modelling and simulation of a district heating network. In: Proceedings of the 9th International Symposium on District Heating and Cooling (2004)

Explicit Receding Horizon Control of Automobiles with Continuously Variable Transmissions

Takeshi Hatanaka, Teruki Yamada, Masayuki Fujita, Shigeru Morimoto, and Masayuki Okamoto

Abstract. This paper presents a novel systematic control scheme of continuously variable transmissions (CVTs) of automobiles based on explicit receding horizon control (ERHC). We use the ERHC controller as a high-level controller in the automotive control design hierarchy i.e. it determines reference signals of the pulley ratio at each time instant, and inputs it to a locally compensated system. We choose as a prediction model a constrained piecewise affine system whose dynamics switches depending on the presence or absence of fuel injection. Then, we formulate a constrained finite time optimal control problem, and compute its explicit solution via Multi-Parametric Toolbox. The effectiveness of our control scheme is finally verified by a simulator developed by Honda R&D.

Keywords: Continuously Variable Transmissions, Explicit Receding Horizon Control, Multi-Parametric Toolbox.

1 Introduction

In recent years, stimulated by increasing applications of electronics technologies to automotive systems, demands for the automobiles have become diverse. For example, environmental conservation, comfortable ride and safety have been required. Especially, reduction of fuel consumption is addressed as one of the highest priority issue in order to prevent environmental damage. To meet this requirement, continuously variable transmissions (CVTs) have

Takeshi Hatanaka, Teruki Yamada, and Masayuki Fujita
Tokyo Institute of Technology, 2-12-1 Ookayama, Meguroku, Tokyo 152-8552, Japan
e-mail: hatanaka@ctrl.titech.ac.jp

Shigeru Morimoto and Masayuki Okamoto
Honda R&D Co.,Ltd. Automobile R&D Center, 4630 Shimotakanezawa, Haga-machi, Haga-gun, Tochigi, 321-3223 Japan

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 561–569.
springerlink.com

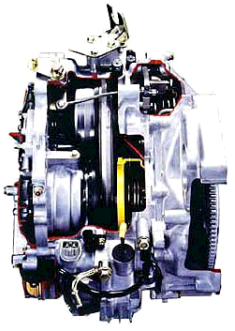


Fig. 1 Belt type CVTs

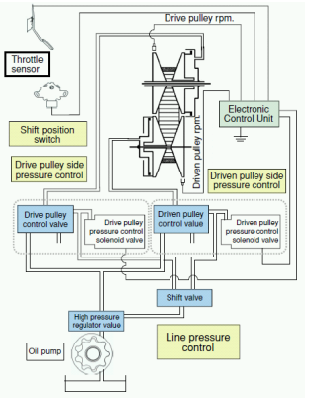


Fig. 2 Structure of CVTs

gained increasing interests. It enables one to improve fuel efficiency and driving performances simultaneously due to its continuously variable pulley ratio characteristics, and has already been installed in most of the recent stock cars [1, 2, 3] in Japan. Though the diffusion of CVTs is currently limited in other countries due to technological problems, differences in driving environment and some social reasons, the rate of automobiles with CVTs tends to increase also in the US because of their technological progress. A belt type CVT handled in this paper is illustrated in Figures 1 and 2.

The automotive industry has employed a so-called map controller to determine the pulley ratio of the CVTs. This controller is designed based on heuristics and a large quantity of empirical data without taking account of the dynamical characteristics of the automobiles explicitly [3]. However, since the controller grows more and more complex, its retuning requires a high degree of professional skills and knowledge on the controller and automotive system. The automotive industry thus desires a new systematic control scheme, and this paper addresses this requirement, where we employ receding horizon control (RHC) as a systematic design scheme.

RHC is a control methodology in which an optimal control problem is solved over a finite horizon, the first one of the computed control moves is implemented and then the optimal control problem is newly solved at the next step with the horizon shifted forward by one time instant [4]. This control method can handle various types of control systems within a unified framework, and has been employed in a variety of industries. This paper especially focuses on the RHC for hybrid systems [4, 5]. Some relevant works applying this control strategy to the automotive systems have already been reported in [6, 7, 8]. In these works, either of the following implementation methods is employed: (i) Model Predictive Control and (ii) Explicit Receding Horizon Control (ERHC). The former solves a mixed integer programming problem on-line at each sampling step, which is in general prohibitive on automotive

control hardware [6]. The latter computes off-line an explicit piecewise affine (PWA) form of the optimal control law and performs only a simple evaluation of the PWA function for a given state at each time instant.

The objective of this paper is to present a novel systematic control scheme of CVTs based on the ERHC. The present scheme has the following benefits compared with the current map controller. (i) Controller retuning becomes easier. (ii) Driving performances are improved. (iii) Dynamical characteristics are taken into account. It should be noted that the present controller is developed within much shorter period and smaller human resources than the current map controller. In contrast to the previous works on the CVT controller design [1, 2, 3], where local controllers stabilizing the CVTs are presented, our control scheme belongs to a higher layer in the automotive control system design hierarchy. Namely, the present controller determines the reference of the pulley ratio and inputs it to the automotive system compensated *a priori*. This scheme brings an advantage that several minor effects on the total automotive system and the corresponding variables can be neglected and each subsystem be simplified. This enables us to employ as a prediction model in the ERHC the total automotive system including all the subsystems.

2 Modeling

This section describes the automotive system model utilized for control design under the following assumptions: (i) It drives on a straight flat road and does not stop. (ii) Drivers do not depress the accelerator and the brake simultaneously. (iii) A fuel cutoff works and fuel injection never occurs whenever the brake is depressed (This situation is denoted by FC = ON) and, conversely, a fuel cutoff never works whenever the accelerator pedal is pressed (FC = OFF). It is in general desirable due to computational and structural complexity of the ERHC controller to find a simple model while capturing the main behavior of the automotive system. However, it is difficult to systematically determine how much simplification is allowed. We thus choose the following model through a trial and error process via numerical experiments.

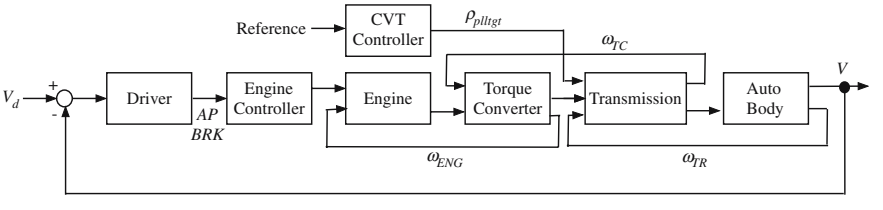


Fig. 3 Configuration of the automotive system

In this paper, we consider the total automotive system illustrated in Figure 3, where AP is the accelerator opening percent, BRK is the brake pedal force, ω_{ENG}[rpm] is the engine speed, ω_{TC}[rpm] is the turbine speed of torque converter, and ω_{TR}[rpm] is the tire speed, ρ_{plltgt} is the pulley ration target V[km/h] is the vehicle speed, and V_d is the reference vehicle speed of a driver. It should be now noted that the system dynamics significantly changes depending on the presence or absence of fuel injection. In this paper, we separately describe the subsystem models in both situations, integrate them and gain a hybrid system model as the automotive system model. Only the overview is described below. The engine and engine controller do not have a strong dynamical characteristics and we first describe them as a static map. Then, we employ linear approximation of the input-output relation in order to suit these models to RHC design. The torque converter, transmission and auto body models are approximately described based on physical modeling.

Integrating the above subsystems yields a total automotive model. Due to complexity of the model, all the equations cannot be shown in this paper, the state equation takes the form of

$$\dot{x_p} = f_{OFF}(x_p) + B_{OFF} \begin{bmatrix} AP \\ \rho_{plltgt} \end{bmatrix}, \quad \text{if } FC = OFF, \tag{1}$$

$$\dot{x_p} = f_{ON}(x_p) + B_{ON} \begin{bmatrix} BRK \\ \rho_{plltgt} \end{bmatrix}, \quad \text{if } FC = ON, \tag{2}$$

$$x_p = \begin{bmatrix} \dot{V} & \dot{\omega}_{ENG} & \dot{\omega}_{TC} & \dot{\rho}_{pll} \end{bmatrix}^\top,$$

where ρ_{pll} is the pulley ratio and f_{OFF} and f_{ON} are nonlinear functions of x_p. The above automotive model (1), (2) is a hybrid system whose dynamics switches depending on the state of FC. However, this model is intractable for RHC design as it is due to its nonlinearity and obscuration of the switching condition. In the following, we moreover approximately establish a discrete-time constrained PWA system model.

We first linearize both of the systems (1) and (2). For this purpose, we have to determine the operating equilibrium state and input. Here, we let the equilibrium state and input be $(20, 876.6197, 876.3416, 0.6)$ and $(0.1768, 0.6)$ respectively for both systems, which is empirically determined. We then discretize the resulting continuous-time linear time invariant system model with zeroth-order hold and sampling period 1[s], which is longer than standard transmission control schemes. It is possible to take such a long period since we consider a high level controller determining not ρ_{pll} but ρ_{plltgt}.

We next derive a switching condition. We see from real data that the fuel cutoff works when ω_{TC} exceeds ω_{ENG}, which means that the engine is rotated by the wheel. Namely, the switching condition is represented by a linear inequality between the state variables ω_{TC} and ω_{ENG}.

We thus get the discrete-time PWA system model

$$x(k+1) = \begin{bmatrix} 0.9502 & 0.0080 & 0.0029 & 0.2244 \\ 4.6843 & 0.0428 & 0.0155 & 3.8368 \\ 4.6856 & 0.0428 & 0.0155 & 3.8318 \\ 0 & 0 & 0 & 0.0183 \end{bmatrix} x(k) + \begin{bmatrix} 1.0056 & -0.8543 \\ 4.6980 & 146.1789 \\ 4.6224 & 146.2213 \\ 0 & 0.9817 \end{bmatrix} u(k)$$

$$\text{if } \ \delta\omega_{ENG}(k) \geq \delta\omega_{TC}(k)$$

$$x(k+1) = \begin{bmatrix} 0.9298 & 0.0078 & 0.0028 & 0.0541 \\ 4.2661 & 0.036 & 0.0130 & 3.0594 \\ 4.2661 & 0.036 & 0.0130 & 3.0505 \\ 0 & 0 & 0 & 0.0183 \end{bmatrix} x(k) + \begin{bmatrix} 0 & -1.3755 \\ 0 & 143.8176 \\ 0 & 143.8243 \\ 0 & 0.9817 \end{bmatrix} u(k)$$

$$+ \begin{bmatrix} -0.1776 \\ -0.8145 \\ -0.8149 \\ 0 \end{bmatrix} BRK, \ \text{if } \ \delta\omega_{ENG}(k) \leq \delta\omega_{TC}(k), \tag{3}$$

$$y(k) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x(k), \tag{4}$$

where the state and input vectors are $x(k) = \left[\delta V \ \delta\omega_{ENG} \ \delta\omega_{TC} \ \delta\rho_{pll}\right]^T$ and $u(k) = \left[\delta AP \ \delta\rho_{plltgt}\right]^T$. The notation δ represents the deviation of each variable from its equilibrium point. We choose $\delta\rho_{pll}$ and $\delta\omega_{ENG}$ as the control output y.

We finally describe state and input constraints. The automotive system has the following constraints due to mechanical limitations and specifications for a comfortable ride, where $\Delta u(k) = u(k) - u(k+1)$.

$$\begin{bmatrix} -0.1768 \\ -0.150 \end{bmatrix} \leq u(k) \leq \begin{bmatrix} 1.8232 \\ 1.85 \end{bmatrix}, \ \begin{bmatrix} -5 \\ -1 \end{bmatrix} \leq \Delta u(k) \leq \begin{bmatrix} 5 \\ 1 \end{bmatrix}, \tag{5}$$

$$\begin{bmatrix} -20.5 \\ -104.7198 \\ -104.7198 \\ -0.15 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 50.5 \\ 628.3185 \\ 628.3185 \\ 1.85 \end{bmatrix}, \ \begin{bmatrix} -0.15 \\ -1000 \end{bmatrix} \leq y(k) \leq \begin{bmatrix} 1.85 \\ 6000 \end{bmatrix}. \tag{6}$$

Remark 1. It is possible to obtain a more accurate model by setting several operating points and corresponding switching conditions. However, we do not adopt this strategy since it does not improve control performances despite of the increase of the computational and structural complexity in design stage.

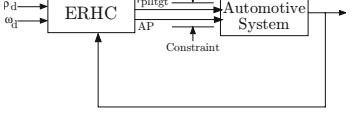


Fig. 4 Prediction Model

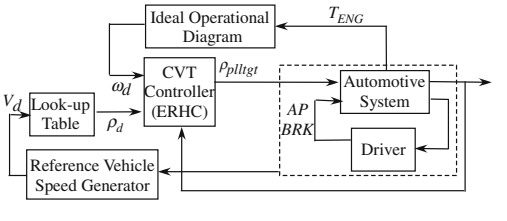


Fig. 5 Total Control System

3 Controller Design

In this section, we design an ERHC for the automotive system model described in the previous section. As shown in the prediction model of Figure 4, the ERHC controller determines ρ_{plltgt} and AP so that the control outputs track to external references. The resulting ρ_{plltgt} is inputted to the transmission. However, AP is not utilized for control since it is originally an uncontrollable human operated force. This implies that the ERHC controller assumes in prediction that the driver takes near optimal actions. Though including the driver into the plant model is an option, it is a highly difficult problem. The practical total control system is depicted in Figure 5. Note that we hereafter denote the real time by t (the time step in the prediction horizon is denoted by k).

3.1 ERHC Design

Let us consider the constrained finite-time optimal control (CFTOC) problem

$$J = \min_{u(0),\ \cdots,\ u(H-1)} \sum_{k=0}^{H-1} \|R\Delta u(k)\|_p + \|Q_y(y(k) - y_d)\|_p \text{ subject to } (3) - (6),$$

where $H = 3,\ p = 2$ and $Q_y = \mathrm{diag}(250, 1),\ R = \mathrm{diag}(10, 1000)$.

Since the reference signal y_d is time varying, the explicit solution of the above CFTOC problem is defined over the augmented state space composed of $\xi(k) = (x(k), u(k-1), y_d(k))$. Namely, the ERHC controller takes the form of $u(t) = F_i\xi(t) + f_i,\ \ \text{if } \xi(t) \in \mathcal{P}_i$ ([9] and the references therein). It should be now noted that all the state variables can be measured by sensors. The polyhedral partition of the controller is illustrated in Figure 6, which is a cross-section with the other state variables fixed at 0. This controller is computed by an Intel(R) Core(TM)2 CPU 6400 (2.13GHz) machine running MATLAB(R) 6.5.1 and MPT 2.6.2 [9].

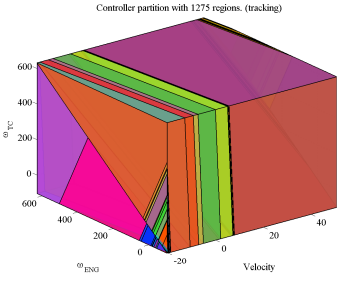
The parameters H, Q_y, R are tuned in numerical experiments until a desired performance is achieved. As reported in [6], by increasing the prediction horizon H, the control performance improves, but at the same time the complexity of the resulting piecewise affine controller increases dramatically and no controller is computed in the worst case. However, short prediction periods result in poor control performances. This is one reason why we choose a long sampling period (1[s]), which allows a prediction over a long period without increasing the complexity. Though this scheme decreases the degree of freedom of control actions, a satisfactory performance is achieved at least for numerical experiments. Though further tuning might be necessary in the stage of actual vehicle tests, the basic strategy of using a long sampling period would be employed. The norm $p = 2$ in the cost function is chosen because of poor control performances of the ERHC controllers with $p = 1$ and $p = \infty$ in numerical experiments.

3.2 Reference Signal Generator

This subsection designs the mechanism generating the reference $y_d = (\rho_d, \omega_d)$.

We let ρ_d be determined by the current reference of vehicle speed according to the graph in Figure 7. Note that V_d is estimated by another mechanism from the current vehicle speed and drivers' operations but we omit detailed explanations on this mechanism since it is beyond the scope of this paper.

We next design the generator of ω_d. Now, the optimality of the engine operation in terms of fuel consumption is in general characterized by the relation between the engine speed and engine torque. The ideal operational diagram can be drawn on the engine speed – engine torque graph [3]. Since the engine torque is measurable at each real time t, the ideal engine speed can be determined by this diagram. We thus employ it as the reference engine speed $\omega_d(t)$.



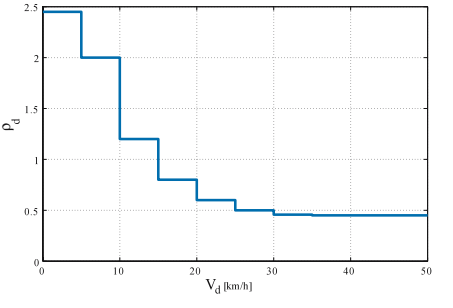


Fig. 6 Polyhedral partition of ERHC **Fig. 7** Relation between V_d and ρ_d controller

4 Simulator Based Verification

In this section, we demonstrate the effectiveness of the present controller through numerical experiments. In this verification, the vehicle motion is simulated by a simulator developed by Honda R & D. We use as the driver model a PID controller whose input is the error of the reference speed and the current speed $(V_d - V)$ and output is accelerator force AP or brake force BRK. The switch of AP and BRK occurs according to the sign of $V_d - V$. Note that for simplicity we employ the following V_d without using the reference vehicle speed generator in Figure 5. The original reference vehicle speed is produced based on the Japanese 10.15 driving mode [3], and the driver model and the ERHC controller uses its filtered signal as the reference V_d.

In the following, we compare the present ERHC with the map controller in terms of the tracking performance. Now let $\phi^{(i)}_{\mathrm{ERHC}}$ and $\phi^{(i)}_{\mathrm{map}}$ be the numerical integration values of the absolute values of the tracking errors for our controller and the map controller respectively, where $i = 1$ denotes that of ω_{ENG} and $i = 2$ that of ρ_{pll}. Then, we see from the values of

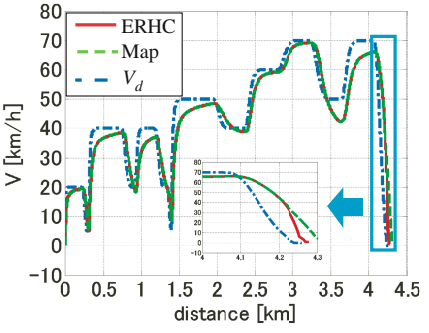


Fig. 8 Vehicle velocity

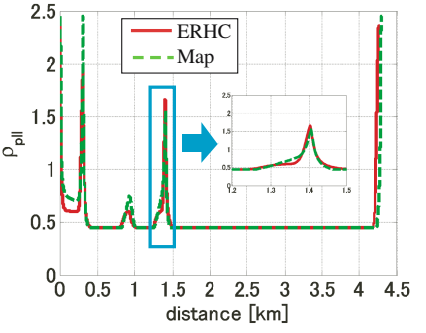


Fig. 9 Pulley-ratio

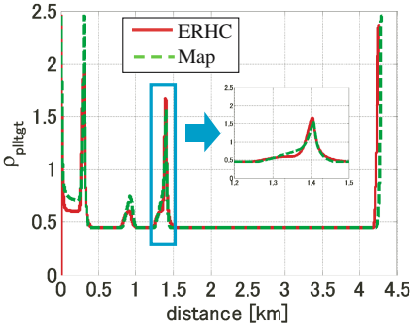


Fig. 10 Pulley-ratio target

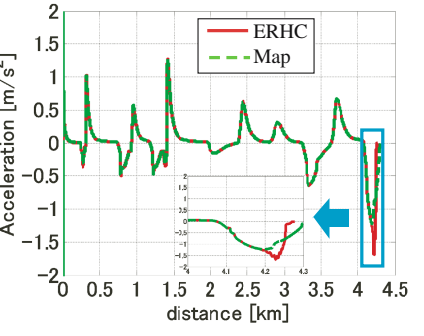


Fig. 11 Vehicle acceleration

$1 - \phi^{(i)}_{\mathrm{ERHC}}/\phi^{(i)}_{\mathrm{map}}$, $i \in \{1, 2\}$ that our controller improves the tracking performance by 3.4781[%] in terms of ω_{ENG} and by 9.7174[%] in terms of ρ_{pll}. It should be noted that both controllers satisfy the specified constraints (5), (6). Though this section shows the effectiveness only for the mode produced based on the 10.15 mode, almost the same tendency is confirmed for other some driving modes.

The above figures also illustrate that our control scheme successfully works despite of the uncontrollable drivers' operations AP and BRK. It works well even for some driving modes and driver models. However, we essentially have to demonstrate its validity through actual vehicle tests, since all the simulations assume some kind of driver models but real drivers do not always behave like the models. Performing experimental tests is one of the future works.

We see from the above results that our controller achieves satisfactory responses. In addition, the controller is designed within much shorter period and smaller human resources than the current map controller.

5 Conclusions

In this paper, we have proposed a novel systematic control scheme of CVTs of automotive systems based on ERHC. Unlike previous works on CVT control, we have investigated a high-level control design problem. By using the high-level control scheme, we could obtain a simple total automotive system model and employ a long sampling period, which is very helpful in reducing the computational and structural complexity of ERHC controller. We have shown our modeling and control procedures, and the effectiveness of our control scheme has been demonstrated through numerical experiments with a simulator developed by Honda R & D.

A future direction of this study is to perform actual vehicle tests. For this purpose, we have to address the following issues: giving physical interpretations of control actions of our controller, simplification of the controller via post-processing, and writing a program which is implementable by electronic control units of automobiles.

Acknowledgement. The authors would like to thank Mr. Tatsuya Miyano sincerely for his invaluable help in improving the quality of this paper.

References

1. Setlur, P., Wagner, J.R., Dawson, D.M., Samuels, B.: Nonlinear Control of Continuously Variable Transmission (CVT) for Hybrid Vehicle Powertrains. In: Proc. of the Amer. Contr. Conf., pp. 1304–1309 (2001)
2. Sakaguchi, S., Kimura, E., Yamamoto, K.: Development of an Engine-CVT Integrated Control System. Proc. of JSAE (9838381) (1998)
3. Togai, K., Koso, M.: Dynamic Scheduling Control for Engine and Gearshifts, Consolidation of Fuel-Economy Optimization and Reserve Power. Mitsubishi Motors Technical Review (18) (2006)
4. Morari, M., Baric, M.: Recent Developments in The Control of Constrained Hybrid Systems. Computers and Chemical Engineering 30(10–12), 1619–1631
5. Christophersen, F.J.: Optimal Control of Constrained Piecewise Affine Systems. LNCIS, vol. 359. Springer, Heidelberg (2007)
6. Borrelli, F., Bemporad, A., Fodor, M., Hrocat, D.: An MPC/Hybrid System Approach to Traction Control. IEEE Trans. on Contr. Sys. Tech. 14(3), 541–552 (2006)
7. Giorgetti, N., Ripaccioli, G., Bemporad, A., Kolmanovsky, I., Hrovat, D.: Hybrid Model Predictive Control of Direct Injection Stratified Change Engines. IEEE/ASME Trans. on Mech. 11(5), 499–506 (2006)
8. Falcone, P., Borrelli, F., Asgari, J., Tseng, H.E., Hrovat, D.: Predictive Active Streering Control for Autonomous Vehicle Systems. IEEE Trans. on Contr. Sys. Tech. 15(3) (2007)
9. Kvasnica, M., Grieder, P., Baotic, M., Christophersen, F.J.: Multi-Parametric Toolbox (MPT), http://control.ee.ethz.ch/~mpt/

Author Index

Lecture Notes in Control and Information Sciences

Edited by M. Thoma, F. Allgöwer, M. Morari

Further volumes of this series can be found on our homepage:
springer.com

Vol. 384: Magni, L.; Raimondo, D.M.; Allgöwer, F. (Eds.):
Nonlinear Model Predictive Control
572 p. 2009 [978-3-642-01093-4]

Vol. 383: Sobhani-Tehrani E.; Khorasani K.;
Fault Diagnosis of Nonlinear Systems Using a Hybrid Approach
360 p. 2009 [978-0-387-92906-4]

Vol. 382: Bartoszewicz A.; Nowacka-Leverton A.;
Time-Varying Sliding Modes for Second and Third Order Systems
192 p. 2009 [978-3-540-92216-2]

Vol. 381: Hirsch M.J.; Commander C.W.; Pardalos P.M.; Murphey R. (Eds.)
Optimization and Cooperative Control Strategies: Proceedings of the 8th International Conference on Cooperative Control and Optimization
459 p. 2009 [978-3-540-88062-2]

Vol. 380: Basin M.
New Trends in Optimal Filtering and Control for Polynomial and Time-Delay Systems
206 p. 2008 [978-3-540-70802-5]

Vol. 379: Mellodge P.; Kachroo P.;
Model Abstraction in Dynamical Systems: Application to Mobile Robot Control
116 p. 2008 [978-3-540-70792-9]

Vol. 378: Femat R.; Solis-Perales G.;
Robust Synchronization of Chaotic Systems Via Feedback
199 p. 2008 [978-3-540-69306-2]

Vol. 377: Patan K.
Artificial Neural Networks for the Modelling and Fault Diagnosis of Technical Processes
206 p. 2008 [978-3-540-79871-2]

Vol. 376: Hasegawa Y.
Approximate and Noisy Realization of Discrete-Time Dynamical Systems
245 p. 2008 [978-3-540-79433-2]

Vol. 375: Bartolini G.; Fridman L.; Pisano A.; Usai E. (Eds.)
Modern Sliding Mode Control Theory
465 p. 2008 [978-3-540-79015-0]

Vol. 374: Huang B.; Kadali R.
Dynamic Modeling, Predictive Control and Performance Monitoring
240 p. 2008 [978-1-84800-232-6]

Vol. 373: Wang Q.-G.; Ye Z.; Cai W.-J.; Hang C.-C.
PID Control for Multivariable Processes
264 p. 2008 [978-3-540-78481-4]

Vol. 372: Zhou J.; Wen C.
Adaptive Backstepping Control of Uncertain Systems
241 p. 2008 [978-3-540-77806-6]

Vol. 371: Blondel V.D.; Boyd S.P.; Kimura H. (Eds.)
Recent Advances in Learning and Control
279 p. 2008 [978-1-84800-154-1]

Vol. 370: Lee S.; Suh I.H.; Kim M.S. (Eds.)
Recent Progress in Robotics: Viable Robotic Service to Human
410 p. 2008 [978-3-540-76728-2]

Vol. 369: Hirsch M.J.; Pardalos P.M.; Murphey R.; Grundel D.
Advances in Cooperative Control and Optimization
423 p. 2007 [978-3-540-74354-5]

Vol. 368: Chee F.; Fernando T.
Closed-Loop Control of Blood Glucose
157 p. 2007 [978-3-540-74030-8]

Vol. 367: Turner M.C.; Bates D.G. (Eds.)
Mathematical Methods for Robust and Nonlinear Control
444 p. 2007 [978-1-84800-024-7]

Vol. 366: Bullo F.; Fujimoto K. (Eds.)
Lagrangian and Hamiltonian Methods for Nonlinear Control 2006
398 p. 2007 [978-3-540-73889-3]

Vol. 365: Bates D.; Hagström M. (Eds.)
Nonlinear Analysis and Synthesis Techniques for Aircraft Control
360 p. 2007 [978-3-540-73718-6]

Vol. 364: Chiuso A.; Ferrante A.; Pinzoni S. (Eds.)
Modeling, Estimation and Control
356 p. 2007 [978-3-540-73569-4]

Vol. 363: Besançon G. (Ed.)
Nonlinear Observers and Applications
224 p. 2007 [978-3-540-73502-1]

Vol. 362: Tarn T.-J.; Chen S.-B.; Zhou C. (Eds.)
Robotic Welding, Intelligence and Automation
562 p. 2007 [978-3-540-73373-7]

Vol. 361: Méndez-Acosta H.O.; Femat R.; González-Álvarez V. (Eds.):
Selected Topics in Dynamics and Control of Chemical and Biological Processes
320 p. 2007 [978-3-540-73187-0]

Vol. 360: Kozlowski K. (Ed.)
Robot Motion and Control 2007
452 p. 2007 [978-1-84628-973-6]

Vol. 359: Christophersen F.J.
Optimal Control of Constrained Piecewise Affine Systems
190 p. 2007 [978-3-540-72700-2]

Vol. 358: Findeisen R.; Allgöwer F.; Biegler L.T. (Eds.): Assessment and Future Directions of Nonlinear Model Predictive Control
642 p. 2007 [978-3-540-72698-2]

Vol. 357: Queinnec I.; Tarbouriech S.; Garcia G.; Niculescu S.-I. (Eds.):
Biology and Control Theory: Current Challenges
589 p. 2007 [978-3-540-71987-8]

Vol. 356: Karatkevich A.:
Dynamic Analysis of Petri Net-Based Discrete Systems
166 p. 2007 [978-3-540-71464-4]

Vol. 355: Zhang H.; Xie L.:
Control and Estimation of Systems with Input/Output Delays
213 p. 2007 [978-3-540-71118-6]

Vol. 354: Witczak M.:
Modelling and Estimation Strategies for Fault Diagnosis of Non-Linear Systems
215 p. 2007 [978-3-540-71114-8]

Vol. 353: Bonivento C.; Isidori A.; Marconi L.; Rossi C. (Eds.)
Advances in Control Theory and Applications
305 p. 2007 [978-3-540-70700-4]

Vol. 352: Chiasson, J.; Loiseau, J.J. (Eds.)
Applications of Time Delay Systems
358 p. 2007 [978-3-540-49555-0]

Vol. 351: Lin, C.; Wang, Q.-G.; Lee, T.H., He, Y.
LMI Approach to Analysis and Control of Takagi-Sugeno Fuzzy Systems with Time Delay
204 p. 2007 [978-3-540-49552-9]

Vol. 350: Bandyopadhyay, B.; Manjunath, T.C.; Umapathy, M.
Modeling, Control and Implementation of Smart Structures 250 p. 2007 [978-3-540-48393-9]

Vol. 349: Rogers, E.T.A.; Galkowski, K.; Owens, D.H.
Control Systems Theory and Applications for Linear Repetitive Processes
482 p. 2007 [978-3-540-42663-9]

Vol. 347: Assawinchaichote, W.; Nguang, K.S.; Shi P.
Fuzzy Control and Filter Design for Uncertain Fuzzy Systems
188 p. 2006 [978-3-540-37011-6]

Vol. 346: Tarbouriech, S.; Garcia, G.; Glattfelder, A.H. (Eds.)
Advanced Strategies in Control Systems with Input and Output Constraints
480 p. 2006 [978-3-540-37009-3]

Vol. 345: Huang, D.-S.; Li, K.; Irwin, G.W. (Eds.)
Intelligent Computing in Signal Processing and Pattern Recognition
1179 p. 2006 [978-3-540-37257-8]

Vol. 344: Huang, D.-S.; Li, K.; Irwin, G.W. (Eds.)
Intelligent Control and Automation
1121 p. 2006 [978-3-540-37255-4]

Vol. 341: Commault, C.; Marchand, N. (Eds.)
Positive Systems
448 p. 2006 [978-3-540-34771-2]

Vol. 340: Diehl, M.; Mombaur, K. (Eds.)
Fast Motions in Biomechanics and Robotics
500 p. 2006 [978-3-540-36118-3]

Vol. 339: Alamir, M.
Stabilization of Nonlinear Systems Using Receding-horizon Control Schemes
325 p. 2006 [978-1-84628-470-0]

Vol. 338: Tokarzewski, J.
Finite Zeros in Discrete Time Control Systems
325 p. 2006 [978-3-540-33464-4]

Vol. 337: Blom, H.; Lygeros, J. (Eds.)
Stochastic Hybrid Systems
395 p. 2006 [978-3-540-33466-8]

Vol. 336: Pettersen, K.Y.; Gravdahl, J.T.; Nijmeijer, H. (Eds.)
Group Coordination and Cooperative Control
310 p. 2006 [978-3-540-33468-2]

Vol. 335: Kozłowski, K. (Ed.)
Robot Motion and Control
424 p. 2006 [978-1-84628-404-5]

Vol. 334: Edwards, C.; Fossas Colet, E.; Fridman, L. (Eds.)
Advances in Variable Structure and Sliding Mode Control
504 p. 2006 [978-3-540-32800-1]

Vol. 333: Banavar, R.N.; Sankaranarayanan, V.
Switched Finite Time Control of a Class of Underactuated Systems
99 p. 2006 [978-3-540-32799-8]