Model Predictive Control

edited by
Tao ZHENG

SCIYO

Model Predictive Control
Edited by Tao ZHENG

Published by Sciyo
Janeza Trdine 9, 51000 Rijeka, Croatia

Publishing Process Manager Jelena Marusic
Technical Editor Sonja Mujacic
Cover Designer Martina Sirotic
Image Copyright Richard Griffin, 2010. Used under license from Shutterstock.com

First published September 2010
Printed in India

A free online edition of this book is available at **www.sciyo.com**
Additional hard copies can be obtained from **publication@sciyo.com**

Model Predictive Control, Edited by Tao ZHENG
p. cm.
ISBN 978-953-307-102-2

Contents

Preface

Since Model Predictive Heuristic Control (MPHC), the earliest algorithm of Model Predictive Control (MPC), was proposed by French engineer Richalet and his colleagues in 1978, the explicit background of industrial application has made MPC develop rapidly to satisfy the increasing request from modern industry. Different from many other control algorithms, the research history of MPC is originated from application and then expanded to theoretical field, while ordinary control algorithms often has applications after sufficient theoretical research.

Nowadays, MPC is not just the name of one or some specific computer control algorithms, but the name of a specific thought in controller design, from which many kinds of computer control algorithms can be derived for different systems, linear or nonlinear, continuous or discrete, integrated or distributed. The basic characters of the thought of MPC can be summarized as the model used for prediction, the online optimization based on prediction and the feedback compensation for model mismatch, while there is no special demands on the form of model, the computational tool for online optimization and the form of feedback compensation.

After three decades' developing, the MPC theory for linear systems is now comparatively mature, so its applications can be found in almost every domain in modern engineering. While, MPC with robustness and MPC for nonlinear systems are still problems for scientists and engineers. Many efforts have been made to solve them, though there are some constructive results, they will remain as the focuses of MPC research for a period in the future.

In first part of this book, to present the recent theoretical improvements of MPC, Chapter 1 will introduce the Robust Model Predictive Control and Chapter 2 to Chapter 5 will introduce some typical methods to establish Nonlinear Model Predictive Control, with more complexity, MPC for multi-variable nonlinear systems will be proposed in Chapter 6 and Chapter 7.

To give the readers an overview of MPC's applications today, in second part of the book, Chapter 8 to Chapter 13 will introduce some successful examples, from plasma stabilization system to satellite system, from linear system to nonlinear system. They can not only help the readers understand the characters of MPC, but also give them the guidance for how to use MPC to solve practical problems.

Authors of this book truly want to it to be helpful for researchers and students, who are concerned about MPC, and further discussions on the contents of this book are warmly welcome.

Finally, thanks to SCIYO and its officers for their efforts in the process of edition and publication, and thanks to all the people who have made contributes to this book.

Editor

Tao ZHENG
University of Science and Technology of China

1

Robust Model Predictive Control Design

Vojtech Veselý and Danica Rosinová
Institute for Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 81219 Bratislava Slovak Republic

1. Introduction

Model predictive control (MPC) has attracted notable attention in control of dynamic systems and has gained the important role in control practice. The idea of MPC can be summarized as follows, (Camacho & Bordons, 2004), (Maciejovski, 2002), (Rossiter, 2003) :

- Predict the future behavior of the process state/output over the finite time horizon.
- Compute the future input signals on line at each step by minimizing a cost function under inequality constraints on the manipulated (control) and/or controlled variables.
- Apply on the controlled plant only the first of vector control variable and repeat the previous step with new measured input/state/output variables.

Therefore, the presence of the plant model is a necessary condition for the development of the predictive control. The success of MPC depends on the degree of precision of the plant model. In practice, modelling real plants inherently includes uncertainties that have to be considered in control design, that is control design procedure has to guarantee robustness properties such as stability and performance of closed-loop system in the whole uncertainty domain. Two typical description of uncertainty, state space polytope and bounded unstructured uncertainty are extensively considered in the field of robust model predictive control. Most of the existing techniques for robust MPC assume measurable state, and apply plant state feedback or when the state estimator is utilized, output feedback is applied. Thus, the present state of robustness problem in MPC can be summarized as follows:
Analysis of robustness properties of MPC.
(Zafiriou & Marchal, 1991) have used the contraction properties of MPC to develop necessary-sufficient conditions for robust stability of MPC with input and output constraints for SISO systems and impulse response model. (Polak & Yang, 1993) have analyzed robust stability of MPC using a contraction constraint on the state.
MPC with explicit uncertainty description.
(Zheng & Morari, 1993), have presented robust MPC schemes for SISO FIR plants, given uncertainty bounds on the impulse response coefficients. Some MPC consider additive type of uncertainty, (delaPena et al., 2005) or parametric (structured) type uncertainty using CARIMA model and linear matrix inequality, (Bouzouita et al., 2007). In (Lovas et al., 2007), for open-loop stable systems having input constraints the unstructured uncertainty is used. The robust stability can be established by choosing a large value for the control input weighting matrix R in the cost function. The authors proposed a new less conservative stability test for determining a sufficiently large control penalty R using bilinear matrix inequality (BMI). In (Casavola

et al., 2004), robust constrained predictive control of uncertain norm-bounded linear systems is studied. The other technique- constrained tightening to design of robust MPC have been proposed in (Kuwata et al., 2007). The above approaches are based on the idea of increasing the robustness of the controller by tightening the constraints on the predicted states.
The mixed H_2/H_∞ control approach to design of MPC has been proposed by (Orukpe et al., 2007) .
Robust constrained MPC using linear matrix inequality (LMI) has been proposed by (Kothare et al., 1996), where the polytopic model or structured feedback uncertainty model has been used. The main idea of (Kothare et al., 1996) is the use of infinite horizon control laws which guarantee robust stability for state feedback. In (Ding et al., 2008) output feedback robust MPC for systems with both polytopic and bounded uncertainty with input/state constraints is presented. Off-line, it calculates a sequence of output feedback laws based on the state estimators, by solving LMI optimization problem. On-line, at each sampling time, it chooses an appropriate output feedback law from this sequence. Robust MPC controller design with one step ahead prediction is proposed in (Veselý & Rosinová , 2009). The survey of optimal and robust MPC design can be consulted in (Mayne et al., 2000). Some interesting results for nonlinear MPC are given in (Janík et al., 2008).
In MPC approach generally, control algorithm requires solving constrained optimization problem on-line (in each sampling period). Therefore on-line computation burden is significant and limits practical applicability of such algorithms to processes with relatively slow dynamics. In this chapter, a new MPC scheme for an uncertain polytopic system with constrained control is developed using model structure introduced in (Veselý et al., 2010). The main contribution of the first part of this chapter is that all the time demanding computations of output feedback gain matrices are realized off-line (for constrained control and unconstrained control cases). The actual value of control variable is obtained through simple on-line computation of scalar parameter and respective convex combination of already computed matrix gains. The developed control design scheme employs quadratic Lyapunov stability to guarantee the robustness and performance (guaranteed cost) over the whole uncertainty domain.
The first part of the chapter is organized as follows. A problem formulation and preliminaries on a predictive output/state model as a polytopic system are given in the next section. In Section 1.2, the approach of robust output feedback predictive controller design using linear matrix inequality is presented. In Section 1.3, the input constraints are applied to LMI feasible solution. Two examples illustrate the effectiveness of the proposed method in the Section 1.4. The second part of this chapter addresses the problem of designing a robust parameter dependent quadratically stabilizing output/state feedback model predictive control for linear polytopic systems without constraints using original sequential approach. For the closed-loop uncertain system the design procedure ensures stability, robustness properties and guaranteed cost. Finally, conclusions on the obtained results are given.
Hereafter, the following notational conventions will be adopted: given a symmetric matrix $P = P^T \in R^{n\times n}$, the inequality $P > 0 (P \geq 0)$ denotes matrix positive definiteness (semidefiniteness). Given two symmetric matrices P, Q, the inequality $P > Q$ indicates that $P - Q > 0$. The notation $x(t + k)$ will be used to define, at time t, k-steps ahead prediction of a system variable x from time t onwards under specified initial state and input scenario. I denotes the identity matrix of corresponding dimensions.

1.1 Problem formulation and preliminaries

Let us start with uncertain plant model described by the following linear discrete-time uncertain system with polytopic uncertainty domain

$$x(t+1) = A(\alpha)x(t) + B(\alpha)u(t) \tag{1}$$

$$y(t) = Cx(t)$$

where $x(t) \in R^n, u(t) \in R^m, y(t) \in R^l$ are state, control and output variables of the system, respectively; $A(\alpha), B(\alpha)$ belong to the convex set

$$S = \{A(\alpha) \in R^{n\times n}, B(\alpha) \in R^{n\times m}\} \tag{2}$$

$$\{A(\alpha) = \sum_{j=1}^{N} A_j\alpha_j \quad B(\alpha) = \sum_{j=1}^{N} B_j\alpha_j, \alpha_j \geq 0\}, j = 1, 2...N, \sum_{j=1}^{N} \alpha_j = 1$$

Matrices A_i, B_i and C are known matrices with constant entries of corresponding dimensions. Simultaneously with (1) we consider the nominal model of system (1) in the form

$$x(t+1) = A_o x(t) + B_o u(t) \quad y(t) = Cx(t) \tag{3}$$

where A_o, B_o are any constant matrices from the convex bounded domain S (2). The nominal model (3) will be used for prediction, while (1) is considered as real plant description providing plant output. Therefore in the robust controller design we assume that for time t output $y(t)$ is obtained from uncertain model (1), predicted outputs for time $t+1, ...t+N_2$ will be obtained from model prediction, where the nominal model (3) is used. The predicted states and outputs of the system (1) for the instant $t+k, k = 1, 2, ...N_2$ are given by

- k=1
  $x(t+2) = A_o x(t+1) + B_o u(t+1) = A_o A(\alpha)x(t) + A_o B(\alpha)u(t) + B_o u(t+1)$
- k=2
  $x(t+3) = A_o^2 A(\alpha)x(t) + A_o^2 B(\alpha)u(t) + A_o B_o u(t+1) + B_o u(t+2)$
- for k

$$x(t+k+1) = A_o^k A(\alpha)x(t) + A_o^k B(\alpha)u(t) + \sum_{i=0}^{k-1} A_o^{k-i-1} B_o u(t+1+i) \tag{4}$$

  and corresponding output is

$$y(t+k) = Cx(t+k) \tag{5}$$

Consider a set of $k = 0, 1, 2, ..., N_2$ state/output model predictions as follows

$$z(t+1) = A_f(\alpha)z(t) + B_f(\alpha)v(t), \quad y_f(t) = C_f z(t) \tag{6}$$

where

$$z(t)^T = [x(t)^T ... x(t+N_2)^T], v(t)^T = [u(t)^T ... u(t+N_u)^T] \tag{7}$$
$$y_f(t)^T = [y(t)^T ... y(t+N_2)^T]$$

and

$$B_f(\alpha) = \begin{bmatrix} B(\alpha) & 0 & ... & 0 \\ A_o B(\alpha) & B_o & ... & 0 \\ ... & ... & ... & 0 \\ A_o^{N_2} B(\alpha) & A_o^{N_2-1} B_o & ... & A_o^{N_2-N_u} B_o \end{bmatrix} \tag{8}$$

$$A_f(\alpha) = \begin{bmatrix} A(\alpha) & 0 & ... & 0 \\ A_o A(\alpha) & 0 & ... & 0 \\ ... & ... & ... & ... \\ A_o^{N_2} A(\alpha) & 0 & ... & 0 \end{bmatrix}, C_f = \begin{bmatrix} C & 0 & ... & 0 \\ 0 & C & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & C \end{bmatrix} \tag{9}$$

where N_2, N_u are output and control prediction horizons of model predictive control, respectively. Note that for output/state prediction in (6) one needs to put $A(\alpha) = A_o, B(\alpha) = B_o$. Matrices dimensions are $A_f(\alpha) \in R^{n(N_2+1)\times n(N_2+1)}, B_f(\alpha) \in R^{n(N_2+1)\times m(N_u+1)}$ and $C_f \in R^{l(N_2+1)\times n(N_2+1)}$.

Consider the cost function associated with the system (6) in the form

$$J = \sum_{t=0}^{\infty} J(t) \tag{10}$$

where

$J(t) = \sum_{k=0}^{N_2} x(t+k)^T Q_k x(t+k) + \sum_{k=0}^{N_u} u(t+k)^T R_k u(t+k) =$

$$= z(t)^T Q z(t) + v(t)^T R v(t) \tag{11}$$

$Q = blockdiag\{Q_i\}_{i=0,1,...N_2}$ $R = blockdiag\{R_i\}_{i=0,1,...N_u}$

The problem studied in this part of chapter can be summarized as follows. Design the robust model predictive controller with output feedback and input constraints in the form

$$v(t) = F y_f(t) = F C_f z(t) \tag{12}$$

where $F^T = [F_0^T ... F_{N_u}^T], \quad F_i = [F_{i0} ... F_{iN_2}], \quad i = 0, 1, 2, ... N_u$

are the output feedback gain matrices which for given prediction horizon N_2 and control horizon N_u ensure the closed-loop system (13) stability, robustness and guaranteed cost.

$$z(t+1) = (A_f(\alpha) + B_f(\alpha) F C_f) z(t) = A_c(\alpha) z(t) \tag{13}$$

Definition 1. Consider the system (6). If there exists a control law $v(t)^*$ and a positive scalar J^* such that the closed-loop system (13) is stable and the closed-loop cost function (10) value J satisfies $J \leq J^*$ then J^* is said to be the guaranteed cost and $v(t)^*$ is said to be the guaranteed cost control law for the system (6).

To guarantee closed-loop stability of uncertain system overall the whole uncertainty domain, the concept of quadratic stability is frequently used. That is, one Lyapunov function works for the whole uncertainty domain. Experience and analysis has shown that quadratic stability is rather conservative in many cases, therefore robust stability with parameter dependent Lyapunov function $P(\alpha)$ has been introduced by (Peaucelle et al., 2000). Using the concept of Lyapunov stability it is possible to formulate the following definition and lemma.

Definition 2. System (13) is robustly stable in the convex uncertainty domain with parameter-dependent Lyapunov function $P(\alpha)$ if and only if there exists a matrix $P(\alpha) = P(\alpha)^T > 0$ such that

$$A_c(\alpha)^T P(\alpha) A_c(\alpha) - P(\alpha) < 0 \tag{14}$$

Lemma 1. (Rosinová et al., 2003), (Krokavec & Filasová, 2003) Consider the closed-loop system (13) with control algorithm (12). Control algorithm (12) is the guaranteed cost control law if and only if there exists a positive definite matrix $P(\alpha)$ and matrix F such that the following condition holds

$$B_e = z(t)^T (A_c(\alpha)^T P(\alpha) A_c(\alpha) - P(\alpha) + Q + C_f^T F^T R F C_f) z(t) \leq 0 \tag{15}$$

where the first term of (15) $\Delta V(t) = z(t)^T(A_c(\alpha)^T P(\alpha) A_c(\alpha) - P(\alpha))z(t)$ is the first difference of closed-loop system Lyapunov function $V(t) = z(t)^T P(\alpha) z(t)$. Moreover, summarizing (15) from initial time t_o to $t \to \infty$ the following inequality is obtained

$$- V(t_o) + J \leq 0 \tag{16}$$

Definition 1 and (16) imply

$$J^* \leq V(t_o) \tag{17}$$

Note, that as a receding horizon strategy is used, only $u(t)$ is sent to the real plant control, control inputs $u(t+k), k = 0, 1, 2, ..., N_u$ are used for predictive outputs $y(t+k)$ calculation. According to (de Oliviera et al., 2000) there is no general and systematic way to formally determine $P(\alpha)$ in (15) as a function of $A_c(\alpha)$. Such a matrix $P(\alpha)$ is called the parameter dependent Lyapunov matrix (PDLM) and for particular structure of $P(\alpha)$ the inequality (15) defines the parameter dependent quadratic stability (PDQS). Formal approach to choose $P(\alpha)$ for real convex polytopic uncertainty (2) can be found in the references. One of the approaches is to take $P(\alpha) = P$, in this case if the solution is feasible the quadratic stability is obtained. Another possibility $P(\alpha) = \sum_{i=1}^{N} P_i \alpha_i, \sum_{i=1}^{N} \alpha_i = 1, P_i = P_i^T > 0$ gives the parameter dependent quadratic stability (PDQS). To decrease the conservatism of PDQS arising from affine parameter dependent Lyapunov function (PDLF), recently, the use of polynomial PDLF (PPDLF) has been proposed in different forms. For more details see (Ebihara et al., 2006).

1.2 Robust model predictive controller design. Quadratic stability

Robust MPC controller design which guarantees quadratic stability and guaranteed cost of closed-loop system is based on (15). Using Schur complement formula inequality (15) can be rewritten to following bilinear matrix inequality (BMI).

$$\begin{bmatrix} -P(\alpha) + Q & C_f^T F^T & A_c(\alpha)^T \\ F C_f & -R^{-1} & 0 \\ A_c(\alpha) & 0 & -P(\alpha)^{-1} \end{bmatrix} \leq 0 \tag{18}$$

For the quadratic stability $P(\alpha) = P = P^T > 0$ in (18). Using linearization approach for P^{-1}, de Oliviera et al. (2000), the following inequality can be derived

$$- P^{-1} \leq Y_k^{-1}(P - Y_k)Y_k^{-1} - Y_k^{-1} = lin(-P^{-1}) \tag{19}$$

where $Y_k, k = 1, 2, ...$ in iteration process $Y_k = P$. We can recast bilinear matrix inequality (18) to the linear matrix inequality (LMI) using linearization (19). The following LMI is obtained for quadratic stability

$$\begin{bmatrix} -P + Q & C_f^T F^T & A_{fi}^T + C_f^T F^T B_{fi}^T \\ F C_f & -R^{-1} & 0 \\ A_{fi} + B_{fi} F C_f & 0 & lin(-P^{-1}) \end{bmatrix} \leq 0 \quad i = 1, 2, ...N \tag{20}$$

where

$$A_f(\alpha) = \sum_{j=1}^{N} A_{fj} \alpha_j \quad B_f(\alpha) = \sum_{j=1}^{N} B_{fj} \alpha_j$$

We can conclude that if the LMIs (20) are feasible with respect to $\varrho * I > P = P^T > 0$ and matrix F then the closed-loop system with control algorithm (12) is quadratically stable with

guaranteed cost (17). Note that due to control horizon strategy only the first m rows of matrix F are used for real plant control, the other part of matrix F serves for predicted output variables calculation. Parameter dependent or Polynomial parameter dependent quadratic stability approach to design robust MPC may decrease the conservatism of quadratic stability. In this case for PDQS we can use the approaches given in (Peaucelle et al., 2000), (Grman et al., 2005) and for (PPDLF) see (Ebihara et al., 2006).

1.3 MPC design for input constraints

In this subsection we propose the off-line calculation of two control gain matrices and using analogy to SVSC approach, (Adamy & Fleming, 2004), we significantly reduce the computational effort for MPC suboptimal control with input constraints.
To design model predictive control (Adamy & Fleming, 2004), (Camacho & Bordons, 2004) with constraints on input, state and output variables at each sampling time, starting from the current state, an open-loop optimal control problem is solved over the defined finite horizon. The first element of the optimal control sequence is applied to the plant. At the next time step, the computation is repeated with new measured variables. Thus, the implementation of the MPC strategy requires a QP solver for the on-line optimization which still requires significant on-line computational effort, which limits MPC applicability.
In our approach the actual output feedback control gain matrix is computed as a convex combination of two gain matrices computed a priori (off-line) : one for constrained and one for unconstrained case such that both gains guarantee performance and robustness properties of closed-loop system. This convex combination is determined by a scalar parameter which is updated on-line in each step. Based on this idea, in the following, the algorithm for constrained control algorithm is developed.
Consider the system (6) where the control $v(t)$ is constrained to evolve in the following set

$$\Gamma = \{v \in R^{mN_u} : |v_i(t)| \leq U_i, i = 1, ...mN_u\} \tag{21}$$

The aim of this part of chapter is to design the stabilizing output feedback control law for system (6) in the form

$$v(t) = FC_f z(t) \tag{22}$$

which guarantees that for the initial state $z_0 \in \Omega(P) = \{z(t) : z(t)^T P z(t) \leq \theta\}$ control $v(t)$ belongs to the set (21) for all $t \geq 0$, where θ is a positive real parameter which determines the size of $\Omega(P)$. Furthermore, $\Omega(P)$ should be such that all $z(t) \in \Omega(P)$ provide $v(t)$ satisfying the relation (21), restricting the values of the control parameters. Moreover, the following ellipsoidal Lyapunov function level set

$$\Omega(P) = \{z(t) \in R^{nN_2} : z(t)^T P z(t) \leq \theta\} \tag{23}$$

can be proven to be a robust positively invariant region with respect to motion of the closed-loop system in the sense of the following definition, (Rohal-Ilkiv, 2004), (Ayd et al., 2008) .
Definition 3. A subset $S_o \in R^{(nN_2)}$ is said to be positively invariant with respect to motion of system (6) with control algorithm (22) if for every initial state $z(0)$ inside S_o the trajectory $z(t)$ remains in S_o for all $t \geq 0$.
Consider that vector f_i denotes the i-th row of matrix F and define

$$L(F) = \{z(t) \in R^{(nN_2)} : |f_i C_f z(t)| \leq U_i, i = 1, 2...mN_u\}$$

The above set can be rewritten as

$$L(F) = \{z(t) \in R^{(nN_2)} : |D_i F C_f z(t)| \leq U_i, i = 1, 2...mN_u\} \tag{24}$$

where $D_i \in R^{1 \times mN_u} = \{dij\}, \quad d_{ij} = 1, \quad i = j, \quad d_{ij} = 0, i \neq j$. The results are summarized in the following theorem.

Theorem 1. The inclusion $\Omega(P) \subseteq L(F)$ is for output feedback control equivalent to

$$\begin{bmatrix} P & C^T F^T D_i^T \\ D_i F C & \lambda_i \end{bmatrix} \geq 0, \quad i = 1, 2, ...mN_u \tag{25}$$

where

$$\lambda_i \in < 0, \frac{U_i^2}{\theta} >$$

Proof. To prove that the inclusion $\Omega(P) \subseteq L(F)$ is equivalent to (25) we use $S-$ procedure in the following way. Rewrite (23) and (24) to the following form

$$p(z) = z^T(t) P z(t) - \theta \leq 0$$

$$g_i(z) = z^T(t) C_f^T F^T D_i^T D_i F C z(t) - U_i^2 \leq 0$$

According to $S-$ procedure the above inclusion is equivalent to the existence of a positive scalar λ_i such that

$$g_i(z) - \lambda_i p(z) \leq 0$$

or equivalently

$$z(t)^T (C_f^T F^T D_i^T D_i F C - \lambda_i P) z(t) - U_i^2 + \lambda_i \theta \leq 0 \tag{26}$$

After some manipulation (26) can be rewritten in the form

$$\begin{bmatrix} C^T F^T D_i^T D_i F C - \lambda_i P & 0 \\ 0 & -U_i^2 + \lambda_i \theta \end{bmatrix} \leq 0 \tag{27}$$

$$i = 1, 2, ...mN_u$$

The above inequality for block diagonal matrix is equivalent to two inequalities. Using Schur complement formula for the first one the inequality (25) is obtained, which proves the theorem.

In order to check the value of θ_i for $i - th$ input we solve the optimization problem $z(t)^T P z(t) \rightarrow$ max, subject to constraints (24), which yields

$$\theta_i = \frac{U_i^2}{D_i F C P^{-1} C^T F^T D_i^T} \tag{28}$$

In the design procedure it should be verified that when parameter θ decreases the obtained robust positively invariant regions $\Omega(P)$ are nested to region obtained for $\theta + \epsilon, \epsilon > 0$.

Assume that we calculate two output feedback gain matrices: F_1 for unconstrained case and F_2 for constrained one. Obviously, closed-loop system with the gain matrix F_2 gives the dynamic behavior slower than the one obtained for F_1. Consider the output feedback gain matrix F in the form

$$F = \gamma F_1 + (1 - \gamma) F_2, \quad \gamma \in (0, 1) \tag{29}$$

For gain matrices $F_i, i = 1,2$ we obtain two closed-loop system in the form (13), $A_{ci} = A_f + B_f F_i C_f, i = 1,2$. Consider the edge between A_{c1} and A_{c2}, that is

$$A_c = \alpha A_{c1} + (1-\alpha) A_{c2}, \quad \alpha \in < 0,1 > \tag{30}$$

The following lemma gives the stability conditions for matrix A_c (30).
Lemma 2. Consider the stable closed-loop system matrices $A_{ci}, i = 1,2$.

- If there exists a positive definite matrix P_q such that

$$A_{ci}^T P_q A_{ci} - P_q \leq 0, \quad i = 1,2 \tag{31}$$

  then matrix A_c (30) is quadratically stable.
- If there exist two positive definite matrices P_1, P_2 such that they satisfy the parameter dependent quadratic stability conditions, see (Peaucelle et al., 2000), (Grman et al., 2005) the closed-loop system A_c is parameter dependent quadratically stable (PDQS).

Remarks

- If closed-loop matrices $A_{ci}, \quad i = 1,2$ satisfy (31) the scalar γ in (29) may be changed with any rate without violating the closed-loop stability.
- If closed-loop matrices $A_{ci}, \quad i = 1,2$ are PDQS, the scalar γ in (29) has to be constant but may be unknown.
- The proposed control algorithm (29) is similar to Soft Variable-Structure Control (SVSC), (Adamy & Fleming, 2004), but in our case, when $|v_i| << U_i$ the feedback gain matrix F (29) gives rather quicker dynamic behavior of the closed-loop system (unconstrained case) than when $|v_i|$ approaches to U_i.

Algorithm for calculation of γ for (29) may be as follows:

$$\gamma = \min_i \frac{U_i - |v_i|}{U_i} \tag{32}$$

If accidentally some $|v_i| > U_i, \gamma = 0$.
The resulting control design procedure is given by the next steps

- Off-line computation stage, compute output feedback gain matrices:
  F_1 for unconstrained case as a solution to (20), where LMI (20) is solved for unknown P and F;
  F_2 for constrained case as a solution to (20) and (25).
- On-line computation- in each step:
  compute the actual value of scalar parameter γ, e.g from (32) (where v_i is obtained from (12) for $F = F_1$;
  compute the actual feedback gain matrix from (29) and respective constrained control vector from (12). All on-line computations follow general MPC scheme, i.e. the first part of computed control vector $u(t)$ is applied on real controlled plant and the other part of control vector is used for model prediction.

1.4 EXAMPLES

Two examples are presented to illustrate the qualities of the control design procedure proposed above, namely its ability to cope with robust stability and input constraints without complex computational load. In each example the results of three simulation experiments are compared for closed-loop with output feedback control:
case 1 Unconstrained case for output feedback gain matrix F_1
case 2 Constrained case for output feedback gain matrix F_2
case 3 The new proposed control algorithm (29) for output feedback gain matrix F.
The input constraint case is studied, in each case maximal value of $u(t)$ is checked; stability is assessed using spectral radius of closed-loop system matrix.
First example serves as a benchmark. The model of double integrator turns to (1) where

$$A_o = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

$$B_o = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

and uncertainty matrices are

$$A_{1u} = \begin{bmatrix} 0.01 & 0.01 \\ 0.02 & 0.03 \end{bmatrix}$$

$$B_{1u} = \begin{bmatrix} 0.001 \\ 0 \end{bmatrix},$$

For the case when number of uncertainty is $p = 1$, the number of the respective polytope vertices is $N = 2^p = 2$, the matrices (2) are calculated as follows

$$A_1 = A_o - A_{1u}, \quad A_2 = A_o + A_{1u}, \quad B_1 = B_o - B_{1u}, \quad B_2 = B_o + B_{1u}$$

For the parameters: $\varrho = 20000, N_2 = 6, N_u = 6, Q_0 = 0.1I, Q_1 = 0.5I, Q_2 = ... = Q_6 = I, R = I$, the following results are obtained for unconstrained and constrained cases

- Unconstrained case: $Closed - loopmaxeig = 0.8495$. Maximal value of control variable is about $u_{max} = 0.24$.
- Constrained case with $U_i = 0.1, \theta = 1000$, $Closed - loopmaxeig = 0.9437$. Maximal value of control variable is about $u_{max} = 0.04$.

Closed-loop step responses for unconstrained and constrained cases are given in Fig.1 and Fig.2, respectively. Closed-loop step responses for the case of in this chapter proposed algorithm are given in Fig.3. Maximal value of control variable is about $u_{max} = 0.08 < 0.1$.
Input constraints conditions were applied only for plant control variable $u(t)$.
Second example has been borrowed from (Camacho & Bordons (2004), p.147). The model corresponds to the longitudinal motion of a Boeing 747 airplane. The multivariable process is controlled using a predictive controller based on the output model of the aircraft. Two of the usual command outputs that must be controlled are airspeed that is, velocity with respect to air, and climb rate. Continuous model has been converted to discrete time one with sampling time of $0.1s$, the nominal model turns to (1) where

$$A_o = \begin{bmatrix} .9996 & .0383 & .0131 & -.0322 \\ -.0056 & .9647 & .7446 & .0001 \\ .002 & -.0097 & .9543 & 0 \\ .0001 & -.0005 & .0978 & 1 \end{bmatrix}$$

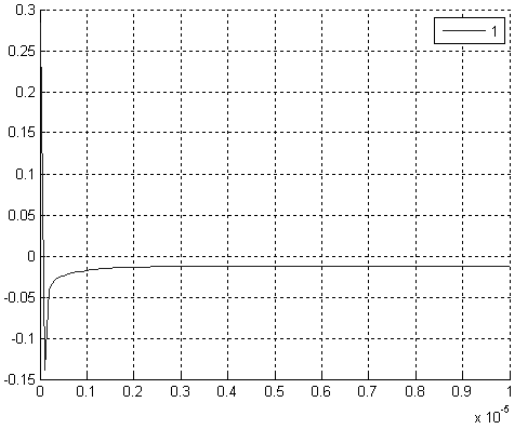


Fig. 1. Dynamic behavior of controlled system for unconstrained case for $u(t)$.

$$B_o = \begin{bmatrix} .0001 & .1002 \\ -.0615 & .0183 \\ -.1133 & .0586 \\ -.0057 & .0029 \end{bmatrix} C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 7.74 \end{bmatrix}$$

and model uncertainty matrices are

$$A_{1u} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.0005 & 0.0017 & 0 \\ 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B_{1u} = \begin{bmatrix} 0 & 0.12 \\ -0.02 & 0.1 \\ -0.12 & 0 \\ 0 & 0 \end{bmatrix} 10^{-3}$$

For the case when number of uncertainty is $p = 1$, the number of vertices is $N = 2^p = 2$, the matrices (2) are calculated as in example 1. Note that nominal model A_o is unstable. Consider $N_2 = N_u = 1, \varrho = 20000$ and weighting matrices $Q_0 = Q_1 = 1I, R_0 = R_1 = I$ the following results are obtained:

- Unconstrained case: maximal closed-loop nominal model eigenvalue is $Closed - loopmaxeig = 0.9983$. Maximal value of control variables are about $u_{1max} = 9.6, u_{2max} = 6.3$.
- Constrained case with $U_i = 1, \theta = 40000$ $Closed - loopmaxeig = 0.9998$ Maximal values of control variables are about $u_{1max} = 0.21, u_{2max} = 0.2$.

Closed-loop nominal model step responses of the above two cases for the input $u(t)$ are given in the Fig.4 and Fig.5, respectively. Closed-loop step responses for in the paper proposed control algorithm (29) and (32) are in Fig.6. Maximal values of control variables are about $u_{1max} = 0.75 < 1, u_{2max} = 0.6 < 1$. Input constraint conditions were applied only for plant control variable $u(t)$. Both examples show that using tuning parameter θ the demanded input

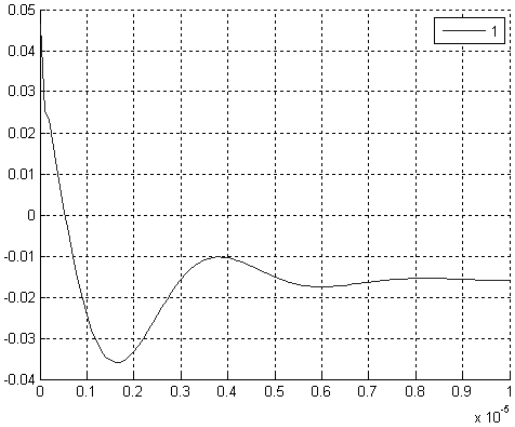


Fig. 2. Dynamic behavior of controlled system for constrained case for $u(t)$.

constraints can be reached with high accuracy. The initial guess of θ can be obtained from (28). It can be seen that the proposed control scheme provides reasonable results : the response in case 3 (Fig.3 , Fig. 6) are quicker than those in case 2 (Fig.2, Fig.5), while the computation load has not much increased comparing to case 2.

2. ROBUST MPC DESIGN: SEQUENTIAL APPROACH

2.1 INTRODUCTION

In this part a new MPC algorithm is proposed pursuing the idea of (Veselý & Rosinová , 2009). The proposed robust MPC control algorithm is designed sequentially. The respective sequential robust MPC design procedure consists from two steps. In the first step and one step ahead prediction horizon, the necessary and sufficient robust stability conditions have been developed for MPC and polytopic system with output feedback, using generalized parameter dependent Lyapunov matrix $P(\alpha)$. The proposed robust MPC algorithm ensures parameter dependent quadratic stability (PDQS) and guaranteed cost. In the second step of design procedure, the nominal plant model is used to design the predicted input variables $u(t+1),...u(t+N_2-1)$ so that the robust closed-loop stability of MPC and guaranteed cost are ensured. Thus, input variable $u(t)$ guarantees the performance and robustness properties of closed-loop system and predicted input variables $u(t+1),...u(t+N_2-1)$ guarantee the performance and closed-loop stability of uncertain plant model and nominal model prediction. Note that within sequentially design procedure the degree of plant model does not change when the output prediction horizon changes.

This part of chapter is organized as follows: Section 2.2 presents preliminaries and problem formulation. In Section 2.3 the main results are given and finally, in Section 2.4 two examples solved using Yalmip BMI solvers show the effectiveness of the proposed method.

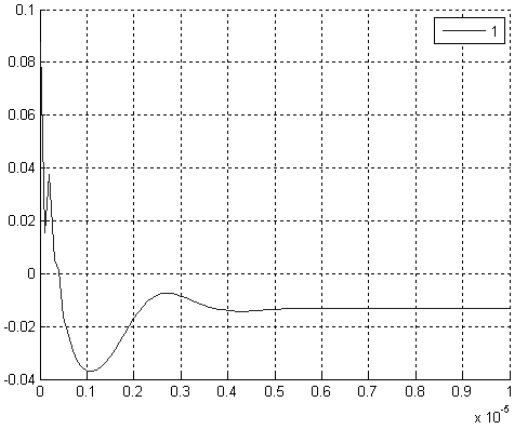


Fig. 3. Dynamic behavior of controlled system with the proposed algorithm for $u(t)$.

2.2 PROBLEM FORMULATION AND PRELIMINARIES

For readers convenience, uncertain plant model and respective preliminaries are briefly recalled. A time invariant linear discrete-time uncertain polytopic system is

$$x(t+1) = A(\alpha)x(t) + B(\alpha)u(t) \tag{33}$$

$$y(t) = Cx(t)$$

where $x(t) \in R^n, u(t) \in R^m, y(t) \in R^l$ are state, control and output variables of the system, respectively; $A(\alpha), B(\alpha)$ belong to the convex set

$$S = \{A(\alpha) \in R^{n\times n}, B(\alpha) \in R^{n\times m}\} \tag{34}$$

$$\{A(\alpha) = \sum_{j=1}^{N} A_j\alpha_j \quad B(\alpha) = \sum_{j=1}^{N} B_j\alpha_j, \alpha_j \geq 0\}, j = 1, 2...N, \sum_{j=1}^{N} \alpha_j = 1$$

Matrix C is constant known matrix of corresponding dimension. Jointly with the system (33), the following nominal plant model will be used.

$$x(t+1) = A_o x(t) + B_o u(t) \tag{35}$$

$$y(t) = Cx(t)$$

where $(A_o, B_o) \in S$ are any matrices with constant entries. The problem studied in this part of chapter can be summarized as follows: in the first step, parameter dependent quadratic stability conditions for output feedback and one step ahead robust model predictive control are derived for the polytopic system (33), (34), when control algorithm is given as

$$u(t) = F_{11}y(t) + F_{12}y(t+1) \tag{36}$$

and in the second step of design procedure, considering a nominal model (35) and a given prediction horizon N_2 a model predictive control is designed in the form:

$$u(t+k-1) = F_{kk}y(t+k-1) + F_{kk+1}y(t+k) \tag{37}$$

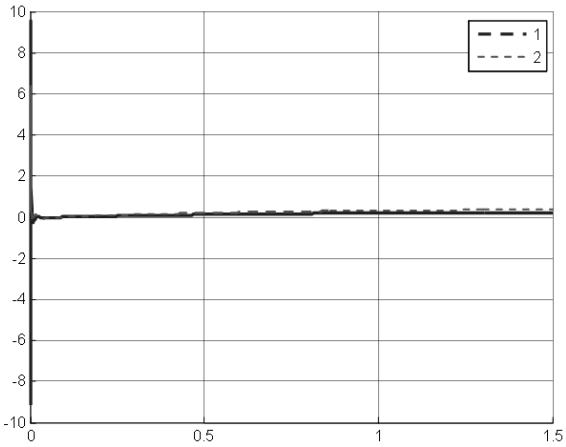


Fig. 4. Dynamic behavior of unconstrained controlled system for $u(t)$.

where $F_{ki} \in R^{m\times l}, k = 2,3,...N_2; i = k+1$ are output (state) feedback gain matrices to be determined so that cost function given below is optimal with respect to system variables. We would like to stress that $y(t+k-1), y(t+1)$ are predicted outputs obtained from predictive model (44).
Substituting control algorithm (36) to (33) we obtain

$$x(t+1) = D_1(j)x(t) \tag{38}$$

where

$$D_1(j) = A_j + B_j K_1(j)$$

$$K_1(j) = (I - F_{12}CB_j)^{-1}(F_{11}C + F_{12}CA_j), j = 1,2,...N$$

For the first step of design procedure, the cost function to be minimized is given as

$$J_1 = \sum_{t=0}^{\infty} J_1(t) \tag{39}$$

where

$$J_1(t) = x(t)^T Q_1 x(t) + u(t)^T R_1 u(t)$$

and Q_1, R_1 are positive definite matrices of corresponding dimensions. For the case of $k = 2$ we obtain

$$u(t+1) = F_{22}CD_1(j)x(t) + F_{23}C(A_o D_1(j)x(t) + B_o u(t+1))$$

or

$$u(t+1) = K_2(j)x(t)$$

and closed-loop system

$$x(t+2) = (A_o D_1(j) + B_o K_2(j))x(t) = D_2(j)x(t), j = 1,2,...N$$

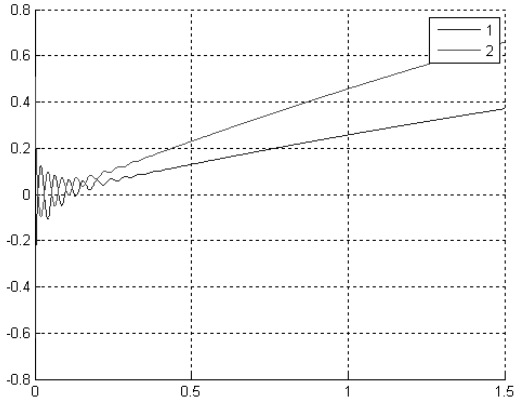


Fig. 5. Dynamic behavior of constrained controlled system for $u(t)$.

Sequentially, for $k = N_2 \geq 2$ step prediction, we obtain the following closed-loop system

$$x(t+k) = (A_o D_{k-1}(j) + B_o K_k(j))x(t) = D_k(j)x(t) \tag{40}$$

where

$$D_0 = I, D_k(j) = A_o D_{k-1}(j) + B_o K_k(j) \quad k = 2, 3, ..., N_2; \quad j = 1, 2, ...N$$
$$K_k(j) = (I - F_{kk+1} C B_o)^{-1}(F_{kk} C + F_{kk+1} C A_o) D_{k-1}(j)$$

For the second step of robust MPC design procedure and k prediction horizon the cost function to be minimized is given as

$$J_k = \sum_{t=0}^{\infty} J_k(t) \tag{41}$$

where

$$J_k(t) = x(t)^T Q_k x(t) + u(t+k-1)^T R_k u(t+k-1)$$

and $Q_k, R_k, k = 2, 3, ...N_2$ are positive definite matrices of corresponding dimensions. We proceed with two corollaries following from Definition 2 and Lemma 1.

Corollary 1

The closed-loop system matrix of discrete-time system (1) is robustly stable if and only if there exists a symmetric positive definite parameter dependent Lyapunov matrix $0 < P(\alpha) = P(\alpha)^T < I\varrho$ such that

$$-P(\alpha) + D_1(\alpha)^T P(\alpha) D_1(\alpha) \leq 0 \tag{42}$$

where $D_1(\alpha)$ is the closed-loop polytopic system matrix for system (33). The necessary and sufficient robust stability condition for closed-loop polytopic system with guaranteed cost is given by the recent result (Rosinová et al., 2003).

Corollary 2

Consider the system (33) with control algorithm (36). Control algorithm (36) is the guaranteed cost control law for the closed-loop system if and only if the following condition holds

$$B_e = D_1(\alpha)^T P(\alpha) D_1(\alpha) - P(\alpha) + Q_1 + (F_{11} C + F_{12} C D_1(\alpha))^T R_1 (F_{11} C + \tag{43}$$

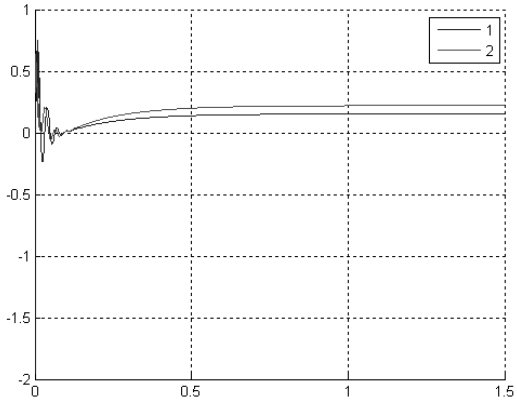


Fig. 6. Dynamic behavior for proposed control algorithm (29) and (32) for $u(t)$.

$$+F_{12}CD_1(\alpha)) \leq 0$$

For the nominal model and $k = 1, 2, ...N_2$ the model prediction can be obtained in the form

$$z(t+1) = A_f z(t) + B_f v(t) \tag{44}$$

$$y_f(t) = C_f z(t)$$

where

$$z(t)^T = [x(t)^T ... x(t+N_2-1)^T]$$
$$v(t)^T = [u(t)^T ... u(t+N_2-1)^T]$$
$$y_f(t)^T = [y(t)^T ... y(t+N_2-1)^T]$$

$$A_f = \begin{bmatrix} A_o & 0 & 0 & ... & 0 \\ A_o D_1 & 0 & 0 & ... & 0 \\ A_o D_2 & 0 & 0 & ... & 0 \\ ... & ... & ... & ... & ... \\ A_o D_{N_2-1} & 0 & 0 & ... & 0 \end{bmatrix} \in R^{nN_2 \times nN_2}$$

$$B_f = blockdiag\{B_o\}_{nN_2 \times mN_2}$$
$$C_f = blockdiag\{C\}_{lN_2 \times nN_2}$$

Remarks

- Control algorithm for $k = N_2$ is $u(t+N_2-1) = F_{N_2 N_2} y(t+N_2-1)$.
- If one wants to use control horizon $N_u < N_2$ (Camacho & Bordons, 2004), the control algorithm is $u(t+k-1) = 0, K_k = 0, F_{N_{u+1} N_{u+1}} = 0, F_{N_{u+1} N_{u+2}} = 0$ for $k > N_u$.
- Note that model prediction (44) is calculated using nominal model (35), that is $D_0 = I, D_k = A_o D_{k-1} + B_o K_k$, $D_k(j)$ is used robust controller design procedure.

2.3 MAIN RESULTS

2.3.1 Robust MPC controller design. First step

The main results for the first step of design procedure can be summarized in the following theorem.

Theorem 2.

The system (33) with control algorithm (36) is parameter dependent quadratically stable with parameter dependent Lyapunov function $V(t) = x(t)^T P(\alpha)x(t)$ if and only if there exist matrices $N_{11}, N_{12}, F_{11}, F_{12}$ such that the following bilinear matrix inequality holds.

$$B_e = \begin{bmatrix} G_{11} & G_{12} \\ G_{12}^T & G_{22} \end{bmatrix} \leq 0 \tag{45}$$

where

$$G_{22} = N_{12}^T A_c(\alpha) + A_c(\alpha)^T N_{12} - P(\alpha) + Q_1 + C^T F_{11}^T R_1 F_{11} C$$
$$G_{12}^T = A_c(\alpha)^T N_{11} + N_{12}^T M_c(\alpha) + C^T F_{11}^T R_1 F_{12} C$$
$$G_{11} = N_{22}^T M_c(\alpha) + M_c(\alpha)^T N_{22} + C^T F_{12}^T R_1 F_{12} C + P(\alpha)$$
$$M_c(\alpha) = B(\alpha) F_{12} C - I$$
$$A_c(\alpha) = A(\alpha) + B(\alpha) F_{11} C$$

Note that (45) is affine with respect to α. Substituting (34) and $P(\alpha) = \sum_{i=1}^{N} \alpha_i P_i$ to (45) the following BMI is obtained for the polytopic system

$$B_{ie} = \begin{bmatrix} G_{11i} & G_{12i} \\ G_{12i}^T & G_{22i} \end{bmatrix} \leq 0 \quad i = 1, 2, ...N \tag{46}$$

where

$$G_{11i} = N_{22}^T M_{ci} + M_{ci}^T N_{22} + C^T F_{12}^T R_1 F_{12} C + P_i$$
$$G_{12i}^T = A_{ci}^T N_{22} + N_{12}^T M_{ci} + C^T F_{11}^T R_1 F_{12} C$$
$$G_{22i} = N_{12}^T A_{ci} + A_{ci}^T N_{12} - P_i + Q_1 + C^T F_{11}^T R_1 F_{11} C$$
$$M_{ci} = B_i F_{12} C - I \quad A_{ci} = A_i + B_i F_{11} C$$

Proof. For the proof of this theorem see the proof of *Theorem 3* .

If the solution of (46) is feasible with respect to symmetric matrices $P_i = P_i^T > 0, i = 1, 2...N$, and matrices N_{11}, N_{12}, within the convex set defined by (34), the gain matrices F_{11}, F_{12} ensure the guaranteed cost and parameter dependent quadratic stability (PDQS) of closed-loop polytopic system for one step ahead predictive control.

Note that:

- For concrete matrix $P(\alpha) = \sum_{i=1}^{N} \alpha_i P_i$ BMI robust stability conditions "if and only if" in (45) reduces in (46) to BMI conditions " if".
- If in (46) $P_i = P_j = P, i \neq j = 1, 2...N$, the feasible solution of (46) with respect to matrices N_{11}, N_{12}, and symmetric positive definite matrix P gives the gain matrices F_{11}, F_{12} guaranteeing quadratic stability and guaranteed cost for one step ahead predictive control for the closed-loop polytopic system within the convex set defined by (34). Quadratic stability gives more conservative results than PDQS. Conservatism of real results depend on the concrete examples.

Assume that the BMI solution of (46) is feasible, then for nominal plant one can calculate matrices D_1 and K_1 using (38). For the second step of MPC design procedure, the obtained nominal model will be used.

2.3.2 Model predictive controller design. Second step

The aim of the second step of predictive control design procedure is to design gain matrices $F_{kk}, F_{kk+1}, k = 2,3,...N_2$ such that the closed-loop system with nominal model is stable with guaranteed cost. In order to design model predictive controller with output feedback in the second step of design procedure we proceed with the following corollary and theorem.

Corollary 3

The closed-loop system (40) is stable with guaranteed cost iff the following inequality holds

$$B_{ek}(t) = \Delta V_k(t) + x(t)^T Q_k x(t) + u(t+k-1)^T R_k u(t+k-1) \leq 0 \tag{47}$$

where $\Delta V_k(t) = V_k(t+k) - V_k(t)$ and $V_k(t) = x(t)^T P_k x(t)$, $P_k = P_k^T > 0, k = 2,3,...N_2$.

Theorem 3

The closed-loop system (40) is robustly stable with guaranteed cost iff for $k = 2,3,...N_2$ there exist matrices

$$F_{kk}, F_{kk+1}, N_{k1} \in R^{n\times n}, N_{k2} \in R^{n\times n}$$

and positive definite matrix $P_k = P_k^T \in R^{n\times n}$ such that the following bilinear matrix inequality holds

$$B_{e2} = \begin{bmatrix} G_{k11} & G_{k12} \\ G_{k12}^T & G_{k22} \end{bmatrix} \leq 0 \tag{48}$$

where

$$G_{k11} = N_{k1}^T M_{ck} + M_{ck}^T N_{k1} + C^T F_{kk+1}^T R_k F_{kk+1} C + P_k$$

$$G_{k12}^T = D_{k-1}(j)^T C^T F_{kk}^T R_k F_{kk+1} C + D_{k-1}(j)^T A_{ck}^T N_{k1} + N_{k2}^T M_{ck}$$

$$\begin{aligned} G_{k22} = Q_k - P_k + D_{k-1}(j)^T C^T F_{kk}^T R_k F_{kk} C D_{k-1}(j) \\ + N_{k2}^T A_{ck} D_{k-1}(j) + D_{k-1}(j)^T A_{ck}^T N_{k2} \end{aligned}$$

and

$$M_{ck} = B_0 F_{kk+1} C - I; \quad A_{ck} = A_0 + B_0 F_{kk} C$$

$$D_k(j) = A_0 D_{k-1}(j) + B_0 K_k(j)$$

$$K_k(j) = (I - F_{kk+1} C B_0)^{-1} (F_{kk} C + F_{kk+1} C A_0) D_{k-1}(j), \quad j = 1,2,...N$$

Proof. Sufficiency.

The closed-loop system (40) can be rewritten as follows

$$x(t+k) = -(M_{ck})^{-1} A_{ck} D_{k-1}(j) x(t) = A_{clk} x(t) \tag{49}$$

Since the matrix (j is omitted)

$$U_k^T = [-D_{k-1}^T A_{ck}^T (M_{ck})^{-1} \quad I]$$

has full row rank, multiplying (48) from left and right side the inequality equivalent to (47) is obtained. Multiplying the results from left by $x(t)^T$ and right by $x(t)$, taking into account the closed-loop matrix (49), the inequality (47) is obtained, which proves the sufficiency.

Necessity.

Suppose that for k-step ahead model predictive control there exists such matrix $0 < P_k =$

$P_k^T < I\rho$ that (48) holds. Necessarily, there exists a scalar $\beta > 0$ such that for the first difference of Lyapunov function in (47) holds

$$A_{clk}^T P_k A_{clk} - P_k \leq -\beta(A_{clk}^T A_{clk}) \tag{50}$$

The inequality (50) can be rewritten as

$$A_{clk}^T (P_k + \beta I) A_{clk} - P_k \leq 0$$

Using Schur complement formula we obtain

$$\begin{bmatrix} -P_k & -A_{clk}^T (P_k + \beta I) \\ (P_k + \beta I) A_{clk} & -(P_k + \beta I) \end{bmatrix} \leq 0 \tag{51}$$

taking

$$\begin{aligned} N_{k1} &= -(M_{ck})^{-1}(P_k + \beta I/2) \\ N_{k2}^T &= -D_{k-1}^T A_{ck}^T (M_{ck}^{-1})^T M_{ck}^{-1} \beta/2 \end{aligned}$$

one obtains

$$\begin{aligned} -A_{clk}^T (P_k + \beta I) &= D_{k-1}^T A_{ck}^T N_{k1} + N_{k2}^T M_{ck} \\ -P_k &= -P_k + N_{k2}^T A_{ck} D_{k-1} + D_{k-1}^T \\ A_{ck}^T N_{k2} + \beta(D_{k-1}^T A_{ck}^T (M_{ck}^{-1})^T M_{ck}^{-1} A_{ck} D_{k-1}) & \\ -(P_k + \beta I) &= 2M_{ck} N_{k1} + P_k \end{aligned} \tag{52}$$

Substituting (52) to (51) for $\beta \to 0$ the inequality (48) is obtained for the case of $Q_k = 0, R_k = 0$. If one substitutes to the second part of (47) for $u(t+k-1)$ from (37), rewrites the obtained result to matrix form and takes sum of it with the above matrix, inequality (48) is obtained, which proves the necessity. It completes the proof.
If there exists a feasible solution of (48) with respect to matrices $F_{kk}, F_{kk+1}, N_{k1} \in R^{n\times n}, N_{k2} \in R^{n\times n}, k = 2,3,...N_2$ and positive definite matrix $P_k = P_k^T \in R^{n\times n}$, then the designed MPC ensures quadratic stability of the closed-loop system and guaranteed cost.

Remarks

- Due to the proposed design philosophy, predictive control algorithm $u(t+k), k \geq 1$ is the function of corresponding performance term (39) and previous closed-loop system matrix.
- In the proposed design approach constraints on system variables are easy to be implemented by LMI using a notion of invariant set (Ayd et al., 2008), (Rohal-Ilkiv, 2004) (see Section 1.3).
- The proposed MPC with sequential design is a special case of classical MPC. Sequential MPC may not provide "better" dynamic behavior than classical one but it is another approach to the design of MPC.
- Note that in the proposed MPC sequential design procedure, the size of system does not change when N_2 increases.
- If there exists feasible solution for both steps in the convex set (34), the proposed control algorithm (37) guarantees the PDQS and robustness properties of closed-loop MPC system with guaranteed cost.

The sequential robust MPC design procedure can be summarized in the following steps:

- Design of robust MPC controller with control algorithm (36) by solving (46).
- Calculate matrices K_1, D_1 and $K_1(j), D_1(j), j = 1, 2, ...N$ given in (38) for nominal and uncertain model of system.
- For a given $k = 2, 3, ...N_2$ and control algorithm (37), sequentially calculate F_{kk}, F_{kk+1} by solving (48) with K_k, D_k given in (40).
- Calculate matrices A_f, B_f, C_f (44) for model prediction.

2.4 EXAMPLES

Example 1. First example is the same as in section 1.5, it serves as a benchmark. The model of double integrator turns to (35) where

$$A_o = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

$$B_o = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

and uncertainty matrices are

$$A_{1u} = \begin{bmatrix} 0.01 & 0.01 \\ 0.02 & 0.03 \end{bmatrix}$$

$$B_{1u} = \begin{bmatrix} 0.001 \\ 0 \end{bmatrix},$$

For the case when number of uncertainties $p = 1$, the number of vertices is $N = 2^p = 2$, the matrices (34) are calculated as

$$A_1 = A_n - A_{1u}, \quad A_2 = A_n + A_{1u}$$

$$B_1 = B_n - B_{1u}, \quad B_2 = B_n + B_{1u}$$

For the parameters: $\varrho = 20000$, prediction and control horizons $N_2 = 4, N_u = 4$, performance matrices $R_1 = ...R_4 = 1, Q_1 = .1I, Q_2 = .5I, Q_3 = I, Q_4 = 5I$, the following results are obtained using the sequential design approach proposed in this part :

- For prediction $k = 1$, the robust control algorithm is given as

  $$u(t) = F_{11}y(t) + F_{12}y(t+1)$$

  From (46), one obtains the gain matrices $F_{11} = 0.9189; F_{12} = -1.4149$. The eigenvalues of closed-loop first vertex system model are as follows

  $$Eig(Closed - loop) = \{0.2977 \pm 0.0644i\}$$

- For $k = 2$, control algorithm is

  $$u(t+1) = F_{22}y(t+1) + F_{23}y(t+2)$$

  In the second step of design procedure control gain matrices obtained solving (48) are $F_{22} = 0.4145; F_{23} = -0.323$. The eigenvalues of closed-loop first vertex system model are

  $$Eig(Closed - loop) = \{0.1822 \pm 0.1263i\}$$

- For k=3, control algorithm is

$$u(t+2) = F_{33}y(t+2) + F_{34}y(t+3)$$

In the second step of design procedure the obtained control gain matrices are $F_{33} = 0.2563; F_{34} = -0.13023$. The eigenvalues of closed-loop first vertex system model are

$$Eig(Closed-loop) = \{0.1482 \pm 0.051i\}$$

- For prediction $k = N_2 = 4$, control algorithm is

$$u(t+3) = F_{44}y(t+3) + F_{45}y(t+4)$$

In the second step the obtained control gain matrices are $F_{44} = 0.5797; F_{45} = 0.0$. The eigenvalues of closed-loop first vertex model system are

$$Eig(Closed-loop) = \{0.1002 \pm 0.145i\}$$

Example 2. Nominal model for the second example is

$$A_o = \begin{bmatrix} 0.6 & 0.0097 & 0.0143 & 0 & 0 \\ 0.012 & 0.9754 & 0.0049 & 0 & 0 \\ -0.0047 & 0.01 & 0.46 & 0 & 0 \\ 0.0488 & 0.0002 & 0.0004 & 1 & 0 \\ -0.0001 & 0.0003 & 0.0488 & 0 & 1 \end{bmatrix}$$

$$B_o = \begin{bmatrix} 0.0425 & 0.0053 \\ 0.0052 & 0.01 \\ 0.0024 & 0.0001 \\ 0 & 0.0012 \end{bmatrix} C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The linear affine type model of uncertain system (34) is in the form

$$A_i = A_n + \theta_1 A_{1u}; \quad B_i = B_n + \theta_1 B_{1u}$$

$$C_i = C, i = 1, 2$$

where A_{1u}, B_{1u} are uncertainty matrices with constant entries, θ_1 is an uncertain real parameter $\theta_1 \in < \underline{\theta_1}, \overline{\theta_1} >$. When lower and upper bounds of uncertain parameter θ_1 are substituted to the affine type model, the polytopic system (33) is obtained. Let $\theta_1 \in < -1, 1 >$ and

$$A_{1u} = \begin{bmatrix} 0.025 & 0 & 0 & 0 & 0 \\ 0 & 0.021 & 0 & 0 & 0 \\ 0 & 0 & 0.0002 & 0 & 0 \\ 0.001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0001 & 0 & 0 \end{bmatrix}$$

$$B_{1u} = \begin{bmatrix} 0.0001 & 0 \\ 0 & 0.001 \\ 0 & 0.0021 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

In this example two vertices ($N = 2$) are calculated. The design problem is: Design two PS(PI) model predictive robust decentralized controllers for plant input $u(t)$ and prediction horizon $N_2 = 5$ using sequential design approach. The cost function is given by the following matrices

$$Q_1 = Q_2 = Q_3 = I, R_1 = R_2 = R_3 = I,$$

$$Q_4 = Q_5 = 0.5I, R_4 = R_5 = I$$

In the first step, calculation for the uncertain system (33) yields the robust control algorithm

$$u(t) = F_{11}y(t) + F_{12}y(t+1)$$

where matrix F_{11} with decentralized output feedback structure containing two PS controllers, is designed. From (46), the gain matrices F_{11}, F_{12} are obtained

$$F_{11} = \begin{bmatrix} -18.7306 & 0 & -42.4369 & 0 \\ 0 & 8.8456 & 0 & 48.287 \end{bmatrix}$$

where decentralized proportional and integral gains for the first controller are

$$K_{1p} = 18.7306, K_{1i} = 42.4369$$

and for the second one

$$K_{2p} = -8.8456, K_{2i} = -48.287$$

Note that in F_{11} sign - shows the negative feedback. Because predicted output $y(t+1)$ is obtained from prediction model (44), for output feedback gain matrix F_{12} there is no need to use decentralized control structure

$$F_{12} = \begin{bmatrix} -22.0944 & 20.2891 & -10.1899 & 18.2789 \\ -29.3567 & 8.5697 & -28.7374 & -40.0299 \end{bmatrix}$$

In the second step of design procedure, using (48) for nominal model, the matrices (37) $F_{kk}, F_{kk+1}, k = 2, 3, 4, 5$ are calculated. The eigenvalues of closed-loop first vertex system model for $N_2 = N_u = 5$ are

$$Eig(Closed - loop) = \{-0.0009; -0.0087; 0.9789; 0.8815; 0.8925\}$$

Feasible solutions of bilinear matrix inequality have been obtained by YALMIP with PENBMI solver.

3. CONCLUSION

The first part of chapter addresses the problem of designing the output/state feedback robust model predictive controller with input constraints for output and control prediction horizons N_2 and N_u. The main contribution of the presented results is twofold: The obtained robust control algorithm guarantees the closed-loop system quadratic stability and guaranteed cost under input constraints in the whole uncertainty domain. The required on-line computation load is significantly less than in MPC literature (according to the best knowledge of authors), which opens possibility to use this control design scheme not only for plants with slow dynamics but also for faster ones. At each sample time the calculation of proposed control algorithm reduces to a solution of simple equation. Finally, two examples illustrate the effectiveness of the proposed method. The second part of chapter studies the problem of design

a new MPC with special control algorithm. The proposed robust MPC control algorithm is designed sequentially, the degree of plant model does not change when the output prediction horizon changes. The proposed sequential robust MPC design procedure consists of two steps: In the first step for one step ahead prediction horizon the necessary and sufficient robust stability conditions have been developed for MPC and the polytopic system with output feedback, using generalized parameter dependent Lyapunov matrix $P(\alpha)$. The proposed robust MPC ensures parameter dependent quadratic stability (PDQS) and guaranteed cost. In the second step of design procedure the uncertain plant and nominal model with sequential design approach is used to design the predicted input variables $u(t+1),...u(t+N_2-1)$ so that to ensure the robust closed-loop stability of MPC with guaranteed cost. Main advantages of the proposed sequential method are that the design plant model degree is independent on prediction horizon N_2; robust controller design procedure ensures PDQS and guaranteed cost and the obtained results are easy to be implemented in real plant. In the proposed design approach, constraints on system variables are easy to be implemented by LMI (BMI) using a notion of invariant set. Feasible solution of BMI has been obtained by Yalmip with PENBMI solver.

4. ACKNOWLEDGMENT

The work has been supported by Grant N 1/0544/09 of the Slovak Scientific Grant Agency.

5. References

Adamy, J. & Flemming, A. (2004) Soft variable-structure controls: a survey, *A*utomatica, 40, 1821-1844.

Ayd, H., Mesquine, F. & Aitrami, M. (2008) Robust control for uncertain linear systems with state and control constraints. In:*P*roc. of the 17th World Congress IFAC, Seoul, Korea, 2008, 1153-1158.

Bouzouita, B., Bouani, F. & Ksouri, M. (2007) Efficient Implementation of Multivariable MPC with Parametric Uncertainties, In:*P*roc. ECC 2007, Kos, Greece, TuB12.4, CD-ROM.

Camacho, E.F & Bordons, C. (2004)*Model predictive control*, *S*pringer-Verlag London Limited.

Casavola, A., Famularo, D. & Franze, G. (2004) Robust constrained predictive control of uncertain norm-bounded linear systems. *A*utomatica, 40, 1865-1776.

Clarke, D.W. & Mohtadi, C. (1989) Properties of generalized predictive control. *A*utomatica, 25(6), 859-875.

Clarke, D.W. & Scattolini, R.(1991) Constrained Receding-horizon Predictive Control. *P*roceedings IEE 138,(4), 347-354.

Dermicioglu, H. & Clarke, D.W. (1993) Generalized predictive control with end-point weighting. *I*EE Proc , 140, Part D(4): 275-282, 1993.

Ding, B., Xi, Y., Cychowski, M.T. & O'Mahony, T. (2008) A synthesis approach for output robust constrained model predictive control, *A*utomatica,44, 258-264.

Ebihara, Y., Peaucelle, D., Arzelier, D. & Hagivara, T. (2006) Robust H_2 Performance Analysis of Uncertain LTI Systems via Polynomially Parameter Dependent Lyapunov functions. In:*P*reprint of the 5th IFAC Symposium on Robust Control Design, ROCOND 06, Toulouse, France, July 5-7, 2006 CD-ROM.

Grman, L., Rosinová, D., Veselý, V. & Kozáková, A. (2005) Robust stability conditions for polytopic systems. *I*nt. Journal of Systems Science, Vol36, N15, 961-973.

Janík, M., Miklovicová, E. & Mrosko, M. (2008) Predictive control of nonlinear systems. *I*CIC Express Letters, Vol. 2, N3, 239-244.

Kothare, M.V., Balakrishnan, V, & Morari, M. (1996) Robust Constrained Model Predictive Control using Linear Matrix Inequalities, *A*utomatica , Vol 32, N10, 1361-1379.

Krokavec, D. & Filasová, A. (2003) Quadratically stabilized discrete-time robust LQ control. In:Control System Design , Proc. of the 2nd IFAC Conf., Bratislava, 375-380.

Kuwata, Y., Richards, A. & How, J. (2007) Robust Receding Horizon using Generalized Constraint Tightening, In:*P*roc. ACC, New Yourk, CD-ROM.

Lovas, Ch. , Seron, M.M. & Goodwin, G.C. (2007) Robust Model Predictive Control of Input-Constrained Stable Systems with Unstructured Uncertainty, In:*P*roc. ECC, Kos, Greece, CD-ROM.

Maciejovski, J.M. (2002) *Predictive Control with Constraints.* Prentice Hall

Mayne, D.Q., Rawlings, J.B.,Rao, C.V. & Scokaert, P.O.M. (2000) Contrained model predictive control: stability and optimality. *A*utomatica 36: 789-814.

de Oliveira, M.C., Camino, J.F. & Skelton, R.E. (2000) A convexifying algorithm for the design of structured linear controllers. In:*P*roc.39th IEEE Conference on Decision and Control, Sydney , 2781-2786.

Orukpe, P.E., Jaimoukha, I.M. & El-Zobaidi, H.M.H. (2007) Model Predictive Control Based on MixedH_2 / H_∞ Control Approach, In:*P*roc. ACC, New York July, 2007, CD-ROM.

Peaucelle, D., Arzelier, D., Bachelier, O. & Bernussou, J. (2000) A new robust D-stability condition for real convex polytopic uncertainty, *S*ystems and Control Letters, 40, 21-30.

delaPena, D.M., Alamo, T., Ramirez, T. & Camacho E. (2005) Min-max Model Predictive Control as a Quadratic Program, In: *P*roc. of 16th IFAC World Congress, Praga, CD-ROM.

Polak, E. & Yang, T.H. (1993) Moving horizon control of linear systems with input saturation and plant uncertainty, *I*nt. J. Control , 53, 613-638.

Rawlings, J. & Muske, K. (1993) The stability of constrained Receding Horizon Control. *I*EEE Trans. on Automatic Control 38, 1512-1516.

Rohal-Ilkiv, B. (2004) A note on calculation of polytopic invariant and feasible sets for linear continuous -time systems. *A*nnual Rewiew in Control , 28, 59-64.

Rosinová, D., Veselý, V. & Kučera, V. (2003) A necessary and sufficient condition for static output feedback stabilizability of linear discrete-time systems, *K*ybernetika, Vol39, N4, 447-459.

Rossiter, J.A. (2003) *Model Based Predictive Control: A Practical Approach*,Control Series.

Veselý, V., Rosinová, D. & Foltin, M. (2010) Robust model predictive control design with input constraints. *I*SA Transactions,49, 114-120.

Veselý, V. & Rosinová, D. (2009) Robust output model predictive control design : BMI approach, *I*JICIC Int. Vol 5, 4, 1115-1123.

Yanou, A., Inoue,A., Deng, M. & Masuda,S. (2008) An Extension of two Degree-of-freedom of Generalized Predictive Control for M-input M-output Systems Based on State Space Approach *I*JICIC, Vol4, N12, 3307-3318.

Zafiriou, E. & Marchal, A. (1991) Stability of SISO quadratic dynamic matrix control with hard output constraints, *A*IChE J. 37, 1550-1560.

Wang, Z., Chen, Z. Sun, Q. & Yuan, Z.(2006) GPC design technique based on MQFT for MIMO uncertain system, *I*nt. J. of Innovitive Computing, Information and Control, Vol2, N3, 519-526.

Zheng, Z.Q. & Morari, M. (1993) Robust Stability of Constrained Model Predictive Control, In *P*roc. ACC, San Francisco, CA, 379-383.

2

Robust Adaptive Model Predictive Control of Nonlinear Systems

Darryl DeHaan and Martin Guay
Dept. Chemical Engineering, Queen's University
Canada

1. Introduction

When faced with making a decision, it is only natural that one would aim to select the course of action which results in the "best" possible outcome. However, the ability to arrive at a decision necessarily depends upon two things: a well-defined notion of what qualities make an outcome desirable, and a *previous* decision[1] defining to what extent it is necessary to characterize the quality of individual candidates before making a selection (i.e., a notion of when a decision is "good enough"). Whereas the first property is required for the problem to be well defined, the later is necessary for it to be tractable.
The process of searching for the "best" outcome has been mathematically formalized in the framework of optimization. The typical approach is to define a scalar-valued *cost function*, that accepts a decision candidate as its argument, and returns a quantified measure of its quality. The decision-making process then reduces to selecting a candidate with the lowest (or highest) such measure.

1.1 The Emergence of Optimal Control

The field of "control" addresses the question of how to manipulate an *input* u in order to drive the *state* x of a dynamical system

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

to some desired target. Ultimately this task can be viewed as decision-making, so it is not surprising that it lends itself towards an optimization-based characterization. Assuming that one can provide the necessary metric for assessing quality of the trajectories generated by (1), there exists a rich body of "optimal control" theory to guide this process of decision-making. Much of this theory came about in the 1950's and 60's, with Pontryagin's introduction of the Minimum (a.k.a. Maximum) Principle Pontryagin (1961), and Bellman's development of Dynamic Programming Bellman (1952; 1957). (This development also coincided with landmark results for linear systems, pioneered by Kalman Kalman (1960; 1963), that are closely related). However, the roots of both approaches actually extend back to the mid-1600's, with the inception of the calculus of variations.

[1] The recursiveness of this definition is of course ill-posed until one accepts that at some level, every decision is ultimately predicated upon underlying assumptions, accepted entirely in faith.

The tools of optimal control theory provide useful benchmarks for characterizing the notion of “best" decision-making, as it applies to control. However applied directly, the tractability of this decision-making is problematic. For example, Dynamic Programming involves the construction of a $n-$dimensional surface that satisfies a challenging nonlinear partial differential equation, which is inherently plagued by the so-called *curse of dimensionality*. This methodology, although elegant, remains generally intractable for problems beyond modest size. In contrast, the Minimum Principle has been relatively successful for use in off-line trajectory planning, when the initial condition of (1) is known. Although it was suggested as early as 1967 in Lee & Markus (1967) that a stabilizing feedback $u = k(x)$ could be constructed by continuously re-solving the calculations online, a tractable means of doing this was not immediately forthcoming.

1.2 Model Predictive Control as Receding-Horizon Optimization

Early development (Richalet et al. (1976),Richalet et al. (1978),Cutler & Ramaker (1980)) of the control approach known today as Model Predictive Control (MPC) originated in the process control community, and was driven much more by industrial application than by theoretical understanding. Modern theoretical understanding of MPC, much of which developed throughout the 1990's, has clarified its very natural ties to existing optimal control theory. Key steps towards this development included such results as Chen & Allgöwer (1998a;b); De Nicolao et al. (1996); Jadbabaie et al. (2001); Keerthi & Gilbert (1988); Mayne & Michalska (1990); Michalska & Mayne (1993); Primbs et al. (2000), with an excellent unifying survey in Mayne et al. (2000).

At its core, MPC is simply a framework for implementing existing tools of optimal control. Taking the current value $x(t)$ as the initial condition for (1), the Minimum Principle is used as the primary basis for identifying the “best" candidate trajectory by predicting the future behaviour of the system using model (1). However, the actual quality measure of interest in the decision-making is generally the total future accumulation (i.e., over an infinite future) of a given instantaneous metric, a quantity rarely computable in a satisfactorily short time. As such, MPC only generates predictions for (1) over a finite time-horizon, and approximates the remaining infinite tail of the cost accumulation using a penalty surface derived from either a *local* solution of the Dynamic Programming surface, or an appropriate approximation of that surface. As such, the key benefit of MPC over other optimal control methods is simply that its finite horizon allows for a convenient trade-off between the online computational burden of solving the Minimum Principle, and the offline burden of generating the penalty surface.

In contrast to other approaches for constructive nonlinear controller design, optimal control frameworks facilitate the inclusion of constraints, by imposing feasibility of the candidates as a condition in the decision-making process. While these approaches can be numerically burdensome, optimal control (and by extension, MPC) provides the only real framework for addressing the control of systems in the presence of constraints - in particular those involving the state x. In practice, the predictive aspect of MPC is unparalleled in its ability to account for the risk of future constraint violation during the current control decision.

1.3 Current Limitations in Model Predictive Control

While the underlying theoretical basis for model predictive control is approaching a state of relative maturity, application of this approach to date has been predominantly limited to “slow" industrial processes that allow adequate time to complete the controller calculations. There is great incentive to extend this approach to applications in many other sectors, moti-

vated in large part by its constraint-handling abilities. Future applications of significant interest include many in the aerospace or automotive sectors, in particular constraint-dominated problems such as obstacle avoidance. At present, the significant computational burden of MPC remains the most critical limitation towards its application in these areas.
The second key weakness of the model predictive approach remains its susceptibility to uncertainties in the model (1). While a fairly well-developed body of theory has been developed within the framework of robust-MPC, reaching an acceptable balance between computational complexity and conservativeness of the control remains a serious problem. In the more general control literature, adaptive control has evolved as an alternative to a robust-control paradigm. However, the incorporation of adaptive techniques into the MPC framework has remained a relatively open problem.

2. Notational and Mathematical Preliminaries

Throughout the remainder of this dissertation, the following is assumed by default (where $s \in \mathbb{R}^{\mathrm{s}}$ and $\mathbb{S}$ represent arbitrary vectors and sets, respectively):

- all vector norms are Euclidean, defining balls $B(s,\delta) \triangleq \{s' \mid \|s - s'\| \leq \delta\}$, $\delta \geq 0$.
- norms of matrices $\mathcal{S} \in \mathbb{R}^{m \times \mathrm{s}}$ are assumed induced as $\|\mathcal{S}\| \triangleq \max_{\|s\|=1} \|\mathcal{S}s\|$.
- the notation $s_{[a,b]}$ denotes the entire continuous-time trajectory $s(\tau)$, $\tau \in [a,b]$, and likewise $\dot{s}_{[a,b]}$ the trajectory of its forward derivative $\dot{s}(\tau)$.
- For any set $\mathbb{S} \subseteq \mathbb{R}^{\mathrm{s}}$, define
  - i) its closure $\mathrm{cl}\{\mathbb{S}\}$, interior $\mathring{\mathbb{S}}$, and boundary $\partial\mathbb{S} = \mathrm{cl}\{\mathbb{S}\} \setminus \mathring{\mathbb{S}}$
  - ii) its orthogonal distance norm $\|s\|_{\mathbb{S}} \triangleq \inf_{s' \in \mathbb{S}} \|s - s'\|$
  - iii) a closed δ-neighbourhood $B(\mathbb{S},\delta) \triangleq \{s \in \mathbb{R}^{\mathrm{s}} \mid \|s\|_{\mathbb{S}} \leq \delta\}$
  - iv) an interior approximation $\overleftarrow{B}(\mathbb{S},\delta) \triangleq \{s \in \mathbb{S} \mid \inf_{s' \in \partial\mathbb{S}} \|s - s'\| \geq \delta\}$
  - v) a (finite, closed, open) cover of $\mathbb{S}$ as any (finite) collection $\{\mathbb{S}^i\}$ of (open, closed) sets $\mathbb{S}^i \subseteq \mathbb{R}^{\mathrm{s}}$ such that $\mathbb{S} \subseteq \cup_i \mathbb{S}^i$.
  - vi) the maximal closed subcover $\overline{\mathrm{cov}}\,\{\mathbb{S}\}$ as the infinite collection $\{\mathbb{S}^i\}$ containing *all possible* closed subsets $\mathbb{S}^i \subseteq \mathbb{S}$; i.e., $\overline{\mathrm{cov}}\,\{\mathbb{S}\}$ is a maximal "set of subsets".

Furthermore, for any arbitrary function $\alpha : \mathbb{S} \to \mathbb{R}$ we assume the following definitions:

- $\alpha(\cdot)$ is C^{m+} if it is at least m-times differentiable, with all derivatives of order m yielding locally Lipschitz functions.
- A function $\alpha : \mathbb{S} \to (-\infty, \infty]$ is *lower semi-continuous* ($\mathcal{LS}$-continuous) at s if it satisfies (see Clarke et al. (1998)):
$$\liminf_{s' \to s} \alpha(s') \geq \alpha(s) \tag{2}$$
- a continuous function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ belongs to *class* $\mathcal{K}$ if $\alpha(0) = 0$ and $\alpha(\cdot)$ is strictly increasing on $\mathbb{R}_{>0}$. It belongs to *class* $\mathcal{K}_\infty$ if it is furthermore radially unbounded.
- a continuous function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ belongs to *class* $\mathcal{KL}$ if i) for every fixed value of τ, it satisfies $\beta(\cdot,\tau) \in \mathcal{K}$, and ii) for each fixed value of s, then $\beta(s,\cdot)$ is strictly decreasing and satisfies $\lim_{\tau \to \infty} \beta(s,\tau) = 0$.
- the scalar operator $\mathrm{sat}_a^b(\cdot)$ denotes saturation of its arguments onto the interval $[a,b]$, $a < b$. For vector- or matrix-valued arguments, the saturation is presumed by default to be evaluated element-wise.

3. Brief Review of Optimal Control

The underlying assumption of optimal control is that at any time, the pointwise cost of x and u being away from their desired targets is quantified by a known, physically-meaningful function $L(x,u)$. Loosely, the goal is to then reach some target in a manner that accumulates the least cost. It is not generally necessary for the "target" to be explicitly described, since its knowledge is built into the function $L(x,u)$ (i.e it is assumed that convergence of x to any invariant subset of $\{x \mid \exists u \quad s.t.\ L(x,u) = 0\}$ is as acceptable). The following result, while superficially simple in appearance, is in fact the key foundation underlying the optimal control results of this section, and by extension all of model predictive control as well. Proof can be found in many references, such as Sage & White (1977).

Definition 3.1 (Principle of Optimality:). *If $u^*_{[t_1,t_2]}$ is an optimal trajectory for the interval $t \in [t_1,t_2]$, with corresponding solution $x^*_{[t_1,t_2]}$ to* (1)*, then for any $\tau \in (t_1,\, t_2)$ the sub-arc $u^*_{[\tau,\,t_2]}$ is necessarily optimal for the interval $t \in [\tau,t_2]$ if* (1) *starts from $x^*(\tau)$.*

4. Variational Approach: Euler, Lagrange and Pontryagin

Pontryagin's Minimum principle (also known as the Maximum principle, Pontryagin (1961)) represented a landmark extension of classical ideas of variational calculus to the problem of control. Technically, the Minimum Principle is an application of the classical Euler-Lagrange and Weierstrass conditions[2] Hestenes (1966), which provide first-order *necessary* conditions to characterize extremal time-trajectories of a cost functional.[3]. The Minimum Principle therefore characterizes minimizing trajectories $(x_{[0,T]},\, u_{[0,T]})$ corresponding to a constrained finite-horizon problem of the form

$$V_T(x_0,u_{[0,T]}) = \int_0^T L(x,u)\,d\tau + W(x(T)) \tag{3a}$$

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

$$\dot{x} = f(x,u), \qquad x(0) = x_0 \tag{3b}$$

$$g(x(\tau)) \le 0, \qquad h(x(\tau),u(\tau)) \le 0, \qquad w(x(T)) \le 0 \tag{3c}$$

where the vectorfield $f(\cdot,\cdot)$ and constraint functions $g(\cdot)$, $h(\cdot,\cdot)$, and $w(\cdot)$ are assumed sufficiently differentiable.

Assume that $g(x_0) < 0$, and, for a given $(x_0,u_{[0,T]})$, let the interval $[0,T)$ be partitioned into (maximal) subintervals as $\tau \in \cup_{i=1}^{p}[t_i,t_{i+1})$, $t_0 = 0$, $t_{p+1} = T$, where the interior t_i represent intersections $g < 0 \Leftrightarrow g = 0$ (i.e., the $\{t_i\}$ represent changes in the active set of g). Assuming that $g(x)$ has constant relative degree r over some appropriate neighbourhood, define the following vector of (Lie) derivatives: $N(x) \triangleq [g(x), g^{(1)}(x), \dots g^{(r-1)}(x)]^T$, which characterizes additional tangency constraints $N(x(t_i)) = 0$ at the corners $\{t_i\}$. Rewriting (3) in multiplier form

$$V_T = \int_0^T \mathcal{H}(x,u) - \lambda^T \dot{x}\,d\tau + W(x(T)) + \mu_w w(x(T)) + \sum_i \mu_N^T(t_i) N(x(t_i)) \tag{4a}$$

$$\mathcal{H} \triangleq L(x,u) + \lambda^T f(x,u) + \mu_h h(x,u) + \mu_g g^{(r)}(x,u) \tag{4b}$$

[2] phrased as a fixed initial point, free endpoint problem

[3] i.e., generalizing the NLP necessary condition $\frac{\partial p}{\partial x} = 0$ for the extrema of a function $p(x)$.

overa Taking the first variation of the right-hand sides of (4a,b) with respect to perturbations in $x_{[0,T]}$ and $u_{[0,T]}$ yields the following set of conditions (adapted from statements in Bertsekas (1995); Bryson & Ho (1969); Hestenes (1966)) which necessarily must hold for V_T to be minimized:

Proposition 4.1 (Minimum Principle)**.** *Suppose that the pair* $(u^*_{[0,T]},\, x^*_{[0,T]})$ *is a minimizing solution of* (3)*. Then for all* $\tau \in [0,T]$*, there exists multipliers* $\lambda(\tau) \geq 0$*,* $\mu_h(\tau) \geq 0$*,* $\mu_g(\tau) \geq 0$*, and constants* $\mu_w \geq 0$*,* $\mu^i_N \geq 0$*,* $i \in \mathcal{I}$*, such that*

i) Over each interval $\tau \in [t_i,\, t_{i+1}]$*, the multipliers* $\mu_h(\tau)$*,* $\mu_g(\tau)$ *are piecewise continuous,* $\mu_N(\tau)$ *is constant,* $\lambda(\tau)$ *is continuous, and with* $(u^*_{[t_i,\,t_{i+1}]},\, x^*_{[t_i,\,t_{i+1}]})$ *satisfies*

$$\dot{x}^* = f(x^*, u^*), \qquad x^*(0) = x_0 \tag{5a}$$

$$\dot{\lambda}^T = \nabla_x \mathcal{H} \quad \textit{a.e., with} \quad \lambda^T(T) = \nabla_x W(x^*(T)) + \mu_w \nabla_x w(x^*(T)) \tag{5b}$$

where the solution $\lambda_{[0,T]}$ *is discontinuous at* $\tau \in \{t_i\}$*,* $i \in \{1, 3, 5...p\}$*, satisfying*

$$\lambda^T(t_i^-) = \lambda^T(t_i^+) + \mu_N^T(t_i^+) \nabla_x N(x(t_i)) \tag{5c}$$

ii) $\mathcal{H}(x^*, u^*, \lambda, \mu_h, \mu_g)$ *is constant over intervals* $\tau \in [t_i,\, t_{i+1}]$*, and for all* $\tau \in [0,T]$ *it satisfies (where* $\mathcal{U}(x) \triangleq \{u \,|\, h(x,u) \leq 0$ *and* $\Big(g^{(r)}(x,u) \leq 0$ *if* $g(x) = 0\Big)\}$*):*

$$\mathcal{H}(x^*, u^*, \lambda, \mu_h, \mu_g) \leq \min_{u \in \mathbb{U}(x)} \mathcal{H}(x^*, u, \lambda, \mu_h, \mu_g) \tag{5d}$$

$$\nabla_u \mathcal{H}(x^*(\tau), u^*(\tau), \lambda(\tau), \mu_h(\tau), \mu_g(\tau)) = 0 \tag{5e}$$

iii) For all $\tau \in [0,T]$*, the following constraint conditions hold*

$$g(x^*) \leq 0 \qquad h(x^*, u^*) \leq 0 \qquad w(x^*(T)) \leq 0 \tag{5f}$$

$$\mu_g(\tau) g^{(r)}(x^*, u^*) = 0 \qquad \mu_h(\tau) h(x^*, u^*) = 0 \qquad \mu_w w(x^*(T)) = 0 \tag{5g}$$

$$\mu_N^T(\tau) N(x^*) = 0 \qquad \Big(\textit{and } N(x^*) = 0, \;\; \forall \tau \in [t_i,\, t_{i+1}], \;\; i \in \{1, 3, 5...p\}\Big) \tag{5h}$$

The multiplier $\lambda(t)$ is called the *co-state*, and it requires solving a two-point boundary-value problem for (5a) and (5b). One of the most challenging aspects to locating (and confirming) a minimizing solution to (5) lies in dealing with (5c) and (5h), since the number and times of constraint intersections are not known a-priori.

5. Dynamic Programming: Hamilton, Jacobi, and Bellman

The Minimum Principle is fundamentally based upon establishing the optimality of a *particular* input trajectory $u_{[0,T]}$. While the applicability to offline, open-loop trajectory planning is clear, the inherent assumption that x_0 is known can be limiting if one's goal is to develop a *feedback* policy $u = k(x)$. Development of such a policy requires the consideration of *all possible* initial conditions, which results in an optimal cost *surface* $J^* : \mathbb{R}^n \to \mathbb{R}$, with an associated control policy $k : \mathbb{R}^n \to \mathbb{R}^m$. A constructive approach for calculating such a surface, referred to as *Dynamic Programming*, was developed by Bellman Bellman (1957). Just as the

Minimum Principle was extended out of the classical trajectory-based Euler-Lagrange equations, Dynamic Programming is an extension of classical Hamilton-Jacobi field theory from the calculus of variations.
For simplicity, our discussion here will be restricted to the unconstrained problem:

$$V^*(x_0) = \min_{u_{[0,\infty)}} \int_0^\infty L(x,u)\, d\tau \tag{6a}$$

$$s.t. \quad \dot{x} = f(x,u), \qquad x(0) = x_0 \tag{6b}$$

with locally Lipschitz dynamics $f(\cdot,\cdot)$. From the Principle of Optimality, it can be seen that (6) lends itself to the following recursive definition:

$$V^*(x(t)) = \min_{u[t,\, t+\Delta t]} \left\{ \int_t^{t+\Delta t} L(x(\tau), u(\tau)) d\tau + V^*(x(t+\Delta t)) \right\} \tag{7}$$

Assuming that V^* is differentiable, replacing $V^*(x(t+\Delta t)$ with a first-order Taylor-series and the integrand with a Riemannian sum, the limit $\Delta t \to 0$ yields

$$0 = \min_u \left\{ L(x,u) + \frac{\partial V^*}{\partial x} f(x,u) \right\} \tag{8}$$

Equation (8) is one particular form of what is known as the Hamilton-Jacobi-Bellman (HJB) equation. In some cases (such as $L(x,u)$ quadratic in u, and $f(x,u)$ affine in u), (8) can be simplified to a more standard-looking PDE by evaluating the indicated minimization in closed-form[4]. Assuming that a (differentiable) surface $V^* : \mathbb{R}^n \to \mathbb{R}$ is found (generally by off-line numerical solution) which satisfies (8), a stabilizing feedback $u = k_{DP}(x)$ can be constructed from the information contained in the surface V^* by simply defining[5] $k_{DP}(x) \triangleq \{u \mid \frac{\partial V^*}{\partial x} f(x,u) = -L(x,u)\}$.
Unfortunately, incorporation of either input or state constraints generally violates the assumed smoothness of $V^*(x)$. While this could be handled by interpreting (8) in the context of *viscosity solutions* (see Clarke et al. (1998) for definition), for the purposes of application to model predictive control it is more typical to simply restrict the domain of $V^* : \Omega \to \mathbb{R}$ such that $\Omega \subset \mathbb{R}^n$ is feasible with respect to the constraints.

6. Inverse-Optimal Control Lyapunov Functions

While knowledge of a surface $V^*(x)$ satisfying (8) is clearly ideal, in practice analytical solutions are only available for extremely restrictive classes of systems, and almost never for systems involving state or input constraints. Similarly, numerical solution of (8) suffers the so-called "curse of dimensionality" (as named by Bellman) which limits its applicability to systems of restrictively small size.
An alternative design framework, originating in Sontag (1983), is based on the following:

Definition 6.1. *A* control Lyapunov function (CLF) *for* (1) *is any* C^1*, proper, positive definite function* $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ *such that, for all* $x \neq 0$*:*

$$\inf_u \quad \frac{\partial V}{\partial x} f(x,u) < 0 \tag{9}$$

[4] In fact, for linear dynamics and quadratic cost, (8) reduces down to the linear Ricatti equation.
[5] $k_{DP}(\cdot)$ is interpreted to incorporate a deterministic selection in the event of multiple solutions. The existence of such a u is implied by the assumed solvability of (8)

Design techniques for deriving a feedback $u = k(x)$ from knowledge of $V(\cdot)$ include the well-known "Sontag's Controller" of Sontag (1989), which led to the development of "Pointwise Min-Norm" control of the form Freeman & Kokotović (1996a;b); Sepulchre et al. (1997):

$$\min_u \gamma(u) \quad s.t. \quad \frac{\partial V}{\partial x} f(x,u) < -\sigma(x) \tag{10}$$

where γ, σ are positive definite, and γ is radially unbounded. As discussed in Freeman & Kokotović (1996b); Sepulchre et al. (1997), relation (9) implies that *there exists* a function $L(x,u)$, derived from γ and σ, for which $V(\cdot)$ satisfies (8). Furthermore, if $V(x) \equiv V^*(x)$, then appropriate selection of γ, σ (in particular that of Sontag's controller Sontag (1989)) results in the feedback $u = k_{clf}(x)$ generated by (9) satisfying $k_{clf}(\cdot) \equiv k_{DP}(\cdot)$. Hence this technique is commonly referred to as "inverse-optimal" control design, and can be viewed as a method for approximating the optimal control problem (6) by replacing $V^*(x)$ directly.

7. Review of Nonlinear MPC based on Nominal Models

The ultimate objective of a model predictive controller is to provide a *closed-loop feedback* $u = \kappa_{mpc}(x)$ that regulates (1) to its target set (assumed here $x = 0$) in a fashion that is optimal with respect to the *infinite-time* problem (6), while enforcing pointwise constraints of the form $(x,u) \in \mathbb{X} \times \mathbb{U}$ in a constructive manner. However, rather than defining the map $\kappa_{mpc} : \mathbb{X} \to \mathbb{U}$ by solving a PDE of the form (8) (i.e thereby pre-computing knowledge of $\kappa_{mpc}(x)$ for *every* $x \in \mathbb{X}$), the model predictive control philosophy is to solve for, at time t, the control move $u = \kappa_{mpc}(x(t))$ for the *particular* value $x(t) \in \mathbb{X}$. This makes the online calculations inherently trajectory-based, and therefore closely tied to the results in Section 4 (with the caveat that the initial conditions are continuously referenced relative to current (t,x)). Since it is not practical to pose (online) trajectory-based calculations over an infinite prediction horizon $\tau \in [t,\infty)$, a truncated prediction $\tau \in [t,\, t+T]$ is used instead. The truncated tail of the integral in (6) is replaced by a (designer-specified) terminal penalty $W : \mathbb{X}_f \to \mathbb{R}_{\geq 0}$, defined over any local neighbourhood $\mathbb{X}_f \subset \mathbb{X}$ of the target $x = 0$. This results in a feedback of the form:

$$u = \kappa_{mpc}(x(t)) \triangleq u^*_{[t,\,t+T]}(t) \tag{11a}$$

where $u^*_{[t,\,t+T]}$ denotes the solution to the $x(t)$-dependent problem:

$$u^*_{[t,\,t+T]} \triangleq \arg\min_{u^p_{[t,\,t+T]}} \left(V_T(x(t), u^p_{[t,\,t+T]}) \triangleq \int_t^{t+T} L(x^p, u^p)\, d\tau + W(x^p(t+T)) \right) \tag{11b}$$

$$s.t.\ \forall \tau \in [t, t+T]: \quad \tfrac{d}{d\tau} x^p = f(x^p, u^p), \quad x^p(t) = x(t) \tag{11c}$$

$$(x^p(\tau), u^p(\tau)) \in \mathbb{X} \times \mathbb{U} \tag{11d}$$

$$x^p(t+T) \in \mathbb{X}_f \tag{11e}$$

Clearly, if one could define $W(x) \equiv V^*(x)$ globally, then the feedback in (11) must satisfy $\kappa_{mpc}(\cdot) \equiv k_{DP}(\cdot)$. While $W(x) \equiv V^*(x)$ is generally unachievable, this motivates the selection of $W(x)$ as a CLF such that $W(x)$ is an inverse-optimal approximation of $V^*(x)$. A more precise characterization of the selection of $W(x)$ is the focus of the next section.

8. General Sufficient Conditions for Stability

A very general proof of the closed-loop stability of (11), which unifies a variety of earlier, more restrictive, results is presented[6] in the survey Mayne et al. (2000). This proof is based upon the following set of sufficient conditions for closed-loop stability:

Criterion 8.1. *The function $W : \mathbb{X}_f \to \mathbb{R}_{\geq 0}$ and set $\mathbb{X}_f$ are such that a local feedback $k_f : \mathbb{X}_f \to \mathbb{U}$ exists to satisfy the following conditions:*

C1) $0 \in \mathbb{X}_f \subseteq \mathbb{X}$, $\mathbb{X}_f$ *closed (i.e., state constraints satisfied in* $\mathbb{X}_f$*)*

C2) $k_f(x) \in \mathbb{U}, \forall x \in \mathbb{X}_f$ *(i.e., control constraints satisfied in* $\mathbb{X}_f$*)*

C3) $\mathbb{X}_f$ *is positively invariant for* $\dot{x} = f(x, k_f(x))$.

C4) $L(x, k_f(x)) + \frac{\partial W}{\partial x} f(x, k_f(x)) \leq 0, \quad \forall x \in \mathbb{X}_f$.

Only existence, not knowledge, of $k_f(x)$ is assumed. Thus by comparison with (9), it can be seen that C4 essentially requires that $W(x)$ be a CLF over the (local) domain $\mathbb{X}_f$, in a manner consistent with the constraints.

In hindsight, it is nearly obvious that closed-loop stability can be reduced entirely to conditions placed upon only the terminal choices $W(\cdot)$ and $\mathbb{X}_f$. Viewing $V_T(x(t), u^*_{[t,t+T]})$ as a Lyapunov function candidate, it is clear from (3) that V_T contains "energy" in both the $\int L\, d\tau$ and terminal W terms. Energy dissipates from the front of the integral at a rate $L(x, u)$ as time t flows, and by the principle of optimality one could implement (11) on a *shrinking* horizon (i.e., $t + T$ constant), which would imply $\dot{V} = -L(x, u)$. In addition to this, C4 guarantees that the energy transfer from W to the integral (as the point $t + T$ recedes) will be non-increasing, and could even dissipate additional energy as well.

9. Robustness Considerations

As can be seen in Proposition 4.1, the presence of inequality constraints on the state variables poses a challenge for numerical solution of the optimal control problem in (11). While locating the times $\{t_i\}$ at which the active set changes can itself be a burdensome task, a significantly more challenging task is trying to guarantee that the tangency condition $N(x(t_{i+1})) = 0$ is met, which involves determining if x lies on (or crosses over) the critical surface beyond which this condition fails.

As highlighted in Grimm et al. (2004), this critical surface poses more than just a computational concern. Since both the cost function and the feedback $\kappa_{mpc}(x)$ are potentially discontinuous on this surface, there exists the potential for *arbitrarily small* disturbances (or other plant-model mismatch) to compromise closed-loop stability. This situation arises when the optimal solution $u^*_{[t,t+T]}$ in (11) switches between disconnected minimizers, potentially resulting in invariant limit cycles (for example, as a very low-cost minimizer alternates between being judged feasible/infeasible.)

A modification suggested in Grimm et al. (2004) to restore nominal robustness, similar to the idea in Marruedo et al. (2002), is to replace the constraint $x(\tau) \in \mathbb{X}$ of (11d) with one of the form $x(\tau) \in \mathbb{X}^o(\tau - t)$, where the function $\mathbb{X}^o : [0, T] \to \mathbb{X}$ satisfies $\mathbb{X}^o(0) = \mathbb{X}$, and the strict containment $\mathbb{X}^o(t_2) \subset \mathbb{X}^o(t_1)$, $t_1 < t_2$. The gradual relaxation of the constraint limit as future predictions move closer to current time provides a safety margin that helps to avoid constraint violation due to small disturbances.

[6] in the context of both continuous- and discrete-time frameworks

The issue of robustness to measurement error is addressed in Tuna et al. (2005). On one hand, nominal robustness to measurement noise of an MPC feedback was already established in Grimm et al. (2003) for discrete-time systems, and in Findeisen et al. (2003) for sampled-data implementations. However, Tuna et al. (2005) demonstrates that as the sampling frequency becomes arbitrarily fast, the margin of this robustness may approach zero. This stems from the fact that the feedback $\kappa_{mpc}(x)$ of (11) is inherently discontinuous in x if the indicated minimization is performed globally on a nonconvex surface, which by Coron & Rosier (1994); Hermes (1967) enables a fast measurement dither to generate flow in any direction contained in the convex hull of the discontinuous closed-loop vectorfield. In other words, additional attractors or unstable/infeasible modes can be introduced into the closed-loop behaviour by arbitrarily small measurement noise.
Although Tuna et al. (2005) deals specifically with situations of obstacle avoidance or stabilization to a target set containing disconnected points, other examples of problematic nonconvexities are depicted in Figure 1. In each of the scenarios depicted in Figure 1, measurement dithering could conceivably induce flow along the dashed trajectories, thereby resulting in either constraint violation or convergence to an undesired equilibrium.
Two different techniques were suggested in Tuna et al. (2005) for restoring robustness to the measurement error, both of which involve adding a hysteresis-type behaviour in the optimization to prevent arbitrary switching of the solution between separate minimizers (i.e., making the optimization behaviour more decisive).

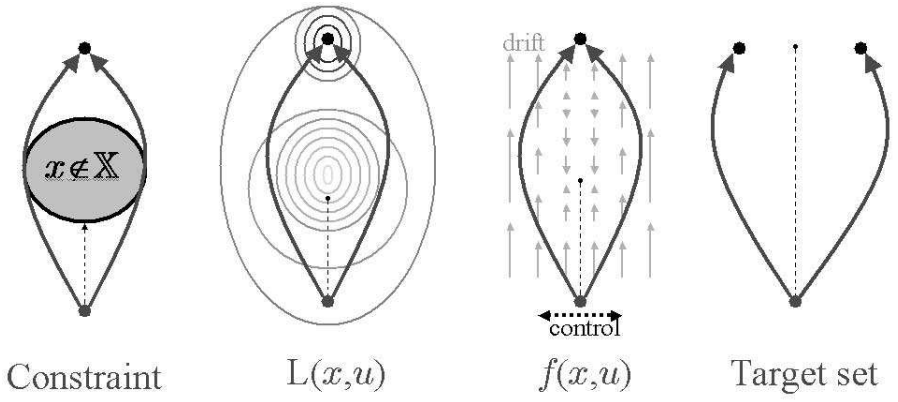


Fig. 1. Examples of nonconvexities susceptible to measurement error

10. Robust MPC

10.1 Review of Nonlinear MPC for Uncertain Systems

While a vast majority of the robust-MPC literature has been developed within the framework of discrete-time systems[7], for consistency with the rest of this thesis most of the discussion will be based in terms of their continuous-time analogues. The uncertain system model is

[7] Presumably for numerical tractability, as well as providing a more intuitive link to game theory.

therefore described by the general form

$$\dot{x} = f(x, u, d) \tag{12}$$

where $d(t)$ represents any arbitrary $\mathcal{L}_\infty$-bounded disturbance signal, which takes point-wise[8] values $d \in \mathcal{D}$. Equivalently, (12) can be represented as the differential inclusion model $\dot{x} \in F(x, u) \triangleq f(x, u, \mathcal{D})$.

In the next two sections, we will discuss approaches for accounting explicitly for the disturbance in the online MPC calculations. We note that significant effort has also been directed towards various means of increasing the inherent robustness of the controller without requiring explicit online calculations. This includes the suggestion in Magni & Sepulchre (1997) (with a similar discrete-time idea in De Nicolao et al. (1996)) to use a modified stage cost $\overline{L}(x, u) \triangleq L(x, u) + \langle \nabla_x V_T^*(x),\ f(x, u) \rangle$ to increase the robustness of a nominal-model implementation, or the suggestion in Kouvaritakis et al. (2000) to use an prestabilizer, optimized offline, of the form $u = Kx + v$ to reduced online computational burden. Ultimately, these approaches can be considered encompassed by the banner of nominal-model implementation.

10.1.1 Explicit robust MPC using Open-loop Models

As seen in the previous chapters, essentially all MPC approaches depend critically upon the Principle of Optimality (Def 3.1) to establish a proof of stability. This argument depends inherently upon the assumption that the predicted trajectory $x^p_{[t,\, t+T]}$ is an invariant set under open-loop implementation of the corresponding $u^p_{[t,\, t+T]}$; i.e., that the prediction model is "perfect". Since this is no longer the case in the presence of plant-model mismatch, it becomes necessary to associate with $u^p_{[t,\, t+T]}$ a *cone* of trajectories $\{x^p_{[t,\, t+T]}\}_{\mathcal{D}}$ emanating from $x(t)$, as generated by (12).

Not surprisingly, establishing stability requires a strengthening of the conditions imposed on the selection of the terminal cost W and domain $\mathbb{X}_f$. As such, W and $\mathbb{X}_f$ are assumed to satisfy Criterion (8.1), but with the revised conditions:

C3a) $\mathbb{X}_f$ is strongly positively invariant for $\dot{x} \in f(x, k_f(x), \mathcal{D})$.

C4a) $L(x, k_f(x)) + \frac{\partial W}{\partial x} f(x, k_f(x), d) \leq 0, \quad \forall (x, d) \in \mathbb{X}_f \times \mathcal{D}$.

While the original C4 had the interpretation of requiring W to be a CLF for the nominal system, so the revised C4a can be interpreted to imply that W should be a robust-CLF like those developed in Freeman & Kokotović (1996b).

Given such an appropriately defined pair $(W, \mathbb{X}_f)$, the model predictive controller explicitly considers all trajectories $\{x^p_{[t,\, t+T]}\}_{\mathcal{D}}$ by posing the modified problem

$$u = \kappa_{mpc}(x(t)) \triangleq u^*_{[t,\, t+T]}(t) \tag{13a}$$

where the trajectory $u^*_{[t,\, t+T]}$ denotes the solution to

$$u^*_{[t,\, t+T]} \triangleq \arg \min_{\substack{u^p_{[t,\, t+T]} \\ T \in [0, T_{max}]}} \left(\max_{d_{[t,\, t+T]} \in \mathcal{D}} V_T(x(t), u^p_{[t,\, t+T]}, d_{[t,\, t+T]}) \right) \tag{13b}$$

[8] The abuse of notation $d_{[t_1,\, t_2]} \in \mathcal{D}$ is likewise interpreted pointwise

The function $V_T(x(t), u^p_{[t,\,t+T]}, d_{[t,\,t+T]})$ appearing in (13) is as defined in (11), but with (11c) replaced by (12). Variations of this type of design are given in Chen et al. (1997); Lee & Yu (1997); Mayne (1995); Michalska & Mayne (1993); Ramirez et al. (2002), differing predominantly in the manner by which they select $W(\cdot)$ and $\mathbb{X}_f$.
If one interprets the word "optimal" in Definition 3.1 in terms of the worst-case trajectory in the optimal cone $\{x^p_{[t,\,t+T]}\}^*_{\mathcal{D}}$, then at time $\tau \in [t,\ t+T]$ there are only two possibilities:

- the actual $x_{[t,\tau]}$ matches the subarc from a worst-case element of $\{x^p_{[t,\,t+T]}\}^*_{\mathcal{D}}$, in which case the Principle of Optimality holds as stated.
- the actual $x_{[t,\tau]}$ matches the subarc from an element in $\{x^p_{[t,\,t+T]}\}^*_{\mathcal{D}}$ which was *not* the worst case, so implementing the remaining $u^*_{[\tau,\,t+T]}$ will achieve overall less cost than the worst-case estimate at time t.

One will note however, that the bound guaranteed by the principle of optimality applies only to the remaining subarc $[\tau,\ t+T]$, and says nothing about the ability to extend the horizon. For the nominal-model results of Chapter 7, the ability to extend the horizon followed from C4) of Criterion (8.1). In the present case, C4a) guarantees that for *each* terminal value $\{x^p_{[t,\,t+T]}(t+T)\}^*_{\mathcal{D}}$ there exists a value of u rendering W decreasing, but not necessarily a single such value satisfying C4a) for *every* $\{x^p_{[t,\,t+T]}(t+T)\}^*_{\mathcal{D}}$. Hence, receding of the horizon can only occur at the discretion of the optimizer. In the *worst* case, T could contract (i.e., $t+T$ remains fixed) until eventually $T = 0$, at which point $\{x^p_{[t,\,t+T]}(t+T)\}^*_{\mathcal{D}} \equiv x(t)$, and therefore by C4a) an appropriate extension of the "trajectory" $u^*_{[t,t]}$ exists.
Although it is not an explicit min-max type result, the approach in Marruedo et al. (2002) makes use of global Lipschitz constants to determine a bound on the the worst-case distance between a solution of the uncertain model (12), and that of the underlying nominal model estimate. This Lipschitz-based uncertainty cone expands at the fastest-possible rate, necessarily containing the actual uncertainty cone $\{x^p_{[t,\,t+T]}\}_{\mathcal{D}}$. Although ultimately just a nominal-model approach, it is relevant to note that it can be viewed as replacing the "max" in (13) with a simple worst-case upper bound.
Finally, we note that many similar results Cannon & Kouvaritakis (2005); Kothare et al. (1996) in the linear robust-MPC literature are relevant, since nonlinear dynamics can often be approximated using uncertain linear models. In particular, linear systems with polytopic descriptions of uncertainty are one of the few classes that can be realistically solved numerically, since the calculations reduce to simply evaluating each node of the polytope.

10.1.2 Explicit robust MPC using Feedback Models

Given that robust control design is closely tied to game theory, one can envision (13) as representing a player's decision-making process throughout the evolution of a strategic game. However, it is unlikely that a player even moderately-skilled at such a game would restrict themselves to preparing only a single *sequence of moves* to be executed in the future. Instead, a skilled player is more likely to prepare a *strategy* for future game-play, consisting of several "backup plans" contingent upon future responses of their adversary.
To be as least-conservative as possible, an ideal (in a worst-case sense) decision-making process would more properly resemble

$$u = \kappa_{mpc}(x(t)) \triangleq u^*_t \tag{14a}$$

where $u_t^* \in \mathbb{R}^m$ is the constant value satisfying

$$u_t^* \triangleq \arg\min_{u_t} \left(\max_{d_{[t,\,t+T]} \in \mathcal{D}} \min_{u^p_{[t,\,t+T]} \in \mathcal{U}(u_t)} V_T(x(t), u^p_{[t,\,t+T]}, d_{[t,\,t+T]}) \right) \tag{14b}$$

with the definition $\mathcal{U}(u_t) \triangleq \{u^p_{[t,\,t+T]} \mid u^p(t) = u_t\}$. Clearly, the "least conservative" property follows from the fact that a separate response is optimized for every possible sequence the adversary could play. This is analogous to the philosophy in Scokaert & Mayne (1998), for system $x^+ = Ax + Bu + d$, in which polytopic $\mathcal{D}$ allows the max to be reduced to selecting the worst index from a finitely-indexed collection of responses; this equivalently replaces the innermost minimization with an augmented search in the outermost loop over *all* input responses in the collection.

While (14) is useful as a definition, a more useful (equivalent) representation involves minimizing over *feedback policies* $k : [t,\, t+T] \times \mathbb{X} \to \mathbb{U}$ rather than trajectories:

$$u = \kappa_{mpc}(x(t)) \triangleq k^*(t, x(t)) \tag{15a}$$

$$k^*(\cdot,\cdot) \triangleq \arg\min_{k(\cdot,\cdot)} \max_{d_{[t,\,t+T]} \in \mathcal{D}} \left(V_T(x(t), k(\cdot,\cdot), d_{[t,\,t+T]}) \right) \tag{15b}$$

$$V_T(x(t), k(\cdot,\cdot), d_{[t,\,t+T]}) \triangleq \int_t^{t+T} L(x^p, k(\tau,\, x^p(\tau)))\, d\tau + W(x^p(t+T)) \tag{15c}$$

$$\text{s.t. } \forall \tau \in [t, t+T]: \qquad \tfrac{d}{d\tau} x^p = f(x^p, k(\tau,\, x^p(\tau)), d), \;\; x^p(t) = x(t) \tag{15d}$$

$$(x^p(\tau), k(\tau,\, x^p(\tau))) \in \mathbb{X} \times \mathbb{U} \tag{15e}$$

$$x^p(t+T) \in \mathbb{X}_f \tag{15f}$$

There is a recursive-like elegance to (15), in that $\kappa_{mpc}(x)$ is essentially defined as a search over future candidates of itself. Whereas (14) explicitly involves *optimization-based* future feedbacks, the search in (15) can actually be (suboptimally) restricted to *any* arbitrary sub-class of feedbacks $k : [t,\, t+T] \times \mathbb{X} \to \mathbb{U}$. For example, this type of approach first appeared in Kothare et al. (1996); Lee & Yu (1997); Mayne (1995), where the cost functional was minimized by restricting the search to the class of linear feedback $u = Kx$ (or $u = K(t)x$).

The error cone $\{x^p_{[t,\,t+T]}\}^*_{\mathcal{D}}$ associated with (15) is typically *much* less conservative than that of (13). This is due to the fact that (15d) accounts for future disturbance attenuation resulting from $k(\tau, x^p(\tau))$, an effect ignored in the open-loop predictions of (13). In the case of (14) and (15) it is no longer necessary to include T as an optimization variable, since by condition C4a one can now envision extending the horizon by appending an increment $k(T+\delta t,\, \cdot) = k_f(\cdot)$.

This notion of feedback MPC has been applied in Magni et al. (2003; 2001) to solve $\mathcal{H}_\infty$ disturbance attenuation problems. This approach avoids the need to solve a difficult Hamilton-Jacobi-Isaacs equation, by combining a specially-selected stage cost $L(x,u)$ with a local HJI approximation $W(x)$ (designed generally by solving an $\mathcal{H}_\infty$ problem for the linearized system). An alternative perspective of the implementation of (15) is developed in Langson et al. (2004), with particular focus on obstacle-avoidance in Raković & Mayne (2005). In this work, a set-invariance philosophy is used to propagate the uncertainty cone $\{x^p_{[t,\,t+T]}\}_{\mathcal{D}}$ for (15d) in the form of a control-invariant tube. This enables the use of efficient methods for constructing control invariant sets based on approximations such as polytopes or ellipsoids.

11. Adaptive Approaches to MPC

The sectionr will be focused on the more typical role of adaptation as a means of coping with uncertainties in the system model. A standard implementation of model predictive control using a nominal model of the system dynamics can, with slight modification, exhibit nominal robustness to disturbances and modelling error. However in practical situations, the system model is only approximately known, so a guarantee of robustness which covers only "sufficiently small" errors may be unacceptable. In order to achieve a more solid robustness guarantee, it becomes necessary to account (either explicitly, or implicitly) for *all* possible trajectories which could be realized by the uncertain system, in order to guarantee feasible stability. The obvious numerical complexity of this task has resulted in an array of different control approaches, which lie at various locations on the spectrum between simple, conservative approximations versus complex, high-performance calculations. Ultimately, selecting an appropriate approach involves assessing, for the particular system in question, what is an acceptable balance between computational requirements and closed-loop performance.
Despite the fact that the ability to adjust to changing process conditions was one of the earliest industrial motivators for developing predictive control techniques, the progress in this area has been negligible. The small amount of progress that has been made is restricted to systems which do not involve constraints on the state, and which are affine in the unknown parameters. We will briefly describe two such results.

11.1 Certainty-equivalence Implementation

The result in Mayne & Michalska (1993) implements a certainty equivalence nominal-model[9] MPC feedback of the form $u(t) = \kappa_{mpc}(x(t), \hat{\theta}(t))$, to stabilize the uncertain system

$$\dot{x} = f(x,u,\theta) \triangleq f_0(x,u) + g(x,u)\theta \tag{16}$$

subject to an input constraint $u \in \mathbb{U}$. The vector $\theta \in \mathbb{R}^p$ represents a set of unknown constant parameters, with $\hat{\theta} \in \mathbb{R}^p$ denoting an identifier. Certainty equivalence implies that the nominal prediction model (11c) is of the same form as (16), but with $\hat{\theta}$ used in place of θ.
At any time $t \geq 0$, the identifier $\hat{\theta}(t)$ is defined to be a (min-norm) solution of

$$\int_0^t g(x(s),u(s))^T \Big(\dot{x}(s) - f_0(x(s),u(s))\Big) ds = \int_0^t g(x(s),u(s))^T g(x(s),u(s)) ds \; \hat{\theta} \tag{17}$$

which is solved over the window of all past history, under the assumption that $\dot{x}$ is measured (or computable). If necessary, an additional search is performed along the nullspace of $\int_0^t g(x,u)^T g(x,u) ds$ in order to guarantee $\hat{\theta}(t)$ yields a controllable certainty-equivalence model (since (17) is controllable by assumption).
The final result simply shows that there must exist a time $0 < t_a < \infty$ such that the regressor $\int_0^t g(x,u)^T g(x,u) ds$ achieves full rank, and thus $\hat{\theta}(t) \equiv \theta$ for all $t \geq t_a$. However, it is only by assumption that the state $x(t)$ does not escape the stabilizable region during the identification phase $t \in [0, t_a]$; moreover, there is no mechanism to decrease t_a in any way, such as by injecting excitation.

[9] Since this result arose early in the development of nonlinear MPC, it happens to be based upon a terminal-constrained controller (i.e., $\mathbb{X}_f \equiv \{0\}$); however, this is not critical to the adaptation.

11.1.1 Stability-Enforced Approach

One of the early stability results for nominal-model MPC in (Primbs (1999); Primbs et al. (2000)) involved the use of a global CLF $V(x)$ instead of a terminal penalty. Stability was enforced by constraining the optimization such that $V(x)$ is decreasing, and performance achieved by requiring the predicted cost to be less than that accumulated by simulation of pointwise min-norm control.
This idea was extended in Adetola & Guay (2004) to stabilize unconstrained systems of the form

$$\dot{x} = f(x,u,\theta) \triangleq f_0(x) + g_\theta(x)\theta + g_u(x)u \tag{18}$$

Using ideas from robust stabilization, it is assumed that a global ISS-CLF[10] is known for the nominal system. Constraining $V(x)$ to decrease ensures convergence to a neighbourhood of the origin, which gradually contracts as the identification proceeds. Of course, the restrictiveness of this approach lies in the assumption that $V(x)$ is known.

12. An Adaptive Approach to Robust MPC

Both the theoretical and practical merits of model-based predictive control strategies for nonlinear systems are well established, as reviewed in Chapter 7. To date, the vast majority of implementations involve an "accurate model" assumption, in which the control action is computed on the basis of predictions generated by an approximate nominal process model, and implemented (un-altered) on the actual process. In other words, the effects of plant-model mismatch are completely ignored in the control calculation, and closed-loop stability hinges upon the critical assumption that the nominal model is a "sufficiently close" approximation of the actual plant. Clearly, this approach is only acceptable for processes whose dynamics can be modelled a-priori to within a high degree of precision.
For systems whose true dynamics can only be approximated to within a large margin of uncertainty, it becomes necessary to directly account for the plant-model mismatch. To date, the most general and rigourous means for doing this involves explicitly accounting for the error in the online calculation, using the robust-MPC approaches discussed in Section 10.1. While the theoretical foundations and guarantees of stability for these tools are well established, it remains problematic in most cases to find an appropriate approach yielding a satisfactory balance between computational complexity, and conservatism of the error calculations. For example, the framework of min-max feedback-MPC Magni et al. (2003); Scokaert & Mayne (1998) provides the least-conservative control by accounting for the effects of future feedback actions, but is in most cases computationally intractable. In contrast, computationally simple approaches such as the openloop method of Marruedo et al. (2002) yield such conservatively-large error estimates, that a feasible solution to the optimal control problem often fails to exist.
For systems involving primarily *static* uncertainties, expressible in the form of unknown (constant) model parameters $\theta \in \Theta \subset \mathbb{R}^p$, it would be more desirable to approach the problem in the framework of adaptive control than that of robust control. Ideally, an adaptive mechanism enables the controller to improve its performance over time by employing a process model which asymptotically approaches that of the true system. Within the context of predictive control, however, the transient effects of parametric estimation error have proven problematic

[10] i.e., a CLF guaranteeing robust stabilization to a neighbourhood of the origin, where the size of the neighbourhood scales with the $\mathcal{L}_\infty$ bound of the disturbance signal

towards developing anything beyond the limited results discussed in Section 11. In short, the development of a general “robust adaptive-MPC" remains at present an open problem.
In the following, we make no attempt to construct such a “robust adaptive" controller; instead we propose an approach more properly referred to as “adaptive robust" control. The approach differs from typical adaptive control techniques, in that the adaptation mechanism does not directly involve a parameter identifier $\hat{\theta} \in \mathbb{R}^p$. Instead, a set-valued description of the parametric uncertainty, Θ, is adapted online by an identification mechanism. By gradually eliminating values from Θ that are identified as being inconsistent with the observed trajectories, Θ gradually contracts upon θ in a nested fashion. By virtue of this nested evolution of Θ, it is clear that an adaptive feedback structure of the form in Figure 2 would retain the stability properties of any underlying robust control design.

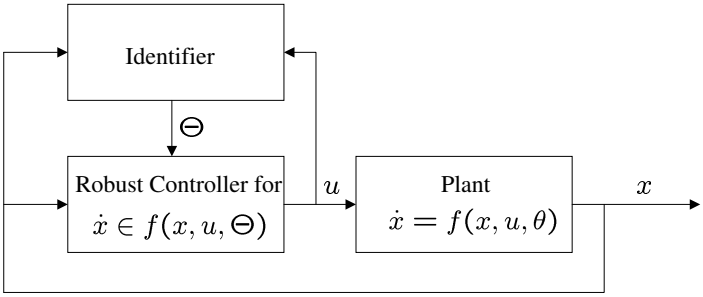


Fig. 2. Adaptive robust feedback structure

The idea of arranging an identifier and robust controller in the configuration of Figure 2 is itself not entirely new. For example the robust control design of Corless & Leitmann (1981), appropriate for nonlinear systems affine in u whose disturbances are bounded and satisfy the so-called “matching condition", has been used by various authors Brogliato & Neto (1995); Corless & Leitmann (1981); Tang (1996) in conjunction with different identifier designs for estimating the disturbance bound resulting from parametric uncertainty. A similar concept for linear systems is given in Kim & Han (2004).
However, to the best of our knowledge this idea has not been well explored in the situation where the underlying robust controller is designed by robust-MPC methods. The advantage of such an approach is that one could then potentially imbed an internal model of the identification mechanism into the predictive controller, as shown in Figure 3. In doing so the effects of future identification are accounted for within the optimal control problem, the benefits of which are discussed in the next section.

13. A Minimally-Conservative Perspective

13.1 Problem Description

The problem of interest is to achieve robust regulation, by means of state-feedback, of the system state to some compact target set $\Sigma_x^o \in \mathbb{R}^n$. Optimality of the resulting trajectories are measured with respect to the accumulation of some instantaneous penalty (i.e., stage cost) $L(x, u) \geq 0$, which may or may not have physical significance. Furthermore, the state and input trajectories are required to obey pointwise constraints $(x, u) \in \mathbb{X} \times \mathbb{U} \subseteq \mathbb{R}^n \times \mathbb{R}^m$.

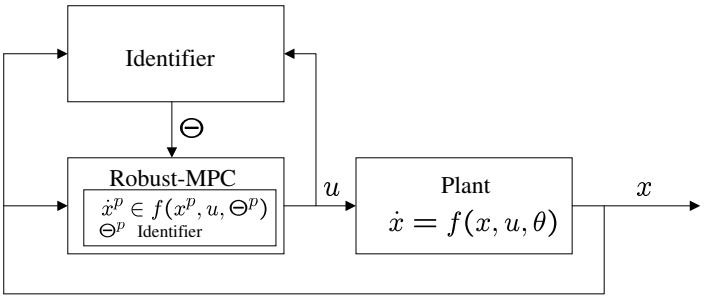


Fig. 3. Adaptive robust MPC structure

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 assumed to be of the general form

$$\dot{x} = f(x, u, \theta, d(t)) \tag{19}$$

where f is a locally Lipschitz vector 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 could be parameters associated with any (finite) set of universal basis functions used to approximate unknown nonlinearities. 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 the truncation error resulting from using a finite basis).

Assumption 13.1. *$\theta \in \Theta^o$, where Θ^o is a known compact subset of $\mathbb{R}^p$.*

Assumption 13.2. *$d(\cdot) \in \mathcal{D}_\infty$, where $\mathcal{D}_\infty$ is the set of all right-continuous $\mathcal{L}^\infty$-bounded functions $d : \mathbb{R} \to \mathcal{D}$; i.e., composed of continuous subarcs $d_{[a,b)}$, and satisfying $d(\tau) \in \mathcal{D}$, $\forall \tau \in \mathbb{R}$, with $\mathcal{D} \subset \mathbb{R}^d$ a compact vectorspace.* ■

Unlike much of the robust or adaptive MPC literature, we do 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:

Assumption 13.3. *Letting $\Sigma_u^o \subseteq \mathbb{U}$ be a chosen compact set, assume that $L : \mathbb{X} \times \mathbb{U} \to \mathbb{R}_{\geq 0}$ is continuous, $L(\Sigma_x^o, \Sigma_u^o) \equiv 0$, and $L(x, u) \geq \underline{\gamma}_L\big(\|(x, u)\|_{\Sigma_x^o \times \Sigma_u^o}\big)$, $\underline{\gamma}_L \in \mathcal{K}_\infty$. As well, assume that*

$$\min_{(u,\theta,d) \in \mathbb{U} \times \Theta^o \times \mathcal{D}} \left(\frac{L(x, u)}{\|f(x, u, \theta, d)\|} \right) \geq \frac{c_2}{\|x\|_{\Sigma_x^o}} \qquad \forall x \in \mathbb{X} \setminus B(\Sigma_x^o, c_1) \tag{20}$$

■

Definition 13.4. *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 $\dot{x} \in f(x, u, \Theta, \mathcal{D})$, using only controls $u \in \Sigma_u^o$.*

Assumption 13.5. *There exists a constant $N_\Sigma < \infty$, and a finite cover of Θ^o (not necessarily unique), denoted $\{\Theta\}^\Sigma$, such that*

i. the collection $\{\mathring{\Theta}\}^{\Sigma}$ *is an open cover for the interior* $\mathring{\Theta}^o$.

ii. $\Theta \in \{\Theta\}^{\Sigma}$ *implies* $\Sigma_x(\Theta) \neq \emptyset$.

iii. $\{\Theta\}^{\Sigma}$ *contains at most* N_{Σ} *elements.* ■

The most important requirement of Assumption 13.3 is that, since the exact location (in $\mathbb{R}^n \times \mathbb{R}^m$) of the equilibrium[11] manifold is not known *a-priori*, $L(x,u)$ must be identically zero on the entire region of equilibrium candidates $\Sigma_x^o \times \Sigma_u^o$. One example of how to construct such a function would be to define $L(x,u) = \rho(x,u)\overline{L}(x,u)$, where $\overline{L}(x,u)$ is an arbitrary penalty satisfying $(x,u) \notin \Sigma_x^o \times \Sigma_u^o \implies \overline{L}(x,u) > 0$, and $\rho(x,u)$ is a smoothed indicator function of the form

$$\rho(x,u) = \begin{cases} 0 & (x,u) \in \Sigma_x^o \times \Sigma_u^o \\ \frac{\|(x,u)\|_{\Sigma_x^o \times \Sigma_u^o}}{\delta_\rho} & 0 < \|(x,u)\|_{\Sigma_x^o \times \Sigma_u^o} < \delta_\rho \\ 1 & \|(x,u)\|_{\Sigma_x^o \times \Sigma_u^o} \geq \delta_\rho \end{cases} \tag{21}$$

The restriction that $L(x,u)$ is strictly positive definite with respect to $\Sigma_x^o \times \Sigma_u^o$ is made for convenience, and could be relaxed to positive semi-definite using an approach similar to that of Grimm et al. (2005) as long as $L(x,u)$ satisfies an appropriate detectability assumption (i.e., as long as it is guaranteed that all trajectories remaining in $\{x \mid \exists u \ s.t. \ L(x,u) = 0\}$ must asymptotically approach $\Sigma_x^o \times \Sigma_u^o$).

The first implication of Assumption 13.5 is that for any $\theta \in \Theta^o$, the target Σ_x^o contains a stabilizable "equilibrium" $\Sigma(\theta)$ such that the regulation problem is well-posed. Secondly, the openness of the covering in Assumption 13.5 implies a type of "local-ISS" property of these equilibria with respect to perturbations in small neighbourhoods Θ of θ. This property ensures that the target is stabilizable given "sufficiently close" identification of the unknown θ, such that the adaptive controller design is tractable.

13.2 Adaptive Robust Controller Design Framework

13.2.1 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 parametric uncertainty set Θ^o is adapted to gradually eliminate sets which do not contain θ. To this end, we define the infimal uncertainty set

$$\mathbb{Z}(\Theta, x_{[a,b]}, u_{[a,b]}) \triangleq \{\, \theta \in \Theta \mid \dot{x}(\tau) \in f(x(\tau), u(\tau), \theta, \mathcal{D}), \ \forall \tau \in [a,b] \,\} \tag{22}$$

By definition, $\mathbb{Z}$ represents the best-case performance that could be achieved by any identifier, given a set of data generated by (19), and a prior uncertainty bound Θ. Since exact online calculation of (22) is generally impractical, we assume that the set $\mathbb{Z}$ is approximated online using an arbitrary estimator Ψ. This estimator must be chosen to satisfy the following conditions.

Criterion 13.6. $\Psi(\cdot,\cdot,\cdot)$ *is designed such that for* $a \leq b \leq c$, *and for any* $\Theta \subseteq \Theta^o$,

C13.6.1 $\mathbb{Z} \subseteq \Psi$

C13.6.2 $\Psi(\Theta,\cdot,\cdot) \subseteq \Theta$, *and closed.*

[11] we use the word "equilibrium" loosely in the sense of control-invariant subsets of the target Σ_x^o, which need not be actual equilibrium *points* in the traditional sense

C13.6.3 $\Psi(\Theta_1, x_{[a,b]}, u_{[a,b]}) \subseteq \Psi(\Theta_2, x_{[a,b]}, u_{[a,b]})$, *for* $\Theta_1 \subseteq \Theta_2 \subseteq \Theta^o$

C13.6.4 $\Psi(\Theta, x_{[a,b]}, u_{[a,b]}) \supseteq \Psi(\Theta, x_{[a,c]}, u_{[a,c]})$

C13.6.5 $\Psi(\Theta, x_{[a,c]}, u_{[a,c]}) \equiv \Psi(\Psi(\Theta, x_{[a,b]}, u_{[a,b]}), x_{[b,c]}, u_{[b,c]})$

The set Ψ represents an approximation of $\mathbb{Z}$ in two ways. First, both Θ^o and Ψ can be restricted *a-priori* to any class of finitely-parameterized sets, such as linear polytopes, quadratic balls, etc. Second, contrary to the actual definition of (22), Ψ can be computed by *removing* values from Θ^o as they are determined to violate the differential inclusion model. As such, the search for infeasible values can be terminated at any time without violating C13.6.
The closed loop dynamics of (19) then take the form

$$\dot{x} = f(x, \kappa_{mpc}(x, \Theta(t)), \theta, d(t)), \quad x(t_0) = x_0 \tag{23a}$$

$$\Theta(t) = \Psi(\Theta^o, x_{[t_0,\,t]}, u_{[t_0,\,t]}) \tag{23b}$$

where $\kappa_{mpc}(x, \Theta)$ represents the MPC feedback policy, detailed in Section 13.2.2. In practice, the (set-valued) controller state Θ could be generated using an update law $\dot{\Theta}$ designed to gradually contract the set (satisfying C13.6). However, the given statement of (23b) is more general, as it allows for $\Theta(t)$ to evolve discontinuously in time, as may happen for example when the sign of a parameter can suddenly be conclusively determined.

13.2.2 Feedback-MPC framework

In the context of min-max robust MPC, it is well known that feedback-MPC, because of its ability to account for the effects of future feedback decisions on disturbance attenuation, provides significantly less conservative performance than standard open-loop MPC implementations. In the following, the same principle is extended to incorporate the effects of future parameter adaptation.
In typical feedback-MPC fashion, the receding horizon control law in (23) is defined by minimizing over *feedback policies* $\kappa : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \times \overline{\text{cov}}\,\{\Theta^o\} \to \mathbb{R}^m$ as

$$u = \kappa_{mpc}(x, \Theta) \triangleq \kappa^*(0, x, \Theta) \tag{24a}$$

$$\kappa^* \triangleq \arg\min_{\kappa(\cdot,\cdot,\cdot)} J(x, \Theta, \kappa) \tag{24b}$$

where $J(x, \Theta, \kappa)$ is the (worst-case) cost associated with the optimal control problem:

$$J(x, \Theta, \kappa) \triangleq \max_{\substack{\theta \in \Theta \\ d(\cdot) \in \mathcal{D}_\infty}} \int_0^T L(x^p, u^p) d\tau + W(x_f^p, \hat{\Theta}_f) \tag{25a}$$

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

$$\tfrac{d}{d\tau} x^p = f(x^p, u^p, \theta, d), \;\; x^p(0) = x \tag{25b}$$

$$\hat{\Theta}(\tau) = \Psi_p(\Theta(t), x^p_{[0,\tau]}, u^p_{[0,\tau]}) \tag{25c}$$

$$x^p(\tau) \in \mathbb{X} \tag{25d}$$

$$u^p(\tau) \triangleq \kappa(\tau, x^p(\tau), \hat{\Theta}(\tau)) \in \mathbb{U} \tag{25e}$$

$$x_f^p \triangleq x^p(T) \in \mathbb{X}_f(\hat{\Theta}_f) \tag{25f}$$

$$\hat{\Theta}_f \triangleq \Psi_f(\Theta(t), x^p_{[0,T]}, u^p_{[0,T]}) \tag{25g}$$

Throughout the remainder, we denote the optimal cost $J^*(x,\Theta) \triangleq J(x,\Theta,\kappa^*)$, and furthermore we drop the explicit constraints (25d)-(25f) by assuming the definitions of L and W have been extended as follows:

$$L(x,u) = \begin{cases} L(x,u) < \infty & (x,u) \in \mathbb{X} \times \mathbb{U} \\ +\infty & \text{otherwise} \end{cases} \tag{26a}$$

$$W(x,\Theta) = \begin{cases} W(x,\Theta) < \infty & x \in \mathbb{X}_f(\Theta) \\ +\infty & \text{otherwise} \end{cases} \tag{26b}$$

The parameter identifiers Ψ_p and Ψ_f in (25) represent internal model approximations of the actual identifier Ψ, and must satisfy both C13.6 as well as the following criterion:

Criterion 13.7. *For identical arguments, $\mathbb{Z} \subseteq \Psi \subseteq \Psi_f \subseteq \Psi_p$.*

Remark 13.8. *We distinguish between different identifiers to emphasize that, depending on the frequency at which calculations are called, differing levels of accuracy can be applied to the identification calculations. The ordering in Criterion 13.7 is required for stability, and implies that identifiers existing within faster timescales provide more conservative approximations of the uncertainty set.*

There are two important characteristics which distinguish (25) from a standard (non-adaptive) feedback-MPC approach. First, the future evolution of $\hat{\Theta}$ in (25c) is fed back into both (25b) and (25e). The benefits of this feedback are analogous to those of adding state-feedback into the MPC calculation; the resulting cone of possible trajectories $x^p(\cdot)$ is narrowed by accounting for the effects of future adaptation on disturbance attenuation, resulting in less conservative worst-case predictions.
The second distinction is that both W and $\mathbb{X}_f$ are parameterized as functions of $\hat{\Theta}_f$, which reduces the conservatism of the terminal cost. Since the terminal penalty W has the interpretation of the "worst-case cost-to-go", it stands to reason that W should decrease with decreased parametric uncertainty. In addition, the domain $\mathbb{X}_f$ would be expected to enlarge with decreased parametric uncertainty, which in some situations could mean that a stabilizing CLF-pair $(W(x,\Theta), \mathbb{X}_f(\Theta))$ can be constructed even when no such CLF exists for the original uncertainty Θ^o. This effect is discussed in greater depth in Section 14.1.1.

13.2.3 Generalized Terminal Conditions

To guide the selection of $W(x_f, \hat{\Theta}_f)$ and $\mathbb{X}_f(\hat{\Theta}_f)$ in (25), it is important to outline (sufficient) conditions under which (23)-(25) can guarantee stabilization to the target Σ_x^o. The statement given here is extended from the set of such conditions for robust MPC from Mayne et al. (2000) that was outlined in Sections 8 and 10.1.1.
For reasons that are explained later in Section 14.1.1, it is useful to present these conditions in a more general context in which $W(\cdot,\Theta)$ is allowed to be $\mathcal{LS}$-continuous with respect to x, as may occur if W is generated by a switching mechanism. This adds little additional complexity to the analysis, since (25) is already discontinuous due to constraints.

Criterion 13.9. *The set-valued terminal constraint function $\mathbb{X}_f : \overline{co}\{\Theta^o\} \to \overline{co}\{\mathbb{X}\}$ and terminal penalty function $W : \mathbb{R}^n \times \overline{co}\{\Theta^o\} \to [0, +\infty]$ are such that for each $\Theta \in \overline{co}\{\Theta^o\}$, there exists $k_f(\cdot,\Theta) : \mathbb{X}_f \to \mathbb{U}$ satisfying*

***C13.9**.1 $\mathbb{X}_f(\Theta) \neq \emptyset$ implies that $\Sigma_x^o \cap \mathbb{X}_f(\Theta) \neq \emptyset$, and $\mathbb{X}_f(\Theta) \subseteq \mathbb{X}$ is closed*

***C13.9**.2 $W(\cdot,\Theta)$ is $\mathcal{LS}$-continuous with respect to $x \in \mathbb{R}^n$*

***C13.9**.3* $k_f(x,\Theta) \in \mathbb{U}$, *for all* $x \in \mathbb{X}_f(\Theta)$.

***C13.9**.4* $\mathbb{X}_f(\Theta)$ *and* $\Sigma_x(\Theta) \subseteq \left\{\Sigma_x^o \cap \mathbb{X}_f(\Theta)\right\}$ *are both strongly positively invariant with respect to the differential inclusion* $\dot{x} \in f(x, k_f(x,\Theta), \Theta, \mathcal{D})$.

***C13.9**.5* $\forall x \in \mathbb{X}_f(\Theta)$, *and denoting* $\mathcal{F} \triangleq f(x, k_f(x,\Theta), \Theta, \mathcal{D})$,

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

■

Although condition C13.9.5 is expressed in a slightly non-standard form, it embodies the standard interpretation that W must be decreasing by at least an amount $-L(x,k_f(x,\Theta))$ along all vectorfields in the closed-loop differential inclusion $\mathcal{F}$; i.e., $W(x,\Theta)$ is a robust-CLF (in an appropriate non-smooth sense) on the domain $\mathbb{X}_f(\Theta)$. Lyapunov stability involving $\mathcal{LS}$-continuous functions is thoroughly studied in Clarke et al. (1998), and provides a meaningful sense in which W can be considered a "robust-CLF" despite its discontinuous nature.
It is important to note that for the purposes of Criterion 13.9, $W(x,\Theta)$ and $\mathbb{X}_f(\Theta)$ are parameterized by the set Θ, but the criterion imposes no restrictions on their functional dependence with respect to the Θ argument. This Θ-dependence is required to satisfy the following criteria:

Criterion 13.10. *For any* $\Theta_1, \Theta_2 \in \overline{cov}\{\Theta^o\}$ *such that* $\Theta_1 \subseteq \Theta_2$,

***C13.10**.1* $\mathbb{X}_f(\Theta_2) \subseteq \mathbb{X}_f(\Theta_1)$

***C13.10**.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 13.9 and 13.10 may appear prohibitively complex; however, the task is greatly simplified by noting that neither criterion imposes 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 14.1.1.

13.2.4 Closed-loop Stability

Theorem 13.11 (Main result)**.** *Given system* (19), *target* Σ_x^o, *and penalty* L *satisfying Assumptions 13.1, 13.2, 13.3, 13.5, assume the functions* Ψ, Ψ_p, Ψ_f, W *and* $\mathbb{X}_f$ *are designed to satisfy Criteria 13.6, 13.7, 13.9, and 13.10. Furthermore, let* $\mathbb{X}_0 \triangleq \mathbb{X}_0(\Theta^o) \subseteq \mathbb{X}$ *denote the set of initial states, with uncertainty* $\Theta(t_0) = \Theta^o$, *for which* (25) *has a solution. Then under* (23), Σ_x^o *is feasibly asymptotically stabilized from any* $x_0 \in \mathbb{X}_0$. ■

Remark 13.12. *As indicated by Assumption 13.5, the existence of an invariant target set* $\Sigma_x^o(\Theta^o)$, *robust to the full parametric uncertainty* Θ^o, *is not required for Theorem 13.11 to hold. The identifier* $\hat{\Theta}_f$ *must be contained in a sufficiently small neighbourhood of (the worst-case)* θ *such that nontrivial* $\mathbb{X}_f(\hat{\Theta}_f)$ *and* $W(\cdot,\hat{\Theta}_f)$ *exist, for* (25) *to be solvable. While this imposes a minimum performance requirement on* Ψ_f, *it enlarges the domain* $\mathbb{X}_0$ *for which the problem is solvable.*

14. Computation and Performance Issues

14.1 Excitation of the closed-loop trajectories

Contrary to much of the adaptive control literature, including adaptive-MPC approaches such as Mayne & Michalska (1993), the result of Theorem 13.11 does not depend on any auxiliary excitation signal, nor does it require any assumptions regarding the persistency or quality of excitation in the closed-loop behaviour.

Instead, any benefits to the identification which result from injecting excitation into the input signal are predicted by (25c) and (25g), and thereby are automatically accounted for in the posed optimization. In the particular case where $\Psi_p \equiv \Psi_f \equiv \Psi$, then the controller generated by (25) will automatically inject the exact type and amount of excitation which optimizes the cost $J^*(x, \Theta)$; i.e., the closed-loop behaviour (23) 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 13.11 remains valid. However, the non-smoothness of $J^*(x, \Theta)$ (with respect to both x and Θ) makes it difficult to quantify the impact of these deviations on the closed-loop behaviour. Qualitatively, small changes to Ψ_p or Ψ_f yielding increasingly conservative identification would generally result in the optimal control solution injecting additional excitation to compensate for the de-sensitized identifier. However, if the changes to Ψ_p or Ψ_f are sufficiently large such that the injection of additional excitation is insufficient to prevent a discontinuous increase in J^*, then it is possible that the optimal solution may suddenly involve *less* excitation than previously, to instead reduce actuation energy. Clearly this behaviour is the result of nonconvexities in the optimal control problem (24), which is inherently a nonconvex problem even in the absence of the adaptive mechanisms proposed here.

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

Proposition 14.1. *Let $\{(W^i, \mathbb{X}_f^i)\}$ denote a finitely-indexed collection of terminal function candidates, with indices $i \in \mathcal{I}$, where each pair $(W^i, \mathbb{X}_f^i)$ satisfies Criteria 13.9 and 13.10. Then*

$$W(x, \Theta) \triangleq \min_{i \in \mathcal{I}} \{W^i(x, \Theta)\}, \qquad \mathbb{X}_f(\Theta) \triangleq \bigcup_{i \in \mathcal{I}} \{\mathbb{X}_f^i(\Theta)\} \tag{27}$$

satisfy Criteria 13.9 and 13.10. ■

Using Proposition 14.1, it is clear that one approach to constructing $W(\cdot, \cdot)$ and $\mathbb{X}_f(\cdot)$ is to use a collection of pairs of the form

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

This collection can be obtained 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 13.9 and 13.10)

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" constitute a cover[12] of 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 as some other controlled-invariant set.
   - 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 not an easy task, this is a very standard robust-control calculation. As such, there is a wealth of tools in the robust control and viability literatures (see, for example Aubin (1991)) to tackle this problem.

3. **Calculate $W(\cdot, \Theta)$ and $\mathbb{X}_f(\Theta)$ online:**

   i. Given Θ, identify all sets that are active: $\mathcal{I}^* = \mathcal{I}^*(\Theta) \triangleq \left\{ j \mid \Theta \subseteq \Theta^j \right\}$. Using the hierarchy, test only immediate children of active parents.

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

Remark 14.2. *Although the above steps assume that Θ^j is selected before $\mathbb{X}_f^j$, an alternative approach would be to design the candidates $W^j(\cdot)$ on the basis of a collection of* parameter values *$\hat{\theta}^j$. Briefly, this could be constructed as follows:*

1. *Generate a grid of values $\{\theta^i\}$ distributed across Θ^o.*
2. *Design $W^j(\cdot)$ based on a certainty-equivalence model for $\hat{\theta}^j$ (for example, by linearization). Specify $\mathbb{X}_f^j$ (likely as a level set of W^j), and then approximate the maximal neighbourhood Θ^j of $\hat{\theta}^j$ such that Criterion 13.9 holds.*
3. *For the same (θ^j, W^j) pair, multiple $(W^j, \mathbb{X}_f^j)$ candidates can be defined corresponding to different Θ^j.*

14.2 Robustness Issues

One could argue that if the disturbance model $\mathcal{D}$ in (19) encompasses *all* possible sources of model uncertainty, then the issue of robustness is completely addressed by the min-max formulation of (25). In practice this is not realistic, since it is generally desirable to explicitly consider significant disturbances only, or to exclude $\mathcal{D}$ entirely if Θ represents the dominant uncertainty. The lack of nominal robustness to model error in constrained nonlinear MPC is a well documented problem, as discussed in Grimm et al. (2004). In particular, Grimm et al.

[12] specifically, the interiors of all peers must together constitute an open cover

(2003); Marruedo et al. (2002) establish nominal robustness (for "accurate-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, that is relevant to the current adaptive framework.

In addition to ensuring robustness of the controller itself, using methods similar to those mentioned above, it is equally important to ensure that the adaptive mechanism Ψ, including its internal models Ψ_f and Ψ_p, exhibits at least some level of nominal robustness to unmodelled disturbances. By Criterion 13.6.4, the online estimation *must* evolve in a nested fashion and therefore the true θ must never be inadvertently excluded from the estimated uncertainty set. Therefore, just as $\mathcal{Z}$ in (22) defined a best-case bound around which the identifiers in the previous sections could be designed, we present here a modification of (22) which quantifies the type of conservatism required in the identification calculations for the identifiers to possess nominal robustness.

For any $\gamma, \delta \geq 0$, and with $\tau_a \triangleq \tau - a$, we define the following modification of (22):

$$\mathcal{Z}^{\delta,\gamma}(\Theta, x_{[a,b]}, u_{[a,b]}) \triangleq \{\, \theta \in \Theta \mid B(\dot{x}, \delta + \gamma\tau_a) \cap f(B(x, \gamma\tau_a), u, \theta, \mathcal{D}) \neq \emptyset,\ \forall \tau \,\}. \tag{28}$$

Equation (28) provides a conservative outer-approximation of (22) such that $\mathcal{Z} \subseteq \mathcal{Z}^{\delta,\gamma}$. The definition in (28) accounts for two different types of conservatism that can be introduced into the identification calculations. First, the parameter $\delta > 0$ represents a minimum tolerance for the distance between actual derivative information from trajectory $x_{[a,b]}$ and the model (19) when determining if a parameter value can be excluded from the uncertainty set. For situations where the trajectory $x_{[a,b]}$ is itself a *prediction*, as is the case for the internal models Ψ_f and Ψ_p, the parameter $\gamma > 0$ represents increasingly relaxed tolerances applied along the length of the trajectory. Throughout the following we denote $\mathcal{Z}^{\delta} \equiv \mathcal{Z}^{\delta,0}$, with analogous notations for Ψ, Ψ_f, and Ψ_p.

The following technical property of definition (28) is useful towards establishing the desired robustness claim:

Claim 14.3. *For any $a < b < c$, $\gamma \geq 0$, and $\delta \geq \delta' \geq 0$, let $x'_{[a,c]}$ be an arbitrary, continuous perturbation of $x_{[a,b]}$ satisfying*

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

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

Then, $\mathcal{Z}^{\delta,\gamma}$ satisfies

$$\mathcal{Z}^{\delta,\gamma}\left(\mathcal{Z}^{\delta'}(\Theta, x'_{[a,b]}, u_{[a,b]}), x'_{[b,c]}, u_{[b,c]}\right) \subseteq \mathcal{Z}^{\delta,\gamma}(\Theta, x_{[a,c]}, u_{[a,c]}). \tag{29}$$

■

Based on (28), we are now able to detail sufficient conditions under which the stability claim of Theorem 13.11 holds in the presence of small, unmodelled disturbances. For convenience, the following proposition is restricted to the situation where the only discontinuities in $W(x, \Theta)$ and $\mathbb{X}_f(\Theta)$ are those generated by a switching mechanism (as per Prop. 14.1) between a set of

candidates $\{W^i(x,\Theta), \mathbb{X}_f^i(\Theta)\}$ that are individually *continuous* on $x \in \mathbb{X}_f^i(\Theta)$ (i.e., a strengthening of C13.9.2). With additional complexity, the proposition can be extended to general $\mathcal{LS}$-continuous penalties $W(x,\Theta)$.

Proposition 14.4. *Assume that the following modifications are made to the design in Section 13.2:*

i. $W(x,\Theta)$ and $\mathbb{X}_f(\Theta)$ are constructed as per Prop. 14.1, but with C13.9.2 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 design parameter $\delta_x > 0$, (26) and (27) are redefined as:

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

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

iii. The individual sets $\mathbb{X}_f^i$ are specified such that there exists $\delta_f > 0$, for which C13.9.4 holds for every inner approximation $\overleftarrow{B}(\mathbb{X}_f^i(\Theta), \delta_x')$, $\delta_x' \in [0, \delta_x]$, where positive invariance is with respect to all flows generated by the differential inclusion $\dot{x} \in B(f(x, k_f^i(x,\Theta), \Theta, \mathcal{D}), \delta_f)$

iv. Using design parameters $\delta > \delta' > 0$ and $\gamma > 0$, the identifiers are modified as follows:

- *Ψ in (23b) is replaced by $\Psi^{\delta'} \equiv \Psi^{\delta',0}$*
- *Ψ_p and Ψ_f in (25) are replaced by $\Psi_p^{\delta,\gamma}$ and $\Psi_f^{\delta,\gamma}$, respectively*

where the new identifiers are assumed to satisfy C13.6, C13.7, and a relation of the form (29).

Then for any compact subset $\bar{\mathbb{X}}_0 \subseteq \mathbb{X}_0(\Theta^o)$, there exists $c^* = c^*(\gamma, \delta_x, \delta_f, \delta, \delta', \bar{\mathbb{X}}_0) > 0$ such that, for all $x_0 \in \bar{\mathbb{X}}_0$ and for all disturbances $\|d_2\| \leq c \leq c^*$, the target Σ_x^o and the actual dynamics

$$\dot{x} = f(x, \kappa_{mpc}(x, \Theta(t)), \theta, d(t)) + d_2(t), \quad x(t_0) = x_0 \tag{30a}$$

$$\Theta(t) = \Psi^{\delta'}(\Theta^o, x_{[t_0,t]}, u_{[t_0,t]}) \tag{30b}$$

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

14.3 Example Problem

To demonstrate the versatility of our approach, we consider the following nonlinear system:

$$\dot{x}_1 = -x_1 + \left|2\sin(x_1 + \pi\theta_1) + 1.5\theta_2 - x_1 + x_2\right| x_1 + d_1(t)$$
$$\dot{x}_2 = 10\,\theta_{4a}\theta_{4b} x_1\,(u + \theta_3) + d_2(t)$$

The uncertainty $\mathcal{D}$ is given by $|d_1|, |d_2| \leq 0.1$, and Θ^o by $\theta_1, \theta_2, \theta_3 \in [-1,1]$, and $\theta_{4a} \in \{-1,+1\}$, $\theta_{4b} \in [0.5,1]$. The control objective is to achieve regulation of x_1 to the set $x_1 \in [-0.2, 0.2]$, subject to the constraints $\mathbb{X} \triangleq \{|x_1| \leq M_1 \text{ and } |x_2| \leq M_2\}$, $\mathbb{U} \triangleq \{|u| \leq M_u\}$, with M_1, $M_2 \in (0, +\infty]$ and $M_u \in (1, +\infty]$ any given constants. The dynamics exhibit several challenging properties: i) state constraints, ii) nonlinear parameterization of θ_1 and θ_2, iii) potential open-loop instability with finite escape, iv) uncontrollable linearization, v) unknown

sign of control gain, and vi) exogenous disturbances. This system is not stabilizable by any non-adaptive approach (MPC or otherwise), and furthermore fits very few, if any, existing frameworks for adaptive control.

One key property of the dynamics (which is arguably necessary for the regulation objective to be well-posed) is that for any *known* $\theta \in \Theta$ the target is stabilizable and nominally robust. This follows by observing that the surface

$$s \triangleq 2\sin(x_1 + \pi\theta_1) + 1.5\theta_2 - x_1 + x_2 = 0$$

defines a sliding mode for the system, with a robustness margin $|s| \leq 0.5$ for $|x_1| \geq 0.2$. This motivates the design choices:

$$\begin{aligned}\mathbb{X}_f(\Theta) &\triangleq \{x \in \mathbb{X} \,|\, -M_2 \leq \underline{\Gamma}(x_1, \Theta) \leq x_2 \leq \overline{\Gamma}(x_1, \Theta) \leq M_2\} \\ \underline{\Gamma} &\triangleq x_1 - 1.5\underline{\theta}_2 - 2\sin(x_1 + \pi\theta_1^{avg}) - 2\pi(\underline{\theta}_1 - \theta_1^{avg}) + 0.5 \\ \overline{\Gamma} &\triangleq x_1 - 1.5\overline{\theta}_2 - 2\sin(x_1 + \pi\theta_1^{avg}) - 2\pi(\overline{\theta}_1 - \theta_1^{avg}) - 0.5\end{aligned}$$

where $\overline{\theta}_i$, $\underline{\theta}_i$ denote upper and lower bounds corresponding to $\Theta \subseteq \Theta^o$, and $\theta^{avg} \triangleq \frac{\overline{\theta}+\underline{\theta}}{2}$. The set $\mathbb{X}_f(\Theta)$ satisfies C13.10 and is nonempty for any Θ such that $\overline{\theta}_2 - \underline{\theta}_2 + \pi(\overline{\theta}_1 - \underline{\theta}_1) \leq 0.5$, that defines minimum thresholds for the performance of Ψ_f and the amount of excitation in solutions to (25).

It can be shown that $|s| \leq 0.5 \overset{\forall\theta\in\Theta^o}{\Longrightarrow} |x_1 - x_2| \leq 4$, and that $\mathbb{X}_f(\Theta)$ is control-invariant using $u \in [-1, 1]$, as long as the sign θ_{4a} is known. This motivates the definitions $\Sigma_u^o \triangleq [-1, 1]$, $\Sigma_1 = [-0.2, 0.2]$, $\Sigma_{12} = [-4, 4]$, and $\Sigma_x^o \triangleq \{x \,|\, (x_1, x_1 - x_2) \in \Sigma_1 \times \Sigma_{12}\}$, plus the modification of $\mathbb{X}_f(\Theta)$ above to contain the explicit requirement $\Theta_{4a} = \{-1, +1\} \implies \mathbb{X}_f(\Theta) = \emptyset$. Then on $x \in \mathbb{X}_f(\Theta)$, the cost functions $W(x, \Theta) \triangleq \frac{1}{2}\|x_1\|^2_{\Sigma_1}$ and $L(x, u) \triangleq \frac{1}{2}\left(\|x_1\|^2_{\Sigma_1} + \|x_1 - x_2\|^2_{\Sigma_{12}} + \|u\|^2_{\Sigma_u^o}\right)$ satisfy all the claims of C13.9, since $W \equiv L \equiv 0$ on $x \in \mathbb{X}_f \cap \Sigma_x^o$, and on $x \in \mathbb{X}_f \setminus \Sigma_x^o$ one has:

$$\dot{W} \leq \|x_1\|_{\Sigma_1}\left(-\tfrac{1}{2}|x_1| + 0.1\right) \leq -\tfrac{1}{2}\|x_1\|^2_{\Sigma_1} \leq -L(x, u).$$

15. Conclusions

In this chapter we have demonstrated the methodology for adaptive MPC, in which the adverse effects of parameter identification error are explicitly minimized using a robust MPC approach. As a result, it is possible to 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 the optimization problem, raising the potential to significantly reduce the conservativeness of the solutions, especially with respect to design of the terminal penalty. While the results presented here are conceptual, in that they are generally intractable to compute due to the underlying min-max feedback-MPC framework, this chapter provides insight into the maximum performance that could be attained by incorporating adaptation into a robust-MPC framework.

16. Proofs for Section 13

16.1 Proof of Theorem 13.11

This proof will follow the so-called "direct method" of establishing stability by directly proving strict decrease of $J^*(x(t),\Theta(t))$, for all $x \notin \Sigma_x^o$. Stability analysis involving $\mathcal{LS}$-continuous Lyapunov functions (for example, (Clarke et al., 1998, Thm4.5.5)) typically involves the proximal subgradient $\partial_p J^*$ (a generalization of ∇J), which is a somewhat ambiguous quantity in the context here given (23b). Instead, this proof exploits an alternative framework involving subderivates (generalized Dini-derivatives), which is equivalent by (Clarke et al., 1998, Prop4.5.3). Together, the following two conditions can be shown sufficient to ensure decrease of J^*, where $\mathcal{F} \triangleq f(x,\kappa_{mpc}(x,\Theta(t)),\Theta(t),\mathcal{D})$

$$\text{(i.)}\ \max_{f\in\mathcal{F}} \overrightarrow{D}J^*(x,\Theta) \triangleq \max_{f\in\mathcal{F}} \liminf_{\substack{v\to f\\ \delta\downarrow 0}} \tfrac{J^*(x+\delta v,\Theta(t+\delta))-J^*(x,\Theta(t))}{\delta} < 0$$

$$\text{(ii.)}\ \min_{f\in\mathcal{F}} \overleftarrow{D}J^*(x,\Theta) \triangleq \min_{f\in\mathcal{F}} \limsup_{\substack{v\to f\\ \delta\downarrow 0}} \tfrac{J^*(x-\delta v,\Theta(t-\delta))-J^*(x,\Theta(t))}{\delta} > 0$$

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, $\hat{\Theta}^p$ correspond to any worst-case minimizing solution of $J^*(x(t),\Theta(t))$, defined on $\tau \in [0,T]$. Additional notations which will be used: $T_\delta \triangleq T+\delta$, $\hat{\Theta}^p_T \triangleq \hat{\Theta}_f(T)$, $\hat{\Theta}^p_{T_\delta} \triangleq \hat{\Theta}_f(T_\delta)$; i.e., both sets represent solutions of the terminal identifier Ψ_f, evaluated along $x^p_{[0,T]}$ and $x^p_{[0,T_\delta]}$, respectively. Likewise, for an arbitrary argument $\mathbb{S} \in \{\hat{\Theta}^p_T,\ \hat{\Theta}^p_{T_\delta}\}$, we define $W^p_T(\mathbb{S}) \triangleq W(x^p(T),\mathbb{S})$ and $W^p_{T_\delta}(\mathbb{S}) \triangleq W(x^p(T_\delta),\mathbb{S})$.

With the above notations, it can be seen that if the minimizing solution $x^p_{[0,T]}$ were extended to $\tau \in [0,T_\delta]$ by implementing the feedback $u^p(\tau) = k_f(x^p(\tau),\hat{\theta}^p_T)$ on $\tau \in [T,T_\delta]$ (i.e., with $\hat{\theta}^p_T$ fixed), then Criterion C13.9.5 guarantees the inequality

$$\lim_{\delta\downarrow 0} \tfrac{1}{\delta}\left(\delta L(x^p_T,k_f(x^p_T,\hat{\Theta}^p_T)) + W^p_{T_\delta}(\hat{\Theta}^p_T) - W^p_T(\hat{\Theta}^p_T)\right) \le 0.$$

Using this fact, the relationship (i.) follows from:

$$\begin{aligned}
\max_{f\in\mathcal{F}} \overrightarrow{D}J^*(x,\Theta) &= \max_{f\in\mathcal{F}} \liminf_{\substack{v\to f\\ \delta\downarrow 0}} \tfrac{1}{\delta}\Big[J^*(x+\delta v,\Theta(t+\delta)) - \textstyle\int_0^T L^p d\tau - W^p_T(\hat{\Theta}^p_T)\Big] \\
&\le \max_{f\in\mathcal{F}} \liminf_{\substack{v\to f\\ \delta\downarrow 0}} \tfrac{1}{\delta}\Big[J^*(x+\delta v,\Theta(t+\delta)) - \textstyle\int_0^\delta L^p d\tau - \int_\delta^T L^p d\tau - W^p_T(\hat{\Theta}^p_T) \\
&\qquad\qquad - \left(\delta L(x^p_T,k_f(x^p_T,\hat{\Theta}^p_T)) + W^p_{T_\delta}(\hat{\Theta}^p_T) - W^p_T(\hat{\Theta}^p_T)\right)\Big] \\
&\le \max_{f\in\mathcal{F}} \liminf_{\substack{v\to f\\ \delta\downarrow 0}} \tfrac{1}{\delta}\Big[J^*(x+\delta v,\Theta(t+\delta)) - \textstyle\int_\delta^T L^p d\tau - \int_T^{T_\delta} L^p d\tau - W^p_{T_\delta}(\hat{\Theta}^p_T) - \delta L^p|_\delta\Big] \\
&\le \max_{f\in\mathcal{F}} \lim_{\delta\downarrow 0} \tfrac{1}{\delta}\Big[J^*(x^p(\delta),\hat{\Theta}^p(\delta)) - \textstyle\int_\delta^{T_\delta} L^p d\tau - W^p_{T_\delta}(\hat{\Theta}^p_{T_\delta})\Big] - \delta L^p|_\delta \\
&\le -L(x,\kappa_{mpc}(x,\Theta))
\end{aligned}$$

The final inequalities are achieved by recognizing:

- the $\int L^p d\tau + W^p$ term is a (potentially) suboptimal cost on the interval $[\delta, T_\delta]$, starting from the point $(x^p(\delta), \hat{\Theta}_p(\delta))$.
- The relation $\hat{\Theta}^p_{T_\delta} \subseteq \hat{\Theta}^p_T$ holds by Criterion C13.6.4, which implies by Criterion C13.10.2 that $W^p_{T_\delta}(\hat{\Theta}^p_{T_\delta}) \leq W^p_{T_\delta}(\hat{\Theta}^p_T)$
- by C13.7, $\Theta(t+\delta) \triangleq \Psi(\Theta(t), x_{[0,\delta]}, u_{[0,\delta]}) \subseteq \Psi_p(\Theta(t), x_{[0,\delta]}, u_{[0,\delta]})$, along any locus connecting x and $x + \delta v$.
- the $\liminf_v$ applies over all sequences $\{v_k\} \to f$, of which the particular 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 $\{v(\delta_k)\}$.
- The cost $J^*(x^p(\delta), \hat{\Theta}^p(\delta))$ is optimal on $[\delta, T_\delta]$, and passes through the same point $(x^p(\delta), \hat{\Theta}_p(\delta))$ as the trajectory defining the L^p and W^p expressions. Thus, the bracketed expression is non-positive.

For the purposes of condition (ii.), let x^v denote a solution to the prediction model (25b) for initial condition $x^v(-\delta) = x - \delta v$. Condition (ii.) then follows from:

$$
\begin{aligned}
\min_{f\in\mathcal{F}} \overleftarrow{D} J^*(x,\Theta) &= \min_{f\in\mathcal{F}} \limsup_{\substack{v\to f\\ \delta\downarrow 0}} \tfrac{1}{\delta}\left[\int_{-\delta}^{T-\delta} L^v d\tau + W^v_{T-\delta}(\hat{\Theta}^v_{T-\delta}) - J^*(x,\Theta)\right] \\
&\geq \min_{f\in\mathcal{F}} \limsup_{\substack{v\to f\\ \delta\downarrow 0}} \tfrac{1}{\delta}\Big[\delta L^v|_{-\delta} + \int_0^{T-\delta} L^v d\tau + W^v_{T-\delta}(\hat{\Theta}^v_{T-\delta}) - J^*(x,\Theta) \\
&\qquad + \Big(\delta L(x^v_{T-\delta}, k_f(x^v_{T-\delta}, \hat{\Theta}^v_{T-\delta})) + W^v_T(\hat{\Theta}^v_{T-\delta}) - W^v_{T-\delta}(\hat{\Theta}^v_{T-\delta})\Big)\Big] \\
&\geq \min_{f\in\mathcal{F}} \limsup_{\substack{v\to f\\ \delta\downarrow 0}} \tfrac{1}{\delta}\left[\delta L^v|_{-\delta} + \int_0^{T} L^v d\tau + W^v_T(\hat{\Theta}^v_{T-\delta}) - J^*(x,\Theta)\right] \\
&\geq \min_{f\in\mathcal{F}} \lim_{\delta\downarrow 0} \tfrac{1}{\delta}\left[\delta L^p|_{-\delta} + \int_0^{T} L^p d\tau + W^p_T(\hat{\Theta}^p_T) - J^*(x,\Theta)\right] \\
&\geq L(x, \kappa_{mpc}(x,\Theta))
\end{aligned}
$$

The above derivation made use of the fact that the reverse subderivate $\overleftarrow{D} W$ satisfies

$$
\min_{f\in\mathcal{F}} \limsup_{\substack{v\to f\\ \delta\downarrow 0}} \left(-L(x-\delta v, k_f(x-\delta v, \Theta)) + \left(\frac{W(x-\delta v,\Theta) - W(x,\Theta)}{\delta}\right)\right) \geq 0
$$

which follows from a combination of C13.9.5 and the $\mathcal{LS}$-continuity of W.

Using the above inequalities for $\overleftarrow{D} J^*(x,\Theta)$ and $\overrightarrow{D} J^*(x,\Theta)$ together with Assumption 13.3, it follows that $J^*(t)$ is strictly decreasing on $x \notin \Sigma^o_x$ and non-increasing on $x \in \Sigma^o_x$. It follows that $\lim_{t\to\infty}(x,\Theta)$ must converge to an invariant subset of $\Sigma^0_x \times \overline{\text{co}}\,\{\Theta^o\}$. Assumption 13.1 guarantees that such an invariant subset exists, since it implies $\exists \delta^* > 0$ such $\Sigma_x(B(\theta^*, \delta^*)) \neq \emptyset$, with θ^* the actual unknown parameter in (19). Continued solvability of (25) as $(x(t), \Theta(t))$ evolve follows by: 1) $x(\tau) \notin \mathbb{X}_0(\Theta(\tau)) \Rightarrow J^*(\tau) = +\infty$, and 2) if $x(t) \in \mathbb{X}_0(\Theta(t))$ and $x(t') \notin \mathbb{X}_0(\Theta(t'))$, then $(t' - t) \downarrow 0$ contradicts either condition (i.) at time t, or (ii.) at time t'.

16.2 Proof of Proposition 14.1

The fact that C13.10 holds is a direct property of the union and min operations for the closed sets $\mathbb{X}_f^i$, and the fact that the Θ-dependence of individual $(W^i, \mathbb{X}_f^i)$ satisfies C13.10. For the purposes of C13.9, the Θ argument is a constant, and is omitted from notation. Properties C13.9.1 and C13.9.2 follow directly by (27), the closure of $\mathbb{X}_f^i$, and (2). Define

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

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

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

It then follows that $u = k_f(x) \triangleq k_f^{i(x)}(x)$ satisfies C13.9.5 for any arbitrary selection rule $i(x) \in \mathcal{I}_f(x)$ (from which C13.9.3 is obvious). Condition C13.9.4 follows from continuity of the $x(\cdot)$ flows, and observing that by (26), C13.9.5 would be violated at any point of departure from $\mathbb{X}_f$.

16.3 Proof of Claim 14.3

By contradiction, let θ^* be a value contained in the left-hand side of (29), but not in the right-hand side. Then by (28), there exists $\tau \in [a, c]$ (i.e., $\tau_a \equiv (\tau - a) \in [0,\ c-a]$) such that

$$f(B(x, \gamma\tau_a), u, \theta^*, \mathcal{D}) \cap B(\dot{x}, \delta + \gamma\tau_a) = \emptyset \tag{31}$$

Using the bounds indicated in the claim, the following inclusions hold when $\tau \in [a, b]$:

$$f(x', u, \theta^*, \mathcal{D}) \subseteq f(B(x, \gamma\tau_a), u, \theta^*, \mathcal{D}) \tag{32a}$$
$$B(\dot{x}', \delta') \subseteq B(\dot{x}, \delta + \gamma\tau_a) \tag{32b}$$

Combining (32) and (31) yields

$$f(x', u, \theta^*, \mathcal{D}) \cap B(\dot{x}', \delta') = \emptyset \quad \Longrightarrow \quad \theta^* \notin \mathbb{Z}^{\delta'}(\Theta, x'_{[a,\tau]}, u_{[a,\tau]}) \tag{33}$$

which violates the initial assumption that θ^* is in the LHS of (29). Meanwhile, for $\tau \in [b, c]$ the inclusions

$$f(B(x', \gamma\tau_b), u, \theta^*, \mathcal{D}) \subseteq f(B(x, \gamma\tau_a), u, \theta^*, \mathcal{D}) \tag{34a}$$
$$B(\dot{x}', \delta + \gamma\tau_b) \subseteq B(\dot{x}, \delta + \gamma\tau_a) \tag{34b}$$

yield the same contradictory conclusion:

$$f(B(x', \gamma\tau_b), u, \theta^*, \mathcal{D}) \cap B(\dot{x}', \delta + \gamma\tau_b) = \emptyset \tag{35a}$$
$$\Longrightarrow\ \theta^* \notin \mathbb{Z}^{\delta,\gamma}\left(\mathbb{Z}^{\delta'}(\Theta, x'_{[a,b]}, u_{[a,b]}), x'_{[b,\tau]}, u_{[b,\tau]}\right) \tag{35b}$$

It therefore follows that the containment indicated in (29) necessarily holds.

16.4 Proof of Proposition 14.4

It can be shown that Assumption 13.3, together with the compactness of Σ_x, is sufficient for an analogue of Claim **??** to hold (i.e., with J^*_∞ interpreted in a min − max sense). In other words, the cost $J^*(x,\Theta)$ satisfies

$$\alpha_l(\|x\|_{\Sigma_x^o},\Theta) \le J^*(x,\Theta) \le \alpha_h(\|x\|_{\Sigma_x^o},\Theta)$$

for some functions α_l, α_h which are class-$\mathcal{K}_\infty$ w.r.t. x, and whose parameterization in Θ satisfies $\alpha_i(x,\Theta_1) \le \alpha_i(x,\Theta_2), \Theta_1 \subseteq \Theta_2$. We then define the compact set $\bar{\mathbb{X}}_0^\uparrow \triangleq \{x \mid \min_{\Theta\in\overline{\text{cov}}\{\Theta^o\}} J^*(x,\Theta) < \max_{x_0\in\bar{\mathbb{X}}_0} \alpha_h(\|x_0\|_{\Sigma_x^o},\Theta^0)\}$.

By a simple extension of (Khalil, 2002, Thm4.19), the ISS property follows if it can be shown that there exists $\alpha_c \in \mathcal{K}$ such that $J^*(x,\Theta)$ satisfies

$$x\in\bar{\mathbb{X}}_0^\uparrow\backslash B(\Sigma_x^o,\alpha_c(c)) \Rightarrow \begin{cases} \max_{f\in\mathcal{F}_c} \overrightarrow{D} J^*(x,\Theta) < 0 \\ \min_{f\in\mathcal{F}_c} \overleftarrow{D} J^*(x,\Theta) > 0 \end{cases} \tag{36}$$

where $\mathcal{F}_c \triangleq B(f(x,\kappa_{mpc}(x,\Theta(t)),\Theta(t),\mathcal{D}),\, c)$. To see this, it is clear that J decreases until $x(t)$ enters $B(\Sigma_x^o,\alpha_c(c))$. While this set is not necessarily invariant, it is contained within an invariant, compact levelset $\Omega(c,\Theta) \triangleq \{x \mid J^*(x,\Theta) \le \alpha_h(\alpha_c(c),\Theta)\}$. By C13.6.4, the evolution of $\Theta(t)$ in (30b) must approach some constant interior bound Θ^∞, and thus $\lim_{t\to\infty} x(t) \in \Omega(c,\Theta^\infty)$. Defining $\alpha_d(c) \triangleq \max_{x\in\Omega(c,\Theta^\infty)} \|x\|_{\Sigma_x^o}$ completes the Proposition, if c^* is sufficiently small such that $B(\Sigma_x^o,\alpha_d(c^*)) \subseteq \bar{\mathbb{X}}_0^\uparrow$.

Next, we only prove decrease in the forward direction, since the reverse direction follows analogously, as it did in the proof of Theorem 13.11. Using similar procedure and notation as the Thm 13.11 proof, $x^p_{[0,T]}$ denotes any worst-case prediction at (t,x,Θ), extended to $[T,T_\delta]$ via k_f, that is assumed to satisfy the specifications of Proposition 14.4. Following the proof of Theorem 13.11,

$$\begin{aligned}
\max_{f\in\mathcal{F}_{c^*}} \overrightarrow{D} J^*(x,\Theta) &\le \max_{f\in\mathcal{F}} \limsup_{\substack{v\to f\\ \delta\downarrow 0}} \tfrac{1}{\delta}\Big[J^*(x+\delta v,\Theta(t+\delta)) - \textstyle\int_\delta^{T_\delta} L^p d\tau - W^p_{T_\delta}(\hat{\Theta}^p_T)\Big] - L^p|_\delta \\
&\le \max_{f\in\mathcal{F}} \limsup_{\substack{v\to f\\ \delta\downarrow 0}} \tfrac{1}{\delta}\Big[J^*(x+\delta v,\Theta(t+\delta)) - \textstyle\int_\delta^{T_\delta} L^v d\tau - W^v_{T_\delta}(\hat{\Theta}^v_{T_\delta})\Big] - L^p|_\delta \\
&\quad + \tfrac{1}{\delta}\Big[\textstyle\int_\delta^{T_\delta} L^v d\tau + W^v_{T_\delta}(\hat{\Theta}^v_{T_\delta}) - \int_\delta^{T_\delta} L^p d\tau - W^p_{T_\delta}(\hat{\Theta}^p_T)\Big]
\end{aligned} \tag{37}$$

where L^v, W^v denote costs associated with a trajectory $x^v_{[0,T_\delta]}$ satisfying the following:

- initial conditions $x^v(0) = x$, $\Theta^v(0) = \Theta$.
- generated by the same worst-case $\hat{\theta}$ and $d(\cdot)$ as $x^p_{[0,T_\delta]}$
- dynamics of form (30) on $\tau \in [0,\delta]$, and of form (25b),(25c) on $\tau \in [\delta, T_\delta]$, with the trajectory passing through $x^v(\delta) = x + \delta v$, $\Theta^v_p(\delta) = \Theta(t+\delta)$.
- the $\min_\kappa$ in (25) is constrained such that $\kappa^v(\tau,x^v,\Theta^v) = \kappa^p(\tau,x^p,\Theta^p)$; i.e., $u^v_{[0,T_\delta]} \equiv u^p_{[0,T_\delta]} \equiv u_{[0,T_\delta]}$.

Let K_f denote a Lipschitz constant of (19) with respect to x, over the compact domain $\bar{\mathbb{X}}_0^{\uparrow} \times \Theta^o \times \mathcal{D}$. Then, using the comparison lemma (Khalil, 2002, Lem3.4) one can derive the bounds

$$\tau \in [0, \delta]: \quad \begin{cases} \|x^v - x^p\| \leq \frac{c}{K_f}(e^{K_f \tau} - 1) \\ \|\dot{x}^v - \dot{x}^p\| \leq c\, e^{K_f \tau} \end{cases} \tag{38a}$$

$$\tau \in [\delta, T_\delta]: \quad \begin{cases} \|x^v - x^p\| \leq \frac{c}{K_f}(e^{K_f \delta} - 1)\, e^{K_f(\tau - \delta)} \\ \|\dot{x}^v - \dot{x}^p\| \leq c\,(e^{K_f \delta} - 1)\, e^{K_f(\tau - \delta)} \end{cases} \tag{38b}$$

As $\delta \downarrow 0$, the above inequalities satisfy the conditions of Claim 14.3 as long as $c^* < \min\{\gamma, (\delta - \delta'), \gamma e^{K_f T}, \frac{\gamma}{K_f} e^{K_f T}\}$, thus yielding

$$\hat{\Theta}_f^v \;=\; \Psi_f^{\delta,\gamma}(\Psi^{\delta'}(\Theta, x^v_{[0,\delta]}, u_{[0,\delta]}), x^v_{[\delta,T_\delta]}, u_{[\delta,T_\delta]}) \;\subseteq\; \Psi_f^{\delta,\gamma}(\Theta, x^p_{[0,T_\delta]}, u_{[0,T_\delta]}) \;=\; \hat{\Theta}_f^p$$

as well as the analogue $\hat{\Theta}_p^v(\tau) \subseteq \hat{\Theta}_p^p(\tau), \forall \tau \in [0, T_\delta]$.

Since $x^p_{[0,T]}$ is a feasible solution of the original problem from (t, x, Θ) with $\tau \in [0, T]$, it follows for the new problem posed at time $t + \delta$ that x^v is feasible with respect to the appropriate inner approximations of $\mathbb{X}$ and $\mathbb{X}_f^{i^*}(\hat{\Theta}_T^p) \subseteq \mathbb{X}_f(\hat{\Theta}_{T_\delta}^v)$ (where i^* denotes an active terminal set for x_f^p) if

$$\|x^v - x^p\| \leq \begin{cases} \delta \frac{\delta_x}{T} & \tau \in [\delta, T] \\ \delta\, \delta_f & \tau \in [T, T_\delta] \end{cases}$$

which holds by (38) as long as $c^* < \min\{\delta_f, \frac{\delta_x}{T}\} e^{-K_f T}$. Using arguments from the proof Theorem 13.11, the first term in (37) can be eliminated, leaving:

$$\begin{aligned} \max_{f \in \mathcal{F}_c} \overrightarrow{D} J^*(x, \Theta) &\leq \max_{f \in \mathcal{F}} \liminf_{\substack{v \to f \\ \delta \downarrow 0}} \tfrac{1}{\delta} \left[\textstyle\int_\delta^{T_\delta} L^v d\tau + W^v_{T_\delta}(\hat{\Theta}^v_{T_\delta}) - \int_\delta^{T_\delta} L^p d\tau - W^p_{T_\delta}(\hat{\Theta}^p_T) \right] - L^p|_\delta \\ &\leq \max_{f \in \mathcal{F}} \liminf_{\substack{v \to f \\ \delta \downarrow 0}} \tfrac{1}{\delta} \left(\textstyle\int_\delta^{T_\delta} K_L \|x^v - x^p\| d\tau + K_W \|x^v(T) - x^p(T)\| - L^p|_\delta \right) \\ &\leq \lim_{\delta \downarrow 0} \left(\frac{c(e^{K_f \delta} - 1)}{K_f \delta} \Big[K_W + T K_L \Big] e^{K_f T} - L^p|_\delta \right) \\ &\leq -L(x, k_{MPC}(x, \Theta)) + c(K_W + T K_L) e^{K_f T} \\ &< 0 \qquad \forall x \in \bar{\mathbb{X}}_0^{\uparrow} \backslash B(\Sigma_x^o, \alpha_c(c)) \end{aligned}$$

with $\alpha_c \in \mathcal{K}$ given by

$$\alpha_c(c) \triangleq \underline{\gamma}_L^{-1} \left(c\,(K_W + T K_L)\, e^{K_f T} \right)$$

where K_W is a Lipschitz constant of $W^{i^*}(x, \Theta)$ over the compact domain $\bar{\mathbb{X}}_0^{\uparrow} \cap \mathbb{X}_f^{i^*}(\Theta)$, maximal over all $\Theta \in \overline{\text{cov}}\{\Theta^o\}$. Likewise, K_L is a Lipschitz constant of $L(x, u)$ with respect to x, maximal over $u \in \mathbb{U}$.

This proves the forward case in (36), with the reverse case following similarly. As argued previously, this is sufficient to yield the ISS property of (30) with respect to $\|d_2\| \leq c \leq c^*$, which completes the proof.

17. References

Adetola, V. & Guay, M. (2004). Adaptive receding horizon control of nonlinear systems, *Proc. IFAC Symposium on Nonlinear Control Systems*, Stuttgart, Germany, pp. 1055–1060.

Aubin, J. (1991). *Viability Theory*, Systems & Control: Foundations & Applications, Birkhäuser, Boston.

Bellman, R. (1952). The theory of dynamic programming, *Proc. National Academy of Science,*, number 38, USA.

Bellman, R. (1957). *Dynamic Programming*, Princeton Press.

Bertsekas, D. (1995). *Dynamic Programming and Optimal Control*, Vol. I, Athena Scientific, Belmont, MA.

Brogliato, B. & Neto, A. T. (1995). Practical stabilization of a class of nonlinear systems with partially known uncertainties, *Automatica* **31**(1): 145 – 150.

Bryson, A. & Ho, Y. (1969). *Applied Optimal Control*, Ginn and Co., Waltham, MA.

Cannon, M. & Kouvaritakis, B. (2005). Optimizing prediction dynamics for robust MPC, **50**(11): 1892–1897.

Chen, H. & Allgöwer, F. (1998a). A computationally attractive nonlinear predictive control scheme with guaranteed stability for stable systems, *Journal of Process Control* **8**(5-6): 475–485.

Chen, H. & Allgöwer, F. (1998b). A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability, *Automatica* **34**(10): 1205–1217.

Chen, H., Scherer, C. & Allgöwer (1997). A game theoretic approach to nonlinear robust receding horizon control of constrained systems, *Proc. American Control Conference*.

Clarke, F., Ledyaev, Y., Stern, R. & Wolenski, P. (1998). *Nonsmooth Analysis and Control Theory*, Grad. Texts in Math. 178, Springer-Verlag, New York.

Corless, M. J. & Leitmann, G. (1981). Continuous state feedback guaranteeing uniform ultimate boundedness for uncertain dynamic systems., *IEEE Trans. Automat. Contr.* **AC-26**(5): 1139 – 1144.

Coron, J. & Rosier, L. (1994). A relation between continuous time-varying and discontinuous feedback stabilization, *Journal of Mathematical Systems, Estimation, and Control* **4**(1): 67–84.

Cutler, C. & Ramaker, B. (1980). Dynamic matrix control - a computer control algorithm, *Proceedings Joint Automatic Control Conference*, San Francisco, CA.

De Nicolao, G., Magni, L. & Scattolini, R. (1996). On the robustness of receding horizon control with terminal constraints, *IEEE Trans. Automat. Contr.* **41**: 454–453.

Findeisen, R., Imsland, L., Allgöwer, F. & Foss, B. (2003). Towards a sampled-data theory for nonlinear model predictive control, *in* C. Kang, M. Xiao & W. Borges (eds), *New Trends in Nonlinear Dynamics and Control, and their Applications*, Vol. 295, Springer-Verlag, New York, pp. 295–313.

Freeman, R. & Kokotović, P. (1996a). Inverse optimality in robust stabilization, *SIAM Journal of Control and Optimization* **34**: 1365–1391.

Freeman, R. & Kokotović, P. (1996b). *Robust Nonlinear Control Design*, Birkh auser.

Grimm, G., Messina, M., Tuna, S. & Teel, A. (2003). Nominally robust model predictive control with state constraints, *Proc. IEEE Conf. on Decision and Control*, pp. 1413–1418.

Grimm, G., Messina, M., Tuna, S. & Teel, A. (2004). Examples when model predictive control is non-robust, *Automatica* **40**(10): 1729–1738.

Grimm, G., Messina, M., Tuna, S. & Teel, A. (2005). Model predictive control: for want of a local control lyapunov function, all is not lost, *IEEE Trans. Automat. Contr.* **50**(5): 617–628.

Hermes, H. (1967). Discontinuous vector fields and feedback control, *in* J. Hale & J. LaSalle (eds), *Differential Equations and Dynamical Systems*, Academic Press, New York, pp. 155–166.

Hestenes, M. (1966). *Calculus of Variations and Optimal Control*, John Wiley & Sons, New York.

Jadbabaie, A., Yu, J. & Hauser, J. (2001). Unconstrained receding-horizon control of nonlinear systems, *IEEE Trans. Automat. Contr.* **46**(5): 776 – 783.

Kalman, R. (1960). Contributions to the theory of optimal control, *Bol. Soc. Mat. Mexicana* **5**: 102–119.

Kalman, R. (1963). Mathematical description of linear dynamical systems, *SIAM J. Control* **1**: 152–192.

Keerthi, S. S. & Gilbert, E. G. (1988). 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.

Khalil, H. (2002). *Nonlinear Systems*, 3rd edn, Prentice Hall, Englewood Cliffs, N.J.

Kim, J.-K. & Han, M.-C. (2004). Adaptive robust optimal predictive control of robot manipulators, *IECON Proceedings (Industrial Electronics Conference)* **3**: 2819 – 2824.

Kothare, M., Balakrishnan, V. & Morari, M. (1996). Robust constrained model predictive control using linear matrix inequalities, *Automatica* **32**(10): 1361–1379.

Kouvaritakis, B., Rossiter, J. & Schuurmans, J. (2000). Efficient robust predictive control, *IEEE Trans. Automat. Contr.* **45**(8): 1545 – 1549.

Langson, W., Chryssochoos, I., Raković, S. & Mayne, D. (2004). Robust model predictive control using tubes, *Automatica* **40**(1): 125 – 133.

Lee, E. & Markus, L. (1967). *Foundations of Optimal Control Theory*, Wiley.

Lee, J. & Yu, Z. (1997). Worst-case formulations of model predictive control for systems with bounded parameters, *Automatica* **33**(5): 763–781.

Magni, L., De Nicolao, G., Scattolini, R. & Allgöwer, F. (2003). Robust model predictive control for nonlinear discrete-time systems, *International Journal of Robust and Nonlinear Control* **13**(3-4): 229–246.

Magni, L., Nijmeijer, H. & van der Schaft, A. (2001). Receding-horizon approach to the nonlinear h_∞ control problem, *Automatica* **37**(3): 429 – 435.

Magni, L. & Sepulchre, R. (1997). Stability margins of nonlinear receding-horizon control via inverse optimality, *Systems and Control Letters* **32**: 241–245.

Marruedo, D., Alamo, T. & Camacho, E. (2002). Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties, *Proc. IEEE Conf. on Decision and Control*, pp. 4619–4624.

Mayne, D. (1995). Optimization in model based control, *Proc. IFAC symposium on dynamics and control, chemical reactors and batch processes (DYCORD)*, Oxford: Elsevier Science., pp. 229–242. plenary address.

Mayne, D. Q. & Michalska, H. (1990). Receding horizon control of non-linear systems, *IEEE Trans. Automat. Contr.* **35**(5): 814–824.

Mayne, D. Q. & Michalska, H. (1993). Adaptive receding horizon control for constrained nonlinear systems, *Proc. IEEE Conf. on Decision and Control*, pp. 1286–1291.

Mayne, D. Q., Rawlings, J. B., Rao, C. V. & Scokaert, P. O. M. (2000). Constrained model predictive control: Stability and optimality, *Automatica* **36**: 789–814.

Michalska, H. & Mayne, D. (1993). Robust receding horizon control of constrained nonlinear systems, *IEEE Trans. Automat. Contr.* **38**(11): 1623 – 1633.

Pontryagin, L. (1961). Optimal regulation processes, *Amer. Math. Society Trans., Series 2* **18**: 321–339.

Primbs, J. (1999). *Nonlinear Optimal Control: A Receding Horizon Approach*, PhD thesis, California Institute of Technology, Pasadena, California.

Primbs, J., Nevistic, V. & Doyle, J. (2000). A receding horizon generalization of pointwise min-norm controllers, *IEEE Trans. Automat. Contr.* **45**(5): 898–909.

Raković, S. & Mayne, D. (2005). Robust time optimal obstacle avoidance problem for constrained discrete time systems, *Proc. IEEE Conf. on Decision and Control*.

Ramirez, D., Alamo, T. & Camacho, E. (2002). Efficient implementation of constrained min-max model predictive control with bounded uncertainties, *Proc. IEEE Conf. on Decision and Control*, pp. 3168–3173.

Richalet, J., Rault, A., Testud, J. & Papon, J. (1976). Algorithmic control of industrial processes, *Proc. IFAC symposium on identification and system parameter estimation*, pp. 1119–1167.

Richalet, J., Rault, A., Testud, J. & Papon, J. (1978). Model predictive heuristic control: Applications to industrial processes, *Automatica* **14**: 413–428.

Sage, A. P. & White, C. C. (1977). *Optimum Systems Control*, 2nd edn, Prentice-Hall.

Scokaert, P. & Mayne, D. (1998). Min-max feedback model predictive control for constrained linear systems, *IEEE Trans. Automat. Contr.* **43**(8): 1136–1142.

Sepulchre, R., Jankovic, J. & Kokotovic, P. (1997). *Constructive Nonlinear Control*, Springer, New York.

Sontag, E. (1989). A “universal" construction of artstein’s theorem on nonlinear stabilization, *Systems and Control Letters* **13**: 117–123.

Sontag, E. D. (1983). Lyapunov-like characterization of asymptotic controllability., *SIAM Journal on Control and Optimization* **21**(3): 462 – 471.

Tang, Y. (1996). Simple robust adaptive control for a class of non-linear systems: an adaptive signal synthesis approach, *International Journal of Adaptive Control and Signal Processing* **10**(4-5): 481 – 488.

Tuna, S., Sanfelice, R., Messina, M. & Teel, A. (2005). Hybrid MPC: Open-minded but not easily swayed, *International Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control*, Freudenstadt-Lauterbad, Germany, pp. 169–180.

3

A new kind of nonlinear model predictive control algorithm enhanced by control lyapunov functions

Yuqing He and Jianda Han
State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences
P.R.China

1. Introduction

With the abilities of handling constraints and performance of optimization, model based predictive control (MPC), especially linear MPC, has been extensively researched in theory and applied in practice since it was firstly proposed in 1970s (Qin & Badgwell, 2003). However, when used in systems with heavy nonlinearities, nonlinear MPC (NMPC) results often in problems of high computational cost or closed loop instability due to their complicated structure. This is the reason why the gaps between NMPC theory and its applications in reality are larger and larger, and why researches on NMPC theory absorbs numerous scholars (Chen & Shaw, 1982; Henson, 1998 ; Mayne, et al., 2000 ; Rawlings, 2000). When the closed loop stability of NMPC is concerned, some extra strategies is necessary, such as increasing the length of the predictive horizon, superinducing state constraints, or introducing Control Lyapunov Functions (CLF).

That infinite predictive/control horizon (in this chapter, predictive horizon is assumed equal to control horizon) can guarantee the closed loop stability is natural with the assumption of feasibility because it implicates zero terminal state, which is a sufficient stability condition in many NMPC algorithm (Chen and Shaw, 1982). In spite of the inapplicability of infinite predictive horizon in real plants, a useful proposition originated from it makes great senses during the development of NMPC theory, i.e., a long enough predictive horizon can guarantee the closed loop stability for most systems (Costa & do Val, 2003; Primbs & Nevistic, 2000). Many existing NMPC algorithm is on the basis of this result, such as Chen & Allgower (1998), Magni et al. (2001). Although long predictive horizon scheme is convenient to be realized, the difficulty to obtain the corresponding threshold value makes this scheme improper in many plants, especially in systems with complicated structure. For these cases, another strategy, superinducing state constraints or terminal constraints, is a good substitue. A typical predictive control algorithm using this strategy is the so called dual mode predictive control(Scokaert et al., 1999 ; Wesselowske and Fierro, 2003 ; Zou et al., 2006), which is originated from the predictive control with zero terminal state constrains and can increase its the stability region greatly. CLF is a new introduced

concept to design nonlinear controller. It is firstly used in NMPC by Primbs et al. in 1999 to obtain two typical predictive control algorithm with guaranteed stability.
Unfortunately, each approach above will result in huge computational burden simultaneously since they bring either more constraints or more optimizing variables. It is well known that the high computational burden of NMPC mainly comes from the online optimization algorithm, and it can be alleviated by decreasing the number of optimized variables. But this often deteriorates the closed loop stability due to the changed structure of optimal control problem at each time step.
In a word, the most important problem during designing NMPC algorithm is that the stability and computational burden are deteriorated by each other. Another problem, seldom referred to but top important, is that the stability can only be guarangteed under the condition of perfect optimization algorithm that is impossible in reality. Thus, how to design a robustly stable and fast NMPC algorithm has been one of the most difficult problems that many researchers are pursued.
In this chapter, we attempt to design a new stable NMPC which can partially solve the problems referred to above. CLF, as a new introduced concept to design nonlinear controller by directly using the idea of Lyapunov stability analysis, is used in this chapter to ensure the stability. Firstly, a generalized pointwise min-norm (GPMN) controller (a stable controller design method) based on the concept of CLF is designed. Secondly, a new stable NMPC algorithm, called GPMN enhanced NMPC (GPMN-ENMPC), is given through parameterized GPMN controller. The new algorithm has the following two advantages, 1) it can not only ensure the closed loop stability but also decrease the computational cost flexibly at the price of sacrificing the optimality in a certain extent; 2) a new tool of guide function is introduced by which some extra control strategy can be considered implicitly. Subsequently, the GPMN-ENMPC algorithm is generalized to obtain a robust NMPC algorithm with respect to the feedback linearizable system. Finally, extensive simulations are conducted and the results show the feasibility and validity of the proposed algorithm.

2. Concept of CLF

The nonlinear system under consideration in this chapter is in the form as:

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ u &\in U \subset R^m \end{aligned} \tag{1}$$

where $x \subset R^n$ is state vector, $u \subset R^m$ is input vector, $f(*)$ and $g(*)$ are nonlinear smooth functions with $f(0) = 0$. U is the control constraint.

Definition I:

For system (1), if there exists a C_1 function $V(x)$: $x \subset R^n \rightarrow R^+ \cup \{0\}$, such that

1) $V(0) = 0$, $V(x) > 0$ if $x \neq 0$;
2) $a_1(||x||) < V(x) < a_2(||x||)$, where $a_1(*)$ and $a_2(*)$ are class K_∞ functions;
3) $\inf_{u \in U \subset R^m} [V_x(x) f(x) + V_x(x) g(x) u] < 0, \quad \forall x \in \Omega_c - \{0\}$,where $\Omega_c \triangleq \{x \in R^n : V(x) \leq c\}$.

then $V(x)$ is called a CLF of system (1). Moreover, if x can be chosen as R^n and $V(x)$ satisfies the following condition,

$$V(x)\rightarrow\infty \Longrightarrow ||x|| \rightarrow\infty$$

then $V(x)$ is called a global CLF of system (1). ∎
If system (1) has uncertainty terms, i.e.,

$$\begin{aligned} \dot{x} &= f(x) + g(x)u + l(x)\omega \\ y &= h(x) \\ u &\in U \subset R^m \end{aligned} \tag{2}$$

where $\omega \in R^q$ is external disturbance; $l(*)$ and $h(*)$ are pre-defined nonlinear smooth functions; y is the interested output. We have the following concept of robust version CLF – called H_∞CLF,

Definition II,

For system (2), if there exists a C_1 function $V(x)$: $x \subset R^n \rightarrow R^+ \cup \{0\}$, such that

1) $V(0) = 0$, $V(x) > 0$ if $x \neq 0$;
2) $a_1(||x||) < V(x) < a_2(||x||)$, where $a_1(*)$ and $a_2(*)$ are class K_∞ functions;
3) $\inf_{u\in R^m}\{V_x(x)[f(x)+g(x)u] + \frac{1}{2\gamma^2}V_x(x)l(x)l^T(x)V_x^T + \frac{1}{2}h^T(x)h(x)\} < 0, \forall x \in \Omega_{c_1} - \Omega_{c_2}$, where $c_1 > c_2$.

then $V(x)$ is called a local H_∞CLF of system (2) in $\Omega_{c_1} - \Omega_{c_2}$. Furthermore, $V(x)$ is called a global H_∞CLF if c_1 can be chosen $+\infty$ with $V(x)\rightarrow\infty$ as $|x|\rightarrow\infty$. ∎

Definition I and II indicate that if we can obtain a CLF or H_∞CLF of system (1) or (2), a 'permitted' control set can be found at every 'feasible' state, and the control action inside the set can guarantee the closed loop stability of system (1) or input output finite gain L_2 stability of system (2). Subsequently, in order to complete the controller design, what one needs to do is just to find an approach to select a sequence of control actions from the 'permitted control set', see Fig. 1.

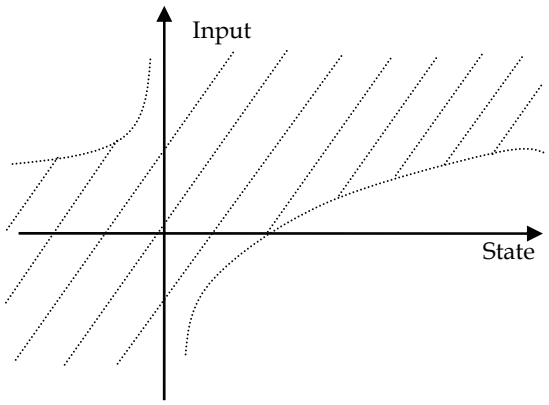


Fig. 1. Sketch of CLF, the shadow indicates the 'permitted' set of (x, u) $\dot{V}(x,u)$ along system (1)

CLF based nonlinear controller design method is also called direct method of Lyapunov function based controller design, and its difficulty is how to ensure the controller's continuousness. Thus, most recently, researchers mainly pay their attentions to designing continuous CLF based controller, and several universal formulas have been revealed. Sontag's formula (Sontag, 1989), for example, originated from the root calculation of 2nd-order equation, can be written as Eq. (3) through slightly modification by Freeman (Freeman & Kokotovic, 1996b),

$$u = \begin{cases} -\left[\dfrac{V_x(x)f + \sqrt{(V_x(x)f(x))^2 + q(x)(V_x(x)g(x)g^T(x)V_x^T(x))}}{V_x(x)g(x)g^T(x)V_x^T(x)}\right] & V_x g \neq 0 \\ 0 & V_x g = 0 \end{cases} \tag{3}$$

where $q(x)$ is a pre-designed positive definite function.
Pointwise Min-Norm (PMN) control is another well known CLF-based approach proposed by Freeman (Freeman & Kokotovic, 1996a),

$$\begin{aligned} \min_u \quad & \|u\| \\ s.t. \quad & V_x(x)[f(x) + g(x)u] \leq -\sigma(x) \\ & u \in U \end{aligned} \tag{4}$$

where $\sigma(x)$ is a pre-selected positive definite function. Controller (4) can also be explicitly denoted as (5) if the constraint set U can be selected big enough.

$$u = \begin{cases} -\dfrac{[V_x(x)f(x) + \sigma(x)]g^T(x)V_x^T(x)}{V_x(x)g(x)g^T(x)V_x^T(x)} & V_x(x)f(x) + \sigma(x) > 0 \\ 0 & V_x(x)f(x) + \sigma(x) \leq 0 \end{cases} \tag{5}$$

(3) and (5) provide two different methods on how to design continuous and stable controller based on CLF with respect to system (1). H_∞CLF with respect to system (2) is a new given concept, and there are no methods can be used to designed robust controller based on it. Although the closed loop stability can be guaranteed using controller (3) or controller (5), selection of parameters $q(x)$ or $\sigma(x)$ is too difficult to be used in real applications. This is mainly because these parameters heavily influence some inconsistent closed loop performance simultaneously. Furthermore, if the known CLF is not global, the selection of $q(x)$ and $\sigma(x)$ will also influence stability margin of the closed loop systems, which makes them more difficult to be selected (Sontag, 1989; Freeman & Kokotovic, 1996a). In this chapter, we will firstly give a new CLF based controller design strategy, which is superior compared to the existing CLF based controller design methods referred to above. Furthermore, the most important is that this new strategy can be used in designing robustly stable and fast NMPC algorithm.

3. GPMN-ENMPC

3.1 CLF based GPMN controller

Since $q(x)$ and $\sigma(x)$ in controller (3) and controller (5) are difficult to select, a guide function is proposed in this subsection into the PMN controller to obtain a new CLF based nonlinear controller with respect to system (1), in the following section, this controller will be generated with respect to system (2). In the new controller, $\sigma(x)$ is only used to ensure the stability of the closed loop, while the other desired performance of the controller, for example tracking performance, can be guaranteed by the guide function, which, as new controller parameters, can be designed without deteriorating the stability. The following proposition is the main result of this subsection.

***Proposition I*:**
If $V(x)$ is a CLF of system (1) in Ω_c and $\xi(x)$: $R^n \rightarrow R^m$ is a continuous guide function such that $\xi(0) = 0$, then, the following controller can stabilize system (1),

$$\begin{aligned} u(x) &= \arg\min_{u \in K_V(x)} \{\|u - \xi(x)\|\} \\ K_V(x) &= \{y \mid V_x(x) f(x) + V_x(x) g(x) y \le -\sigma(x),\ y \in U\} \end{aligned} \tag{6}$$

where $\sigma(x)$ is a positive definite function of state, and $\xi(x)$, called guide function, is a continuous state function.
Proof of Proposition I:
Let $V(x)$ be a Lyapunov function candidate for system (1), then we have

$$\dot{V}(x) = V_x(x) f(x) + V_x(x) g(x) u \tag{7}$$

Substitute Eq. (6) into (7), it is not difficult to obtain the following inequality,

$$\dot{V}(x) = V_x(x) f(x) + V_x(x) g(x) u \le -\sigma(x)$$

Because $\sigma(x)$ is a positive definite function, proposition I is proved. ∎

Controller (6) is called Generalized Pointwise Min-Norm (GPMN) controller. The difference between the proposed GPMN controller and the normal PMN controller of Eq. (4) can be illustrated in Fig.2: for the normal PMN algorithm (Fig. 2a), the controller output in each state point has the minimum 'permitted' norm (close to the state-axis as much as possible), while the GPMN controller's output has nearest distance from the guide function $\xi(x)$ (Fig. 2b). Thus, $\xi(x)$ in GPMN controller is actual a performance criterion which the controller is expected to pursue, while $\sigma(x)$ dedicates only on providing the 'permitted' stable control input sets.
Up to now, the design of new GPMN controller has been completed. However, in order to use a GPMN controller in reality or in NMPC algorithm, analytical form of the solution of Eq. (6) is necessary to be studied.
Firstly, if there are no input constraints (or the input constraint sets are big enough), the analytical form of controller (6) can be obtained as follows, based on the projection theory,

$$u_{\xi(x)}(x)=\begin{cases}-\dfrac{[V_x f+\sigma+V_x g\xi(x)]g^T V_x^T}{V_x g g^T V_x^T}+\xi(x), & V_x f+\sigma+V_x g\xi(x)>0\\ \xi(x) \quad , & V_x f+\sigma+V_x g\xi(x)\le 0\end{cases} \tag{8}$$

Secondly, if there exist input constraints, the analytical expression of controller (6) might be very complicated or even inexistent. Thus in this subsection, only analytical form of controller (6) with a typical super ball input constraint is researched, i.e., input constraints is as

$$U=\{(u_1,\cdots,u_m)\,|\,u_1^2+\cdots+u_m^2\le r^2\} \tag{9}$$

where (u_1, … ,u_m) is the input vector, and r is the radius of the super ball.
In order to obtain the analytical expression of Eq. (6) with input constraint as Eq. (9), we propose the following 4 steps (For a general control input constraint U, one can always find a maximal inscribed super ball B of it, and then use B replacing U before continuing the following processes):

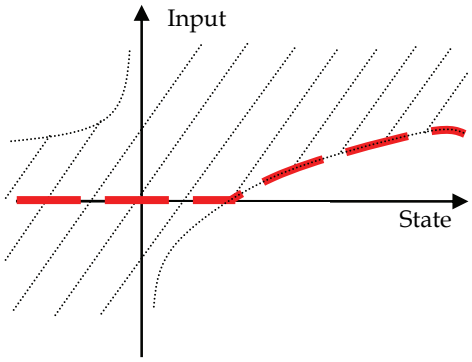


Fig. 2a. the sketch of PMN

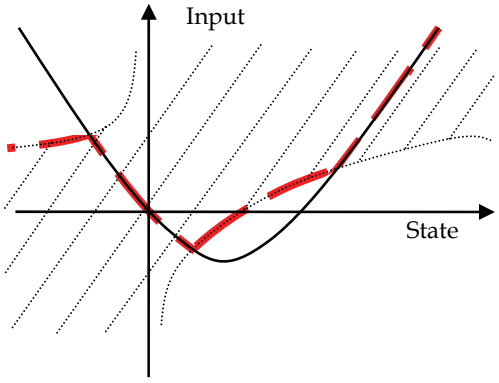


Fig. 2b. the sketch of GPMN
* the dashed line is the PMN controller in a) and the GPMN control in b); the solid line denotes the guide function of ξ(x).

Step1: For each state x, the following equation denotes a super plane in R^m ($u \in R^m$).

$$V_x f(x) + \sigma(x) + V_x g(x) u = 0 \tag{10}$$

Let d be the distance from zero to the super plane (10), we have,

$$d = \frac{\left| V_x f(x) + \sigma(x) \right|}{\sqrt{V_x g(x) g^T(x) V_x^T}} \tag{11}$$

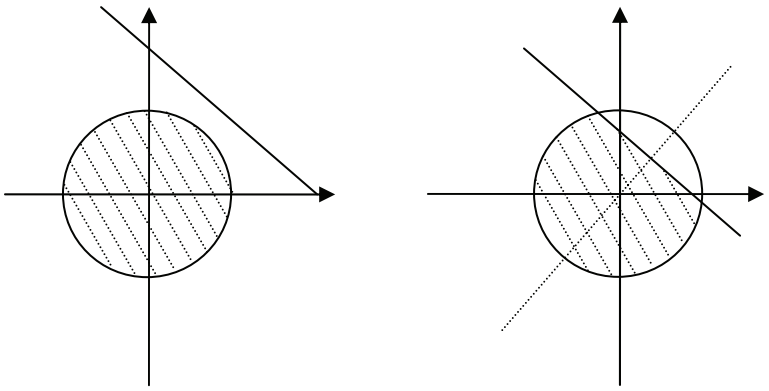


Fig. 3a

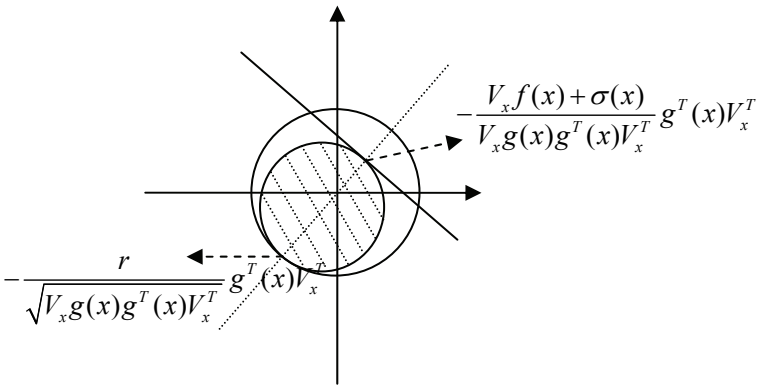


Fig. 3b.
* Sketch of the process to build the analytic GPMN controller

Step2: From Eq. (11), the 'permitted' stable control input set $K_V(x)$ in controller (6) can be denoted as Fig. 3a, where the right figure (left figure) is the case that the super plane of (10) intersects (does not intersect) with the super ball (9), and the region filled by the dotted line is the 'permitted' stable control input set. For the case denoted by the left figure of Fig. 3a, it is easy to obtain a minimal distance from any point p to $K_V(x)$, and the corresponding point, i.e., the controller's output, in $K_V(x)$ with minimal distance from p can also be obtained (the

point of intersection of the super ball (9) and the beeline connecting the centre of it and p). With respect to the case of the right figure, the maximally inscribed super ball B' is used to replace $K_V(x)$ (see Fig. 3b). Thus, the same processes as above can be used to obtain the output of controller (6).

Step 3: A new 'permitted' stable control input sets $\overline{K_V(x)}$ is defined,

$$\overline{K_V(x)} = \begin{cases} U & \dfrac{\left|V_x f(x) + \sigma(x)\right|}{\sqrt{V_x g(x) g^T(x) V_x^T}} > r \\ \{u \mid \left\|u - \gamma(x)\right\| = R^2(x)\} & \dfrac{\left|V_x f(x) + \sigma(x)\right|}{\sqrt{V_x g(x) g^T(x) V_x^T}} \le r \end{cases} \tag{12}$$

where

$$\gamma(x) = -\left(\frac{V_x f(x) + \sigma(x)}{2 V_x g(x) g^T(x) V_x^T} + \frac{r}{2\sqrt{V_x g(x) g^T(x) V_x^T}}\right) g^T(x) V_x^T$$

$$R(x) = \frac{\left| r - \dfrac{[V_x f(x) + \sigma(x)]}{\sqrt{V_x g(x) g^T(x) V_x^T}} \right|}{2}$$

It is obvious that $\overline{K_V(x)} \subseteq K_V(x)$, thus the stability of the closed loop can be ensured from Proposition I.

Step 4: The analytical expression of GPMN controller with super-ball input constraint can thus be described as

$$\overline{u}_{\xi(x)}(x) = \begin{cases} \xi(x) & \left\|\xi(x) - \gamma(x)\right\| \le R(x) \\ \dfrac{R(x)}{\left\|\xi(x) - \gamma(x)\right\|}[\xi(x) - \gamma(x)] + \gamma(x) & else \end{cases} \tag{13}$$

where $\xi(x)$ is the guide function of controller (6). ∎

From the preceding procedure, it is evidently that Eq. (13) is the solution of Eq. (6) with $K_V(x)$ being placed by $\overline{K_V(x)}$.

3.2 GPMN-ENMPC

In order to achieve a stable NMPC with reduced computational burden, we propose to use the GPMN to parameterize the control input sequence in NMPC. Assuming that $\vartheta(x, \theta)$ is a function of state x, where θ is the vector of unknown parameters, the following NMPC can be formulated,

$$
\begin{aligned}
&u^* = \vartheta(x,\theta^*)\\
&\theta^* = \arg\min_{\theta\subset IR^l} J(x,\theta)\\
&J(x,\theta) = \int_t^{t+T} l(x,\vartheta(x,\theta))d\tau\\
&s.t.\quad \dot{x} = f(x) + g(x)\vartheta(x,\theta)\\
&\qquad \vartheta(x,\theta)\in U,\ \forall t\in[t,t+T]
\end{aligned}
\tag{14}
$$

NMPC algorithm of (14) is different from the normal NMPC in the following aspect: in normal NMPC algorithm, one tries to optimize the continuous control profile of u (Mayne et al., 2000), while controller (14) tries to achieve good performance by optimizing the parameter vector θ. Thus, the computational cost of controller (14) dependents mainly on dimension of θ instead of that of control input profile in normal NMPC algorithm. The most important problem of the latter algorithm is that its computational cost increases rapidly with the control horizon. Based on (14), our new designed NMPC controller is introduced in the following proposition.

Proposition II:
Assuming $V(x)$ is a known CLF of system (1), Ω_c is the stability region of $V(x)$, then controller (14) with the following GPMN controller $\vartheta(x,\theta)$,

$$
\vartheta(x,\theta) = u(x,\theta) = \arg\min_{u\in K_V(x)} \{\|u-\xi(x,\theta)\|\}
\tag{15}
$$

($u(x,\theta)$ is the GPMN control and $\xi(x,\theta)$ the guide function in Eq. (6)), is stable in Ω_c. Furthermore, if $V(x)$ is a global CLF, controller of (14) combined with (15) is stable over R^n. (14), combined with (15), is called GPMN-Enhanced NMPC (GPMN-ENMPC).
Proof of Proposition II:
At any time instant t, by assuming that θ^* is the optimal parameters at t, control input at t can be represented as $u(x,\theta^*)$. From Proposition I, we can conclude that the control inputs $u(x,\theta^*)$ can guarantee a negative definite $\dot{V}(x)$. Due to the randomicity of t, GPMN-ENMPC actually makes the $\dot{V}(x)$ negative in any time instant, which means that the closed loop stability of controller (14) and (15) is guaranteed. ∎

3.3 Selection of $\xi(x,\theta)$

Theoretically, $\xi(x,\theta)$ in (15) can be selected in any forms since it does not influence the closed loop stability, which is guaranteed by GPMN. However, it is natural that $\xi(x,\theta)$ will influence other closed loop performances of the GPMN-ENMPC except the stability.
Since optimality is the main concern in designing NMPC algorithm, the Bellman's Optimization Principle (BOP, Lewis & Syrmos, 1995) is used to design $\xi(x,\theta)$ in this sub-section.
In BOP, with the following quadratic cost function,

$$
J(x,\theta) \overset{\Delta}{=} \int_t^{t+T} (x^T P x + u^T Q u)d\tau
\tag{16}
$$

and $J^*(x_0,\theta)$ denoting the optimal value function of $J(x_0,\theta)$ in state x_0, the following controller of system (1) is optimal,

$$u^* = -\frac{1}{2}(Q^{-1})^T g^T(x)\frac{\partial J^*}{\partial x} \tag{17}$$

Unfortunately, in most applications, it is impossible to obtain $J^*(x^*,\theta)$.
Based on the Stone-Weierstrass theorem (Brinkhuis & Tikhomirov, 2005), any continuous function defined in a bounded set can be uniformly approximated by a polynomial function,

$$B_k^{J^*}(x_1,\cdots,x_n) = \sum_{\substack{v_1,\cdots,v_n\geq 0\\ v_1+\cdots+v_n\leq k}} J^*(\frac{v_1}{k},\cdots,\frac{v_n}{k}) p_{k;v_1,\cdots,v_n}(x_1,\cdots,x_n) \tag{18}$$

where

$$\begin{aligned} p_{k;v_1,\cdots,v_n}(x_1,\cdots,x_n) &= \binom{k}{v_1,\cdots,v_n} x_1^{v_1}\cdots x_n^{v_n}(1-x_1-\cdots-x_n)^{k-v_1-\cdots-v_n},\\ \binom{k}{v_1,\cdots,v_n} &= \frac{k!}{v_1!v_2!\cdots v_n!(k-v_1-\cdots-v_n)!}, \end{aligned} \tag{19}$$

and

$$\lim_{\substack{k_i\to\infty\\ (i=1,\cdots,k)}} B_{k_1,\cdots,k_n}^{J^*}(x_1,\cdots,x_n) = J^*(x_1,\cdots,x_n) \tag{20}$$

Thus, take the coefficients of the Bernstein polynomial as the parameters θ, and select θ optimally using the NMPC algorithm, a 'quasi-optimal' function closed to $J^*(x^*,\theta)$ can be obtained. That means we can complete the design of GPMN-ENMPC algorithm by taking

$$\xi(x,\theta) = \sum_{\substack{v_1,\cdots,v_n\geq 0\\ v_1+\cdots+v_n\leq k}} \lambda_{v_1,\cdots,v_n} p_{k;v_1,\cdots,v_n}(x_1,\cdots,x_n) \tag{21}$$

where $\lambda_{v_1,\ldots,v_n}$, $v_1,\ldots,v_n \geq 0$ and $v_1+\ldots+v_n \leq k$ are the parameters to be optimized, k is the order of the Bernstein polynomial, and

$$\theta = [\lambda_{k_1,k_2,\ldots,k_n}]_{n^k\times 1} \tag{22}$$

It should be noted that the order of the Bernstein polynomial determines the consequent optimization cost, i.e., the higher the order is, the higher the computational cost is. About the GPMN-ENMPC, we have the following remarks:
Remark-1: Selection of $\xi(x,\theta)$ as Eq. (21) provides a feasible way to complete the GPMN-ENMPC of (20) and (21). By this way, the computation cost is controllable, namely, one can select the order of k to meet the CPU capability of a specific real system. This makes the GPMN-ENMPC feasible to be implemented.

Remark-2: The selection of k does not influence the closed loop stability, which has already guaranteed by the GPMN scheme. But there still exist trade-offs between computation cost and the optimal performance which is determined by $\xi(x,\theta)$.

Remark-3: Compared to nominal NMPC algorithm, the huge computational burden problem of GPMN-ENMPC algorithm is improved due to the following two reasons: 1) the dimension of optimizing variables is one of key elements which increase the computational burden of NMPC, while that of GPMN-ENMPC algorithm is independent of the predictive horizon; 2) online considerations of control input constraints are not necessary in GPMN-ENMPC algorithm since it can be dealt with offline during designing GPMN controller.

3.4 The Feasibility of GPMN-ENMPC

Another important problem, normally called the feasibility problem of NMPC, is that general NMPC algorithm may not guarantee that a control set always exists to meet all of the input and state constraints, while the proposed GPMN-ENMPC can guarantee such a control sequence always exists. This is because for any θ, from the proposition-I, one can always obtain a stable GPMN controller, i.e., $u(x,\theta)$ of (6) meeting all input and state constraints. Therefore, by Eq. (14) and (15), there will always exist a feasible control $u = \vartheta(x,\theta)$, and the task left is just to find an optimal parameter set of θ to minimize the cost function of $J(x,\theta)$ in Eq. (14).

4. H_∞ GPMN-ENMPC

In section 3, GPMN-ENMPC algorithm is introduced with respect to system (1). In this section, it will be generalized to deal with the disturbed system as Eq. (2). Firstly, an H_∞ controller with partially known disturbances is given, and then it is used to design H_∞GPMN controller, which followed by the designing process of H_∞GPMN-ENMPC.

4.1 H_∞ Control With Partially Known Disturbances

Suppose the following two assumptions are satisfied with respect to system (2),

***Assumption I*:**
System (2) is static feedback linearizable, i.e., there exists a state feedback controller $u = k(x)$ such that (2) can be transformed into a linear system without considering ω.

Assumption II:
The disturbances of system (2) are ***partially obtainable***, i.e., the variables ω can be used to construct controller.

Assumption II is reasonable because the uncertainty information ω can often be measured or estimated in reality (He & Han, 2007; Chen, 2004). Moreover, the tracking problem of general nonlinear system, where ω is composed of known desired trajectory, can also be modeled as Eq. (2). However, the higher order derivative of the disturbances with respective to time is often difficult to be obtained due to the heavy additive noise. Thus, the disturbances are often '*partially obtainable*'.

Based on assumption I, system (2) can be changed into the following equations through some coordination transformation,

$$\begin{aligned} \dot{z}_1 &= z_2 + F_1(z)\Delta \\ &\vdots \\ \dot{z}_n &= f_1(z) + g_1(z)u + F_n(z)\Delta \\ y &= z_1 \end{aligned} \tag{23}$$

where $z = [z_1, z_2, \ldots, z_n]^{\mathrm{T}}$ is the new state variable.
An H_∞ robust controller for system (23) can be designed based on the following Theorem,

***Theorem I*:**
Consider system (23), if there exists a control $u = u_1(z)$ and a radially unbounded function $V(x)$ to satisfy the following inequality,

$$\begin{aligned} &\sum_{i=1}^{n-1} V_{z_i} z_{i+1} + V_{z_n}[f_1(z) + g_1(z)u_1(z)] + \frac{2}{\gamma^2} V_z \left[F_1^T(z) \quad F_2^T(z) \quad \cdots \quad F_n^T(z) \right]^T \times \\ &\left[F_1^T(z) \quad F_2^T(z) \quad \cdots \quad F_n^T(z) \right] V_z^T + z_1^2 \le 0 \end{aligned} \tag{24}$$

Then, controller

$$\begin{aligned} u = g_1^{-1}(z)[f_1(\bar{z}) + g_1(\bar{z})u_1 - f_1(z) + \sum_{i=1}^{n} \bar{F}_2(\bar{z}, \rho, \cdots, \rho^{(i-1)})\rho^{(n-i)}] + [\bar{V}_{\bar{z}_n} g_1(z)]^{-1} \times \\ \{\frac{2}{\gamma^2} \bar{V}_{\bar{z}} \begin{bmatrix} F_1(\bar{z}) - \bar{F}_1(\bar{z}, \rho) \\ F_2(\bar{z}) - \bar{F}_2(\bar{z}, \rho, \dot{\rho}) \\ \vdots \\ F_n(\bar{z}) - \bar{F}_n(\bar{z}, \rho, \cdots, \rho^{(n-1)}) \end{bmatrix} \begin{bmatrix} F_1(\bar{z}) - \bar{F}_1(\bar{z}, \rho) \\ F_2(\bar{z}) - \bar{F}_2(\bar{z}, \rho, \dot{\rho}) \\ \vdots \\ F_n(\bar{z}) - \bar{F}_n(\bar{z}, \rho, \cdots, \rho^{(n-1)}) \end{bmatrix}^T \bar{V}_{\bar{z}}^T \} \end{aligned} \tag{25}$$

can make the system (23) finite gain L_2 stable from $\Delta+\rho$ to y, and the gain is less than or equal to γ. ρ is a new defined signal to further attenuate the disturbances.
Proof of Theorem I:
Define new variables,

$$\begin{aligned} \bar{z}_1 &= z_1 \\ \bar{z}_2 &= z_2 - F_1(z)\rho \\ &\vdots \\ \bar{z}_n &= z_n - \sum_{i=1}^{n-1} F_i(z)\rho^{(n-i)} \end{aligned} \tag{26}$$

Then, system (23) can be written as

$$\begin{aligned}
&\dot{\bar{z}}_1 = \bar{z}_2 + \bar{F}_1(\bar{z},\rho)(\Delta+\rho)\\
&\dot{\bar{z}}_2 = \bar{z}_3 + \bar{F}_2(\bar{z},\rho,\dot{\rho})(\Delta+\rho)\\
&\quad\vdots\\
&\dot{\bar{z}}_n = f_1(z) + g_1(z)u - \sum_{i=1}^{n}\bar{F}_i(\bar{z},\rho,\cdots,\rho^{(i-1)})\rho^{(n-i)} + \bar{F}_n(\bar{z},\rho,\cdots,\rho^{(n-1)})(\Delta+\rho)\\
&y = \bar{z}_1
\end{aligned} \tag{27}$$

where

$$\bar{F}_j(\bar{z},\rho,\cdots,\rho^{(j-1)}) \triangleq F_j(z)\Big|_{z_i=\bar{z}_j+\sum_{j=1}^{i-1}F_i(z)\rho^{(n-i)}}$$

Let

$$\bar{V}(\bar{z}) = V(z)\,|_{z=\bar{z}} \tag{28}$$

where $\bar{z} = \begin{bmatrix}\bar{z}_1 & \bar{z}_2 & \cdots & \bar{z}_n\end{bmatrix}^T$. Computing the HJI equation (Khalil, 2002) of system (27) with respect to $\bar{V}(\bar{z})$, we have,

$$\begin{aligned}
&\sum_{i=1}^{n-1}\bar{V}_{z_i}\bar{z}_{i+1} + \bar{V}_{\bar{z}_n}[f_1(z)+g_1(z)u - \sum_{i=1}^{n}\bar{F}_2(\bar{z},\rho,\cdots,\rho^{(i-1)})\rho^{(n-i)}] +\\
&\frac{2}{\gamma^2}\bar{V}_{\bar{z}}\begin{bmatrix}\bar{F}_1^T(\bar{z},\rho) & \bar{F}_2^T(\bar{z},\rho,\dot{\rho}) & \cdots & \bar{F}_n(\bar{z},\rho,\cdots,\rho^{(n-1)})\end{bmatrix}^T \times\\
&\begin{bmatrix}\bar{F}_1^T(\bar{z},\rho) & \bar{F}_2^T(\bar{z},\rho,\dot{\rho}) & \cdots & \bar{F}_n(\bar{z},\rho,\cdots,\rho^{(n-1)})\end{bmatrix}\bar{V}_{\bar{z}}^T + \bar{z}_1^2
\end{aligned} \tag{29}$$

Thus, combine controller (25) and Eq. (29), we have,

$$\begin{aligned}
(29) &= \sum_{i=1}^{n-1}\bar{V}_{z_i}\bar{z}_{i+1} + \bar{V}_{\bar{z}_n}[f_1(z)+g_1(z)u - \sum_{i=1}^{n}\bar{F}_2(\bar{z},\rho,\cdots,\rho^{(i-1)})\rho^{(n-i)}] +\\
&\quad\frac{2}{\gamma^2}\bar{V}_{\bar{z}}\begin{bmatrix}\bar{F}_1^T(\bar{z},\rho) & \bar{F}_2^T(\bar{z},\rho,\dot{\rho}) & \cdots & \bar{F}_n^T(\bar{z},\rho,\cdots,\rho^{(n-1)})\end{bmatrix}^T \times\\
&\quad\begin{bmatrix}\bar{F}_1^T(\bar{z},\rho) & \bar{F}_2^T(\bar{z},\rho,\dot{\rho}) & \cdots & \bar{F}_n^T(\bar{z},\rho,\cdots,\rho^{(n-1)})\end{bmatrix}\bar{V}_{\bar{z}}^T + \bar{z}_1^2\\
&= \{\sum_{i=1}^{n-1}V_{z_i}z_{i+1} + V_{\bar{z}_n}[f_1(z)+g_1(z)u] + \frac{2}{\gamma^2}V_z\begin{bmatrix}F_1^T(z) & F_2^T(z) & \cdots & F_n^T(z)\end{bmatrix}^T \times\\
&\quad\begin{bmatrix}F_1^T(z) & F_2^T(z) & \cdots & F_n^T(z)\end{bmatrix}V_z^T + z_1^2\}\,|_{z=\bar{z}}\\
&\le 0
\end{aligned} \tag{30}$$

Based on theorem 5.5 in reference (Khalil, 2002), controller (25) can make system (23) finite gain L_2 stable from $\Delta+\rho$ to y, and the L_2 gain is less than or equal to γ. ∎

Furthermore, ρ can be used to further attenuate the disturbances which are partially obtainable from assumption II by the following equation,

$$\rho(s) = \frac{B(s)}{A(s)}\Delta(s) \tag{31}$$

where s is the Laplace operator. Thus, the new external disturbances $\Delta+\rho$ can be denoted as,

$$\Delta(s) + \rho(s) = \frac{A(s) + B(s)}{A(s)}\Delta(s) \tag{32}$$

From Eq. (32), proper A(s) and B(s) is effective for attenuating the influence of external disturbances on the closed loop system. Thus, we have designed an H_∞ controller (25) and (31) with partially known uncertainty information.

4.2 H_∞ GPMN Controller Based on Control Lyapunov Functions

In this sub-section, by using the concept of H_∞CLF, H_∞ GPMN controller is designed as following proposition,

***Proposition III**:*

If $V(x)$ is a local H_∞CLF of system (23), and $\xi(x)$: $R^n \rightarrow R^m$ is a continuous guide function such that $\xi(0) = 0$, then, the following controller, called H_∞GPMN, can make system (23) finite gain L_2 stable from Δ to output $\boldsymbol{y}$,

$$u^{H\infty}(x) = \arg\min_{u \in K_V^{H\infty}(x)} \{\|u - \xi(x)\|\} \tag{33}$$

where

$$K_V^{H_\infty}(x) = \{u \in U(x) : V_x[f(x) + g(x)u] + \frac{1}{2\gamma^2}V_x l(x) l^T(x) V_x^T + \frac{1}{2}h^T(x)h(x) \le -\sigma(x)\} \tag{34}$$

Proof of Proposition III can be easily done based on the definition of finite gain L_2 stability and H_∞CLF. The analytical form of controller (33) can also be obtained as steps in section 3. Here only the analytical form of controller without input constraints is given,

$$u^{H\infty}(x) = \begin{cases} \xi - \dfrac{[V_x(f + \frac{1}{2\gamma^2}ll^TV_x^T + g\xi) + \frac{1}{2}h^Th + \sigma]g^TV_x^T}{V_x g g^T V_x^T} & \lambda > 0 \\ \xi & \lambda \le 0 \end{cases} \tag{35}$$

where

$$\lambda = V_x f + \sigma + V_x g\xi; \quad f = f(x); \quad g = g(x); \quad \xi = \xi(x);$$
$$\sigma = \sigma(x); \quad V_x = V_x(x); \quad h = h(x); \quad l = l(x)$$

It is not difficult to show that H_∞GPMN satisfies inequality (24) of Theorem I, thus, it can be used as $u_1(z)$ in controller (25) to bring the advantages of H_∞GPMN controller to the robust controller in section 4.1.

4.3 H_∞GPMN-ENMPC

As far as the external disturbances are concerned, nominal model based NMPC, where the prediction is made through a nominal certain system model, is an often used strategy in reality. And the formulation of it is very similar to non-robust NMPC, so dose the GPMN-ENMPC.

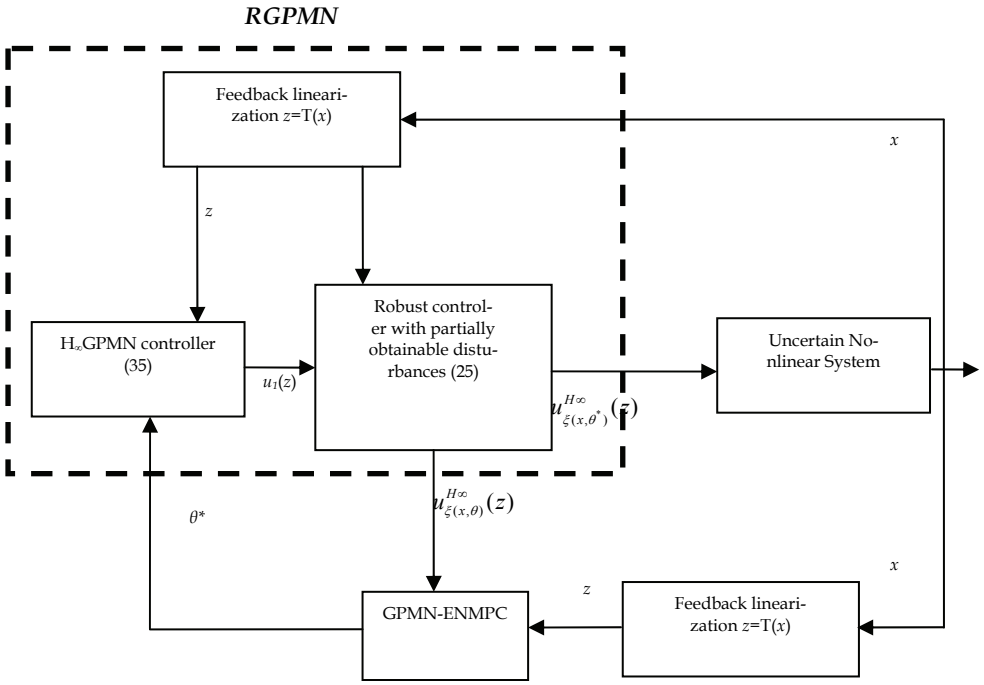


Fig. 4. Structure of new designed RNRHC controller

However, for disturbed nonlinear system like Eq. (23), GPMN-ENMPC algorithm can hardly be used in real applications due to weak robustness. Thus, in this subsection, we will combine it to the robust controller from sub-section 4.1 and sub-section 4.2 to overcome the drawbacks originated from both GPMN-ENMPC algorithm and the robust controller (25) and (35). The structure of the new parameterized H_∞GPMN-ENMPC algorithm based on controller (25) and (35) is as Fig. 4.

Eq. (36) is the new designed H_∞GPMN-ENMPC algorithm. Compared to Eq. (14), it is easy to find out that the control input in the H_∞GPMN-ENMPC algorithm has a pre-defined structure given in section 4.1 and 4.2.

$$
\begin{aligned}
& u^* = u^{H\infty}(x,\theta^*) \\
& \theta^* \overset{\Delta}{=} \arg\min_{u\in U} J(x,u) \\
& J(x,u) \overset{\Delta}{=} \int_t^{t+T} l(x(\tau),u(\tau))d\tau \\
& s.t. \quad \dot{x} = f(x) + g(x)u \\
& \qquad u(t) = u^{H\infty}(x,\theta)
\end{aligned}
\tag{36}
$$

5. Practical Considering

Both GPMN-ENMPC algorithm and H_∞GPMN-ENMPC algorithm can be divided into two processes, including the implementation process and the optimization process as Fig.5.

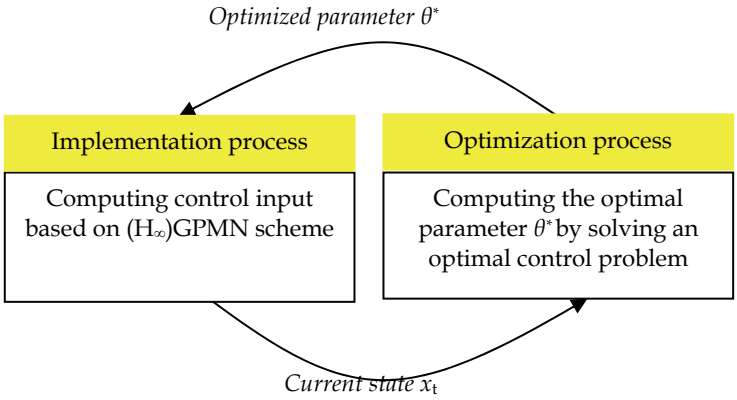


Fig. 5. The process of (H_∞)GPMN-ENMPC

The implementation process and the optimization process in Fig. 5 are independent. In implementation process, the (H_∞)GPMN scheme is used to ensure the closed loop (L_2) stability, and in the optimization process, the optimization algorithm is responsible to improving the optimality of the controller. And the interaction of the two processes is realized through the optimized parameter θ^* (from optimization process to implementation process) and the measured states (form implementation process to optimization process).

5.1 Time Interval Between Two Neighboring Optimizing Process

Sample time in controller implemented using computer is often very short, especially in mechatronic system. This is very challenging to implement complicated algorithm, such as GPMN-ENMPC in this chapter. Fortunately, the optimization process of the new designed controller will end up with a group of parameters which are used to form a stable (H_∞)GPMN controller, and the optimization process itself does not influence the closed loop stability at all. Thus, theoretically, any group of optimized parameters can be used for several sample intervals without destroying the closed loop stability.

Fig.6 denotes the scheduling of (H_∞)GPMN-ENMPC algorithm. In Fig.6, t is the current time instant; T is the prediction horizon; T_S is the sample time of the (H_∞)GPMN controller; and T_I is the duration of every optimal parameter $\theta^*(t)$, i.e., the same parameter θ^* is used to implement the (H_∞)GPMN controller from time t to time $t+T_I$.

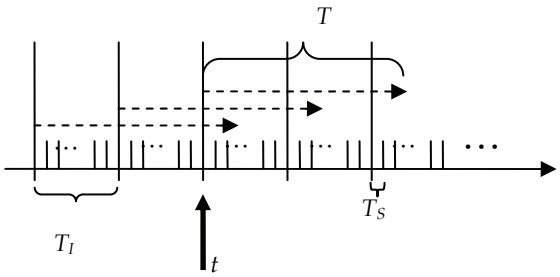


Fig. 6. Scheduling of ERNRHC

5.2 Numerical Integrator

How to predict the future's behavior is very important during the implementation of any kind of MPC algorithms. In most applications, the NMPC algorithm is realized by computers. Thus, for the continuous systems, it will be difficult and time consuming if some accurate but complicated numerical integration methods are used, such as Newton-Cotes integration and Gaussian quadratures, etc. In this chapter, we will discretize the continuous system (1) as follows (take system (1) as an example),

$$x(kT_O + T_O) = f(x(kT_O))T_O + g(x(kT_O))u(kT_O)T_O \tag{37}$$

where T_o is the discrete sample time. Thus, the numerical integrator can be approached by the operation of cumulative addition.

5.3 Index Function

Replace $x(kT_o)$ with $x(k)$, the index function can be designed as follows,

$$J(x(k_0), \theta_c) \overset{\Delta}{=} \theta_l^{T*} Z \theta_l^* + \sum_{i=k_0}^{k_0+N} J(x(i), \theta_c) \tag{38}$$

where k_0 denotes the current time instant; N is the predictive horizon with N=Int(T/T_o) (here Int(*) is the operator to obtain a integer nearest to *); θ_c is the parameter vector to be optimized at current time instant; and θ_l^* is the last optimization result; Q, Z, R are constant matrix with Q>0, Z>0, and R≥0.

The new designed item $\theta_l^{T*} Z \theta_l^*$ is used to reduce the difference between two neighboring optimized parameter vector, and improve the smoothness of the optimized control inputs u.

6. Numerical Examples

6.1 Example1 (GPMN-ENMPC without control input constrains)

Consider the following pendulum equation (Costa & do Va, 2003),

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \dfrac{19.6\sin x_1 - 0.2x_2^2 \sin 2x_1}{4/3 - 0.2\cos^2 x_1} + \dfrac{-0.2\cos x_1}{4/3 - 0.2\cos^2 x_1} u \end{cases} \tag{39}$$

A local CLF of system (39) can be given as,

$$V(x) = x^T P x = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 151.57 & 42.36 \\ 42.36 & 12.96 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{40}$$

Select

$$\sigma(x) = 0.1(x_1^2 + x_2^2) \tag{41}$$

The normal PMN control can be designed according to (5) as,

$$u = \begin{cases} -\dfrac{\rho(x)(4/3 - 0.2\cos^2 x_1)}{0.4\cos x_1 (42.36x_1 + 12.96x_2)} & \rho(x) < 0 \\ 0 & \rho(x) \le 0 \end{cases}$$

$$\rho(x) = 2[(151.57x_1 + 42.36x_2)x_2 + (42.36x_1 + 12.96x_2)\frac{19.6\sin x_1 - 0.2x_2^2 \sin 2x_1}{4/3 - 0.2\cos^2 x_1}] + \tag{42}$$

$$(10.54x_2 + 1.27x_1)^2 + x_2^2$$

Given initial state $x_0 = [x_1, x_2]^T = [-1, 2]^T$, and desired state $x_d = [0, 0]^T$, time response of the closed loop for PMN controller is shown in Fig. 7 in solid line. It can be seen that the closed loop with PMN controller (42) has a very low convergence rate for state x_1. This is mainly because the only regulable parameter to change the closed loop performance is $\sigma(x)$, which is difficult to be properly selected due to its great influence on the stability region.

To design GPMN-ENMPC, two different guide functions are selected based on Eq. (21),

$$\xi(x, \theta) = \theta_{0,0}(1 - x_1 - x_2) + \theta_{1,0}x_1 + \theta_{0,1}x_2 \tag{43}$$

$$\xi(x, \theta) = \theta_{0,0}(1 - x_1 - x_2)^2 + 2(\theta_{0,1}x_2 + \theta_{1,0}x_1)(1 - x_1 - x_2) + 2\theta_{1,1}x_1x_2 + \theta_{0,2}x_2^2 + \theta_{2,0}x_1^2 \tag{44}$$

CLF $V(x)$ and $\sigma(x)$ are given in Eq. (40) and Eq. (41), and others conditions in GPMN-ENMPC are designed as follows,

$$J = \int_0^T (x^T \begin{bmatrix} 20 & 0 \\ 0 & 1 \end{bmatrix} x + 0.01u^2) dt \tag{45}$$

$$l(x,u)=x^T\begin{bmatrix}20 & 0\\ 0 & 1\end{bmatrix}x+0.01u^2;\ f(x)=\begin{bmatrix}x_2\\ \dfrac{19.6\sin x_1-0.2x_2^2\sin 2x_1}{4/3-0.2\cos^2 x_1}\end{bmatrix};$$
$$g(x)=\begin{bmatrix}0\\ \dfrac{-0.2\cos x_1}{4/3-0.2\cos^2 x_1}\end{bmatrix};z=0.1I \tag{46}$$

Integral time interval T_o in Eq. (37) is 0.1s. Genetic algorithm (GA) in MATLAB toolbox is used to solve the online optimization problem. Time response of GPMN-ENMPC algorithm with different predictive horizon T and approaching order are presented in Fig. 7, where the dotted line denotes the case of T = 0.6s with guide function (43), and the dashed line is the case of T = 1.5s with guide function (44). From Fig. 7, it can be seen that the convergence performance of the proposed NMPC algorithm is better than PMN controller, and both the prediction horizon and the guide function will result in the change of the closed loop performance.

The improvement of the optimality is the main advantage of MPC compared with others controller. In view of this, we propose to estimate the optimality by the following index function,

$$J=\lim_{\Gamma\to\infty}\int_0^{\Gamma}(x^T\begin{bmatrix}20 & 0\\ 0 & 1\end{bmatrix}x+0.01u^2)dt \tag{47}$$

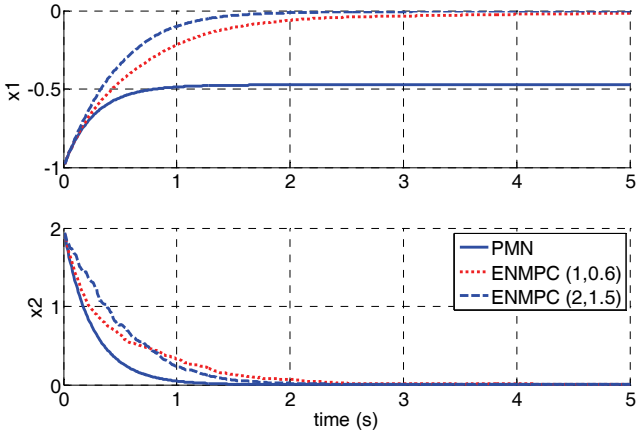


Fig. 7. Time response of different controller, where the (a,b) indicates that the order of $\xi(x,\theta)$ is a, and the predictive horizon b

The comparison results are summarized in Table 1, from which the following conclusions can be obtained, 1) GPMN-ENMPC has better optimizing performance than PMN controller in terms of optimization. 2) In most cases, GPMN-ENMPC with higher order $\xi(x,\theta)$ will usually result in a smaller cost than that with lower order $\xi(x,\theta)$. This is mainly because

higher order $\xi(x,\theta)$ indicates larger inherent optimizing parameter space. 3) A longer prediction horizon will usually be followed by a better optimal performance.

J	ENMPC				PMN	
	x_0 = (-1,2)		x_0 = (0.5,1)		x_0 = (-1,2)	x_0 = (0.5,1)
	k = 1	k = 2	K = 1	k = 2	----	
T=0.6	*29.39*	*28.87*	*6.54*	*6.26*	→+∞	→+∞
T=0.8	*23.97*	*23.83*	*5.02*	*4.96*	→+∞	→+∞
T=1.0	*24.08*	*24.07*	*4.96*	*4.90*	→+∞	→+∞
T=1.5	*26.31*	*24.79*	*5.11*	*5.28*	→+∞	→+∞

Table 1. the cost value of different controller
* k is the order of Bernstein polynomial used to approach the optimal value function; T is the predictive horizon; x_0 is the initial state

Another advantage of the GPMN-ENMPC algorithm is the flexibility of the trade offs between the optimality and the computational time. The computational time is influenced by the dimension of optimizing parameters and the parameters of the optimizing algorithm, such as the maximum number of iterations and the size of the population (the smaller these values are selected, the less the computational cost is). However, it will be natural that the optimality maybe deteriorated to some extent with the decreasing of the computational burden. In preceding paragraphs, we have researched the optimality of GPMN-ENMPC algorithm with different optimizing parameters, and now the optimality comparisons among the closed loop systems with different GA parameters will be done. And the results are listed in Table 2, from which the certain of the optimality loss with the changing of the optimizing algorithm's parameters can be observed. This can be used as the criterion to determine the trade-off between the closed loop performance and the computational efficiency of the algorithm.

OP	G=100 PS=50	G=50 PS=50	G=50 PS=30	G=50 PS=20	G=50 PS=10
cost	26.2	28.1	30.8	43.5	45.7

Table 2. The relation between the computational cost and the optimality
*x_0 = (-1,2), T=1.5, k = 1, OP means Optimization Prameters, G means Generations, PS means Population Size

Finally, in order to verify that the new designed algorithm is improved in the computational burden, simulations comparing the performance of the new designed algorithm and algorithm in (Primbs, 1999) are conducted with the same optimizing algorithm. Time interval of two neighboured optimization (T_I in Table 3) in Primbs' algorithm is important since control input is assumed to be constant at every time slice. Generally, large time interval will result in poor stability.
While our new GPMN-ENMPC results in a group of controller parameter, and the closed loop stability is independent of T_I. Thus different T_I is considered in these simulations of Primbs'

algorithm and Table 3 lists the results. From Table 3, the following items can be concluded: 1) with same GA parameters, Primbs' algorithm is more time-consuming and poorer in optimality than GPMN-ENMPC. This is easy to be obtained through comparing results of Ex-2 and Ex-5; 2) in order to obtain similar optimality, GPMN-ENMPC takes much less time than Primbs' algorithm. This can be obtained by comparing results of Ex-1/Ex-4 and Ex-6, as well as Ex-3 and Ex-5. The reasons for these phenomena have been introduced in ***Remark 3***.

	Algorithm in (Primbs, 1999)				GPMN-ENMPC	
	Ex-1	Ex-2	Ex-3	Ex-4	Ex-5	Ex-6
TI	0.1		0.05		0.1	
OP	G=100 PS=50	G=50 PS=50	G=100 PS=50	G=50 PS=50	G=50 PS=50	G=50 PS=30
Average Time Consumption	2.2075	1.8027	2.9910	2.2463	1.3961	0.8557
Cost	31.2896	35.7534	27.7303	31.8055	28.1	31.1043

Table 3. Performance comparison of GPMN-ENMPC and Primbs' algorithm
*x_0 = (-1,2), TI means time interval of two neighbored optimization; OP means Optimization Prameters; G means Generations, PS means Population Size. Other parameters of GPMN-ENMPC are T=1.5, k = 1

6.2 Example 2 (GPMN-ENMPC with control input constraint)

In order to show the performance of the GPMN-ENMPC in handling input constraints, we give another simulation using the dynamics of a mobile robot with orthogonal wheel assemblies (Song, 2007). The dynamics can be denoted as Eq. (48),

$$\dot{x} = f(x) + g(x)u \tag{48}$$

where

$$f(x) = \begin{bmatrix} x_2 \\ -2.3684x_4x_6 - 0.5921x_2 \\ x_4 \\ 2.3684x_2x_6 - 0.5921x_4 \\ x_6 \\ -0.2602x_6 \end{bmatrix}$$

$$g(x) = \begin{bmatrix} 0 & 0 & 0 \\ 0.8772(\sqrt{3}\sin x_5 - \cos x_5) & 0.8772*2\cos x_5 & 0.8772(\sqrt{3}\sin x_5 - \cos x_5) \\ 0 & 0 & 0 \\ 0.8772(\sqrt{3}\cos x_5 - \sin x_5) & 0.8772*2\sin x_5 & 0.8772(-\sqrt{3}\cos x_5 - \sin x_5) \\ 0 & 0 & 0 \\ -1.4113 & -1.4113 & -1.4113 \end{bmatrix}$$

$x_1 = x_w; x_2 = \dot{x}_w; x_3 = y_w; x_4 = \dot{y}_w; x_5 = \varphi_w; x_6 = \dot{\varphi}_w$; x_w, y_w, φ_w are respective the x-y positions and yaw angle; u_1, u_2, u_3 are motor torques.
Suppose that control input is limited in the following closed set,

$$U = \{(u_1, u_2, u_3) \mid (u_1^2 + u_2^2 + u_3^2)^{1/2} \leq 20\} \tag{49}$$

System (48) is feedback linearizable, and by which we can obtain a CLF of system (48) as follows,

$$V(x) = x^T P x \tag{50}$$

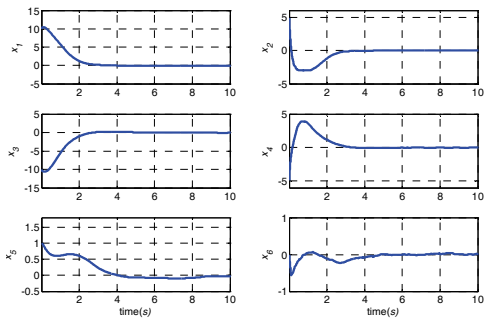
where

$$P = \begin{bmatrix} 1.125 & 0.125 & 0 & 0 & 0 & 0 \\ 0.125 & 0.156 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.125 & 0.125 & 0 & 0 \\ 0 & 0 & 0.125 & 0.156 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.125 & 0.125 \\ 0 & 0 & 0 & 0 & 0.125 & 1.156 \end{bmatrix}$$

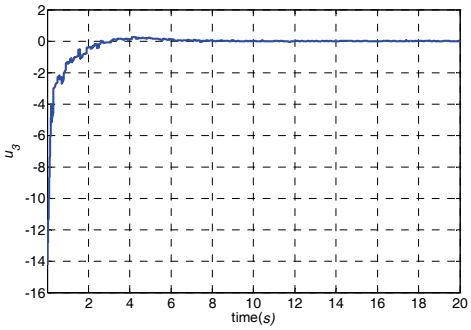
The cost function $J(x)$ and $\sigma(x)$ are designed as,

$$\begin{aligned} J(x) &= \int_{t_0}^{t_0+T} (3x_1^2 + 3x_3^2 + 3x_5^2 + x_2^2 + x_4^2 + x_6^2 + 5u_1^2 + 5u_2^2 + 5u_3^2)dt + \theta^T(k-1)Z\theta(k-1); \\ \sigma(x) &= 0.1(x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2); Z = 0.1I \end{aligned} \tag{51}$$

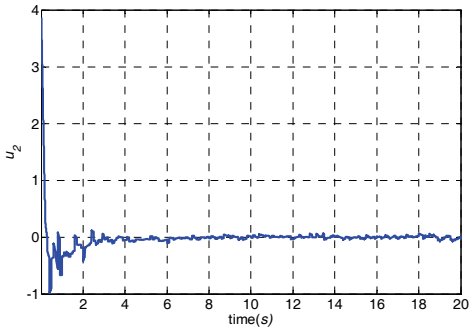
System (48) has 6 states and 3 inputs, which will introduce large computational burden if using the GPMN-ENMPC method. Fortunately, one of the advantages of GPMN-ENMPC is that the optimization does not destroy the closed loop stability. Thus, in order to reduce the computation burden, we reduce the frequency of the optimization in this simulation, i.e., one optimization process is conducted every 0.1s while the controller of (13) is calculated every 0.002s, i.e., T_I = 0.1s, T_s = 0.002s.



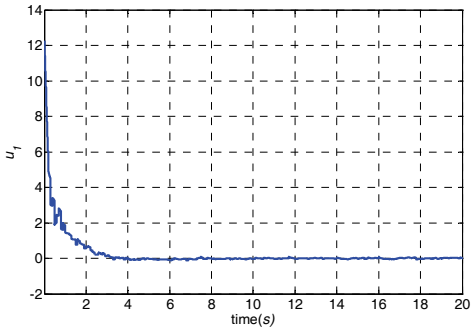
a) states response



b) control input u_1



c) control input u_2



d) control input u_3

Fig. 8. GPMN-ENMPC controller simulation results on the mobile robot with input constraints

Initial States (x_1; x_2; x_3; x_4; x_5; x_6)	Feedback linearization controller	GPMN-NMPC
(10; 5; 10; 5; 1; 0)	2661.7	1377.0
(10; 5; 10; 5; -1; 0)	3619.5	1345.5
(-10; -5; 10; 5; 1; 0)	2784.9	1388.5
(-10; -5; 10; 5; -1; 0)	8429.2	1412.0
(-10; -5; -10; -5; 1; 0)	394970.0	1349.9
(-10; -5; -10; -5; -1; 0)	4181.6	1370.9
(10; 5; -10; -5; 1; 0)	3322	1406
(10; 5; -10; -5; -1; 0)	1574500000	1452.1
(-5; -2; -10; -5; 1; 0)	1411.2	856.1
(-10; -5; -5; -2; 1; 0)	1547.5	850.9

Table 4. The comparison of the optimality

Simulation results are shown in Fig.8 with the initial state (10; 5; -10; -5; 1; 0), From Fig.8, it is clear that GPMN-ENMPC controller has the ability to handling input constraints.
In order to evaluate the optimal performance of the GPMN-ENMPC, we proposed the following cost function according to Eq. (51),

$$\cos t = \lim_{\Gamma \to +\infty} \int_0^{\Gamma} (3x_1^2 + 3x_3^2 + 3x_5^2 + x_2^2 + x_4^2 + x_6^2 + 5u_1^2 + 5u_2^2 + 5u_3^2)dt \tag{52}$$

Table 4 lists the costs by feedback linearization controller and GPMN-ENMPC for several different initial states, from which it can be seen that the cost of GPMN-ENMPC is less than the half of the cost of feedback linearization controller when the initial is (10; 5; -10; -5; 1; 0). And in most cases listed in Table 4, the cost of GPMN-ENMPC is about one second of that of feedback linearization controller. Actually, in some special cases, such as the initial of (10; 5; -10; -5; -1; 0), the cost ratio of feedback linearization controller to GPMN-ENMPC is more than 1000000.

6.3 Example 3 (H_∞GPMN-ENMPC)

In this section, a simulation will be given to verify the feasibility of the proposed H_∞GPMN-ENMPC algorithm with respect to the following planar dynamic model of helicopter,

$$\begin{cases} \ddot{x} = -9.8\cos\phi\sin\theta + \Delta_1 \\ \ddot{y} = 9.8\sin\phi + \Delta_2 \\ \ddot{\phi} = 0.05\dot{\theta}^2\sin\phi\cos\phi + \dot{\phi}\dot{\theta}\tan\theta(0.5+0.05\cos^2\phi) + \\ \qquad 0.07\cos\phi\tan\theta + L + M\sin\phi\tan\theta + \Delta_3 \\ \ddot{\theta} = -0.05\dot{\phi}\dot{\theta}\sin\phi\cos\phi - 0.07\sin\phi + M\cos\phi + \Delta_4 \end{cases} \tag{53}$$

where Δ_1, Δ_2, Δ_3, Δ_4 are all the external disturbances, and are selected as following values,

$$\begin{cases} \Delta_1 = 3; \quad \Delta_2 = 3 \\ \Delta_3 = 10\sin(0.5t) \\ \Delta_4 = 10\sin(0.5t) \end{cases}$$

Firstly, design an H_∞CLF of system (53) by using the feedback linearization method,

$$V = X^T \overline{P} X \tag{54}$$

where,

$$X = [x, \dot{x}, \ddot{x}, \dddot{x}, y, \dot{y}, \ddot{y}, \dddot{y}]^T$$

$$\overline{P} = \begin{bmatrix} 14.48 & 11.45 & 3.99 & 0.74 & 0 & 0 & 0 & 0 \\ 11.45 & 9.77 & 3.44 & 0.66 & 0 & 0 & 0 & 0 \\ 3.99 & 3.44 & 1.28 & 0.24 & 0 & 0 & 0 & 0 \\ 0.74 & 0.66 & 0.24 & 0.05 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 14.48 & 11.45 & 3.99 & 0.74 \\ 0 & 0 & 0 & 0 & 11.45 & 9.77 & 3.44 & 0.66 \\ 0 & 0 & 0 & 0 & 3.99 & 3.44 & 1.28 & 0.24 \\ 0 & 0 & 0 & 0 & 0.74 & 0.66 & 0.24 & 0.05 \end{bmatrix}$$

Thus, the robust predictive controller can be designed as Eq. (25), (35) and (36) with the following parameters,

$$\sigma(x) = X^T X$$

$$J = \theta_l^{T*} I \theta_l^* + \sum_{i=1}^{N} [x^{\mathrm{T}}(iT_o) P x(iT_o) + u^{\mathrm{T}}(iT_o) Q u(iT_o)] T_o$$

$$\xi(x,\beta) = \begin{bmatrix} \beta_1 + \beta_2 x + \beta_3 \dot{x} + \beta_4 y + \beta_5 \dot{y} + \beta_6 \phi + \beta_7 \dot{\phi} + \beta_8 \theta + \beta_9 \dot{\theta} \\ \beta_{10} + \beta_{11} x + \beta_{12} \dot{x} + \beta_{13} y + \beta_{14} \dot{y} + \beta_{15} \phi + \beta_{16} \dot{\phi} + \beta_{17} \theta + \beta_{18} \dot{\theta} \end{bmatrix}$$

$$P = \begin{bmatrix} 50000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 50000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; T_o = 0.1s; T_s = 0.02s; T_l = 1s; N = 20; Z = I$$

Time response of the H_∞GPMN-ENMPC is as solid line of Fig.9 and Fig.10. Furthermore, the comparisons between the performance of the closed loop controlled by the proposed H_∞GPMN-ENMPC and some other controller design method are done. The dashed line in Fig.9 and Fig.10 is the time response of the feedback linearization controller. From Fig.9 and Fig.10, the disturbance attenuation performance of the H_∞GPMN-ENMPC is apparently better than that of feedback linearization controller, because the penalty gain of position signals, being much larger than other terms, can be used to further improve the ability.

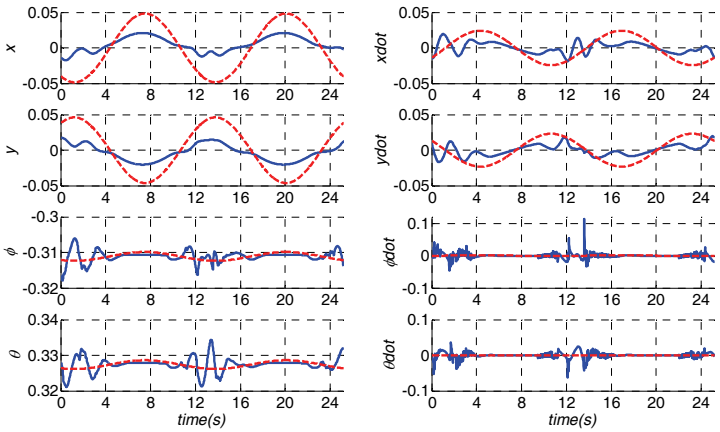


Fig. 9. Time response of states

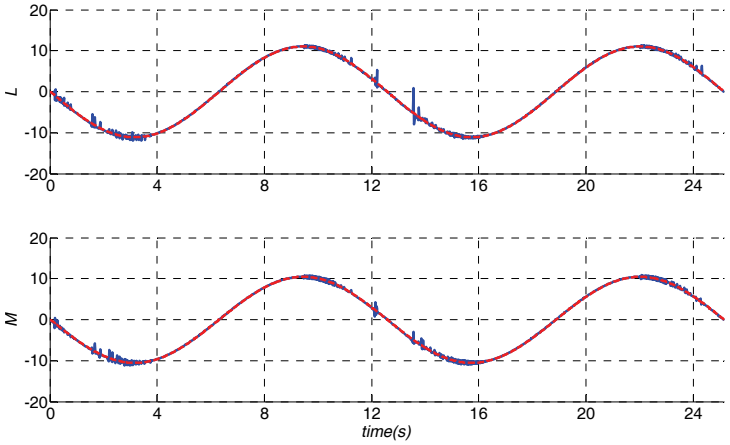


Fig. 10. Control inputs

Simultaneously, the following index is used to compare the optimality of the two different controllers,

$$J = \lim_{\Gamma \to \infty} \int_0^{0+\Gamma} [x^{\mathrm{T}}(t)Px(t) + u^{\mathrm{T}}(t)Qu(t)]dt \tag{55}$$

The optimality performance of H_∞GPMN-ENMPC, computed from Eq. (55), is about 3280, and the feedback linearization controller is about 5741, i.e., the H_∞GPMN-ENMPC has better optimality than the feedback linearization controller.

7. Conclusion

In this paper, nonlinear model predictive control (NMPC) is researched and a new NMPC algorithm is proposed. The new designed NMPC algorithm, called GPMN-enhancement NMPC (GPMN-ENMPC), has the following three advantages: 1) closed loop stability can be always guaranteed; 2) performance other than optimality and stability can be considered in the new algorithm through selecting proper guide function; 3) computational cost of the new NMPC algorithm is regulable according to the performance requirement and available CPU capabilities. Also, the new GPMN-ENMPC is generalized to a robust version with respect to input-output feedback linearizable nonlinear system with partially known uncertainties. Finally, extensive simulations have been conducted, and the results have shown the feasibility and validity of the new designed method.

8. References

Brinkhuis, J. & Tikhomirov, V., *Optimization : insights and applications*, Princeton University Press, ISBN : 978-0-691-10287-0, Oxfordshire, United Kingdom

Chen, C. & Shaw, L., On receding horizon feedback control, *Automatica*, Vol. 18, No. 3, May, 1982, 349-352, ISSN : 0005-1098

Chen, H. & Allgower, F., A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, Vol. 34, No. 10, Oct, 1998, 1205-1217, ISSN : 0005-1098

Chen, W., Disturbance observer based control for nonlinear systems, *IEEE/ASME Transactions on mechatronics*, Vol. 9, No. 4, 2004, 706-710, ISSN : 1083-4435

Costa, E. & do Val, J., Stability of receding horizon control of nonlinear systems, *Proceedings of The 42nd IEEE Conference on Decision and Control*, pp. 2077-2801, ISSN : 0191-2216, HI, USA, Dec, 2003, IEEE, Maui

Freeman, R. & Kokotovic, P, Inverse optimality in robust stabilization, *SIAM Journal on Control and Optimization*, Vol. 34, No. 4, Aug, 1996a, 1365-1391, ISSN : 0363-0129

Freeman, R. & Kokotovic, P.(1996b), *Robust nonlinear control design: state-space and Lyapunov techniques*, Birkhauser, ISBN: 978-0-8176-4758-2, Boston

Henson, M., Nonlinear model predictive control: current status and future directions. *Computers & Chemical Engineering*, Vol. 23, No. 2, Dec, 1998, 187-202, ISSN : 0098-1354

He, Y., and Han, J., Acceleration feedback enhanced H_∞ disturbance attenuation control, *Proceedings of The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pp. 839-844, ISSN : 1553-572X, Taiwan, Nov, 2007, IEEE, Taipei

Khalil, H. (2002), *Nonlinear systems*, 3rd edition, Printice Hall, ISBN: 0-13-067389-7, NJ, USA

Lewis, F. & Syrmos, V. (1995), *Optimal control*, John Wiley & Sons, ISBN : 0-471-03378-2, Bangalore, India

Magni, L., De Nicolao, G., Magnani, L. & Scattolini, R., A stabilizing model-based predictive control algorithm for nonlinear systems. *Automatica*, Vol. 37, No. 9, Sep, 2001, 1351-1362, ISSN : 0098-1354

Mayne, D., Rawlings, J., Rao, C. & Scokaert, P., Constrained model predictive control: stability and optimality. *Automatica*, Vol. 36, No. 6, Jun, 2000, 789-814, ISSN : 0005-1098

Pothin, R., Disturbance decoupling for a class of nonlinear MIMO systems by static measurement feedback, *Systems & Control Letters*, Vol. 43, No. 2, Jun, 2001, 111-116, ISSN : 0167-6911

Primbs, J. A., Nevistic, V. & Doyle, J. C., Nonlinear optimal control: a control Lyapunov function and receding horizon perspective, *Asian Journal of Control*, Vol. 1, No. 1, Jan, 1999, 14-24, ISSN: 1561-8625

Primbs, J. & Nevistic, V., Feasibility and stability of constrained finite receding horizon control, *Automatica*, Vol. 36, No. 7, Jul, 2000, 965-971, ISSN : 0005-1098

Qin, S., & Badgwell, T., A survey of industrial model predictive control technology. *Control Engineering Practice*, Vol. 11, No. 7, Jul, 2003, 733-764, ISSN :0967-0661

Rawlings, J., Tutorial overview of model predictive control, *IEEE Control System Magazine*, Vol. 20, No. 3, Jun, 2000, 38-52, ISSN : 0272-1708

Scokaert, P., Mayne, D. & Rawlings, J., Suboptimal model predictive control (feasibility implies stability), *IEEE Transactions on Automatica Control*, Vol. 44, No. 3, Mar, 1999, 648-654, ISSN : 0018-9286

Song, Q., Jiang, Z. & Han, J., Noise covariance identification-based adaptive UKF with application to mobile robot system, *Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2007)*, pp. 4164-4169, ISSN: 1050-4729, Italy, May, 2007, Roma

Sontag, E., A 'universal' construction of Artstein's theorem on nonlinear stabilization, *Systems & Control Letters*, Vol. 13, No. 2, Aug, 1989, 117-123, ISSN : 0167-6911

Zou, T., Li, S. & Ding, B., A dual-mode nonlinear model predictive control with the enlarged terminal constraint sets, *Acta Automatica Sinica*, Vol. 32, No. 1, Jan, 2006, 21-27, ISSN : 0254-4156

Wesselowske, K. & Fierro, R., A dual-mode model predictive controller for robot formations, Proceedings of the 42nd IEEE Conference on Decision and Control, pp. 3615-3620, ISSN : 0191-2216, HI, USA, Dec, 2003, IEEE, Maui

Robust Model Predictive Control Algorithms for Nonlinear Systems: an Input-to-State Stability Approach

D. M. Raimondo
Automatic Control Laboratory, Electrical Engineering, ETH Zurich, Physikstrasse 3, 8092 Zurich Switzerland

D. Limon, T. Alamo
Departamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla, Escuela Superior de Ingenieros, Camino de los Descubrimientos s/n 41092 Sevilla Spain

L. Magni
Dipartimento di Informatica e Sistemistica, Università di Pavia, via Ferrata 1, 27100 Pavia Italy

This paper presents and compares two robust *MPC* controllers for constrained nonlinear systems based on the minimization of a nominal performance index. Under suitable modifications of the constraints of the Finite Horizon Optimization Control Problems (*FHOCP*), the derived controllers ensure that the closed loop system is Input-to-State Stable (*ISS*) with a robust invariant region, with relation to additive uncertainty/disturbance. Assuming smoothness of the model function and of the ingredients of the *FHOCP*, the effect of each admissible disturbance in the predictions is considered and taken into account by the inclusion in the problem formulation of tighter state and terminal constraints. A simulation example shows the potentiality of both the algorithms and highlights their complementary aspects.

Keywords: Robust *MPC*, Input to State Stability, Constraints, Robust design.

1. Introduction

Model predictive control (*MPC*) is an optimal control technique which deals with constraints on the states and the inputs. This strategy is based on the solution of a finite horizon optimization problem (*FHOCP*), which can be posed as a mathematical programming problem. The control law is obtained by means of the receding horizon strategy that requires the solution of the optimization problem at each sample time Camacho & Bordons (2004); Magni et al. (2009); Rawlings & Mayne (2009).
It is well known that considering a terminal cost and a terminal constraint in the optimization problem, the *MPC* stabilizes asymptotically a constrained system in absence of disturbances or uncertainties. If there exist uncertainties in the process model, then the stabilizing properties may be lost Magni & Scattolini (2007); Mayne et al. (2000) and these must be taken into account in the controller design. Recent results have revealed that nominal *MPC* may have

zero robustness, i.e. stability or feasibility may be lost if there exist model mismatches Grimm et al. (2004). Therefore it is quite important to analyze when this situation occurs and to find design procedures to guarantee certain degree of robustness. In Limon et al. (2002b); Scokaert et al. (1997) it has been proved that under some regularity condition on the optimal cost, the *MPC* is able to stabilize the uncertain system; however, this regularity condition may be not ensured due to constraints, for instance.

The synthesis of *NMPC* algorithms with robustness properties for uncertain systems has been developed by minimizing a nominal performance index while imposing the fulfillment of constraints for each admissible disturbance, see e.g. Limon et al. (2002a) or by solving a min-max optimization problem, see e.g. Chen et al. (1997); Fontes & Magni (2003); Magni et al. (2003); Magni, Nijmeijer & van der Schaft (2001); Magni & Scattolini (2005). The first solution calls for the inclusion in the problem formulation of tighter state, control and terminal constraints. The main advantage is that the on-line computational burden is substantially equal to the computational burden of the nominal *NMPC*. In fact, nominal prediction based robust predictive controllers can be thought as a nominal MPC designed in such a way that a certain degree of robustness is achieved. The main limitation is that it can lead to very conservative solutions. With a significant increase of the computational burden, less conservative results can be achieved by solving a min-max optimization problem.

Input-to-State Stability (*ISS*) is one of the most important tools to study the dependence of state trajectories of nonlinear continuous and discrete time systems on the magnitude of inputs, which can represent control variables or disturbances. The concept of *ISS* was first introduced in Sontag (1989) and then further exploited by many authors in view of its equivalent characterization in terms of robust stability, dissipativity and input-output stability, see e.g. Jiang & Wang (2001), Huang et al. (2005), Angeli et al. (2000), Jiang et al. (1994), Nešić & Laila (2002). Now, several variants of *ISS* equivalent to the original one have been developed and applied in different contexts (see e.g. Sontag & Wang (1996), Gao & Lin (2000), Sontag & Wang (1995), Huang et al. (2005)). The *ISS* property has been recently introduced also in the study of nonlinear perturbed discrete-time systems controlled with Model Predictive Control (*MPC*), see e.g. Limon et al. (2009), Raimondo et al. (2009), Limon et al. (2002a), Magni & Scattolini (2007), Limon et al. (2006), Franco et al. (2008), Magni et al. (2006). In fact, the development of *MPC* synthesis methods with enhanced robustness characteristics is motivated by the widespread success of MPC and by the availability of many MPC algorithms for nonlinear systems guaranteeing stability in nominal conditions and under state and control constraints.

In this paper two algorithms based on the solution of a minimization problem with respect to a nominal performance index are proposed. The first one, following the algorithm presented in Limon et al. (2002a), proves that if the terminal cost is a Lyapunov function which ensures a nominal convergence rate (and hence some degree of robustness), then the derived nominal *MPC* is an Input-to-State stabilizing controller. The size of allowable disturbances depends on the one step decreasing rate of the terminal cost.

The second algorithm, first proposed in a preliminary version in Raimondo & Magni (2006), shares with de Oliveira Kothare & Morari (2000) the idea to update the state of the nominal system with the value of the real one only each M step to check the terminal constraint. The use of a prediction horizon larger than a time varying control horizon is aimed to provide more robust results by means of considering the decreasing rate in a number of steps.

Both controllers are based on the Lipschitz continuity of the prediction model and of some of the ingredients of the *MPC* functional such as stage cost function and the terminal cost

function. Under the same assumptions they ensure that the closed loop system is Input-to-State-Stable (ISS) with relation to the additive uncertainty.
A simulation example shows the potentiality of both the algorithms and highlights their complementary aspects.
The paper is organized as follows: first some notations and definitions are presented. In Section 3 the problem is stated. In Section 4 the Regional Input-to-State Stability is introduced. In Section 5 the proposed MPC controllers are presented. In Section 6 the benefits of the proposed controllers are illustrated with several examples. Section 7 contains the conclusions. All the proofs are gathered in an Appendix in order to improve the readability.

2. Notations and basic definitions

Let $\mathcal{R}$, $\mathcal{R}_{\geq 0}$, $\mathcal{Z}$ and $\mathcal{Z}_{\geq 0}$ denote the real, the non-negative real, the integer and the non-negative integer numbers, respectively. For a given $M \in \mathcal{Z}_{\geq 0}$, the following set is defined $\mathcal{T}_M \triangleq \{kM, k \in \mathcal{Z}_{\geq 0}\}$. Euclidean norm is denoted as $|\cdot|$. Given a signal w, the signal's sequence is denoted by $\mathbf{w} \triangleq \{w(0), w(1), \cdots\}$ where the cardinality of the sequence is inferred from the context. The set of sequences of w, whose values belong to a compact set $\mathcal{W} \subseteq \mathcal{R}^m$ is denoted by $\mathcal{M}_{\mathcal{W}}$, while $\mathcal{W}^{sup} \triangleq \sup_{w \in \mathcal{W}}\{|w|\}$, $\mathcal{W}^{inf} \triangleq \inf_{w \in \mathcal{W}}\{|w|\}$. Moreover $\|\mathbf{w}\| \triangleq \sup_{k \geq 0}\{|w(k)|\}$ and $\|\mathbf{w}_{[\tau]}\| \triangleq \sup_{0 \leq k \leq \tau}\{|w(k)|\}$. The symbol id represents the identity function from $\mathcal{R}$ to $\mathcal{R}$, while $\gamma_1 \circ \gamma_2$ is the composition of two functions γ_1 and γ_2 from $\mathcal{R}$ to $\mathcal{R}$. Given a set $A \subseteq \mathcal{R}^n$, $|\zeta|_{\mathcal{A}} \triangleq \inf\{|\eta - \zeta|, \eta \in A\}$ is the point-to-set distance from $\zeta \in \mathcal{R}^n$ to A. The difference between two given sets $A \subseteq \mathcal{R}^n$ and $B \subseteq \mathcal{R}^n$ with $B \subseteq A$, is denoted by $A \backslash B \triangleq \{x : x \in A, x \notin B\}$. Given two sets $A \subseteq \mathcal{R}^n$ and $B \subseteq \mathcal{R}^n$, then the Pontryagin difference set C is defined as $C = A \sim B \triangleq \{x \in \mathcal{R}^n : x + \xi \in A, \forall \xi \in B\}$. Given a closed set $A \subseteq \mathcal{R}^n$, ∂A denotes the border of A. A function $\gamma : \mathcal{R}_{\geq 0} \to \mathcal{R}_{\geq 0}$ is of class $\mathcal{K}$ (or a "$\mathcal{K}$-function") if it is continuous, positive definite and strictly increasing. A function $\gamma : \mathcal{R}_{\geq 0} \to \mathcal{R}_{\geq 0}$ is of class $\mathcal{K}_\infty$ if it is a $\mathcal{K}$-function and $\gamma(s) \to +\infty$ as $s \to +\infty$. A function $\beta : \mathcal{R}_{\geq_0} \times \mathcal{Z}_{\geq_0} \to \mathcal{R}_{\geq_0}$ is of class $\mathcal{KL}$ if, for each fixed $t \geq 0$, $\beta(\cdot, t)$ is of class $\mathcal{K}$, for each fixed $s \geq 0$, $\beta(s, \cdot)$ is decreasing and $\beta(s, t) \to 0$ as $t \to \infty$.

3. Problem statement

In this paper it is assumed that the plant to be controlled is described by discrete-time nonlinear model:

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

where $x(k) \in \mathcal{R}^n$ is the state of the system, $u(k) \in \mathcal{R}^m$ is the control variable, and $w(k) \in \mathcal{R}^n$ is the additive uncertainty. Notice that the additive uncertainty can model perturbed systems and a wide class of model mismatches. Take into account that these ones might depend on the state and on the input of the system, consider a real plant $x_{k+1} = \tilde{f}(x(k), u(k))$. Then the additive uncertainty can be taken as $w(k) = [\tilde{f}(x(k), u(k)) - f(x(k), u(k))]$. Note that if, as it will be assumed, x and u are bounded and f is Lipschitz, then w can be modeled as a bounded uncertainty. This kind of model uncertainty has been used in previous papers about robustness in MPC , as in Michalska & Mayne (1993) and Mayne (2000).
In the following assumption, the considered structure of such a model is formally presented.

Assumption 1.

1. *The uncertainty belongs to a compact set $\mathcal{W} \subset \mathcal{R}^n$ containing the origin, defined as*

$$\mathcal{W} \triangleq \{w \in \mathcal{R}^n : |w| \leq \gamma\} \tag{2}$$

where $\gamma \in \mathcal{R}_{\geq 0}$.

2. *The system has an equilibrium point at the origin, that is $f(0,0) = 0$.*
3. *The control and state of the plant must fulfill the following constraints on the state and the input:*

$$x(k) \in X \tag{3}$$

$$u(k) \in U \tag{4}$$

where X is and U are compact sets, both of them containing the origin.

4. *The state of the plant $x(k)$ can be measured at each sample time.* □

The control objective consists in designing a control law $u = \kappa(x)$ such that it steers the system to (a neighborhood of) the origin fulfilling the constraints on the input and the state along the system evolution for any possible uncertainty and yielding an optimal closed performance according to certain performance index.

4. Regional Input-to-State Stability

In this section the *ISS* framework for discrete-time autonomous nonlinear systems is presented and Lyapunov-like sufficient conditions are provided. This will be employed in the paper to study the behavior of perturbed nonlinear systems in closed-loop with *MPC* controllers. Consider a nonlinear discrete-time system described by

$$x(k+1) = F(k, x(k), w(k)),\ k \geq t,\ x(t) = \bar{x} \tag{5}$$

where $F : \mathcal{Z}_{\geq 0} \times \mathcal{R}^n \times \mathcal{R}^r \to \mathcal{R}^n$ is locally Lipschitz continuous, $F(k,0,0) = 0$, $x(k) \in \mathcal{R}^n$ is the state, $w(k) \in \mathcal{R}^p$ is the input (disturbance), limited in a compact set $\mathcal{W}$ containing the origin $w(k) \in \mathcal{W}$. The solution to the difference equation (5) at time k, starting from state $x(0) = \bar{x}$ and for inputs $\mathbf{w}$ is denoted by $x(k, \bar{x}, \mathbf{w})$. Consider the following definitions.

Definition 1 (Robust positively invariant set)**.** *A set $\Xi(k) \subseteq \mathcal{R}^n$ is a robust positively invariant set for the system (5), if $x(k, \bar{x}, \mathbf{w}) \in \Xi(k)$, $\forall k \geq t$, $\forall \bar{x} \in \Xi(t)$ and $\forall \mathbf{w} \in \mathcal{M}_{\mathcal{W}}$.* □

Definition 2 (Magni et al. (2006) Regional *ISS* in $\Xi(k)$)**.** *Given a compact set $\Xi(k) \subset \mathcal{R}^n$ containing the origin as an interior point, the system (5) with $\mathbf{w} \in \mathcal{M}_{\mathcal{W}}$, is said to be ISS (Input-to-State Stable) in $\Xi(k)$, if $\Xi(k)$ is robust positively invariant for (5) and if there exist a $\mathcal{KL}$-function β and a $\mathcal{K}$-function γ such that*

$$|x(k, \bar{x}, \mathbf{w})| \leq \beta(|\bar{x}|, k) + \gamma(\|\mathbf{w}_{[k-1]}\|),\ \forall k \geq t,\ \forall \bar{x} \in \Xi(t). \tag{6}$$

□

Definition 3 (Magni et al. (2006) ISS-Lyapunov function in Ξ)**.** *A function V: $\mathcal{R}^n \to \mathcal{R}_{\geq}0$ is called an ISS-Lyapunov function in $\Xi(k) \subset \mathcal{R}^n$ for system (5) with respect to w, if:*

1) $\Xi(k)$ is a closed robust positively invariant set containing the origin as an interior point.

2) *there exist a compact set* $\Omega \subseteq \Xi(k)$, $\forall k \geq t$ (containing *the origin as an interior point), a pair of suitable* $\mathcal{K}_\infty$*-functions* α_1, α_2 *such that:*

$$V(x) \geq \alpha_1(|x|), \forall x \in \Xi(k), \forall k \geq t \tag{7}$$

$$V(x) \leq \alpha_2(|x|), \forall x \in \Omega \tag{8}$$

3) *there exist a suitable* $\mathcal{K}_\infty$*-function* α_3*, a* $\mathcal{K}$*-function* σ *such that:*

$$\begin{array}{rcl} \Delta V(x) & \triangleq & V(F(k,x,w)) - V(x) \\ & \leq & -\alpha_3(|x|) + \sigma(|w|), \forall x \in \Xi(k), \forall k \geq t, \ \forall w \in \mathcal{W} \end{array} \tag{9}$$

4) *there exist a suitable* $\mathcal{K}_\infty$*-functions* ρ *(with* ρ *such that* $(id - \rho)$ *is a* $\mathcal{K}_\infty$*-function) and a suitable constant* $c_\theta > 0$*, such that there exists a nonempty compact set* $\Theta \subset \{x : x \in \Omega, d(x, \delta\Omega) > c_\theta\}$ *(containing the origin as an interior point) defined as follows:*

$$\Theta \quad \triangleq \quad \{x : V(x) \leq b(\mathcal{W}^{sup})\} \tag{10}$$

where $b \triangleq \alpha_4^{-1} \circ \rho^{-1} \circ \sigma$*, with* $\alpha_4 \triangleq \alpha_3 \circ \alpha_2^{-1}$.

□

The following sufficient condition for regional *ISS* of system (5) can be stated.

Theorem 1. *If system (5) admits an ISS-Lyapunov function in* $\Xi(k)$ *with respect to w, then it is ISS in* $\Xi(k)$ *with respect to w and* $\lim_{k\to\infty} |x(k, \bar{x}, \mathbf{w})|_\Theta = 0$.

Remark 1. *In order to analyse the control algorithm reported in Section 5.2, a time-varying system has been considered. However, because all the bounds introduced in the ISS Lyapunov function are time-invariant, Theorem 1 can be easily derived by the theorem reported in Magni et al. (2006) for time-invariant systems.* □

5. Nonlinear Model Predictive Control

In this section, the results derived in Theorem 1, are used to analyze the *ISS* property of two open-loop formulations of stabilizing *MPC* algorithms for nonlinear systems. The idea on the base of the two algorithms is the same one. However, there are important differences that, based on the dynamic system under consideration, give advantages to an algorithm rather than to the other in terms of domain of attraction and robustness. Notably, in the following it is not necessary to assume the regularity of the value function and of the resulting control law.

5.1 *MPC* with constant optimization horizon

The system (1) with $w(k) = 0, k \geq t$, is called *nominal model*. Let denote $u_{t_1,t_2} \triangleq \{u(t_1), u(t_1 + 1), \ldots, u(t_2)\}$, $t_2 \geq t_1$, a sequence of vectors and $u_{t_1,t_2}(t_3)$ the vector u_{t_1,t_2} at time t_3. If it is clear on the context the subscript will be omitted. The vector $\hat{x}(k|t)$ is the predicted state of the system at time k $(k \geq t)$ obtained applying the sequence of inputs $u_{t,k-1}$ to the nominal model, starting from the real state $x(t)$ at time t, i.e. $\hat{x}(k|t) = f(\hat{x}(k-1|t), u(k-1))$, $k > t$, $\hat{x}(t|t) = x(t)$.

Assumption 2. *The function $f(\cdot,\cdot)$ is Lispchitz with respect to x and u in $X \times U$, with Lipschitz constants L_f and L_{fu} respectively.*

Remark 2. *Note that the following results could be easily extended to the more general case of $f(\cdot,\cdot)$ uniformly continuous with respect to x and u in $X \times U$. Moreover, note that in virtue of the Heine-Cantor, if X and U are compact, as assumed, then continuity is sufficient to guarantee uniform continuity Limon (2002); Limon et al. (2009).*

Definition 4 (Robust invariant region)**.** *Given a control law $u = \kappa(x)$, $\bar{X} \subseteq X$ is a robust invariant region for the closed-loop system (1) with $u(k) = \kappa(x(k))$, if $\bar{x} \in \bar{X}$ implies $x(k) \in \bar{X}$ and $\kappa(x(k)) \in U$, $\forall w(k) \in \mathcal{W}$, $k \geq t$.* □

Since there are mismatches between real system and nominal model, the predicted evolution using nominal model might differ from the real evolution of the system. In order to consider this effect in the controller synthesis, a bound on the difference between the predicted and the real evolution is given in the following lemma:

Lemma 1. *Limon et al. (2002a) Consider the system (1) satisfying Assumption 2. Then, for a given sequence of inputs, the difference between the nominal prediction of the state $\hat{x}(k|t)$ and the real state of the system $x(k)$ is bounded by*

$$|\hat{x}(k|t) - x(k)| \leq \frac{L_f^{k-t} - 1}{L_f - 1}\gamma, \quad k \geq t.$$

□

To define the *NMPC* algorithms first let

$$\begin{array}{lcl} B_\gamma^{k-t} & \triangleq & \{z \in \mathcal{R}^n : |z| \leq \frac{L_f^{k-t}-1}{L_f-1}\gamma\} \\ X_{k-t} & \triangleq & X \sim B_\gamma^{k-t} \\ & = & \{x \in \mathcal{R}^n : x + y \in X, \forall y \in B_\gamma^{k-t}\} \end{array}$$

then define the following Finite Horizon Optimal Control Problem.

Definition 5 ($FHOCP^1$)**.** *Given the positive integer N, the stage cost l, the terminal penalty V_f and the terminal set X_f, the Finite Horizon Optimal Control Problem ($FHOCP^1$) consists in minimizing, with respect to $u_{t,t+N-1}$, the performance index*

$$J(\bar{x}, u_{t,t+N-1}, N) \triangleq \sum_{k=t}^{t+N-1} l(\hat{x}(k|t), u(k)) + V_f(\hat{x}(t+N|t))$$

subject to

(i) the nominal state dynamics (1) with $w(k) = 0$ and $x(t) = \bar{x}$;

(ii) the state constraints $\hat{x}(k|t) \in X_{k-t}$, $k \in [t, t+N-1]$;

(iii) the control constraints (4), $k \in [t, t+N-1]$;

(iv) the terminal state constraint $\hat{x}(t+N|t) \in X_f$. □

It is now possible to define a "prototype" of the first one of two nonlinear *MPC* algorithms: at every time instant t, define $\bar{x} = x(t)$ and find the optimal control sequence $u^o_{t,t+N-1}$ by solving the $FHOCP^1$. Then, according to the Receding Horizon (RH) strategy, define $\kappa^{MPC}(\bar{x}) = u^o_{t,t}(\bar{x})$ where $u^o_{t,t}(\bar{x})$ is the first column of $u^o_{t,t+N-1}$, and apply the control law

$$u = \kappa^{MPC}(x). \tag{11}$$

Although the $FHOCP^1$ has been stated for nominal conditions, under suitable assumptions and by choosing appropriately the terminal cost function V_f and the terminal constraint X_f, it is possible to guarantee the *ISS* property of the closed-loop system formed by (1) and (11), subject to constraints (2)-(4).

Assumption 3. *The function $l(x,u)$ is such that $l(0,0) = 0$, $l(x,u) \geq \alpha_l(|x|)$ where α_l is a $\mathcal{K}_\infty$-function. Moreover, $l(x,u)$ is Lipschitz with respect to x and u, in $X \times U$, with constant L_l and L_{lu} respectively.*

Remark 3. *Notice that if the stage cost $l(x,u)$ is a piece-wise differentiable function in X and U (as for instance the standard quadratic cost $l(x,u) = x'Qx + u'Ru$) and X and U are bounded sets, then the previous assumption is satisfied.*

Assumption 4. *The design parameter V_f and the set $\Phi \triangleq \{x : V_f(x) \leq \alpha\}$, $\alpha > 0$, are such that, given an auxiliary control law κ_f,*

1. $\Phi \subseteq X_{N-1}$;
2. $\kappa_f(x) \in U$, $\forall x \in \Phi$;
3. $f(x, \kappa_f(x)) \in \Phi$, $\forall x \in \Phi$;
4. $\alpha_{V_f}(|x|) \leq V_f(x) < \beta_{V_f}(|x|)$, $\forall x \in \Phi$, where α_{V_f} and β_{V_f} are $\mathcal{K}_\infty$-functions;
5. $V_f(f(x, \kappa_f(x))) - V_f(x) \leq -l(x, \kappa_f(x))$, $\forall x \in \Phi$;
6. V_f is Lipschitz in Φ with a Lipschitz constant L_v.

Remark 4. *The assumption above can appear quite difficult to be satisfied, but it is standard in the development of nonlinear stabilizing MPC algorithms. Moreover, many methods have been proposed in the literature to compute V_f, Φ satisfying the Assumption 4 (see for example Chen & Allgöwer (1998); De Nicolao et al. (1998); Keerthi & Gilbert (1988); Magni, De Nicolao, Magnani & Scattolini (2001); Mayne & Michalska (1990)).*

Assumption 5. *The design parameter $X_f \triangleq \{x \in \mathcal{R}^n : V_f(x) \leq \alpha_v\}$, $\alpha_v > 0$, is such that for all $x \in \Phi$, $f(x, k_f(x)) \in X_f$.*

Remark 5. *If Assumption 4 is satisfied, then, a value of α_v satisfying Assumption 5 is the following*

$$\alpha_v = (id + \alpha_l \circ \beta_{V_f}^{-1})^{-1}(\alpha).$$

For each $x(k) \in \Phi$ there could be two cases. If $V_f(x(k)) \leq \alpha_v$, then, by Assumption 4, $V_f(x(k+1)) \leq \alpha_v$. If $V(x(k)) > \alpha_v$, then, by point 4 of Assumption 4, $\beta_{V_f}(|x(k)|) \geq V_f(x(k)) > \alpha_v$, that means $|x(k)| > \beta_{V_f}^{-1}(\alpha_v)$. Therefore, by Assumption 3 and point 4 of Assumption 4, one has

$$\begin{aligned} V_f(x(k+1)) &\leq V_f(x(k)) - l(x(k), \kappa_f(x(k))) \leq V_f(x(k)) - \alpha_l(|x(k)|) \\ &\leq \alpha - \alpha_l \circ \beta_{V_f}^{-1}(\alpha_v) \end{aligned}$$

for all $V_f(x(k+1)) \leq \alpha_v$. *Then,* $\alpha_v = \alpha - \alpha_l \circ \beta_{V_f}^{-1}(\alpha_v)$ *satisfy the previous equation. After some manipulations one has* $\alpha_v = (id + \alpha_l \circ \beta_{V_f}^{-1})^{-1}(\alpha)$. □

Let $X^{MPC}(N)$ be the set of states of the system where an admissible solution of the $FHOCP^1$ optimization problem exists.

Definition 6. *Let* $\alpha_1 = \alpha_3 = \alpha_l$, $\alpha_2 = \beta_{V_f}$, $\Xi = X^{MPC}(N)$, $\Omega = \Phi$, $\sigma = L_J$, *where* $L_J \triangleq L_{V_f} L_f^{N-1} + L_l \frac{L_f^{N-1}-1}{L_f-1}$.

Assumption 6. *The values w are such that point 4 of Definition 2 is satisfied with* $V(x) \triangleq J(x, u^o_{t,t+N-1}, N)$.

Remark 6. *From this assumption it is inferred that the allowable size of disturbances is related with the size of the local region* Ω *where the upper bound of the terminal cost is found. This region can be enlarged following the way suggested in Limon et al. (2006). However, this might not produce an enlargement of the allowable size since the new obtained bound is more conservative.* □

The main peculiarities of this $NMPC$ algorithm are the use in the $FHOCP^1$ of: (i) tightened state constraints along the optimization horizon; (ii) terminal set that is only a subset of the region where the auxiliary control law satisfies Assumption 4 in order to guarantee robustness (see Assumptions 4 and 5).

Let introduce now following theorem.

Theorem 2. *Let a system be described by a model given by (1). Assume that Assumptions 1-6 are satisfied. Then the closed loop system (1), (11) is ISS with robust invariant region* $X^{MPC}(N)$ *if the uncertainty is such that*

$$\gamma \leq \frac{\alpha - \alpha_v}{L_v L_f^{N-1}} \tag{12}$$

5.2 MPC with time-varying control horizon

In this sub-section the second algorithm will be shown. It is based on the same ideas of the first one and it is motivated by the attempt to reduce its intrinsic conservativity.
The second Finite Horizon Optimal Control Problem ($FHOCP^2$) to be introduced is characterized by using a time varying control horizon $N_c(t)$ and a (time invariant) prediction horizon N_p. The control horizon is given by

$$N_c(t) \triangleq \left(\left\lfloor \frac{t}{M} \right\rfloor + 1 \right) M - t$$

where $\lfloor \cdot \rfloor$ indicates the integer part operator and M is a parameter which determines its maximum value, i.e. $N_c(t) \in [1, M]$.

Definition 7 ($FHOCP^2$)**.** *Given a stabilizing control law* κ_f *the maximum control horizon M, the prediction horizon* N_p, *the stage cost l, and the terminal penalty* V_f, *the Finite Horizon Optimal Control Problem (*$FHOCP^2$*) consists in minimizing, with respect to* $u_{t,t+N_c(t)-1}$, *the performance index*

$$J(\bar{x}, u_{t,t+N_c(t)-1}, N_c(t), N_p) \triangleq \sum_{k=t}^{t+N_p-1} l(\hat{x}(k|t), u(k)) + V_f(\hat{x}(t+N_p|t))$$

subject to

(i) the nominal state dynamics (1) with $w(k) = 0$ and $\bar{x} = x(t)$;

(ii) the state constraints $\hat{x}(k|t) \in X_{k-t}$, $k \in [t, ..., t + N_c(t) - 1]$;

(iii) the control constraints (4), $k \in [t, ..., t + N_c(t) - 1]$;

(iv) the terminal state constraint $\tilde{x}(t + N_c(t)|t + N_c(t) - M) \in X_f$ where $\tilde{x}$ denotes the nominal prediction of the system considering as initial condition $x(t + N_c(t) - M)$ and applying the sequence of control inputs $\tilde{u}_{t+N_c(t)-M,t+N_c(t)-1}$ defined as

$$\tilde{u}_{t+N_c(t)-M,t+N_c(t)-1}(k) = \begin{cases} u^o_{k,k} & \text{if } k < t \\ u_{t,t+N_c(t)-1}(k) & \text{if } k \geq t \end{cases}$$

(v) the control signal

$$u(k) = \begin{cases} u_{t,t+N_c(t)-1}(k),\ k \in [t, t + N_c(t) - 1] \\ \kappa_f(\hat{x}(k|t)),\ k \in [t + N_c(t), t + N_p - 1] \end{cases} \tag{13}$$

□

It is now possible to introduce the second $NMPC$ algorithm in the following way: at every time instant t, define $\bar{x} = x(t)$ and find the optimal control sequence $u^o_{t,t+N_c(t)-1}$ by solving the $FHOCP^2$. Then, according to the RH strategy, define $\kappa^{MPC}(t, \bar{x}, \tilde{x}(t|t + N_c(t) - M)) = u^o_{t,t}(\bar{x}, \tilde{x}(t|t + N_c(t) - M))$ where $u^o_{t,t}(\bar{x}, \tilde{x}(t|t + N_c(t) - M))$ is the first column of $u^o_{t,t+N_c(t)-1}$, and apply the control law

$$u(t) = \kappa^{MPC}(t, x(t), \tilde{x}(t|t + N_c(t) - M)). \tag{14}$$

Note that the control law is time variant (periodic) due to the time variance of the control horizon $N_c(t)$ and depends also on $\tilde{x}(t|t + N_c(t) - M)$.
Therefore, defining

$$\xi(t) = \begin{bmatrix} x(t) \\ \tilde{x}(t|t + N_c(t) - M)) \end{bmatrix} = \begin{bmatrix} \xi_1(t) \\ \xi_2(t) \end{bmatrix} \in \mathcal{R}^{2n},$$

the closed-loop system formed by (1) and (14) is given by

$$\xi(k + 1) = \tilde{F}(k, \xi(k), w(k)),\ k \geq t, \xi(t) = \bar{\xi} \tag{15}$$

where

$$\tilde{F}(k, \xi(k), w(k)) = \begin{bmatrix} f(\xi_1(k), \kappa^{MPC}(k, \xi_1(k), \xi_2(k))) + w(k) \\ \begin{cases} f(\xi_2(k), \kappa^{MPC}(k, \xi_1(k), \xi_2(k))),\ \forall (k+1) \notin \mathcal{T}_M \\ f(\xi_1(k), \kappa^{MPC}(k, \xi_1(k), \xi_2(k))) + w(k),\ \forall (k+1) \in \mathcal{T}_M \end{cases} \end{bmatrix}$$

Definition 8. *Let $X^{MPC}(t, N_p) \in \mathcal{R}^{2n}$ be the set of states $\xi(t)$ where an admissible solution of the $FHOCP^2$ exists.* □

Noting that $x(t) = \tilde{x}(t|t + N_c(t) - M))$, $\forall t \in \mathcal{T}_M$ since $N_c(t) = M$, the closed-loop system (1), (14) for $k \in \mathcal{T}_M$ is time invariant since the control law is time invariant and

$$x(k+M) = \bar{F}(x(k), w_{k,k+M-1}),\ \forall k \in \mathcal{T}_M,\ k \geq t,\ x(t) = \bar{x}. \tag{16}$$

Definition 9. *Let $X_M^{MPC}(N_p) \in \mathcal{R}^n$ be the set x of states of the system (1) where an admissible solution of the $FHOCP^2$ exists $\forall t \in \mathcal{T}_M$.* □

As in the previous algorithm, although the $FHOCP^2$ has been stated for nominal conditions, under suitable assumptions and by choosing accurately the terminal cost function V_f and the terminal constraint X_f, it is possible to guarantee the ISS property of the closed-loop system formed by (1) and (14), subject to constraints (2)-(4).

Assumption 7. *The auxiliary control law κ_f is Lipschitz in Φ with a Lipschitz constant L_κ where $\Phi \triangleq \{x \in X_{M-1} : V_f(x) \leq \alpha\},\ \alpha > 0$.*

Remark 7. *Note that, an easy way to satisfy Assumption 7 is to choose κ_f linear, e.g. the solution of the infinite horizon optimal control problem for the unconstrained linear system.*

Assumption 8. *The design parameter $X_f \triangleq \{x \in \mathcal{R}^n : V_f(x) \leq \alpha_v\}$ is such that, considering the system (1), with $u = \kappa_f(x)$ and $w(k) = 0$, for all $x(t) \in \Phi$ results $\hat{x}(t+M|t) \in X_f$ and $\hat{x}(k|t) \in X_{k-t}, k \in [t, t+M-1]$.*

Definition 10. *Let $\alpha_1 = \alpha_3 = \alpha_l$, $\alpha_2 = \beta_{V_f}$, $\Xi = X_M^{MPC}(N_p)$, $\Omega = \Phi$, $\sigma = L_J^M$, where*

$$L_J^M \triangleq \sum_{k=t}^{t+M-1} \left\{ L_l \frac{L_f^{N_c(k)-1} - 1}{L_f - 1} + L_{lx} L_f^{N_c(k)-1} \frac{L_x^{N_p - N_c(k)+1} - 1}{L_x - 1} + L_v L_f^{N_c(k)-1} L_x^{N_p - N_c(k)+1} \right\}$$

with $L_x \triangleq (L_f + L_{fu} L_\kappa)$ and $L_{lx} \triangleq (L_l + L_{lu} L_\kappa)$.

Assumption 9. *The values w are such that point 4 of Definition 2 is satisfied with $V(x) \triangleq J(x, u^o_{t,t+M-1}, M, N_p)$.*

The main peculiarities of this $NMPC$ algorithm, with respect to the one previously presented, are the use in the $FHOCP^2$ of: (i) a time varying control horizon; (ii) a control horizon that is different from prediction horizon; (iii) the fact that the real value of the state is updated only each M step to check the terminal constraint while it is updated at each step for the computation of cost. These modifications allows to relax Assumption 5 with Assumption 8. In this way it could be possible to enhance the robustness. The idea to use the measure of the state only each M step has been already used in an other context in contractive MPC de Oliveira Kothare & Morari (2000).

Theorem 3. *Let a system be described by a model given by (1). Assume that Assumptions 1-4, 7-9 are satisfied. Then the closed loop system (15) is ISS with robust invariant region $X^{MPC}(t, N_p)$ if the uncertainty is such that*

$$\gamma \leq \frac{\alpha - \alpha_v}{L_v \frac{L_f^M - 1}{L_f - 1}} \tag{17}$$

Different from Magni, De Nicolao, Magnani & Scattolini (2001) the use of a prediction horizon longer than the control horizon does not affect the size of the robust invariant region because the terminal inequality constraint has been imposed at the end of the control horizon. However the following theorem proves that this choice has positive effect on the performance.

Theorem 4. *Magni, De Nicolao, Magnani & Scattolini (2001) Letting* $l(x,u) = x'Qx + u'Ru$, $Q > 0, R > 0$, $u = -K^{LQ}x$ *the solution of the infinite horizon optimal control problem for the unconstrained linear system*

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

with $A = \partial f(x,u)/\partial x|_{x=0,u=0}$, $B = \partial f(x,u)/\partial u|_{x=0,u=0}$, *for each given* N_c, *if* $\kappa_f(x) = -K^{LQ}x$, *then* $\lim_{N_p\to\infty} \partial\kappa^{MPC}(x)/\partial x|_{x=0} = K^{LQ}$.

In conclusion, Theorems 2 and 3 proven that both the algorithm guarantee the *ISS* of the closed-loop system. However a priori it is not possible to establish which of the two algorithms give more robustness. This because of the dependance from the values of L_f, M, N_p of the bounded on the maximum disturbance allowed. Therefore, based on the dynamic system in object, it will be used an algorithm rather than the other.

6. Examples

The objective of the examples is to show that, based on the values of certain parameters, one algorithm can be better than the other. In particular two examples are shown: in the first one the algorithm based on $FHOCP^1$ is better than the one based on $FHOCP^2$ in terms of robustness; in the second one the contrary happens.

6.1 Example 1

Consider the uncertain nonlinear system given by

$$\begin{aligned} x_1(k+1) &= 0.55x_1(k) + 0.12x_2(k) + (0.01 - 0.6x_1(k) + x_2(k) + \Lambda_1)u(k) \\ x_2(k+1) &= 0.67x_2(k) + (0.15 + x_1(k) - 0.8x_2(k) + \Lambda_2)u(k) \end{aligned}$$

where Λ_1 and Λ_2 are the parameters of the system model uncertainty. The control is constrained to be $|u| \leq u_{\max} = 0.2$. Defining $w = [\Lambda_1 u^T \ \Lambda_2 u^T]^T$ the disturbance is in the form (1) and the nominal system is in the form $x(k+1) = Ax + Bu + Cxu$. Considering the ∞-norm, the Lipschitz constant of the system is

$$L_f = \max_u(|A + Cu|_\infty) = \max\{|A + 3C|_\infty, |A - 3C|_\infty\} = 1.03.$$

In the formulation of the $FHOCP^1$ and $FHOCP^2$ the stage is $l(x,u) = x'Qx + u'Ru$ with $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $R = 1$ and the auxiliary control law $u = -K^{LQ}x$ is derived by solving an Infinite Horizon optimal control problem for the linearized system around the origin

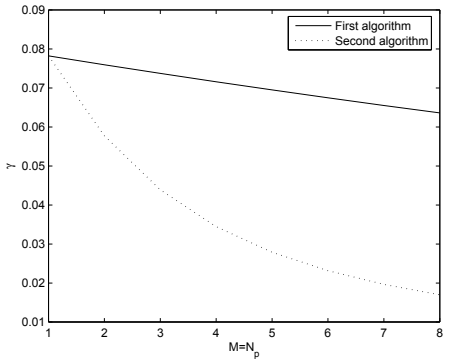
$$\begin{aligned} x_1(k+1) &= 0.55x_1(k) + 0.12x_2(k) + 0.01u(k) \\ x_2(k+1) &= 0.67x_2(k) + 0.15u(k) \end{aligned}$$

with the same stage cost. The solution of the associated Riccati Equation is $P = \begin{bmatrix} 1.4332 & 0.1441 \\ 0.1441 & 1.8316 \end{bmatrix}$ so that the value of K^{LQ} is $K^{LQ} = \begin{bmatrix} -0.0190 & -0.1818 \end{bmatrix}$. The value of the

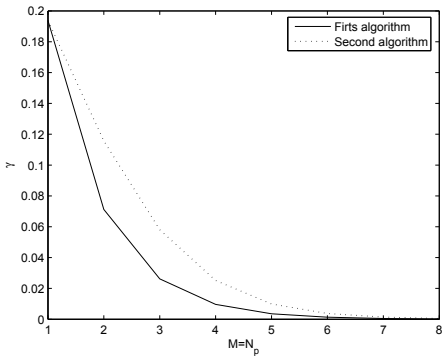
Lipschitz constant L_κ of the auxiliary control law is $L_\kappa = |K^{LQ}|_\infty = 0.1818$. The terminal penalty $V_f(x) = \beta x'Px$, where $\beta = 1.2$ satisfies

$$\lambda_{\max}(Q + K^{LQ\prime}RK^{LQ}) < \beta\lambda_{\min}(Q + K^{LQ\prime}RK^{LQ})$$

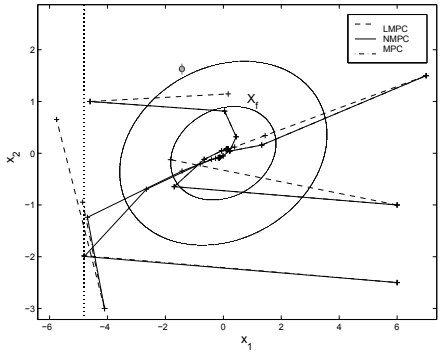
in order to verify Assumption 7. Therefore, considering the presence of the constraint on the control, the linear controller $u = -K^{LQ}x$ stabilizes the system only in the invariant set Φ, $\Phi = \{x : 1.2x'Px \leq \alpha = 0.2\}$ The value of the Lipschitz constant L_v is $L_v = \max_{x\in\Phi} |2\beta Px|_\infty = 2.4|Px|_\infty = 1.3222$. For the algorithm based on $FHOCP^2$ the final constraint X_f depends on the value M while for the algorithm based on $FHOCP^1$ it results $X_f = \{x : 3x'Px \leq 0.0966\}$. In Figure 1.a the maximum value of γ that satisfies (12) (solid line) and the one that satisfies the (17) (dotted line) for different values of M, are reported. In this example the algorithm based on the $FHOCP^1$ guarantees major robustness than the one based on $FHOCP^2$.



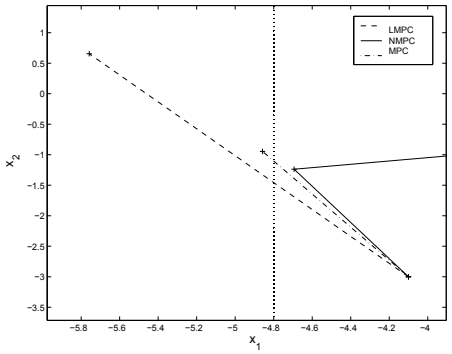
(a) Example 1: comparison of γ between the two algorithms.



(b) Example 2: comparison of γ between the two algorithms.



(c) Example 2: closed loop state evolution.



(d) Example 2: detail of the closed-loop state evolution with initial state (-4.1;-3).

6.2 Example 2

This example shows a case in which the algorithm based on $FHOCP^2$ gives a better solution. Consider the uncertain nonlinear system

$$\begin{aligned} x_1(k+1) &= x_2(k) + (0.3x_2(k) + \Lambda_1)u \\ x_2(k+1) &= -0.32x_1(k) + 1.8x_2(k) + (1 - 0.2x_2(k) + \Lambda_2)u \end{aligned}$$

where Λ_1 and Λ_2 are the parameters of the system model uncertainty. The control is constrained to be $|u| \leq u_{\max} = 3$ and the state x_1 is constrained to be $x_1 \geq -4.8$. Considering the ∞-norm, the Lipschitz constant of the system is

$$L_f = \max_u(|A + Cu|_\infty) = \max\{|A + 3C|_\infty, |A - 3C|_\infty\} = 2.72.$$

In the formulation of the $FHOCP^1$ and $FHOCP^2$ the stage is $l(x,u) = x'Qx + u'Ru$ with $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $R = 1$ and the auxiliary control law $u = -K^{LQ}x$ is derived by solving an Infinite Horizon optimal control problem for the linearized system around the origin

$$\begin{aligned} x_1(k+1) &= x_2(k) \\ x_2(k+1) &= -0.32x_1(k) + 1.8x_2(k) + u \end{aligned}$$

with the same stage cost. The solution of the associated Riccati Equation is $P = \begin{bmatrix} 1.0834 & -0.4428 \\ -0.4428 & 4.3902 \end{bmatrix}$ so that the value of K^{LQ} is $K^{LQ} = \begin{bmatrix} -0.2606 & 1.3839 \end{bmatrix}$. The value of the Lipschitz constant L_κ of the auxiliary control law is $L_\kappa = |K^{LQ}|_\infty = 1.3839$. The terminal penalty $V_f(x) = \beta x'Px$, where $\beta = 3$, satisfies

$$\lambda_{\max}(Q + K^{LQ\prime}RK^{LQ}) < \beta\lambda_{\min}(Q + K^{LQ\prime}RK^{LQ})$$

in order to verify Assumption 7. Therefore, considering the presence of the constraint on the control, the linear controller $u = -K^{LQ}x$ stabilizes the system only in the invariant set Φ, $\Phi = \{x : 3x'Px \leq \alpha = 40.18\}$. The value of the Lipschitz constant L_v is $L_v = \max_{x\in\Phi} |2\beta Px|_\infty = 6|Px|_\infty = 45.9926$. For the algorithm based on $FHOCP^2$ the final constraint X_f depends on the value M while for the algorithm based on $FHOCP^1$ it results $X_f = \{x : 3x'Px \leq 31.2683\}$. In Figure 1.b the maximum value of γ that satisfies (12) (solid line) and the one that satisfies the (17) (dotted line) for different values of M, are reported. In this example, the advantage of the algorithm based on the $FHOCP^2$ with respect to first one is due to the fact that the auxiliary control law can lead the state of the nominal system from Φ to X_f in M steps rather than in only one. Hence, since the difference between Φ and X_f is bigger, then a bigger perturbation can be tolerated. In Figure 1.c the state evolutions of the nonlinear system obtained with different control strategies with initial condition

x_{01}	6	−4.1	7	6	−4.6
x_{02}	−2.5	−3	1.5	−1	1

and $\gamma = 0.0581$ are reported: in solid line, using the new algorithm ($NMPC$), with $N_p = 10$ and $M = 3$, in dashed line, using the new algorithm but with the linearized system in the solution of the $FHOCP$ ($LMPC$) and in dash-dot line the results of a nominal MPC (MPC) with $N_p = 10$ and $N_c = 3$. It is clear that, since the model used for the $FHOCP$ differs from the nonlinear model, using $LMPC$ feasibility is not guaranteed along the trajectory as shown with

initial states $[-4.6; 1]$, $[-4.1; -3]$, $[6; -1]$. Also with the nominal *MPC*, as shown with initial states $[-4.1; -3]$, $[6; -2.5]$, since uncertainty is not considered, feasibility is not guaranteed. Figure 1.d shows a detail of the unfeasibility phenomenon from the first to the second time instant with initial state $[-4.1; -3]$. The state constraint infact is robustly fulfilled only with the *NMPC* algorithm. For the other initial states, the evolutions of the three strategies are close.

7. Conclusions

In this paper two design procedures of nominal *MPC* controllers are presented. The objective of these algorithms is to provide some degree of robustness when model mismatches are present. Regional Input-to-State Stability (*ISS*) has been used as theoretical framework of the closed loop analysis. Both controllers assume the Lipschitz continuity of the model and of the stage cost and terminal cost functions. Robust constraint satisfaction is ensured by introducing restricted constraints in the optimization problem based on the estimation of the maximum effect of the uncertainty. The main differences between the proposed algorithms are that the second one uses a time varying control horizon and, in order to check the terminal constraints, it updates the state with the real one just only each M steps. Theorem 2 and Theorem 3 give sufficient condition on the maximum uncertainty in order to guarantee regional *ISS*. The bounds depend on both system parameters and control algorithm parameters. These conditions, even if only sufficient, give an idea on the algorithm that it is better to use for a particular system.

8. Appendix

Lemma 2. *Let $x \in X_{k-t}$ and $y \in \mathcal{R}^n$ such that $|y - x| \leq L_f^{k-t-1}\gamma$. Then $y \in X_{k-t-1}$.*

Proof : Consider $e_{k-t-1} \in B_\gamma^{k-t-1}$, and let denote $z = y - x + e_{k-t-1}$. It is clear that

$$|z| \leq |y - x| + |e_{k-t-1}| \leq L_f^{k-t-1}\gamma + \frac{L_f^{k-t-1} - 1}{L_f - 1}\gamma = \frac{L_f^{k-t} - 1}{L_f - 1}\gamma$$

thus, $z \in B_\gamma^{k-t}$. Taking into account that $x \in X_{k-t}$, for all $e_{k-t-1} \in B_\gamma^{k-t-1}$, it results that $y + e_{k-t-1} = (x + z) \in X$. This yields that $y \in X_{k-t-1}$. ■

Proof of Theorem 2: Firstly, it will be shown that region $X^{MPC}(N)$ is robust positively invariant for the closed loop system: if $x(t) \in X^{MPC}(N)$, then $x(t+1) = f(x(t), u^o(t)) + w(t) \in X^{MPC}(N)$ for all $w(t) \in W$. This is achieved by proving that for all $x(t) \in X^{MPC}(N)$, there exists an admissible solution of the optimization problem in $t+1$, based on the optimal solution in t, i.e. $\bar{u}_{t+1,t+N} = [u^o_{t+1,t+N-1}, k_f(\hat{x}(t+N|t+1))]$. Let denote $\bar{x}(k|t+1)$ the state obtained applying the input sequence $\bar{u}_{t+1,k-1}$ to the nominal model with initial condition $x(t+1)$. In order to prove that the sequence $\bar{u}_{t+1,t+N}$ is admissible, it is necessary that

a) $\bar{u}(k) \in U$, $k \in [t+1, t+N]$: it follows from the feasibility of $u^o_{t,t+N-1}$ and the fact that $\kappa_f(x) \in U, \forall x \in X_f \subseteq \Phi$.

b) $\bar{x}(t+N+1|t+1) \in X_f$: first, it is going to be shown that $\bar{x}(t+N|t+1) \in \Phi$. Taking into account that $|\bar{x}(t+N|t+1) - \hat{x}(t+N|t)| \leq L_f^{N-1}\gamma$ then

$$V_f(\bar{x}(t+N|t+1)) \leq V_f(\hat{x}(t+N|t)) + L_v L_f^{N-1}\gamma \leq \alpha_v + L_v L_f^{N-1}\gamma \leq \alpha.$$

Therefore $\bar{x}(t+N|t+1) \in \Phi$ and hence, applying the auxiliary control law, $\bar{x}(t+N+1|t+1) \in X_f$.

c) $\bar{x}(k|t+1) \in X_{k-t-1}$, $k \in [t+1, t+N]$: considering that $|x(t+1) - \hat{x}(t+1|t)| \leq \gamma$ by recursion $|\bar{x}(k|t+1) - \hat{x}(k|t)| \leq L_f^{k-t-1}\gamma$ for $k \in [t+1, t+N]$. Since $\hat{x}(k|t) \in X_{k-t}$, then, by Lemma 2, $\bar{x}(k|t+1) \in X_{k-t-1}$. Moreover, since $\bar{x}(t+N|t+1) \in \Phi \subseteq X_{N-1}$, the proof is completed.

Now, in order to show that the closed loop system (1), (11) is ISS in $X^{MPC}(N)$, let verify that $V(\bar{x}, N) \triangleq J(\bar{x}, u^o_{t,t+N-1}, N)$ is an ISS-Lyapunov function in $X^{MPC}(N)$. First note that by Assumption 3

$$V(\bar{x}, N) \geq \alpha_l(|\bar{x}|),\ \forall \bar{x} \in X^{MPC}(N). \tag{18}$$

Moreover, in view of Assumption 4, $\tilde{u}_{t,t+N} = [u^o_{t,t+N-1}, k_f(\hat{x}(t+N|t))]$ is an admissible, possible suboptimal, control sequence for the $FHOCP$[1] with horizon $N+1$ at time t with cost

$$\begin{aligned} J(\bar{x}, \tilde{u}_{t,t+N}, N+1) &= V(\bar{x}, N) - V_f(\hat{x}(t+N|t)) + V_f(\hat{x}(t+N+1|t)) \\ &\quad + l(\hat{x}(t+N|t), k_f(\hat{x}(t+N|t))). \end{aligned}$$

Since $\tilde{u}_{t,t+N}$ is a suboptimal sequence, $V(\bar{x}, N+1) \leq J(\bar{x}, \tilde{u}_{t,t+N}, N+1)$ and, using point 5 of Assumption 4, it follows that $J(\bar{x}, \tilde{u}_{t,t+N}, N+1) \leq V(\bar{x}, N)$. Then

$$V(\bar{x}, N+1) \leq V(\bar{x}, N),\ \forall \bar{x} \in X^{MPC}(N)$$

with $V(\bar{x}, 0) = V_f(\bar{x})$, $\forall \bar{x} \in \Phi$. Therefore

$$V(\bar{x}, N) \leq V(\bar{x}, N-1) \leq V_f(\bar{x}) < \beta_{V_f}(|\bar{x}|),\ \forall \bar{x} \in \Phi. \tag{19}$$

Moreover, let define ΔJ as

$$\begin{aligned} \Delta J &\triangleq J(x(t+1), \bar{u}_{t+1,t+N}, N) - J(x(t), u^o_{t,t+N-1}, N) \\ &= -l(x(t), u^o(t)) + \sum_{k=t+1}^{k=t+N-1} \{l(\bar{x}(k|t+1), \bar{u}(k)) - l(\hat{x}(k|t), u^o(k))\} \\ &\quad + l(\bar{x}(t+N|t+1), \bar{u}(t+N)) + V_f(\bar{x}(t+N+1|t+1) - V_f(\hat{x}(t+N|t)). \end{aligned} \tag{20}$$

From the definition of $\bar{u}$, $\bar{u}(k) = u^o(k)$, for $k \in [t+1, t+N-1]$, and hence $l(\bar{x}(k|t+1), \bar{u}(k)) - l(\hat{x}(k|t), u^o(k)) \leq L_l L_f^{k-t-1}\gamma$ and analogously

$$V_f(\bar{x}(t+N|t+1) - V_f(\hat{x}(t+N|t)) \leq L_v L_f^{N-1}\gamma.$$

Substituting these expressions in (20) and considering that $\bar{x}(t+N|t+1) \in \Phi$, from Assumption 4, there is

$$\begin{aligned} \Delta J &\leq [l(\bar{x}(t+N|t+1), \bar{u}(t+N)) + V_f(\bar{x}(t+N+1|t+1) - V_f(\bar{x}(t+N|t+1)] \\ &\quad - l(x(t), u^o(t)) + L_J\gamma \leq -l(x(t), u^o(t)) + L_J\gamma \end{aligned}$$

where $L_J \triangleq L_v L_f^{N-1} + L_l \frac{L_f^{N-1}-1}{L_f-1}$. Considering that by Assumption 3, $l(x,u) \geq \alpha_l(|x|)$ and the optimality of the solution, then

$$V(x(t+1),N) - V(x(t),N) \leq \Delta J \leq -\alpha_l(|x(t)|) + L_J\gamma, \ \forall x \in X^{MPC}(N) \tag{21}$$

Therefore, by (18), (19) and (21), $V(\bar{x},N)$ is an *ISS*-Lyapunov function of the closed loop system (1), (11), and hence, the closed-loop system is *ISS* with robust invariant region $X^{MPC}(N)$. ■

Proof of Theorem 3: Firstly, it will be shown that region $X^{MPC}(t,N_p)$ is robust positively invariant for the closed-loop system. This is achieved by proving that for all $\xi(t) \in X^{MPC}(t,N_p)$, there exists an admissible solution $\bar{u}_{t+1,t+1+N_c(t+1)-1}$ of the optimization problem in $t+1$, based on the optimal solution in t. This sequence is given by

$$\bar{u}_{t+1,t+1+N_c(t+1)-1}(k) = \begin{cases} u^o_{t,t+N_c(t)-1}(k) & \text{if } t+1 \notin \mathcal{T}_M \\ \kappa_f(\hat{x}(k|t+1)) & \text{if } t+1 \in \mathcal{T}_M \end{cases}$$

for $k \in [t+1, \cdots, t+1+N_c(t+1)-1]$. Notice that if $t+1 \notin \mathcal{T}_M$, $N_c(t+1) = N_c(t)-1$ and hence the sequence is well defined.
Moreover, since necessary for the *ISS* proof, it will be shown that, starting from the (nominal) state $\hat{x}(t+1|t)$, the sequence $\bar{u}'_{t+1,t+1+N_c(t+1)-1}$ is admissible. This is given by

$$\bar{u}'_{t+1,t+1+N_c(t+1)-1}(k) = \begin{cases} u^o_{t,t+N_c(t)-1}(k) & \text{if } t+1 \notin \mathcal{T}_M \\ \kappa_f(\hat{x}(k|t)) & \text{if } t+1 \in \mathcal{T}_M \end{cases}$$

for $k \in [t+1, \cdots, t+1+N_c(t+1)-1]$.

In order to prove that the two sequences are admissible, it is necessary that
1) $\tilde{x}(t+1+N_c(t+1)|t+1+N_c(t+1)-M) \in X_f$ with $\tilde{u}_{t+1+N_c(t+1)-M,t+1+N_c(t+1)-1}$ derived from both $\bar{u}$ and $\bar{u}'$;
2) $\hat{x}(k|t+1) \in X_{k-t-1}$, $k \in [t+1, t+1+N_c(t+1)-1]$ with input $\bar{u}$;
3) $\hat{x}(k|t) \in X_{k-t}$, $k \in [t+1, t+1+N_c(t+1)-1]$ with input $\bar{u}'$;
4) $\bar{u}(k) \in U$, $\bar{u}'(k) \in U$, $k \in [t+1, t+1+N_c(t+1)-1]$.

1) First note that if $t+1 \notin \mathcal{T}_M$, then $\bar{u}(k) = \bar{u}'(k) = u^o(k)$, $k \in [t+1, t+1+N_c(t+1)-1]$. This yields to $\tilde{x}(k|t+N_c(t)-M) = \tilde{x}(k|t+1+N_c(t+1)-M)$ for all $k \in [t+1+N_c(t+1)-M, t+1+N_c(t+1)]$ and hence

$$\tilde{x}(t+1+N_c(t+1)|t+1+N_c(t+1)-M) = \tilde{x}(t+N_c(t)|t+N_c(t)-M) \in X_f.$$

On the contrary, if $t+1 \in \mathcal{T}_M$ then $\bar{u}_{t+1,t+1+N_c(t+1)-1}(k) = \kappa_f(\hat{x}(k|t+1))$ and $\bar{u}'_{t+1,t+1+N_c(t+1)-1}(k) = \kappa_f(\hat{x}(k|t))$. We are going to prove that both sequence satisfies the terminal constraint:

- Consider the sequence $\bar{u}$ and let denote $\tilde{u}$ and $\tilde{x}$ the sequence and predictions derived from $\bar{u}$. In virtue of Lemma 1 and the fact that $N_c(t) = 1$, the following inequality holds

$$|x(t+1) - \tilde{x}(t+1|t+N_c(t)-M)| \leq \frac{L_f^M - 1}{L_f - 1}\gamma \tag{22}$$

and by point 5 of Assumption 4 it follows that

$$V_f(x(t+1)) - V_f(\tilde{x}(t+1|t+N_c(t)-M)) \\ \leq L_v|x(t+1) - \tilde{x}(t+1|t+N_c(t)-M)| \leq L_v \frac{L_f^M - 1}{L_f - 1}\gamma$$

Hence, considering that $\tilde{x}(t+1|t+N_c(t)-M) \in X_f$ and the uncertainty satisfies (17), then

$$V_f(x(t+1)) \leq V_f(\tilde{x}(t+1|t+N_c(t)-M)) + L_v \frac{L_f^M - 1}{L_f - 1}\gamma \leq \alpha_v + L_v \frac{L_f^M - 1}{L_f - 1}\gamma \leq \alpha \tag{23}$$

and therefore $x(t+1) \in \Phi$. Hence, from Assumption 8, $\kappa_f(\hat{x}(k|t+1))$ steers the nominal state in X_f in M steps. Then $\bar{u}_{t+1,t+N_c(t+1)-1}$ satisfies the constraint.

- Let consider now $\bar{u}'$ and let denote $\tilde{u}'$ and $\tilde{x}'$ the sequence and predictions derived from $\bar{u}'$. Since $\hat{x}(t+1|t) = f(x(t), u^o_{t,t})$ we have that

$$\begin{array}{rcl} & |\hat{x}(t+1|t) - \tilde{x}'(t+1|t+N_c(t)-M)| & \\ = & |f(x(t), u^o(t)) - f(\tilde{x}'(t|t+N_c(t)-M), u^o(t))| & \\ \leq & L_f|x(t) - \tilde{x}'(t|t+N_c(t)-M)| & \end{array}$$

and from (22) $|\hat{x}(t+1|t) - \tilde{x}'(t+1|t+N_c(t)-M)| \leq L_f \frac{L_f^{M-1}-1}{L_f-1}\gamma$. Finally, following the same idea used to derive (23)

$$\begin{array}{rcl} V_f(\hat{x}(t+1|t)) & \leq & V_f(\tilde{x}'(t+1|t+N_c(t)-M)) + L_v L_f \dfrac{L_f^{M-1}-1}{L_f-1}\gamma \\ & < & \alpha_v + L_v \dfrac{L_f^M - 1}{L_f - 1}\gamma \leq \alpha. \end{array} \tag{24}$$

Therefore $V_f(\hat{x}(t+1|t)) < \alpha$ and consequently $\hat{x}(t+1|t) \in \Phi$. Hence $\kappa_f(\hat{x}(k|t))$ steers the nominal state in X_f in M steps. Then $\bar{u}'_{t+1,t+N_c(t+1)-1}$ satisfies the constraint.

2) Consider the sequence of inputs $\bar{u}$ and assume that $t+1 \notin \mathcal{T}_M$, then, since by optimality of solution at time t, $\hat{x}(k|t) \in X_{k-t}$ and

$$|\hat{x}(k|t+1) - \hat{x}(k|t)| \leq L_f^{k-t-1}\gamma,\ k \in [t+1, t+1+N_c(t+1)-1]$$

from Lemma 2, it follows that $\hat{x}(k|t+1) \in X_{k-t-1}$. If $t \in \mathcal{T}_M$ then $x(t+1) \in \Phi$ as shown in (23), and from Assumptions 4, 7, the constraints satisfaction is directly derived.

3) Consider that the sequence $\bar{u}'_{t+1,t+1+N_c(t+1)-1}$ is applied from the state $\hat{x}(t+1|t)$. If $t+1 \notin \mathcal{T}_M$ then the constraints are satisfied since $\hat{x}(k|t) \in X_{k-t}$. If $t+1 \in \mathcal{T}_M$, as shown in (24), $\hat{x}(t+1|t) \in \Phi$ and then, by Assumptions 4, 7, constraints satisfaction is directly derived.

4) From the admissibility of $u^o_{t,t+N_c(t)-1}$ and the fact that for all $x \in \Phi$, $\kappa_f(x) \in U$, it follows that $\bar{u}(k) \in U$, $\bar{u}'(k) \in U$, $k \in [t+1, t+1+N_c(t+1)-1]$.

Now, in order to show that the closed loop system (15) is *ISS* in $X^{MPC}(t, N_p)$, it is first proven that the closed-loop system (16), defined for each $t \in \mathcal{T}_M$, is *ISS* in $X_M^{MPC}(N_p)$.

In order to prove the first part let verify that $V(\bar{x}, M, N_p) \triangleq J(\bar{x}, u^o_{t,t+M-1}, M, N_p)$, is an ISS-Lyapunov function for the system (16).

Let denote $\bar{x}(k|t+1)$ and $\bar{x}'(k|t)$ the state evolution obtained with input $\bar{u}(k)$ and initial state $x(t+1)$ and with input $\bar{u}'(k)$ and initial state $x(t+1|t)$ respectively. Let call $J^*(t,x)$, $\bar{J}(x)$ and $\bar{J}'(x)$ the optimal cost and the cost relative to the admissible sequences $\bar{u}$ and $\bar{u}'$ respectively. First note that by Assumption 3

$$V(\bar{x}, M, N_p) \geq \alpha_l(|\bar{x}|),\ \forall \bar{x} \in X_M^{MPC}(N_p). \tag{25}$$

Moreover, $\tilde{u}_{t,t+M-1} = u^o_{t,t+M-1}$, where $u^o_{t,t+M-1}$ is the optimal control sequence for the $FHOCP^2$ with prediction horizon N_p, is an admissible, possible suboptimal, control sequence for the $FHOCP^2$ with control horizon M and prediction horizon N_p+1 at time t with cost

$$\begin{aligned} J(\bar{x}, \tilde{u}_{t,t+M-1}, M, N_p+1) &= V(\bar{x}, M, N_p) - V_f(\hat{x}(t+N_p|t)) + V_f(\hat{x}(t+N_p+1|t)) \\ &\quad + l(\hat{x}(t+N_p|t), k_f(\hat{x}(t+N_p|t))). \end{aligned}$$

Since $\tilde{u}_{t,M-1}$ is a suboptimal sequence $V(\bar{x}, M, N_p+1) \leq J(\bar{x}, \tilde{u}_{t,t+M-1}, M, N_p+1)$ and, using point 5 of Assumption 4, it follows that $J(\bar{x}, \tilde{u}_{t,t+M-1}, N_p+1) \leq V(\bar{x}, M, N_p)$. Then $V(\bar{x}, M, N_p+1) \leq V(\bar{x}, M, N_p)$, $\forall \bar{x} \in X_M^{MPC}(N_p)$, $N_p \geq M$. In particular, it is true that $V(\bar{x}, M, N_p) \leq V(\bar{x}, M, M)$, $\forall \bar{x} \in X_M^{MPC}(M)$. Now, in view of Assumption 4, $\tilde{u}_{t,t+M} = [u^o_{t,t+M-1}, k_f(\hat{x}(t+M|t))]$ is an admissible, possible suboptimal, control sequence for the $FHOCP^2$ with horizon $M+1$ with cost

$$\begin{aligned} J(\bar{x}, \tilde{u}_{t,t+M}, M+1, M+1) &= V(\bar{x}, M, M) - V_f(\hat{x}(t+M|t)) + V_f(\hat{x}(t+M+1|t)) \\ &\quad + l(\hat{x}(t+M|t), k_f(\hat{x}(t+M|t))). \end{aligned}$$

Since $\tilde{u}_{t,t+M}$ is a suboptimal sequence $V(\bar{x}, M+1, M+1) \leq J(\bar{x}, \tilde{u}_{t,t+M}, M+1, M+1)$ and, using point 5 of Assumption 4, it follows that $J(\bar{x}, \tilde{u}_{t,t+M}, M+1) \leq V(\bar{x}, M, M)$. Then $V(\bar{x}, M+1, M+1) \leq V(\bar{x}, M, M)$, $\forall \bar{x} \in X_M^{MPC}(M)$ with $V(\bar{x}, 0, 0) = V_f(\bar{x})$, $\forall \bar{x} \in \Phi$. Therefore

$$V(\bar{x}, M, M) \leq V(\bar{x}, M-1, M-1) \leq V_f(\bar{x}) < \beta_{V_f}(|\bar{x}|),\ \forall \bar{x} \in \Phi. \tag{26}$$

Moreover, let calculate

$$\begin{aligned} &\bar{J}'(\hat{x}(t+1|t)) - J^*(t, x(t)) \\ = &\sum_{k=t+1}^{t+1+N_c(t+1)-1} l(\bar{x}'(k|t), \bar{u}'(k)) + \sum_{k=t+1+N_c(t+1)}^{t+N_p} l(\bar{x}'(k|t), \kappa_f(\bar{x}'(k|t))) \\ &- \sum_{k=t}^{t+N_c(t)-1} l(\hat{x}(k|t), u^o(k)) - \sum_{k=t+N_c(t)}^{t+N_p-1} l(\hat{x}(k|t), \kappa_f(\hat{x}(k|t))) \\ &+ V_f(\bar{x}'(t+1+N_p|t)) - V_f(\hat{x}(t+N_p|t)). \end{aligned}$$

Since, both the state evolutions are obtained with initial condition $\hat{x}(t+1|t)$ and the same input sequence from time $t+1$ and until $t+N_p-1$, there is $\bar{x}'(k|t) = \hat{x}(k|t),\ k \in [t+1, t+N_p]$ so that

$$\begin{aligned} \bar{J}'(\hat{x}(t+1|t)) - J^*(t, x(t)) &= l(\hat{x}(t+N_p|t), \kappa_f(\hat{x}(t+N_p|t))) - l(x(t), u^o(t)) \\ &\quad + V_f(\hat{x}(t+1+N_p|t)) - V_f(\hat{x}(t+N_p|t)). \end{aligned}$$

Using point 5 of the Assumption 4

$$\bar{J}'(\hat{x}(t+1|t)) - J^*(t,x(t)) \leq -l(x(t),\kappa^{MPC}(t,x(t))). \tag{27}$$

Let consider now the difference

$$\begin{aligned} J(x(t+1)) - \bar{J}'(\hat{x}(t+1|t)) \quad = \quad & \sum_{k=t+1}^{t+N_c(t)-1} \{l(\bar{x}(k|t+1),\bar{u}(k)) - l(\bar{x}'(k|t),\bar{u}'(k))\} \\ & + \sum_{k=t+N_c(t)}^{t+N_p} \{l(\bar{x}(k|t+1),\kappa_f(\bar{x}(k|t+1))) \\ & - l(\bar{x}'(k|t),\kappa_f(\bar{x}'(k|t)))\} \\ & + V_f(\bar{x}(t+1+N_p|t+1)) - V_f(\bar{x}'(t+1+N_p|t)). \end{aligned}$$

Note that $\bar{u}(k) = \bar{u}'(k)$, $k \in [t+1, t+N_c(t)-1]$, while the signals are different for $k > t+N_c(t)-1$. Since $|\bar{x}(k|t+1) - \bar{x}'(k|t)| \leq L_f^{k-t-1}\gamma$ from Assumption 3 it is derived that

$$|l(\bar{x}(k|t+1),\bar{u}(k)) - l(\bar{x}'(k|t),\bar{u}'(k))| \quad \leq \quad L_l L_f^{k-t-1}\gamma,\ k \in [t+1,....,t+N_c(t)-1]$$

Therefore, an upper bound for the first part of the summation is given by

$$\sum_{k=t+1}^{t+N_c(t)-1} \{l(\bar{x}(k|t+1),\bar{u}(k)) - l(\bar{x}'(k|t),\bar{u}'(k))\} \quad \leq \quad L_l \frac{L_f^{N_c(t)-1}-1}{L_f-1}\gamma. \tag{28}$$

For $k > t+N_c(t)$, where $\bar{u}'$ and $\bar{u}$ are obtained applying the auxiliary control law to $\bar{x}(k|t+1)$ and $\bar{x}'(k|t)$ respectively, the upper bound is obtained using Assumptions 3 and 7, $l(\bar{x}(k|t+1),\bar{u}(k|t+1)) - l(\bar{x}'(k|t),\bar{u}'(k|t)) \leq (L_l + L_{lu}L_\kappa)|\bar{x}(k|t+1) - \bar{x}'(k|t)|$ and Assumption 2, $|\bar{x}(k+1|t+1) - \bar{x}'(k+1|t)| \leq (L_f + L_{fu}L_\kappa)|\bar{x}(k|t+1) - \bar{x}'(k|t)|$.Moreover $|\bar{x}(t+N_c(t)|t+1) - \bar{x}'(t+N_c(t)|t)| \leq L_f^{N_c(t)-1}\gamma$ and defining $L_x \triangleq (L_f + L_{fu}L_\kappa)$ and $L_{lx} \triangleq (L_l + L_{lu}L_\kappa)$, the following upper bound is obtained

$$\sum_{k=t+N_c(t)}^{t+N_p} \{l(\bar{x}(k|t+1),\bar{u}(k|t+1)) - l(\bar{x}'(k|t),\bar{u}'(k|t))\} \leq L_{lx} \sum_{k=t+N_c(t)}^{t+N_p} |\bar{x}(k|t+1) - \bar{x}'(k|t)|$$

$$\begin{aligned} & \leq \quad L_{lx} \sum_{k=t+N_c(t)}^{t+N_p} L_x^{k-t-N_c(t)} |\bar{x}(t+N_c(t)|t+1) - \bar{x}'(t+N_c(t)|t)| \\ & \leq \quad L_{lx} L_f^{N_c(t)-1} \frac{L_x^{N_p-N_c(t)+1}-1}{L_x-1}\gamma. \end{aligned}$$

Finally in order to compute an upper bound for the difference of terminal penalties note that $|\bar{x}(t+N_p+1|t+1) - \bar{x}'(t+N_p+1|t)| \leq L_f^{N_c(t)-1}L_x^{N_p-N_c(t)+1}\gamma$ and using point 6 of Assumption 4, $V_f(\bar{x}(t+N_p+1|t+1)) - V_f(\bar{x}'(t+N_p+1|t)) \leq L_v L_f^{N_c(t)-1}L_x^{N_p-N_c(t)+1}\gamma$. Therefore the following bound is obtained

$$\begin{aligned} J(x(t+1)) - \bar{J}'(\hat{x}(t+1|t)) \quad \leq \quad & L_l \frac{L_f^{N_c(t)-1}-1}{L_f-1}\gamma + L_{lx} L_f^{N_c(t)-1} \frac{L_x^{N_p-N_c(t)+1}-1}{L_x-1}\gamma \\ & + L_v L_f^{N_c(t)-1} L_x^{N_p-N_c(t)+1}\gamma. \end{aligned}$$

Defining

$$L_J(t) \triangleq L_l \frac{L_f^{N_c(t)-1}-1}{L_f-1} + L_{lx} L_f^{N_c(t)-1} \frac{L_x^{N_p-N_c(t)+1}-1}{L_x-1} + L_v L_f^{N_c(t)-1} L_x^{N_p-N_c(t)+1}$$

it follows that $\bar{J}(x(t+1)) \leq \bar{J}'(\hat{x}(t+1|t)) + L_J(t)\gamma$. Considering that $J^*(t+1, x(t+1))$ is the optimal solution at time $t+1$, $J^*(t+1, x(t+1)) \leq \bar{J}(x(t+1)) \leq \bar{J}'(\hat{x}(t+1|t)) + L_J(t)\gamma$. From (27) it is possible to conclude $J^*(t+1, x(t+1)) - J^*(t, x(t)) \leq -l(x(t), \kappa^{MPC}(t, x(t))) + L_J(t)\gamma$. and by Assumption 3

$$J^*(t+1, x(t+1)) - J^*(t, x(t)) \leq -\alpha_l(|x(t)|) + L_J(t)\gamma. \tag{29}$$

Now, since $V(x(t), M, N_p) = J^*(t, x(t))$, $\forall t \in \mathcal{T}_M$, using (29), there is

$$\begin{aligned} V(x(t+M), M, N_p) - V(x(t), M, N_p) &\leq \sum_{k=t}^{t+M-1} -\alpha_l(|x(k)|) + L_J(k)\gamma \\ &\leq -\alpha_l(|x(t)|) + \sum_{k=t}^{t+M-1} L_J(k)\gamma. \end{aligned} \tag{30}$$

Therefore, by (25), (26) and (30), $V(x, M, N_p)$ is an ISS-Lyapunov function for the closed-loop system (16) and hence, the closed-loop system is ISS with robust invariant region $X_M^{MPC}(N_p)$. Now, to conclude the proof, it is necessary to demonstrate that, for $t \notin \mathcal{T}_M$, the system (15) is ISS in $X^{MPC}(t, N_p)$. Since the model predictive control law (14) is admissible for the $FHOCP^2$, the closed-loop system (15) is such that $\xi_1(t+nM) \in \Phi$, $\forall t \in \mathcal{T}_M$, $\forall n \in \mathcal{Z}_{>0}$. Hence, in order to prove that the system (15) is ISS in $X^{MPC}(t, N_p)$, it is sufficient to prove that the system (15) is ISS in Φ.
Noting that

$$\begin{aligned} \alpha_l(|x(t+i)|) \leq J^*(t+i, x(t+i)) &\leq V(x(t), M, N_p) - \alpha_l(|x(t)|) + \sum_{k=t}^{t+i-1} L_J(k)\gamma \\ &\leq \beta_{V_f}(|x(t)|) + \sum_{k=t}^{t+i-1} L_J(k)\gamma,\ \forall x(t) \in \Phi \end{aligned}$$

considering that for any $\mathcal{K}_\infty$-function γ, $\gamma(r+s) \leq \gamma(2r) + \gamma(2s)$, there is

$$|x(t+Mn+i)| \leq \alpha_l^{-1}(2\beta_{V_f}(|x(t+Mn)|)) + \alpha_l^{-1}(2\sum_{k=t}^{t+i-1} L_J(k)\gamma),\ \forall x(t) \in \Phi \tag{31}$$

for all $n \in \mathcal{Z}_{>0}$ and $i \in [0, \cdots, M-1]$. Since the closed-loop system (16) is ISS with robust invariant region $X_M^{MPC}(N_p)$, there exist a $\mathcal{KL}$-function $\beta(\cdot,\cdot)$, and a $\mathcal{K}_\infty$-function λ such that $|x(t+Mn)| \leq \beta(|x(t)|, n) + \lambda(\gamma)$, $\forall n \in \mathcal{Z}_{\geq 0}, \forall x(t) \in X_M^{MPC}(N_p)$. Applying this to (31), there is $|x(t+Mn+i)| \leq \tilde{\beta}(|x(t)|, n) + \tilde{\lambda}(\gamma)$, $\forall x(t) \in \Phi$. Hence, in conclusion, the system (15) is ISS in $X^{MPC}(t, N_p)$. ■

9. References

Angeli, D., Sontag, E. D. & Wang, Y. (2000). A characterization of integral input-to-state stability, *IEEE Trans. on Automatic Control* **45**: 1082–1097.

Camacho, E. F. & Bordons, C. (2004). *Model Predictive Control*, Springer.

Chen, H. & Allgöwer, F. (1998). A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability, *Automatica* **34**: 1205–1217.

Chen, H., Scherer, C. W. & Allgöwer, F. (1997). A game theoretical approach to nonlinear robust receding horizon control of constrained systems, *American Control Conference '97*.

De Nicolao, G., Magni, L. & Scattolini, R. (1998). Stabilizing receding-horizon control of nonlinear time-varying systems, *IEEE Trans. on Automatic Control* **AC-43**: 1030–1036.

de Oliveira Kothare, S. L. & Morari, M. (2000). Contractive model predictive control for constrained nonlinear systems, *IEEE Trans. on Automatic Control* pp. 1053–1071.

Fontes, F. A. C. C. & Magni, L. (2003). Min-max model predictive control of nonlinear systems using discontinuous feedbacks, *IEEE Trans. on Automatic Control* **48**: 1750–1755.

Franco, E., Magni, L., Parisini, T., Polycarpou, M. & Raimondo, D. M. (2008). Cooperative constrained control of distributed agents with nonlinear dynamics and delayed information exchange: A stabilizing receding-horizon approach, *IEEE Trans. on Automatic Control* **53**(1): 324–338.

Gao, K. & Lin, Y. (2000). On equivalent notions of input-to-state stability for nonlinear discrete time systems, *IASTED International Conference on Control and Applications, Cancun, Mexico*, pp. 81–86.

Grimm, G., Messina, M. J., Tuna, S. E. & Teel, A. R. (2004). Examples when nonlinear model predictive control is nonrobust, *Automatica* **40**: 1729–1738.

Huang, S., James, M. R., Nesic, D. & Dower, P. M. (2005). A unified approach to controller design for achieving ISS and related properties, *IEEE Trans. on Automatic Control* **50**: 1681–1697.

Jiang, Z.-P., Teel, A. R. & Praly, L. (1994). Small-gain theorem for ISS systems and applications, *Mathematics of Control, Signals, and Systems* **7**: 95–120.

Jiang, Z.-P. & Wang, Y. (2001). Input-to-state stability for discrete-time nonlinear systems, *Automatica* **37**: 857–869.

Keerthi, S. S. & Gilbert, E. G. (1988). Optimal, infinite-horizon feedback laws for a general class of constrained discrete-time systems, *J. Optimiz. Th. Appl.* **57**: 265–293.

Limon, D. (2002). *Predictive control of constrained nonlinear systems: stability and robustness*, PhD thesis, Universidad de Sevilla. In spanish.

Limon, D., Alamo, T. & Camacho, E. F. (2002a). Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties, *IEEE CDC*, pp. 4619–4624.

Limon, D., Alamo, T. & Camacho, E. F. (2002b). Stability analysis of systems with bounded additive uncertainties based on invariant sets: Stability and feasibility of MPC, *ACC02*, pp. 364–369.

Limon, D., Alamo, T., Raimondo, D. M., Muñoz de la Peña, D., Bravo, J. M., Ferramosca, A. & F. Camacho, E. (2009). Input-to-state stability: a unifying framework for robust model predictive control, *in* L. Magni, D. M. Raimondo & F. Allgöwer (eds), *Nonlinear Model Predictive Control: Towards New Challenging Applications*, Springer-Verlag, pp. 1–26.

Limon, D., Alamo, T., Salas, F. & Camacho, E. F. (2006). Input to state stability of min-max MPC controllers for nonlinear systems with bounded uncertainties, *Automatica* **42**: 797–803.

Magni, L., De Nicolao, G., Magnani, L. & Scattolini, R. (2001). A stabilizing model-based predictive control for nonlinear systems, *Automatica* **37**: 1351–1362.

Magni, L., De Nicolao, G., Scattolini, R. & F.Allgöwer (2003). Robust model predictive control of nonlinear discrete-time systems, *International Journal of Robust and Nonlinear Control* **13**: 229–246.

Magni, L., Nijmeijer, H. & van der Schaft, A. J. (2001). A receding-horizon approach to the nonlinear H_∞ control problem, *Automatica* **37**: 429–435.

Magni, L., Raimondo, D. M. & Allgöwer, F. (eds) (2009). *Nonlinear Model Predictive Control: Towards New Challenging Applications*, Springer-Verlag.

Magni, L., Raimondo, D. M. & Scattolini, R. (2006). Regional input-to-state stability for nonlinear model predictive control, *IEEE Trans. on Automatic Control* **51**: 1548–1553.

Magni, L. & Scattolini, R. (2005). Control design for nonlinear systems: Trading robustness and performance with the model predictive control approach, *IEE Proceedings - Control Theory & Application* pp. 333–339.

Magni, L. & Scattolini, R. (2007). Robustness and robust design of MPC for nonlinear discrete-time systems, *R. Findeisen et al. (Eds.): Assessment and Future Directions*, Vol. LNCIS 358, Springer-Verlag Berlin Heidelberg, pp. 239–254.

Mayne, D. Q. (2000). Nonlinear model predictive control: Challenges and opportunities, *in* F. Allgöwer & A. Zheng (eds), *Nonlinear Model Predictive Control*, Progress in Systems and Control Theory, Birkhauser Verlag, pp. 23–44.

Mayne, D. Q. & Michalska, H. (1990). Receding horizon control of nonlinear systems, *IEEE Trans. on Automatic Control* **35**: 814–824.

Mayne, D. Q., Rawlings, J. B., Rao, C. V. & Scokaert, P. O. M. (2000). Constrained model predictive control: Stability and optimality, *Automatica* **36**: 789–814.

Michalska, H. & Mayne, D. Q. (1993). Robust receding horizon control of constrained nonlinear systems, *IEEE Trans. on Automatic Control* **38**: 1623–1633.

Nešić, D. & Laila, D. S. (2002). A note on input-to-state stabilization for nonlinear sampled-data systems, *IEEE Transactions on Automatic Control* **47**: 1153–1158.

Raimondo, D. M., Limon, D., Lazar, M., Magni, L. & Camacho, E. F. (2009). Min-max model predictive control of nonlinear systems: a unifying overview on stability, *European Journal of Control* **15**(1): 5–21.

Raimondo, D. M. & Magni, L. (2006). A robust model predictive control algorithm for nonlinear systems with low computational burden, *IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems*, Grenoble, France.

Rawlings, J. B. & Mayne, D. Q. (2009). *Model Predictive Control: Theory and Design*, Nob Hill Publishing.

Scokaert, P. O. M., Rawlings, J. B. & Meadows, E. S. (1997). Discrete-time stability with perturbations: Application to model predictive control, *Automatica* **33**: 463–470.

Sontag, E. D. (1989). Smooth stabilization implies coprime factorization, *IEEE Trans. on Automatic Control* **34**: 435–443.

Sontag, E. D. & Wang, Y. (1995). On characterizations of the input-to-state stability property, *System & Control Letters* **24**: 351–359.

Sontag, E. D. & Wang, Y. (1996). New characterizations of input-to state-stability, *IEEE Trans. on Automatic Control* **41**: 1283–1294.

5

Model predictive control of nonlinear processes

Ch. Venkateswarlu
Indian Institute of Chemical Technology

1. Historical background

Process control has become an integral part of process plants. An automatic controller must be able to facilitate the plant operation over a wide range of operating conditions. The proportional-integral (PI) or proportional-integral-derivative (PID) controllers are commonly used in many industrial control systems. These controllers are tuned with different tuning techniques to deliver satisfactory plant performance.

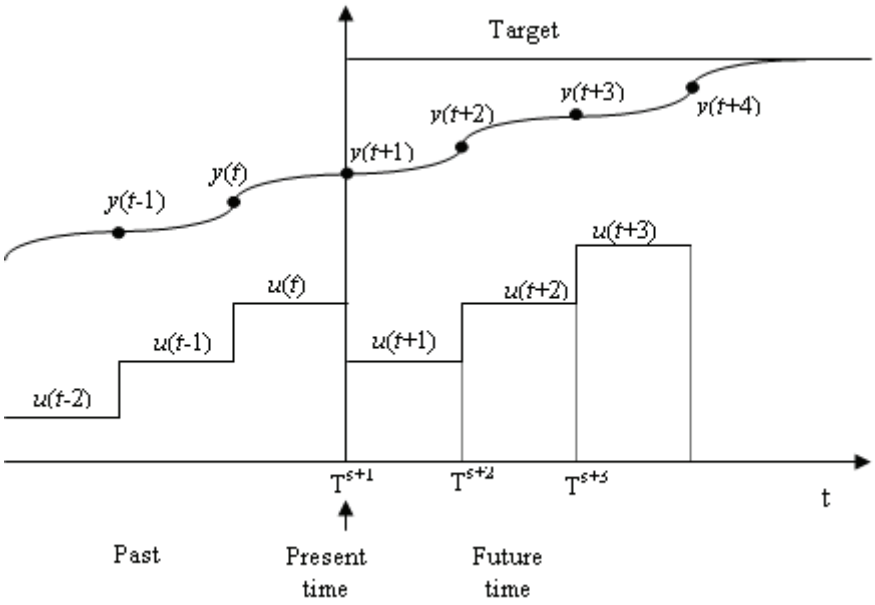


Fig. 1. MPC multi-step prediction scheme.

However, specific control problems associated with the plant operations severely limit the performance of conventional controllers. The increasing complexity of plant operations

together with tougher environmental regulations, rigorous safety codes and rapidly changing economic situations demand the need for more sophisticated process controllers.
Model predictive control (MPC) is an important branch of automatic control theory. MPC refers to a class of control algorithms in which a process model is used to predict and optimize the process performance. MPC has been widely applied in industry (Qin and Badgwell, 1997). The idea of MPC is to calculate a control function for the future time in order to force the controlled system response to reach the reference value. Therefore, the future reference values are to be known and the system behavior must be predictable by an appropriate model. The controller determines a manipulated variable profile that optimizes some open-loop performance objective over a finite horizon extending from the current time into the future. This manipulated variable profile is implemented until a plant measurement becomes available. Feedback is incorporated by using the measurement to update the optimization problem for the next time step. Figure 1 explains the basic idea of MPC showing how the past input-output information is used to predict the future process behavior at the current time and how this information is extended to future to track the desired setpoint trajectory. The notation y, u and T_s refer process output, control action and sample time, respectively.

2. Model predictive control scheme

Model predictive control (MPC) refers to a wide class of control algorithms that use an explicit process model to predict the behavior of a plant. The most significant feature that distinguishes MPC from other controllers is its long range prediction concept. This concept enables MPC to perform current computations to account the future dynamics, thus facilitating it to overcome the limitations of process dead time, non-minimum phase behavior and slow dynamics. In addition, MPC exhibits superior performance by systematically handling constraints violation.

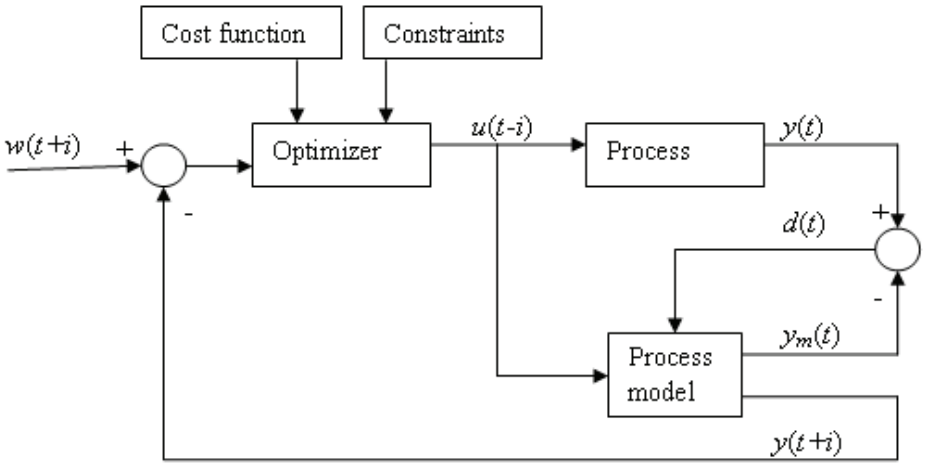


Fig. 2. MPC block diagram.

The fundamental framework of MPC algorithms is common for any kind of MPC schemes. The main differences in many MPC algorithms are the types models used to represent the plant dynamics and the cost function to be minimized. The multi-step model predictive control scheme shown in Figure 1 can be realized from the block diagram represented in Figure 2.

The basic elements in the block diagram are defined as follows. An appropriate model is used to predict the process outputs, $y(t+i), i=1,....,N$ over a future time interval known as prediction horizon, N. A sequence of control actions, $\Delta u(t+j)$, j=1,…., m over the control horizon m are calculated by minimizing some specified objective which is a function of predicted outputs, $y(t+i)$, set-point values, $w(t+i)$ and control actions, $\Delta u(t)$. The first control move, $u(t)$ of the sequence is implemented and the calculations are repeated for the subsequent sampling instants. In order to account the plant-model mismatch, a prediction error, $d(t)$, that is calculated based on plant measurement, $y(t)$ and model prediction, $y_m(t)$ is used to update the future predictions.

In MPC, the control law generates a control sequence, which forces the future system response to be equal to the reference values. The system response is based on future control actions, model parameters and the actual system states. Many methods for updating the optimization problem are possible, such as estimating model parameters and/or states, inferring about disturbances etc. MPC design considers different types of process models. These include first principle models, auto regressive moving average models, polynomial models, neural network models, fuzzy models etc. The attraction for MPC is due to its capability of handling various constraints directly in the formulation through on-line optimization. A variety of model predictive control techniques have been reported for controlling the processes of various complexities.

This chapter presents different linear and nonlinear model predictive controllers with case studies illustrating their application to real processes.

3. Linear model predictive control

Linear MPC (LMPC) algorithms employ linear or linearized models to obtain the predictive response of the controlled process. These algorithms include the Model Algorithmic Control (MAC) (Richalet et al., 1978), the Dynamic Matrix Control (DMC) (Cutler and Ramaker, 1980) and the Generalized Predictive Control (GPC) (Clarke et al., 1987). These algorithms are all similar in the sense that they rely on process models to predict the behavior of the process over some future time interval, and the control calculations are based on these model predictions. The models used for these predictions have usually been derived from linear approximations of the process or experimentally obtained step response data. A survey of theory and applications of such algorithms have been reported by Garcia et al. (1989).

3.1 LMPC design

A classical autoregressive moving average (ARX) model structure that relates the plant output with the present and past plant input-output can be used to formulate a predictive model. The model parameters can be determined a priori by using the known input-output data to form a fixed predictive model or these parameters are updated at each sampling

time by an adaptive mechanism. The one step ahead predictive model can be recursively extended to obtain future predictions for the plant output. The minimization of a cost function based on future plant predictions and desired plant outputs generates an optimal control input sequence to act on the plant. The strategy is described as follows.

Predictive model

The relation between the past input-output data and the predicted output can be expressed by an ARX model of the form

$$y(t+1) = a_1 y(t) + \ldots + a_{ny} y(t-ny+1) + b_1 u(t) + \ldots\ldots + b_{nu} u(t-nu+1) \tag{1}$$

where $y(t)$ and $u(t)$ are the process and controller outputs at time t, $y(t+1)$ is the one-step ahead model prediction at time t, a's and b's represent the model coefficients and the nu and ny are input and output orders of the system.

Model identification

The model output prediction can be expressed as

$$y_m(t+1) = \phi\, x_m(t) \tag{2}$$

where

$$\phi = [\alpha_1 \ldots \alpha_{ny}\ \beta_1 \ldots \beta_{nu}] \tag{3}$$

and

$$x_m(t) = [y(t) \ldots y(t-ny+1)\ \ u(t) \ldots u(t-nu+1)]^T \tag{4}$$

with α and β as identified model parameters.

One of the most widely used estimators for model parameters and covariance is the popular recursive least squares (RLS) algorithm (Goodwin and Sin, 1984). The RLS algorithm provides the updated parameters of the ARX model in the operating space at each sampling instant or these parameters can be determined a priori using the known data of inputs and outputs for different operating conditions. The RLS algorithm is expressed as

$$\begin{aligned} \phi(t+1) &= \phi(t) + K(t)\,[y(t+1) - y_m(t+1)] \\ K(t) &= P(t)\, x_m(t+1) \,/\, [1 + x_m(t+1)^T P(t)\, x_m(t+1)] \\ P(t+1) &= 1/\lambda\, [P(t) - \{(P(t)\, x_m(t+1)\, x_m(t+1)^T P(t)) \,/\, (1 + x_m(t+1)^T P(t)\, x_m(t+1))\}] \end{aligned} \tag{5}$$

where $\phi(t)$ represents the estimated parameter vector, λ is the forgetting factor, $K(t)$ is the gain matrix and $P(t)$ is the covariance matrix.

Controller formulation

The N time steps ahead output prediction over a prediction horizon is given by

$$y_p(t+N) = \alpha_1\ y(t+N-1)+\ldots+\alpha_{ny} y(t-ny+N)+\beta_1 u(t+N-1)+\ldots+\beta_{nu} u(t-nu+N)+err(t) \tag{6}$$

where $y_p(t+N)$ represent the model predictions for N steps and $err(t)$ is an estimate of the modeling error which is assumed as constant for the entire prediction horizon. If the control horizon is m, then the controller output, u after m time steps can be assumed to be constant.

An internal model is used to eliminate the discrepancy between model and process outputs, *error*(*t*), at each sampling instant

$$error(t) = y(t) - y_m(t) \tag{7}$$

where $y_m(t)$ is the one-step ahead model prediction at time (t-1). The estimate of the error is then filtered to produce *err*(*t*) which minimizes the instability introduced by the modeling error feedback. The filter error is given by

$$err(t) = (1\text{-}K_f)\, err(t\text{-}1) + K_f\, error(t) \tag{8}$$

where K_f is the feedback filter gain which has to be tuned heuristically.
Back substitutions transform the prediction model equations into the following form

$$\begin{aligned} & y_p(t+N) = f_{N,1}y(t) + ... + \\ & f_{N,ny}y(t-ny+1) + f_{N,ny+1}u(t-1) + \\ & .....+ f_{N,ny+nu-1}u(t-nu+1) + \\ & g_{N,1}u(t) + ... + g_{N,m}u(t+m-1) \\ & + e_N err(t) \end{aligned} \tag{9}$$

The elements *f*, *g* and *e* are recursively calculated using the parameters α and β of Eq. (3). The above equations can be written in a condensed form as

$$Y(t) = F\,X(t) + G\,U(t) + E\,err(t) \tag{10}$$

where

$$Y(t) = [y_p(t\text{+}1) \ldots y_p(t\text{+}N)]^T \tag{11}$$

$$X(t) = [y(t)\ \ y(t\text{-}1) \ldots y(t\text{-}ny\text{+}1)\ \ u(t\text{-}1) \ldots u(t\text{-}nu\text{+}1)]^T \tag{12}$$

$$U(t) = [u(t) \ldots u(t\text{+}m\text{-}1)]^T \tag{13}$$

$$F = \begin{bmatrix} f_{11}\ f_{12}\ \cdots\cdots f_{1(ny+nu-1)} \\ f_{11}\ f_{12}\ \cdots\cdots f_{2(ny+nu-1)} \\ \vdots \\ \vdots \\ f_{N1}\ f_{N12}\ \cdots\cdots f_{N(ny+nu-1)} \end{bmatrix}$$

$$G = \begin{bmatrix} g_{11} & 0 & 0 & \ldots & 0 \\ g_{21} & g_{21} & 0 & \ldots & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ g_{m1} & g_{m2} & g_{m3} & \cdots & g_{mm} \\ . & . & . & \ldots & . \\ . & . & . & \ldots & . \\ . & . & . & \ldots & . \\ g_{N1} & g_{N2} & g_{N3} & \cdots & g_{Nm} \end{bmatrix}$$

$$E = [e_1 \ldots e_N]^T$$

In the above, $Y(t)$ represents the model predictions over the prediction horizon, $X(t)$ is a vector of past plant and controller outputs and $U(t)$ is a vector of future controller outputs. If the coefficients of F, G and E are determined then the transformation can be completed. The number of columns in F is determined by the ARX model structure used to represent the system, where as the number of columns in G is determined by the length of the control horizon. The number of rows is fixed by the length of the prediction horizon.

Consider a cost function of the form

$$J = \sum_{i=1}^{N} \left[y_p(t+i) - w(t+i) \right]^2 + \sum_{i=1}^{m} \gamma \left[u(t+i-1 \right]^2$$

$$= \sum_{i=1}^{N} \left[Y(t) - W(t) \right]^T \left[Y(t) - W(t) \right] + \sum_{i=1}^{m} \gamma U(t)^T U(t) \tag{14}$$

where $W(t)$ is a setpoint vector over the prediction horizon

$$W(t) = [\, w(t+1) \ldots . w(t+N)]^T \tag{15}$$

The minimization of the cost function, J gives optimal controller output sequence

$$U(t) = [G^T G + \gamma I\,]^{-1} G^T [W(t) - FX(t) - Eerr(t)] \tag{16}$$

The vector $U(t)$ generates control sequence over the entire control horizon. But, the first component of $U(t)$ is actually implemented and the whole procedure is repeated again at the next sampling instant using latest measured information.
Linear model predictive control involving input-output models in classical, adaptive or fuzzy forms is proved useful for controlling processes that exhibit even some degree of nonlinear behavior (Eaton and Rawlings, 1992; Venkateswarlu and Gangiah, 1997 ; Venkateswarlu and Naidu, 2001).

3.2 Case study: linear model predictive control of a reactive distillation column

In this study, a multistep linear model predictive control (LMPC) strategy based on autoregressive moving average (ARX) model structure is presented for the control of a reactive distillation column. Although MPC has been proved useful for a variety of chemical and biochemical processes (Garcia et al., 1989 ; Eaton and Rawlings, 1992), its application to a complex dynamic system like reactive distillation is more interesting.

The process and the model

Ethyl acetate is produced through an esterfication reaction between acetic acid and ethyl alcohol

$$CH_3COOH + C_2H_5OH \xleftrightarrow{H+} H_2O + CH_3COOC_2H_5 \tag{17}$$

The achievable conversion in this reversible reaction is limited by the equilibrium conversion. This quaternary system is highly non-ideal and forms binary and ternary

azeotropes, which introduce complexity to the separation by conventional distillation. Reactive distillation can provide a means of breaking the azeotropes by altering or eliminating the conditions for azeotrope formation. Thus reactive distillation becomes attractive alternative for the production of ethyl acetate.
The rate equation of this reversible reaction in the presence of a homogeneous acid catalyst is given by (Alejski and Duprat, 1996)

$$\begin{aligned} r_1 &= k_1 C_1 C_2 - \frac{k_1}{K_c} C_3 C_4 \\ k_1 &= (4.195 C_k + 0.08815) \exp(-6500.1/T) \\ K_c &= 7.558 - 0.012T \end{aligned} \tag{18}$$

Vora and Daoutidis (2001) have presented a two feed column configuration for ethyl acetate reactive distillation and found that by feeding the two reactants, ethanol and acetic acid, on different trays counter currently allows to enhance the forward reaction on trays and results in higher conversion and purity over the conventional column configuration of feeding the reactants on a single tray. All plates in the column are considered to be reactive. The column consists of 13 stages including the reboiler and the condenser. The less volatile acetic acid enters the 3 rd tray and the more volatile ethanol enters the 10 th tray. The steady state operating conditions of the column are shown in Table 1.

Acetic acid feed flow rate, F_{Ac}	6.9 mol/s
Ethanol flow rate, F_{Eth}	6.865 mol/s
Reflux flow rate, L_o	13.51 mol/s
Distillate flow rate, D	6.68 mol/s
Bottoms flow rate, B	7.085 mol/s
Reboiler heat duty, Q_r	5.88 x 10^5 J/mol
Boiling points, ^oK	391.05, 351.45, 373.15, 350.25
(Acetic acid, ethanol, water, ethyl acetate)	
Distillate composition	0.0842, 0.1349, 0.0982, 0.6827
(Acetic acid, ethanol, water, ethyl acetate)	
Bottoms composition	0.1650, 0.1575, 0.5470, 0.1306
(Acetic acid, ethanol, water, ethyl acetate)	

Table 1. Design conditions for ethyl acetate reactive distillation column

The dynamic model representing the process operation involves mass and component balance equations with reaction terms, along with energy equations supported by vapor-liquid equilibrium and physical properties (Alejski & Duprat, 1996). The assumptions made in the formulation of the model include adiabatic column operation, negligible heat of reaction, negligible vapor holdup, liquid phase reaction, physical equilibrium in streams leaving each stage, negligible down comer dynamics and negligible weeping of liquid through the openings on the tray surface. The equations representing the process are given as follows.

Total mass balance

Total condenser:

$$\frac{dM_1}{dt} = V_2 - (L_1 + D) + \Delta R_1 \tag{19}$$

Plate j:

$$\frac{dM_j}{dt} = FV_j + V_{j+1} + FL_j + L_{j-1} - V_j - L_j + \Delta R_j \tag{20}$$

Reboiler :

$$\frac{dM_n}{dt} = L_{n-1} - V_n - L_n + \Delta R_n \tag{21}$$

Component mass balance

Total condenser :

$$\frac{d(M_1 x_{i,1})}{dt} = V_2 y_{i,2} - (L_1 + D)x_{i,1} + R_{i,1} \tag{22}$$

Plate j:

$$\frac{d(M_j x_{i,j})}{dt} = FV_j yf_{i,j} + V_{j+1} y_{i,j+1} + FL_j xf_{i,j} + L_{j-1} x_{i,j-1} - V_j y_{i,j} - L_j x_{i,j} + R_{i,j} \tag{23}$$

Reboiler:

$$\frac{d(M_n x_{i,n})}{dt} = L_{n-1} x_{i,n-1} - V_n y_{i,n} - L_n x_{i,n} + R_{i,n} \tag{24}$$

Total energy balance

Total condenser :

$$\frac{dE_1}{dt} = V_2 H_2 - (L_1 + D)h_1 + Q_1 \tag{25}$$

Plate j:

$$\frac{dE_j}{dt} = FV_j Hf_j + V_{j+1} H_{j+1} + FL_j hf_j + L_{j-1} h_{j-1} - V_j H_j - L_j h_j + Q_j \tag{26}$$

Reboiler:

$$\frac{dE_n}{dt} = L_{n-1} h_{n-1} - V_n H_n - L_n h_n + Q_n \tag{27}$$

Level of liquid on the tray

$$L^{liq} = \frac{M_n MW_{av}}{A_{tray} \rho_{av}} \tag{28}$$

Flow of liquid over the weir

$$\text{If } (L^{liq} < h_{weir}) \text{ then } L_n = 0 \tag{29}$$

else

$$L_n = 1.84 \frac{L_{weir}\rho_{av}}{MW_{av}} (L^{liq} - h_{weir})^{3/2} \tag{30}$$

Mole fraction normalization

$$\sum_{i=1}^{NC} x_i = \sum_{i=1}^{NC} y_i = 1 \tag{31}$$

VLE calculations

For the column operation under moderate pressures, the VLE equation assumes the ideal gas model for the vapor phase, thus making the vapor phase activity coefficient equal to unity. The VLE relation is given by

$$y_i P = x_i \gamma_i P_i^{sat} \qquad (i = 1,2,\ldots,NC) \tag{32}$$

The liquid phase activity coefficients are calculated using UNIFAC method (Smith et al., 1996).

Enthalpies Calculation
The relations for the liquid enthalpy h, the vapor enthalpy H and the liquid density ρ are:

$$\begin{aligned} h &= h(P, T, x) \\ H &= H(P, T, y) \\ \rho^{liq} &= \rho^{liq}(P, T, x) \end{aligned} \tag{33}$$

Control scheme

The design and implementation of the control strategy is studied for the single input-single output (SISO) control of the ethyl acetate reactive distillation column with its double feed configuration. The objective is to control the desired product purity in the distillate stream inspite disturbances in column operation. This becomes the main control loop. Since reboiler and condenser holdups act as pure integrators, they also need to be controlled. These become the auxiliary control loops. Reflux flow rate is used as a manipulated variable to control the purity of the ethyl acetate. Distillate flow rate is used as a manipulated variable to control the condenser holdup, while bottom flow rate is used to control the reboiler holdup. In this work, it is proposed to apply a multistep model predictive controller for the main loop and conventional PI controllers for the auxiliary control loops. This control scheme is shown in the Figure 3.

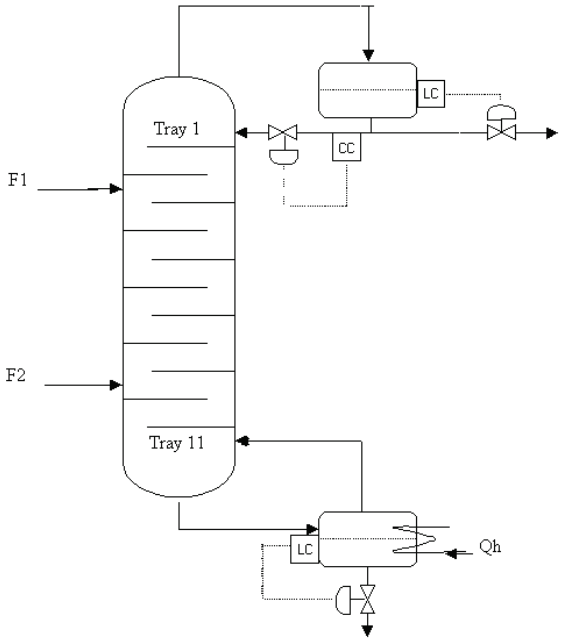


Fig. 3. Control structure of two feed ethyl acetate reactive distillation column.

Analysis of Results

The performance of the multistep linear model predictive controller (LMPC) is evaluated through simulation. The product composition measurements are obtained by solving the model equations using Euler's integration with sampling time of 0.01 s. The input and output orders of the predictive model are considered as n_u = 2 and n_y = 2. The diagonal elements of the initial covariance matrix, $P(0)$ in the RLS algorithm are selected as 10.0, 1.0, 0.01, 0.01, respectively. The forgetting factor, λ used in recursive least squares is chosen as 5.0. The feedback controller gain K_f is assigned as 0.65. The tuning parameter γ in the control law is set as 0.115 x 10^{-6}. The PI controller parameters of ethyl acetate composition are evaluated by using the continuous cycling method of Ziegler and Nichols. The tuned controller settings are k_c = 11.15 and τ_I = 1.61 x 10^4 s. The PI controller parameters used for reflux drum and reboiler holdups are k_c = - 0.001 and τ_I = 5.5 h, and k_c = - 0.001 and τ_I = 5.5 h, respectively (Vora and Daoutidis, 2001).

The LMPC is implemented by adaptively updating the prediction model using recursive least squares. On evaluating the effect of different prediction and control horizons, it is observed that the LMPC with a prediction horizon of around 5 and a control horizon of 2 has shown reasonably better control performance. The LMPC is also referred here as MPC. Figure 4 shows the results of MPC and PI controller when they are applied for tracking series of step changes in ethyl acetate composition. The regulatory control performance of MPC and PI controller for 20% decrease in feed rate of acetic acid is shown in Figure 5. The results thus show the effectiveness of the multistep linear model predictive control strategy for the control of highly nonlinear reactive distillation column.

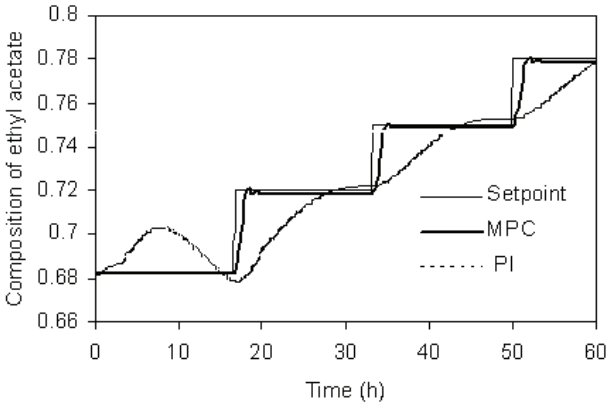


Fig. 4. Performance of MPC and PI controller for tracking series of step changes in distillate composition.

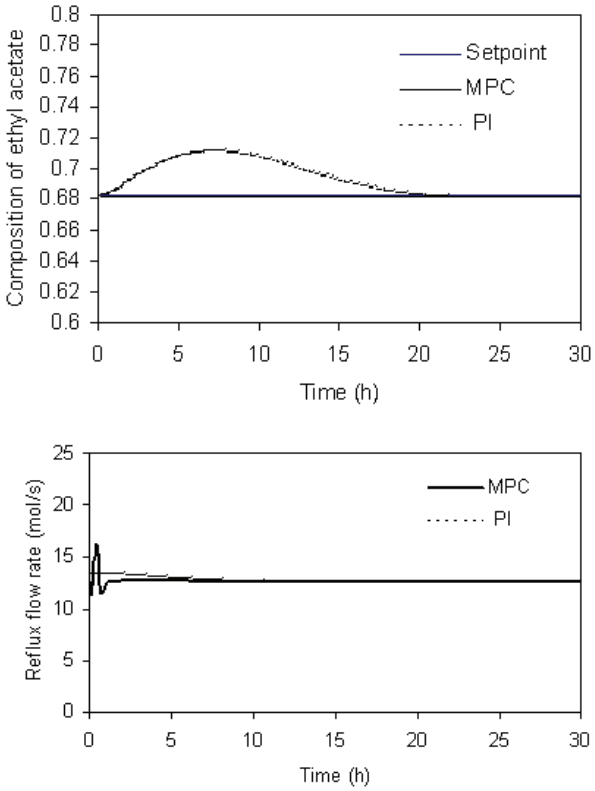


Fig.5. Output and input profiles for MPC and PI controller for 20% decrease in the feed rate of acetic acid.

4. Generalized predictive control

The generalized predictive control (GPC) is a general purpose multi-step predictive control algorithm (Clarke et al., 1987) for stable control of processes with variable parameters, variable dead time and a model order which changes instantaneously. GPC adopts an integrator as a natural consequence of its assumption about the basic plant model. Although GPC is capable of controlling such systems, the control performance of GPC needs to be ascertained if the process constraints are to be encountered in nonlinear processes. Camacho (1993) proposed a constrained generalized predictive controller (CGPC) for linear systems with constrained input and output signals. By this strategy, the optimum values of the future control signals are obtained by transforming the quadratic optimization problem into a linear complementarity problem. Camacho demonstrated the results of the CGPC strategy by carrying out a simulation study on a linear system with pure delay. Clarke et al. (1987) have applied the GPC to open-loop stable unconstrained linear systems. Camacho applied the CGPC to constrained open-loop stable linear system. However, most of the real processes are nonlinear and some processes change behavior over a period of time. Exploring the application of GPC to nonlinear process control will be more interesting.
In this study, a constrained generalized predictive control (CGPC) strategy is presented and applied for the control of highly nonlinear and open-loop unstable processes with multiple steady states. Model parameters are updated at each sampling time by an adaptive mechanism.

4.1 GPC design

A nonlinear plant generally admits a local-linearized model when considering regulation about a particular operating point. A single-input single-output (SISO) plant on linearization can be described by a Controlled Autoregressive Integrated Moving Average (CARIMA) model of the form

$$A(q^{-1})y(t) = B(q^{-1})q^{-d}\,u(t) + C\,(q^{-1})e(t\,)/\Delta \tag{34}$$

where A, B and C are polynomials in the backward shift operator q^{-1}. The $y(t)$ is the measured plant output, $u(t)$ is the controller output, $e(t)$ is the zero mean random Gaussian noise, d is the delay time of the system and Δ is the differencing operator $1\text{-}q^{-1}$.
The control law of GPC is based on the minimization of a multi-step quadratic cost function defined in terms of the sum of squares of the errors between predicted and desired output trajectories with an additional term weighted by projected control increments as given by

$$J(N_1, N_2, N_3) = E\left\{ \sum_{j=N_1}^{N_2} \{[y(t+j|t) - w(t+j)]^2 \; + \sum_{j=1}^{N_3} \lambda[\Delta u(t+j-1)]^2 \right\} \tag{35}$$

where $E\{.\}$ is the expectation operator, $y(t + j \mid t\,)$ is a sequence of predicted outputs, $w(t + j)$ is a sequence of future setpoints, $\Delta u(t + j\text{ -}1)$ is a sequence of predicted control increments and λ is the control weighting factor. The N_1 , N_2 and N_3 are the minimum costing horizon, the maximum costing horizon and the control horizon, respectively. The values of N_1 , N_2 and N_3 of Eq. (35) can be defined by $N_1 = d + 1$, $N_2 = d + N$, and $N_3 = N$, respectively.

Predicting the output response over a finite horizon beyond the dead-time of the process enables the controller to compensate for constant or variable time delays. The recursion of the Diophantine equation is a computationally efficient approach for modifying the predicted output trajectory. An optimum *j*-step a head prediction output is given by

$$y(t+j \mid t) = G_j(q^{-1})\,\Delta u(t+j-d-1) + F_j(q^{-1})y(t) \tag{36}$$

where $G_j(q^{-1}) = E_j(q^{-1})B(q^{-1})$, and E_j and F_j are polynomials obtained recursively solving the Diophantine equation,

$$1 = E_j(q^{-1})A\Delta + q^{-j}F_j(q^{-1}) \tag{37}$$

The *j*-step ahead optimal predictions of y for $j = 1, \ldots, N_2$ can be written in condensed form

$$Y = Gu + f \tag{38}$$

where f contains predictions based on present and past outputs up to time t and past inputs and referred to free response of the system, i.e., $f = [f_1, f_2, \ldots, f_N]$. The vector u corresponds to the present and future increments of the control signal, *i.e.*, $u = [\Delta u(t), \Delta u(t+1), \ldots, \Delta u(t+N-1)]^T$. Eq. (35) can be written as

$$J = \left(Gu + f - w\right)^T \left(Gu + f - w\right) + \lambda\, u^T u \tag{39}$$

The minimization of J gives unconstrained solution to the projected control vector

$$u = (G^T G + \lambda I)^{-1} G^T (w - f) \tag{40}$$

The first component of the vector u is considered as the current control increment $\Delta u(t)$, which is applied to the process and the calculations are repeated at the next sampling instant. The schematic of GPC control law is shown in Figure 6, where K is the first row of the matrix $(G^T G + \lambda I)^{-1} G^T$.

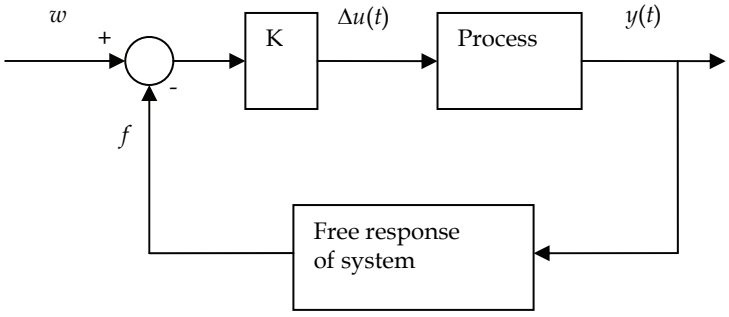


Fig. 6. The GPC control law

4.2 Constrained GPC design

In practice, all processes are subject to constraints. Control valves are limited by fully closed and fully open positions and maximum slew rates. Constructive and safety reasons as well as sensor ranges cause limits in process variables. Moreover, the operating points of plants are determined in order to satisfy economic goals and usually lie at the intersection of certain constraints. Thus, the constraints acting on a process can be manipulated variable limits (u_{min}, u_{max}), slew rate limits of the actuator (du_{min}, du_{max}), and limits on the output signal (y_{min}, y_{max}) as given by

$$\begin{aligned} u_{\min} &\le u(t) \le u_{\max} \\ du_{\min} &\le u(t) - u(t-1) \le du_{\max} \\ y_{\min} &\le y(t) \le y_{\max} \end{aligned} \tag{41}$$

These constraints can be expressed as

$$\begin{aligned} lu_{\min} &\le Tu + u(t-1)l \le lu_{\max} \\ ldu_{\min} &\le u \le ldu_{\max} \\ ly_{\min} &\le Gu + f \le ly_{\max} \end{aligned} \tag{42}$$

where l is an N vector containing ones, and T is an N x N lower triangular matrix containing ones. By defining a new vector $x = u - ldu_{min}$, the constrained equations can be transformed as

$$\begin{aligned} x &\ge 0 \\ Rx &\le c \end{aligned} \tag{43}$$

with

$$R = \begin{bmatrix} I_{NxN} \\ T \\ -T \\ G \\ -G \end{bmatrix} \quad ; \qquad c = \begin{bmatrix} l(du_{\max} - du_{\min}) \\ lu_{\max} - Tldu_{\min} - u(t-1)l \\ -lu_{\min} + Tldu_{\min} + u(t-1)l \\ ly_{\max} - f - Gldu_{\min} \\ -ly_{\min} + f + Gldu_{\min} \end{bmatrix} \tag{44}$$

Eq. (39) can be expressed as

$$J = \frac{1}{2} u^T Hu + bu + f_o \tag{45}$$

where

$$H = 2(G^T G + \lambda I)$$
$$b = G^T (f - w) + (f - w)^T G$$
$$f_o = (f - w)^T (f - w)$$

The minimization of J with no constraints on the control signal gives

$$u = -H^{-1} b \tag{46}$$

Eq. (45) in terms of the newly defined vector x becomes

$$J = \frac{1}{2} x^T H x + ax + f_1 \tag{47}$$

where

$$a = b + u_{\min} l^T H$$
$$f_1 = f_o + \frac{1}{2} d u_{\min}^2 l^T H l + b l d u_{\min}$$

The solution of the problem can be obtained by minimization of Eq. (47) subject to the constraints of Eq. (43). By using the Lagrangian multiplier vectors v_1 and v for the constraints, $x \geq 0$ and $Rx \leq c$, respectively, and introducing the slack variable vector v_2, the Kuhn-Tucker conditions can be expressed as

$$\begin{aligned} & Rx + v_2 = c \\ & -Hx - R^T v + v_1 = a \\ & x^T v_1 = 0 \\ & v^T v_2 = 0 \\ & x, v, v_1, v_2 \geq 0 \end{aligned} \tag{48}$$

Camacho (1993) has proposed the solution of this problem with the help of Lemke's algorithm (Bazaraa and Shetty, 1979) by expressing the Kuhn-Tucker conditions as a linear complementarity problem starting with the following tableau

	v_2	x	v	v_1	z_0	
v_2	I_{mxn}	O_{mxN}	$-RH^{-1}R^T$	RH^{-1}	-1	$v_{2\min}$
x	O_{Nxm}	I_{NxN}	$H^{-1}R^T$	$-H^{-1}$	-1	$x_{\min}$

(49)

Here, z_0 is the artificial variable which will be driven to zero iteratively.

In this study, the above stated constrained generalized predictive linear control of Camacho (1993) is extended to open-loop unstable constrained control of nonlinear processes. In this

strategy, process nonlinearities are accounted through adaptation of model parameters while taking care of input and output constraints acting on the process. The following recursive least squares formula (Hsia, 1977) is used for on-line estimation of parameters and the covariance matrix after each new sample:

$$\begin{aligned}\theta(k+1) &= \theta(k) + \gamma(k+1)P(k)v(k+1)\left[y(k+1) - v^T(k+1)\theta(k)\right] \\ P(k+1) &= \frac{1}{\lambda}\left[P(k) - \gamma(k+1)P(k)v(k+1)v^T(k+1)P(k)\right] \\ \gamma(k+1) &= 1 \Big/ \left[1 + v^T(k+1)P(k)v(k+1)\right]\end{aligned} \tag{50}$$

where θ is the parameter vector, γ is the intermediate estimation variable, P is the covariance matrix, v is the vector of input-output variables, y is the output variable, and $0 < \lambda < 1$ is the forgetting factor. The initial covariance matrix and exponential forgetting factor are selected based on various trials so as to provide reasonable convergence in parameter estimates.

The CGPC strategy of nonlinear processes is described in the following steps:

1. Specify the controller design parameters N_1, N_2, N_3 and also the initial parameter estimates and covariance matrix for recursive identification of model parameters.
2. Update the model parameters using recursive least squares method.
3. Initialize the polynomials E_1 and F_1 of Diophantine identity, Eq. (37), using the estimated parameters. Further initialize G_1 as $E_1 B$.
4. Compute the polynomials E_j , F_j and G_j over the prediction horizon and control horizon using the recursion of Diophantine.
5. Compute matrices H, R, and G, and vectors f and c using the polynomials determined in step 4.
6. Compute the unconstrained solution $x_{min} = - H^{-1} a$.
7. Compute $v_{2min} = c - Rx_{min}$. If x_{min} and v_{2min} are nonnegative, then go to step 10.
8. Start Lemke's algorithm with x and v_2 in the basis with the tableau, Eq. (49).
9. If x_1 is not in the first column of the tableau, make it zero; otherwise, assign it the corresponding value.
10. Compute $u(t) = x_1 + du_{min} + u(t - 1)$.
11. Implement the control action, then shift to the next sampling instant and go to step 2.

4.3 Case study: constrained generalized predictive control (CGPC) of open-loop unstable CSTR

The design and implementation of the CGPC strategy is studied by applying it for the control of a nonlinear open-loop unstable chemical reactor (Venkateswarlu and Gangiah, 1997).

Reactor

A continuous stirred tank reactor (CSTR) in which a first order exothermic irreversible reaction occurs is considered as an example of an unstable nonlinear process. The dynamic equations describing the process can be written as

$$V\frac{dC_A}{dt'} = F(C_{Af} - C_A) - Vk_0 \exp\left[-E / R_g T_r\right] C_A \tag{51}$$

$$V\rho C_p \frac{dT_r}{dt'} = \rho C_p F(T_f - T_r) + V(-\Delta H)k_0 \exp\left[-E / R_g T_r\right] C_A - UA_h(T_r - T_c) \tag{52}$$

where C_A and T_r are reactant concentration and temperature, respectively. The coolant temperature T_c is assumed to be the manipulated variable. Following the analysis of Uppal et al. (1974), the model is made dimensionless by introducing the parameters as

$$\phi = \frac{E}{R_g T_{fo}}, \quad B_h = \frac{(-\Delta H)C_{Afo}}{\rho C_p T_{fo}}, \quad D_a = \frac{k_o e^{-\phi} V}{F_o}, \quad \beta = \frac{UA_h}{F_o \rho C_p} \tag{53}$$

where F_o, C_{Afo} and T_{fo} are the nominal characteristic values of volumetric flow rate, feed composition and feed temperature, respectively. The corresponding dimensionless variables are defined by

$$t = \frac{t'F_o}{V}, \quad x_1 = \frac{C_{Afo} - C_A}{C_{Afo}}, \quad x_2 = \frac{T_r - T_{fo}}{T_{fo}}, \quad u = \frac{T_c - T_{co}}{T_{fo}}\phi \tag{54}$$

where T_{co} is some reference value for the coolant temperature.
The modeling equations can be written in dimensionless form (Calvet and Arkun, 1988; Hernandez and Arkun, 1992) as

$$\dot{x}_1 = -x_1 + Da(1 - x_1)\exp\left(\frac{x_2}{1 + x_2/\phi}\right)$$

$$\dot{x}_2 = -x_2 + B_h Da(1 - x_1)\exp\left(\frac{x_2}{1 + x_2/\phi}\right) + \beta(u - x_2) \tag{55}$$

$$y = x_1$$

where x_1 and x_2 are the dimensionless reactant concentration and temperature, respectively. The input u is the cooling jacket temperature, D_a is the Damkohler number, ϕ is the dimensionless activation energy, B_h is the heat of reaction and β is the heat transfer coefficient. If the physical parameters chosen are $D_a = 0.072$, $\phi = 20.0$, $B_h = 8.0$, and $\beta = 0.3$, then the system can exhibit up to three steady states, one of which is unstable as shown in Figure 7. Here the task is to control the reactor at and around the unstable operating point. The cooling water temperature is the input u, which is the manipulated variable to control the reactant concentration, x_1.

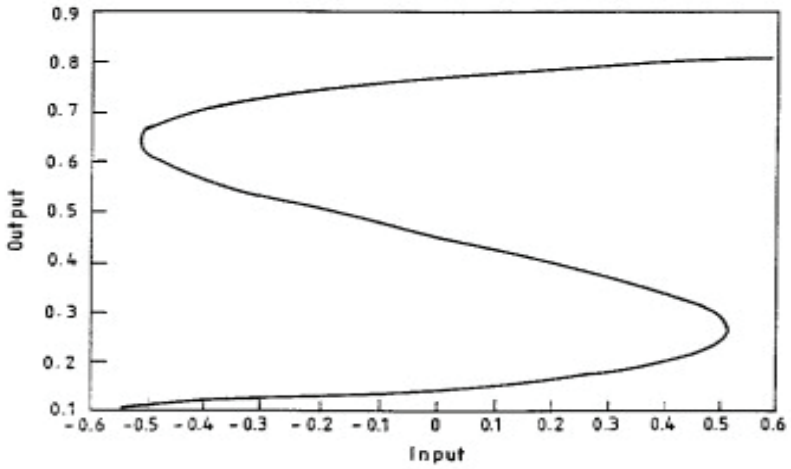


Fig. 7. Steady state output vs. steady state input for CSTR system.

Analysis of Results

Simulation studies are carried out in order to evaluate the performance of the Constrained Generalized Predictive Control (CGPC) strategy. The results of unconstrained Generalized Predictive Control (GPC) are also presented as a reference. The CGPC strategy considers an adaptation mechanism for model parameters.

N_a	2
N_b	2
N_1	2
N_2	7
N_3	6
λ	0.2
u_{min}	-1.0
u_{max}	1.0
du_{min}	-0.5
du_{max}	0.5
y_{min}	0.1
y_{max}	0.5
Forgetting factor	0.95
Initial covariance matrix	$1.0x10^9$
Sample time	0.5

Table 2. Constraints and parameters of CSTR system.

The controller and design parameters as well as the constraints employed for the CSTR system are given in Table 2. The same controller and design parameters are used for both the CGPC and GPC. Two set-point changes are introduced for the output concentration of the system and the corresponding results of CGPC and GPC are analyzed. A step change is

introduced in the output concentration of *CSTR* from a stable equilibrium point (x_1 = 0.2, x_2 = 1.33, u = 0.42) to an unstable operating point (x_1 = 0.5, x_2 = 3.303, u = - 0.2). The input and output responses of both CGPC and GPC are shown in Figure 8. Another step change is introduced for the set-point from a stable operating point (x_1 = 0.144, x_2 = 0.886, u = 0.0) to an unstable operating point (x_1 = 0.445, x_2 = 2.75, u = 0.0). The input and output responses of CGPC and GPC for this case are shown in Figure 9. The results show that for the specified controller and design parameters, CGPC provides better performance over GPC.

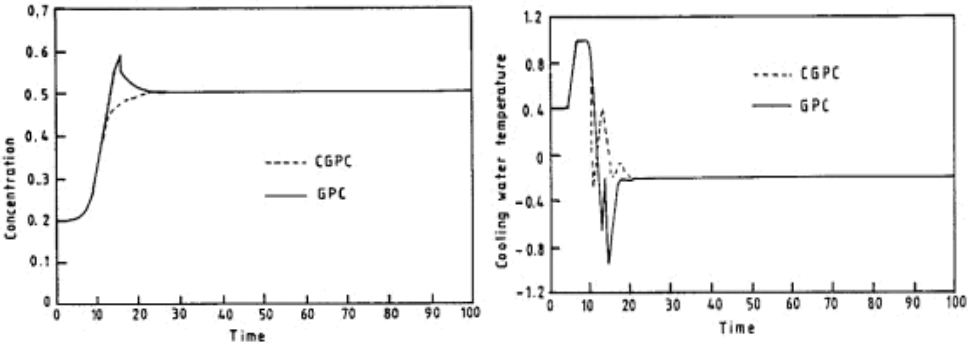


Fig. 8. Cooling water temperature and concentration plots of CSTR for a step change in concentration from 0.20 to 0.50.

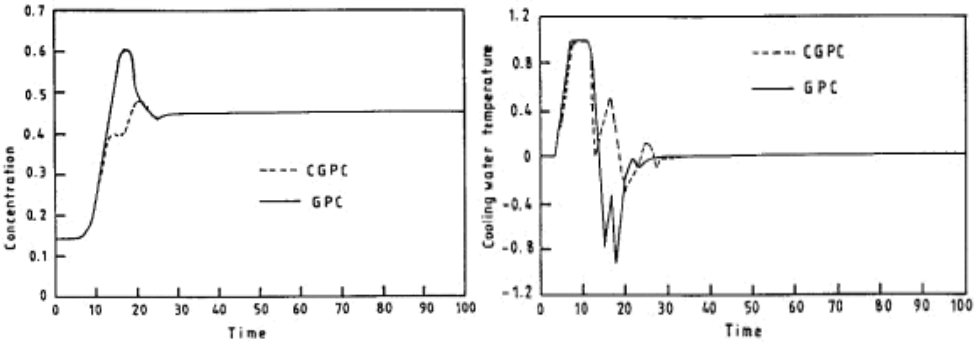


Fig. 9. Cooling water temperature and concentration plots of CSTR for a step change in concentration from 0.144 to 0.445.

The results illustrate the better performance of CGPC for SISO control of nonlinear systems that exhibit multiple steady states and unstable behavior.

5. Nonlinear model predictive control

Linear MPC employs linear or linearized models to obtain the predictive response of the controlled process. Linear MPC is proved useful for controlling processes that exhibit even some degree of nonlinear behavior (Eaton and Rawlings, 1992; Venkateswarlu and Gangiah, 1997). However, the greater the mismatch between the actual process and the representative

model, the degree of deterioration in the control performance increases. Thus the control of a highly nonlinear process by MPC requires a suitable model that represents the salient nonlinearities of the process. Basically, two different approaches are used to develop nonlinear dynamic models. These approaches are developing a first principle model using available process knowledge and developing an empirical model from input-output data. The first principle modeling approach results models in the form of coupled nonlinear ordinary differential equations and various model predictive controllers based on this approach have been reported for nonlinear processes (Wright and Edgar, 1994 ; Ricker and Lee, 1995). The first principle models will be larger in size for high dimensional systems thus limiting their usage for MPC design. On the other hand, the input-output modeling approach can be conveniently used to identify nonlinear empirical models from plant data, and there has been a growing interest in the development of different types of MPCs based on this approach (Hernandez and Arkun, 1994; Venkateswarlu and Venkat Rao, 2005). The other important aspect in model predictive control of highly nonlinear systems is the optimization algorithm. Efficient optimization algorithms exist for convex optimization problems. However, the optimization problem often becomes nonconvex in the presence of nonlinear characteristics/constraints and is usually more complex than convex optimization. Thus, the practical usefulness of nonlinear predictive control is hampered by the unavailability of suitable optimization tools (Camacho and Bordons, 1995). Sequential quadratic programming (SQP) is widely used classical optimization algorithm to solve nonlinear optimization problems. However, for the solution of large problems, it has been reported that gradient based methods like SQP requires more computational efforts (Ahn et al., 1999). More over, classical optimization methods are more sensitive to the initialization of the algorithm and usually leads to unacceptable solutions due to convergence to local optima. Consequently, efficient optimization techniques are being used to achieve the improved performance of NMPC.

This work presents a NMPC based on stochastic optimization technique. Stochastic approach based genetic algorithms (GA) and simulated annealing (SA) are potential optimization tools because of their ability to handle constrained, nonlinear and nonconvex optimization problems. These methods have the capacity to escape local optima and find solutions in the vicinity of the global optimum. They have the ability to use the values from the model in a black box optimization approach with out requiring the derivatives. Various studies have been reported to demonstrate the ability of these methods in providing efficient optimization solutions (Hanke and Li, 2000 ; Shopova and Vaklieva-Bancheva, 2006).

5.1 NMPC design

In NMPC design, the identified input-output nonlinear process model is explicitly used to predict the process output at future time instants over a specified prediction horizon. A sequence of future control actions over a specified control horizon is calculated using a stochastic optimizer which minimizes the objective function under given operating constraints. In this receding horizon approach, only the first control action in the sequence is implemented. The horizons are moved towards the future. The structure of the stochastic optimizer based NMPC is shown in Figure 10.

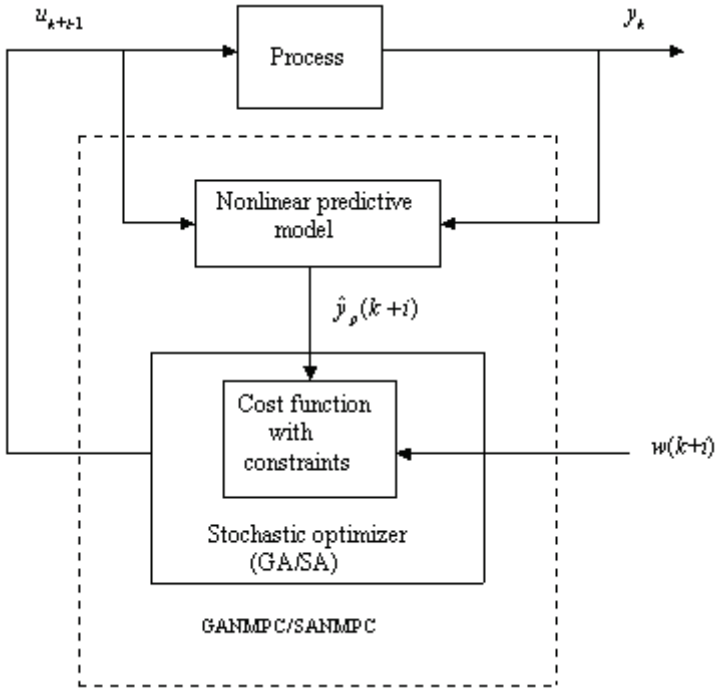


Fig.10. Structure of stochastic optimization based NMPC.

Simulated Annealing

Simulated annealing (SA) is analogous to the process of atomic rearrangement of a substance into a highly ordered crystalline structure by way of slowly cooling-annealing the substance through successive stages. This method is found to be a potential tool to solve a variety of optimization problems (Kirkpatrick et al., 1983 ; Dolan et al., 1989). Crystalline structure with a high degree of atomic order is the purest form of the substance, indicating the minimum energy state. The principle of SA mimics the annealing process of slow cooling of molten metal to achieve the minimum function value. The cooling phenomena is simulated by controlling a temperature like parameter introduced with the concept of the Bolzmann probability distribution, which determines the energy distribution probability, P of the system at the temperature, T according to the equation:

$$P(E) = \exp(-E / k_B T_A) \tag{56}$$

where k_B is the Bolzmann constant. The Bolzmann distribution concentrates around the state with the lowest energy. For $T \to 0$, $P(E) \to 0$ and only the state with the lowest energy can have a probability greater than zero. However, cooling the system too fast could result in a higher state of energy and may lead to frozen the system to a metastable state.

The SA is a point by point method based on Monte Carlo approach. The algorithm begins at an initial random point called u and a high temperature T, and the function value at this point is evaluated as $E(k)$. A second point is created in the vicinity of the initial point u and the function value corresponding to this point is obtained as $E(k+1)$. The difference in function values at these points ΔE is obtained as

$$\Delta E = E(k+1) - E(k) \tag{57}$$

If $\Delta E \leq 0$, the second point is accepted, otherwise the point is accepted probabilistically, governed by the temperature dependent Bolzmann probability factor

$$P_r = \exp(-\Delta E / k_B T_A) \tag{58}$$

The annealing temperature, T_A is a parameter in the optimization algorithm and is set by a predefined annealing schedule which starts at a relatively high temperature and steps slowly downward at a prescribed rate in accordance with the equation

$$T_A^{m+1} = \alpha T_A^m, \qquad 0 < \alpha < 1 \tag{59}$$

As the temperature decreases, the probability of the acceptance of the point u will be decreased according to Eq. (58). The parameter α is set such that at the point of convergence, the temperature T_A reaches a small value. The procedure is iteratively repeated at each temperature with the generation of new points and the search is terminated when the convergence criterion set for the objective is met.

Nonlinear modeling and model identification

Various model structures such as Volterra series models, Hammerstein and Wiener models, bilinear models, state affine models and neural network models have been reported in literature for identification of nonlinear systems. Haber and Unbehauen (1990) presented a comprehensive review on these model structures. The model considered in this study for identification of a nonlinear process has a polynomial ARMA structure of the form

$$\begin{aligned}\hat{y}(k) = \theta_0 + \sum_{i=1}^{n_y} \theta_{1,i}\, y(k-i) + \sum_{i=1}^{n_u} \theta_{2,i}\, u(k-i) + \sum_{i=1}^{n_y+1} \theta_{3,i}\, y(k-i)u(k-i) + \\ \sum_{i=1}^{n_u} \sum_{j=1}^{i+1} \theta_{4,ij}\, u(k-i)u(k-j) + ...\end{aligned} \tag{60}$$

or simply

$$\hat{y}(k) = f\left(y(k-1),....., y(k-n_y), u(k-1),....., u(k-n_u)\right) \tag{61}$$

Here k refers the sampling time, y and u are the output and input variables, and n_y and n_u refer the number of output and input lags, respectively. This type of polynomial model structures have been used by various researchers for process control (Morningred et al.,

1992 ; Hernandez and Arkun, 1993). The main advantage of this model is that it represents the process nonlinearities in a structure with linear model parameters, which can be estimated by using efficient parameter estimation methods such as recursive least squares. Thus the model in (61) can be rearranged in a linear regression form as

$$\hat{y}(k) = \theta^T(k-1)\varphi(k-1) + \varepsilon(k) \tag{62}$$

where θ is a parameter vector, φ represents input-output process information and ε is the estimation error. The parameters in the model can be estimated by using recursive least squares based on a priori process knowledge representing the process characteristics over a wide range of operating conditions.

Predictive Model Formulation

The primary purpose of NMPC is to deal with complex dynamics over an extended horizon. Thus, the model must predict the process dynamics over a prediction horizon enabling the controller to incorporate future set point changes or disturbances. The polynomial input-output model provides one step ahead prediction for process output. By feeding back the model outputs and control inputs, the one step-a head predictive model can be recurrently cascaded to itself to generate future predictions for process output. The N step predictions can be obtained as follows

$$\begin{aligned}
\hat{y}(k+1/k) &= f\left(y(k),.....,y(k+1-n_y),u(k),.....,u(k+1-n_u)\right)\\
\hat{y}(k+2/k) &= f\left(\hat{y}(k+1/k),.....,y(k+2-n_y),u(k+1),.....,u(k+M-n_u)\right)\\
\hat{y}(k+N/k) &= f\begin{pmatrix}\hat{y}(k+N-1/k),.....,\hat{y}(k+N-n_y/k),u(k+M-1),.....,\\ u(k+M-n_u)\end{pmatrix}
\end{aligned} \tag{63}$$

where N is the prediction horizon and M is the control horizon.

Objective function

The optimal control input sequence in NMPC is computed by minimizing an objective function based on a desired output trajectory over a prediction horizon:

$$\begin{aligned}
&Min \quad J = \sum_{i=1}^{N}\lambda\left(w(k+i)-\hat{y}_p(k+i)\right)^2 + \sum_{i=1}^{M}\gamma\left(\Delta u(k+i-1)^T\ \Delta u(k+i-1)\right)\\
&u(k),\ u(k+1),.....,u(k+M-1)
\end{aligned} \tag{64}$$

subject to constraints:

$$\begin{aligned}
y_{\min} \le \hat{y}_p(k+i) \le y_{\max} \qquad & (i=1,.....,N)\\
u_{\min} \le u(k+i) \le u_{\max} \qquad & (i=0,.....,M-1)
\end{aligned}$$

$$\Delta u_{\min} \le \Delta u(k+i) \le \Delta u_{\max} \qquad (i = 0, \ldots\ldots, M-1)$$

where $\hat{y}_p(k+i)$, i=1, …., N, are the future process outputs predicted over the prediction horizon, w_{k+i}, i=1, …., N, are the setpoints and $u(k+i)$, i=0, ….., M-1, are the future control signals. The λ and γ represent the output and input weightings, respectively. The u_{min} and u_{max} are the minimum and maximum values of the manipulated inputs, and Δu_{min} and Δu_{max} represent their corresponding changes, respectively. Computation of future control signals involves the minimization of the objective function so as to bring and keep the process output as close as possible to the given reference trajectory, even in the presence of load disturbances. The control actions are computed at every sampling time by solving an optimization problem while taking into consideration of constraints on the output and inputs. The control signal, u is manipulated only with in the control horizon, and remains constant afterwards, i.e., $u(k+i) = u(k+M-1)$ for $i = M, \ldots, N-1$. Only the first control move of the optimized control sequence is implemented on the process and the output measurements are obtained. At the next sampling instant, the prediction and control horizons are moved ahead by one step, and the optimization problem is solved again using the updated measurements from the process. The mismatch d_k between the process $y(k)$ and the model $\hat{y}(k)$ is computed as

$$d_k = b(y(k) - \hat{y}(k)) \tag{65}$$

where b is a tunable parameter lying between 0 and 1. This mismatch is used to compensate the model predictions in Eq. (62):

$$\hat{y}_p(k+i) = \hat{y}(k+i) + d_k \qquad (\text{for all } i = 1 \text{ to } N) \tag{66}$$

These predictions are incorporated in the objective function defined by Eq. (64) along with the corresponding setpoint values.

NMPC based on stochastic optimization

NMPC design based on simulated annealing (SA) requires to specify the energy function and random number selection for control input calculation. The control input is normalized and constrained with in the specified limits. The random numbers used for the control input, Δu equals the length of the control horizon, and these numbers are generated so that they satisfy the constraints. A penalty function approach is considered to satisfy the constraints on the input variables. In this approach, a penalty term corresponding to the penalty violation is added to the objective function defined in Eq. (64). Thus the violation of the constraints on the variables is accounted by defining a penalty function of the form

$$P = \sum_{i=1}^{N} \mu \left(\Delta u(k+i)\right)^2 \tag{67}$$

where the penalty parameter, μ is selected as a high value. The penalized objective function is then given by

$$f(x) = J + P \tag{68}$$

where J is defined by Eq. (64). At any instant, the current control signal, u_k and the prediction output based on this control input, $\hat{y}(k+i)$ are used to compute the objective function $f(x)$ in Eq. (68) as the energy function, $E(k+i)$. The $E(k+i)$ and the previously evaluated $E(k)$ provides the ΔE as

$$\Delta E(k) = E(k+i) - E(k) \tag{69}$$

The comparison of the ΔE with the random numbers generated between 0 and 1 determines the probability of acceptance of $u(k)$. If $\Delta E \leq 0$, all $u(k)$ are accepted. If $\Delta E > 0$, $u(k)$ are accepted with a probability of $\exp(-\Delta E/T_A)$. If n_m be the number of variables, n_k be the number of function evaluations and n_T be the number of temperature reductions, then the total number of function evaluations required for every sampling condition are $(n_T \times n_k \times n_m)$. Further details of NMPC based on stochastic optimization can be referred elsewhere (Venkateswarlu and Damodar Reddy, 2008).

Implementation procedure

The implementation of NMPC based on SA proceeds with the following steps.

1. Set T_A as a sufficiently high value and let n_k be the number of function evaluations to be performed at a particular T_A. Specify the termination criterion, ε. Choose the initial control vector, u and obtain the process output predictions using Eq. (63). Evaluate the objective function, Eq. (68) as the energy function $E(k)$.
2. Compute the incremental input vector Δu_k stochastically and update the control vector as

$$u(k+i) = u(k) + \Delta u(k) \tag{70}$$

Calculate the objective function, $E(k+i)$ as the energy function based on this vector.

3. Accept $u(k+i)$ unconditionally if the energy function satisfies the condition

$$E(k+i) \leq E(k) \tag{71}$$

Otherwise, accept $u(k+i)$ with the probability according to the Metropolis criterion

$$\exp\left(-\frac{\left(E(k+i)-E(k)\right)}{T_A^{'}}\right) \geq r \tag{72}$$

where $T_A^{'}$ is the current annealing temperature and r represents random number. This step proceeds until the specified function evaluations, n_k are completed.

4. Carry out the temperature reduction in the outer loop according to the decrement function

$$T_A^{/} = \alpha T_A \tag{73}$$

where α is temperature reduction factor. Terminate the algorithm if all the differences are less than the prespecified ε.

5. Go to step 2 and repeat the procedure for every measurement condition based on the updated control vector and its corresponding process output.

5.2 Case study: nonlinear model predictive control of reactive distillation column

The performance of NMPC based on stochastic optimization is evaluated through simulation by applying it to a ethyl acetate reactive distillation column.

Analysis of Results

The process, the column details, the mathematical model and the control scheme of ethyl acetate reactive distillation column given in Section 3.2 is used for NMPC implementation. In this operation, since the ethyl acetate produced is withdrawn as a product in the distillate stream, controlling the purity of this main product is important in spite of disturbances in the column operation. This becomes the main control loop for NMPC in which reflux flow rate is used as a manipulated variable to control the purity of ethyl acetate. Since reboiler and condenser holdups act as pure integrators, they also need to be controlled. These become the auxiliary control loops and are controlled by conventional PI controllers in which the distillate flow rate is considered as a manipulated variable to control the condenser molar holdup and the bottom flow rate is used to control the reboiler molar holdup. The tuning parameters used for both the PI controllers of reflux drum and reboiler holdups are k_c = - 0.001 and τ_I = 1.99 x 10^4 (Vora and Dauotidis, 2001). The SISO control scheme for the column with the double feed configuration used in this study is shown in the Fig. 3.

The input-output data to construct the nonlinear empirical model is obtained by solving the model equations using Euler's integration with a step size of 2.0 s. A PI controller with a series of step changes in the set point of ethyl acetate composition is used for data generation. The input data (reflux flow) is normalized and used along with the outputs (ethyl acetate composition) in model building. The reflux flow rate is constrained with in the limits of 20 mol/s and 5 mol/s. A total number of 25000 data sets is considered to develop the model. The model parameters are determined by using the well known recursive least squares algorithm (Goodwin and Sin, 1984), the application of which has been shown elsewhere (Venkateswarlu and Naidu, 2001). After evaluating model structure in Eq. (60) for different orders of n_y and n_u , the model with the order n_y=2 and n_u=2 is found to be more appropriate to design and implement the NMPC with stochastic optimization. The structure of the model is in the form

$$\hat{y}_k = \theta_0 + \theta_1 y_{k-1} + \theta_2 u_{k-1} + \theta_3 y_{k-1} u_{k-1} + \theta_4 y_{k-2} u_{k-2} + \theta_5 u_{k-1} u_{k-2} \qquad (74)$$

The parameters of this model are determined as θ_0=-0.000774, θ_1=1.000553, θ_2=0.002943, θ_3=-0.003828, θ_4=0.000766 and θ_5=-0.000117. This identified model is then used to derive the future predictions for the process output by cascading the model to it self as in Eq. (63). These model predictions are added with the modeling error, $d(k)$ defined by Eq. (65), which is considered to be constant for the entire prediction horizon. The weightings λ and γ in the objective function, Eq. (64) are set as 1.0 x 10^7 and 7.5 x 10^4, respectively. The penalty parameter, μ in Eq. (67) is assigned as 1.0 x 10^5. The cost function used in NMPC is the penalized objective function, eq. (68), based on which the SA search is computed. The incremental input, Δu in SA search is constrained with in the limits -0.0025 and 0.0025, respectively. The actual input, u involved with the optimization scheme is a normalized value and is constrained between 0 and 1. The objective function in Eq. (68) is evaluated as the energy function at each instant. The initial temperature T is chosen as 500 and the

number of iterations at each temperature is set as 250. The temperature reduction factor, α in Eq. (73) is set as 0.5. The control input determined by the stochastic optimizer is denormalized and implemented on the process. A sample time of 2 s is considered for the implementation of the controller.

The performance of NMPC based on SA is evaluated by applying it for the servo and regulatory control of ethyl acetate reactive distillation column. On evaluating the results with different prediction and control horizons, the NMPC with a prediction horizon of around 10 and a control horizon of around 1 to 3 is observed to provide better performance. The results of NMPC are also compared with those of LMPC presented in Section 3 and a PI controller. The tuning parameters of the PI controller are set as k_C = 10.0 and τ_I = 1.99 x 10^4 (Vora and Dauotidis, 2001). The servo and regulatory results of NMPC along with the results of LMPC and PI controller are shown in Figures 11-14. Figure 11 compares the input and output profiles of NMPC with LMPC and PI controller for step change in ethyl acetate composition from 0.6827 to 0.75. The responses in Figure 12 represent 20% step decrease in ethanol feed flow rate, and the responses in Figure 13 correspond to 20% step increase in reboiler heat load. These responses show the better performance of NMPC over LMPC and PI controller. Figure 14 compares the performance of NMPC and LMPC in tracking multiple step changes in setpoint of the controlled variable. The results thus show the stability and robustness of NMPC towards load disturbances and setpoint changes.

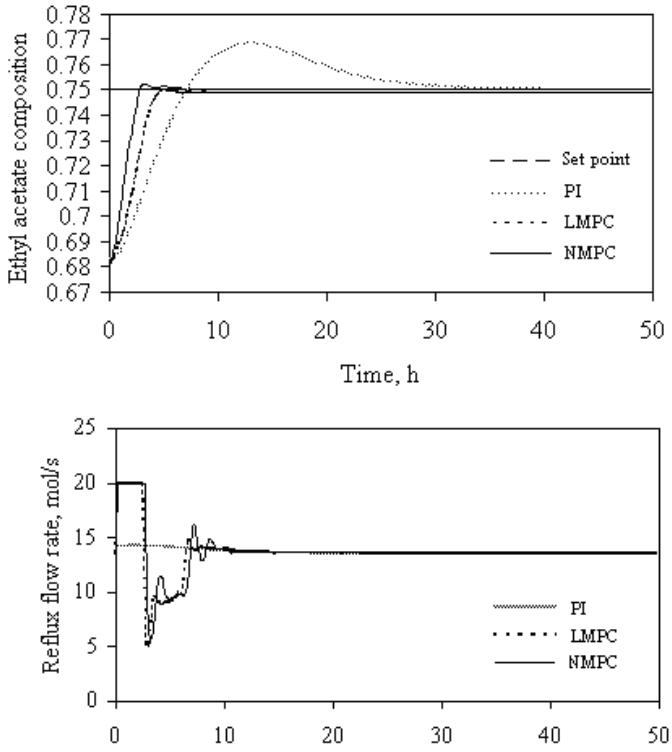


Fig.11. Output and input profiles for step increase in ethyl acetate composition setpoint.

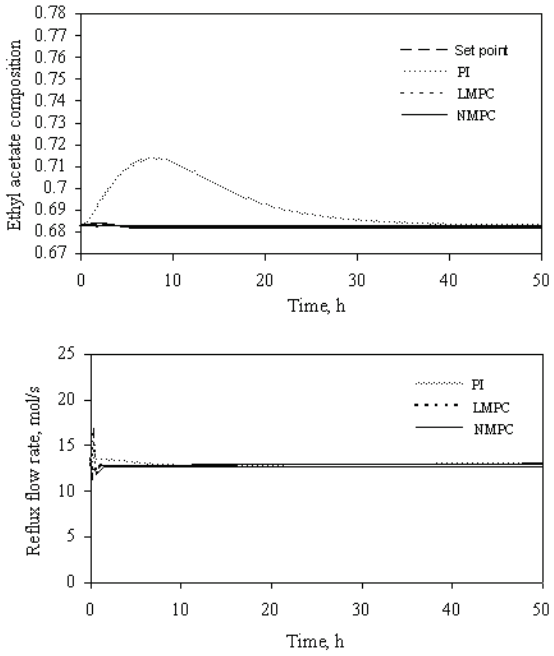


Fig.12. Output and input profiles for step decrease in ethanol feed flow rate.

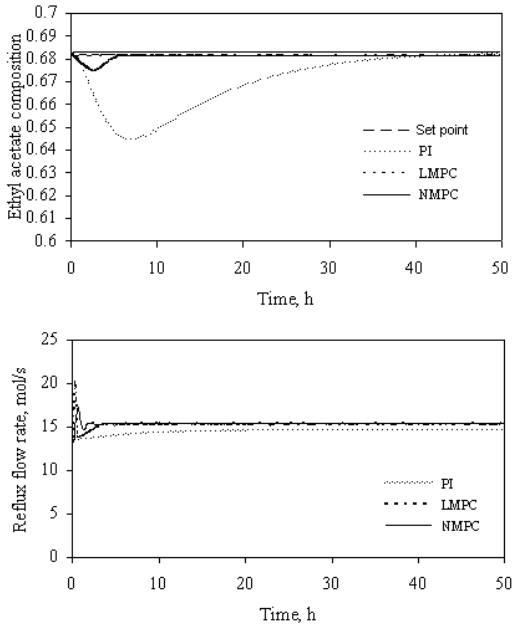


Fig.13. Output and input profiles for step increase in reboiler heat load.

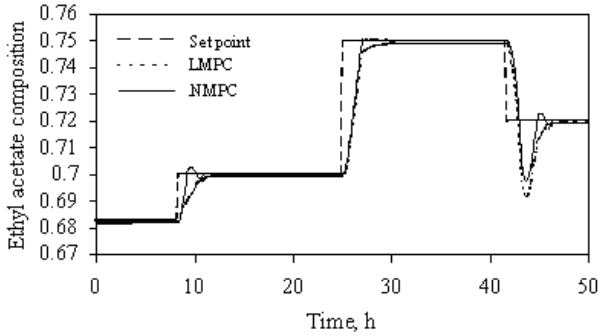


Fig. 14. Output responses for multiple setpoint changes in ethyl acetate composition

6. Conclusions

Model predictive control (MPC) is known to be a powerful control strategy for a variety of processes. In this study, the capabilities of linear and nonlinear model predictive controllers are explored by designing and applying them to different nonlinear processes. A linear model predictive controller (LMPC) is presented for the control of an ethyl acetate reactive distillation. A generalized predictive control (GPC) and a constrained generalized predictive control (CGPC) are presented for the control of an unstable chemical reactor. Further, a nonlinear model predictive controller (NMPC) based on simulated annealing is presented for the control of a highly complex nonlinear ethyl acetate reactive distillation column. The results of these controllers are evaluated under different disturbance conditions for their servo and regulatory performance and compared with the conventional controllers. From these results, it is observed that though linear model predictive controllers offer better control performance for nonlinear processes over conventional controllers, the nonlinear model predictive controller provides effective control performance for highly complex nonlinear processes.

Nomenclature

ARX autoregressive moving average
A_h heat transfer area, m^2
A_{tray} tray area, m^2
B bottom flow rate, mol s^{-1}
B_h dimensionless heat of reaction
C concentration, mol m^{-3}
C_A reactant concentration, mol m^{-3}
C_{Af} feed concentration, mol m^{-3}
C_k catalyst concentration, % vol
C_p specific heat capacity, J kg^{-1} K^{-1}
D distillate flow rate, mol s^{-1}
D_a Damkohler number
du_{min} lower limit of slew rate

du_{max} upper limit of slew rate
E total enthalpy of liquid on plate, kJ
FL liquid feed flow rate on plate, mol s^{-1}
FV vapor feed on plate, mol s^{-1}
F_{Ac} acetic acid feed flow rate, mol s^{-1}
F_{Eth} ethanol feed flow rate, mol s^{-1}
F_o volumetric feed rate, m^3 s^{-1}
H molar enthalpy of vapor stream, kJ mol^{-1}
h molar enthalpy of liquid stream, kJ mol^{-1}
k_1 reaction rate constant, m^3 mol^{-1} s^{-1}
h_{weir} weir height, m
K_C constant of reaction equilibrium
L molar liquid flow rate, mol s^{-1}
L_{weir} weir length, m
L^{liquid} liquid level on tray, m
M molar holdup on plate, m
MW_{av} average molecular weight, g mol^{-1}
N_1 minimum costing horizon
N_2 maximum costing horizon
N_3 control horizon
P pressure on plate, pascal
Q heat exchange, kJ
R number of moles reacted, mol s^{-1}
R_g gas constant, J mol^{-1} K^{-1}
RLS recursive least squares
r rate of reaction, mol s^{-1} m^{-3}
ρ_{av} average density, g m^{-3}
T temperature, K
T_c coolant temperature, K
T_f feed temperature, K
T_r reactor temperature, K
U heat transfer coefficient, J m^{-2} s^{-1} K^{-1}
u controller output
u_{min} lower limit of manipulated variable
u_{max} upper limit of manipulated variable
VLE vapor-liquid equilibrium
V molar vapor flow rate, mol s^{-1}
x mole fraction in liquid phase
x_1 dimensionless reactant concentration
x_2 dimensionless reactant temperature
y mole fraction in vapor phase
y_{min} lower limit of output variable
y_{max} upper limit of output variable
ρ_{av} average density, g m^{-3}

7. References

Ahn, S.M., Park, M.J., Rhee, H.K. Extended Kalman filter based nonlinear model predictive control of a continuous polymerization reactor. *Industrial &. Engineering Chemistry Res*earch, 38: 3942-3949, 1999.

Alejski, K., Duprat, F. Dynamic simulation of the multicomponent reactive distillation. *Chemical Engineering Sci*ence, 51: 4237-4252, 1996.

Bazaraa, M.S., Shetty, C.M. *Nonlinear Programming,* 437-443 (John Wiley & Sons, New York), 1979.

Calvet, J P., Arkun, Y. Feedforward and feedback linearization of nonlinear systems and its implementation using internal model control (IMC). *Industrial &. Engineering Chemistry Res*earch, 27: 1822-1831, 1988.

Camacho, E. F. Constrained generalized predictive control. *IEEE Trans Aut Contr*, 38: 327-332, 1993.

Camacho, E. F., Bordons, C. *Model Predictive Control in the Process Industry*; Springer Verlag: Berlin, Germany, 1995.

Clarke, D.W., Mohtadi, C and Tuffs, P.S. Generalized predictive control – Part I. The basic algorithm. *Automatica*, 23: 137-148, 1987.

Cutler, C.R. and Ramker, B.L. Dynamic matrix control – a computer control algorithm, *Proceedings Joint Automatic Control Conference*, Sanfrancisco, CA.,1980.

Dolan, W.B., Cummings, P.T., Le Van, M.D. Process optimization via simulated annealing: application to network design. *AIChE Journal*. 35: 725-736, 1989.

Garcia, C.E., Prett, D.M., and Morari, M. Model predictive control: Theory and Practice - A survey. *Automatica*, 25: 335-348, 1989.

Eaton, J.W., Rawlings, J.B. Model predictive control of chemical processes. *Chemical Engineering Sci*ence, 47: 705-720, 1992.

Goodwin, G.C., Sin, K.S. Adaptive Filtering Prediction and Control (Printice Hall, Englewood Cliffs, New Jersey), 1984.

Haber, R., Unbehauen, H. Structure identification of nonlinear dynamical systems -a survey on input/output approaches. *Automatica*, 26: 651-677, 1990.

Hanke, M., Li, P. Simulated annealing for the optimization of batch distillation process. *Computers and Chemical Engineering,* 24: 1-8, 2000.

Hernandez, E., Arkun, Y., Study of the control relevant properties of backpropagation neural network models of nonlinear dynamical systems. *Computers & Chemical Engineering*, 16: 227-240, 1992.

Hernandez, E., Arkun, Y. Control of nonlinear systems using polynomial ARMA models. *AIChE Journal*, 39: 446-460, 1993.

Hernandez, E., Arkun, Y. On the global solution of nonlinear model predictive control algorithms that use polynomial models. *Computers and Chemical Engineering,* 18: 533-536, 1994.

Hsia, T.C. System Identification: Least Square Methods (Lexington Books, Lexington, MA), 1977.

Kirkpatrick, S., Gelatt Jr, C.D., Veccchi, M.P. Optimization by simulated annealing. *Scienc,* 220: 671-680, 1983.

Morningred, J.D., Paden, B.E., Seborg D.E., Mellichamp, D.A., An adaptive nonlinear predictive controller. *Chemical Engineering Science*, 47: 755-762, 1992.

Qin, J., Badgwell, T. An overview of industrial model predictive control technology; In: V th International Conference on Chemical Process Control (Kantor, J.C., Garcia, C.E., Carnhan, B., Eds.): *AIChE Symposium Series, 93*: 232-256, 1997.

Richalet, J., Rault, A., Testud, J. L. and Papon, J. Model predictive heuristic control: Application to industrial processes. *Automatica*, 14: 413-428, 1978.

Ricker, N.L., Lee, J.H. Nonlinear model predictive control of the Tennessee Eastman challenging process. *Computers and Chemical Engineering,* 19: 961-981, 1995.

Smith, J.M., Van Ness, H.C. Abbot, M.M., A Text Book on Introduction to Chemical Engineering Thermodynamics, 5 th Ed., Mc-Graw Gill International. 1996.

Shopova, E.G., Vaklieva-Bancheva, N.G. BASIC-A genetic algorithm for engineering problems solution. *Computers and Chemical Engineering,* 30: 1293-1309, 2006.

Venkateswarlu, Ch., Gangiah, K. Constrained generalized predictive control of unstable nonlinear processes. *Transactions of Insitution of Chemical Engineers*, 75: 371-376, 1997.

Venkateswarlu, Ch., Naidu, K.V.S. Adaptive fuzzy model predictive control of an exothermic batch chemical reactor. *Chemical Engineering Communications,* 186: 1-23, 2001.

Venkateswarlu, Ch., Venkat Rao, K. Dynamic recurrent radial basis function network model predictive control of unstable nonlinear processes. *Chemical Engineering Sci*ence, 60: 6718-6732, 2005.

Venkateswarlu, Ch., Damodar Reddy, D. Nonlinear model predictive control of reactive distillation based on stochastic optimization. *Industrial Engineering & Chemistry Research*, 47: 6949-6960, 2008.

Vora, N., Daoutidis, P. Dynamics and control of ethyl acetate reactive distillation column. *Industrial &. Engineering Chemistry Res*earch, 40: 833-849, 2001.

Uppal, A., Ray, W.H., Poore, A. B. On the dynamic behavior of continuous stirred tank reactors. *Chemical Engineering Sci*ence, 29: 967- 985,1974.

Wright, G. T., Edgar, T. F. Nonlinear model predictive control of a fixed-bed water-gas shift reactor: an experimental study. *Computers and Chemical Engineering,* 18: 83-102, 1994.

6

Approximate Model Predictive Control for Nonlinear Multivariable Systems

Jonas Witt and Herbert Werner
Hamburg University of Technology
Germany

1. Introduction

The control of multi-input multi-output (MIMO) systems is a common problem in practical control scenarios. However in the last two decades, of the advanced control schemes, only linear model predictive control (MPC) was widely used in industrial process control (Maciejowski, 2002). The fundamental common idea behind all MPC techniques is to rely on predictions of a plant model to compute the optimal future control sequence by minimization of an objective function. In the predictive control domain, *Generalized Predictive Control* (GPC) and its derivatives have received special attention. Particularly the ability of GPC to be applied to unstable or time-delayed MIMO systems in a straight forward manner and the low computational demands for static models make it interesting for many different kinds of tasks. However, this method is limited to linear models.

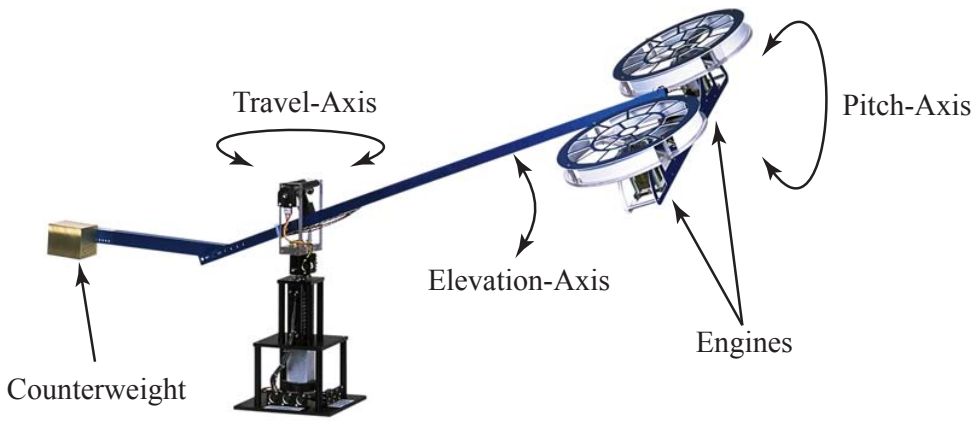


Fig. 1. Quanser 3-DOF Helicopter

If nonlinear dynamics are present in the plant a linear model might not yield sufficient predictions for MPC techniques to function adequately. A related technique that can be applied to nonlinear plants is *Approximate (Model) Predictive Control* (APC). It uses an instantaneous linearization of a nonlinear model based on a neural network in each sampling instant. It is

similar to GPC in most aspects except that the instantaneous linearization of the neural network yields an adaptive linear model. Previously this technique has already successfully been applied to a pneumatic servomechanism (Nørgaard et al., 2000) and gas turbine engines (Mu & Rees, 2004), however both only in simulation.
The main challenges in this work were the nonlinear, unstable and comparably fast dynamics of the 3-DOF helicopter by Quanser Inc. (2005) (see figure 1). APC as proposed by Nørgaard et al. (2000) had to be extended to the MIMO case and model parameter filtering was proposed to achieve the desired control and disturbance rejection performance.
This chapter covers the whole design process from nonlinear MIMO system identification based on an artificial neural network (ANN) in section 2 to controller design and presentation of enhancements in section 3. Finally the results with the real 3-DOF helicopter system are presented in section 4. On the way pitfalls are analyzed and practical application hints are given.

2. System Identification

The correct identification of a model is of high importance for any MPC method, so special attention has to be paid to this part of controller design. The success of the identification will determine the performance of the final controlled system directly or even whether the system is stable at all.
Basically there are a few points one has to bear in mind during the experiment design (Ljung, 1999):

- The sampling rate should be chosen appropriately.
- The experimental conditions should be close to the situation for which the model is going to be used. Especially for MIMO systems this plays an important role as this may be nontrivial.
- The identification signal should be sufficiently rich to excite all modes of the system. For nonlinear systems not only the frequency spectrum but also the excitation of different amplitudes should be sufficient.
- Periodic inputs have the advantage that they reduce the influence of noise on the output signal but increase the experiment length.

The following sections guide through the full process of the MIMO identification by means of the practical experiences with the helicopter model.

2.1 Excitation Signal

The type of the excitation signal plays an important role as it should exhibit a few properties which affect the outcome essentially. Generally the input signal should be persistently exciting of at least twice the system order. There are many different types of input signals which are not covered here (see Ljung (1999) for further reading). Despite the desirable optimal Crest factor, for nonlinear system identification binary signals are not an option due to the lack of excitation of different amplitudes. For this work an excitation signal comprised of independent multi-sine signals as described in (Evan et al., 2000) was designed. This is explored in the following section.

2.1.1 Assembling of Multisine Signals

A multisine is basically a sum of sinusoids:

$$u(t) = \sum_{k=1}^{n_s} A_k cos(\omega_k t + \phi_k)$$

where n_s is the number of present frequencies. This parameter should be large enough to guarantee persistent excitation.
A favourable attribute of multisine signals is that the spectrum can be determined directly. By this property it is possible to just include the frequency ranges that excite the system which is done by splitting the spectrum in a low (or main) and a high frequency band. As a rule of thumb one should choose the upper limit of the main frequency band ω_c around the system bandwidth ω_b, since choosing ω_c too low may result in unexcited modes, while $\omega_c \gg \omega_b$ does not yield additional information (Ljung, 1999). In a relay feedback experiment the bandwidth of the helicopter's pitch axis was measured to be $f_b \approx 0.67Hz$. As one can see in figure 2 the upper limit of the main frequency band $f_c = \omega_c/2\pi = 1.5Hz$ was chosen about twice as large but the higher frequencies from ω_c up to the Nyquist frequency ω_n are not entirely absent. This serves the purpose of making the mathematical model resistant to high frequency noise as the real system will typically not react to this high frequency band.

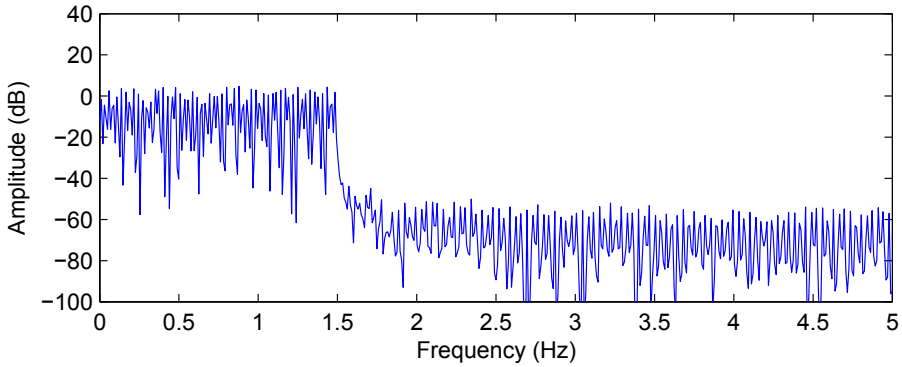


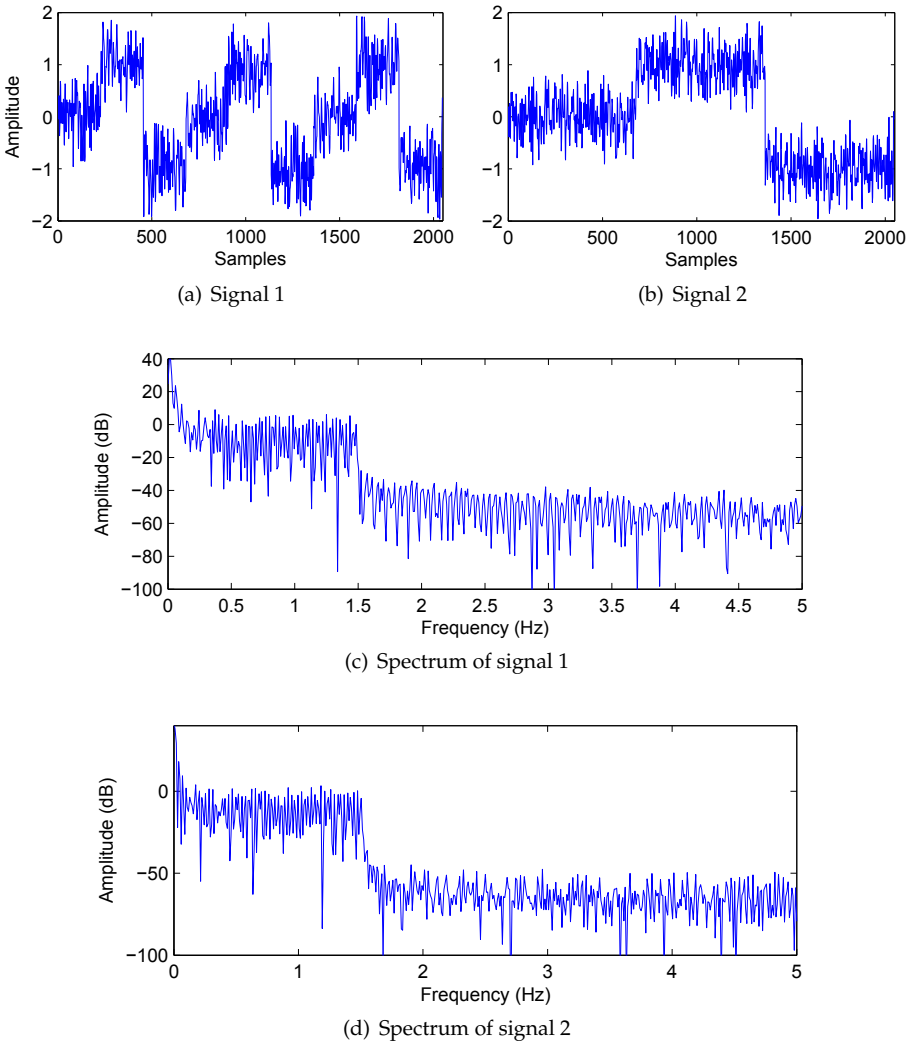
Fig. 2. Spectrum of the multisine excitation signal for the helicopter

2.1.2 Periodic Signals

To reduce the influence of noise present in the output signal of the plant, taking an integer number of periods of the input signal can be considered. If K periods of the input signal are taken, the signal to noise ratio is improved by this factor K. A drawback of periodic inputs is that they generally can not inject as much excitation into the system over a given time span as non-periodic inputs, since a signal of length N can at most excite a system of order N (Ljung, 1999). But as a periodic signal of length $N = KM$ consists of K periods of length M it has the same level of excitation as one period.
In the case of the helicopter three signal periods were chosen, as this proved to give consistent results for the present noise level.

2.1.3 MIMO Considerations

For MIMO systems the design of the input signal is a bit more complex, as there may be cross couplings between the different inputs which drive the outputs out of desired limits or the signal does not excite all modes sufficiently. The design process of the identification signal involves the consideration of system specifics and can not be generalized.



(a) Signal 1

(b) Signal 2

(c) Spectrum of signal 1

(d) Spectrum of signal 2

Fig. 3. MIMO signals with appropriate setpoints

In the case of the helicopter three axes need to be excited in all modes. A first attempt was to directly apply multisine signals to both inputs. For this attempt both inputs were limited

to low amplitudes though, as coincidental add-up effects quickly drove the system out of the operating bounds. Naturally this yielded bad models that did not resemble the actual plant very well.
A way to drive a MIMO system to different operating states is the use of setpoints that are added to the multisine signal. This enables the selective identification of certain modes of the system. At the same time this can be used as a means to keep the outputs inside of valid operating bounds, since the amplitude of the multisine signal can be chosen to be much lower than without setpoints. This enables much safer operation during the experiment, as the energy of the random signal can be reduced. Of course one has to keep in mind that the actual excitation signals amplitude has to be as large as possible to assure maximal excitation around each setpoint.
The spectrum of a multisine signal with additive setpoints does not differ much from the original multisine (figure 2) as can be seen in figure 3. The only difference is a peak in the low frequency band and a general small lifting in the upper band. Both signals are composed of multisines of same spectrum with unit variance and additive setpoints in the range of $[-1, 1]$. This assures overlapping amplitude ranges, which is desirable for a consistent model.

2.2 Closed Loop Identification

In recent years the interest in closed loop identification has generally risen, due to its importance for practical system identification. In many industrial processes existing control loops cannot be switched off during system identification for safety reasons or process restrictions. Likewise when dealing with unstable systems every experiment setup must involve stabilizing control loops to keep the output in a valid operating range. Another advantage of models computed from closed loop data is their better approximation of the behavior of a process under feedback which is important for successful controller design (Pico & Martinez, 2002). There are a few different approaches to closed loop identification of which the two most general are covered here:

- Direct Approach: ignore the presence of the feedback and directly identify the plant by plant input and output data. This has the advantage that no knowledge about the type of control feedback or even linearity of the controller is required.
- Indirect Approach: identify the closed loop and obtain the open loop model by deconvolution if possible. Obtaining the open loop model is only possible if the controller is known and both the closed loop plant model and the controller are linear.

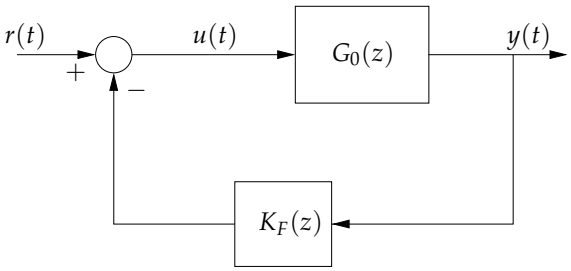


Fig. 4. Closed loop setup for identification

This section is limited to the techniques and practical experiences of the identification of the helicopter model and does not cover the whole theory of closed loop identification. For further information on this matter see Pico & Martinez (2002) or Ljung (1999).

2.2.1 Direct Identification

This is the natural approach to closed loop identification as it is similar to open loop identification if one keeps some fallacies in mind. In general this method applies a straightforward identification by taking the raw data from plant input $u(t)$ and plant output $y(t)$ and thus computes the model as if in open loop. Figure 4 shows the basic setup of the experiment.
A few general statements can be made about the direct identification in closed loop (Pico & Martinez, 2002):

- The experiment is informative if $r(t)$ is persistently exciting.
- Even if $r(t)$ is not a rich signal, the experiment can be informative if the system is driven by the output-noise and the feedback mechanism is of adequate structure avoiding a linear dependency between $y(t)$ and $u(t)$.
- Problems can arise if the amplitude of $r(t)$ is small in comparison to $u(t)$ and the feedback mechanism is approximately linear, i.e. $u(t) \approx -K_F(z)y(t)$.
- The direct approach can be problematic if the open loop plant is unstable, since the spectrum of $u(t)$ may be altered in a suboptimal way.

So if it is possible to use a rich signal for $r(t)$, the experiment is informative. But as this may not always be the case in a practical scenario, informative experiments can still be achieved by choosing an appropriate controller. Generally if $r(t)$ has a very low amplitude in comparison to $u(t)$ or if $r(t)$ is not a rich signal, problems arise if a dependency like $u(t) \approx -K_F(z)y(t)$ exists. But this can be avoided by choosing high order, time varying or nonlinear feedback mechanisms. For further information about appropriate controllers see (Pico & Martinez, 2002).

Issues with Unstable Systems

In the case of the helicopter the multisine from section 2.1 was used which exhibits a high level of excitation, thus a simple proportional derivative controller (PD controller) could be used for stabilization since the main source of exitation is the reference signal $r(t)$ in this case. Figure 5 shows the spectrum of the input signal $u(t)$ that was recorded during a SISO experiment to identify the helicopters pitch axis. The elevation axis was controlled separately by a PID controller. As one can see, the spectrum has changed significantly if compared to the original spectrum in figure 2. The low frequencies are heavily damped which comes naturally for an unstable plant like the helicopter's pitch axis, as a constant input would drive the system to infinity. For identification though this is unfavourable, since the low frequencies are not sufficiently excited although these frequencies lie in the normal operating band.
More problems arise during validation, since with the direct approach unstable plants yield unstable models directly and validating these is tedious. The problem with validating unstable models is that the validation is usually open loop, as the recorded input sequence is applied to the model and the output is compared to the recorded actual output. Errors are not compensated and thus add up as there are no control loops during validation so the model response may ascend to infinity - even with a decent model. A better way to verify the quality of an unstable model is to look at a k-step-ahead prediction, because errors do not have as much time to add up.

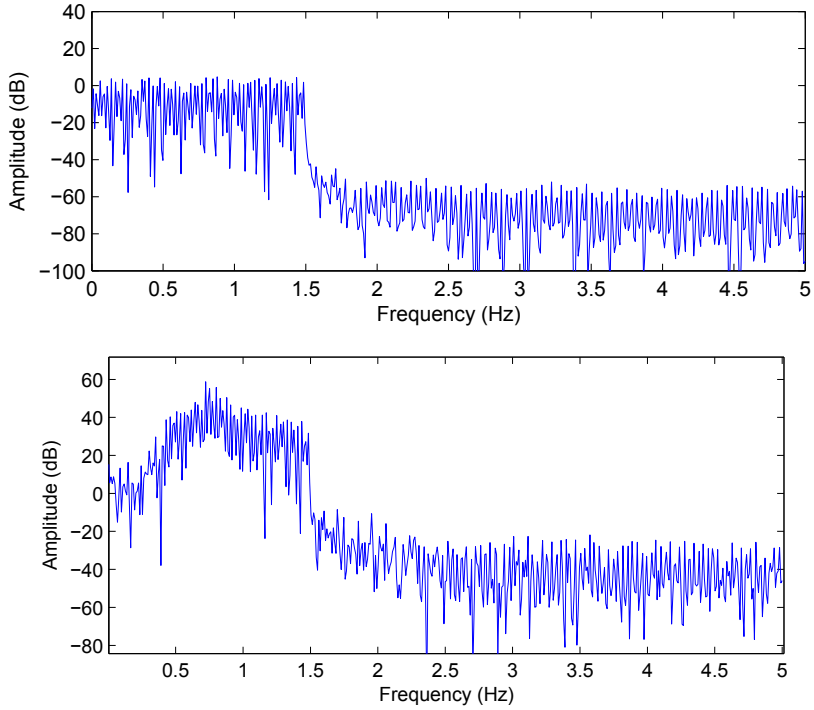


Fig. 5. Spectrum of the plant input $u(t)$ in closed loop excited by multisine signal (lower picture). Note that this signal is scaled in comparison to the original spectrum of the reference $r(t)$ (upper picture).

A comparison of the direct and the indirect approach by example of the helicopter's unstable pitch axis is presented in section 2.2.3.

2.2.2 Indirect Identification

The basic concept of indirect identification is to identify the closed loop as a whole and validate this model of the closed loop in the first place. Consecutive steps may include a deconvolution of the plant and controller to obtain the open loop model. This requires the regulator and the plant model to be both linear, and of course the regulator transfer function must be known. So in the case of nonlinear system identification these steps are not possible, even if the controller is linear and known, because in this case the equation can usually not be solved for $G_0(z)$ analytically.

In contrast to the direct approach the indirect approach avoids any alteration of the input signal's spectrum since the input to the closed loop directly is the reference signal $r(t)$ which has no dependence on other signals. This has the advantage, that even if $r(t)$ is small compared to $u(t)$ and the feedback mechanism is of a simple linear kind, an informative experiment is still achieved as long as $r(t)$ is persistently exciting (which the multisine signal from section 2.1 ensures). Another advantage is that unstable systems can be handled intuitively as the resulting model is stable. This eliminates the problems discussed in 2.2.1 in the validation phase

since the closed loop model does not have unstable poles as the open loop model would. A drawback of this method is, that the feedback mechanism increases the model order since it is identified along with the actual open loop model.
For the case of a known linear controller and a linear closed loop model, the open loop model can be obtained by deconvolution as was already mentioned. The closed loop transfer function corresponding to figure 4 is:

$$G_{cl}(z) = \frac{G_0(z)}{1 + G_0(z)K_F(z)}. \quad (1)$$

Solving for $G_0(z)$ yields:

$$G_0(z) = \frac{G_{cl}(z)}{1 - G_{cl}(z)K_F(z)}, \quad (2)$$

which is the final formula for obtaining the open loop model $G_0(z)$. So if either $G_{cl}(z)$ or $K_F(z)$ are nonlinear both formulas cannot be applied and a deconvolution is not possible. However for the linear case it will be shown that a controller design for the closed loop model $G_{cl}(z)$ can yield exactly the same overall system dynamics as a controller design for the open loop model $G_0(z)$. For control strategies that utilize linearizations of a nonlinear model like APC, this similarly implies that the direct use of a (in this case nonlinear) closed loop model has no adverse effects on the final performance.

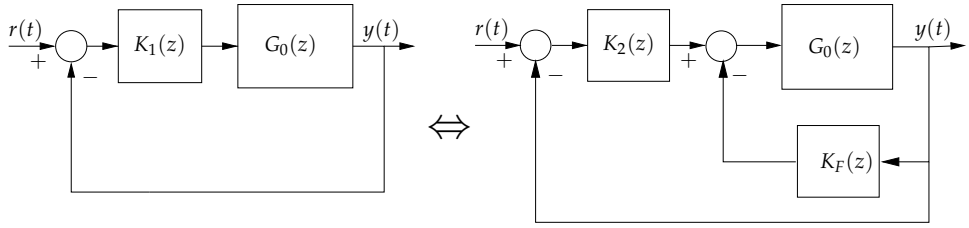


Fig. 6. Controller Setup for open loop and closed loop models

Theorem. *Given the closed loop system $G_{cl}(z)$ consisting of the open loop plant $G_0(z)$ and the controller $K_F(z)$, a controller $K_2(z)$ can be found that transforms the system to an equivalent system consisting of an arbitrary controller $K_1(z)$ that is applied to the plant $G_0(z)$ directly.*

Proof. The two system setups are depicted in figure 6. The transfer function of the left system is:

$$G_1(z) = \frac{K_1(z)G_0(z)}{1 + K_1(z)G_0(z)}$$

while the right system has the transfer-function:

$$\begin{aligned} G_2(z) &= \frac{K_2(z)G_{cl}(z)}{1 + K_2(z)G_{cl}(z)} \\ &= \frac{K_2(z)\frac{G_0(z)}{1+G_0(z)K_F(z)}}{1 + K_2(z)\frac{G_0(z)}{1+G_0(z)K_F(z)}} \\ &= \frac{K_2(z)G_0(z)}{1 + (K_F(z) + K_2(z))G_0(z)}. \end{aligned}$$

Now it has to be proved that there exists a controller $K_2(z)$ that transforms $G_2(z)$ to $G_1(z)$ for any given $K_1(z)$. It is clear that the system $G_2(z)$ with the $K_F(z)$ feedback controller can achieve exactly the same performance as the $G_1(z)$ system if this is the case.

$$\begin{aligned} \frac{K_1(z)G_0(z)}{1+K_1(z)G_0(z)} &= \frac{K_2(z)G_0(z)}{1+(K_F(z)+K_2(z))G_0(z)} \\ K_1(z)G_0(z)+K_1(z)K_F(z)G_0(z)G_0(z) &= K_2(z)G_0(z) \\ K_2(z) &= K_1(z)+K_1(z)K_F(z)G_0(z) \\ &= K_1(z)(1+K_F(z)G_0(z)) \end{aligned}$$

2.2.3 Indirect vs. Direct Approach

The quality of the models heavily depends on the experiment setup and the identification approach chosen. This is valid even more for the identification of unstable systems. This will be shown in the following with the example of the SISO identification of the helicopter's pitch axis. Consider the experiment setup from figure 4. As discussed in section 2.1 three periods of a multisine signal with spectrum as in figure 2 are applied to the reference input $r(t)$. The feedback controller $K_F(z)$ is a hand tuned PD controller. The input data for the identification process are $r(t)$ for the indirect approach and $u(t)$ for the direct approach respectively, the output data is $y(t)$ in both cases. The spectrum of $u(t)$ is shown in figure 5 (the spectrum of $r(t)$ is just the one of the multisine in figure 2). To obtain a fair comparison of the approaches the open loop model is computed for both (for the indirect approach (2) is used to compute $G_0(z)$). Since direct validation of unstable models is not very meaningful in most cases, a stabilizing controller is added for the simulation. Here, the natural choice is the same PD feedback controller as used with the real helicopter. The comparison of the model outputs to the original helicopter output is shown in figure 7.

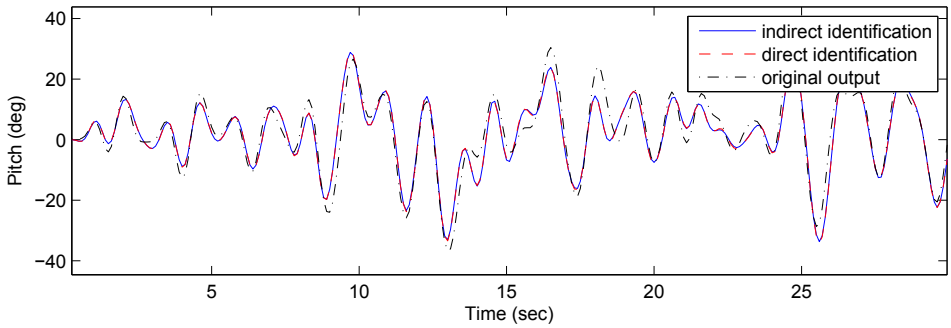


Fig. 7. Simulated closed loop response of models from direct and indirect approach compared to experimental measurement

Judging from the predicted outputs of both models they seem almost identical, as it is even difficult to distinguish between both model outputs. Both are not perfectly tracking the real output but it seems that decent models have been acquired. In figure 8 the bode plots of both

open loop models are shown and this illustrates that the models are not as similar as it had seemed in the closed loop validation, since the static gain differs in a few orders of magnitude. The high frequency part of the plot is comparable, though.

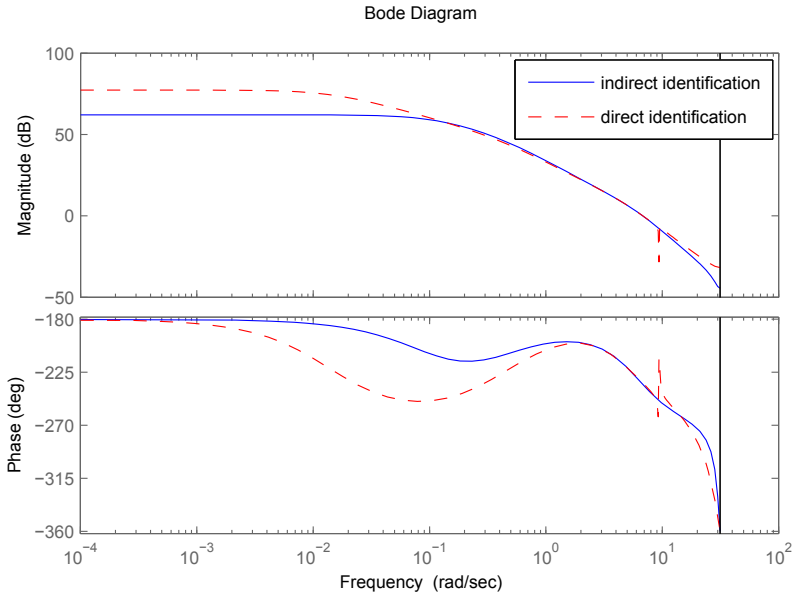


Fig. 8. Bode plot of models from direct and indirect approach

This correlates with the spectra of $r(t)$ and $u(t)$ since they are similar for higher frequencies, too. Taking a close look at the step responses in figure 9 of both stabilized open loop models, although looking similar over all, the model of the direct approach shows a small oscillation for a long time period which seems negligible at first.
To see the consequences of the differences in the models they have to be used in a controller design process and tested on the actual plant. Figure 10 shows the responses of the helicopter's pitch axis to a rectangular reference stabilized by two LQG controllers. Both controllers were designed with the same parameters differing only in the employed plant models.
The controller designed with the model of the indirect approach performs well and is also very robust to manual disturbances. In contrast the LQG controller designed with the model of the direct approach even establishes a static oscillation indicating that the model is not a good representation of the real plant. During all identification approaches the indirect method performed superiorly, which led to the conclusion that the direct approach is not ideal for our setup.

2.3 Linear Identification Results

With the bad results for the direct identification in the SISO case the MIMO identification was attempted with the indirect approach only.
The output of a MIMO model computed from a data set with usage of setpoints as described in 2.1.3 is shown in figure 11. The model used for the output in figure 11 is a state space model of order 16 computed with the prediction error/maximum likelihood (PEM) method of the

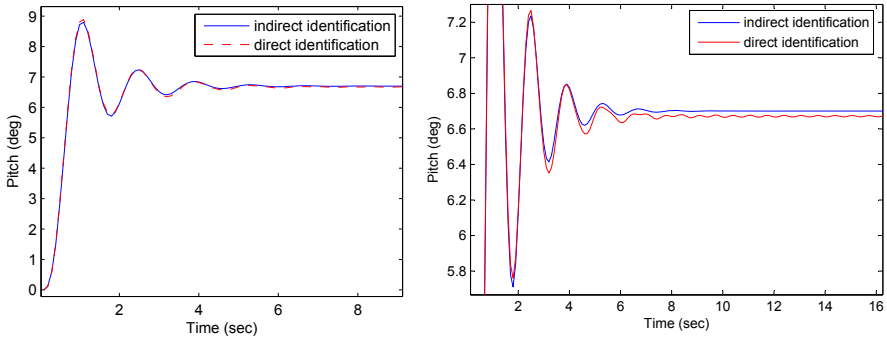


Fig. 9. Simulated step response of models from direct and indirect approach (plotted at different scales)

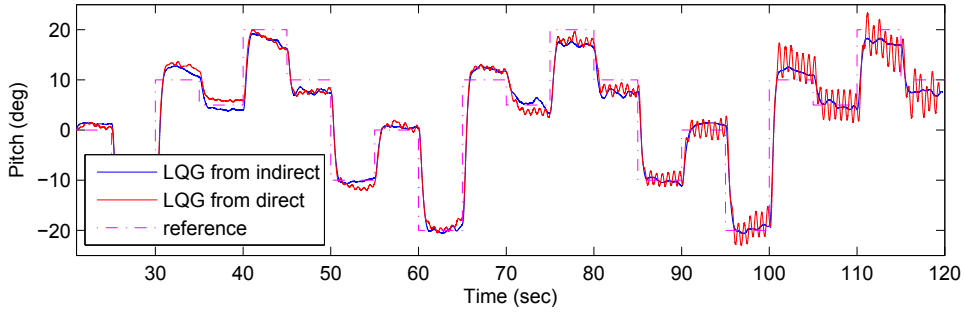


Fig. 10. Experimental results of controllers tracking a rectangular reference on the pitch axis. The LQG controller design was done with models from the direct and indirect approach respectively.

identification toolbox in Matlab. This method uses an iterative search starting at the result of the subspace-method. Other methods like MIMO ARX or directly the sub-space method also yielded good results.
From the model output it can be seen that the characteristics of the model seem to resemble the real ones correctly. During more dynamic maneuvers a discrepancy between the measurement and the prediction becomes visible, though.

2.4 Neural Networks for System Identification

Opposed to the common linear plant models with widely spread structures like ARX (AutoRegressive with eXogenous input), ARMAX (AutoRegressive Moving Average with eXogenous input) or state space models the use of neural networks for system identification is a relatively new approach. Traditionally the identification of models with neural networks falls in the category of black box modelling as typically very few information about the system can be incorporated in the process of identification. Neural networks as they are used most often are very general approximators which can be trained to resemble any given function (given that the network complexity is sufficient). Many different approaches and network structures

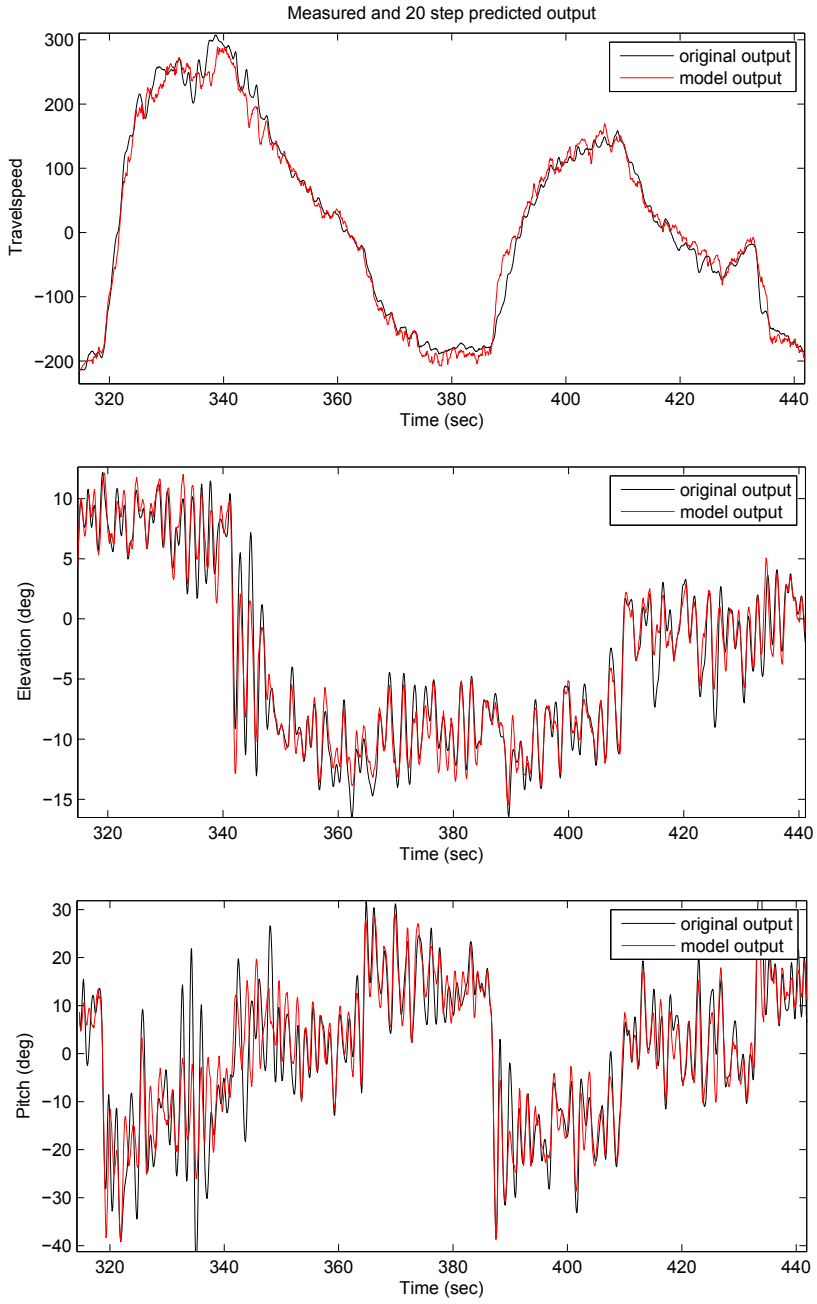


Fig. 11. 20-step ahead prediction output of the linear model for a validation data set

exist. For an introduction to the field of neural networks the reader is referred to Engelbrecht (2002). The common structures and specifics of neural networks for system identification are examined in Nørgaard et al. (2000).

2.4.1 Network Structure

The network that was chosen as nonlinear identification structure in this work is of NNARX format (Neural Network ARX, corresponding to the linear ARX structure), as depicted by figure 12. It is comprised of a multilayer perceptron network with one hidden layer of sigmoid units (or tanh units which are similar) and linear output units. In particular this network structure has been proven to have a universal approximation capability (Hornik et al., 1989). In practice this is not very relevant knowledge though, since no statement about the required number of hidden layer units is made. Concerning the total number of neurons it may still be advantageous to introduce more network layers or to introduce higher order neurons like product units than having one big hidden layer.

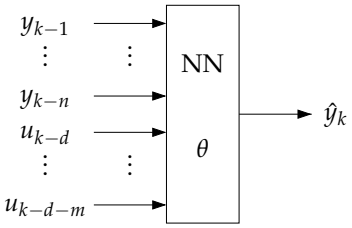


Fig. 12. SISO NNARX model structure

The prediction function of a general two-layer network with tanh hidden layer and linear output units at time k of output l is

$$\hat{y}_l(k) = \sum_{j=1}^{s^1} w_{lj}^2 \tanh\left(\sum_{i=1}^{r} w_{ji}^1 \varphi_i(k) + w_{j0}^1\right) + w_{l0}^2 \tag{3}$$

where w_{ji}^1 and w_{j0}^1 are the weights and biases of the hidden layer, w_{lj}^2 and w_{l0}^2 are the weights and biases of the output layer respectively and $\varphi_i(k)$ is the ith entry of the network input vector (regression vector) at time k which contains past inputs and outputs in the case of the NNARX structure. The choice of an appropriate hidden layer structure and input vector are of great importance for satisfactory prediction performance. Usually this decision is not obvious and has to be determined empirically. For this work a brute-force approach was chosen, to systematically explore different lag space and hidden layer setups, as illustrated in figure 13. From the linear system identification can be concluded that significant parts of the dynamics can be described by linear equations approximately. This knowledge can pay off during the identification using neural networks. If only sigmoid units are used in the hidden layer the network is not able to learn linear dynamics directly. It can merely approximate the linear behavior which would be wasteful. Consequently in this case it is beneficial to introduce linear neurons to the hidden layer. The benefits are twofold as training speed is greatly improved when using linear units (faster convergence) and the linear behavior can be learned "natively". Since one linear neuron in the hidden layer can represent a whole difference equation for an output the number of linear neurons should not exceed the number of system outputs.

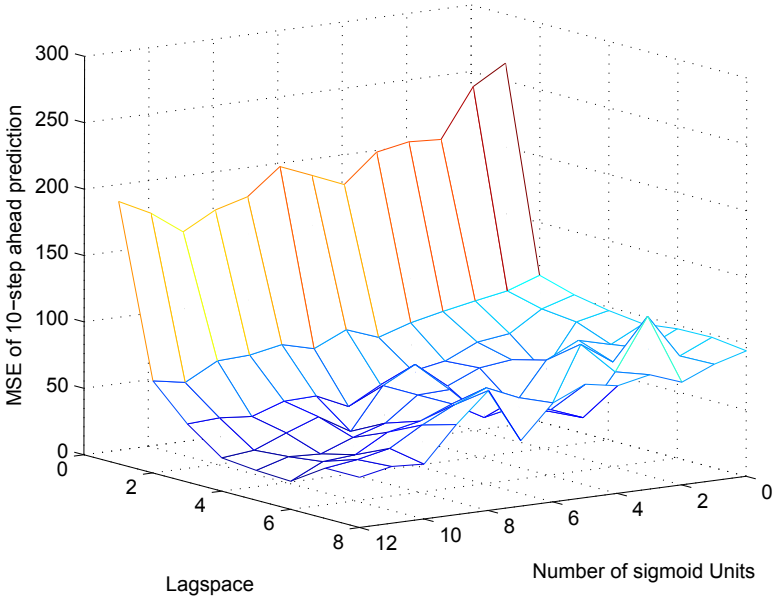


Fig. 13. Comparison of network structures according to their MSE of the 10-step ahead prediction using a validation data set (all networks include three linear units in the hidden layer). Each data point reresents the best candidate network of 10 independent trainings.

The final structure that was chosen according to the results depicted by figure 13 includes three linear and twelve sigmoid units in the hidden layer with a lag space of six for both inputs and the three outputs. For this network accordingly $((2+3)\cdot 6+1)\cdot(12+3)+(12+3+1)\cdot 3 = 513$ weights had to be optimized.

2.4.2 Instantaneous Linearization

To implement APC, linearized MIMO-ARX models have to be extracted from the nonlinear NNARX model in each sampling instant. The coefficients of a linearized model can be obtained by the partial derivative of each output with respect to each input (Nørgaard et al., 2000). Applying the chain rule to (3) yields

$$\frac{\partial \hat{y}_l(k)}{\partial \varphi_i(k)} = \sum_{j=1}^{s^1} w_{lj}^2 w_{ji}^1 \left[1 - \tanh^2\left(\sum_{i=1}^{r} w_{ji}^1 \varphi_i(k) + w_{j0}^1\right)\right] \tag{4}$$

for tanh units in the hidden layer. For linear hidden layer units in both the input and the output layer one yields

$$\frac{\partial \hat{y}_l(k)}{\partial \varphi_i(k)} = \sum_{j=1}^{s^1} w_{lj}^2 w_{ji}^1 \,. \tag{5}$$

2.4.3 Network Training

All networks were trained with *Levenberg Marquardt Backpropagation* (Hagan & Menhaj, 1994). Due to the monotonic properties of linear and sigmoid units, networks using only these unit types have the inherent tendency to have only few local minima, which is beneficial for local optimization algorithms like backpropagation. The size of the final network (513 weights) that was used in this work even makes global optimization techniques like *Particle Swarm Optimization* or *Genetic Algorithms* infeasible. Consequently for a network of the presented size, higher order units such as product units cannot be incorporated due to the increased amount of local minima, requiring global optimization techniques (Ismail & Engelbrecht, 2000).
But also with only sigmoid units, based on the possibility of backpropagation getting stuck in local minima, always a set of at least 10 networks with random initial parameters were trained. To minimize overfitting a *weight decay* of $D = 0.07$ was used. The concept of regularization to avoid overfitting using a weight decay term in the cost function is thoroughly explored by Nørgaard et al. (2000).

2.5 Nonlinear Identification Results

For the nonlinear identification the same excitation signal and indirect measurement setup was used as for the linear identification. Thus a stabilized closed-loop model was acquired. The controller that was inevitably identified along with the unstable plant model cannot be removed from the model analytically. In section 2.2.2 we showed that the stabilizing controller will not hinder the final control performance in the case of APC, though.
The prediction of the finally chosen network with a validation data set is depicted in figure 14. If one compares the neural network prediction with the prediction of the linear model in figure 11 it is obvious that the introduction of nonlinear neurons benefited the prediction accuracy. This is underlined by figure 13 also visualizing a declining prediction error for increasing sigmoid unit numbers. Whether the improvements in the model can be transferred to an improved controller remains to be seen, though.

2.6 Conclusion

This section demonstrated successful experiment design for an unstable nonlinear MIMO system and showed some pitfalls that may impede effective identification. The main approaches to closed loop identification have been presented and compared by means of the helicopters unstable pitch axis. It was shown that the identification of unstable systems can be just as successful as for stable systems if the presented issues are kept in mind. Both linear and nonlinear identifications can be regarded as successful, although the nonlinear predictions outperform the linear ones.

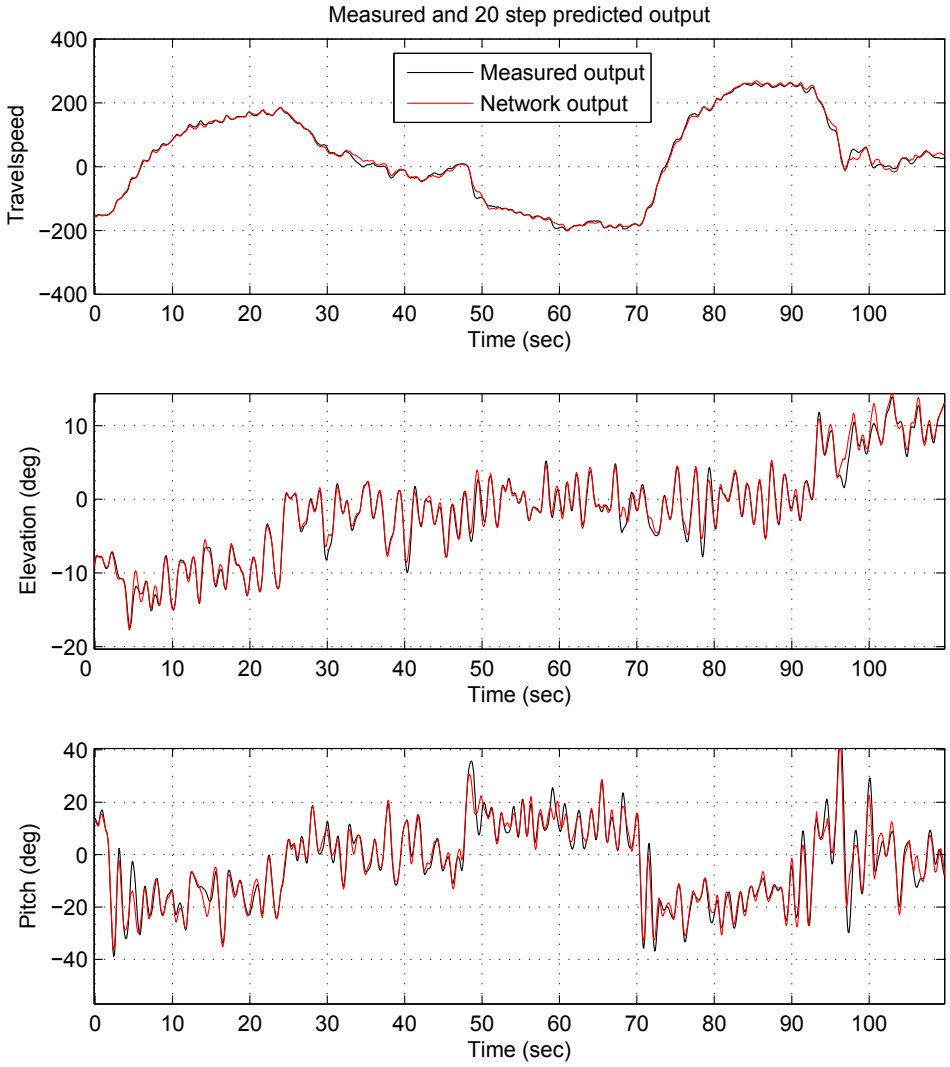


Fig. 14. 20-step ahead prediction output of the best network for a validation data set

3. Approximate Model Predictive Control

The predictive controller that is discussed in this chapter is a nonlinear adaptation of the popular *Generalized Predictive Control* (GPC), proposed in (Clarke et al., 1987a;b). *Approximate (Model) Predictive Control* (APC) as proposed by Nørgaard et al. (2000) uses the GPC principle on instantaneous linearizations of a neural network model. Although presented as a single-input single-output (SISO) algorithm, its extension to the multi-input multi-output (MIMO) case with MIMO-GPC (Camacho & Borbons, 1999) is straightforward. The scheme is visualized in figure 15.

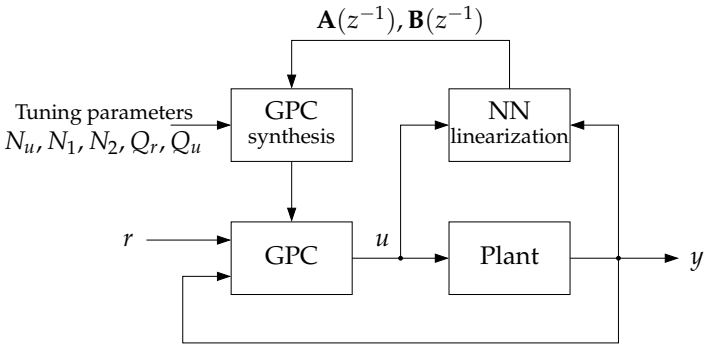


Fig. 15. Approximate predictive control scheme

The linearized model that is extracted from the neural network at each time step (as described in section 2.4.2) is used for the computation of the optimal future control sequence according to the objective function:

$$\begin{aligned} J(k) \quad &= \quad \sum_{i=N_1}^{N_2} \Big(r(k+i) - \hat{y}(k+i)\Big)^T Q_r \Big(r(k+i) - \hat{y}(k+i)\Big) \\ &+ \quad \sum_{i=1}^{N_u} \Delta u^T(k+i-1)\; Q_u\; \Delta u(k+i-1) \end{aligned} \tag{6}$$

where N_1 and N_2 are the two prediction horizons which determine how many future samples the objective function considers for minimization and N_u denotes the length of the control sequence that is computed. As common in most MPC methods, a *receding horizon strategy* is used and thus only the first control signal that is computed is actually applied to the plant to achieve loop closure.

A favourable property of quadratic cost functions is that a closed-form solution exists, enabling its application to fast processes under hard realtime constraints (since the execution time remains constant). If constraints are added, an iterative optimization method has to be used in either way, though. The derivation of MIMO-GPC is given in the following section for the sake of completeness.

3.1 Generalized Predictive Control for MIMO Systems

In GPC, usually a modified ARX (AutoRegressive with eXogenous input) or ARMAX (AutoRegressive Moving Average with eXogenous input) structure is used. In this work a structure like

$$\mathbf{A}(z^{-1})y(k) = \mathbf{B}(z^{-1})u(k) + \frac{1}{\Delta}e(k) \tag{7}$$

is used for simplicity, with $\Delta = 1 - z^{-1}$ where $y(k)$ and $u(k)$ are the output and control sequence of the plant and $e(k)$ is zero mean white noise. This structure is called ARIX and basically extends the ARX structure by integrated noise. It has a high relevance for practical applications as the coloring polynomials for an integrated ARMAX structure are very difficult to estimate with sufficient accuracy, especially for MIMO systems (Camacho & Borbons, 1999). The integrated noise term is introduced to eliminate the effects of step disturbances.
For an n-output, m-input MIMO system $\mathbf{A}(z^{-1})$ is an $n \times n$ monic polynomial matrix and $\mathbf{B}(z^{-1})$ is an $n \times m$ polynomial matrix defined as:

$$\begin{aligned} \mathbf{A}(z^{-1}) &= I_{n\times n} + A_1 z^{-1} + A_2 z^{-2} + ... + A_{n_a} z^{-n_a} \\ \mathbf{B}(z^{-1}) &= B_0 + B_1 z^{-1} + B_2 z^{-2} + ... + B_{n_b} z^{-n_b} \end{aligned}$$

The output $y(k)$ and noise $e(k)$ are $n \times 1$-vectors and the input $u(k)$ is an $m \times 1$-vector for the MIMO case. Looking at the cost function from (6) one can see that it is already in a MIMO compatible form if the weighting matrices Q_r and Q_u are of dimensions $n \times n$ and $m \times m$ respectively. The SISO case can easily be deduced from the MIMO equations by inserting $n = m = 1$ where $\mathbf{A}(z^{-1})$ and $\mathbf{B}(z^{-1})$ degenerate to polynomials and $y(k)$, $u(k)$ and $e(k)$ become scalars.

To predict future outputs the following Diophantine equation needs to be solved:

$$I_{n\times n} = \mathbf{E}_j(z^{-1})(\mathbf{A}(z^{-1})\Delta) + z^{-j}\mathbf{F}_j(z^{-1}) \tag{8}$$

where $\mathbf{E}_j(z^{-1})$ and $\mathbf{F}_j(z^{-1})$ are both unique polynomial matrices of order $j-1$ and n_a respectively. This special Diophantine equation with $I_{n\times n}$ on the left hand side is called Bizout identity, which is usually solved by recursion (see Camacho & Borbons (1999) for the recursive solution). The solution to the Bizout identity needs to be found for every future sampling point that is to be evaluated by the cost function. Thus $N_2 - N_1 + 1$ polynomial matrices $\mathbf{E}_j(z^{-1})$ and $\mathbf{F}_j(z^{-1})$ have to be computed. To yield the j step ahead predictor, (7) is multiplied by $\mathbf{E}_j(z^{-1})\Delta z^j$:

$$\mathbf{E}_j(z^{-1})\Delta\mathbf{A}(z^{-1})y(k+j) = \mathbf{E}_j(z^{-1})\mathbf{B}(z^{-1})\Delta u(k+j-1) + \mathbf{E}_j(z^{-1})e(k+j) \tag{9}$$

which by using equation 8 can be transformed into:

$$y(k+j) = \underbrace{\mathbf{E}_j(z^{-1})\mathbf{B}(z^{-1})\Delta u(k+j-1)}_{past\ and\ future\ inputs} + \underbrace{\mathbf{F}_j(z^{-1})y(k)}_{free\ response} + \underbrace{\mathbf{E}_j(z^{-1})e(k+j)}_{future\ noise} \tag{10}$$

Since the future noise term is unknown the best prediction is yielded by the expectation value of the noise which is zero for zero mean white noise. Thus the expected value for $y(k+j)$ is:

$$\hat{y}(k+j|k) = \mathbf{E}_j(z^{-1})\mathbf{B}(z^{-1})\Delta u(k+j-1) + \mathbf{F}_j(z^{-1})y(k) \tag{11}$$

The term $\mathbf{E}_j(z^{-1})\mathbf{B}(z^{-1})$ can be merged into the new polynomial matrix $\mathbf{G}_j(z^{-1})$:

$$\mathbf{G}_j(z^{-1}) = G_0 + G_1 z^{-1} + \ldots + G_{j-1} z^{-(j-1)} + (G_j)_j z^{-j} + \ldots + (G_{j-1+n_b})_j z^{-(j-1+n_b)}$$

where $(G_{j+1})_j$ is the $(j+1)$th coefficient of $\mathbf{G}_j(z^{-1})$ and n_b is the order of $\mathbf{B}(z^{-1})$. So the coefficients up to $(j-1)$ are the same for all $\mathbf{G}_j(z^{-1})$ which stems from the recursive properties of $\mathbf{E}_j(z^{-1})$ (see Camacho & Borbons (1999)). With this new matrix it is possible to separate the first term of (10) into past and future inputs:

$$\begin{aligned}\mathbf{G}_j(z^{-1})\Delta u(k+j-1) \quad = \quad & \underbrace{G_0\Delta u(k+j-1) + G_1\Delta u(k+j-2) + \ldots + G_{j-1}\Delta u(k)}_{future\ inputs} \\ & + \underbrace{(G_j)_j\Delta u(k-1) + (G_{j+1})_j\Delta u(k-2) + \ldots + (G_{j-1+n_b})_j\Delta u(k-n_b)}_{past\ inputs}\end{aligned}$$

Now it is possible to separate all past inputs and outputs from the future ones and write this in matrix form:

$$\underbrace{\begin{bmatrix}\hat{y}(k+1|k)\\ \hat{y}(k+2|k)\\ \vdots\\ \hat{y}(k+N_u|k)\\ \vdots\\ \hat{y}(k+N_2|k)\end{bmatrix}}_{\hat{\mathbf{y}}} = \underbrace{\begin{bmatrix}G_0 & 0 & \cdots & 0\\ G_1 & G_0 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ G_{N_u-1} & G_{N_u-2} & \cdots & G_0\\ \vdots & \vdots & \ldots & \vdots\\ G_{N_2-1} & G_{N_2-2} & \cdots & G_{N_2-N_u}\end{bmatrix}}_{\mathbf{G}} \underbrace{\begin{bmatrix}\Delta u(k)\\ \Delta u(k+1)\\ \vdots\\ \Delta u(k+N_u-1)\end{bmatrix}}_{\tilde{\mathbf{u}}} + \underbrace{\begin{bmatrix}\mathbf{f}_1\\ \mathbf{f}_2\\ \vdots\\ \mathbf{f}_{N_u}\\ \vdots\\ \mathbf{f}_{N_2}\end{bmatrix}}_{\mathbf{f}} \tag{12}$$

which can be condensed to :

$$\hat{\mathbf{y}} \;=\; \mathbf{G}\tilde{\mathbf{u}} \;+\; \mathbf{f} \tag{13}$$

where $\mathbf{f}$ represents the influence of all past inputs and outputs and the columns of $\mathbf{G}$ are the step responses to future $\tilde{\mathbf{u}}$ (for further reading, see (Camacho & Borbons, 1999)). Since each G_i is an $n \times m$ matrix $\mathbf{G}$ has block matrix structure.

Now that we obtained a j-step ahead predictor form of a linear model this can be used to compute the optimal control sequence with respect to a given cost function (like (6)). If (6) is written in vector form and with (13) one yields:

$$\begin{aligned}J(k) \quad &= \quad (\mathbf{r}-\hat{\mathbf{y}})^T Q_r(\mathbf{r}-\hat{\mathbf{y}}) + \tilde{\mathbf{u}}^T Q_u \tilde{\mathbf{u}} \\ &= \quad (\mathbf{r}-\mathbf{G}\tilde{\mathbf{u}}-\mathbf{f})^T Q_r(\mathbf{r}-\mathbf{G}\tilde{\mathbf{u}}-\mathbf{f}) + \tilde{\mathbf{u}}^T Q_u \tilde{\mathbf{u}}\end{aligned}$$

where

$$\mathbf{r} = [r(k+1),\ r(k+2),\ \ldots,\ r(k+N_2)]^T$$

In order to minimize the cost function $J(k)$ for the future control sequence $\tilde{\mathbf{u}}$ the derivative $dJ(k)/d\tilde{\mathbf{u}}$ is computed and set to zero:

$$\begin{aligned}\frac{dJ(k)}{d\tilde{\mathbf{u}}} &= 0 \\ &= 2\mathbf{G}^T Q_r \mathbf{G}\tilde{\mathbf{u}} - 2\mathbf{G}^T Q_r(\mathbf{r}-\mathbf{f}) + 2Q_u\tilde{\mathbf{u}}\end{aligned}$$

$$(\mathbf{G}^T Q_r \mathbf{G} + Q_u)\tilde{\mathbf{u}} = \mathbf{G}^T Q_r(\mathbf{r}-\mathbf{f}) \tag{14}$$

$$\tilde{\mathbf{u}} = \underbrace{(\mathbf{G}^T Q_r \mathbf{G} + Q_u)^{-1}\mathbf{G}^T Q_r}_{K}(\mathbf{r}-\mathbf{f}) \tag{15}$$

Thus the optimization problem can be solved analytically without any iterations which is true for all quadratic cost functions in absence of constraints. This is a great advantage of GPC since the computation effort can be very low for time-invariant plant models as the main computation of the matrix K can be carried out off-line. Actually just the first m rows of K must be saved because of the receding horizon strategy using only the first input of the whole sequence $\tilde{\mathbf{u}}$. Therefore the resulting control law is linear, each element of K weighting the predicted error between the reference and the free response of the plant.
Finally for a practical implementation of APC one has to bear in mind that the matrix $(\mathbf{G}^T Q_r \mathbf{G} + Q_u)$ can be singular in some instances. In the case of GPC this is not a problem since the solution is not computed online. For APC in this work a special Gauss solver was used which assumes zero control input where no unambiguous solution can be found.

3.2 Reducing Overshoot with Reference Filters

With the classic quadratic cost function it is not possible to control the overshoot of the resulting controller in a satisfying manner. If the overshoot needs to be influenced one can choose three possible ways. The obvious and most elaborate way is to introduce constraints, however the solution to the optimization problems becomes computationally more expensive. Another possible solution is to change the cost function, introducing more tuning polynomials, as mentioned by Nørgaard et al. (2000) referring to *Unified Predictive Control*.
A simple but yet effective way to reduce the overshoot for any algorithm that minimizes the standard quadratic cost function (like LQG, GPC or APC) is to introduce a reference prefilter which smoothes the steep areas like steps in the reference. For the helicopter, the introduction of prefilters made it possible to eliminate overshoot completely, retaining comparably fast rise times. The utilized reference prefilters are of first order low-pass kind

$$G_{RF} = \frac{1-l}{1-lz^{-1}}$$

which have a steady-state gain of one and can be tuned by the parameter l to control the smoothing.

3.3 Improving APC Performance by Parameter Filtering

A problem with APC is that a network that has a good prediction capability does not necessarily translate into a good controller, as for APC the network dynamics need to be smooth for consistent linear models which is not a criterion the standard Levenberg-Marquardt backpropagation algorithm trains the network for. A good way to test whether the network dynamics are sufficiently smooth is to start a simulation with the same neural network as the plant and

as the predictive controllers system model. If one sees unnecessary oscillation this is good evidence that the network dynamics are not as smooth as APC desires for optimal performance. The first solution to this is simply training more networks and test whether they provide a better performance in the simulation.

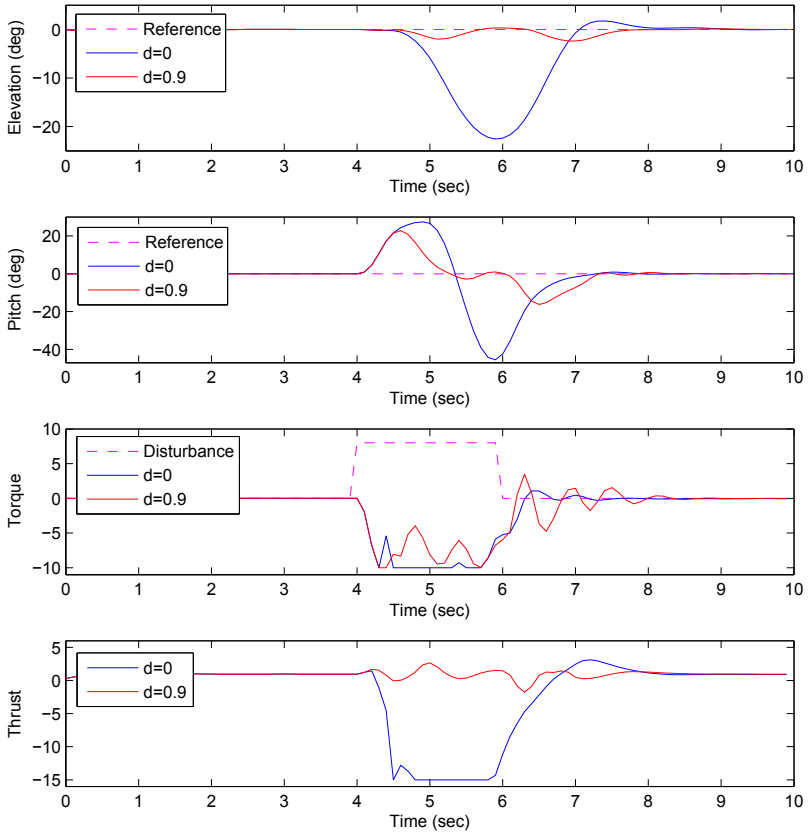


Fig. 16. Simulation results of disturbance rejection with parameter filtering. Top two plots: Control outputs. Bottom two plots : Control inputs

In the case of the helicopter a neural network with no unnecessary oscillation in the simulation could not be found, though. If one assumes sufficiently smooth nonlinearities in the real system, one can try to manually smooth linearizations of the neural network from sample to sample, as proposed in (Witt et al., 2007). Since APC is not able to control systems with nonlinearities that are not reasonably smooth within the prediction horizon anyway, the idea of smoothing the linearizations of the network does not interfere with the basic idea of APC being able to control nonlinear systems. It is merely a means to flatten out local network areas where the linearized coefficients start to jitter within the prediction horizon.

This idea has been realized by a first order low-pass filter:

$$G_{PF} = \frac{1-d}{1-dz^{-1}}$$

with tuning parameter d. When applied to the polynomial matrix $\mathbf{A}(z^{-1})$, (3.3) results in the following formula:

$$\hat{\mathbf{A}}_k(z^{-1}) = (1-d)\mathbf{A}_k(z^{-1}) + d\hat{\mathbf{A}}_{k-1}(z^{-1})$$

where $\hat{\mathbf{A}}_k(z^{-1})$ contains the filtered polynomial coefficients $\mathbf{A}_k(z^{-1})$. For prediction horizons around $N_2 = 10...20$ a good starting value for the tuning parameter d was found to be 0.9, however this parameter depends on the sampling rate.
If the filtering parameter d is increased, the adaptivity of the model decreases and shifts towards a linear model (in the case of $d = 1$). The importance of parameter filtering in the case of the helicopter is displayed in figure 16 where an input disturbance acts on the torque input of a standard APC controller and the parameter filtered version.

4. Experimental Results

During the practical experiments the setup shown in figure 17 was used. It necessarily incorporates the stabilizing proportional derivative controller that is included in our nonlinear model from section 2. The sampling time was 0.1 seconds and the experiments were run on a 1 GHz Intel Celeron CPU. All APC related algorithms were implemented in C++ to achieve the computational performance that was necessary to be able to compute the equations in realtime on this system at the given sampling rate.

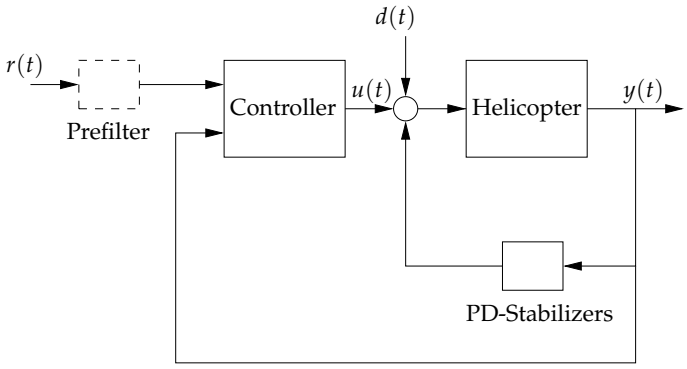


Fig. 17. Control setup for helicopter with inner stabilizing control loop and reference prefilter.

For our experiments only the control of the pitch and elevation axis was considered as the travelspeed axis has significantly longer rise times (about factor 15) than the other two axes, making predictive control with the same sampling rate and prediction horizons impractical. To control the travelspeed axis in this setup one could design an outer cascaded control loop with a slower sampling rate, but this is beyond the scope of this work.
APC as well as GPC were tuned with the same 5 parameters, being the horizons N_1, N_2, N_u and the weighting matrices Q_r and Q_u. The tuning was done as suggested in (Clarke et al., 1987a;b) and resulted in $N_1 = 1$, $N_2 = 10$, $N_u = 10$ and the weighting matrices $Q_r =$

$diag(0,1,1)$ and $Q_u = diag(20,10)$. The choice of Q_r disables weighting for the first output which is the uncontrolled travelspeed-axis.
The computational limits of the test platform were found at horizons of $N_2 = N_u = 20$ which does not leave too much headroom.

4.1 Tracking Performance

APC has been benchmarked with both tracking and disturbance rejection experiments. We also designed a linear GPC and an integrator augmented LQG controller for comparison. The benchmark reference signals are designed to cover all operating ranges for all outputs. All controllers were benchmarked with identically parameterized reference prefilters to eliminate overshoot.
In figure 18 it can be seen that LQG achieves a suitable performance only for the pitch axis while performance on the elevation axis is much poorer than both APC and GPC. For both outputs, APC yields slightly better performance than linear GPC which is most visible for the large reference steps on the more nonlinear elevation axis. However looking at the plant input signals one can see that the APC signals have less high frequency oscillation than for GPC which is also an important issue because of actuator stress in practical use. Parameter filtering does not change the response to the benchmark sequence up to about $d = 0.9$ but significantly improves the performance for disturbance rejection as will be shown in the next section.

4.2 Disturbance Rejection

The performance of the benchmarked controllers becomes more diverse when disturbance rejection is considered. In figure 19 one can see the response to disturbances applied to the two inputs. Again LQG can be tuned to satisfactory performance only for the pitch axis, but also the standard APC and GPC do not give satisfying results. Considering input disturbance rejection the standard APC even shows a lower stability margin than GPC. The introduction of parameter filtering however changes this aspect significantly. With parameter filtering of $d = 0.9$ the stability margin of APC becomes much larger than the one of GPC and it can be seen in the plot that it shows the best disturbance response of all tested controllers - especially note the low input signal amplitude, while superiorly managing the disturbance.

4.3 Conclusion

With this work it has been shown that MIMO APC for a fast process is indeed feasible with mid-range embedded hardware. It was found that standard APC can be problematic if the network dynamics are unsmooth. For this purpose, parameter filtering was presented as an improvement to the standard APC implementation with which it was possible to enhance the stability margin and overall performance of APC in the face of disturbances significantly. Still the acquisition of a decent model should be the first step before one should tune the performance with parameter filtering, since it remains the most important constituent to good control performance.
Finally although the helicopter is not a highly nonlinear system, APC with parameter filtering was able to outperform the linear GPC while being the more generally applicable control scheme.

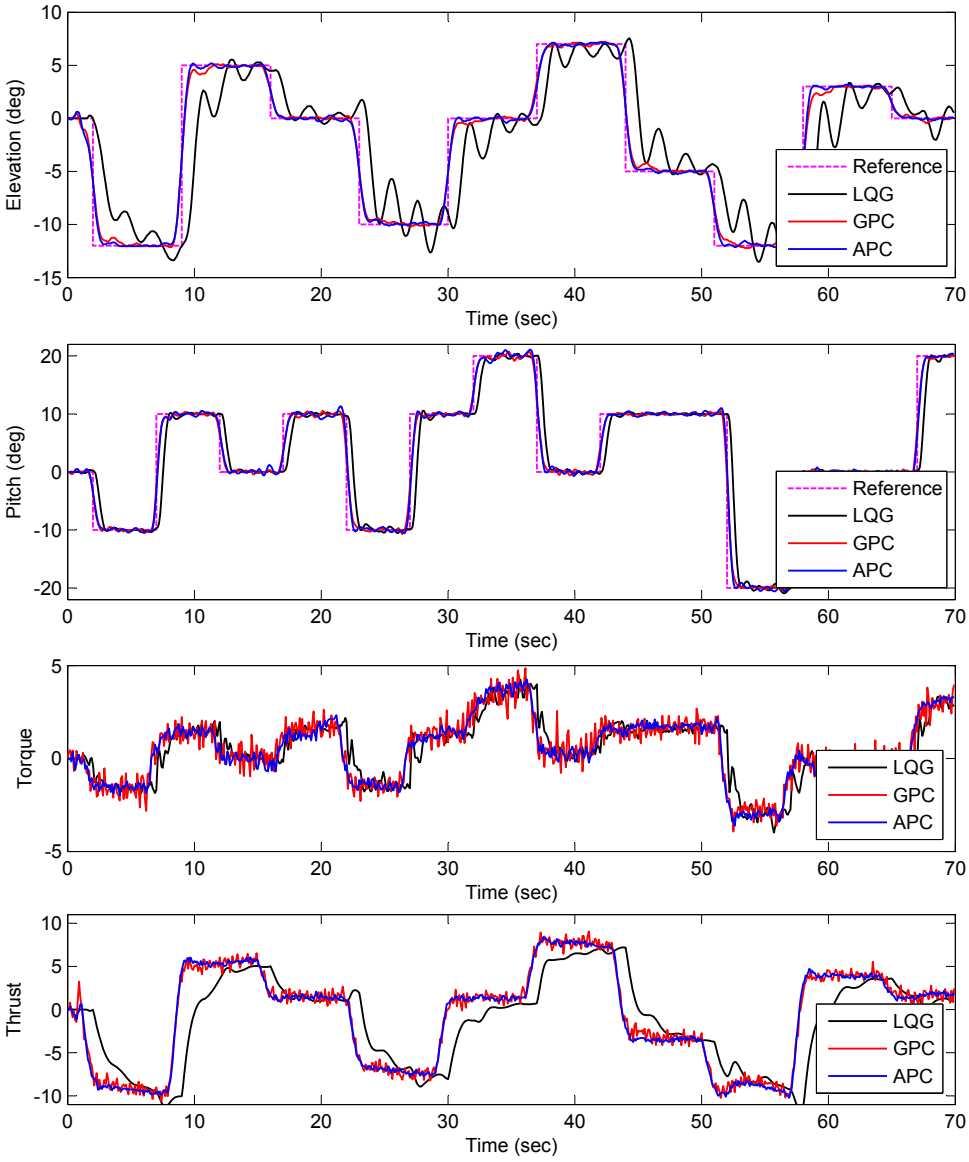


Fig. 18. Experimental results for tracking performance of APC compared to GPC and LQG. Top two plots: Control outputs. Bottom two plots: Control inputs.

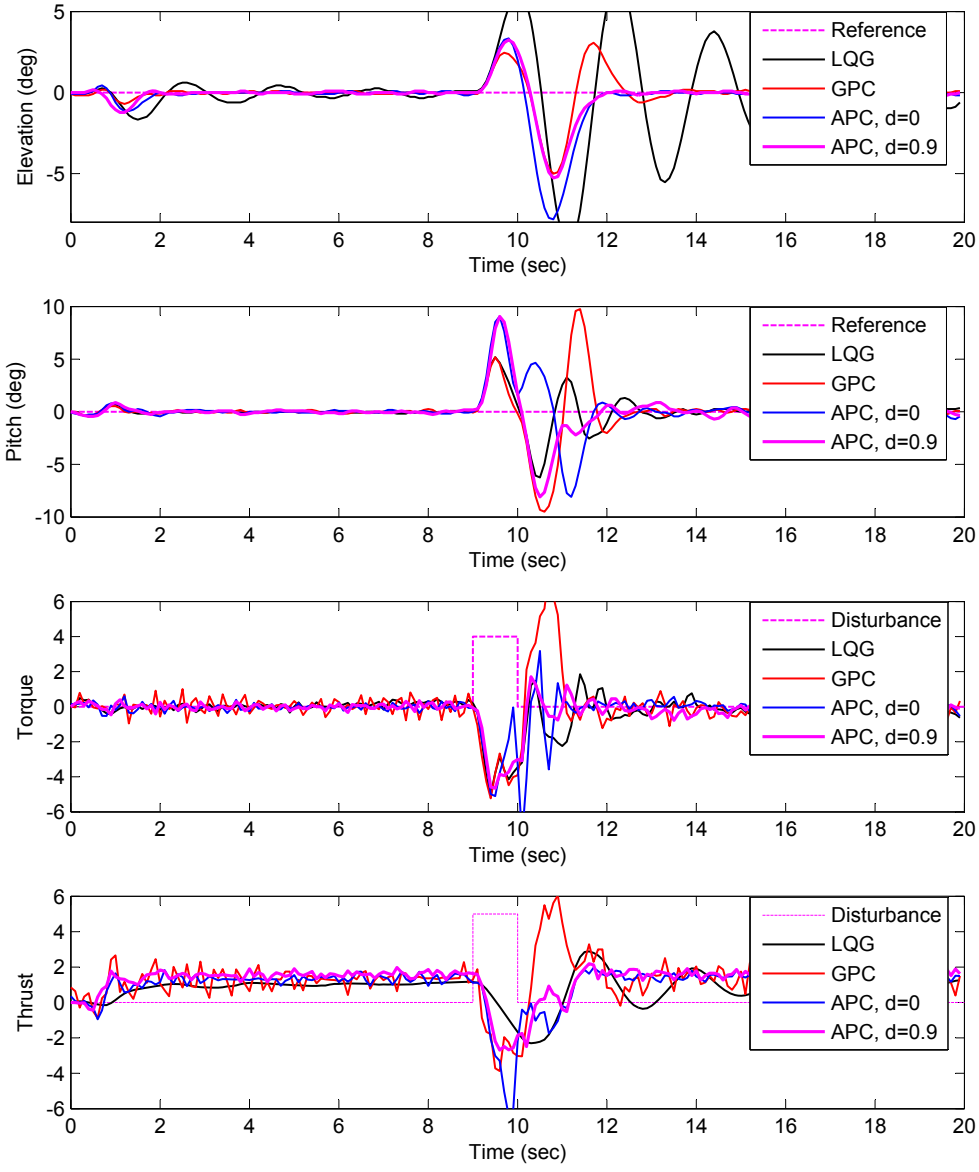


Fig. 19. Experimental results for disturbance rejection performance. Top two plots: Control outputs. Bottom two plots: Control inputs.

5. References

Camacho, E. & Borbons, C. (1999). *Model Predictive Control*, Springer-Verlag, London.

Clarke, D., Mohtadi, C. & Tuffs, P. (1987a). Generalized Predictive Control – Part I. The basic algorithm, *Automatica* **23**(2): 137–148.

Clarke, D., Mohtadi, C. & Tuffs, P. (1987b). Generalized Predictive Control – Part II. Extension and Interpretations, *Automatica* **23**(2): 149–160.

Engelbrecht, A. (2002). *Computational Intelligence: An Introduction*, Halsted Press New York, NY, USA.

Evan, C., Rees, D. & Borrell, A. (2000). Identification of aircraft gas turbine dynamics using frequency-domain techniques, *Control Engineering Practice* **8**: 457–467.

Hagan, M. & Menhaj, M. (1994). Training feedforward networks with the Marquardt algorithm, *IEEE transactions on Neural Networks* **5**(6): 989–993.

Hornik, K., Stinchcombe, M. & White, H. (1989). Multilayer Feedforward Networks are Universal Approximators, *Neural networks* **2**(5): 359–366.

Ismail, A. & Engelbrecht, A. (2000). Global Optimization Algorithms for Training Product Unit Neural Networks, *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, Vol. 1, Como, Italy, pp. 132–137.

Ljung, L. (1999). *System Identification Theory for the User*, 2nd edn, Prentice Hall PTR, Upper Saddle River, NJ.

Maciejowski, J. (2002). *Predictive Control with Constraints*, Prentice Hall.

Mu, J. & Rees, D. (2004). Approximate model predictive control for gas turbine engines, *Proceedings of the 2004 American Control Conference*, Boston, Massachusetts, USA, pp. 5704–2709.

Nørgaard, M., Ravn, O., Poulsen, N. K. & Hansen, L. K. (2000). *Neural Networks for Modelling and Control of Dynamic Systems*, Springer-Verlag, London, UK.

Pico, J. & Martinez, M. (2002). *Iterative Identification and Control*, Springer-Verlag London, chapter System Identification. Performance and Closed-loop Issues.

Quanser Inc. (2005). *3 DOF Helicopter System*, www.quanser.com.

Witt, J., Boonto, S. & Werner, H. (2007). Approximate model predictive control of a 3-dof helicopter, *Proceedings of the 46th IEEE Conference on Decision and Control*, New Orleans, LA, USA, pp. 4501–4506.

7

Multi-objective Nonlinear Model Predictive Control: Lexicographic Method

Tao ZHENG, Gang WU, Guang-Hong LIU and Qing LING
University of Science and Technology of China
China

1. Introduction

The design of most process control is essentially a dynamic multi-objective optimization problem (Meadowcroft *et al.*, 1992), sometimes with nonlinear characters, and in which both economic benefit and social benefit should be considered. Commonly speaking, there are contradictory objectives such as quantity of products, quality of products, safety of manufacturing, cost of manufacturing, environment protection and so on. Since the different relative importance of these objectives cannot be ignored in the process of the controller design, we should manage the different priority of each objective correctly and exactly. Therefore, multivariable process control could be formulated as a complicated dynamic multi-objective optimization problem.

Traditionally, a multi-objective control problem could be transformed into a single-objective dynamic optimization with the quadratic objective function, where the weights denote the different relative importance of different objectives. This method is easy to understand, but the value of the weight coefficients usually could be only decided by try-and-error method, based on engineering experiences, repeating simulations and other information, while there is no accurate theoretical analysis of these weight coefficients yet. So it can be seen that, the design process of the traditional method is complicated and time-consuming indeed. Especially, when the situation of manufacturing changes (such as sudden load increasing of a power supplier and so on), it is very hard for operators to renew the weights rapidly. Therefore, a new framework of multi-objective controller is desired, it should be driven by the relative importance of different objectives, which reflect the practical requirement of control problems, and it also should be convenient to redesign for engineers and operators, when the values or priorities of the objectives are changed.

Using lexicographic method, which also called completely stratified method, Meadowcroft *et al.* proposed a priority-driven framework of controller: Modular Multivariable Controller (MMC), and analyzed its steady-state properties (Meadowcroft *et al.*, 1992). It sorts objectives sequentially according to their relative importance, and then satisfies them as many as possible in the corresponding control modules by the order as Fig. 1., where one module handles with only one objective. Later, because of its advantages, researchers have extended MMC to the dynamic optimization of linear systems with model predictive control (MPC) and other controllers in past years (Ocampo-Martinez *et al.*, 2008, Wu *et al.*, 2000).

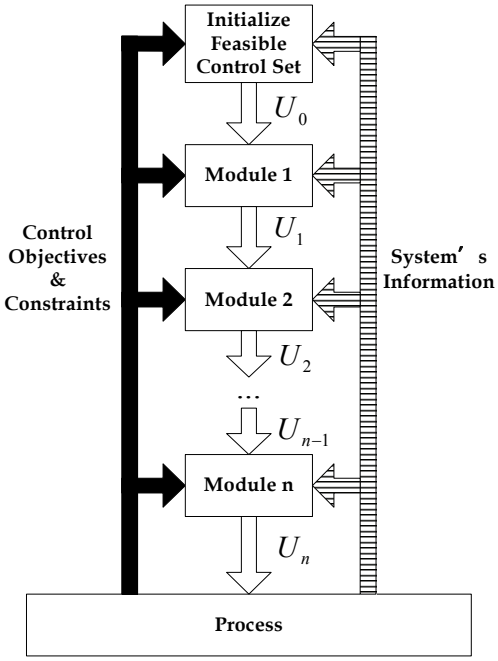


Fig. 1. Lexicographic structure of Modular Multivariable Controller

While has the mentioned advantages, the lexicographic structure still has some serious problems. First, in this structure, the priorities of objectives are absolute and rigid, if an objective cannot be completely satisfied (usually a objective with a setpoint form or an extremum form), the objectives with lower priorities than it will not be considered any more, even if they can be satisfied without any bad influence on other objectives. Second, in some practical cases, it is hard to distinguish the difference on priorities between some "parallel objectives", and it is also not necessary indeed. In practical need, the number of priorities is no need to equals to the number of objectives, it can be smaller, that means a certain priority may have several objectives. So sometimes, the partially stratified structure is more flexible than completely stratified structure (lexicographic structure), the number of priorities could be determined by the essential control problem, and the objectives with relatively lower importance can be handled in a same priority together for simplicity.

Besides the structure of the controller, the control algorithm is also important in multi-objective control nowadays. Since the control demand of modern process industry is heightening continuously, nonlinearity of systems cannot be ignored in controller design, to utilize the advantages of MPC in process control, nonlinear model predictive control (NMPC) now are developing rapidly (Alessio & Bemporad, 2009, Cannon, 2004). Naturally, for multi-objective NMPC in many industrial cases, the priority-driven method is also necessary. We have tried to combine lexicographic structure (or partially stratified structure) and NMPC directly, as dynamic MMC of linear systems (Ocampo-Martinez *et al.*, 2008, Wu *et al.*, 2000). But the nonlinear character makes it difficult to obtain analytic solution of control problem, and the modular form for stratified structure seems to be too complex for

nonlinear systems in some extent. Both these facts lead us to find a new way for the nonlinear multi-objective control problem. Genetic algorithm (GA) now is recognized as an efficient computing means for single-objective NMPC already (Yuzgec *et al.*, 2006), and it also can be used to solve lexicographic optimization (Coello, 2000). So, in this chapter, a series of dynamic coefficients are used to make up a combined fitness function of GA, which makes GA be able to handle lexicographic optimization or partially stratified optimization in multi-objective NMPC. It can solve the nonlinear multi-objective control problem in the same way as MMC, but with a simple structure and much little computational load.

Since the partially stratified structure could be modified from lexicographic structure easily (or lexicographic structure can be seen as a special case of partially stratified structure), in this chapter, we will introduce lexicographic method as the main content, then the corresponding content of partially stratified method can be obtained directly. The rest of this chapter is organized as follow, Section 2 will introduce the basic theory of lexicographic optimization and partially stratified optimization, then the modified GA for them will be proposed in Section 3, lexicographic NMPC and partially stratified NMPC based on the proposed GA will be studied in Section 4, using the control problem of a two-tank system as a case study. At last, conclusions and acknowledgements will be done in Section 5.

2. Lexicographic optimization and partially stratified optimization

2.1 Lexicographic optimization

Lexicographic optimization is a strategy of multivariable optimization derived from priority-driven thought, without loss of generality, we just considers the minimization of multi-objective problem in this chapter.

Suppose a complex goal $g=\{g_1,g_2,\cdots,g_n\}$ contains n objectives, and the subscript also describe the relative importance of each objective, where g_1 is the most important one and g_{i-1} is always more important than g_i. The solution $g^{(1)}=\{g_1^{(1)},g_2^{(1)},\cdots,g_n^{(1)}\}$ is better than the solution $g^{(2)}=\{g_1^{(2)},g_2^{(2)},\cdots,g_n^{(2)}\}$, if and only if $g_k^{(1)}<g_k^{(2)}$ and $g_i^{(1)}=g_i^{(2)}=\min g_i$ hold for certain $k\le n$ and all $i<k$. It means that, before priority k, all objectives are satisfied in both $g^{(1)}$ and $g^{(2)}$, but on priority k, $g^{(1)}$ is preferred to $g^{(2)}$, so it is a better solution for the whole multi-objective optimization, no matter what will be on the objectives of lower priorities than g_k. Thus the formulation of the lexicographic minimization problem can be written as follow (Meadowcroft *et al.*, 1992):

$$\begin{aligned} &\min g_k, k=1,2,\cdots n \\ &\text{s. t.}\quad g_i=\min g_i, i<k \end{aligned} \tag{1}$$

Therefore, lexicographic optimization would be defined as the computing process of a lexicographic minimum solution of a multi-objective problem (or sometimes maybe a maximum solution). This solution usually is not the optimal solution of any quadratic objective function and vice versa. As mentioned in Section 1, in lexicographic optimization, one priority can have only one objective, so it also called completely stratified optimization. If needed, the readers can find more about the definition of lexicographic optimization from other references.

2.2 Partially stratified optimization

Still suppose the complex goal $g=\{g_1,g_2,\cdots,g_n\}$ contains n objectives, and all objectives need to be minimized. If these n objectives can be divided into m priorities ($m\le n$), the complex goal can be rewritten as $G=\{G_1,G_2,\cdots,G_m\}$, where $G_i=\sum_j \lambda_{ij}g_{ij}$ is a combined goal of a certain priority i that contains j goals, and the goals in the same priority still could be combined with weight coefficients.

Because the relation between priorities is still lexicographic, the subscript of G_i also describes the relative importance, where G_1 is the most important and G_{i-1} is always more important than G_i. The solution $G^{(1)}=\{G_1^{(1)},G_2^{(1)},\cdots,G_m^{(1)}\}$ is better than the solution $G^{(2)}=\{G_1^{(2)},G_2^{(2)},\cdots,G_m^{(2)}\}$, if and only if $G_k^{(1)}<G_k^{(2)}$ and $G_i^{(1)}=G_i^{(2)}=\min G_i$ hold for certain $k\le m$ and all $i<k$. Similar to de definition of lexicographic minimization in (1), the partially stratified optimization now can be defined as the computing process of a partially stratified minimum solution:

$$\begin{aligned}&\min G_k, k=1,2,\cdots m\\ &\text{s. t.}\quad G_i=\min G_i, i<k\end{aligned}\tag{2}$$

Simply speaking, partially stratified multi-objective optimization here means lexicographic method between priorities and traditional weight coefficients method on goals in the same priority. Specially, if the number of priorities equals to the number of goals ($m=n$), partially stratified multi-objective optimization will equal to lexicographic multi-objective optimization.

3. GA for lexicographic optimization and partially stratified optimization

3.1 GA for lexicographic optimization

In GA, the survival opportunity and competitiveness of individuals are only determined by fitness function. So the key to a lexicographic genetic algorithm is a special fitness function, which is suitable for lexicographic optimization for multi-objective control.

Still suppose a complex goal $g=\{g_1,g_2,\cdots,g_n\}$ contains n objectives, and the fitness function of each objective is $F_i\in[0,1],1\le i\le n$, while $F_i=1$ means objective i has been completely satisfied. Since lexicographic optimization can only deal with a certain objective when all the objectives with higher priority have been achieved already, a series of dynamic coefficients is introduced to describe this decision procedure:

$$\delta_i=\begin{cases}1 & i=1\\ 1 & F_1=F_2=\cdots F_{i-1}=1, 2\le i\le n\\ 0 & \text{otherwise}\end{cases}\tag{3}$$

Here, since F_1 is the fitness function of the most important objective, which has no objective with higher priority than itself, so $\delta_1=1$ should be held all the time. Then the combined fitness function could be:

$$F=\delta_1F_1+\delta_2F_2+\cdots+\delta_nF_n=\sum_{i=1}^{n}\delta_iF_i\tag{4}$$

Using this lexicographic combined fitness function in multi-objective GA, the lexicographic optimum solution can be obtained directly, and there are no special rules on coding method, crossover operator, mutation operator or any other parameters of GA. Constraints on the value of individuals can be matched by lethal penalty or other kinds of penalties in GA, and to ensure the solution's convergence to the optimal solution, the best individual should be remained in every evolution. For the convenience to readers, we will describe the steps of this modified GA for lexicographic optimization briefly as follow:

Step 1: create M initial parent individuals randomly.

Step 2: create M offspring individuals by crossover operator, mutation operator with proper operation on constraints.

Step 3: compute the fitness of all the 2M individuals (parents and offspring) respectively by (4).

Step 4: choose M individuals with higher fitness among the 2M individuals as new parent individuals.

Step 5: if the ending condition for evolution computation is matched, output the individual of the highest fitness, or return to Step 2.

3.2 GA for partially stratified optimization and some discussion

The only difference between GA for lexicographic optimization (LMGA) and partially stratified GA (PSMGA) is the definition of F_i, in LMGA it is the fitness function for a single objective, but in PSMGA it is the fitness function for all the objectives in a same priority. Since the form of fitness fuction is depend on the problem will be solved, the fitness fuction of NMPC based on proposed GA will be introduced in detail later in the Section 4.

Both LMGA and PSMGA are quite different from many other multi-objective GA. Traditional multi-objective GA usually need to find out Pareto Surface (Coello, 2000), which contains a set of Pareto optimal solutions, then choose a best solution by the given criterion. But LMGA and PSMGA don't need this additional selection after evolutionary computing, since the optimal solution of the multi-objective optimization can be obtained directly. For controller design, what we need is just an optimal solution, no matter what the Pareto Surface is, so PSMGA and LMGA's disposal is quite suitable and time-saving.

In LMGA and PSMGA, if the priority order of objectives changes, we only need to modify the logical descriptions of the priorities in the combined fitness function, and if the value of objectives changes, we only need to modify the numerical description of the combined fitness function, while there is no parameters need to be tuned.

4. Multi-objective NMPC based on GA: a case study

4.1 The model of the two-tank system

To be used in this chapter to carry out simulations, the nonlinear model of a two-tank system in Fig. 2. would be introduced here as (5), which is obtained by mechanism modelling, and the sample time of this discrete system is 1 second. Here outputs $y_1(k), y_2(k)$ denote the height of water in two tanks T_1 and T_2 respectively, and control input $u(k)$ is the water fed into tank T_1 from the valve V_3. The manual valve V_1 and V_2 are kept open at the maximal position all the time, and magnetic valve V_3 is controlled by PC to be the actuator of the system, to control the fluid speed of water from pulp P_1. (5-1)

and (5-2) is the fluid mechanical character of T_1 and T_2 and (5-3) is the constraints on outputs, input, and the increment of input respectively. For convenience, all the variables in the model are normalized to the scale 0%-100%.

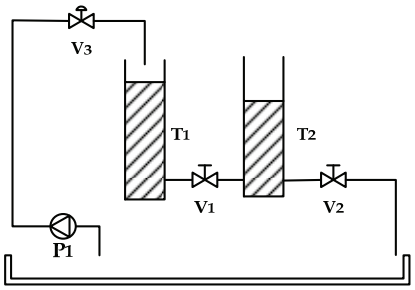


Fig. 2. Structure of the two-tank system

$$y_1(k+1) = y_1(k) + 0.01573 \cdot u(k) - 0.2232 \cdot \text{sign}(y_1(k) - y_2(k)) \cdot \sqrt{|y_1(k) - y_2(k)|} \quad (5\text{-}1)$$

$$y_2(k+1) = y_2(k) - 0.1191\sqrt{y_2(k)} + 0.2232 \cdot \text{sign}(y_1(k) - y_2(k))\sqrt{|y_1(k) - y_2(k)|} \quad (5\text{-}2)$$

$$\text{s. t.} \quad y_1(k), y_2(k) \in [0\%, 100\%]$$

$$u(k) \in [20\%, 80\%]$$

$$u(k) - u(k-1) = \Delta u(k) \in [-5\%, 5\%] \quad (5\text{-}3)$$

where the sign function is

$$\text{sign}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}.$$

4.2 The basic control problem of the two-tank system

The NMPC of the two-tank system would have two forms of objective functions, according to two forms of practical goals in control problem: setpoint and restricted range.

For goals in the form of restricted range $g: y_i(k) \in [y_{i-low}, y_{i-high}], i = 1,2$, suppose the predictive horizon contains p sample time, k is the current time and the predictive value at time k of future output is denoted by $\hat{y}_i(\cdot \mid k)$, the objective function can be chosen as:

$$J(k) = \sum_{j=1}^{p} [\text{pos}(\hat{y}_i(k+j \mid k) - y_{i-high}) + \text{neg}(\hat{y}_i(k+j \mid k) - y_{i-low})]^2, i = 1,2 \quad (6)$$

where the positive function and negative function are

$$\text{pos}(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}, \quad \text{neg}(x) = \begin{cases} 0 & x \geq 0 \\ -x & x < 0 \end{cases}.$$

In (6), if the output is in the given restricted range, the value of objective function J(k) is zero, which means this objective is completely satisfied.

For goals in the form of setpoint $g: y_i(k) = y_{i-set}, i = 1,2$, since the output cannot reach the setpoint from recent value immediately, we can use the concept of reference trajectories, and

the output will reach the set point along it. Suppose the future reference trajectories of output $y_i(k)$ are $w_i(k), i = 1,2$, in most MPC (NMPC), these trajectories often can be set as exponential curves as (7) and Fig. 3. (Zheng *et al.*, 2008)

$$w_i(k+j) = \alpha_i \cdot w_i(k+j-1) + (1-\alpha_i) \cdot y_{i-set}, 1 \le j \le p, i = 1,2 \tag{7}$$

where

$$w_i(k) = y_i(k) \text{ and } 0 \le \alpha_i < 1 .$$

Then the objective function of a setpoint goal would be:

$$J(k) = \sum_{j=1}^{p} (\hat{y}_i(k+j \mid k) - w_i(k+j))^2, i = 1,2 \tag{8}$$

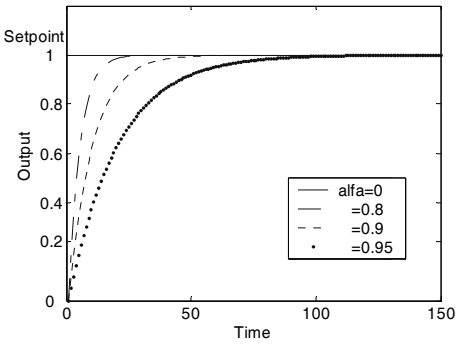


Fig. 3. Description of exponential reference trajectory

4.3 The stair-like control strategy

To enhance the control quality and lighten the computational load of dynamic optimization of NMPC, especially the computational load of GA in this chapter, stair-like control strategy (Wu *et al.*, 2000) could be used here. Suppose the first unknown increment of instant control input is $\Delta u(k) = u(k) - u(k-1)$, and the stair constant β is a positive real number, in stair-like control strategy, the future control inputs could be decided as follow (Wu *et al.*, 2000, Zheng *et al.*, 2008):

$$\Delta u(k+j) = \beta \cdot \Delta u(k+j-1) = \beta^j \cdot \Delta u(k), 1 \le j \le p-1 \tag{9}$$

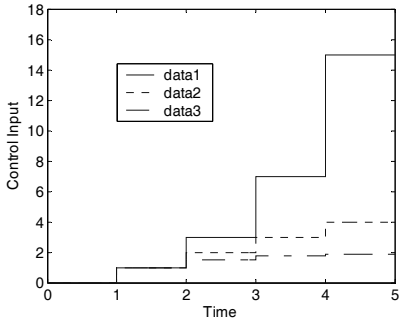


Fig. 4. Description of stair-like control strategy

With this disposal, the elements in the future sequence of control input $u(k)\ \ u(k+1)\ \cdots\ u(k+p-1)$ are not independent as before, and the only unknown variable here in NMPC is the increment of instant control input $\Delta u(k)$, which can determine all the later control input. The dimension of unknown variable in NMPC now decreases from $i \cdot p$ to i remarkably, where i is only the dimension of control input, thus the computational load is no longer depend on the length of the predictive horizon like many other MPC (NMPC). So, it is very convenient to use long predictive horizon to obtain better control quality without additional computational load under this strategy. Because MPC (NMPC) will repeat the dynamic optimization at every sample time, and only $u(k) = \Delta u(k) + u(k-1)$ will be carried out actually in MPC (NMPC), this strategy is surely efficient here. At last, in stair-like control strategy, it also supposes the future increment of control input will change in the same direction, which can prevent the frequent oscillation of control input's increment, while this kind of oscillation is very harmful to the actuators of practical control plants. A visible description of this control strategy is shown in Fig. 4.

4.4 Multi-objective NMPC based on GA

Based on the proposed LMGA and PSMGA, the NMPC now can be established directly. Because NMPC is an online dynamic optimal algorithm, the following steps of NMPC will be executed repeatedly at every sample time to calculate the instant control input.

Step 1: the LMGA (PSMGA) initialize individuals as different $\Delta u(k)$ (with population M) under the constraints in (5-3) with historic data $u(k-1)$.

Step 2: create M offspring individuals by evolutionary operations as mentioned in the end of Section 3.1. In control problem, we usually can use real number coding, linear crossover, stochastic mutation and the lethal penalty in GA for NMPC. Suppose P_1, P_2 are parents and O_1, O_2 are offspring, linear crossover operator $0 < \gamma < 1$ and stochastic mutation operator ε is Gaussian white noise with zero mean, the operations can be described briefly as bellow:

$$\begin{aligned} O_1 &= \gamma \cdot P_1 + (1-\gamma) \cdot P_2 + \varepsilon_1 \\ O_2 &= \gamma \cdot P_2 + (1-\gamma) \cdot P_1 + \varepsilon_2 \end{aligned}, \quad 0 < \gamma < 1 \tag{10}$$

Step 3: predictions of future outputs ($\hat{y}_i(k+1|k)\ \ \hat{y}_i(k+2|k)\ \cdots\ \hat{y}_i(k+p|k)$, i=1,2) are carried out by (5-1) and (5-2) on all the 2M individuals (M parents and M offspring), and their fitness will be calculated. In this control problem, the fitness function F of each objective is transformed from its objective function J easily as follow, to meet the value demand of $F \in [0,1]$, in which J is described by (6) or (8):

$$F = 1/(J+1) \tag{11}$$

To obtain the robustness to model mismatch, feedback compensation can be used in prediction, thus the latest predictive errors $e_i(k) = y_i(k) - \hat{y}_i(k|k-1), i = 1,2$ should be added into every predictive output $\hat{y}_i(k+j|k), 1 = 1,2, 1 \le j \le p$.

Step 4: the M individuals with higher fitness in the 2M individuals will be remained as new parents.

Step 5: if the condition of ending evaluation is met, the best individual will be the increment of instant control input Δu(k) of NMPC, which is taken into practice by the actuator. Else, the process should go back to Step 2, to resume dynamic optimization of NMPC based on LMGA (PSMGA).

4.5 Simulations and analysis of lexicographic multi-objective NMPC

First, the simulation about lexicographic Multi-objective NMPC will be carried out. Choose control objectives as: $g_1 : y_1(k) \in [40\%,60\%]$, $g_2 : y_2(k) \in [20\%,40\%]$, $g_3 : y_2 = 30\%$. Consider the physical character of the system, two different order of priorities can be chosen as: [A]: $g_1 > g_2 > g_3$, [B]: $g_2 > g_1 > g_3$, and they will have the same initial state as $y_1(0) = 80\%$, $y_2(0) = 0\%$ and $u(0) = 20\%$. Parameters of NMPC are $\alpha = 0.95, \beta = 0.85$ for both y_1 and y_2, and parameters of GA are $\gamma = 0.9$, while ε is a zero mean Gaussian white noise, whose variance is 5. Since the feasible control input set is relatively small in our problem according to constraints (5-3), it is enough to have only 10 individuals in our simulation, and they will evolve for 20 generations. While in process control practice, because the sample time is often has a time scale of minutes, even hours, we can have much more individuals and they can evolve much more generations to get a satisfactory solution. (In following figures, dash-dot lines denote g_1, g_2, dot line denote g_3 and solid lines denote y_1, y_2, u)

Compare Fig. 5. and Fig. 6. with Fig. 7. and Fig. 8., although the steady states are the same in these figures, the dynamic responses of them are with much difference, and the objectives are satisfied as the order appointed before respectively under all the constraints. The reason of these results is the special initial state: $y_1(0)$ is higher than g_1 (the most important objective in order [A]: $g_1 > g_2 > g_3$), while $y_2(0)$ is lower than g_2 (the most important objective in order [B]: $g_2 > g_1 > g_3$). So the most important objective of the two orders must be satisfied with different control input at first respectively. Thus the difference can be seen from the different decision-making of the choice in control input more obviously: in Fig. 5. and Fig. 6. the input stays at the lower limit of the constraints at first to meet g_1, while in Fig. 7. and Fig. 8. the input increase as fast as it can to satisfy g_2 at first. The lexicographic character of LMGA is verified by these comparisons.

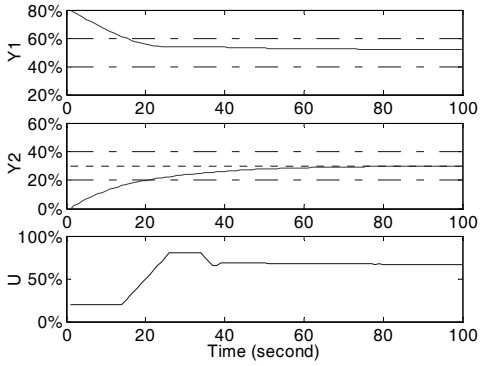


Fig. 5. Control simulation: priority order [A] and p=1

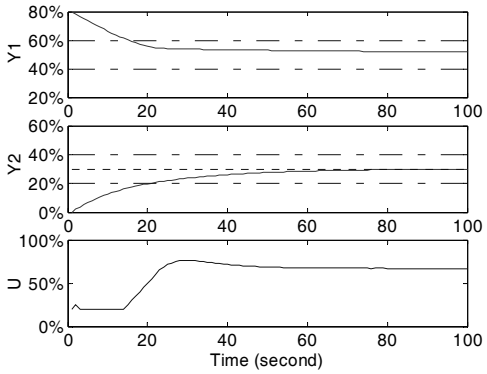


Fig. 6. Control simulation: priority order [A] and p=20

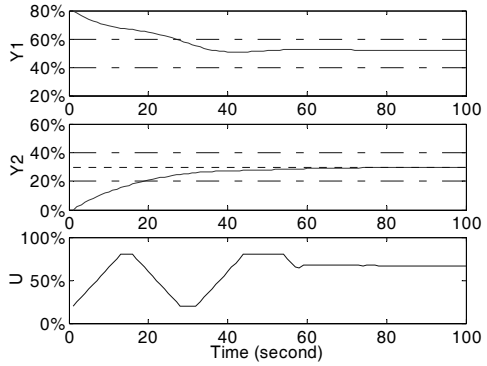


Fig. 7. Control simulation: priority order [B] and p=1

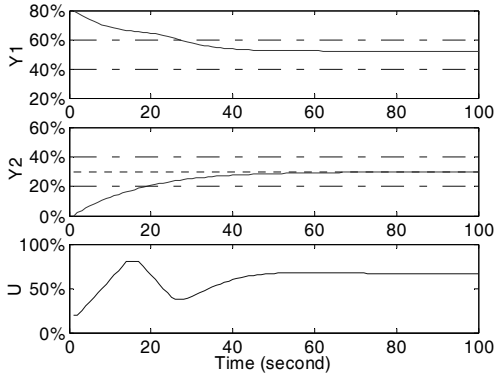


Fig. 8. Control simulation: priority order [B] and p=20

And the difference in control input with different predictive horizon can also be observed from above figures: the control input is much smoother when the predictive horizon

becomes longer, while the output is similar with the control result of shorter predictive horizon. It is the common character of NMPC.

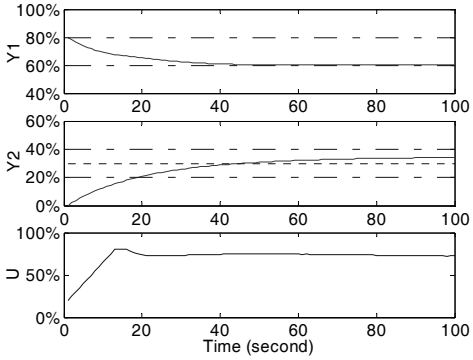


Fig. 9. Control simulation: when an objective cannot be satisfied

In Fig. 9., g_1 is changed as $y_1 \in [60\%,80\%]$, while other objectives and parameters are kept the same as those of Fig. 6., so that g_3 can't be satisfied at steady state. The result shows that y_1 will stay at lower limit of g_1 to reach set-point of g_3 as close as possible, when g_1 must be satisfied first in order [A]. This result also shows the lexicographic character of LMGA obviously.

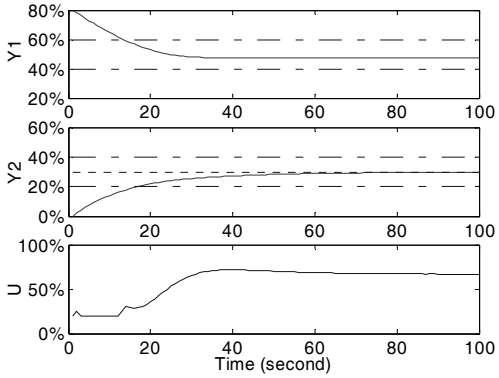


Fig. 10. Control simulation: when model mismatch

Finally, we would consider about of the model mismatch, here the simulative plant is changed, by increasing the flux coefficient 0.2232 to 0.25 in (5-1) and (5-2), while all the objectives, parameters and predictive model are kept the same as those of Fig. 6. The result in Fig. 10. shows the robustness to model mismatch of the controller with error compensation in prediction, as mentioned in Section 4.4.

4.6 Simulations and analysis of partially stratified multi-objective NMPC

To obtain evident comparison to Section 4.5, simulations are carried out with the same parameters ($\alpha = 0.95, \beta = 0.85$ for both y_1 and y_2, predictive horizon p=20 and the same GA parameters), and the only difference is an additional objective on y_1 in the form of a setpoint.

The four control objectives now are $g_1 : y_1(k) \in [40\%, 60\%]$, $g_2 : y_2(k) \in [20\%, 40\%]$, $g_3 : y_2 = 30\%$, $g_4 : y_1 = 50\%$, and then choose the new different order of priorities as: [A]: $g_1 > g_2 > g_3 > g_4$, [B]: $g_2 > g_1 > g_3 > g_4$, if we still use lexicographic multi-objective NMPC as Section 4.5, the control result in Fig. 11. and Fig. 12. is completely the same as Fig. 6. and Fig. 8., when there are only three objectives g_1, g_2, g_3. That means, the additional objective g_4 (setpoint of y_1) could not be considered by the controller in both situations above, because the solution of g_3 (setpoint of y_2) is already a single-point set of u. (In following figures, dash-dot lines denote g_1, g_2, dot line denote g_3, g_4 and solid lines denote y_1, y_2, u)

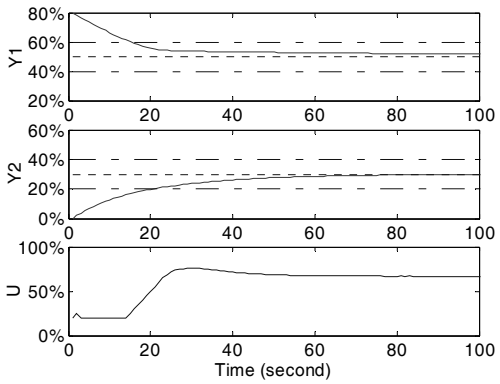


Fig. 11. Control simulation: priority order [A] of four objectives, NMPC based on LMGA

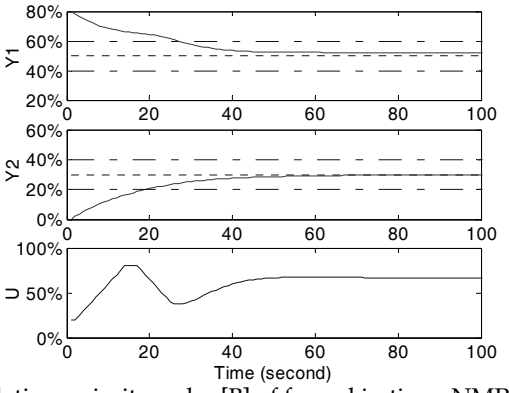


Fig. 12. Control simulation: priority order [B] of four objectives, NMPC based on LMGA

In another word, in lexicographic multi-objective NMPC based on LMGA, if optimization of an objective uses out all the degree of freedom on control inputs (often an objective in the form of setpoint), or an objective cannot be completely satisfied (often an objective in the form of extremum, such as minimization of cost that can not be zero), the objectives with lower priorities will not be take into account at all. But this is not rational in most practice cases, for complex process industrial manufacturing, there are often many objectives in the form of setpoint in a multi-objective control problem, if we handle them with the lexicographic method, usually, we can only satisfy only one of them. Take the proposed two-tank system as example, g_3 and g_4 are both in the form of setpoint, seeing about the steady-state control result in Fig. 13. and Fig. 14., if we want to satisfy $g_3 : y_2 = 30\%$, then y_1 will stay at 51.99%, else if we want to satisfy $g_4 : y_1 = 50\%$, then y_2 will stay at 28.92%, the errors of the dissatisfied output are both more than 1%.

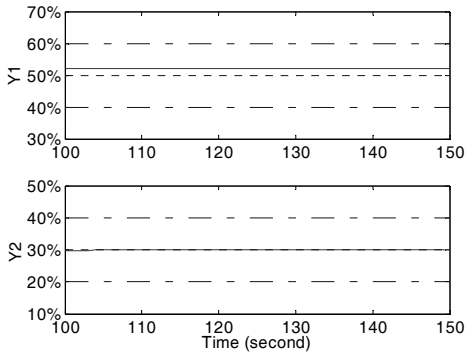


Fig. 13. Steady-state control result when g_3 is completely satisfied

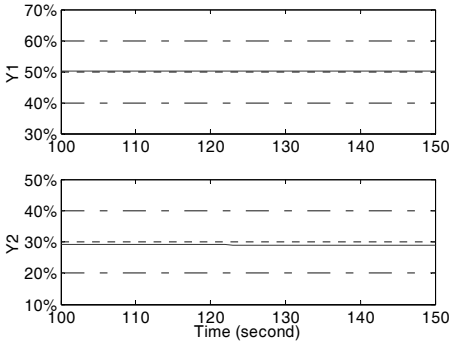


Fig. 14. Steady-state control result when g_4 is completely satisfied

In the above analysis, the mentioned disadvantage comes from the absolute, rigid management of lexicographic method, if we don't develop it, NMPC based on LMGA can only be used in very few control practical problem. Actually, in industrial practice, objectives in the form of setpoint or extremum are often with lower importance, they are usually objectives for higher demand on product quality, manufacturing cost and so on,

which is much less important than the objectives about safety and other basic manufacturing demand. Especially, for objectives in the form of setpoint, under many kinds of disturbances, it always can not be accurately satisfied while it is also not necessary to satisfy them accurately.

A traditional way to improve it is to add slack variables into objectives in the form of setpoint or extremum. Setpoint may be changed into a narrow range around it, and instead of an extremum, the satisfaction of a certain threshold value will be required. For example, in the two-tank system's control problem, setpoint $g_3 : y_2 = 30\%$ could be redefined as $g_3 : y_2 \in 30\% \pm 1\%$.

Another way is modified LMGA into PSMGA as mentioned in Section 3, because sometimes there is no need to divide these objectives with into different priorities respectively, and they are indeed parallel. Take order [A] for example, we now can reform the multi-objective control problem of the two-tank system as: $G_1 > G_2 > G_3 = g_1 > g_2 > \lambda_3 g_3 + \lambda_4 g_4$. Choose weight coefficients as $\lambda_3 = 30, \lambda_4 = 1$ and other parameters the same as those of Fig. 6., while NMPC base on PSMGA has the similar dynamic state control result to that of NMPC based on LMGA, the steady state control result is evidently developed as in Fig. 15. and Fig. 16., y_1 stays at 50.70% and y_2 stays at 29.27%, both of them are in the 0.8% neighborhood of setpoint in g_3, g_4.

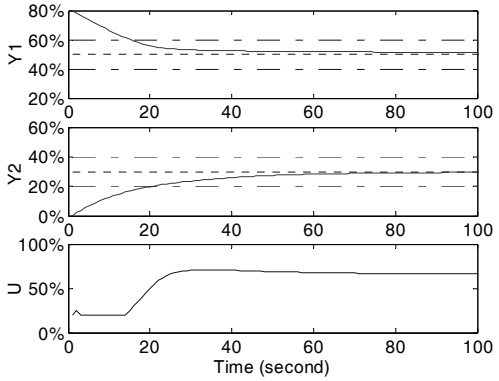


Fig. 15. NMPC based on PSMGA: priority order [A]

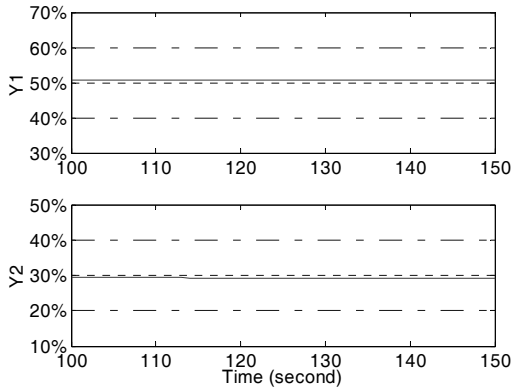


Fig. 16. Steady-state control result of NMPC base on PSMGA

4.7 Some discussions

In the above simulative examples, there is only one control input, but for many practical systems, coordinated control of multi-input is also a serious problem. The brief discussions on multi-input proposed NMPC can be achieved if we still use priorities for inputs. If all the inputs have the same priority, in another word, it is no obvious difference among them in economic cost or other factors, we can just increase the dimension of GA's individual. But, in many cases, the inputs actually also have different priorities: for certain output, different input often has different gain on it with different economic cost. The cheap ones with large gain are always preferred by manufacturers. In this case, we can form anther priority list, and then inputs will be used to solve the control problem one by one, using single input NMPC as the example in Section 4, that can divide an MIMO control problem into some SIMO control problems.

We should point out that, the two kinds of stratified structures proposed in this paper are basic structures for multi-objective controllers, though we use NMPC to realize them in this chapter, they are independent with control algorithms indeed. For certain multi-objective control problem, other proper controllers and computational method can be used.

Another point must be mentioned is that, NMPC proposed in this paper is based on LMGA and PSMGA, because it is hard for most NMPC to get an online analytic solution. But the LMGA and PSMGA are also suitable for other control algorithms, the only task is to modify the fitness function, by introducing the information from the control algorithm which will be used.

At last, all the above simulations could been done in 40-200ms by PC (with 2.7 GHz CPU, 2.0G Memory and programmed by Matlab 6.5), which is much less than the sample time of the system (1 second), that means controllers proposed in this chapter are actually applicable online.

5. Conclusion

In this chapter, to avoid the disadvantages of weight coefficients in multi-objective dynamic optimization, lexicographic (completely stratified) and partially stratified frameworks of multi-objective controller are proposed. The lexicographic framework is absolutely priority-driven and the partially stratified framework is a modification of it, they both can solve the multi-objective control problem with the concept of priority for objective's relative importance, while the latter one is more flexible, without the rigidity of lexicographic method.

Then, nonlinear model predictive controllers based on these frameworks are realized based on the modified genetic algorithm, in which a series of dynamic coefficients is introduced to construct the combined fitness function. With stair–like control strategy, the online computational load is reduced and the performance is developed. The simulative study of a two-tank system indicates the efficiency of the proposed controllers and some deeper discussions are given briefly at last.

The work of this chapter is supported by Fund for Excellent Post Doctoral Fellows (K. C. Wong Education Foundation, Hong Kong, China and Chinese Academy of Sciences), Science and Technological Fund of Anhui Province for Outstanding Youth (08040106910), and the authors also thank for the constructive advices from Dr. De-Feng HE, College of Information Engineering, Zhejiang University of Technology, China.

6. References

Alessio A. & Bemporad A. (2009). A survey on explicit model predictive control. Lecture Notes in Control and Information Sciences (Nonlinear Model Predictive Control: Towards New Challenging Applications), Vol. 384, pp 345-369, ISSN 0170-8643.

Cannon M. (2004). Efficient nonlinear model predictive control algorithms. Annual Reviews in Control. Vol.28, No.2, pp229-237, ISSN 1367-5788.

Coello C. A. C. (2000). An Updated Survey of GA-Based Multiobjective Optimization Techniques, ACM Computing Surveys, Vol.32, No.2, pp109-143, ISSN 0360-0300.

Meadowcroft T. A.; Stephanopoulos G. & Brosilow C. (1992). The modular multivariable controller: I: steady-state properties. AIChE Journal, Vol.38, No.8, pp1254-1278, ISSN 0001-1541.

Ocampo-Martinez C.; Ingimundarson A.; Vicenç P. & J. Quevedo. (2008). Objective prioritization using lexicographic minimizers for MPC of sewer networks. IEEE Transactions on Control Systems Technology, Vol. 16, No.1, pp113-121, ISSN 1063-6536.

Wu G.; Lu X. D.; Ying A. G.; Xue M. S.; Zhang Z. G. & Sun D. M. (2000). Modular Multivariable Self-tuning Regulator. Acta Automatica Sinica, Vol.26, No.6, pp811-815, ISSN 0254-4156.

Yuzgec U.; Becerikli Y. & Turker M. (2006). Nonlinear Predictive Control of a Drying Process Using Genetic Algorithms, ISA Transactions, Vol.45, No.4, pp589-602, ISSN 0019-0578.

Zheng T.; Wu G.; He D. F.; Yue D. Z. (2008). An Efficient Model Nonlinear Predictive Control Algorithm Based on Stair-like Control Strategy, Proceedings of the 27th Chinese Control Conference, Vol.3, pp557-561, ISBN 9787811243901, Kunming, China, July, 2008, Beihang University Press, Beijing, China.

8

Model Predictive Trajectory Control for High-Speed Rack Feeders

Harald Aschemann and Dominik Schindele
Chair of Mechatronics, University of Rostock
18059 Rostock, Germany

1. Introduction

Rack feeders represent the commonly used handling systems for the automated operation of high-bay rackings. To further increase the handling capacity by shorter transport times, control measures are necessary for the reduction of excited structural oscillations, see also Aschemann & Ritzke (2009). One possible approach is given by flatness-based feedforward control, where the desired control inputs are determined by dynamic system inversion using the desired trajectories for the flat outputs as in Bachmayer et al. (2008) and M. Bachmayer & Ulbrich (2008). However, both publications consider only a constant mass position in vertical direction on an elastic beam without any feedback control. A variational approach is presented in Kostin & Saurin (2006) to compute an optimal feedforward control for an elastic beam. Unfortunately, feedforward control alone is not sufficient to guarantee small tracking errors when model uncertainty is present or disturbances act on the system. For this reason in this contribution a model predictive control (MPC) design is presented for fast trajectory control. In general, at model predictive control the optimal input vector is mostly calculated by minimising a quadratic cost function as, e.g., in Wang & Boyd (2010) or Magni & Scattolini (2004). In contrast, the here considered MPC approach aims at reducing future state errors, see Jung & Wen (2004), and allows for a relatively small computational effort as required in a real-time implementation. Hence, the proposed MPC algorithm is well suited for systems with fast dynamics, e.g., a high-speed linear axis with pneumatic muscles as presented in Schindele & Aschemann (2008) or high-speed rack feeders as in the given case. A further attractive characteristic of this MPC approach is its applicability to linear as well as nonlinear systems.
For the experimental investigation of modern control approaches to active oscillation damping as well as tracking control, a test rig of a high-speed rack feeder has been build up at the Chair of Mechatronics at the University of Rostock, see Figure 1. The experimental set-up consists of a carriage driven by an electric DC servo motor via a toothed belt, on which an elastic beam as the vertical supporting structure is mounted. On this beam structure, a cage with variable load mass is guided relocatably in vertical direction. This cage with the coordinate $y_K(t)$ in horizontal direction and $x_K(t)$ in vertical direction represents the tool center point (TCP) of the rack feeder that should track desired trajectories as accurate as possible. The movable cage is driven by a tooth belt and an electric DC servo motor as well. The angles of the actuators are measured by internal angular transducers, respectively. Additionally, the horizontal position of the carriage is detected by a magnetostrictive transducer. Both axes are operated with a fast underlying velocity control on the current converter. Consequently, the

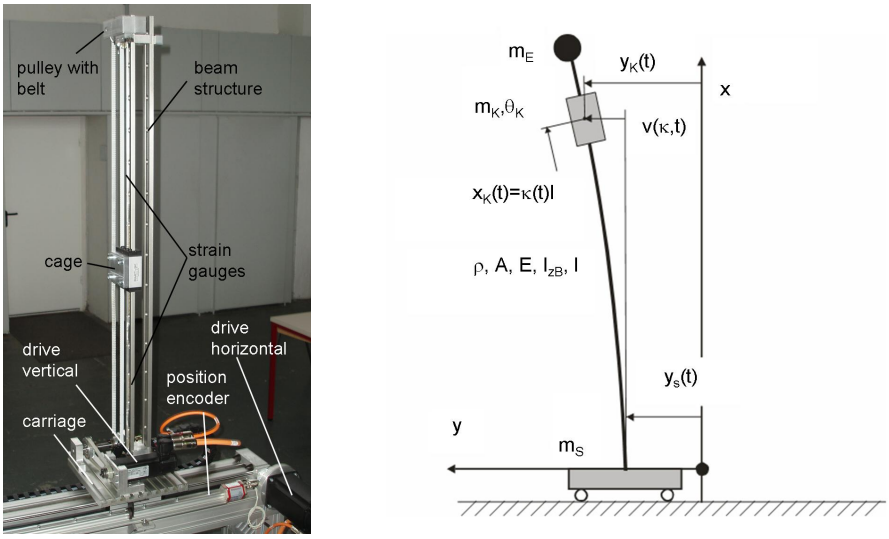


Fig. 1. Experimental set-up of the high-speed rack feeder (left) and the corresponding elastic multibody model (right).

corresponding velocities deal as new control input, and the implementational effort is tremendously reduced as compared to the commonly used force or torque input, like in Staudecker et al. (2008), where passivity techniques were employed for feedback control of a similar set-up. Two strain gauges are used to determine the bending deformation of the elastic beam.
Basis of the control design for the rack feeder is a planar elastic multibody system, where for the mathematical description of the bending deflection of the elastic beam a Ritz ansatz is introduced, covering for instance the first bending mode. The decentralised feedforward and feedback control design for both axes is performed employing a linearised state space representation, respectively. Given couplings between both axes are taken into account by the gain-scheduling technique with the normalised vertical cage position as scheduling parameter, see also Aschemann & Ritzke (2010). This leads to an adaptation of the whole control structure for the horizontal axis. The capability of the proposed control concept is shown by experimental results from the test set-up with regard to tracking behaviour and damping of bending oscillations. Especially the artificial damping introduced by the closed control loop represents a main improvement. The maximum velocity of the TCP during the tracking experiments is approx. 2.5 m/s.

2. Control-oriented modelling of the mechatronic system

Elastic multibody models have proven advantageously for the control-oriented modelling of flexible mechanical systems. For the feedforward and feedback control design of the rack feeder a multibody model with three rigid bodies - the carriage (mass m_S), the cage movable on the beam structure (mass m_K, mass moment of inertia θ_K), and the end mass at the tip of the beam (mass m_E) - and an elastic Bernoulli beam (density ρ, cross sectional area A, Youngs modulus E, second moment of area I_{zB}, and length ℓ) is chosen. The varying vertical position

$x_K(t)$ of the cage on the beam is denoted by the dimensionless system parameter

$$\kappa\left(t\right)=\frac{x_\kappa\left(t\right)}{l}. \tag{1}$$

The elastic degrees of freedom of the beam concerning the bending deflection can be described by the following Ritz ansatz

$$v\left(x,t\right)=\bar{v}_1\left(x\right)v_1\left(t\right)=\left[\frac{3}{2}\left(\frac{x}{l}\right)^2-\frac{1}{2}\left(\frac{x}{l}\right)^3\right]v_1\left(t\right), \tag{2}$$

which takes into account only the first bending mode. The vector of generalised coordinates results in

$$\boldsymbol{q}\left(t\right)=\left[\begin{array}{c} y_S\left(t\right) \\ v_1\left(t\right) \end{array}\right]. \tag{3}$$

The nonlinear equations of motion can be derived either by Lagrange's equations or, advantageously, by the Newton-Euler approach, cf. Shabana (2005). After a linearisation for small bending deflections, the equations of motion can be stated in M-D-K form

$$\boldsymbol{M}\ddot{\boldsymbol{q}}\left(t\right)+\boldsymbol{D}\dot{\boldsymbol{q}}\left(t\right)+\boldsymbol{K}\boldsymbol{q}\left(t\right)=\boldsymbol{h}\cdot\left[F_{SM}\left(t\right)-F_{SR}\left(\dot{y}_S\left(t\right)\right)\right]. \tag{4}$$

The symmetric mass matrix is given by

$$\boldsymbol{M} \;=\; \left[\begin{array}{cc} m_S+\rho Al+m_K+m_E & \frac{3}{8}\rho Al+\frac{m_K\kappa^2}{2}\left[3-\kappa\right]+m_E \\ \frac{3}{8}\rho Al+\frac{m_K\kappa^2}{2}\left[3-\kappa\right]+m_E & m_{22} \end{array}\right], \tag{5}$$

with $m_{22}=\frac{33}{140}\rho Al+\frac{6\rho I_{zB}}{5l}+\frac{m_K\kappa^2}{4}\left[3-\kappa\right]^2+\frac{9\theta_K\kappa^2}{l^2}\left[1-\kappa+\frac{\kappa^2}{4}\right]+m_E$. The damping matrix, which is specified with stiffness-proportional damping properties, and the stiffness matrix become

$$\boldsymbol{D} \;=\; \left[\begin{array}{cc} 0 & 0 \\ 0 & \frac{3k_dEI_{zB}}{l^3} \end{array}\right], \tag{6}$$

$$\boldsymbol{K} \;=\; \left[\begin{array}{cc} 0 & 0 \\ 0 & \frac{3EI_{zB}}{l^3}-\frac{3}{8}\rho Ag-\frac{3m_Kg\kappa^3}{l}\left[1+\frac{3\kappa^2}{20}-\frac{3\kappa}{4}\right]-\frac{6m_Eg}{5l} \end{array}\right]. \tag{7}$$

The input vector of the generalised forces, which accounts for the control input as well as the disturbance input, reads

$$\boldsymbol{h} \;=\; \left[\begin{array}{cc} 1 & 0 \end{array}\right]^T. \tag{8}$$

The electric drive for the carriage is operated with a fast underlying velocity control on the current converter. The resulting dynamic behaviour is characterised by a first-order lag system with a time constant T_{1y}

$$T_{1y}\ddot{y}_S\left(t\right)+\dot{y}_S\left(t\right)=v_S\left(t\right). \tag{9}$$

This differential equation replaces now the equation of motion for the carriage in the mechanical system model, which leads to a modified mass matrix as well as a modified damping matrix

$$\boldsymbol{M}_y \;=\; \left[\begin{array}{cc} T_{1y} & 0 \\ \frac{3}{8}\rho Al+\frac{m_K\kappa^2}{2}\left[3-\kappa\right]+m_E & m_{22} \end{array}\right], \tag{10}$$

$$\boldsymbol{D}_y = \begin{bmatrix} 1 & 0 \\ 0 & \frac{3k_d E I_{zB}}{l^3} \end{bmatrix} . \tag{11}$$

The stiffness matrix $\boldsymbol{K} = \boldsymbol{K}_y$ and the input vector for the generalised forces $\boldsymbol{h} = \boldsymbol{h}_y$, however, remain unchanged. Hence, the equations of motion are given by

$$\ddot{\boldsymbol{q}} = -\boldsymbol{M}_y^{-1}\boldsymbol{K}_y\boldsymbol{q} - \boldsymbol{M}_y^{-1}\boldsymbol{D}_y\dot{\boldsymbol{q}} + \boldsymbol{M}_y^{-1}\boldsymbol{h}_y v_S . \tag{12}$$

For control design, the system representation is reformulated in state space form

$$\dot{\boldsymbol{x}}_y = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \ddot{\boldsymbol{q}} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{M}_y^{-1}\boldsymbol{K}_y & -\boldsymbol{M}_y^{-1}\boldsymbol{D}_y \end{bmatrix}}_{\boldsymbol{A}_y} \underbrace{\begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \end{bmatrix}}_{\boldsymbol{x}_y} + \underbrace{\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{M}_y^{-1}\boldsymbol{h}_y \end{bmatrix}}_{\boldsymbol{b}_y} \underbrace{v_S}_{u_y} . \tag{13}$$

The design model for the vertical movement of the cage can be directly stated in state space representation. Here, an underlying velocity control is employed on the current converter, which is also described by a first-order lag system

$$T_{1x}\ddot{x}_K(t) + \dot{x}_K(t) = v_K(t) . \tag{14}$$

The corresponding state space description follows immediately in the form

$$\dot{\boldsymbol{x}}_x = \begin{bmatrix} \dot{x}_K \\ \ddot{x}_K \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{T_{1x}} \end{bmatrix}}_{\boldsymbol{A}_x} \underbrace{\begin{bmatrix} x_K \\ \dot{x}_K \end{bmatrix}}_{\boldsymbol{x}_x} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{T_{1x}} \end{bmatrix}}_{\boldsymbol{b}_x} \underbrace{v_K}_{u_x} . \tag{15}$$

Whereas the state space respresentation for the horizontal y-axis depends on the varying system parameter $\kappa(t)$, the description of the x-axis is invariant. A gain-scheduling, hence, is necessary only for the horizontal axis in y-direction.

3. Decentralised control design

As for control, a decentralised approach is followed, at which the coupling of the vertical cage motion with the horizontal axis is taken into account by gain-scheduling techniques. For the control of the cage position $x_K(t)$ a simple proportional feedback in combination with feedforward control, which is based on the inverse transfer function of this axis, is sufficient

$$v_K(t) = K_R\left(x_{Kd}(t) - x_K(t)\right) + \dot{x}_{Kd}(t) + T_{1x}\ddot{x}_{Kd}(t) . \tag{16}$$

For this purpose, the desired trajectory $x_{Kd}(t)$ and its first two time derivatives are available from trajectory planning. The design of the state feedback for the horizontal motion is carried out by the MPC approach, which is explained in the following chapter.

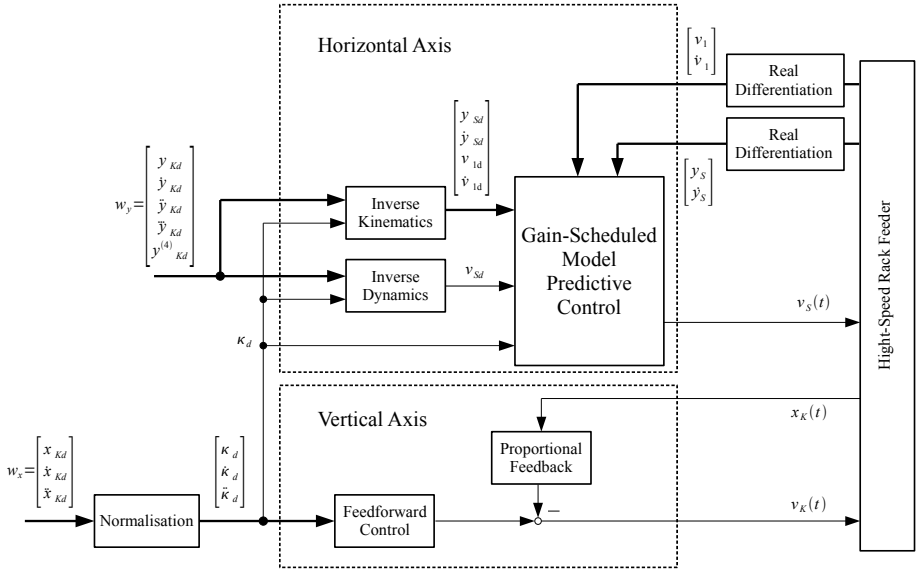


Fig. 2. Implementation of the control structure.

4. Model Predictive Control

The main idea of the control approach consists in a minimisation of a future tracking error in terms of the predicted state vector based on the actual state and the desired state vector resulting from trajectory planning, see Lizarralde et al. (1999), Jung & Wen (2004). The minimisation is achieved by repeated approximate numerical optimisation in each time step, in the given case using the Newton-Raphson technique. The optimisation is initialised in each time step with the optimisation result of the preceding time step in form of the input vector. The MPC-algorithm is based on the following discrete-time state space representation

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{b}u_k \,, \tag{17}$$

$$y_k = \boldsymbol{c}^T\boldsymbol{x}_k \,, \tag{18}$$

with the state vector $\boldsymbol{x}_k \in \mathbb{R}^n$, the control input $u_k \in \mathbb{R}$ and the output vector $y_k \in \mathbb{R}$. The constant M specifies the prediction horizon T_P as a multiple of the sampling time t_s, i.e. $T_P = M \cdot t_s$. The predicted input vector at time k becomes

$$\boldsymbol{u}_{k,M} = \left[u_1^{(k)}, ..., u_M^{(k)}\right]^T \,, \tag{19}$$

with $\boldsymbol{u}_{k,M} \in \mathbb{R}^M$. The predicted state vector at the end of the prediction horizon $\boldsymbol{\phi}_M(\boldsymbol{x}_k, \boldsymbol{u}_{k,M})$ is obtained by repeated substitution of k by $k+1$ in the discrete-time state equation (17)

$$\begin{aligned} \boldsymbol{x}_{k+2} &= \boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{b}u_{k+1} = \boldsymbol{A}^2\boldsymbol{x}_k + \boldsymbol{A}\boldsymbol{b}u_k + \boldsymbol{b}u_{k+1} \\ &\vdots \\ \boldsymbol{x}_{k+M} &= \boldsymbol{A}^M\boldsymbol{x}_k + \boldsymbol{A}^{M-1}\boldsymbol{b}u_k + \boldsymbol{A}^{M-2}\boldsymbol{b}u_{k-1} + \ldots + \boldsymbol{b}u_{k+M-1} \\ &= \boldsymbol{\phi}_M(\boldsymbol{x}_k, \boldsymbol{u}_{k,M}) \,. \end{aligned} \tag{20}$$

The difference of $\boldsymbol{\phi}_M(\boldsymbol{x}_k, \boldsymbol{u}_{k,M})$ and the desired state vector $\boldsymbol{x}_d$ yields the final control error

$$\boldsymbol{e}_{M,k} = \boldsymbol{\phi}_M(\boldsymbol{x}_k, \boldsymbol{u}_{k,M}) - \boldsymbol{x}_d \,, \tag{21}$$

i.e. to the control error at the end of the prediction horizon. The cost function to be minimised follows as

$$J_{MPC} = \frac{1}{2} \cdot \boldsymbol{e}_{M,k}^T \boldsymbol{e}_{M,k} \,, \tag{22}$$

and, hence, the necessary condition for an extremum can be stated as

$$\frac{\partial J_{MPC}}{\partial \boldsymbol{e}_{M,k}} = \boldsymbol{e}_{M,k} \stackrel{!}{=} 0 \,. \tag{23}$$

A Taylor-series expansion of (23) at $\boldsymbol{u}_{k,M}$ in the neighbourhood of the optimal solution leads to the following system of equations

$$\boldsymbol{0} = \boldsymbol{e}_{M,k} + \frac{\partial \boldsymbol{\phi}_M}{\partial \boldsymbol{u}_{k,M}} \Delta \boldsymbol{u}_{k,M} + T.h.O.. \tag{24}$$

The vector $\Delta \boldsymbol{u}_{k,M}$ denotes the difference which has to be added to the input vector $\boldsymbol{u}_{k,M}$ to obtain the optimal solution. The n equations (24) represent an under-determined set of equations with $m \cdot M$ unknowns having an infinite number of solutions. An unique solution for $\Delta \boldsymbol{u}_{k,M}$ can be determined by solving the following L_2-optimisation problem with (24) as side condition

$$J = \frac{1}{2} \cdot \Delta \boldsymbol{u}_{k,M}^T \Delta \boldsymbol{u}_{k,M} + \boldsymbol{\lambda}^T \left(\boldsymbol{e}_{M,k} + \frac{\partial \boldsymbol{\phi}_M}{\partial \boldsymbol{u}_{k,M}} \Delta \boldsymbol{u}_{k,M} \right) \,. \tag{25}$$

Consequently, the necessary conditions can be stated as

$$\begin{aligned} \frac{\partial J}{\partial \Delta \boldsymbol{u}_{k,M}} &\stackrel{!}{=} 0 = \Delta \boldsymbol{u}_{k,M} + \left(\frac{\partial \boldsymbol{\phi}_M}{\partial \boldsymbol{u}_{k,M}} \right)^T \boldsymbol{\lambda}, \\ \frac{\partial J}{\partial \boldsymbol{\lambda}} &\stackrel{!}{=} 0 = \boldsymbol{e}_{M,k} + \frac{\partial \boldsymbol{\phi}_M}{\partial \boldsymbol{u}_{k,M}} \Delta \boldsymbol{u}_{k,M}, \end{aligned} \tag{26}$$

which results in $\boldsymbol{e}_{M,k}$

$$\boldsymbol{e}_{M,k} = \underbrace{\frac{\partial \boldsymbol{\phi}_M}{\partial \boldsymbol{u}_{k,M}} \left(\frac{\partial \boldsymbol{\phi}_M}{\boldsymbol{u}_{k,M}} \right)^T}_{\boldsymbol{S}(\boldsymbol{\phi}_M, \boldsymbol{u}_{k,M})} \boldsymbol{\lambda} \,. \tag{27}$$

If the matrix $\boldsymbol{S}\left(\boldsymbol{\phi}_M, \boldsymbol{u}_{k,M}\right)$ is invertible, the vector $\boldsymbol{\lambda}$ can be calculated as follows

$$\boldsymbol{\lambda} = \boldsymbol{S}^{-1}\left(\boldsymbol{\phi}_M, \boldsymbol{u}_{k,M}\right) \boldsymbol{e}_{M,k} \,. \tag{28}$$

An almost singular matrix $\boldsymbol{S}\left(\boldsymbol{\phi}_M, \boldsymbol{u}_{k,M}\right)$ can be treated by a modification of (28)

$$\boldsymbol{\lambda} = \left[\mu \boldsymbol{I} + \boldsymbol{S}\left(\boldsymbol{\phi}_M, \boldsymbol{u}_{k,M}\right)\right]^{-1} \boldsymbol{e}_{M,k} \,, \tag{29}$$

where $\boldsymbol{I}$ denotes the unity matrix. The regularisation parameter $\mu > 0$ in (29) may be chosen constant or may be calculated by a sophisticated algorithm. The latter solution improves the convergence of the optimisation but increases, however, the computational complexity. Solving (26) for $\Delta \boldsymbol{u}_{k,M}$ and inserting $\boldsymbol{\lambda}$ according to (28) or (29), directly yields the L_2-optimal solution

$$\Delta \boldsymbol{u}_{k,M} = -\left(\frac{\partial \boldsymbol{\phi}_M}{\partial \boldsymbol{u}_{k,M}}\right)^T \boldsymbol{S}^{-1}\left(\boldsymbol{\phi}_M, \boldsymbol{u}_{k,M}\right) \boldsymbol{e}_{M,k} = -\left(\frac{\partial \boldsymbol{\phi}_M}{\partial \boldsymbol{u}_{k,M}}\right)^{\dagger} \boldsymbol{e}_{M,k} \,. \tag{30}$$

Here, $\left(\frac{\partial \boldsymbol{\phi}_M}{\partial \boldsymbol{u}_{k,M}}\right)^{\dagger}$ denotes the Moore-Penrose pseudo inverse of $\frac{\partial \boldsymbol{\phi}_M}{\partial \boldsymbol{u}_{k,M}}$. The overall MPC-algorithm can be described as follows:
Choice of the initial input vector $\boldsymbol{u}_{0,M}$ at time $k = 0$, e.g. $\boldsymbol{u}_{0,M} = \boldsymbol{0}$, and repetition of steps a) - c) at each sampling time $k \geq 0$:

a) Calculation of an improved input vector $\boldsymbol{v}_{k,M}$ according to

$$\boldsymbol{v}_{k,M} = \boldsymbol{u}_{k,M} - \eta_k \left(\frac{\partial \boldsymbol{\phi}_M}{\partial \boldsymbol{u}_{k,M}}\right)^{\dagger} \boldsymbol{e}_{M,k} \,. \tag{31}$$

The step width η_k can be determined with, e.g., the Armijo-rule.

b) For the calculation of $\boldsymbol{u}_{k+1,M}$ the elements of the vector $\boldsymbol{v}_{k,M}$ have to be shifted by one element and the steady-state input u_d corresponding to the final state has to be inserted at the end

$$\boldsymbol{u}_{k+1,M} = \begin{bmatrix} \boldsymbol{0}_{((M-1)\times 1)} \\ 1 \end{bmatrix} u_d + \begin{bmatrix} \boldsymbol{0}_{((M-1)\times 1)} & \boldsymbol{I}_{(M-1)} \\ 0 & \boldsymbol{0}_{(1\times(M-1))} \end{bmatrix} \boldsymbol{v}_{k,M}. \tag{32}$$

In general, the steady-state control input u_d can be computed from

$$\boldsymbol{x}_d = \boldsymbol{A}\boldsymbol{x}_d + \boldsymbol{b}u_d. \tag{33}$$

Alternatively, the desired input vector u_d can be calculated by an inverse system model. If the system is differentially flat, see Fliess et al. (1995) the desired input u_d can be calculated exactly by the flat system output and a finite number of its time derivative. For non-flat outputs -as in the given case- the approach presented in chapter 4.4 is useful.

c) The first element of the improved input vector $\boldsymbol{v}_{k,M}$ is applied as control input at time k

$$u_k = \begin{bmatrix} 1 & \boldsymbol{0}_{(1\times(M-1))} \end{bmatrix} \boldsymbol{v}_{k,M} \,. \tag{34}$$

In the proposed algorithm only one iteration is performed per time step. A similar approach using several iteration steps is described in Weidemann et al. (2004). An improvement of the trajectory tracking behaviour can be achieved if an input vector resulting from an inverse system model is used as initial vector for the subsequent optimisation step instead of the last input vector. The slightly modified algorithm can be stated as follows

a) Calculation of the ideal input vector $\boldsymbol{u}_{k,M}^{(d)}$ by evaluating an inverse system model with the specified reference trajectory as well as a certain number $\beta \in \mathbb{N}$ of its time derivatives

$$\boldsymbol{u}_{k,M}^{(d)} = \boldsymbol{u}_{k,M}^{(d)} \left(\boldsymbol{y}_d, \dot{\boldsymbol{y}}_d, ..., \overset{(\beta)}{\boldsymbol{y}}_d \right) . \tag{35}$$

b) Calculation of the improved input vector $\boldsymbol{v}_{k,M}$ based on the equation

$$\boldsymbol{v}_{k,M} = \boldsymbol{u}_{k,M}^{(d)} - \eta_k \left(\frac{\partial \boldsymbol{\phi}_M}{\partial \boldsymbol{u}_{k,M}} \right)^{\dagger} \boldsymbol{e}_{M,k} . \tag{36}$$

c) Application of the first element of $\boldsymbol{v}_{k,M}$ to the process

$$u_k = \begin{bmatrix} 1 & \mathbf{0}_{(1\times(M-1))} \end{bmatrix} \boldsymbol{v}_{k,M} . \tag{37}$$

If the reference trajectory is known in advance, the according reference input vector $\boldsymbol{u}_{k,M}^{(d)}$ can be computed offline. Consequently, the online computational time remains unaffected.

4.1 Numerical calculations

The analytical computation of the Jacobian $\frac{\partial \boldsymbol{\phi}_M}{\partial \boldsymbol{u}_{k,M}}$ becomes increasingly complex for larger values of M. Therefore, a numerical approach is preferred taking advantage of the chain rule with $i = 0, ..., M-1$

$$\frac{\partial \boldsymbol{\phi}_M}{\partial \boldsymbol{u}_{i+1}^{(k)}} = \frac{\partial \boldsymbol{\phi}_M}{\partial \boldsymbol{x}_{k+M-1}} \cdot \frac{\partial \boldsymbol{x}_{k+M-1}}{\partial \boldsymbol{x}_{k+M-2}} \cdot \ldots \cdot \frac{\partial \boldsymbol{x}_{k+i+2}}{\partial \boldsymbol{x}_{k+i+1}} \cdot \frac{\partial \boldsymbol{x}_{k+i+1}}{\partial u_{i+1}^{(k)}} . \tag{38}$$

In this way, the Jacobian can be computed as follows

$$\frac{\partial \boldsymbol{\phi}_M}{\partial \boldsymbol{u}_{k,M}} = [\boldsymbol{A}^{M-1}\boldsymbol{b}, \boldsymbol{A}^{M-2}\boldsymbol{b}, ..., \boldsymbol{A}\boldsymbol{b}, \boldsymbol{b}] . \tag{39}$$

For the inversion of the symmetric and positive definite matrix $\boldsymbol{S}\left(\boldsymbol{\phi}_M, \boldsymbol{u}_{k,M}\right) = \frac{\partial \boldsymbol{\phi}_M}{\partial \boldsymbol{u}_{k,M}} \left(\frac{\partial \boldsymbol{\phi}_M}{\partial \boldsymbol{u}_{k,M}} \right)^T$ the Cholesky-decomposition has proved advantageous in terms of computational effort.

4.2 Choice of the MPC design parameters

The most important MPC design parameter is the prediction horizon T_P, which is given as the product of the sampling time t_s and the constant value M. Large values of T_P lead to a slow and smooth transient behaviour and result in a robust and stable control loop. For fast trajectory tracking, however, a smaller value T_P is desirable concerning a small tracking error. The choice of the sampling time t_s is crucial as well: a small sampling time is necessary regarding both discretisation error and stability; however, the MPC-algorithm has to be evaluated in real-time within the sampling inverval. Furthermore, the smaller t_s, the larger becomes M for a given prediction horizon, which in turn increases the computational complexity of the optimisation step. Consequently, a system-specific trade-off has to be made for the choice of M and t_s. This paper follows the moving horizon approach with a constant prediction horizon and, hence, a constant dimension $m \cdot M$ of the corresponding optimisation problem in constrast to the shrinking horizon approach according to Weidemann et al. (2004).

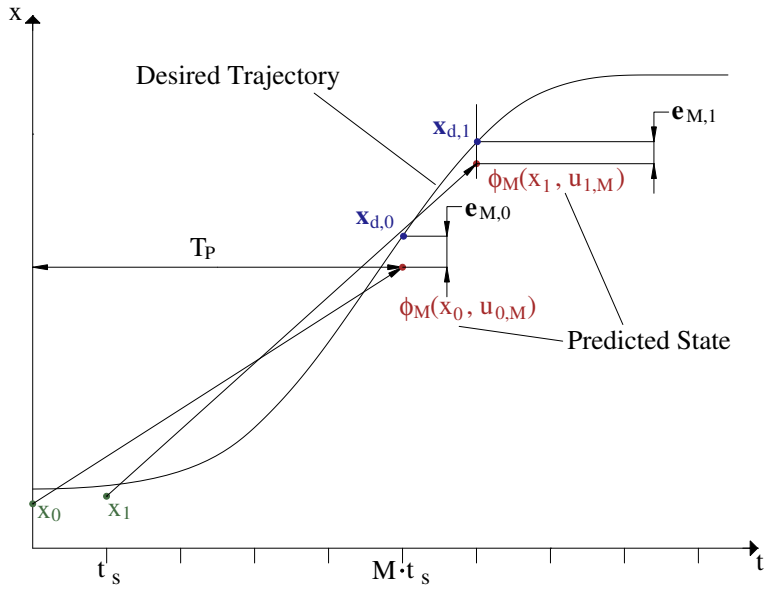


Fig. 3. Design parameters.

4.3 Input constraints

One major advantage of predictive control is the possibility to easily account for input constraints, which are present in almost all control applications. To this end, the cost function can be extended with a corresponding term

$$h(u_j^{(k)}) = \begin{cases} 0 & u_{min} \leq u_j^{(k)} \leq u_{max} \\ g_1(u_j^{(k)}) \text{ for } & u_j^{(k)} > u_{max} \\ g_2(u_j^{(k)}) & u_j^{(k)} < u_{min} \end{cases}, \tag{40}$$

which has to be evaluated componentwise, i.e. for each input variable at each sampling time. Thus, the contribution of the additional input constraints depending on $\boldsymbol{u}_{k,M}$ is given by

$$z(\boldsymbol{u}_{k,M}) = \sum_{j=1}^{M} h(u_j^{(k)}). \tag{41}$$

Instead of $\boldsymbol{e}_{M,k}$, the extended vector $\left[\boldsymbol{e}_{M,k}^T, z\right]^T$ has to be minimised in the MPC-algorithm.

4.4 MPC of the horizontal cage position

The state space representation for the cage position control in y-direction design is given by (13). The discrete-time representation of the continous-time system is obtained by Euler discretisation

$$\boldsymbol{x}_{y,k+1} = \left(\boldsymbol{I} + t_s \cdot \boldsymbol{A}_y(\kappa)\right) \boldsymbol{x}_{y,k} + t_s \cdot \boldsymbol{b}_y(\kappa)\, u_{y,k}. \tag{42}$$

Using this simple discretisation method, the computational effort for the MPC-algorithm can be kept acceptable. By the way, no significant improvement could be obtained for the given system with the Heun discretisation method because of the small sampling time $t_s = 3\ ms$. Only in the case of large sampling times, e.g. $t_s > 20\ ms$, the increased computational effort caused by a sophisticated time discretisation method is advantageous. Then, the smaller discretisation error allows for less time integration steps for a specified prediction horizon, i.e. a smaller number M. As a result, the smaller number of time steps can overcompensate the larger effort necessary for a single time step.
The ideal input $u_d(t)$ can be obtained in continous time as function of the output variable

$$y_K(t) = \boldsymbol{c}_y^T \boldsymbol{x}_y(t) = \begin{bmatrix} 1 & \frac{1}{2}\kappa^2\,(3-\kappa) & 0 & 0 \end{bmatrix} \boldsymbol{x}_y(t)\,, \tag{43}$$

and a certain number of its time derivatives. For this purpose the corresponding transfer function of the system under consideration is employed

$$\frac{Y_K\,(s)}{U_d\,(s)} = \boldsymbol{c}_y^T\,(s\boldsymbol{I} - \boldsymbol{A}_y)^{-1}\,\boldsymbol{b}_y = \frac{\left(b_0 + b_1 \cdot s + b_2 \cdot s^2\right)}{N\,(s)}\,. \tag{44}$$

Obviously, the numerator of the control transfer function contains a second degree polynomial in s, leading to two transfer zeros. This shows that the considered output $y_K(t)$ represents a non-flat output variable that makes computing of the feedforward term more difficult. A possible way for calculating the desired input variable is given by a modification of the numerator of the control transfer function by introducing a polynomial ansatz for the feedforward action according to

$$U_d\,(s) = \left[k_{V0} + k_{V1} \cdot s + \ldots + k_{V4} \cdot s^4\right] Y_{Kd}\,(s)\,. \tag{45}$$

For its realisation the desired trajectory $y_{Kd}(t)$ as well as the first four time derivatives are available from a trajectory planning module. The feedforward gains can be computed from a comparison of the corresponding coefficients in the numerator as well as the denominator polynomials of

$$\begin{aligned} \frac{Y_K\,(s)}{Y_{Kd}\,(s)} &= \frac{\left(b_0 + \ldots + b_2 \cdot s^2\right)\left[k_{V0} + \ldots + k_{V4} \cdot s^4\right]}{N\,(s)} \\ &= \frac{b_{V0}\left(k_{Vj}\right) + b_{V1}\left(k_{Vj}\right) \cdot s + \ldots + b_{V6}\left(k_{Vj}\right) \cdot s^6}{a_0 + a_1 \cdot s + \ldots + s^4} \end{aligned} \tag{46}$$

according to

$$a_i = b_{Vi}\left(k_{Vj}\right), i = 0, \ldots, n = 4\,. \tag{47}$$

This leads to parameter-dependent feedforward gains $k_{Vj} = k_{Vj}(\kappa)$. It is obvious that due the higher numerator degree in the modified control transfer function a remaining dynamics must be accepted. Lastly, the desired input variable in the time domain is represented by

$$u_d(t) = u_d\left(\dot{y}_{Kd}(t), \ddot{y}_{Kd}(t), \dddot{y}_{Kd}(t), y_{Kd}^{(4)}(t), \kappa\right)\,. \tag{48}$$

To obtain the desired system states as function of the output trajectory the output equation

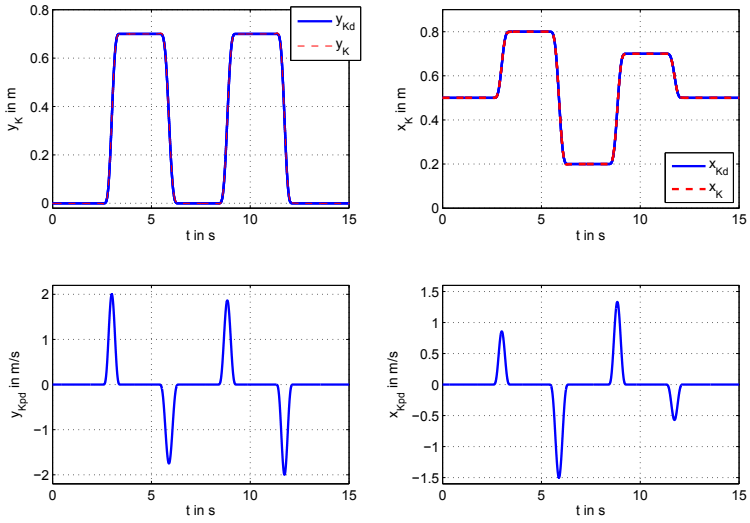


Fig. 4. Desired trajectories for the cage motion: desired and actual position in horizontal direction (upper left corner), desired and actual position in vertical direction (upper right corner), actual velocity in horizontal direction (lower left corner) and actual velocity in vertical direction (lower right corner).

and its first three time derivatives are considered. Including the equations of motion (12) yields the following set of equations

$$y_{Kd}(t) = y_S(t) + \tfrac{1}{2}\kappa^2 (3-\kappa) \cdot v_1(t), \tag{49}$$

$$\dot{y}_{Kd}(t) = \dot{y}_S(t) + \tfrac{1}{2}\kappa^2 (3-\kappa) \cdot \dot{v}_1(t), \tag{50}$$

$$\ddot{y}_{Kd}(t) = \ddot{y}_S(t) + \tfrac{1}{2}\kappa^2 (3-\kappa) \cdot \ddot{v}_1(t) = \ddot{y}_K\left(v_1(t), \dot{y}_S(t), \dot{v}_1(t), u_d(t), \kappa\right), \tag{51}$$

$$\dddot{y}_{Kd}(t) = \dddot{y}_K\left(v_1(t), \dot{y}_S(t), \dot{v}_1(t), u_d(t), \dot{u}_d(t), \kappa\right) . \tag{52}$$

Solving equation (49) to (52) for the system states results in the desired state vector

$$\boldsymbol{x}_d(t) = \begin{bmatrix} y_{Sd}\left(y_{Kd}(t), \dot{y}_{Kd}(t), \ddot{y}_{Kd}(t), \dddot{y}_{Kd}(t), u_d(t), \dot{u}_d(t), \kappa\right) \\ v_{1d}\left(\dot{y}_{Kd}(t), \ddot{y}_{Kd}(t), \dddot{y}_{Kd}(t), u_d(t), \dot{u}_d(t), \kappa\right) \\ \dot{y}_{Sd}\left(\dot{y}_{Kd}(t), \ddot{y}_{Kd}(t), \dddot{y}_{Kd}(t), u_d(t), \dot{u}_d(t), \kappa\right) \\ \dot{v}_{1d}\left(\dot{y}_{Kd}(t), \ddot{y}_{Kd}(t), \dddot{y}_{Kd}(t), u_d(t), \dot{u}_d(t), \kappa\right) \end{bmatrix} . \tag{53}$$

This equation still contains the inverse dynamics $u_d(t)$ and its time derivative $\dot{u}_d$. Substituting u_d for equation (48) and $\dot{u}_d(t)$ for the time derivative of (48), which can be calculated analyti-

cally, finally leads to

$$\boldsymbol{x}_d(t) = \begin{bmatrix} y_{Sd}\left(y_{Kd}(t), \dot{y}_{Kd}(t), \ddot{y}_{Kd}(t), \dddot{y}_{Kd}(t), y_{Kd}^{(4)}(t), y_{Kd}^{(5)}(t), \kappa\right) \\ v_{1d}\left(y_{Kd}(t), \dot{y}_{Kd}(t), \ddot{y}_{Kd}(t), \dddot{y}_{Kd}(t), y_{Kd}^{(4)}(t), y_{Kd}^{(5)}(t), \kappa\right) \\ \dot{y}_{Sd}\left(y_{Kd}(t), \dot{y}_{Kd}(t), \ddot{y}_{Kd}(t), \dddot{y}_{Kd}(t), y_{Kd}^{(4)}(t), y_{Kd}^{(5)}(t), \kappa\right) \\ \dot{v}_{1d}\left(y_{Kd}(t), \dot{y}_{Kd}(t), \ddot{y}_{Kd}(t), \dddot{y}_{Kd}(t), y_{Kd}^{(4)}(t), y_{Kd}^{(5)}(t), \kappa\right) \end{bmatrix}. \tag{54}$$

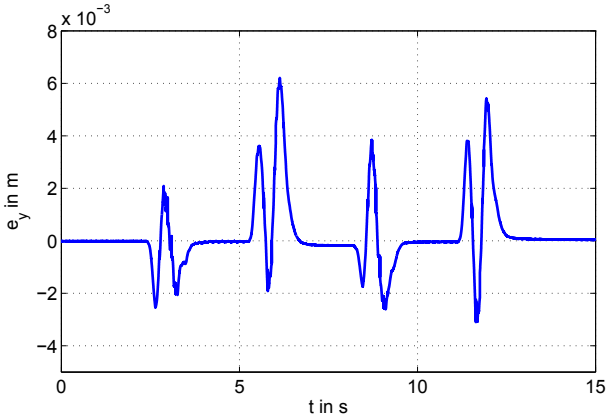


Fig. 5. Tracking error $e_y\,(t)$ for the cage motion in horizontal direction.

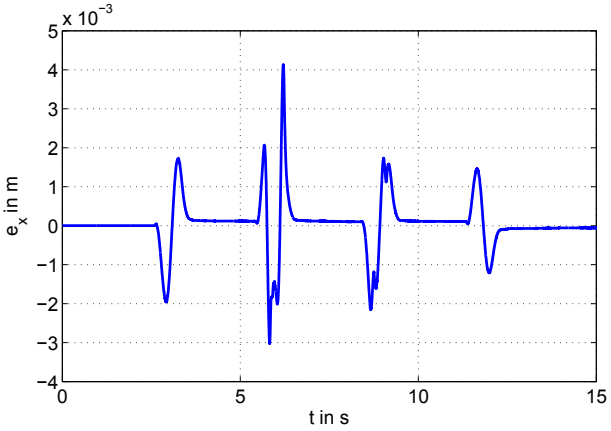


Fig. 6. Tracking error $e_x\,(t)$ for the cage motion in vertical direction.

5. Experimental validation on the test rig

The benefits and the efficiency of the proposed control measures shall be pointed out by experimental results obtained from the test set-up available at the Chair of Mechatronics, University of Rostock. For this purpose, a synchronous four times continuously differentiable desired trajectory is considered for the position of the cage in both x- and y-direction. The desired trajectory is given by polynomial functions that comply with specified kinematic constraints, which is achieved by taking advantage of time scaling techniques. The desired trajectory shown in Figure 4 comprises a sequence of reciprocating motions with maximum velocities of 2 m/s in horizontal direction and 1.5 m/s in vertical direction. The resulting tracking errors

$$e_y(t) = y_{Kd}(t) - y_K(t) \tag{55}$$

and

$$e_x(t) = x_{Kd}(t) - x_K(t) \tag{56}$$

are depicted in Figure 5 and Figure 6. As can be seen, the maximum position error in y-direction during the movements is about 6 mm and the steady-state position error is smaller than 0.2 mm, whereas the maximum position error in x-direction is approx. 4 mm. Figure 7

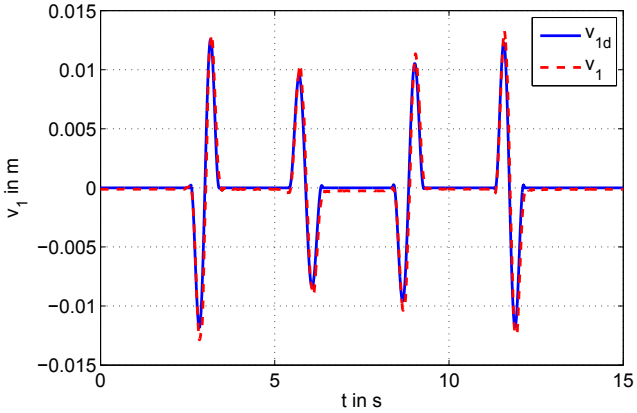


Fig. 7. Comparison of the desired values $v_{1d}(t)$ and the actual values $v_1(t)$ for the bending deflection.

shows the comparison of the bending deflection measured by strain gauges attached to the flexible beam with desired values. During the acceleration as well as the deceleration intervals, physically unavoidable bending deflections could be noticed. The achieved benefit is given by the fact the remaining oscillatons are negligible when the rack feeder arrives at its target position. This underlines both the high model accuracy and the quality of the active damping of the first bending mode. Figure 8 depicts the disturbance rejection properties due to an external excitation by hand. At the beginning, the control structure is deactivated, and the excited bending oscillations decay only due to the very weak material damping. After approx. 2.8 seconds, the control structure is activated and, hence, the first bending mode is actively damped. The remaining oscillations are characterised by higher bending modes that decay with material damping. In future work, the number of Ritz ansatz functions shall be

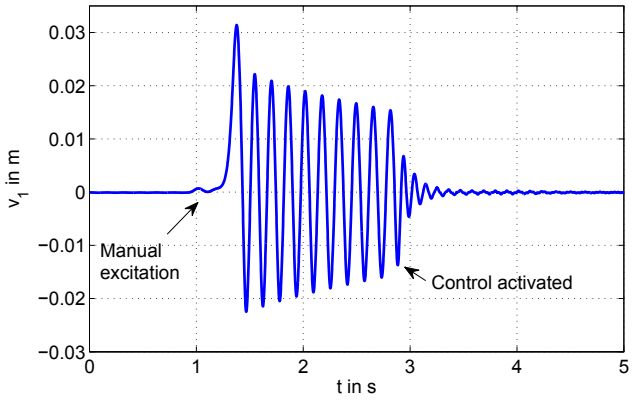


Fig. 8. Transient response after a manual excitation of the bending deflection: at first without feedback control, after approx. 2.8 seconds with active control.

increased to include the higher bending modes as well in the active damping. The corresponding elastic coordinates and their time derivatives can be determined by observer techniques.

6. Conclusions

In this paper, a gain-scheduled fast model predictive control strategy for high-speed rack feeders is presented. The control design is based on a control-oriented elastic multibody system. The suggested control algorithm aims at reducing the future tracking error at the end of the prediction horizon. Beneath an active oscillation damping of the first bending mode, an accurate trajectory tracking for the cage position in x- and y-direction is achieved. Experimental results from a prototypic test set-up point out the benefits of the proposed control structure. Experimental results show maximum tracking errors of approx. 6 mm in transient phases, whereas the steady-state tracking error is approx. 0.2 mm. Future work will address an active oscillation damping of higher bending modes as well as an additional gain-scheduling with respect to the varying payload.

7. References

Aschemann, H. & Ritzke, J. (2009). Adaptive aktive Schwingungsdämpfung und Trajektorienfolgeregelung für hochdynamische Regalbediengeräte (in German), *Schwingungen in Antrieben, Vorträge der 6. VDI-Fachtagung in Leonberg, Germany*. (in German).

Aschemann, H. & Ritzke, J. (2010). Gain-scheduled tracking control for high-speed rack feeders, *Proc. of the first joint international conference on multibody system dynamics (IMSD), 2010, Lappeenranta, Finland* .

Bachmayer, M., Rudolph, J. & Ulbrich, H. (2008). Flatness based feed forward control for a horizontally moving beam with a point mass, *European Conference on Structural Control, St. Petersburg* pp. 74–81.

Fliess, M., Levine, J., Martin, P. & Rouchon, P. (1995). Flatness and defect of nonlinear systems: Introductory theory and examples, *Int. J. Control* **61**: 1327–1361.

Jung, S. & Wen, J. (2004). Nonlinear model predictive control for the swing-up of a rotary inverted pendulum, *ASME J. of Dynamic Systems, Measurement and Control* **126**(3): 666–673.

Kostin, G. V. & Saurin, V. V. (2006). The Optimization of the Motion of an Elastic Rod by the Method of Integro-Differential Relations, *Journal of computer and Systems Sciences International*, Vol. 45, Pleiades Publishing, Inc., pp. 217–225.

Lizarralde, F., Wen, J. & Hsu, L. (1999). A new model predictive control strategy for affine nonlinear control systems, *Proc of the American Control Conference (ACC '99), San Diego* pp. 4263 – 4267.

M. Bachmayer, J. R. & Ulbrich, H. (2008). Acceleration of linearly actuated elastic robots avoiding residual vibrations, *Proceedings of the 9th International Conference on Motion and Vibration Control, Munich, Germany*.

Magni, L. & Scattolini, R. (2004). Model predictive control of continuous-time nonlinear systems with piecewise constant control, *IEEE Transactions on automatic control* **49**(6): 900–906.

Schindele, D. & Aschemann, H. (2008). Nonlinear model predictive control of a high-speed linear axis driven by pneumatic muscles, *Proc. of the American Control Conference (ACC), 2008, Seattle, USA* pp. 3017–3022.

Shabana, A. A. (2005). *Dynamics of multibody systems*, Cambridge University Press, Cambridge.

Staudecker, M., Schlacher, K. & Hansl, R. (2008). Passivity based control and time optimal trajectory planning of a single mast stacker crane, *Proc. of the 17th IFAC World Congress, Seoul, Korea* pp. 875–880.

Wang, Y. & Boyd, S. (2010). Fast model predictive control using online optimization, *IEEE Transactions on control systems technology* **18**(2): 267–278.

Weidemann, D., Scherm, N. & Heimann, B. (2004). Discrete-time control by nonlinear online optimization on multiple shrinking horizons for underactuated manipulators, *Proceedings of the 15th CISM-IFToMM Symposium on Robot Design, Dynamics and Control, Montreal* .

9

Plasma stabilization system design on the base of model predictive control

Evgeny Veremey and Margarita Sotnikova
Saint-Petersburg State University,
Faculty of Applied Mathematics and Control Processes
Russia

1. Introduction

Tokamaks, as future nuclear power plants, currently present exceptionally significant research area. The basic problems are electromagnetic control of the plasma current, shape and position. High-performance plasma control in a modern tokamak is the complex problem (Belyakov et al., 1999). This is mainly connected with the design requirements imposed on magnetic control system and power supply physical constraints. Besides that, plasma is an extremely complicated dynamical object from the modeling point of view and usually control system design is based on simplified linear system, representing plasma dynamics in the vicinity of the operating point (Ovsyannikov et al., 2005). This chapter is focused on the control systems design on the base of Model Predictive Control (MPC) (Camacho & Bordons, 1999; Morari et al., 1994). Such systems provide high-performance control in the case when accurate mathematical model of the plant to be controlled is unknown. In addition, these systems allow to take into account constraints, imposed both on the controlled and manipulated variables (Maciejowski, 2002). Furthermore, MPC algorithms can base on both linear and nonlinear mathematical models of the plant. So MPC control scheme is quite suitable for plasma stabilization problems.

In this chapter two different approaches to the plasma stabilization system design on the base of model predictive control are considered. First of them is based on the traditional MPC scheme. The most significant drawback of this variant is that it does not guarantee stability of the closed-loop control circuit. In order to eliminate this problem, a new control algorithm is proposed. This algorithm allows to stabilize control plant in neighborhood of the plasma equilibrium position. Proposed approach is based on the ideas of MPC and modal parametric optimization. Within the suggested framework linear closed-loop system eigenvalues are placed in the specific desired areas on the complex plane for each sample instant. Such areas are located inside the unit circle and reflect specific requirements and constraints imposed on closed-loop system stability and oscillations.

It is well known that the MPC algorithms are very time-consuming, since they require the repeated on-line solution of the optimization problem at each sampling instant. In order to reduce computational load, algorithms parameters tuning are performed and a special method is proposed in the case of modal parametric optimization based MPC algorithms.

The working capacity and effectiveness of the MPC algorithms is demonstrated by the example of ITER-FEAT plasma vertical stabilization problem. The comparison of the approaches is done.

2. Control Problem Formulation

2.1 Mathematical model of the plasma vertical stabilization process in ITER-FEAT tokamak

The dynamics of plasma control process can be commonly described by the system of ordinary differential equations (Misenov, 2000; Ovsyannikov et al., 2006)

$$\frac{d\Psi}{dt} + RI = V, \tag{1}$$

where Ψ is the poloidal flux vector, R is a diagonal resistance matrix, I is a vector of active and passive currents, V is a vector of voltages applied to coils. The vector Ψ is given by nonlinear relation

$$\Psi = \Psi(I,\, I_p), \tag{2}$$

where I_p is the plasma current. The vector of output variables is given by

$$y = y(I, I_p). \tag{3}$$

Linearizing equations (1)–(3) in the vicinity of the operating point, we obtain a linear model of the process in the state space form. In particular, the linear model describing plasma vertical control in ITER-FEAT tokamak is presented below.

ITER-FEAT tokamak (Gribov et al., 2000) has a separate fast feedback loop for plasma vertical stabilization. The Vertical Stabilization (VS) converter is applied in this loop. Its voltage is evaluated in the feedback controller, which uses the vertical velocity of plasma current centroid as an input. So the linear model can be written as follows

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{b}u, \\ y &= \mathbf{c}\mathbf{x} + du, \end{aligned} \tag{4}$$

where $\mathbf{x} \in \mathbf{E}^{58}$ is a state space vector, $u \in E^1$ is the voltage of the VS converter, $y \in E^1$ is the vertical velocity of the plasma current centroid.

Since the order of this linear model is very high, an order reduction is desirable to simplify the controller synthesis problem. The standard Matlab function ***schmr*** was used to perform model reduction **from 58th to 3rd** order. As a result, we obtain a transfer function of the reduced SISO model (from input u to output y)

$$P(s) = \frac{1.732 \cdot 10^{-6}(s - 121.1)(s + 158.2)(s + 9.641)}{(s + 29.21)(s + 8.348)(s - 12.21)}. \tag{5}$$

This transfer function has poles which dominate the dynamics of the initial plant. The unstable pole corresponds to vertical instability. It is natural to assume that two other poles are determined by the virtual circuit dynamic related to the most significant elements in the tokamak vessel construction. The quality of the model reduction can be illustrated by the comparison of the Bode diagram for both initial and reduced models. Fig. 1 shows the Bode diagrams for initial and reduced 3^{rd} order models on the left and for initial and reduced 2^{nd} order model on the right. It is easy to see that the curves for initial model and reduced 3^{rd} order model are actually indistinguishable, contrary to the 2^{nd} order model.

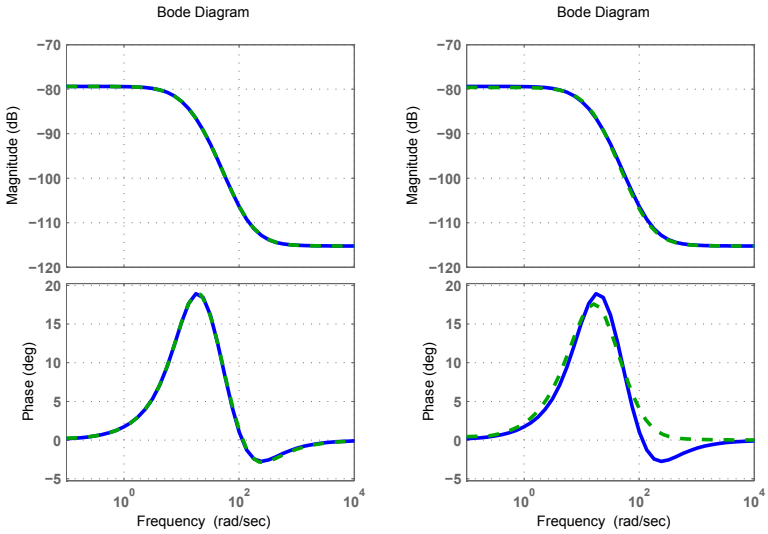


Fig. 1. Bode diagrams for initial (solid lines) and reduced (dotted lines) models.

In addition to plant model (5), we must take into account the following limits that are imposed on the power supply system

$$V_{\max}^{VS} = 0.6kV,\ I_{\max}^{VS} = 20.7kA, \tag{6}$$

where $V_{\max}^{VS}$ is the maximum voltage, $I_{\max}^{VS}$ is the maximum current in the VS converter. So, the linear model (5) together with constraints (6) is considered in the following as the basis for controller synthesis.

2.2 Optimal control problem formulation

The desired controller must stabilize vertical velocity of the plasma current centroid. One of the approaches to control synthesis is based on the optimal control theory. In this framework, plasma vertical stabilization problem can be stated as follows. One needs to find a feedback control algorithm $u = u(t, y)$ that provides a minimum of the quadratic cost functional

$$J = J(u) = \int_0^\infty (y^2(t) + \lambda u^2(t))dt, \tag{7}$$

subject to plant model (5) and constraints (6), and guarantees closed-loop stability. Here λ is a constant multiplier setting the trade-off between controller's performance and control energy costs.

Specifically, in order to find an optimal controller, LQG-synthesis can be performed. Such a controller has high stabilization performance in the unconstrained case. However, it is perhaps not the best choice in the presence of constraints.

Contrary to this, the MPC synthesis allows to take into account constraints. Its basic scheme implies on-line optimization of the cost functional (7) over a finite horizon subject to plant model (5) and imposed constraints (6).

3. Model Predictive Control Algorithms

3.1 MPC Basic Principles

Suppose we have a mathematical model, which approximately describes control process dynamics

$$\dot{\tilde{\mathbf{x}}}(\tau) = \mathbf{f}(\tau, \tilde{\mathbf{x}}(\tau), \tilde{\mathbf{u}}(\tau)), \quad \tilde{\mathbf{x}}|_{\tau=t} = \mathbf{x}(t). \tag{8}$$

Here $\tilde{\mathbf{x}}(\tau) \in \mathbf{E}^n$ is a state vector, $\tilde{\mathbf{u}}(\tau) \in \mathbf{E}^m$ is a control vector, $\tau \in [t, \infty)$, $\mathbf{x}(t)$ is the actual state of the plant at the instant t or its estimation based on measurement output.

This model is used to predict future outputs of the process given the programmed control $\tilde{\mathbf{u}}(\tau)$ over a finite time interval $\tau \in [t, t + T_p]$. Such a model is called **prediction model** and the parameter T_p is named **prediction horizon**. Integrating system (8) we obtain $\tilde{\mathbf{x}}(\tau) = \tilde{\mathbf{x}}(\tau, \mathbf{x}(t), \tilde{\mathbf{u}}(\tau))$—predicted process evolution over time interval $\tau \in [t, t + T_p]$.

The programmed control $\tilde{\mathbf{u}}(\tau)$ is chosen in order to minimize quadratic cost functional over the prediction horizon

$$J = J(\mathbf{x}(t), \tilde{\mathbf{u}}(\cdot), T_p) = \int_t^{t+T_p} ((\tilde{\mathbf{x}} - \mathbf{r}_x)' \mathbf{R}(\tilde{\mathbf{x}})(\tilde{\mathbf{x}} - \mathbf{r}_x) + (\tilde{\mathbf{u}} - \mathbf{r}_u)' \mathbf{Q}(\tilde{\mathbf{x}})(\tilde{\mathbf{u}} - \mathbf{r}_u)) d\tau, \tag{9}$$

where $\mathbf{R}(\tilde{\mathbf{x}})$, $\mathbf{Q}(\tilde{\mathbf{x}})$ are positive definite symmetric weight matrices, $\mathbf{r}_x$, $\mathbf{r}_u$ are state and control input reference signals. In addition, the programmed control $\tilde{\mathbf{u}}(\tau)$ should satisfy all of the constraints imposed on the state and control variables. Therefore, the programmed control $\tilde{\mathbf{u}}(\tau)$ over prediction horizon is chosen to provide minimum of the following optimization problem

$$J(\mathbf{x}(t), \tilde{\mathbf{u}}(\cdot), T_p) \to \min_{\tilde{\mathbf{u}}(\cdot) \in \Omega_u}, \tag{10}$$

where Ω_u is the admissible set given by

$$\Omega_u = \left\{ \tilde{\mathbf{u}}(\cdot) \in \mathbf{K}_n^0[t, t + T_p] : \tilde{\mathbf{u}}(\tau) \in \mathbf{U},\ \tilde{\mathbf{x}}(\tau, \mathbf{x}(t), \tilde{\mathbf{u}}(\tau)) \in \mathbf{X},\ \forall \tau \in [t, t + T_p] \right\}. \tag{11}$$

Here, $\mathbf{K}_n^0[t, t + T_p]$ is the set of piecewise continuous vector functions over the interval $[t, t + T_p]$, $\mathbf{U} \subset \mathbf{E}^m$ is the set of feasible input values, $\mathbf{X} \subset \mathbf{E}^n$ is the set of feasible state values. Denote by $\tilde{\mathbf{u}}^*(\tau)$ the solution of the optimization problem (10), (11). In order to implement feedback loop, the obtained optimal programmed control $\tilde{\mathbf{u}}^*(\tau)$ is used as the input only on the time interval $[t, t + \delta]$, where $\delta << T_p$. So, only a small part of $\tilde{\mathbf{u}}^*(\tau)$ is implemented. At time $t + \delta$ the whole procedure—prediction and optimization—is repeated again to find new optimal programmed control over time interval $[t + \delta,\ t + \delta + T_p]$. Summarizing, the basic MPC scheme works as follows:

1. Obtain the state estimation $\hat{\mathbf{x}}$ on the base of measurements $\mathbf{y}$.
2. Solve the optimization problem (10), (11) subject to prediction model (8) with initial conditions $\tilde{\mathbf{x}}|_{\tau=t} = \hat{\mathbf{x}}(t)$ and cost functional (9).
3. Implement obtained optimal control $\tilde{\mathbf{u}}^*(\tau)$ over time interval $[t, t + \delta]$.
4. Repeat the whole procedure 1–3 at time $t + \delta$.

From the previous discussion, the most significant MPC features can be noted:

- Both linear and nonlinear model of the plant can be used as a prediction model.
- MPC allows taking into account constraints imposed both on the input and output variables.

- MPC is the feedback control with the discrete entering of the measurement information at each sampling instant $0, \delta, 2\delta, \ldots$.
- MPC control algorithms imply the repeated (at each sampling instant with interval δ) on-line solution of the optimization problems. It is especially important from the real-time implementation point of view, because fast calculations are needed.

3.2 MPC real-time implementation

In order for real-time implementation, piece-wise constant functions are used as a programmed control over the prediction horizon. That is, the programmed control $\tilde{\mathbf{u}}(\tau)$ is presented by the sequence$\{\tilde{\mathbf{u}}_k, \tilde{\mathbf{u}}_{k+1}, ..., \tilde{\mathbf{u}}_{k+P-1}\}$, where $\tilde{\mathbf{u}}_i \in \mathbf{E}^m$ is the control input at the time interval $[i\delta, (i+1)\delta]$, δ is the sampling interval. Note that, P is a number of sampling intervals over the prediction horizon, that is $T_p = P\delta$. Likewise, general MPC formulation presented above consider nonlinear prediction model in the discrete form

$$\begin{array}{l} \tilde{\mathbf{x}}_{i+1} = \mathbf{f}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{u}}_i), \quad i = k+j, \quad j = 0, 1, 2, ..., \quad \tilde{\mathbf{x}}_k = \mathbf{x}_k, \\ \tilde{\mathbf{y}}_i = \mathbf{C}\tilde{\mathbf{x}}_i. \end{array} \tag{12}$$

Here $\tilde{\mathbf{y}}_i \in \mathbf{E}^r$ is the vector of output variables, $\mathbf{x}_k \in \mathbf{E}^n$ is the actual state of the plant at time instant k or its estimation on the base of measurement output. We shall say that the sequence of vectors $\{\tilde{\mathbf{y}}_{k+1}, \tilde{\mathbf{y}}_{k+2}, ..., \tilde{\mathbf{y}}_{k+P}\}$ represents the prediction of future plant behavior.
Similar to the cost functional (9), consider also its discrete analog given by

$$\begin{array}{rl} J_k = J_k(\bar{\mathbf{y}}, \bar{\mathbf{u}}) = \sum_{j=1}^{P} \big[& (\tilde{\mathbf{y}}_{k+j} - \mathbf{r}^y_{k+j})^T \mathbf{R}_{k+j} (\tilde{\mathbf{y}}_{k+j} - \mathbf{r}^y_{k+j}) \\ & + (\tilde{\mathbf{u}}_{k+j-1} - \mathbf{r}^u_{k+j-1})^T \mathbf{Q}_{k+j} (\tilde{\mathbf{u}}_{k+j-1} - \mathbf{r}^u_{k+j-1}) \big] , \end{array} \tag{13}$$

where $\mathbf{R}_{k+j}$ and $\mathbf{Q}_{k+j}$ are the weight matrices as in the functional (9), $\mathbf{r}^y_i$ and $\mathbf{r}^u_i$ are the output and input reference signals,

$$\begin{array}{l} \bar{\mathbf{y}} = \begin{pmatrix} \tilde{\mathbf{y}}_{k+1} & \tilde{\mathbf{y}}_{k+2} & ... & \tilde{\mathbf{y}}_{k+P} \end{pmatrix}^T \in \mathbf{E}^{rP}, \\ \bar{\mathbf{u}} = \begin{pmatrix} \tilde{\mathbf{u}}_k & \tilde{\mathbf{u}}_{k+1} & ... & \tilde{\mathbf{u}}_{k+P-1} \end{pmatrix}^T \in \mathbf{E}^{mP} \end{array}$$

are the auxiliary vectors.
The optimization problem (10), (11) can now be stated as follows

$$J_k(\mathbf{x}_k, \tilde{\mathbf{u}}_k, \tilde{\mathbf{u}}_{k+1}, ...\tilde{\mathbf{u}}_{k+P-1}) \to \min_{\{\tilde{\mathbf{u}}_k, \tilde{\mathbf{u}}_{k+1}, ..., \tilde{\mathbf{u}}_{k+P-1}\} \in \Omega \in \mathbf{E}^{mP}}, \tag{14}$$

where $\Omega = \left\{ \bar{\mathbf{u}} \in \mathbf{E}^{mP} : \tilde{\mathbf{u}}_{k+j-1} \in \mathbf{U},\ \tilde{\mathbf{x}}_{k+j} \in \mathbf{X},\ j = 1, 2, ..., P \right\}$ is the admissible set.
Generally, the function $J(\mathbf{x}_k, \tilde{\mathbf{u}}_k, \tilde{\mathbf{u}}_{k+1}, ...\tilde{\mathbf{u}}_{k+P-1})$ is a nonlinear function of mP variables and Ω is a non-convex set. Therefore, the optimization task (14) is a nonlinear programming problem.
Now real-time MPC algorithm can be presented as follows:

1. Obtain the state estimation $\hat{\mathbf{x}}_k$ based on measurements $\mathbf{y}_k$ using the observer.
2. Solve the nonlinear programming problem (14) subject to prediction model (12) with initial conditions $\tilde{\mathbf{x}}_k = \hat{\mathbf{x}}_k$ and cost functional (13). It should be noted, that the value of the function $J_k(\mathbf{x}_k, \tilde{\mathbf{u}}_k, \tilde{\mathbf{u}}_{k+1}, ...\tilde{\mathbf{u}}_{k+P-1})$ is obtained by numerically integrating the prediction model (12) and then substituting the predicted behavior $\bar{\mathbf{x}} \in \mathbf{E}^{nP}$ into the cost function (13) given the programmed control $\{\tilde{\mathbf{u}}_k, \tilde{\mathbf{u}}_{k+1}, ..., \tilde{\mathbf{u}}_{k+P-1}\}$ over the prediction horizon and initial conditions $\hat{\mathbf{x}}_k$.

3. Let $\{\tilde{\mathbf{u}}_k^*, \tilde{\mathbf{u}}_{k+1}^*, ..., \tilde{\mathbf{u}}_{k+P-1}^*\}$ be the solution of the problem (14). Implement only the first component $\tilde{\mathbf{u}}_k^*$ of the obtained optimal sequence over time interval $[k\delta, (k+1)\delta]$.
4. Repeat the whole procedure 1–3 at next time instant $(k+1)\delta$.

Note, that the algorithm stated above implies real-time solution of the nonlinear programming problem at each sampling instant. The complexity of such a problem is determined by the number of sampling intervals P.
The simplest way to reduce the optimization problem order is to decrease the prediction horizon. But, it is necessary to keep in mind that the performance of the closed-loop system depends strongly on the number P of samples. The quality of the processes is decreased if the prediction horizon is reduced. Moreover, the system can lose stability if the quantity P is sufficiently small.
So, the following approaches to reduce computational load can be proposed:

1. Using the **control horizon**. The positive integer number $M < P$ is called the **control horizon** if the following condition hold:

   $$\tilde{\mathbf{u}}_{k+M-1} = \tilde{\mathbf{u}}_{k+M} = ... = \tilde{\mathbf{u}}_{k+P-1}.$$

   Thus, the number of independent variables is decreased from mP to mM. This approach allows to essentially reduce the optimization problem order. However, if the control horizon M is too small, the closed-loop stability can be compromised and the quality of the processes can decrease.
2. Increasing the sampling interval δ and reducing the number P of samples over the prediction horizon. This also allows to decrease the optimization problem order while preserving the value of the prediction horizon.
3. The computational consumption also depends on the prediction model used. So, one needs to use as simple models as possible. But the prediction model should adequately reflect the dynamics of the plant considered. The simplest case is using the linear prediction model.

3.3 Linear MPC

In this particular case, MPC is based on the linear prediction model. These algorithms are computationally efficient which is especially important from the real-time implementation point of view.
Generally, linear prediction model is presented by

$$\begin{aligned} &\tilde{\mathbf{x}}_{i+1} = \mathbf{A}\tilde{\mathbf{x}}_i + \mathbf{B}\tilde{\mathbf{u}}_i, \quad i = k+j, \quad j = 0, 1, 2, ..., \quad \tilde{\mathbf{x}}_k = \mathbf{x}_k, \\ &\tilde{\mathbf{y}}_i = \mathbf{C}\tilde{\mathbf{x}}_i. \end{aligned} \tag{15}$$

Suppose $\bar{\mathbf{u}} = \begin{pmatrix} \tilde{\mathbf{u}}_k & \tilde{\mathbf{u}}_{k+1} & ... & \tilde{\mathbf{u}}_{k+P-1} \end{pmatrix}^T$ is the programmed control over the prediction horizon. Then, integrating (15) we obtain future outputs of the plant in the form

$$\bar{\mathbf{y}} = \mathbf{L}\mathbf{x}_k + \mathbf{M}\bar{\mathbf{u}}, \tag{16}$$

where

$$\mathbf{L} = \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^P \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} \mathbf{CB} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{CAB} & \ddots & & \\ \vdots & & \ddots & \\ \mathbf{CA}^{P-1}\mathbf{B} & \dots & \mathbf{CAB} & \mathbf{CB} \end{bmatrix}.$$

Substituting (16) into (13) we get

$$J_k = J_k(\mathbf{x}_k, \bar{\mathbf{u}}) = \bar{\mathbf{u}}^T \mathbf{H} \bar{\mathbf{u}} + 2\mathbf{f}^T \bar{\mathbf{u}} + g. \tag{17}$$

Here we assumed that all weight matrices are equal, that is

$$\begin{array}{l} \mathbf{R}_{k+1} = \mathbf{R}_{k+2} = ... = \mathbf{R}_{k+P} = \mathbf{R}, \\ \mathbf{Q}_{k+1} = \mathbf{Q}_{k+2} = ... = \mathbf{Q}_{k+P} = \mathbf{Q}. \end{array}$$

The matrix $\mathbf{H}$ and vector $\mathbf{f}$ in (17) are as follows

$$\mathbf{H} = \mathbf{M}'\mathbf{R}\mathbf{M} + \mathbf{Q}, \quad \mathbf{f} = \mathbf{M}'\mathbf{R}\mathbf{L}\mathbf{x}_k. \tag{18}$$

It can easily be shown that in this case the optimization problem (14) is reduced to the quadratic programming problem of the form

$$J_k(\mathbf{x}_k, \tilde{\mathbf{u}}_k, \tilde{\mathbf{u}}_{k+1}, ..., \tilde{\mathbf{u}}_{k+P-1}) = \bar{\mathbf{u}}^T \mathbf{H} \bar{\mathbf{u}} + 2\mathbf{f}^T \bar{\mathbf{u}} + g \to \min_{\bar{\mathbf{u}} \in \Omega \subset \mathbf{E}^{mP}}. \tag{19}$$

Here $\mathbf{H}$ is a positive definite matrix and Ω is a convex set defined by the system of linear constraints. On-line solution of the optimization problem (19) at each sampling instant generally leads to nonlinear feedback control law.
Note that the optimization problem (19) can be solved analytically for the unconstrained case. The result is the linear controller

$$\tilde{\mathbf{u}}_k = \mathbf{K}\tilde{\mathbf{x}}_k, \tag{20}$$

which converges to the LQR-optimal one as P is increased. This convergence is obvious, because the discrete LQR controller minimizes the functional (13) with infinity prediction horizon for linear model (15).

4. Model Predictive Control on the base of modal parametrical optimization

In this section a new approach to MPC control algorithm synthesis is considered. The key feature of corresponding algorithms is that they guarantee linear closed-loop system stability at each sampling period. It is necessary to remark that in the case of traditional MPC algorithm implementation, described above, closed-loop system stability can be provided only for the simplest case when we have a linear prediction model, quadratic cost functional and without constraints.
Let us assume that the mathematical model of the plant to be controlled is described by the following system of difference equations

$$\begin{array}{l} \hat{\mathbf{x}}_{k+1} = \mathbf{F}(\hat{\mathbf{x}}_k, \hat{\mathbf{u}}_k, \hat{\varphi}_k), \\ \hat{\mathbf{y}}_k = \mathbf{C}\hat{\mathbf{x}}_k. \end{array} \tag{21}$$

Here $\hat{\mathbf{y}}_k \in \mathbf{E}^s$ is the vector of output variables, $\hat{\mathbf{x}}_k \in \mathbf{E}^n$ is the state space vector, $\hat{\mathbf{u}}_k \in \mathbf{E}^m$ is the vector of controls, $\hat{\varphi}_k \in \mathbf{E}^l$ is the vector of external disturbances.
Equations (21) can be used as a basis for nonlinear prediction model construction. Suppose that obtained prediction model is given by

$$\begin{array}{l} \tilde{\mathbf{x}}_{i+1} = \mathbf{f}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{u}}_i), \quad i = k + j, \quad j = 0, 1, 2, ..., \quad \tilde{\mathbf{x}}_k = \mathbf{x}_k, \\ \tilde{\mathbf{y}}_i = \mathbf{C}\tilde{\mathbf{x}}_i. \end{array} \tag{22}$$

Here $\mathbf{x}_k \in \mathbf{E}^n$ is the actual state of the plant at time instant k or its estimation on the base of measurement output.
Let desired object dynamics is presented by the given vector sequences $\{\mathbf{r}_k^x\}$ and $\{\mathbf{r}_k^u\}$, k = 0,1,2,... . The linear mathematical model of the plant, describing its behavior in the neighbourhood of the desired trajectory, can be obtained by performing the equations (21) linearization. As a result of this action, we get the linear system of difference equations

$$\begin{array}{l} \bar{\mathbf{x}}_{k+1} = \mathbf{A}\bar{\mathbf{x}}_k + \mathbf{B}\bar{\mathbf{u}}_k + \mathbf{H}\bar{\varphi}_k, \\ \bar{\mathbf{y}}_k = \mathbf{C}\bar{\mathbf{x}}_k, \end{array} \tag{23}$$

where $\bar{\mathbf{x}}_k \in \mathbf{E}^n$, $\bar{\mathbf{u}}_k \in \mathbf{E}^m$, $\bar{\mathbf{y}}_k \in \mathbf{E}^s$, $\bar{\varphi}_k \in \mathbf{E}^l$ are the vectors of the state, control input, measurements and external disturbances respectively. These vectors represent the deviations from the desired trajectory. Next we shall consider only such situations when all matrices in equations (23) have constant elements. In the framework of proposed approach, the control input over the prediction horizon is generated by the controller of the form

$$\bar{\mathbf{u}}_k = \mathbf{W}(q, \mathbf{h})\bar{\mathbf{y}}_k. \tag{24}$$

Here q is the shift operator, $\mathbf{W}(q, \mathbf{h})$ is the controller transfer function with the fixed structure (that is the degrees of the polynomials in the numerator and denominator of all its components are given and fixed), $\mathbf{h} \in \mathbf{E}^r$ is the vector of tuned parameters, which must be chosen on the stage of control design.
The prediction model equations (22), closed by the feedback (24), can be presented as follows

$$\begin{array}{l} \tilde{\mathbf{x}}_{i+1} = \mathbf{f}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{u}}_i), \quad i = k+j, \quad j = 0, 1, 2, ..., \quad \tilde{\mathbf{x}}_k = \mathbf{x}_k, \\ \tilde{\mathbf{u}}_i = \mathbf{r}_i^u + \mathbf{W}(q, \mathbf{h})\mathbf{C}(\tilde{\mathbf{x}}_i - \mathbf{r}_i^x). \end{array} \tag{25}$$

Let us assume that parameters vector $\mathbf{h}$ is chosen and fixed. Then we can solve system of difference equations (25) with a given initial conditions for the instants $i = k, k+1, ..., k+P-1$. As a result we obtain vectors sequence $\{\tilde{\mathbf{x}}_i\}$, $(i = k+1, ..., k+P)$, which represents the prediction of future plant behavior over the prediction horizon P. It must be noted, that the control sequence $\tilde{\mathbf{u}}_k, \tilde{\mathbf{u}}_{k+1}, ...\tilde{\mathbf{u}}_{k+P-1}$ over this horizon is determined uniquely by the choice of parameter vector $\mathbf{h}$. So, in this case the problem of control is reduced to the problem of parameters vector $\mathbf{h}$ tuning.
The controlled processes quality over the prediction horizon P can be presented by the following cost functional

$$J_k = J_k(\{\tilde{\mathbf{x}}_i\}, \{\tilde{\mathbf{u}}_i\}) = J_k(\mathbf{W}(q, \mathbf{h})) = J_k(\mathbf{h}) \geq 0, \tag{26}$$

where $\{\tilde{\mathbf{x}}_i\}$, $i = k+1, ..., k+P$, $\{\tilde{\mathbf{u}}_i\}$, $i = k, ..., k+P-1$ are the state and control vectors sequences correspondently, which satisfies the system of equations (25). It is easy to see, that the cost functional (26) is reduced to the function of parameter vector $\mathbf{h}$.
Let us consider the following optimization problem

$$J_k = J_k(\mathbf{h}) \rightarrow \inf_{\mathbf{h} \in \Omega_H}, \tag{27}$$

where Ω_H is a set of parameter vectors providing that the eigenvalues of the closed-loop system (23), (24) are placed in the desired area C_Δ inside a unit circle.
It is necessary to remark that the problem (27) is a nonlinear programming problem with an extremely complicated definition of the cost function, which, in generall, has no analytical

representation and is given only algorithmically. Besides that, the specific character of the problem (27) is also defined by the complicated constraints imposed, which determines the admissible areas of eigenvalues displacement. It must be noted, that the dimension of the optimization problem (27) is defined only by the dimension of parameter vector $\mathbf{h}$ and it does not depend on the prediction horizon P value.

Definition 1. We shall say that the controller (24) has a **full structure** if the degrees of polynomials in the numerators and denominators of the matrix $\mathbf{W}(q, \mathbf{h})$ components and the structure of parameter vector $\mathbf{h}$ are such that it is possible to assign any given roots for closed-loop system (23),(24) characteristic polynomial $\Delta(z, \mathbf{h})$ by appropriate selection of parameter vector $\mathbf{h}$.

In order to get another form of the presented definition, consider the equations of the closed-loop system (23),(24). They can be represented in the normal form as follows

$$\begin{array}{l} \bar{\mathbf{x}}_{k+1} = \mathbf{A}\bar{\mathbf{x}}_k + \mathbf{B}\bar{\mathbf{u}}_k + \mathbf{H}\bar{\varphi}_k, \\ \bar{\mathbf{y}}_k = \mathbf{C}\bar{\mathbf{x}}_k, \\ \bar{\xi}_{k+1} = \mathbf{A}_c(\mathbf{h})\bar{\xi}_k + \mathbf{B}_c(\mathbf{h})\bar{\mathbf{y}}_k, \\ \bar{\mathbf{u}}_k = \mathbf{C}_c(\mathbf{h})\bar{\xi}_k + \mathbf{D}_c(\mathbf{h})\bar{\mathbf{y}}_k, \end{array} \tag{28}$$

where $\bar{\xi}_k \in \mathbf{E}^{\nu}$ is a controller (24) state vector. After applying Z-transformation to the system of equations (28) with zero initial conditions, obtain

$$\begin{array}{l} (\mathbf{E}_n z - \mathbf{A})\bar{\mathbf{x}} = \mathbf{B}\bar{\mathbf{u}} + \mathbf{H}\bar{\varphi}, \\ (\mathbf{E}_\nu z - \mathbf{A}_c(\mathbf{h}))\bar{\xi} = \mathbf{B}_c(\mathbf{h})\mathbf{C}\bar{\mathbf{x}}, \\ \bar{\mathbf{u}} = \mathbf{C}_c(\mathbf{h})\bar{\xi} + \mathbf{D}_c(\mathbf{h})\mathbf{C}\bar{\mathbf{x}}, \\ \bar{\mathbf{y}} = \mathbf{C}\bar{\mathbf{x}}, \end{array}$$

or

$$\begin{pmatrix} \mathbf{E}_n z - \mathbf{A} - \mathbf{B}\mathbf{D}_c(\mathbf{h})\mathbf{C} & -\mathbf{B}\mathbf{C}_c(\mathbf{h}) \\ -\mathbf{B}_c(\mathbf{h})\mathbf{C} & \mathbf{E}_\nu z - \mathbf{A}_c(\mathbf{h}) \end{pmatrix} \begin{pmatrix} \bar{\mathbf{x}} \\ \bar{\xi} \end{pmatrix} = \begin{pmatrix} \mathbf{H} \\ \mathbf{0} \end{pmatrix} \bar{\varphi}.$$

Therefore, the closed-loop system characteristic polynomial $\Delta(z, \mathbf{h})$ is given by

$$\Delta(z, \mathbf{h}) = det \begin{pmatrix} \mathbf{E}_n z - \mathbf{A} - \mathbf{B}\mathbf{D}_c(\mathbf{h})\mathbf{C} & -\mathbf{B}\mathbf{C}_c(\mathbf{h}) \\ -\mathbf{B}_c(\mathbf{h})\mathbf{C} & \mathbf{E}_\nu z - \mathbf{A}_c(\mathbf{h}) \end{pmatrix}.$$

Let us denote the degree of the polinomial $\Delta(z, \mathbf{h})$ by n_d.
Let find parameter vector $\mathbf{h}$, which provide a given roots for the system (28) characteristic polynomial. In other words, it is nesessary to find such parameter vector $\mathbf{h}$ that provide the following identity

$$\Delta(z, \mathbf{h}) \equiv \tilde{\Delta}(z),$$

where $\tilde{\Delta}(z)$ is a given polynomial with degree n_d, having desired roots. In order to find vector $\mathbf{h}$, equate the correspondent coefficients for the same degrees of z-variable. As a result obtain the system of $(n_d + 1)$ nonlinear equations with r unknown components of vector $\mathbf{h}$ in the form

$$\mathbf{L}(\mathbf{h}) = \gamma. \tag{29}$$

It is evident that the controller (24) has a full structure if and only if the system of equations (33) has a solution for any vector γ.

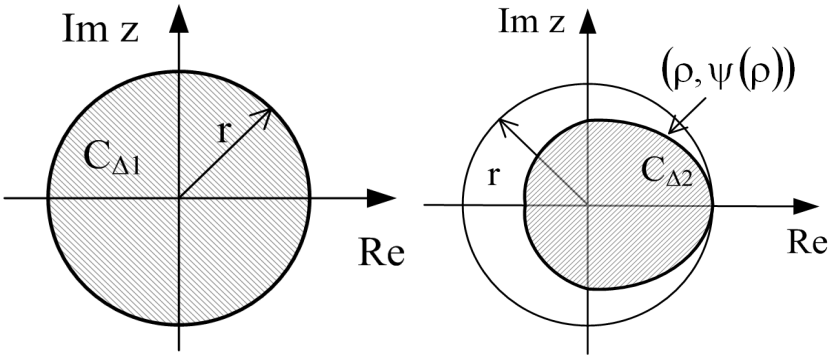
It can be easy shown that if the parameter vector $\mathbf{h}$ consists of the coefficients of numerator and denominator polynomials of matrix $\mathbf{W}(q, \mathbf{h})$, then the system (29) reduced to the linear system of the form

$$\mathbf{L}\mathbf{h} = \gamma, \tag{30}$$

where L is a constant matrix. Note that for any case, the controller (24) has a full structure only if the system (23) is fully controllable and observable.
Let us now refine the optimization problem (27) statement in suppose that the controller (23) has a full structure and that the following set Ω_H is determined as admissible set of the form

$$\Omega_H = \{\mathbf{h} \in \mathbf{E}^r : \delta_i(\mathbf{h}) \in C_\Delta,\ i = 1, 2, ..., n_d\}. \tag{31}$$

Here δ_i is the roots of the characteristic polynomial $\Delta(z, \mathbf{h})$, $n_d = deg\Delta(z, \mathbf{h})$.
Let consider two different variants of the desired areas C_Δ, depicted in Fig. 2. This areas are located inside a unit circle, i. e. $r < 1$.



(a) area $C_{\Delta 1}$ (b) area $C_{\Delta 2}$

Fig. 2. The areas $C_{\Delta 1}$ and $C_{\Delta 2}$ of the desired root displacement

The formalized description for the desired areas C_Δ are as follows:
$C_\Delta = C_{\Delta 1} = \{z \in \mathbf{C}^1 : |z| \leq r\}$, where $r \in (0, 1)$ is a given real number;
$C_\Delta = C_{\Delta 2} = \{z \in \mathbf{C}^1 : z = \rho exp(\pm i\varphi),\ 0 \leq \rho \leq r,\ 0 \leq \varphi \leq \psi(\rho)\}$, where $r \in (0, 1)$ is a given real number, $\psi(\xi)$ is a real function of variable $\xi \in (0, r]$, which takes the values on the interval $[0, \pi]$ and $\psi(r) = 0$.
The reasons of these areas introduction is obvious. The first area $C_{\Delta 1}$ determines the lower bound for the closed-loop system stability margin and, therefore, the settling time for transient processes. Second area $C_{\Delta 2}$ determines stability bound and, in addition, constraints on the closed-loop system oscillations.
In order to form the algorithm for the problem (27) solution on the admissible set (31), let us firstly perform the parametrization of the considered areas C_Δ with the n-dimensional real vectors on the base of the following statement.

Theorem 1. *For any real vector $\gamma \in \mathbf{E}^{n_d}$ the roots of the polynomial $\Delta^*(z, \gamma)$, given by the formulas presented below, are located inside the area $C_{\Delta 1}$ or on its bound. And reversly, if the roots of the some*

polynomial $\Delta(z)$ *are located inside the area* $C_{\Delta 1}$ *and, in addition, all its real roots are positive, then it can be found such a vector* $\gamma \in \mathbf{E}^{n_d}$ *that the following identity holds* $\Delta(z) \equiv \Delta^*(z,\gamma)$. *Here*

$$\Delta^*(z,\gamma) = \prod_{i=1}^{d}(z^2 + a_i^1(\gamma,r)z + a_i^0(\gamma,r)), \tag{32}$$

if n_d *is even,* $d = n_d/2$;

$$\Delta^*(z,\gamma) = (z - a_{d+1}(\gamma,r))\prod_{i=1}^{d}(z^2 + a_i^1(\gamma,r)z + a_i^0(\gamma,r)), \tag{33}$$

if n_d *is odd,* $d = [n_d/2]$;

$$\begin{gathered} a_i^1(\gamma,r) = -r\left(exp\left(-\frac{\gamma_{i1}^2}{2} + \sqrt{\frac{\gamma_{i1}^4}{4} - \gamma_{i2}^2}\right) + exp\left(-\frac{\gamma_{i1}^2}{2} - \sqrt{\frac{\gamma_{i1}^4}{4} - \gamma_{i2}^2}\right)\right), \\ a_i^0(\gamma,r) = r^2 exp\left(-\gamma_{i1}^2\right),\ i = 1,...,d, \quad a_{d+1}(\gamma,r) = r\,exp\left(-\gamma_{d0}^2\right), \end{gathered} \tag{34}$$

$$\gamma = \{\gamma_{11},\gamma_{12},\gamma_{21},\gamma_{22},...,\gamma_{d1},\gamma_{d2},\gamma_{d0}\}. \tag{35}$$

Proof If the n_d is even, then the proof of the direct and reverse propositions arises from the elementary properties of the quadratic trinomials in the formula (32). Really, for any given pair of the real numbers γ_{i1}, γ_{i2} the roots of the trinomial $\Delta_i^*(z)$ in (32) are presented by the expression

$$z_{1,2}^i = r \cdot exp\left(-\frac{\gamma_{i1}^2}{2} \pm \sqrt{\frac{\gamma_{i1}^4}{4} - \gamma_{i2}^2}\right).$$

From this expression it follows that $|z_{1,2}^i| \le r$ and, therefore, the roots $z_{1,2}^i$ of the trinomial are located inside the area $C_{\Delta 1}$ or on its bound, and this proves the direct proposition.
In order to prove reverse one, let consider some quadratic trinomial of the form $\Delta_i(z) = z^2 + \beta_1 z + \beta_0$. By the conditions of the reverse proposition, the roots $z_{1,2}$ of this trinomial are located inside the area $C_{\Delta 1}$ and, if the roots are real numbers, then they are positive. In order to locate the roots $z_{1,2}$ inside the area $C_{\Delta 1}$, it is necessary and sufficient that the following relations holds

$$1 - \frac{\beta_1}{r} + \frac{\beta_0}{r^2} \ge 0,\ 1 - \frac{\beta_0}{r^2} \ge 0,\ 1 + \frac{\beta_1}{r} + \frac{\beta_0}{r^2} \ge 0. \tag{36}$$

Besides that, the roots product $z_1 z_2$ is positive in anycase if they are being complex conjugated pair or positive real numbers. Therefore, the following inequality is true

$$\beta_0 > 0. \tag{37}$$

Let find such numbers γ_{i1} and γ_{i2} that the identity $\Delta_i^*(z) \equiv \Delta_i(z)$ is satisfied. By equating the correspondent coefficients for the same degrees of z-variable, obtain

$$-r\left(exp\left(-\frac{\gamma_{i1}^2}{2} + \sqrt{\frac{\gamma_{i1}^4}{4} - \gamma_{i2}^2}\right) + exp\left(-\frac{\gamma_{i1}^2}{2} - \sqrt{\frac{\gamma_{i1}^4}{4} - \gamma_{i2}^2}\right)\right) = \beta_1,$$

$$r^2 exp(-\gamma_{i1}^2) = \beta_0,$$

and consequently

$$\gamma_{i1} = \sqrt{-ln\left(\beta_0/r^2\right)},$$
$$\gamma_{i2} = \sqrt{-\frac{1}{4} ln\left(w\frac{r^2}{\beta_0}\right) ln\left(w\frac{\beta_0}{r^2}\right)}, \quad where \quad w = \frac{\beta_1^2}{2\beta_0} - 1 + \sqrt{\left(\frac{\beta_1^2}{2\beta_0} - 1\right)^2 - 1}. \tag{38}$$

Now let verify that the γ_{i1} and γ_{i2}, given by the formulas (38), are the real numbers.
Really, from the inequalities (36), (37) it follows that $0 < \beta_0/r^2 \leq 1$, therefore $-ln\left(\beta_0/r^2\right) \geq 0$ and γ_{i1} is a real number.
Let show that the expression under radical in the formula for γ_{i2} is nonnegative. For the first, consider the case when the trinomial $\Delta_i(z)$ has two real positive roots $z_{1,2}$. Then his coefficients must satisfies to the condition $\beta_1^2 - 4\beta_0 \geq 0$, whence it follows that $w \geq 1$ – is a real number. As a result, taking into account (36), we obtain

$$ln\left(w \cdot r^2/\beta_0\right) \geq 0. \tag{39}$$

It could be noted that the inequalities (36) implies also the satisfaction of the inequality $\beta_1^2 - 2\beta_0 \leq r^2 + \beta_0/r^2$. Hence, we have

$$w\beta_0 \leq r^2, \quad and \quad -ln\left(w\beta_0/r^2\right) \geq 0. \tag{40}$$

Thus from the inequalities (39) and (40) it is easy to see that the expression under radical in the formula for γ_{i2} is nonnegative and γ_{i2} is a real number.
Consider now a case, when the trinomial $\Delta_i(z)$ has a pair of complex-conjugate roots $z_{1,2}$. Then the following inequality is hold $\beta_1^2 - 4\beta_0 < 0$, and therefore w is a complex number, which can be presented in the form $w = \beta_1^2/2\beta_0 - 1 + i\sqrt{1 - \left(\beta_1^2/2\beta_0 - 1\right)^2}$. It is not difficult to see that $|w| = 1$, hence, the expression under the radical for γ_{i2} has a form

$$\gamma_{i2} = \sqrt{-\frac{1}{4}\left(ln\left(\frac{r^2}{\beta_0}\right) + i \cdot argw\right)\left(ln\left(\frac{\beta_0}{r^2}\right) + i \cdot argw\right)} = \sqrt{\frac{1}{4}\left(ln^2\left(\frac{r^2}{\beta_0}\right) + arg^2 w\right)},$$

i.e. it is nonnegative and γ_{i2} is a real number.
If the n_d is odd, the polynomial Δ^* has, in according to (33), an additional linear binomial, for which the propositions of the theorem are evident.■
Now consider more difficult second variant of the admissible set C_Δ. Let us prove the analogous theorem, which allows to perform parametrization of this area.

Theorem 2. *For any real vector $\gamma \in \mathbf{E}^{n_d}$ the roots of the polynomial $\Delta^*(z,\gamma)$ (32),(33) are located inside the area $C_{\Delta 2}$, and reversly, if the roots of the some polynomial $\Delta(z)$ are located inside the area $C_{\Delta 2}$ and, in addition, all its real roots are positive, then it can be found such a vector $\gamma \in \mathbf{E}^{n_d}$ that the following identity holds $\Delta(z) \equiv \Delta^*(z,\gamma)$. Here*

$$a_i^1(\gamma, r) = -r\left(exp\left(-\gamma_{i1}^2 + \nu_i\right) + exp\left(-\gamma_{i1}^2 - \nu_i\right)\right),$$
$$a_i^0(\gamma, r) = r^2 exp\left(-2\gamma_{i1}^2\right),\ i = 1, ..., d, \quad a_{d+1}(\gamma, r) = r \cdot exp(-\gamma_{d0}^2), \tag{41}$$

where $\nu_i = \sqrt{\gamma_{i1}^4 - f\left(\gamma_{i2}\right)\left(\psi^2\left(r \cdot exp\left(-\gamma_{i1}^2\right)\right) + \gamma_{i1}^4\right)}, i = 1, 2, ..., d;\ \gamma = \left\{\gamma_{11}, \gamma_{12}, \gamma_{21}, \gamma_{22}, ..., \gamma_{d1}, \gamma_{d2}, \gamma_{d0}\right\}$.
The function f is such that $f(\cdot) : (-\infty, +\infty) \to (0,1)$ and its inverse function exists in the whole region of the definition; the function $\psi(\xi)$ is a real function from the variable $\xi \in (0, r]$, which takes the values in the interval $[0, \pi]$ and $\psi(r) = 0$.

Proof Similar to theorem 1, consider the properties of the quadratic trinomials in (32). Firstly, let prove a direct proposition.
For any given pair of the real numbers γ_{i1}, γ_{i2} the roots of the trinomial $\Delta_i^*(z)$ in (32) is given by the expression $z_{1,2}^i = r \cdot exp(-\gamma_{i1}^2 \pm \nu_i)$. Here two different variants are possible. If ν_i is a real number, then the roots $z_{1,2}^i$ are also real. Besides that, taking into account the properties of the function f, the following inequality holds $\gamma_{i1}^4 - f(\gamma_{i2})\left(\psi^2\left(r \cdot exp(-\gamma_{i1}^2)\right) + \gamma_{i1}^4\right) \leq \gamma_{i1}^4$. Hence the roots are positive and $|z_{1,2}^i| \leq r$, that is $z_{1,2}^i \in C_{\Delta 2}$.
If ν_i is a complex number, then $z_{1,2}^i$ is the pair of complex-conjugated roots and $|z_{1,2}^i| = \rho = r \cdot exp(-\gamma_{i1}^2) \leq r$. Taking into account the properties of the function f, the following inequality is valid

$$\varphi = \sqrt{f(\gamma_{i2})\left(\psi^2\left(r \cdot exp(-\gamma_{i1}^2)\right) + \gamma_{i1}^4\right) - \gamma_{i1}^4} \leq \sqrt{\psi^2\left(r \cdot exp(-\gamma_{i1}^2)\right)} = \psi(\rho). \qquad (42)$$

Since the $arg\, z_{1,2}^i = \pm\varphi$ and, accordingly to (42), $0 \leq \varphi \leq \psi(\rho)$, then the roots $z_{1,2}^i$ are located inside the area $C_{\Delta 2}$, so the direct proposition is proven.
Let consider the reverse proposition. The roots $z_{1,2}$ of some trinomial $\Delta_i(z) = z^2 + \beta_1 z + \beta_0$ are located inside the area $C_{\Delta 2}$ in accordance with the reverse proposition if these roots are positive real numbers. Notice that the coefficients of this trinomial must satisfy to the inequalities (36),(37), because $|z_{1,2}| \leq r$ and the roots product $z_1 z_2$ is positive in any way.
Let find such numbers γ_{i1}, γ_{i2} that the identity $\Delta_i^*(z) \equiv \Delta_i(z)$ holds. By equating the correspondent coefficients for the same degrees of z-variable, obtain

$$-r\left(exp\left(-\gamma_{i1}^2 + \nu_i\right) + exp\left(-\gamma_{i1}^2 - \nu_i\right)\right) = \beta_i, \;\; r^2 exp(-2\gamma_{i1}^2) = \beta_0,$$

hence

$$\gamma_{i1} = \sqrt{-0.5 \cdot ln(\beta_0 / r^2)},$$

$$f(\gamma_{i2}) = \frac{1}{4\left(\psi^2\left(r \cdot exp(-\gamma_{i1}^2)\right) + \gamma_{i1}^4\right)}\left(ln^2\left(\frac{\beta_0}{r^2}\right) - ln^2\left(\frac{\beta_1^2}{2\beta_0} - 1 + \sqrt{\left(\frac{\beta_1^2}{2\beta_0} - 1\right)^2 - 1}\right)\right).$$

Let us show that the γ_{i2} is a real number. For γ_{i1} the proof is equivalent to the such one in the first theorem.
It is evident that the equation with respect to γ_{i2} has a solution, if the expression in the right part of it takes the values inside the interval (0,1). Let denote this expression by h. Notice that the denominator for h is equal to zero only if $z_1 = z_2 = r$, but in this case γ_{i2} can be chosen as any real number. In general case, taking into account the proof of the theorem 1, it is not difficult to see that $h > 0$. Besides that the following inequality holds

$$h < 1 - ln^2\left(\beta_1^2/2\beta_0 - 1 + \sqrt{\left(\beta_1^2/2\beta_0 - 1\right)^2 - 1}\right) / ln^2\left(\beta_0/r^2\right),$$

hence the real number γ_{i2} exists and this one is determined as a solution of the equation $f(\gamma_{i2}) = h$.
If the n_d is odd, the polynomial Δ^* has, in accordance with (33), an additional linear binomial, for which the propositions of the theorem are evident.■

Now let us show how introced areas $C_{\Delta 1}$ and $C_{\Delta 2}$ are related to the standart areas on the complex plane, which are commonly used in the analysis and synthesis of the continuos time systems.
Primarily, it may be noticed that the eigenvalues of the continues linear model and the discrete linear model are connected by the following rule (Hendricks et al., 2008): if s is the eigenvalue of the continuos time system matrix, then $z = e^{sT}$ is the correspondent eigenvalue of the discrete time system matrix, where T is the sampling period. Taking into account this relation, let consider the examples of the mapping of some standart areas for continuous systems to the areas for discrete systems.
Example 1 Let we have given area $C = \{s = x \pm yj \in \mathbf{C}^1 : x \leq -\alpha\}$, depicted in Fig. 3. It is evident that the points of the line $x = -\alpha$ are mapped to the points of the circle $|z| = e^{-\alpha T}$. The area C itself is mapped on the disc $|z| \leq e^{-\alpha T}$, as shown in Fig.3. This disc corresponds to the area $C_{\Delta 1}$, which defines the degree of stability for discrete system.

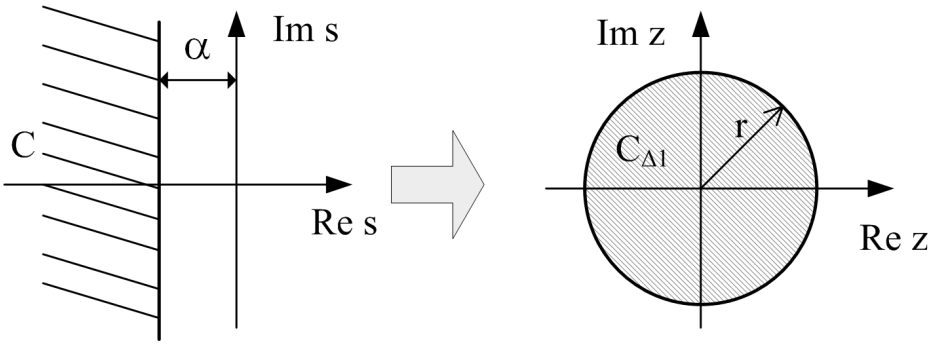


Fig. 3. The correspondence of the areas for continuous and discrete system

Example 2 Consider the area

$$C = \{s = x \pm yj \in \mathbf{C}^1 : x \leq -\alpha, 0 \leq y \leq (-x - \alpha)tg\beta\},$$

depicted in Fig. 4, where $0 \leq \beta < \frac{\pi}{2}$ and $\alpha > 0$ is a given real numbers.
Let perform the mapping of the area C on the z-plane. It is evident that the vertex of the angle $(-\alpha, 0)$ is mapped to the point with polar coordinates $r = e^{-\alpha T}, \varphi = 0$ on the plane z. Let now map each segment from the set

$$L_\gamma = \{s = x \pm yj \in \mathbf{C}^1 : x = \gamma, \gamma \leq -\alpha, 0 \leq y \leq (-\gamma - \alpha)tg\beta\}$$

to the z-plane. Each point $s = \gamma \pm yj$ of the segment L_γ is mapped to the point $z = e^{sT} = e^{\gamma T \pm jyT}$ on the plane z. Therefore, the points of the segment L_γ are mapped to the arc of the circle with radius $e^{\gamma T}$ if the following condition holds $-\alpha - \pi/(Ttg\beta) < \gamma \leq -\alpha$, and to the whole circle if $\gamma \leq -\alpha - \pi/(Ttg\beta)$. Therefore, the maximum radius of the circle, which is fullfilled by the points of the segment, is equal to $r' = e^{-\alpha T - \pi/tg\beta}$, corresponding with the equality $\gamma_0 = -\alpha - \pi/(Ttg\beta)$. Notice that the rays, which constitutes the angle, mapped to the logarithmic spirals. Moreover, the bound of the area on the plane z is formed by the arcs of these spirals in accordace with the x varying from $-\alpha$ to γ_0.

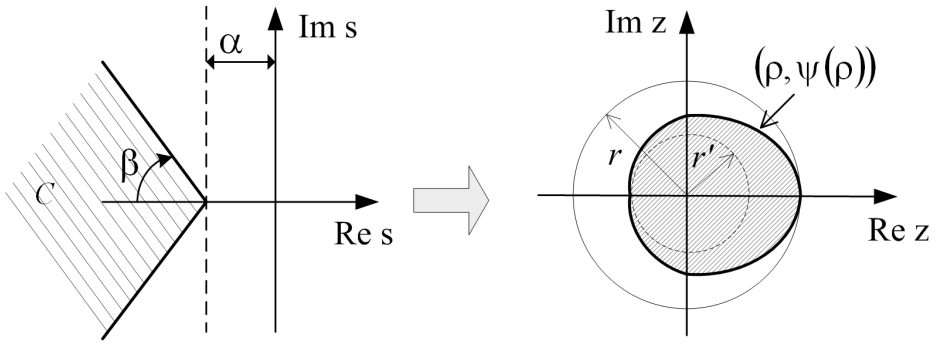


Fig. 4. The correspondence of the areas for continuous and discrete time systems

Let introduce the notation $\rho = e^{xT}$, and define the function $\psi(\rho)$, which represents the constraints on the argument values while the radius ρ of the circle is fixed:

$$\psi(\rho) = \begin{cases} (-ln\rho - \alpha T)tg\beta, & if\ \rho \in [r', r], \\ \pi, & if\ \rho \in [0, r']. \end{cases}$$

The result of the mapping is shown on the Fig. 4. It can be noted that the obtained area reflects the desired degree of the discrete time system stability and oscillations.

Let us use the results of the theorem 2 in order to formulate the computational algoritm for the optimization problem (27) solution on the admissible set Ω_H taking into account the condition $C_\Delta = C_{\Delta 2}$. It is evident that the first case, where $C_\Delta = C_{\Delta 2}$, is a particular case of the second one.

Consider a real vector $\gamma \in \mathbf{E}^{n_d}$ and form the polynomial $\Delta^*(z, \gamma)$ with the help of formulas (32),(33),(41). Let require that the tuned parameters of the controller (24), defined by the vector $\mathbf{h} \in \mathbf{E}^r$, provides the identity

$$\Delta(z, \mathbf{h}) \equiv \Delta^*(z, \gamma), \tag{43}$$

where $\Delta(z, \mathbf{h})$ is the characteristic polynomial of the closed-loop system with the degree n_d. By equating the correspondent coefficients for the same degrees of z-variable, we obtain the following system of nonlinear equations

$$\mathbf{L}(\mathbf{h}) = \chi(\gamma) \tag{44}$$

with respect to unknown components of the parameters vector $\mathbf{h}$. The last system has a solution for any given $\gamma \in \mathbf{E}^{n_d}$ due to the controller (24) has a full structure. Let consider that, in general case, the system (44) has a nonunique solution. Then the vector $\mathbf{h}$ can be presented as a set of two vectors $\mathbf{h} = \{\bar{\mathbf{h}}, \mathbf{h}_c\}$, where $\mathbf{h}_c \in \mathbf{E}^{n_c}$ is a free component, $\bar{\mathbf{h}}$ is the vector that is uniquely defined by the solution of the system (44) for the given vector $\mathbf{h}_c$.

Let introduce the following notation for the general solution of the system (44)

$$\mathbf{h} = \mathbf{h}^* = \{\bar{\mathbf{h}}^*(\mathbf{h}_c, \gamma), \mathbf{h}_c\} = \mathbf{h}^*(\gamma, \mathbf{h}_c) = \mathbf{h}^*(\epsilon),$$

where $\epsilon = \{\gamma, \mathbf{h}_c\}$ is a vector of the independent parameters with the dimension λ given by

$$\lambda = dim\, \epsilon = dim\, \gamma + dim\, \mathbf{h}_c = n_d + n_c.$$

Let form the equations of the prediction model, closed by the controller (24) with the obtained parameter vector $\mathbf{h}^*$

$$\begin{array}{l}\tilde{\mathbf{x}}_{i+1} = \mathbf{f}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{u}}_i), \quad i = k+j, \quad j = 0, 1, 2, ..., \quad \tilde{\mathbf{x}}_k = \mathbf{x}_k, \\ \tilde{\mathbf{u}}_i = \mathbf{r}_i^u + \mathbf{W}(q, \mathbf{h}^*(\epsilon))\mathbf{C}(\tilde{\mathbf{x}}_i - \mathbf{r}_i^x).\end{array} \tag{45}$$

Now the functional J_k, which is given by (26) and computed on the solutions of the system (45), becomes the function of the vector ϵ:

$$J_k = J_k\left(\{\tilde{\mathbf{x}}_i\}, \{\tilde{\mathbf{u}}_i\}\right) = J_k^*\left(\mathbf{W}\left(q, \mathbf{h}^*(\epsilon)\right)\right) = J_k^*(\epsilon). \tag{46}$$

Theorem 3. *Consider the optimization problem (27), where Ω_H is the admissible set, given by (31), and the desired area $C_\Delta = C_{\Delta 2}$. If the extremum of this problem is achieved at the some point $\mathbf{h}_{k0} \in \Omega_H$, then there exists a vector $\epsilon \in \mathbf{E}^\lambda$ such that*

$$\mathbf{h}_{k0} = \mathbf{h}^*(\epsilon_{k0}), \ \textit{with} \ \epsilon_{k0} = arg \min_{\epsilon \in \mathbf{E}^\lambda} J_k^*(\epsilon). \tag{47}$$

And reversly, if there exists such a vector $\epsilon_{k0} \in \mathbf{E}^\lambda$, that satisfies to the condition (47), then the following vector $\mathbf{h}_{k0} = \mathbf{h}^*(\epsilon_{k0})$ is the solution of the optimization problem (27). In other words, **the problem (27) is equivalent to the unconstrained optimization problem of the form**

$$J_k^* = J_k^*(\epsilon) \to \inf_{\epsilon \in \mathbf{E}^\lambda}. \tag{48}$$

Proof Assume that the following condition is hold

$$\mathbf{h}_{k0} = arg \min_{\mathbf{h} \in \Omega_H} J_k(\mathbf{h}), J_{k0} = J_k(\mathbf{h}_{k0}). \tag{49}$$

In this case, the characteristic polynomial $\Delta(z, \mathbf{h}_{k0})$ of the closed-loop system (28) has the roots that are located inside the area $C_{\Delta 2}$. Then, accordingly to the theorem 2, it can be found such a vector $\gamma = \gamma_{k0} \in \mathbf{E}^{n_d}$, that $\Delta(z, \mathbf{h}_{k0}) \equiv \Delta^*(z, \gamma_{k0})$, where Δ^* is a polynomial formed by the formulas (32), (33). Hence, there exists such a vector $\epsilon = \{\gamma_{k0}, \mathbf{h}_{k0c}\}$, for which the following conditions is hold $\mathbf{h}_{k0} = \mathbf{h}^*(\epsilon_{k0}), J_k^*(\epsilon_{k0}) = J_{k0}$. Here $\mathbf{h}_{k0c}$ is the correspondent constituent part of the vector $\mathbf{h}_{k0}$.
Now it is only remain to show that there no exists a vector $\epsilon_{01} \in \mathbf{E}^\lambda$ that the condition $J_k^*(\epsilon_{01}) < J_{k0}$ is valid. Really, let suppose that such vector exists. But then for the vector $\mathbf{h}^*(\epsilon_{01})$ the following inequality takes place $J_k(\mathbf{h}^*(\epsilon_{01}) = J_k^*(\epsilon_{01}) < J_{k0}$. But this is not possible due to the condition (49). The reverse proposition is proved analogously.■
Let formulate the computational algorithm in order to get the solution of the optimization problem (27) on the base of the theorems proved above.
The algorithm consists of the following operations:

1. Set any vector $\gamma \in \mathbf{E}^{n_d}$ and construct the polynomial $\Delta^*(z, \gamma)$ by formulas (32),(33), (41).
2. In accordance with the identity $\Delta(z, \mathbf{h}) \equiv \Delta^*(z, \gamma)$, form the system of nonlinear equations
$$\mathbf{L}(\mathbf{h}) = \chi(\gamma), \tag{50}$$
which has a solution for any vector γ. If the system (50) has a nonunique solution, assign the vector of the free parameters $\mathbf{h}_c \in \mathbf{E}^{n_c}$.
3. For a given vector $\epsilon = \{\gamma, \mathbf{h}_c\} \in \mathbf{E}^\lambda$ solve the system of equations (50). As a result, obtain vector $\mathbf{h}^*(\epsilon)$.

4. Form the equations of the prediction model closed by the controller (24) with the parameter vector $\mathbf{h}^*(\epsilon)$ and compute the value of the cost function $J_k^*(\epsilon)$ (46).
5. Solve the problem (48) by using any numerical method for unconstrained minimization and repeating the steps 3–5.
6. When the optimal solution $\epsilon_{k0} = arg \min\limits_{\epsilon \in \mathbf{E}^\lambda} J_k^*(\epsilon)$ is found, compute the parameter vector $\mathbf{h}_{k0} = \mathbf{h}^*(\epsilon_{k0})$ and accept them as a solution.

Now real-time MPC algorithm, which is based on the on-line solution of the problem (27), can be formulated. This algorithm consists of the following steps:

- Obtain the state estimation $\hat{\mathbf{x}}_k$ on the base of measurements $\mathbf{y}_k$.
- Solve the optimization problem (27), using the algorithm stated above, subject to the prediction model (22) with initial conditions $\tilde{\mathbf{x}}_k = \hat{\mathbf{x}}_k$.
- Let $\mathbf{h}_{k0}$ be the solution of the problem (27). Implement controller (24) with the parameter vector $\mathbf{h}_{k0}$ over time interval $[k\delta, (k+1)\delta]$, where δ is the sampling period.
- Repeat the whole procedure 1–3 at next time instant $(k+1)\delta$.

As a result, let notice the following important features of the proposed MPC-algorithm. For the first, the linear closed-loop system stability is provided at each sampling interval. Secondly, the control is realised in the feedback loop. Thirdly, the dimension of the unconstrained optimization problem is fixed and does not depend on the length of prediction horizon P.

5. Plasma Vertical Stabilization Based on the Model Predictive Control

Let us remember that SISO model (5) represents plasma dynamics in the vertical stabilization process and limits (6) are imposed on the power supply system. It is necessary to transform the system (5) to the state-space form for MPC algorithms implementation. Besides that, in order to take into account the constraint imposed on the current, one more equation should be added to the model (5). Finally, the linear model of the stabilization process is given by

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{b}u, \\ \mathbf{y} &= \mathbf{c}\mathbf{x} + \mathbf{d}u, \end{aligned} \tag{51}$$

where $\mathbf{x} \in \mathbf{E}^4$ and the last component of $\mathbf{x}$ corresponds to VS converter current, $\mathbf{y} = (y_1, y_2) \in \mathbf{E}^2$, y_1 is the vertical velocity and y_2 is the current in the VS-converter. We shall assume that the model (51) describes the process accurately.
We can obtain a linear prediction model in the form (15) by the system (51) discretization. As a result, we get

$$\begin{aligned} &\tilde{\mathbf{x}}_{i+1} = \mathbf{A}_d\tilde{\mathbf{x}}_i + \mathbf{b}_d\tilde{u}_i, \qquad \tilde{\mathbf{x}}_k = \mathbf{x}_k, \\ &\tilde{\mathbf{y}}_i = \mathbf{C}_d\tilde{\mathbf{x}}_i. \end{aligned} \tag{52}$$

The constraints (6) form the system of linear inequalities given by

$$\begin{aligned} &\tilde{u}_i \le V_{max}^{VS}, \quad i = k, ..., k+P-1; \\ &\tilde{y}_{i2} \le I_{max}^{VS}, \quad i = k+1, ..., k+P. \end{aligned} \tag{53}$$

These constraints define the admissible convex set Ω. The discrete analog of the cost functional (7) with $\lambda = 1$ is given by

$$J_k = J_k(\bar{\mathbf{y}}, \bar{\mathbf{u}}) = \sum_{j=1}^{P} \left(\tilde{y}_{k+j,1}^2 + \tilde{u}_{k+j-1}^2 \right). \tag{54}$$

So, in this case MPC algorithm leads to real-time solution of the quadratic programming problem (19) with respect to the prediction model (52), constraints (53) and the cost functional (54). From the experiments the following values for the sampling time and number of sampling intervals over the horizon were obtained

$$\delta = 0.004 \text{ sec}, \quad P = 250.$$

Hence, we have the following prediction horizon

$$T_p = P\delta = 1 \text{ sec}.$$

Let us consider the MPC controller synthesis without taking into account the constraints imposed. Remember that in this case we obtain a linear controller (20) that is practically the same as the LQR-optimal one. The transient response of the system closed by the controller is presented in Fig. 5. The initial state vector $\mathbf{x}(0) = \mathbf{h}$ is used, where $\mathbf{h}$ is a scaled eigenvector of the matrix $\mathbf{A}$ corresponding to the only unstable eigenvalue. The eigenvector $\mathbf{h}$ is scaled to provide the initial vertical velocity $y_1 = 0.03$ m/sec. It can be seen from the figure that the constraints (6) imposed on the voltage and current are violated.

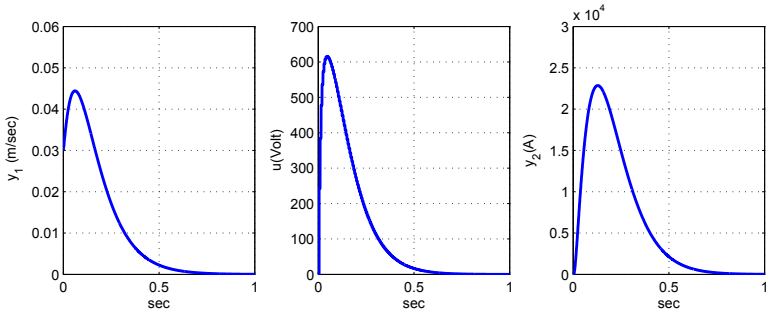


Fig. 5. Transient response of the closed-loop system with unconstrained MPC-controller

Now consider the MPC algorithm synthesis with constraints. Fig. 6 shows transient response of the closed-loop system with constrained MPC-controller. It is not difficult to see that all constraints imposed are satisfied. In order to reduce computational consumptions, the approaches proposed above in Section 3.2 can be implemented.

1. Experiments with using the control horizon were carried out. This experiments show that the quality of stabilization remains approximately the same with control horizon $M = 50$ and prediction horizon $P = 250$. So, optimization problem order can be significantly reduced.
2. Another approach is to increase the sampling interval up to $\delta = 0.005$ sec and reduce the number of samples down to $P = 200$. Hence, prediction horizon has the same value $T_p = P\delta = 1$ sec. The optimization problem order is also reduced in this case and consequently time consumptions at each sampling instant is decreased. However, further increase of δ tends to compromise closed-loop system stability.

Now consider the processes of the plasma vertical stabilization on the base of new MPC-scheme.

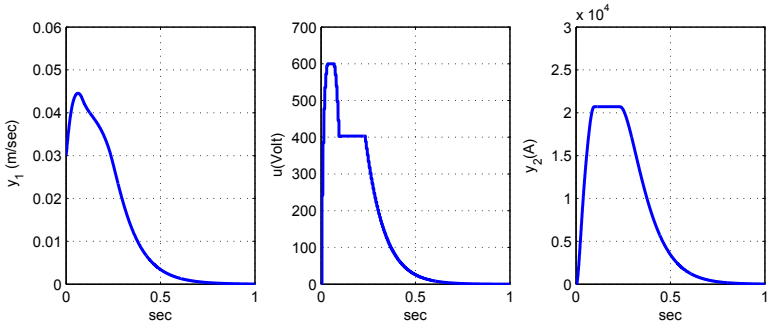


Fig. 6. Transient response of the closed-loop system with constrained MPC-controller

Let us, for the first, transform system (5) into the state space form. As a result, we get

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{b}u, \\ y &= \mathbf{c}\mathbf{x} + \mathbf{d}u, \end{aligned} \tag{55}$$

where $\mathbf{x} \in \mathbf{E}^3$, y is the vertical velocity, u is the voltage in the VS-converter. We shall assume that this model describes the process accurately.
As early, we can obtain linear prediction model by the system (55) discretization. So, we have the following prediction model

$$\begin{aligned} &\tilde{\mathbf{x}}_{i+1} = \mathbf{A}_d\tilde{\mathbf{x}}_i + \mathbf{b}_d\tilde{u}_i, \qquad \tilde{\mathbf{x}}_k = \mathbf{x}_k, \\ &\tilde{y}_i = \mathbf{C}_d\tilde{\mathbf{x}}_i. \end{aligned} \tag{56}$$

Let also form the discrete linear model of the process, describing its behavior in the neighbourhood of the zero equilibrium position. Such a model is obtained by the system (55) discretization and can be presented as follows

$$\begin{aligned} \bar{\mathbf{x}}_{k+1} &= \mathbf{A}_d\bar{\mathbf{x}}_k + \mathbf{b}_d\bar{u}_k, \\ \bar{y}_k &= \mathbf{C}_d\bar{\mathbf{x}}_k, \end{aligned} \tag{57}$$

where $\bar{\mathbf{x}}_k \in \mathbf{E}^3$, $\bar{u}_k \in \mathbf{E}^1$, $\bar{y}_k \in \mathbf{E}^1$. We shall form the control over the prediction horizon by the linear proportional controller, that is given by

$$\bar{u}_k = \mathbf{K}\bar{x}_k, \tag{58}$$

where $\mathbf{K} \in \mathbf{E}^3$ is the parameter vector of the controller. In the real processes control input (58) is computed on the base of the state estimation, obtained with the help of asymptotic observer. It must be noted that the controller (58) has a full structure, because the matrices of the controllability and observability for the system (57) have a full rank.
Now consider the equations of the prediction model (56), closed by the controller (58). As a result, we get

$$\begin{aligned} &\tilde{\mathbf{x}}_{i+1} = (\mathbf{A}_d + \mathbf{b}_d\mathbf{K})\tilde{\mathbf{x}}_i, \qquad \tilde{\mathbf{x}}_k = \mathbf{x}_k, \\ &\tilde{y}_i = \mathbf{C}_d\tilde{\mathbf{x}}_i. \end{aligned} \tag{59}$$

The controlled processes quality over the prediction horizon P is presented by the cost functional

$$J_k = J_k(\mathbf{K}) = \sum_{j=1}^{P} \left(\tilde{y}_{k+j}^2 + \tilde{u}_{k+j-1}^2 \right). \tag{60}$$

It is easy to see that the cost functional (60) becomes the function of three variables, which are the components of the parameter vector $\mathbf{K}$. It is important to note that the cost function remains essentialy nonlinear for this variant of the MPC approach even in the case when the prediction model is linear. It is a price for providing stability of the closed-loop linear system. Consider the optimization problem (27) statement for the particular case of plasma vertical stabilization processes

$$J_k = J_k(\mathbf{K}) \to \min_{\mathbf{K}\in\Omega_K}, \ where \ \ \Omega_K = \{\mathbf{K} \in \mathbf{E}^3 : \delta_i(\mathbf{K}) \in C_\Delta, i = 1,2,3\}. \tag{61}$$

Here δ_i are the roots of the closed-loop system (57), (58) characteristic polynomial $\Delta(z,\mathbf{K})$ with the degree $n_d = 3$. Let given desirable area be $C_\Delta = C_{\Delta 2}$, where $r = 0.97$ and the function $\psi(\rho)$ is presented by the formula

$$\psi(\rho) = \begin{cases} ln\left(\frac{r}{\rho}\right) tg\beta, & re^{-\pi/tg\beta} \le \rho \le r, \\ \pi, & if \ 0 < \rho \le re^{-\pi/tg\beta}, \end{cases}$$

where $\beta = \pi/10$. This area is presented on the Fig. 7.

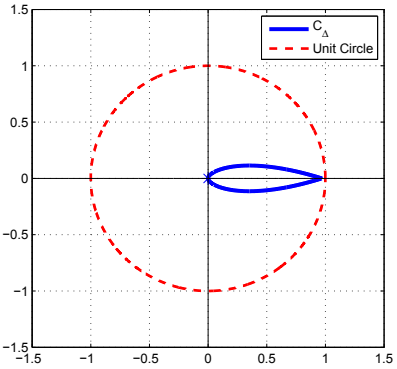


Fig. 7. The area C_Δ of the desired roots location

Let construct now the system of equations in accordance with the identity $\Delta(z,\mathbf{K}) \equiv \Delta^*(z,\gamma)$, where $\gamma \in \mathbf{E}^3$ and the polynomial $\Delta^*(z,\gamma)$ is defined by the formulas (33), (41). As a result, we obtain linear system with respect to unknown parameter vector $\mathbf{K}$

$$\mathbf{L}_0 + \mathbf{L}_1\mathbf{K} = \chi(\gamma). \tag{62}$$

Here vector $\mathbf{L}_0$ and square matrix $\mathbf{L}_1$ are constant for any sampling instant k. These are fully defined by the matrices of the system (57). Besides that, the matrix $\mathbf{L}_1$ is nonsingular, hence we can find the unique solution for system (62)

$$\mathbf{K} = \tilde{\mathbf{L}}_0 + \tilde{\mathbf{L}}_1\chi(\gamma), \tag{63}$$

where $\tilde{\mathbf{L}}_1 = \mathbf{L}_1^{-1}$ and $\tilde{\mathbf{L}}_0 = -\mathbf{L}_1^{-1}\mathbf{L}_0$. Substituting (63) into the prediction model (59) and then into the cost functional (60), we get $J_k = J_k(\mathbf{K}) = J_k^*(\gamma)$. That is the functional J_k becomes

the function of three indepent variables. Then, accordingly to the theorem 3, optimization problem (61) is equivalent to the unconstrained minimization

$$J_k^* = J_k^*(\gamma) \to \min_{\gamma \in \mathbf{E}^3}. \tag{64}$$

Thus, in conformity with the algorithm of the MPC real-time implementation, presented in the section 4 above, in order to form control input we must solve the unconstrained optimization problem (64) at each sampling instant.

Consider now the processes of the plasma vertical stabilization. For the first, let us consider the unconstrained case. Remember that the structure of the controller (58) is linear. So, if the roots of the characteristic polynomial for the system (57) closed by the LQR-controller are located inside the area C_Δ then parameter vector $\mathbf{K}$ will be practically equivalent to the matrix of the LQR-controller. The roots of the system closed by the discrete LQR are the following $z_1 = 0.9591, z_2 = 0.8661, z_3 = 0.9408$. This roots are located inside the area C_Δ. So, the transient responce of the system closed by the MPC-controller, which is based on the optimization (64), is approximately the same as presented in Fig. 5.

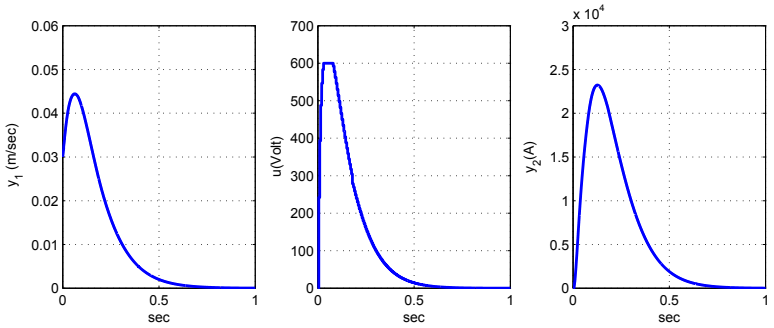


Fig. 8. Transient response of the closed-loop system with constrained MPC-controller

Consider now the processes of plasma stabilization with the constraints (53) imposed. As mentioned above, in order to take into account the constraint imposed on the current, the additional equation should be added. It is necessary to remark that in the presence of the constraints, the optimization problem (64) becomes the nonlinear programming problem. Fig.8 shows transient responce of the closed-loop system with MPC-controller when the only constraint on the VS converter voltage is taked into account. It can be seen from the figure that the constraint imposed on the voltage is satisfied, but the constraint on the current is violated. Fig.9 shows transient responce of the closed-loop system with MPC-controller when both the constraint on the VS converter voltage and current are taken into account. It is not difficult to see that all the imposed constraints are satisfied.

6. Conclusion

The problem of plasma vertical stabilization based on the model predictive control has been considered. It is shown that MPC algorithms are superior compared to the LQR-optimal controller, because they allow taking constraints into account and provide high-performance control. It is also shown that in the case of the traditional MPC-scheme it is possible to reduce

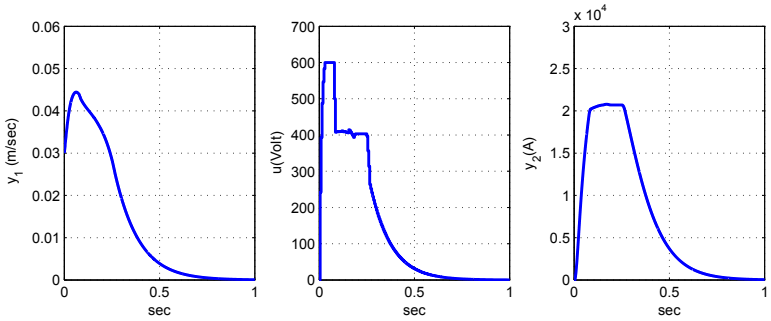


Fig. 9. Transient response of the closed-loop system with constrained MPC-controller

the computational load significantly using relatively small control horizon or by increasing sample interval while preserving the processes quality in the closed-loop system.
New MPC approach was provided. This approach allows us to guarantee linear closed-loop system stability. It's implementation in real-time is connected with the on-line solution of the unconstrained nonlinear optimization problem if there is not constraint imposed and with the nonlinear programming problem in the presence of constraints. The significant feature of this approach is that the dimension of the optimization problem is not depend on the prediction horizon P. The algorithm for the real-time implementation of the suggested approach was described. It allows us to use MPC algorithms to solve plasma vertical stabilization problem.

7. References

Belyakov, V., Zhabko, A., Kavin, A., Kharitonov, V., Misenov, B., Mitrishkin, Y., Ovsyannikov, A. & Veremey, E. (1999). Linear quadratic Gaussian controller design for plasma current, position and shape control system in ITER. *Fusion Engineering and Design*, Vol. 45, No. 1, pp. 55–64.

Camacho E.F. & Bordons C. (1999). *Model Predictive Control*, Springer-Verlag, London.

Gribov, Y., Albanese, R., Ambrosino, G., Ariola, M., Bulmer, R., Cavinato, M., Coccorese, E., Fujieda, H., Kavin A. et. al. (2000). ITER-FEAT scenarios and plasma position/shape control, *Proc. 18th IAEA Fusion Energy Conference*, Sorrento, Italy, 2000, ITERP/02.

Hendricks, E., Jannerup, O. & Sorensen, P.H. (2008) *Linear Systems Control: Deterministic and Stochastic Methods*, Springer-Verlag, Berlin.

Maciejowski, J. M. (2002). *Predictive Control with Constraints*, Prentice Hall.

Misenov, B.A., Ovsyannikov, D.A., Ovsyannikov, A.D., Veremey, E.I. & Zhabko, A.P. (2000). Analysis and synthesis of plasma stabilization systems in tokamaks, *Proc. 11th IFAC Workshop. Control Applications of Optimization*, Vol.1, pp. 255-260, New York.

Morari, M., Garcia, C.E., Lee, J.H. & Prett D.M. (1994). *Model Predictive Control*, Prentice Hall, New York.

Ovsyannikov, D. A., Ovsyannikov, A. D., Zhabko, A. P., Veremey, E. I., Makeev I. V., Belyakov V. A., Kavin A. A. & McArdle G. J. (2005). Robust features analysis for the MAST plasma vertical feedback control system.(2005). *2005 International Conference on Physics and Control, PhysCon 2005, Proceedings, 2005*, pp. 69–74.

Ovsyannikov D. A., Veremey E. I., Zhabko A. P., Ovsyannikov A. D., Makeev I. V., Belyakov V. A., Kavin A. A., Gryaznevich M. P. & McArdle G. J.(2005) Mathematical methods of plasma vertical stabilization in modern tokamaks, in *Nuclear Fusion*, Vol.46, pp. 652-657 (2006).

10

Predictive Control of Tethered Satellite Systems

Paul Williams
Delft University of Technology
Australia

1. Introduction

Tethered satellite systems have many potential applications, ranging from upper atmospheric research (Colombo et al., 1975) to momentum transfer (Nordley & Forward, 2001; Williams et al., 2004). The major dynamical features of the system have been studied extensively (Misra & Modi, 1986), but there still remain open questions with regard to control (Blanksby & Trivailo, 2000). Many of the open issues stem from the fact that there have been limited flight tests. The most recent flight of the Young Engineers' Satellite 2 (YES-2) highlighted from its results that tether dynamic modelling is relatively mature, but that there is a need to provide fault tolerant design in the control and sensor subsystems (Kruijff et al., 2009).

In applications such as momentum transfer and payload capture, it is imperative that robust, accurate and efficient controllers can be designed. For example, although it is conceivable to use onboard thrusters to manipulate the motion of the tethered satellite, this negates some of the advantages of using tethers, i.e., little to no fuel expenditure in ideal circumstances. The main source of control, therefore, has to be sought from manipulating the length of deployed tether. This has two main aims: first, the length of tether directly controls the distance of the tether tip from the main spacecraft, and second, changes in tether length induce Coriolis-type forces on the system due to the orbital motion, which allows indirect control over the swing motion of the tether (librations). Typically, control over the tether length is achieved via manipulating the tension at the mother satellite (Rupp, 1975; Lorenzini et al., 1996). This can help to prevent the tether from becoming slack – a situation that can lead to loss of control of the system.

A variety of different control strategies have been proposed in the literature on tethered systems. Much of the earlier work focused on controlling the deployment and retrieval processes (Xu et al., 1981; Misra & Modi, 1982; Fujii & Anazawa, 1994). This was usually achieved by combining an open-loop length control scheme with feedback of the tether states, either appearing linearly or nonlinearly. Other schemes were devoted to ensuring nonlinear asymptotic stability through the use of Lyapunov's second method (Fujii & Ishijima, 1989; Fujii, 1991; Vadali & Kim, 1991). Most of these techniques do not ensure well-behaved dynamics, and can be hard to tune to make the deployment and retrieval fast.

Because deployment and retrieval is an inherent two-point boundary value problem, it makes much more sense to approach the problem from the point-of-view of optimal control. Several examples of the application of optimal control theory to tethered satellite systems can be found (Fujii & Anazawa, 1994; Barkow, 2003; Lakso & Coverstone, 2000). However, the direct application of the necessary conditions for optimality leads to an extremely numerically sensitive two-point boundary value problem. The state-costate equations are well-known to suffer from instability, but the tethered satellite problem is notorious because of the instability of the state equations to small errors in the control tension. More recent work has focused on the application of direct transcription methods to the tethered satellite problem (Lakso & Coverstone, 2000; Williams, 2008; Williams & Blanksby, 2008). This provides advantages with respect to robustness of convergence and is typically orders of magnitude faster than other methods.

In recent work, the effect of the performance index used in solving the optimal control problem for tethered satellites was examined in detail (Williams, 2008). The work in (Williams, 2008) was prompted by the fact that bang-bang tension control trajectories have been proposed (Barkow, 2003), which is extremely undesirable for controlling a flexible tether. The conclusions reached in (Williams, 2008) suggest that an inelastic tether model can be sufficient to design the open-loop trajectory, provided the cost function is suitably selected. Suitable costs include the square of the tether length acceleration, tension rate or tension acceleration. These trajectories lead to very smooth variations in the dynamics, which ultimately improves the tracking capability of feedback controllers, and reduces the probability of instabilities.

Much of the previous work on optimal control of tethered satellites has focused on obtaining solutions, as opposed to obtaining *rapid* solutions. Some of the ideas that will be explored in this paper have been discussed in (Williams, 2004), which presented two approaches for implementing an optimal-based controller for tethered satellites. One of the methods was based on quasilinearization of the necessary conditions for optimality combined with a pseudospectral discretization, whereas the second was a direct discretization of the continuous optimal control problem. In (Williams, 2004), NPSOL was used as the nonlinear programming (NLP) solver, which implements methods based on dense linear algebra, and is significantly slower than the sparse counterpart SNOPT (Gill et al., 2002).

The aforementioned YES-2 mission had the aim of deploying a 32 km long tether in two phases. The first phase had the objective of stabilizing the tether swinging motion (librations) at the local vertical with the tether length at 3.5 km. The second phase had the objective of inducing a sufficient swinging motion at the end of deployment to allow a specially designed payload to re-enter the atmosphere and be recovered in Khazikstan. The deployment controller consisted of using a reference trajectory computed offline via direct transcription (Williams et al., 2008), in combination with a feedback controller to stabilize the deployment dynamics. The feedback controller used a time-varying feedback gain calculated via a receding horizon approach documented in (Williams, 2005). Flight results showed that despite very large perturbations from nominal, the tether was deployed successfully in the first phase. An issue with one of the sensors that measured the deployment rate caused the feedback controller to believe that the tether was being deployed too slowly. As a consequence, the tether was deployed too quickly. It has been

shown that the tether was nonetheless fully deployed, making it the longest tether ever deployed in space.
The aim of this Chapter is to explore the possibility of providing real-time optimal control for a tethered satellite system. A realistic tether model is combined with a nonlinear Kalman filter for estimating the tether state based on available measurements. A nonlinear model predictive controller is implemented to satisfy the mission requirements.

2. System Model

In order to generate rapid optimal trajectories and test closed-loop performance for a real system, it is necessary to introduce mathematical models of varying fidelity. In this chapter, two models are distinguished: 1) a high fidelity truth model, 2) a low fidelity control model. A truth model is required for testing the closed-loop performance of the controller in a representative environment. Typically, the truth model will incorporate effects that are not present in the model used by the controller. In the simplest case, these can be environmental disturbances. Truth models are usually of higher fidelity than the control model, and as such, they become difficult to use for real-time closed-loop control. For this reason, it is necessary to employ a reduced order model in the controller. It should be pointed out that a truth model will typically include a set of parameter perturbations that alter the characteristics of the simulated system compared to the assumptions made in the control model. Such perturbations are used in Monte Carlo simulations of the closed-loop system to gather statistics on the controller performance.
For the particular case of a tethered satellite system, there are a number of important dynamics that exist in the real system: 1) Rigid-body, librations of the tether in- and out-plane, 2) Lateral string oscillations of the tether between the tether attachment points, 3) Longitudinal spring-mass oscillations of the tether, 4) Rigid body motions of the end bodies, 5) Orbital perturbations caused by exchange of angular momentum from the tethered system with orbital angular momentum. All of these dynamic modes are coupled to varying degrees. However, the dominant dynamics are due to (1) and (2) as these directly impact the short-term response of the system.
The following subsections derive the fundamental equations of motion for modeling the tethered system taking into account the dominant dynamics. A simplified model suitable for model predictive control is then developed.

2.1 Truth Model

The most sophisticated models for tethered satellite systems treat the full effects of tether elasticity and flexibility. Examples include models based on discretization by assumed modes (Xu et al., 1986) or discretization by lumped masses (Kim & Vadali, 1995). In a typical lumped mass model, the tether is discretized into a series of point masses connected by elastic springs. The tension in each element can be computed explicitly based on the positions of the adjacent lumped masses. It is well known that the equations of motion for the system are 'stiff', referring to the fact that the dynamics occur over very different timescales, requiring small integration step sizes to capture the very high frequency modes. For a tethered satellite system, the high frequency modes are the longitudinal elastic modes, followed by the string modes of the tether, libration modes, and finally the orbital motion. For short duration missions or analysis, the longitudinal modes are unlikely to have a

significant effect on the overall motion (provided the tether remains taut). Thus, in this model the effects of longitudinal vibrations are ignored, and the tether is divided into a series of point masses connected via inelastic links. The geometric shortening of the distance to the tether tip is accounted for due to the changes in geometry of the system, but stretching of the tether is not. The degree of approximation is controlled by the number of discretized elements that are used.

The tether is modeled as consisting of a series of n point masses connected via inelastic links, as shown in Fig. 1. The (x, y, z) coordinate system rotates at the orbit angular velocity and is assumed to be attached at the center of mass of the orbit (mother satellite). Although not a necessary assumption in the model, it is assumed that the orbit of the mother satellite is prescribed and remains Keplerian. In general, this coordinate system would orbit in a plane defined by the classical orbital elements (argument of perigee, inclination, longitude of ascending node). In the presence of a Newtonian gravitational field, the orientation of the orbital plane does not affect the system dynamics. However, it does affect any aerodynamic or electrodynamic forces due to the nature of the Earth's rotating atmosphere and magnetic field. These effects are not considered here.

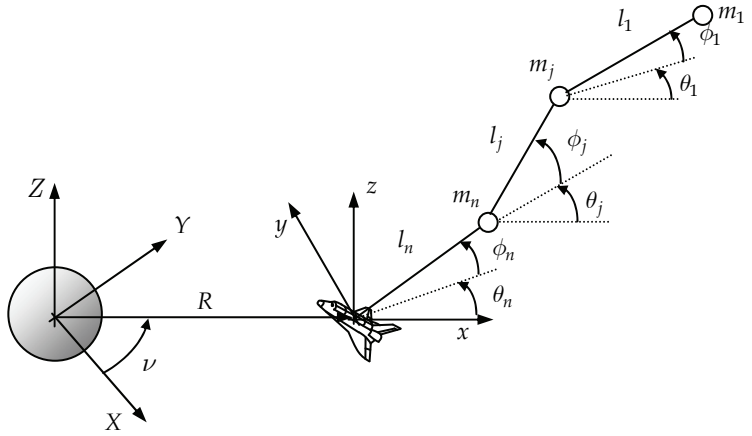


Fig. 1. Discretized multibody tether model.

The acceleration of a mass in the rotating frame is given by

$$\ddot{\boldsymbol{r}} = \left(\ddot{x} - \dot{\omega}y - 2\omega\dot{y} - \omega^2 x\right)\boldsymbol{i} + \left(\ddot{y} + \dot{\omega}x + 2\omega\dot{x} - \omega^2 y\right)\boldsymbol{j} + \ddot{z}\boldsymbol{k} \tag{1}$$

where $\omega = \kappa^2\sqrt{\mu / p^3}$ is the orbital angular velocity, $p = a(1-e^2)$ is the semilatus rectum, μ is the Earth's gravitational parameter, e is the orbit eccentricity, $\kappa = 1 + e\cos\nu$, and a is the orbit semimajor axis. The contribution of forces due to the gravity-gradient is given by

$$\boldsymbol{F}_j^g = 2m_j\frac{\mu x_j}{R^3}\boldsymbol{i} - m_j\frac{\mu y_j}{R^3}\boldsymbol{j} - m_j\frac{\mu z_j}{R^3}\boldsymbol{k} = 2m_j\frac{\omega^2 x_j}{\kappa}\boldsymbol{i} - m_j\frac{\omega^2 y_j}{\kappa}\boldsymbol{j} - m_j\frac{\omega^2 z_j}{\kappa}\boldsymbol{k} \tag{2}$$

Note that in Equation (1), the contributions due to the center of mass motion R and corresponding true anomaly ν are cancelled with the Newtonian gravity terms for the system center of mass. This is valid if the system is assumed to be in a Keplerian orbit.
Define the tension vector in the j th segment as

$$\boldsymbol{T}_j = T_j\left(\cos\theta_j\cos\phi_j\boldsymbol{i} + \sin\theta_j\cos\phi_j\boldsymbol{j} + \sin\phi_j\boldsymbol{k}\right) \tag{3}$$

Also, define the j th mass in terms of the generalized coordinates,

$$x_j = \sum_{k=j}^{n} l_k\cos\theta_k\cos\phi_k \tag{4}$$

$$y_j = \sum_{k=j}^{n} l_k\sin\theta_k\cos\phi_k \tag{5}$$

$$z_j = \sum_{k=j}^{n} l_k\sin\phi_k \tag{6}$$

The tension forces on the j th mass are given by

$$\boldsymbol{F}_j^{\text{tension}} := \begin{cases} \boldsymbol{T}_{j-1} - \boldsymbol{T}_j, & 1 < j \le n \\ -\boldsymbol{T}_j, & j = 1 \end{cases} \tag{7}$$

The equations of motion can be expressed in component form as

$$\begin{aligned} \ddot{x}_j - \dot{\omega}y_j - 2\omega\dot{y}_j - \omega^2 x_j - 2\frac{\omega^2}{\kappa}x_j &= \frac{F_j^x}{m_j} \\ \ddot{y}_j + \dot{\omega}x_j + 2\omega\dot{x}_j - \omega^2 y_j + \frac{\omega^2}{\kappa}y_j &= \frac{F_j^y}{m_j} \\ \ddot{z}_j + \frac{\omega^2}{\kappa}z_j &= \frac{F_j^z}{m_j} \end{aligned} \tag{8}$$

where m_j is the mass of the jth cable mass, and (F_j^x, F_j^y, F_j^z) is the vector of external forces acting on the jth mass in the orbital frame. Substitution of Equations (4) through (6) into Equation (8) gives the governing equations of motion in spherical coordinates. The equations of motion may be decoupled by employing a matrix transformation and forward substitution of the results. By multiplying the vector of Equation (8) by the matrix

$$\left[C_j\right] = \begin{bmatrix} -\sin\theta_j & \cos\theta_j & 0 \\ -\cos\theta_j\sin\phi_j & -\sin\theta_j\sin\phi_j & \cos\phi_j \\ \cos\theta_j\cos\phi_j & \sin\theta_j\cos\phi_j & \sin\phi_j \end{bmatrix} \tag{9}$$

the general decoupled equations of motion can be expressed as

$$\begin{aligned}
\ddot{\theta}_j &= -\dot{\omega} + 2\left(\dot{\theta}_j + \omega\right)\left[\dot{\phi}_j \tan\phi_j - \frac{\dot{l}_j}{l_j}\right] - 3\frac{\omega^2}{\kappa}\sin\theta_j\cos\theta_j - \frac{T_{j-1}\sin\theta_j\cos\theta_{j-1}\cos\phi_{j-1}}{m_j l_j\cos\phi_j} \\
&\quad - \frac{F_j^x\sin\theta_j}{m_j l_j\cos\phi_j} + \frac{T_{j-1}\cos\theta_j\sin\theta_{j-1}\cos\phi_{j-1}}{m_j l_j\cos\phi_j} + \frac{T_{j+1}\cos\theta_j\sin\theta_{j+1}\cos\phi_{j+1}}{m_{j+1} l_j\cos\phi_j} \\
&\quad - \frac{T_{j+1}\sin\theta_j\cos\theta_{j+1}\cos\phi_{j+1}}{m_{j+1} l_j\cos\phi_j} + \frac{F_j^y\cos\theta_j}{m_j l_j\cos\phi_j} + \frac{F_{j+1}^x\sin\theta_j}{m_{j+1} l_j\cos\phi_j} - \frac{F_{j+1}^y\cos\theta_j}{m_{j+1} l_j\cos\phi_j}
\end{aligned} \tag{10}$$

$$\begin{aligned}
\ddot{\phi}_j &= -2\frac{\dot{l}_j}{l_j}\dot{\phi}_j - \left[\left(\dot{\theta}_j + \omega\right)^2 + \frac{3\omega^2\cos^2\theta_j}{\kappa}\right]\sin\phi_j\cos\phi_j \\
&\quad + \frac{T_{j-1}}{m_j l_j}\left(\cos\phi_j\sin\phi_{j-1} - \cos\theta_j\sin\phi_j\cos\theta_{j-1}\cos\phi_{j-1} - \sin\theta_j\sin\phi_j\sin\theta_{j-1}\cos\phi_{j-1}\right) \\
&\quad + \frac{T_{j+1}}{m_{j+1} l_j}\left(\cos\phi_j\sin\phi_{j+1} - \cos\theta_j\sin\phi_j\cos\theta_{j+1}\cos\phi_{j+1} - \sin\theta_j\sin\phi_j\sin\theta_{j+1}\cos\phi_{j+1}\right) \\
&\quad + \frac{F_j^z\cos\phi_j}{m_j l_j} - \frac{F_{j+1}^z\cos\phi_j}{m_{j+1} l_j} + \frac{F_{j+1}^y\sin\theta_j\sin\phi_j}{m_{j+1} l_j} - \frac{F_j^y\sin\theta_j\sin\phi_j}{m_j l_j} - \frac{F_j^x\cos\theta_j\sin\phi_j}{m_j l_j} \\
&\quad + \frac{F_{j+1}^x\cos\theta_j\sin\phi_j}{m_{j+1} l_j}
\end{aligned} \tag{11}$$

$$\begin{aligned}
\ddot{l}_j &= l_j\left[\left(\dot{\theta}_j + \omega\right)^2\cos^2\phi_j + \dot{\phi}_j^2 - \frac{\omega^2}{\kappa}\left(1 - 3\cos^2\theta_j\cos^2\phi_j\right)\right] - \frac{T_j}{m_j} - \frac{T_j}{m_{j+1}} \\
&\quad + \frac{F_j^z\sin\phi_j}{m_j} - \frac{F_{j+1}^z\sin\phi_j}{m_{j+1}} + \frac{F_j^y\sin\theta_j\cos\phi_j}{m_j} - \frac{F_{j+1}^y\sin\theta_j\cos\phi_j}{m_{j+1}} \\
&\quad + \frac{F_j^x\cos\theta_j\cos\phi_j}{m_j} - \frac{F_{j+1}^x\cos\theta_j\cos\phi_j}{m_{j+1}} \\
&\quad \frac{T_{j-1}}{m_j}\left[\cos\theta_j\cos\phi_j\cos\theta_{j-1}\cos\phi_{j-1} + \sin\phi_j\sin\phi_{j-1} + \sin\theta_j\cos\phi_j\sin\theta_{j-1}\cos\phi_{j-1}\right] \\
&\quad \frac{T_{j+1}}{m_{j+1}}\left[\cos\theta_j\cos\phi_j\cos\theta_{j+1}\cos\phi_{j+1} + \sin\phi_j\sin\phi_{j+1} + \sin\theta_j\cos\phi_j\sin\theta_{j+1}\cos\phi_{j+1}\right]
\end{aligned} \tag{12}$$

These equations may be nondimensionalized by utilizing the following relationships

$$\frac{\mathrm{d}}{\mathrm{d}t} = \omega\frac{\mathrm{d}}{\mathrm{d}\nu}, \qquad \frac{\mathrm{d}^2}{\mathrm{d}t^2} = \dot{\omega}\frac{\mathrm{d}}{\mathrm{d}\nu} + \omega^2\frac{\mathrm{d}^2}{\mathrm{d}\nu^2} \tag{13}$$

$$l_j = \Lambda_j L,\ u_j = \frac{T_j}{m_j\omega^2 L} \tag{14}$$

$$\dot{\omega} = -2\frac{\omega^2}{\kappa}e\sin\nu \tag{15}$$

Thus, the following nondimensional equations of motion are obtained

$$\theta_j'' = 2\left(\theta_j'+1\right)\left[\frac{e\sin\nu}{\kappa}+\phi_j'\tan\phi_j-\frac{\Lambda_j'}{\Lambda_j}\right]-\frac{3}{\kappa}\sin\theta_j\cos\theta_j+\frac{u_{j+1}\cos\theta_j\sin\theta_{j+1}\cos\phi_{j+1}}{\Lambda_j\cos\phi_j}$$
$$+\frac{m_{j-1}}{m_j}\frac{u_{j-1}\cos\theta_j\sin\theta_{j-1}\cos\phi_{j-1}}{\Lambda_j\cos\phi_j}-\frac{m_{j-1}}{m_j}\frac{u_{j-1}\sin\theta_j\cos\theta_{j-1}\cos\phi_{j-1}}{\Lambda_j\cos\phi_j}$$
$$-\frac{u_{j+1}\sin\theta_j\cos\theta_{j+1}\cos\phi_{j+1}}{\Lambda_j\cos\phi_j}+\frac{F_j^y\cos\theta_j}{m_j\omega^2L\Lambda_j\cos\phi_j}-\frac{F_j^x\sin\theta_j}{m_j\omega^2L\Lambda_j\cos\phi_j}+\frac{F_{j+1}^x\sin\theta_j}{m_{j+1}\omega^2L\Lambda_j\cos\phi_j} \tag{16}$$
$$-\frac{F_{j+1}^y\cos\theta_j}{m_{j+1}\omega^2L\Lambda_j\cos\phi_j}$$

$$\phi_j'' = 2\frac{e\sin\nu}{\kappa}\phi'-2\frac{\Lambda_j'}{\Lambda_j}\phi_j'-\left[\left(\theta_j'+1\right)^2+\frac{3\cos^2\theta_j}{\kappa}\right]\sin\phi_j\cos\phi_j+\frac{m_{j-1}}{m_j}\frac{u_{j-1}}{\Lambda_j}(\cos\phi_j\sin\phi_{j-1}$$
$$-\cos\theta_j\sin\phi_j\cos\theta_{j-1}\cos\phi_{j-1}-\sin\theta_j\sin\phi_j\sin\theta_{j-1}\cos\phi_{j-1})+\frac{u_{j+1}}{\Lambda_j}(\cos\phi_j\sin\phi_{j+1}$$
$$-\cos\theta_j\sin\phi_j\cos\theta_{j+1}\cos\phi_{j+1}-\sin\theta_j\sin\phi_j\sin\theta_{j+1}\cos\phi_{j+1})+\frac{F_j^z\cos\phi_j}{m_j\omega^2L\Lambda_j} \tag{17}$$
$$-\frac{F_{j+1}^z\cos\phi_j}{m_{j+1}\omega^2L\Lambda_j}+\frac{F_{j+1}^y\sin\theta_j\sin\phi_j}{m_{j+1}\omega^2L\Lambda_j}-\frac{F_j^y\sin\theta_j\sin\phi_j}{m_j\omega^2L\Lambda_j}-\frac{F_j^x\cos\theta_j\sin\phi_j}{m_j\omega^2L\Lambda_j}+\frac{F_{j+1}^x\cos\theta_j\sin\phi_j}{m_{j+1}\omega^2L\Lambda_j}$$

$$\Lambda_j'' = 2\frac{e\sin\nu}{\kappa}\Lambda_j'+\Lambda_j\left[\left(\theta_j'+1\right)^2\cos^2\phi_j+\phi_j'^2-\frac{1}{\kappa}\left(1-3\cos^2\theta_j\cos^2\phi_j\right)\right]-u_j-\frac{m_j}{m_{j+1}}u_j$$
$$+\frac{F_j^z\sin\phi_j}{m_j\omega^2L}-\frac{F_{j+1}^z\sin\phi_j}{m_{j+1}\omega^2L}+\frac{F_j^y\sin\theta_j\cos\phi_j}{m_j\omega^2L}-\frac{F_{j+1}^y\sin\theta_j\cos\phi_j}{m_{j+1}\omega^2L}$$
$$+\frac{F_j^x\cos\theta_j\cos\phi_j}{m_j\omega^2L}-\frac{F_{j+1}^x\cos\theta_j\cos\phi_j}{m_{j+1}\omega^2L}+\frac{m_{j-1}}{m_j}u_{j-1}(\cos\theta_j\cos\phi_j\cos\theta_{j-1}\cos\phi_{j-1} \tag{18}$$
$$+\sin\phi_j\sin\phi_{j-1}+\sin\theta_j\cos\phi_j\sin\theta_{j-1}\cos\phi_{j-1})+u_{j+1}(\cos\theta_j\cos\phi_j\cos\theta_{j+1}\cos\phi_{j+1}$$
$$+\sin\phi_j\sin\phi_{j+1}+\sin\theta_j\cos\phi_j\sin\theta_{j+1}\cos\phi_{j+1})$$

Equations (16) through (18) utilize the orbit true anomaly ν as independent variable, and L is a scaling length representing the length of each tether element when fully deployed. The applicable boundary conditions are

$$m_0 = 0, \qquad u_0 = 0, \qquad m_{n+1} = \infty, \qquad u_{n+1} = 0 \tag{19}$$

The equations (16) through (18) define the dynamics of the tethered satellite system using spherical coordinates. These are not as general as Cartesian coordinates due to the singularity introduced when $\phi_j = -\frac{\pi}{2}, \frac{\pi}{2}$. This represents very large out of plane librational motion or very large out of plane lateral motion. Although this is a limitation of the model, such situations need to be avoided for most practical missions.

2.1.1 Variable Length Case

The tether is modeled as a collection of lumped masses connected by inelastic links, which makes dealing with the case of a variable length tether more difficult than if the tether was modeled as a single link. In particular, it is necessary to have a state vector of variable dimension and to add and subtract elements from the model at appropriate times. When the tether is treated as elastic, great care needs to be exercised to ensure that the introduction of new elements does not create unnecessary cable oscillations. This can happen if the position of the new mass results in the incorrect tension in the new element. However, for an inelastic tether, the introduction of a new mass occurs such that it is placed along the same line as the existing element. Thus, the new initial conditions for the incoming element are that it has the same angles and angle rates as the existing element (closest to the deployer). Alternative formulations based on the variation principle of Hamilton-Ostrogradksi and which transform the deployed length to a fixed interval by means of a new spatial coordinate have also been used (Wiedermann et al., 1999). However, this was not considered in this work.

If the critical length for introduction of a new element is defined as $\Lambda^* \triangleq 1 + k^*$, then the new element is initialized with a length of k^* in nondimensional units, and the same length rate as the previous nth element. During retrieval, elements must be removed. Here, the nth element to be removed and the (n-1)th element need to be used to update the initial conditions for the new n^* th element. In this work, the position and velocity of the (n-1)th mass is used to initialize the n^* th element. Thus, let

$$\begin{aligned} x_{n-1} &= \Lambda_n \cos\theta_n \cos\phi_n + \Lambda_{n-1} \cos\theta_{n-1} \cos\phi_{n-1} \\ y_{n-1} &= \Lambda_n \sin\theta_n \cos\phi_n + \Lambda_{n-1} \sin\theta_{n-1} \cos\phi_{n-1} \\ z_{n-1} &= \Lambda_n \sin\phi_n + \Lambda_{n-1} \sin\phi_{n-1} \end{aligned} \tag{20}$$

From which

$$\Lambda_{n^*} = \sqrt{x_{n-1}^2 + y_{n-1}^2 + z_{n-1}^2},\ \theta_{n^*} = \text{atan2}(y_{n-1}, x_{n-1}),\ \phi_{n^*} = \sin^{-1}(z_{n-1} \,/\, \Lambda_{n^*}) \tag{21}$$

where atan2 represents the four quadrant inverse tangent where the usual arctangent is defined by $\tan^{-1}(\frac{y_{n-1}}{x_{n-1}})$. Similarly, the relative velocity of the (n-1)th mass in the rotating frame is given by

$$\begin{aligned} x'_{n-1} &= \Lambda'_n \cos\theta_n \cos\phi_n - \Lambda_n \theta'_n \sin\theta_n \cos\phi_n - \Lambda_n \phi'_n \cos\theta_n \sin\phi_n \\ &\quad + \Lambda'_{n-1} \cos\theta_{n-1} \cos\phi_{n-1} - \Lambda_{n-1} \theta'_{n-1} \sin\theta_{n-1} \cos\phi_{n-1} - \Lambda_{n-1} \phi'_{n-1} \cos\theta_{n-1} \sin\phi_{n-1} \\ y'_{n-1} &= \Lambda'_n \sin\theta_n \cos\phi_n + \Lambda_n \theta'_n \cos\theta_n \cos\phi_n - \Lambda_n \phi'_n \sin\theta_n \sin\phi_n \\ &\quad + \Lambda'_{n-1} \sin\theta_{n-1} \cos\phi_{n-1} + \Lambda_{n-1} \theta'_{n-1} \cos\theta_{n-1} \cos\phi_{n-1} - \Lambda_{n-1} \phi'_{n-1} \sin\theta_{n-1} \sin\phi_{n-1} \\ z'_{n-1} &= \Lambda'_n \sin\phi_n + \Lambda_n \phi'_n \cos\phi_n + \Lambda'_{n-1} \sin\phi_{n-1} + \Lambda_{n-1} \phi'_{n-1} \cos\phi_{n-1} \end{aligned} \tag{22}$$

From which

$$\begin{aligned} \Lambda'_{n^*} &= x'_{n-1} \cos\phi_{n^*} \cos\theta_{n^*} + y'_{n-1} \cos\phi_{n^*} \sin\theta_{n^*} + z'_{n-1} \sin\phi_{n^*} \\ \theta'_{n^*} &= \left(y'_{n-1} \cos\theta_{n^*} - x'_{n-1} \sin\theta_{n^*} \right) / \left(\Lambda_{n^*} \cos\phi_{n^*} \right) \\ \phi'_{n^*} &= \left(z'_{n-1} \cos\phi_{n^*} - x'_{n-1} \sin\phi_{n^*} \cos\theta_{n^*} - y'_{n-1} \sin\phi_{n^*} \sin\theta_{n^*} \right) / \Lambda_{n^*} \end{aligned} \tag{23}$$

It should be noted that these updates keep the position and velocity of the (*n*-1)th mass the same across the update. The reason for this is that the positions and velocities of all subsequent masses depend on the position/velocity of the *n*th mass. Hence, if this is changed, then the position and velocity of all masses representing the tether change instantaneously. The accuracy of the updates depend on the transition parameter k^{**}, which is used to monitor the length of the *n*th segment. An element is removed when $\Lambda_n < k^{**}$. Because the tether is inelastic, altering the length of the new *n*th element does not keep the total tether length or mass constant unless the *n*th and (*n*-1)th elements are tangential. Therefore, by choosing k^{**} small enough, the errors in the approximation can be made small.

For control purposes, it is assumed that the rate of change of reel-rate is controlled. Thus, Λ_n'' is specified or determined through a control law. This means that the *n*th element is allowed to vary in length, but all other segments remained fixed in length. The problem is to then solve for the unknown tension constraints that enforce constant total length of the remaining segments, as well as the acceleration of the *n*th segment. Once these are known, they are back-substituted into Equations (16) and (17), as well as Equation (18) for the *n*th element. The equations formed by the set (18) are linear in the tensions u_j, and can thus be solved using standard techniques. This assumes that the segment lengths, length rates, and length accelerations are specified. In this work, LAPACK is utilized in solving the simultaneous equations.

2.1.2 Fixed Length Case

To simulate the case of a fixed length tether, Equations (18) are set to zero for $j = 1,...,n$, allowing the unknown tensions $u_j, j = 1,...,n$ to be determined. The resulting tensions are substituted back into the librational dynamics to determine the evolution of the system dynamics.

2.2 Control Model

The predominant modeling assumption that is used in the literature insofar as control of tethered satellite systems is concerned is that the system can be modeled with three degrees of freedom (Williams, 2008). In other words, when dealing with the librational motion of the system, it is sufficient to model it using spherical coordinates representing the dynamics of the subsatellite. This effectively treats the tether as a straight body, which can either be modeled as an inelastic or elastic rod. Early work has neglected the tether mass since its contribution to the librational motion can be considered relatively small (Fujii & Anazawa, 1994). This is due to the fact that the tether is axisymmetric. When large changes in length are considered, the effect of tether mass becomes more important. Moreover, it is essential to include the effects of tether mass when designing tension control laws because there is a nonlinear relationship between tension and tether mass. However, when performing preliminary analyses, it is sufficient to ignore such effects and compensate for these later in the design.

Although the assumption of treating the tether as a straight rod is often a good one, it can create some problems in practice. For example, all tether string vibrations are neglected,

which play a very important role in electrodynamic systems or systems subjected to long-term perturbations. Furthermore, large changes in deployment velocity can induce significant distortions to the tether shape, which ultimately affects the accuracy of the deployment control laws. Earlier work focused much attention on the dynamics of tethers during length changes, particularly retrieval (Misra & Modi, 1986). In the earlier work, assumed modes was typically the method of choice (Misra & Modi, 1982). However, where optimal control methods are employed, high frequency dynamics can be difficult to handle even with modern methods. For this reason, most optimal deployment/retrieval schemes consider the tether as inelastic.

2.1 Straight, Inelastic Tether Model

In this model, the tether is assumed to be straight and inextensible, uniform in mass, the end masses are assumed to be point masses, and the tether is deployed from one end mass only. The generalized coordinates are selected as the tether in-plane libration angle, θ, the out-of-plane tether libration angle, ϕ, and the tether length, l.

The radius vector to the center of mass may be written in inertial coordinates as

$$\boldsymbol{R} = R\cos\nu\boldsymbol{i} + R\sin\nu\boldsymbol{j} \tag{24}$$

From which the kinetic energy due to translation of the center of mass is derived as

$$T_{\mathrm{t}} = \tfrac{1}{2}m\left(\dot{R}^2 + R^2\dot{\nu}^2\right) \tag{25}$$

where $m = m_1 + m_t + m_2$ is the total system mass, $m_1 = m_1^0 - m_t$ is the mass of the mother satellite, m_t is the tether mass, m_2 is the subsatellite mass, and m_1^0 is the mass of the mother satellite prior to deployment of the tether.

The rotational kinetic energy is determined via

$$T_{\mathrm{r}} = \tfrac{1}{2}\boldsymbol{\omega}^T[I]\boldsymbol{\omega} \tag{26}$$

where $\boldsymbol{\omega}$ is the inertial angular velocity of the tether in the tether body frame

$$\boldsymbol{\omega} = \left(\dot{\nu}\sin\phi + \dot{\theta}\sin\phi\right)\boldsymbol{i} - \left(\dot{\phi}\right)\boldsymbol{j} + \left(\dot{\nu}\cos\phi + \dot{\theta}\cos\phi\right)\boldsymbol{k} \tag{27}$$

Thus we have that

$$T_{\mathrm{r}} = \tfrac{1}{2}m^*l^2[\dot{\phi}^2 + \left(\dot{\nu} + \dot{\theta}\right)^2\cos^2\phi] \tag{28}$$

and $m^* = \left(m_1 + \frac{m_t}{2}\right)\left(m_2 + \frac{m_t}{2}\right) / m - m_t / 6$ is the system reduced mass. The kinetic energy due to deployment is obtained as

$$T_{\mathrm{e}} = \tfrac{1}{2}\frac{m_1(m_2+m_t)}{m}\dot{l}^2 \tag{29}$$

which accounts for the fact that the tether is modeled as stationary inside the deployer and is accelerated to the deployment velocity after exiting the deployer. This introduces a thrust-like term into the equations of motion, which affects the value of the tether tension.
The system gravitational potential energy is (assuming a second order gravity-gradient expansion)

$$V = -\frac{\mu m}{R} + \frac{\mu m^* l^2}{2R^3}\left(1-3\cos^2\theta\cos^2\phi\right) \tag{30}$$

The Lagrangian may be formed as

$$\begin{aligned} L = {} & \tfrac{1}{2}m\left(\dot{R}^2 + R^2\dot{\nu}^2\right) + \tfrac{1}{2}m^* l^2[\dot{\phi}^2 + \left(\dot{\nu}+\dot{\theta}\right)^2\cos^2\phi] \\ & + \tfrac{1}{2}\frac{m_1(m_2+m_t)}{m}\dot{l}^2 + \frac{\mu m}{R} - \frac{\mu m^* l^2}{2R^3}\left(1-3\cos^2\theta\cos^2\phi\right) \end{aligned} \tag{31}$$

Under the assumption of a Keplerian reference orbit for the center of mass, the nondimensional equations of motion can be written as

$$\theta'' = 2(\theta'+1)\left[\frac{e\sin\nu}{\kappa} + \phi'\tan\phi - \frac{m_1\left(m_2+\frac{m_t}{2}\right)}{mm^*}\frac{\Lambda'}{\Lambda}\right] - \frac{3}{\kappa}\sin\theta\cos\theta + \frac{Q_\theta}{m^*\Lambda^2 L_r^2\dot{\nu}^2\cos^2\phi} \tag{32}$$

$$\phi'' = \frac{2e\sin\nu}{\kappa}\phi' - 2\frac{m_1\left(m_2+\frac{m_t}{2}\right)}{mm^*}\frac{\Lambda'}{\Lambda}\phi' - \left[(\theta'+1)^2 + \frac{3}{\kappa}\cos^2\theta\right]\sin\phi\cos\phi + \frac{Q_\phi}{m^*\Lambda^2 L_r^2\dot{\nu}^2} \tag{33}$$

$$\begin{aligned} \Lambda'' = {} & \frac{2e\sin\nu}{\kappa}\Lambda' - \frac{(2m_1-m)\frac{m_t}{2}}{m_1(m_2+m_t)}\frac{\Lambda'^2}{\Lambda} + \left(\frac{m_2+\frac{m_t}{2}}{m_2+m_t}\right)\Lambda[\phi'^2 + (\theta'+1)^2\cos^2\phi \\ & + \frac{1}{\kappa}\left(3\cos^2\theta\cos^2\phi - 1\right)] - \frac{T}{m_1\dot{\nu}^2 L_r(m_2+m_t)/m} \end{aligned} \tag{34}$$

where $\Lambda = l/L_r$ is the nondimensional tether length, L_r is a reference tether length, T is the tether tension, and $()' = \mathrm{d}()/\mathrm{d}\nu$. The generalized forces Q_θ and Q_ϕ are due to distributed forces along the tether, which are typically assumed to be negligible.

3. Sensor models

The full dynamic state of the tether is not directly measurable. Furthermore, the presence of measurement noise means that some kind of filtering is usually necessary before directly using measurements from the sensors in the feedback controller. The following measurements are assumed to be available: 1) Tension force at the deployer, 2) Deployment rate, 3) GPS position of the subsatellite. Models of each of these are developed in the subsections below.

3.1 Tension Model

The tension force measured at the deployer differs from the force predicted by the control model due to the presence of tether oscillations and sensor noise. The magnitude and direction of the force in the tether is obtained from the multibody tether model. The tension force in the orbital frame is given by

$$\begin{aligned} T_x &= u_n m_n \omega^2 L \cos\theta_n \cos\phi_n + w_{T_x} \\ T_y &= u_n m_n \omega^2 L \sin\theta_n \cos\phi_n + w_{T_y} \\ T_z &= u_n m_n \omega^2 L \sin\phi_n + w_{T_z} \end{aligned} \tag{35}$$

where the w terms are zero mean, Gaussian measurement noise with covariance R_T.

3.2 Reel-Rate Model

In general, the length of the deployed tether can be measured quite accurately. In this chapter, the reel-rate is measured at the deployer according to

$$\dot{L} = \Lambda_n' \omega L_n + w_{\dot{L}} \tag{36}$$

where $w_{\dot{L}}$ is a zero mean, Gaussian measurement noise with covariance $R_{\dot{L}}$.

3.3 GPS Model

GPS measurements of the two end bodies significantly improve the estimation performance of the system. The position of the mother satellite is required to form the origin of the orbital coordinate system (in case of non-Keplerian motion), and the position of the subsatellite allows observations of the subsatellite range and relative position (libration state). Only position information is used in the estimator. The processed relative position is modeled in the sensor model, as opposed to modeling the satellite constellation and pseudoranges. The processed position error is modeled as a random walk process

$$\delta\dot{x} = \frac{w_x}{\tau_{GPS}},\ \delta\dot{y} = \frac{w_y}{\tau_{GPS}},\ \delta\dot{z} = \frac{w_z}{\tau_{GPS}} \tag{37}$$

where $w_{x,y,z}$ are zero mean white noise processes with covariance R_{GPS}, and τ_{GPS} is a time constant. This model takes into account that the GPS measurement errors are in fact time-correlated.

4. State Estimation

In order to estimate the full tether state, it is necessary to combine all of the measurements obtained from the sensors described in Section 3. The most optimal way to combine the measurements is by applying a Kalman filter. Various forms of the Kalman filter are available for nonlinear state estimation problems. The two most commonly used filter implementations are the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF). The UKF is more robust to filter divergence because it captures the propagation of

uncertainty in the filter states to a higher order than the EKF, which only captures the propagation to first order. The biggest drawback of the UKF is that it is significantly more expensive than the EKF. Consider a state vector of dimension n_x. The EKF only requires the propagation of the mean state estimate through the nonlinear model, and three matrix multiplications of the size of the state vector ($n_x \times n_x$). The UKF requires the propagation of $2n_x + 1$ state vectors through the nonlinear model, and the sum of vector outer products to obtain the state covariance matrix. The added expense can be prohibitive for embedded real-time systems with small sampling times (i.e., on the order of milliseconds). For the tethered satellite problem, the timescales of the dynamics are long compared to the available execution time. Hence, higher-order nonlinear filters can be used to increase performance of the estimation without loss of real-time capability.
Recently, an alternative to the UKF was introduced that employs a spherical-radial-cubature rule for numerically integrating the moment integrals needed for nonlinear estimation. The filter has been called the Cubature Kalman Filter (CKF). This filter is used in this chapter to perform the nonlinear state estimation.

4.1 Cubature Kalman Filter

In this section, the CKF main steps are summarized. The justification for the methodology is omitted and may be found in (Guess & Haykin, 2009).
The CKF assumes a discrete time process model of the form

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{v}_k, t_k) \tag{38}$$

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k, t_k) \tag{39}$$

where $\boldsymbol{x}_k \in \mathbb{R}^{n_x}$ is the system state vector, $\boldsymbol{u}_k \in \mathbb{R}^{n_u}$ is the system control input, $\boldsymbol{y}_k \in \mathbb{R}^{n_y}$ is the system measurement vector, $\boldsymbol{v}_k \in \mathbb{R}^{n_v}$ is the vector of process noise, assumed to be white Gaussian with zero mean and covariance $\boldsymbol{Q}_k \in \mathbb{R}^{n_v \times n_v}$, $\boldsymbol{w}_k \in \mathbb{R}^{n_w}$ is a vector of measurement noise, assumed to be white Gaussian with zero mean and covariance $\boldsymbol{R}_k \in \mathbb{R}^{n_w \times n_w}$. For the results in this paper, the continuous system is converted to a discrete system by means of a fourth-order Runge-Kutta method.
In the following, the process and measurement noise is implicitly augmented with the state vector as follows

$$\boldsymbol{x}_k^a = \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{v}_k \\ \boldsymbol{w}_k \end{bmatrix} \tag{40}$$

The first step in the filtering process is to compute the set of cubature points as follows

$$\mathcal{X}_{k-1} = \left[\hat{\boldsymbol{x}}_{k-1}^a + \boldsymbol{I}_{n_a \times n_a} \sqrt{\boldsymbol{P}_k}, \hat{\boldsymbol{x}}_{k-1}^a - \boldsymbol{I}_{n_a \times n_a} \sqrt{\boldsymbol{P}_k} \right] \tag{41}$$

where $\hat{\boldsymbol{x}}^a$ is the mean estimate of the augmented state vector, and $\boldsymbol{P}_k$ is the covariance matrix. The cubature points are then propagated through the nonlinear dynamics as follows

$$\mathcal{X}^*_{k|k-1} = f(\mathcal{X}_{k-1}, \boldsymbol{u}_k, t_k) \tag{42}$$

The predicted mean for the state estimate is calculated from

$$\hat{\boldsymbol{x}}_k^- = \frac{1}{2n_a}\sum_{i=0}^{2n_a}\mathcal{X}^*_{i,k|k-1} \tag{43}$$

The covariance matrix is predicted by

$$\boldsymbol{P}_k^- = \frac{1}{2n_a}\sum_{i=0}^{2n_a}\mathcal{X}^*_{i,k|k-1}\mathcal{X}^{*T}_{i,k|k-1} - \hat{\boldsymbol{x}}_k^-\hat{\boldsymbol{x}}_k^{-T} \tag{44}$$

When a measurement is available, the augmented sigma points are propagated through the measurement equations

$$\mathcal{Y}_{k|k-1} = \boldsymbol{h}(\mathcal{X}_{k|k-1}, \boldsymbol{u}_k, t_k) \tag{45}$$

The mean predicted observation is obtained by

$$\hat{\boldsymbol{y}}_k^- = \frac{1}{2n_a}\sum_{i=0}^{2n_a}\mathcal{Y}_{i,k|k-1} \tag{46}$$

The innovation covariance is calculated using

$$\boldsymbol{P}_k^{yy} = \frac{1}{2n_a}\sum_{i=0}^{2n_a}\mathcal{Y}_{i,k|k-1}\mathcal{Y}^T_{i,k|k-1} - \hat{\boldsymbol{y}}_k^-\hat{\boldsymbol{y}}_k^{-T} \tag{47}$$

The cross-correlation matrix is determined from

$$\boldsymbol{P}_k^{xy} = \frac{1}{2n_a}\sum_{i=0}^{2n_a}\mathcal{X}_{i,k|k-1}\mathcal{Y}^T_{i,k|k-1} - \hat{\boldsymbol{x}}_k^-\hat{\boldsymbol{y}}_k^{-T} \tag{48}$$

The gain for the Kalman update equations is computed from

$$\mathcal{K}_k = \boldsymbol{P}_k^{xy}(\boldsymbol{P}_k^{yy})^{-1} \tag{49}$$

The state estimate is updated with a measurement of the system $\boldsymbol{y}_k$ using

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \mathcal{K}_k\left(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^-\right) \tag{50}$$

and the covariance is updated using

$$\boldsymbol{P}_k^+ = \boldsymbol{P}_k^- - \mathcal{K}_k \boldsymbol{P}_k^{yy} \mathcal{K}_k^T \tag{51}$$

It is often necessary to provide numerical remedies for covariance matrices that do not maintain positive definiteness. Such measures are not discussed here.

5. Optimal Trajectory Generation

Most of the model predictive control strategies that have been suggested in the literature are based on low-order discretizations of the system dynamics, such as Euler integration. Dunbar et al. (2002) applied receding horizon control to the Caltech Ducted Fan based on a B-spline parameterization of the trajectories. In recent years, pseudospectral methods, and in particular the Legendre pseudospectral (PS) method (Elnagar, 1995; Ross & Fahroo, 2003), have been used for real-time generation of optimal trajectories for many systems. The traditional PS approach discretizes the dynamics via differentiation operators applied to expansions of the states in terms of Lagrange polynomial bases. Another approach is to discretize the dynamics via Gauss-Lobatto quadratures. The approach has been more fully described by Williams (2006). The latter approach is used here.

5.1 Discretization approach

Instead of presenting a general approach to solving optimal control problems, the Gauss-Lobatto approach presented in this section is restricted to the form of the problem solved here. The goal is to find the state and control history $\{\boldsymbol{x}(t), \boldsymbol{u}(t)\}$ to minimize the cost function

$$\mathcal{J} = \mathcal{M}\left[\boldsymbol{x}(t_f)\right] + \int_{t_0}^{t_f} \mathcal{L}\left[\boldsymbol{x}(t^*), \boldsymbol{u}(t^*), t^*\right] \mathrm{d}t^* \tag{52}$$

subject to the nonlinear state equations

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left[\boldsymbol{x}(t), \boldsymbol{u}(t), t\right] \tag{53}$$

the initial and terminal constraints

$$\boldsymbol{\psi}_0\left[\boldsymbol{x}(t_0)\right] = \boldsymbol{0} \tag{54}$$

$$\boldsymbol{\psi}_f\left[\boldsymbol{x}(t_f)\right] = \boldsymbol{0} \tag{55}$$

the mixed state-control path constraints

$$\boldsymbol{g}_L \leq \boldsymbol{g}\left[\boldsymbol{x}(t), \boldsymbol{u}(t), t\right] \leq \boldsymbol{g}_U \tag{56}$$

and the box constraints

$$\boldsymbol{x}_L \leq \boldsymbol{x}(t) \leq \boldsymbol{x}_U, \qquad \boldsymbol{u}_L \leq \boldsymbol{u}(t) \leq \boldsymbol{u}_U \tag{57}$$

where $\boldsymbol{x} \in \mathbb{R}^{n_x}$ are the state variables, $\boldsymbol{u} \in \mathbb{R}^{n_u}$ are the control inputs, $t \in \mathbb{R}$ is the time, $\mathcal{M} : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}$ is the Mayer component of cost function, i.e., the terminal, non-integral cost in Eq. (52), $\mathcal{L} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}$ is the Bolza component of the cost function, i.e., the integral cost in Eq. (52), $\boldsymbol{\psi}_0 \in \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_0}$ are the initial point conditions, $\boldsymbol{\psi}_f \in \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_f}$ are the final point conditions, and $\boldsymbol{g}_L \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}^{n_g}$ and $\boldsymbol{g}_U \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}^{n_g}$ are the lower and upper bounds on the path constraints.

The basic idea behind the Gauss-Lobatto quadrature discretization is to approximate the vector field by an N th degree Lagrange interpolating polynomial

$$\boldsymbol{f}(t) \approx \boldsymbol{f}_N(t) \tag{58}$$

expanded using values of the vector field at the set of Legendre-Gauss-Lobatto (LGL) points. The LGL points are defined on the interval $\tau \in [-1,1]$ and correspond to the zeros of the derivative of the N th degree Legendre polynomial, $L_N(\tau)$, as well as the end points –1 and 1. The computation time is related to the time domain by the transformation

$$t = \frac{(t_f - t_0)}{2}\tau + \frac{(t_f + t_0)}{2} \tag{59}$$

The Lagrange interpolating polynomials are written as

$$\boldsymbol{f}_N(t) = \sum_{k=0}^{N} \boldsymbol{f}_k \phi_k(\tau) \tag{60}$$

where $t = t(\tau)$ because of the shift in the computational domain. The Lagrange polynomials may be expressed in terms of the Legendre polynomials as

$$\phi_k(\tau) = \frac{\left(\tau^2 - 1\right) L_N'(\tau)}{(\tau - \tau_k) N(N+1) L_N(\tau_k)}, \qquad k = 0,...,N \tag{61}$$

Approximations to the state equations are obtained by integrating Eq. (60),

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + \frac{(t_f - t_0)}{2} \int_{-1}^{1} \sum_{j=0}^{N} \phi_j(\tau) \boldsymbol{f}(t_j) \, \mathrm{d}\tau, k = 1,...,N \tag{62}$$

Eq. (62) can be re-written in the form of Gauss-Lobatto quadrature approximations as

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + \frac{(t_f - t_0)}{2} \sum_{j=0}^{N} \mathcal{I}_{k-1,j} \boldsymbol{f}(t_j), \qquad k = 1,...,N \tag{63}$$

where the entries of the $N \times (N+1)$ integration matrix $\mathcal{I}$ are derived by Williams (2006). The cost function is approximated via a full Gauss-Lobatto quadrature as

$$\mathcal{J}_N = \mathcal{M}[\boldsymbol{x}_N] + \frac{(t_f - t_0)}{2} \sum_{j=0}^{N} \mathcal{L}[\boldsymbol{x}_j, \boldsymbol{u}_j, t_j]\, w_j \tag{64}$$

Thus the discrete states and controls at the LGL points $(\boldsymbol{x}_0,...,\boldsymbol{x}_N,\boldsymbol{u}_0,...,\boldsymbol{u}_N)$ are the optimization parameters, which means that the path constraints and box constraints are easily enforced. The continuous problem has been converted into a large-scale parameter optimization problem. The resulting nonlinear programming problem is solved using SNOPT in this work. In all cases analytic Jacobians of the cost and discretized equations of motion are provided to SNOPT.

Alternatives to utilization of nonlinear optimization strategies have also been suggested. An example of an alternative is the use of iterative linear approximations, where the solution is linearized around the best guess of the optimal trajectory. This approach is discussed in more detail for the pseudospectral method in (Williams, 2004).

5.2 Optimal Control Strategy

Using the notation presented above, the basic notion of the real-time optimal control strategy is summarized in Fig. 2. For a given mission objective, a suitable cost function and final conditions would usually be known a priori. This is input into the two-point boundary value problem (TPBVP) solver, which generates the open-loop optimal trajectories $\boldsymbol{x}^*(t), \boldsymbol{u}^*(t)$. The optimal control input is then used in the real-system, denoted by the "Control Actuators" block, producing the observation vector $\boldsymbol{y}(t_k)$. This is fed into the CKF to produce a state estimate, which is then fed back to update the optimal trajectory by letting $t_0 = t$, and using $t_f - t$ as the time to go.

Imposing hard terminal boundary conditions can make the optimization problem infeasible as $t_f - t \to 0$. In many applications of nonlinear optimal control, a receding horizon strategy is used, whereby the constraints are always imposed at the end of a finite horizon $T = t_f - t$, where T is a constant, rather than at a fixed time. This can provide advantages with respect to robustness of the controller. This strategy, as well as some additional strategies, are discussed below.

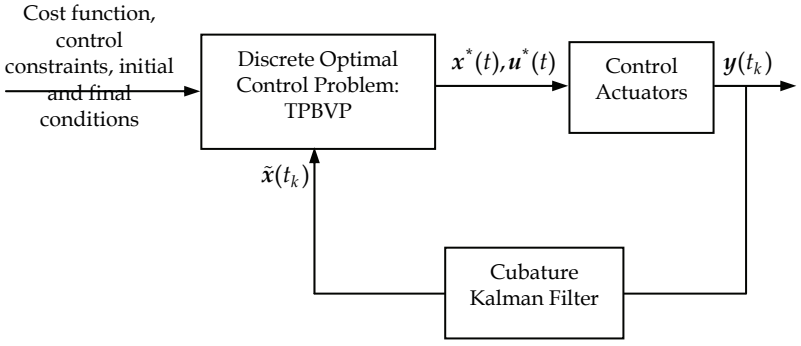


Fig. 2. Real-Time Optimal Control Strategy.

5.3 Issues in Real-Time Optimal Control

Although the architecture for solving the optimal control problem presented in the previous section is capable of rapidly generating optimal trajectories, there are several important issues that need to be taken into consideration before implementing the method. Some of these have already been discussed briefly, but because of their importance they will be reiterated in the following subsections.

5.3.1 Initial Guess

One issue that governs the success of the NLP finding a solution rapidly is the initial guess that is provided. Although convergence of SNOPT can be achieved from random guesses (Ross & Gong, 2008), the ability to converge from a bad guess is not really of significant benefit. The main issue is the speed with which a feasible solution is generated as a function of the initial guess. It is conceivable for many scenarios that good initial guesses are available. For example, for tethered satellite systems, deployment and retrieval would probably occur from fixed initial and terminal points. Therefore, one would expect that this solution would be readily available. In fact, in this work, it is assumed that these "reference" trajectories have already been determined. Hence, each re-optimization would take place with the initial guess provided from the previous solution, and the first optimization would take place using the stored reference solution. In most circumstances then, the largest disturbance or perturbation would occur at the initial time, where the initial state may be some "distance" from the stored solution. Nevertheless, the stored solution is still a "good" guess for optimizing the trajectory. This essentially means that the study of the computational performance should be focused on the initial sample, which would conceivably take much longer than the remaining samples.

5.3.2 Issues in Updating the Control

For many systems, the delay in computing the new control sequences is not negligible. Therefore, it is preferable to develop methods that adequately deal with the computational delay for the general case. The simplest way of updating the control input is illustrated in Fig. 3. The method uses only the latest information and does not explicitly account for the time delay. At the time $t = t_i$, a sample of the system states is taken $x(t_i)$. This information is used to generate a new optimal trajectory $x_i(t), u_i(t)$. However, the computation time required to calculate the trajectory is given by $\Delta t_i = t_{i+1} - t_i$. During the delay, the previous optimal control input $u_{i-1}(t)$ is applied. As soon as the new optimal control is available it is applied (at $t = t_{i+1}$). However, the new control contains a portion of time that has already expired. This means that there is likely to be a discontinuity in the control at the new sample time $t = t_{i+1}$. The new control is applied until the new optimal trajectory, corresponding to the states sampled at $x(t_{i+1})$, is computed. At this point, the process repeats until $t = t_f$. Note that although the updates occur in discrete time, the actual control input is applied at the actuator by interpolation of the reference controls.

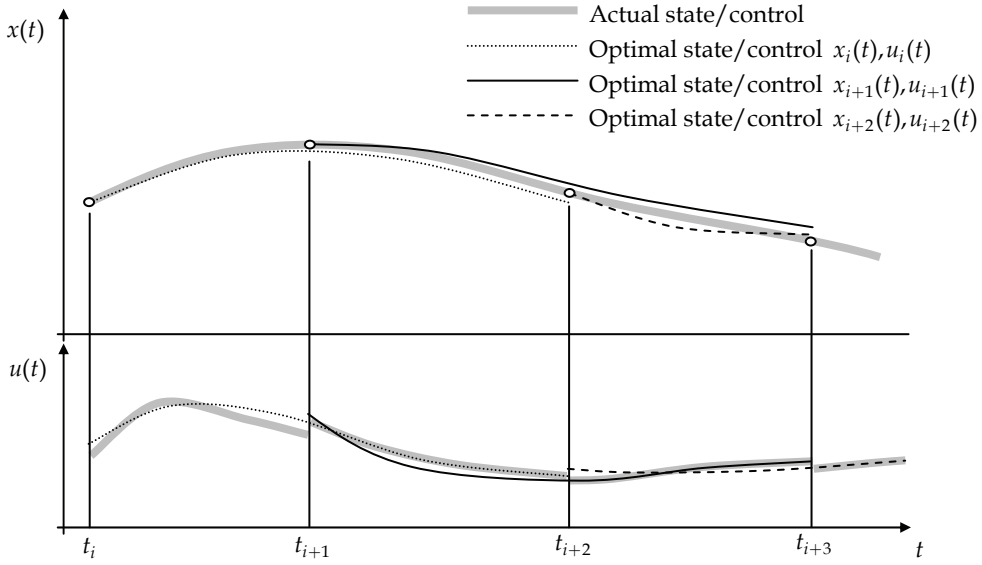


Fig. 3. Updating the Optimal Control using Only Latest Information.

Due to sensor noise and measurement errors, the state sampled at the new sample time $x(t_{i+1})$ is unlikely to correspond to the optimal trajectory that is computed from $x_i(t_{i+1})$. Therefore, in this approach, it is possible that the time delay could cause instability in the algorithm because the states are never matching exactly at the time the new control is implemented. To reduce the effect of this problem, it is possible to employ model prediction to estimate the states. In this second approach, the sample time is not determined by the time required to compute the trajectory, but is some prescribed value. The sampling time must be sufficient to allow the prediction of the states and to solve the resulting optimal control problem, t_{sol}. Hence, $\Delta t_i > t_{\text{sol}}$. The basic concept is illustrated in Fig. 4. At time $t = t_i$, a system state measurement is made $x(t_i)$. This measurement, together with the previously determined optimal control and the system model, allows the system state to be predicted at the new sample time $t = t_{i+1}$,

$$\hat{x}(t_{i+1}) \approx x(t_i) + \int_{t_i}^{t_{i+1}} \dot{x}\left(u_i(t)\right)\, \mathrm{d}t \tag{65}$$

The new optimal control is then computed from the state $\hat{x}(t_{i+1})$. When the system reaches $t = t_{i+1}$, the new control signal is applied, $u_{i+1}(t)$. At the same time, a new measurement is taken and the process is repeated. This process is designed to reduce instabilities in the system and to make the computations more accurate. However, it still does not prevent discontinuities in the control, which for a tethered satellite system could cause elastic vibrations of the tether. One way to produce a smooth control signal is to constrain the initial value of the control in the new computation so that

$$u_{i+1}(t_{i+1}) = u_i(t_{i+1}) \tag{66}$$

That is, the initial value of the new control is equal to the previously computed control at time $t = t_{i+1}$. It should be noted that the use of prediction assumes coarse measurement updates from sensors. Higher update rates would allow the Kalman filter to be run up until the control sampling time, achieving the same effect as the state prediction (except that the prediction has been corrected for errors). Hence, Fig. 4 shows the procedure with the predicted state replaced by the estimated state.

5.3.3 Implementing Terminal Constraints

In standard model predictive control, the future horizon over which the optimal control problem is solved is usually fixed in length. Thus, the implementation of terminal constraints does not pose a theoretical problem because the aim is usually for stability, rather than hitting a target. However, there are many situations where the final time may be fixed by mission requirements, and hence as $t_f - t \to 0$ the optimal control problem becomes more and more ill-posed. This is particularly true if there is a large disturbance near the final time, or if there is some uncertainty in the model. Therefore, it may be preferable to switch from hard constraints to soft constraints at some prespecified time $t = t_{\text{crit}}$, or if the optimization problem does not converge after n_{crit} successive attempts. It is important to note that if the optimization fails, the previously converged control is used until a new control becomes available. Therefore, after n_{crit} failures, soft terminal constraints are used under the assumption that the fixed terminal conditions can not be achieved within the control limits. The soft terminal constraints are defined by

$$\mathcal{M} = \tfrac{1}{2}\left[\boldsymbol{x}(t_f) - \boldsymbol{x}_f\right]^{\top} \boldsymbol{S}_f \left[\boldsymbol{x}(t_f) - \boldsymbol{x}_f\right] \tag{67}$$

The worst case scenario is for fixed time missions. However, where stability is the main issue, receding horizon strategies with fixed horizon length can be used. Alternatively, the time to go can be used up until $t = t_{\text{crit}}$, at which point the controller is switched from a fixed terminal time to one with a fixed horizon length defined by $T = t_f - t_{\text{crit}}$. In this framework, the parameters t_{crit} and n_{crit} are design parameters for the system.

It should also be noted that system requirements would typically necessitate an inner-loop controller be used to track the commands generated by the outer loop (optimal trajectory generator). An inner-loop is required for systems that have associated uncertainty in modeling, control actuation, or time delays. In this chapter, the control is applied completely open-loop between control updates using a time-based lookup table. The loop is closed only at coarse sampling times.

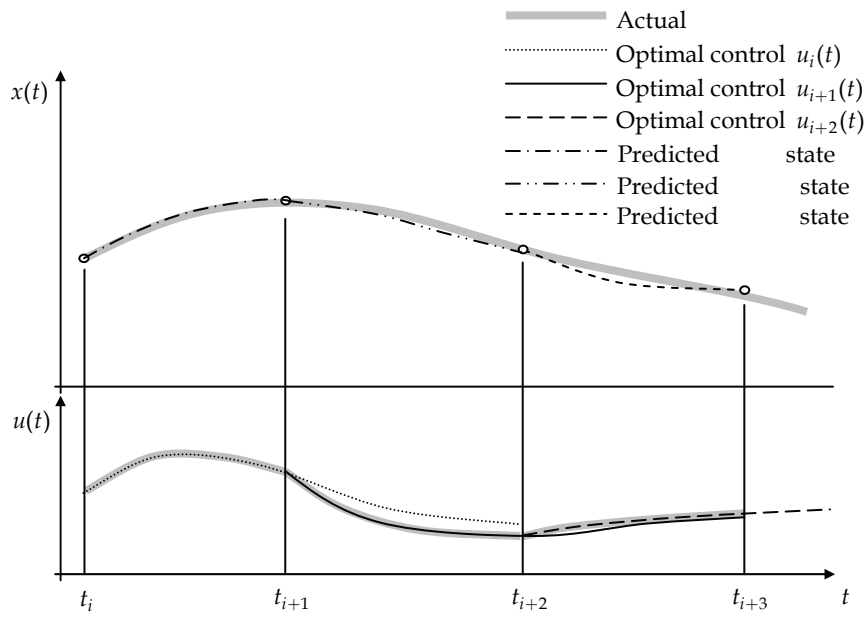


Fig. 4. Updating the Optimal Control with Prediction and Initial Control Constraint.

5.4 Rigid Model In-Loop Tests

To explore the possibilities of real-time control for tethered satellite systems, a simple, but representative test problem is utilized. Deployment and retrieval are two benchmark problems that provide good insight into the capability of a real-time controller. Williams (2008) demonstrated that deployment and retrieval to and from a set of common boundary conditions leads to an exact symmetry in the processes. That is, for every optimal deployment trajectory to and from a set of boundary conditions, there exists a retrieval trajectory that is mirrored about the local vertical. However, it is also known that retrieval is unstable, in that small perturbations near the beginning of retrieval are amplified, whereas small perturbations near the beginning of deployment tend to remain bounded. Therefore, to test the effectiveness of a real-time optimal controller, the retrieval phase is an ideal test case.

The benchmark problem is defined in terms of the nondimensional parameters as: Minimize the cost

$$J = \int_{t_0}^{t_f} (\Lambda'')^2 \, \mathrm{d}t \tag{68}$$

subject to the boundary conditions

$$[\theta, \theta', \Lambda, \Lambda']_{t=t_0} = [0,0,1,0], \qquad [\theta, \theta', \Lambda, \Lambda']_{t=t_f} = [0,0,0.1,0] \tag{69}$$

and the tension control inequality

$$0.01 \le u \le 4 \tag{70}$$

which is designed to prevent the tether from becoming slack, and to prevent the tether from severing. The control input for this test case is defined as $u = T / [m_1 \dot{\nu}^2 L_r (m_2 + m_t) / m]$.

5.4.1 Preliminary Study on Computation Time

To gauge the effectiveness of performing computations of the optimal control in real-time, the problem of tether retrieval was solved using cold-starts with random perturbations to the initial conditions. Since the computation of the control is most critical at the initial time (because the initial state may be very far from the reference state), a numerical study of the performance of the solution algorithm was run for 1000 computations. In terms of actual implementation, if the sampling time is short enough, subsequent convergence is almost always quicker than the initial computation.

The retrieval problem is posed in nondimensional units, with a nondimensional time of 6 rad. For a tether system in low Earth orbit at an altitude of 500 km, the total maneuver time is roughly 5450 sec. The update time with a good guess of the trajectory averages 0.09 sec in MATLAB 2009a on a Core 2 Processor running Windows XP. Clearly, this easily allows real-time computation of the trajectory with over 50000 samples. However, as noted, the critical time is the first update when the trajectory may be far from the reference or when a good initial guess may not be available. A study of 1000 computations with different initial conditions, but with the same infeasible guess for the trajectory was performed. The initial conditions were distributed randomly in the ranges $|\delta\theta(0)| \le 0.2$ rad, $|\delta\theta'(0)| \le 0.1$, and $|\delta\Lambda(0)| \le 0.02$. Fig. 5 shows a summary of the results from these computations. The level of discretization was set to be N = 30 for this study. The mean computation time was determined to be 0.164 sec.

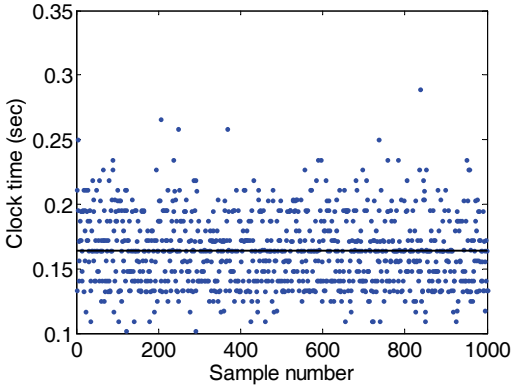


Fig. 5. Summary of Results from Study of Computation of Optimal Trajectories.

The minimum time was 0.102 sec and the maximum time was 0.290 sec. Even in the worst possible case, it would still be possible to implement a sampled-data feedback controller (using MATLAB) with roughly 18000 samples. It should also be noted that convergence was achieved in every case. The CPU time as calculated in Windows represents the worst

case that could be achieved using a dedicated embedded system. The Windows scheduler can schedule the control process in- and out- at different times. The resolution of the scheduler can be seen in the discrete banding of the mean CPU time in Fig. 5, rather than completely random times.

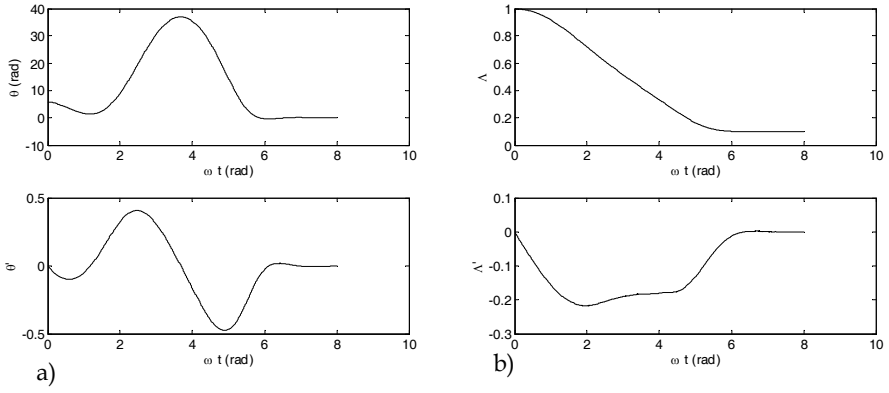
5.4.2 Closed-Loop Control

To examine the actual performance of the controller for dealing with disturbances, the control model is used with external perturbations included via the Q_θ and Q_Λ terms in the equations of motion. For simplicity, the perturbations are generated randomly such that $|Q_\theta| \leq 0.05$ and $|Q_\Lambda| \leq 0.05$. This corresponds to disturbances on the subsatellite on the order of several Newtons, whose actual values depend on the system geometry. The number of major iterations was limited to 50.

The terminal weighted matrix is selected as $\boldsymbol{S}_f = \text{diag}[100,100,100,100]$, and the controller is switched at 4 rad from hard terminal constraints to soft constraints. Numerical results are shown in Fig. 6. Fig. 6a and 6b shows that the terminal constraints are met reasonably accurately, despite not being enforced with hard constraints. The mean CPU time for the whole trajectory is 0.159 sec, the standard deviation is 0.0744 sec, the minimum time is 0.04 sec, and the maximum time is 1.442 sec. Prior to the change in controller, the mean CPU time is 0.1265 sec, whereas after the change the mean CPU time increases to 0.223 sec. Therefore, the smooth control input in the terminal phases of the trajectory comes at the expense of a 76% increase in mean computation time. This is still well within the sampling time of the controller.

6. Closed-Loop Control in Simulation Environment

The results presented in the previous section utilized tension as the control input. Tension has been widely used as the control input in the literature, but it has several drawbacks. It introduces long-term errors in the trajectories because of inaccuracies in the system properties, errors in the gravity model, and tether oscillations. A better choice is to control the reel speed or rate of change of reel speed. In the high fidelity simulation environment, the control is implemented as the rate of change of nondimensional reel rate.



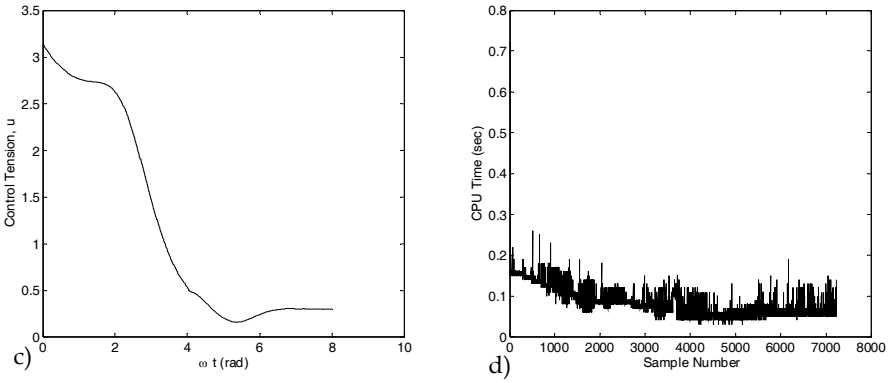


Fig. 6. Real-Time Computation of Retrieval Trajectory with 1 sec Sampling Time, Receding Horizon after $\omega t = 4$ rad and Model Prediction of States with Continuous Control Enforced, a) Libration Dynamics, b) Length Dynamics, c) Control Tension, d) Computation Time.

6.1 Simulation Environment

The simulation environment used for testing the closed-loop control behavior is built in Simulink™, which is itself based on the MATLAB environment. Simulink provides a graphical approach for modeling and control of complex systems. It has the distinct advantage of being able to provide generated C-code targeting real-time operation directly from the underlying model. This feature requires additional supporting tools available from Mathworks. In the context of the current chapter, a Simulink model is used to simulate four distinct elements of the system. Fig. 7 illustrates the interconnections of the four system elements. These are: 1) Variable-Step, Multibody Propagation (bead tether model), 2) Sensor models, 3) Tether state estimation, and 4) Pseudospectral predictive control. One of the complicating factors in simulating the predictive control system is that a high-fidelity, variable step integration algorithm is needed to propagate the multibody dynamic equations.

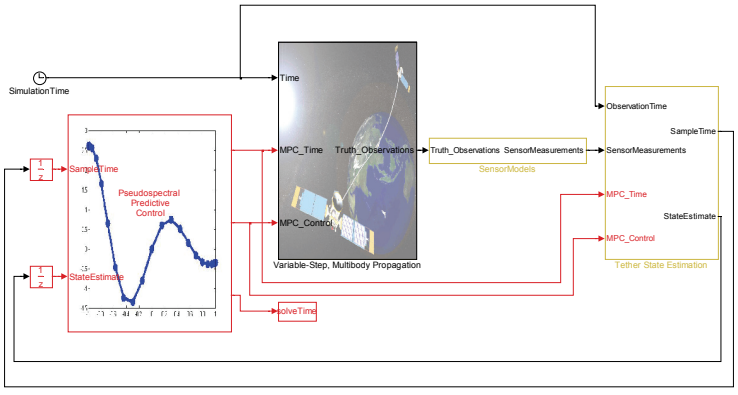
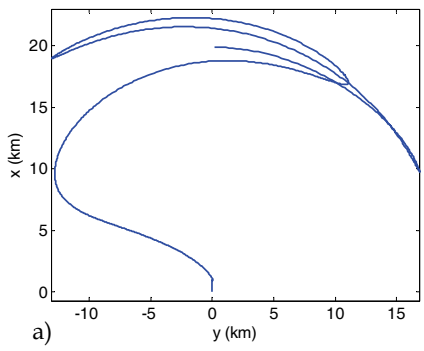


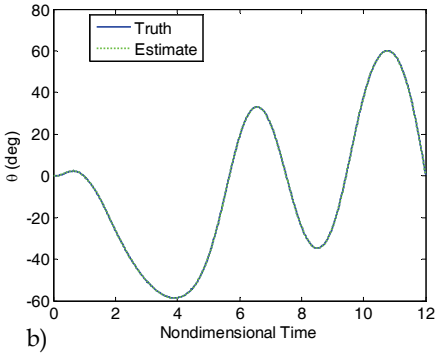
Fig. 7. Simulink simulation model for closed-loop model predictive control.

Although Simulink supports variable-step integration algorithms, it does not easily allow for the combination of variable-step integration and discrete sampling updates of the system being propagated. For example, the multibody model requires regular checks on the length of the deploying segment for the introduction or removal of an element from the model. To overcome this, a custom S-function block is used which employs the LSODA variable-step integration library. The LSODA library is coded in Fortran, but was ported to C via f2c.
The sensor models block implements the tension and GPS models for the system. The tether state estimation block implements the Kalman filter for estimating the tether state in a discrete-time manner. Finally, the pseudospectral predictive control block implements the predictive controller.

6.2 Example: Closed-Loop Control with State Estimator

One of the future applications of tethered satellite systems is for capture and rendezvous of a satellite in a coplanar orbit. In such an application, timing is critical for mission success. A similar application where timing is not as critical is the deorbit of a payload, similar to the idea of the YES-2 mission. In these examples, the control objective is similar in that it requires the generation of a large in-plane swinging motion. As an example, the control objective of rendezvous with a target satellite is used. The rendezvous conditions have been derived in detail by Williams (2006) for the general case of circular and elliptical orbits as a function of tether length.
The objective in this section is to deploy the tether from a length of 1 km to a length of 20 km to achieve a nondimensional in-plane libration rate of -1.5. For a target satellite in a circular orbit, the reel-rate at capture must be zero. The cost function that aids in minimizing tether oscillations is given in Eq. (68). The tether mass density is 1 kg/km, the subsatellite mass is 200 kg, and the orbit radius is 500 km. The tension measurement noise is 0.5 N, the reel-rate noise is 0.05 m/s, and the GPS error terms noise are R_{GPS} = 25 m², τ_{GPS} = 0.01 (nondimensional). Solutions are obtained using N = 30, with a fixed sampling time of 0.01 rad ≈ 9 sec. The final time is set at 12 rad in nondimensional units.





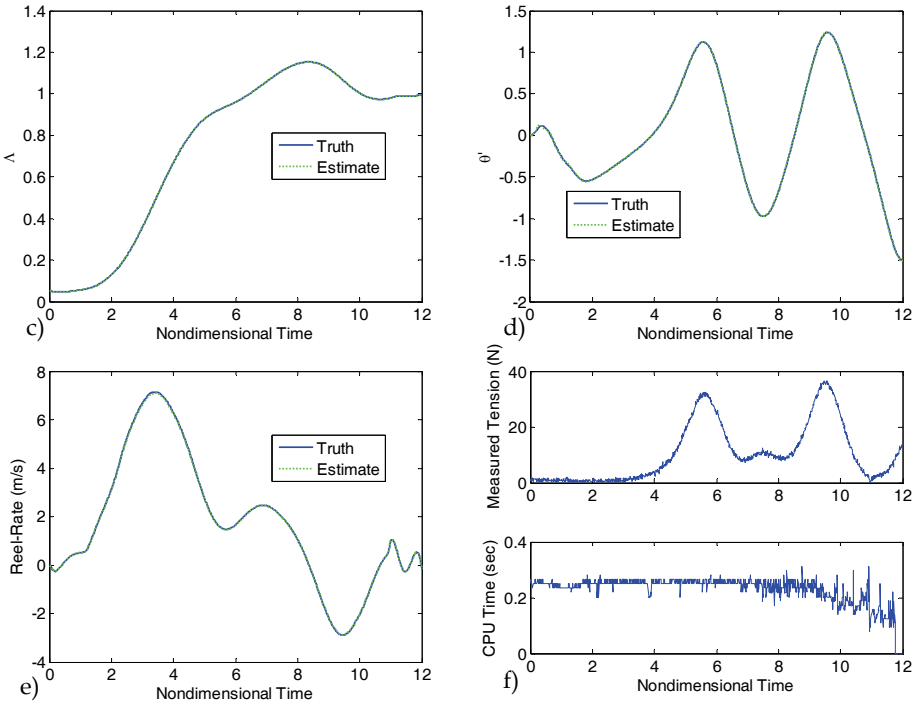


Fig. 8. Closed-loop optimal control of tethered satellite system, a) Tether tip trajectory, b) In-plane libration angle, c) Nondimensional tether length, d) Nondimensional libration rate, e) Reel-rate, f) Measured tension and computation time.

Fig. 8 shows the results of a closed-loop simulation in Simulink using the multibody tether model in combination with the CKF. The results show that the tether is initially over-deployed by about 20%, then reeled back-in to generate the swing velocity required for capture. The final conditions are met to within a fraction of a percent in all state variables despite the measurement errors and uncertainties. The peak reel-rate is approximately 7 m/s, and the variation in reel-rate is smooth throughout the entire maneuver. The average CPU time is 0.23 sec, peaking to 0.31 sec.

7. Conclusion

Modern computing technology allows the real-time generation of optimal trajectories for tethered satellites. An architecture that implements a closed-loop controller with a nonlinear state estimator using a subset of available measurements has been demonstrated for accurately deploying a tether for a rendezvous application. This strategy allows the controller to adapt to large disturbances by recalculating the entire trajectory to satisfy the mission requirements, rather than trying to force the system back to a reference trajectory computer offline.

8. References

Barkow, B.; Steindl, A.; Troger, H. & Wiedermann, G. (2003). Various methods of controlling the deployment of a tethered satellite. *Journal of Vibration and Control*, Vol. 9, 187-208.

Blanksby, C. & Trivailo, P. (2000). Assessment of actuation methods for manipulating tip position of long tethers. *Space Technology*, Vol. 20, No. 1, 31-39.

Colombo, G.; Gaposchkin, E. M.; Grossi, M. D. & Weiffenbach G. C. (1975). The 'skyhook': a shuttle-borne tool for low-orbital-altitude research. *Meccanica*, March, 3-20.

Dunbar, W. B.; Milam, M. B.; Franz, R. & Murray, R. M. (2002). Model predictive control of a thrust-vectored flight control experiment. 15th IFAC World Congress on Automatic Control, Barcelona, Spain.

Elnagar, G.; Kazemi, M. A. & Razzaghi, M. (1995). The pseudospectral legendre method for discretizing optimal control problems. *IEEE Transactions on Automatic Control*, Vol. 40, No. 10, 1793-1796.

Fujii, H. & Ishijima, S. (1989). Mission function control for deployment and retrieval of a subsatellite. *Journal of Guidance, Control, and Dynamics*, Vol. 12, No. 2, 243-247.

Fujii, H. A. & Anazawa, S. (1994). Deployment/retrieval control of tethered subsatellite through an optimal path. *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 6, 1292-1298.

Fujii, H.; Uchiyama, K. & Kokubun, K. (1991). Mission function control of tethered subsatellite deployment/retrieval: In-plane and out-of-plane motion. *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 2, 471-473.

Gill, P. E.; Murray, W. & Saunders, M. A. (2002). SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, Vol. 12, No. 4, 979-1006.

Arasaratnam, I. & Haykin, S. (2009). Cubature kalman filters. *IEEE Transactions on Automatic Control*, Vol. 54, 1254-1269.

Kim, E. & Vadali, S. R. (1995). Modeling issues related to retrieval of flexible tethered satellite systems. *Journal of Guidance, Control, and Dynamics*, Vol. 18, 1169-1176.

Kruijff, M.; van der Heide, E. & Ockels, W. (2009). Data analysis of a tethered spacemail experiment. *Journal of Spacecraft and Rockets*, Vol. 46, No. 6, 1272-1287.

Lakso, J. & Coverstone, V. L. (2000). Optimal tether deployment/retrieval trajectories using direct collocation. *AIAA/AAS Astrodynamics Specialist Conference*, 14-17 Aug. 2000, AIAA Paper 2000-4349.

Lorenzini, E. C.; Bortolami, S. B.; Rupp, C. C. & Angrilli, F. (1996). Control and flight performance of tethered satellite small expendable deployment system-II. *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 5, 1148-1156.

Misra, A. K. & Modi, V. J. (1982). Deployment and retrieval of shuttle supported tethered satellites. *Journal of Guidance, Control, and Dynamics*, Vol. 5, No. 3, 278-285.

Misra, A. K. & Modi, V. J. (1986). A Survey on the dynamics and control of tethered satellite systems. *Advances in the Astronautical Sciences*, Vol. 62, 667-719.

Nordley, G. D. & Forward, R. L. (2001). Mars-earth rapid interplanetary tether transport system. I – Initial feasibility analysis. *Journal of Propulsion and Power*, Vol. 17, No. 3, 499-507.

Ross, I. M. & Fahroo, F. (2003). Legendre pseudospectral approximations of optimal control problems. *Lecture Notes in Control and Information Sciences*, Vol. 295, 327-342.

Ross, I. M. & Gong, Q. (2008). Guess-free trajectory optimization. *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, August, Honolulu.

Rupp, C. C. (1975). A tether tension control law for tethered subsatellites deployed along the local vertical. *NASA TM X-64963*, Marshall Space Flight Center, Alabama.

Vadali, S. R. & Kim, E. S. (1991). Feedback control of tethered satellites using lyapunov stability theory. *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 4, 729-735.

Wiedermann, G.; Schagerl, M.; Steindl, A. & Troger, H. (1999). Simulation of deployment and retrieval processes in a tethered satellite system mission. Paper presented at the International Astronautical Congress, Amsterdam, The Netherlands, Oct.

Williams, P. & Blanksby, C. (2008). Optimal prolonged spacecraft rendezvous using tethers. *International Review of Aerospace Engineering*, Vol. 1, No. 1, 93-103.

Williams, P. (2004). Application of pseudospectral methods for receding horizon control. *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 2, 310-314.

Williams, P. (2004). Guidance and control of tethered satellite systems using pseudospectral methods. *AAS/AIAA Spaceflight Mechanics Meeting*, Wailea, Hawaii, Feb., Paper AAS 04-169.

Williams, P. (2005). Receding horizon control using gauss-lobatto quadrature approximations. *AAS/AIAA Astrodynamics Specialist Conference*, Aug. 7-11, Embassy Suites Hotel, Lake Tahoe Resort, Paper AAS 05-349.

Williams, P. (2006). Dynamics and control of spinning tethers for rendezvous in elliptic orbits. *Journal of Vibration and Control*, Vol. 12, No. 7, 737-771.

Williams, P. (2006). A gauss-lobatto quadrature approach for solving optimal control problems. *ANZIAM Journal* (E), Vol. 47, July, C101-C115.

Williams, P. (2008). Optimal deployment/retrieval of tethered satellites. *Journal of Spacecraft and Rockets*, Vol. 45, No. 2, 324-343.

Williams, P.; Blanksby, C. & Trivailo, P. (2004). Tethered planetary capture maneuvers. *Journal of Spacecraft and Rockets*, Vol. 41, No. 4, 603-613.

Williams, P.; Hyslop, A.; Stelzer, M. & Kruijff, M. (2008) YES2 optimal trajectories in presence of eccentricity and aerodynamic drag. *Acta Astronautica*, to be published.

Xu, D. M.; Misra, A. K. & Modi, V. J. (1981). Three dimensional control of the shuttle supported tethered satellite systems during deployment and retrieval. *Proceedings of the 3rd VPISU/AIAA Symposium on Dynamics and Control of Large Flexible Spacecraft*, Blacksburg, VA, 453-469.

Xu, D. M.; Misra, A. K. & Modi, V. J. (1986). Thruster-augmented active control of a tethered subsatellite system during its retrieval. *Journal of Guidance, Control, and Dynamics*, Vol. 9, 663-672.

11

MPC in urban traffic management

Tamás Tettamanti[1], István Varga[1,2] and Tamás Péni[2]
[1]*Budapest University of Technology and Economics*
[2]*HAS Computer and Automation Research Institute*
Hungary

1. Introduction

More and more people are concerned about the negative phenomenon resulted by the negative effects of the growing traffic motorization. Traffic congestion is the primary direct impact which became everyday occurrence in the last decade. As world trade is continuously increasing, it is obvious that congestions represent also a growing problem. The capacity of the traffic networks saturates during rush hours. At the same time, the traditional traffic management is getting less effective in sustaining a manageable traffic flow. Therefore, external impacts appear causing new costs for the societies. As a possible solution the predictive control based strategy can be applied. The chapter investigates the applicability of MPC strategy specialized in urban traffic management in order to relieve traffic congestion, to reduce travel time and to improve homogeneous traffic flow. Over the theory the realization of the control method is also presented. Firstly we give a historical summary of adaptive traffic control. The brief results of MPC and its related methods in urban traffic control are presented. Then we introduce the modeling possibilities of urban traffic as the appropriate model means an important aspect of the control design. The use of MPC requires a state space theory approach. Therefore the so called Store-and-forward model is chosen which can be directly translated to state space. We analyze the model in details showing the real meaning of system matrices. The constraints of urban traffic system is also discussed which heavily influence modeling and control. The next section presents the simulation environment which is used to demonstrate the developed control methods. Thereinafter we present the main results of MPC in traffic application. The idea to apply MPC in urban traffic network is induced by the fact that the distance is relatively short between several intersections with traffic lights. Hence it is advisable to coordinate the operation of the intersection controller devices. Several intersections are near to each other in smaller or bigger networks, primarily in cities, the coordination is especially emphasized. The development of new control strategies is a real demand of nowadays. One of the possible solutions is the practical application of MPC. The aim of the control is to increase capacity. To test and validate our control strategy we apply it to a real-word transportation network where the actual system is not efficient enough to manage the traffic in rush hours. The simulation results show the effectiveness of the control design. After the presentation of the practical urban traffic MPC the distributive solution of MPC has to be discussed. As the computational demand depends on the size of the network an efficient calculation method is

sought to solve online the MPC problem. The classical scheme for adaptive road traffic management structure is usually based on control center which processes and computes all signal control for the network. Another method for the control system architecture is the decentralized and distributed control scheme. This approach has numerous economical and technological advantages. Distributed traffic control is developed using iterative solution. The so-called Jacobi iteration algorithm is an efficient method to solve constrained and nonlinear programming problem which the original problem can be transformed for. An additional feature of the developed strategy is the ability to manage priority. If a preferred vehicle arrives to any junction of the network it will be automatically indicated. Its stage will be handled with priority getting maximum green time as possible in every cycle until the vehicle will not leave the intersection. It means practically that the cost function is dynamically modified by the system weights depending of presence of any preferred vehicles. Finally we would like to introduce the robust MPC problem in traffic management as our future work. The robustness of the traffic management means that even with the presence of some disturbances the system is able to find optimal control solution. We discuss the modification of the traffic model introduced in third section since the chosen method requires a special model structure.

2. Brief historical summary of adaptive road traffic control

In case the distance is relatively short between several intersections with traffic lights it is advisable to co-ordinate the operation of the intersection controller devices. The coordination may include public transport devices and pedestrian traffic besides vehicles. Where several intersections are near to each other in smaller or bigger networks, primarily in cities, the coordination is especially emphasized.

In the 1970's a new control strategy appears in the road traffic management. Beside the already extant fixed-time and traffic-actuated strategies the traffic-adaptive control is invented. A traffic control system that continuously optimizes the signal plan according to the actual traffic load is called an adaptive traffic control system. The essence of the functioning is that the changes to the active signal plan parameters are automatically implemented in response to the current traffic demand as measured by a vehicle detection system. Such system can be used as local or network-wide control.

The appearance of the adaptivity induces new developments of traffic control methods. The first adaptive systems like SCOOT (Hunt et al., 1982) or SCATS (Lowrie, 1982) are based on heuristic optimization algorithms. In the 1980's new optimization methods are introduced based on rolling horizon optimization using dynamic programming. Some examples are OPAC (Gartner, 1983), PRODYN (Farges et al. 1983), and RHODES (Sen & Head, 1997).

In the middle of the 1990's the first control method is introduced which adopts results of the modern control theory. The TUC system (Diakaki et al., 1999) applies a multivariable regulator approach to calculate in real time the network splits, while cycle time and offsets are calculated by other parallel algorithms. The basic methodology employed for split control by TUC is the formulation of the urban traffic control problem as a linear-quadratic (LQ) optimal control problem. The advantage of LQ control is the simplicity of the required real-time calculations which is an important aspect in network-wide signal control. However the algorithm has a main disadvantage. LQ control is not able to manage constraints on the control input (its importance is discussed in the next section). Therefore a

posteriori application is needed to force the constraints which may lead to suboptimal solution.
In the early 2000's the first results are published in the subject of MPC based traffic control. However these publications (e.g. Bellemans et al., 2002; Hegyi et al., 2003) are related to ramp metering and variable speed limit control of the freeway traffic management. MPC based urban traffic control approach is published by Tettamanti et al. (2008). The paper consists theory, realization and a real-word example. The main result is the possibility to overcome the disadvantage of the LQ problem mentioned above as the MPC method can take the constraints into consideration. These results constitute the basis of the chapter. The paper of Aboudolas et al. (2009) is published investigating large-scale traffic control problem and introducing the open-loop quadratic-programming control (QPC) as a possible method for optimal traffic management. The paper concludes that for the application of the QPC methodology in real time, the corresponding algorithms may be embedded in a rolling-horizon (model-predictive) scheme which constitutes the part of future works.
In 2010 as a development result of Tettamanti et al. (2008) the paper of Tettamanti & Varga (2010) is published which introduces a distributed realization of an MPC based traffic control system. The publication's results will be also enlightened in detail in the chapter.

3. Urban traffic modeling

Modeling and control are coherent notions in control theory as the model highly determines the applicable methods for control. In the previous chapter various control approaches were presented. All of them use an appropriate traffic modeling technique for functioning. Apparently, the modern control theory based traffic management strategies apply the state space approach. The state space modeling is derived from the so called Store-and-forward model (Gazis & Potts, 1963) which introduces a model simplification that enables the mathematical description of the traffic flow process. This modeling technique opens the way to the application of a number of highly efficient optimization methods such as LQ control, MPC, or robust LMI based control. Before to begin to investigate the MPC based traffic control the properties of the model have to be discussed in detail.

3.1 From Store-and-forward traffic modeling to state space representation

The following derivation of the state space model reflects the results of the paper of Diakaki et al. (1999).

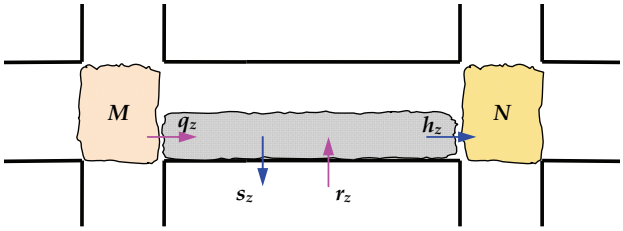


Fig. 1. The Store-and-forward traffic model

The two basic parts of an urban road traffic network are intersection and link. The combination of these elements constitutes the traffic network with link $z \in Z$ and junction

$j \in J$ which are defined geometrically exactly. Each signalized junction j has its own sets of incoming I_j and outgoing O_j links. Figure 1. shows the coherence (link z) of two neighboring intersections (M, N) in the transportation network where $z \in O_M$ and $z \in I_N$. The dynamic of link z is described by the conservation equation:

$$x_z(k+1) = x_z(k) + T\left[q_z(k) - h_z(k) + r_z(k) - s_z(k)\right] \tag{1}$$

where $x_z(k)$ measures the number of vehicles within link z, practically the length of queue, at time kT. $q_z(k)$ and $h_z(k)$ are the inflow and outflow, $r_z(k)$ and $s_z(k)$ are the demand and the exit flow during the sample period $[kT,(k+1)T]$. T is the control interval and $k = 0,1..$ is the discrete time index. For simplicity we assume henceforth that the cycle times are equal for each junction $j \in J$, namely $T_{c,j} = T_c$. Moreover T is also equal to T_c. $r_z(k)$ and $s_z(k)$ represent typically the fluctuation between a parking lot and link z or the effects of any non-controlled intersection between M and N. These disturbing flows can be considered as known perturbations if they can be well measured or estimated. In case of unknown disturbances robust control system is needed.

Equation (1) is linear scalar equation for the portrayal of vehicles movement of a given link. But if we wish to define a whole traffic network each link has to be described by its conservation equation and what is more the equations needs to be interconnected. At this point we can change for state space representation which means the appearance of the state and control input vectors together with the coefficient system matrices. The general discrete LTI state space representation is the following:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Ed(k) \\ y(k) &= Cx(k) \end{aligned} \tag{2}$$

Using Equation (2), it is possible to describe the dynamics of an arbitrary urban traffic network (see Fig. 2 as an example).

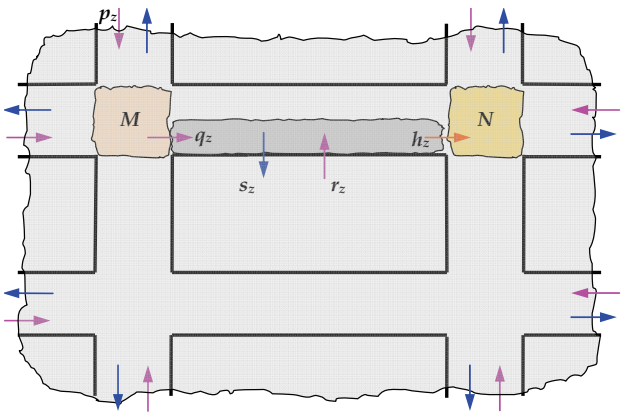


Fig. 2. Dynamics in the urban traffic network

The physical meaning of matrices and vectors is elementary to understand the model. The state equation form can be achieved using all conservation equations, arranging them in one linear matrix equality. In our case the state matrix A is practically considered as an identity matrix. The elements of the state vector $x(k)$ represent the number of vehicles of each controlled link. The second term of the state equation is the product of input matrix B and control input u. Vector u contains the green times of all stages. Their numerical values are the results of a corresponding controller at each cycle. Naturally the number of states is equal to the number of controlled links in the network. The product $Bu(k)$ is arising from the part $T[q_z(k)-h_z(k)]$ of Equation (1) which means the difference of the inflow and the outflow of a link during the control interval. $q_z(k)$ and $h_z(k)$ are directly related to control input (green time), saturation flow (S) and turning rate (t) in a signalized network. To understand the construction of B the parameters S and t have to be discussed. Saturation flow represents the outflow capacity of link $z \in Z$ during its green time. A standard value for saturation flow is $S = 0.5[veh/\sec]$ which is considered constant in practice. Turning rate represents the distribution of turnings of vehicles from link $z \in O_j$ to links $w \in I_N$. These parameters are defined by the geometry and the rights of way in the traffic network and assumed to be known and constant or time varying. Then matrix $B = \lfloor b_{ij} \rfloor$ can be constructed by the appropriate allocation of the combinations of saturation flow and turning rates. The diagonal values of B are negative S_z as the product $S_z u_z(k)$ represents the outflow from link z. At the same time the inflow to the link z has to be also characterized. Therefore the products $S_z t_{w,z}$ are placed in matrix B such that $b_{ij} = S_z t_{w,z}$ when $i \neq j$. The parameters $t_{w,z}$ ($w \in I_M$) are the turning rates towards link z from the links that enter junction M. Hence the inflow is resulted from the appropriate matrix-vector multiplication for all z.

In state space representation the third term $Ed(k)$ of Equation (2) represents an additive disturbance where $E = I$. $d(k)$ is composed of two type of data. On the one hand it is coming from the part $T[r_z(k)-s_z(k)]$ of Equation (1) where $r_z(k)$ and $s_z(k)$ are considered as measured disturbances. They reflect the difference of the demand and the exit flows of a link during the control interval. On the other hand there is demand $p_z(k)$ at the boundary of the traffic network (Figure 2.) which also has to be taken into consideration in the model. The traffic $p_z(k)$ intending to enter is a measurable value. Therefore it is simply added to the appropriate row of $d(k)$.

To end the state space description of the urban traffic the measurement equation has to be mentioned. As each output inside of the network is a measured state (number of vehicles of the link $z \in Z$) the output equation is simplified to $y(k) = x(k)$ with $C = I$. Note that as the exit links of the network are not controlled they do not have to be confused with the outputs $y(k)$.

Finally, as three of the system matrices are identity matrix (discussed above) the general discrete LTI state space representation for urban traffic simplifies to the following form:

$$\begin{aligned} x(k+1) &= x(k) + Bu(k) + d(k) \\ y(k) &= x(k) \end{aligned} \tag{3}$$

3.2 Constraints of urban traffic control

As store-and-forward modeling technique tries to express the real dynamics and states of the urban traffic there are several constraints which have to be taken into account. The most essential constraints of the urban network are determined by the geometry. It is evident that the maximum number of vehicles is defined by the length of link between two junctions. Naturally the vehicles are considered as passenger car unit (PCU) resulting from appropriate transformation (Webster & Cobbe, 1966). Thus the states are subject to the constraints:

$$0 \le x_z(k) \le x_{z,max} \tag{4}$$

If we consider a network the use of the states constraints can contribute to avoid the oversaturation in the controlled traffic area. In a control scheme beside the state constraints one can define output limitations too. However in our case the states constraints are identically to the output constraints as $C = I$.

The control input is the next variable restricted by some constraints. The first constraint on u is the interval of seconds of green time:

$$u_{z,min} \le u_z(k) \le u_{z,max} \tag{5}$$

Depending on the system setting $u_{z,min}$ (for lack of vehicles on link z) can be zero. It means permanent red signal for the stage in the next control interval. The second control input constraint is represented by the linear combination of green times at junction $j \in J$. The sum of the green times has to be lower as $T_{j,max}$:

$$\sum_{z=1}^{Oj} u_z(k) \le T_{j,max}, \quad j = 1, 2, ..., J, \tag{6}$$

where O_j is the number of stages at junction j, $T_{j,max} = T - L_j$ (L_j is the fixed lost time resulted from the geometry of junction j), and J is the number of controlled intersections.

4. Simulation environment

In the previous sections traffic modeling was introduced which can be used in control design. Moreover the simulation environment has to be discussed similarly as all the methods presented in this chapter were simulated and tested. For simulation we used traffic simulator (VISSIM, 2010), numerical computing software (MATLAB, 2010) and C++ programming language.

VISSIM is a microscopic traffic simulation software for analyzing traffic operations. It is able to simulate network consisting of several intersections and allow the use of external control algorithm in the control processes. These properties make it suitable to use this software by reason of the several junctions and the control algorithms written in MATLAB. VISSIM uses a so-called psycho-physical driver behavior model based on the car-following model of (Wiedemann, 1974). The model describes all the cars found in the system. The vehicles are defined by both physical and psychical parameters (origin, destination, speed, driver

behavior, vehicle type, etc.). The VISSIM simulation is based on an iteration process of acceleration and deceleration.
The communication does not work directly between MATLAB and VISSIM as the simulation can only be accessed via Component Object Model (COM) interface (Roca, 2005). To control the communication a C++ application has to be created. The created C++ program manages the simulation process and controls the data transfer between the software (Figure 3.).

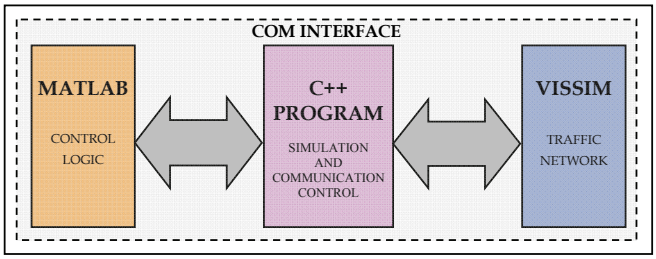


Fig. 3. The simulation process of the system model

5. MPC based urban traffic management

The aim of our research was to elaborate a control process related to network consisting of several junctions which perform the control of all the traffic lights in its sphere of action in a coordinated way depending on the traffic. The controller must be able to dynamically make the traffic signal set of the intersections. From the point of view of realization, this means that before every period a new traffic sign must be generated regarding all the traffic lights, in harmony with the present traffic. To solve the above, MPC technology was chosen since it is able to take all the constraints into consideration in course of the control input setting. To show the efficiency of MPC the control design was tested simulating a real-word traffic network.

5.1 The MPC cost function

The control objective is the minimization and balancing of the numbers of vehicles within the streets of the controlled network. This control objective is approached through the appropriate manipulation of the green splits at urban signalized junctions, assuming given cycle times and offsets. By employing the predictive control model, the dynamic determination (per cycle) of the traffic light's period is possible either with the consideration of the natural constraints existing in the system introduced in Section 3.2.
The state space equation for MPC design can be given as follows:

$$\underbrace{\begin{bmatrix} x(k+1\,|\,k) \\ x(k+2\,|\,k) \\ \vdots \\ x(k+N\,|\,k) \end{bmatrix}}_{\tilde{x}(k+1)} = \underbrace{\begin{bmatrix} x(k)+d(k) \\ x(k)+2d(k) \\ \vdots \\ x(k)+Nd(k) \end{bmatrix}}_{c(k)} + \underbrace{\begin{bmatrix} B & 0 & \cdots & 0 \\ B & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ B & B & \cdots & B \end{bmatrix}}_{\tilde{B}} \underbrace{\begin{bmatrix} u(k\,|\,k) \\ u(k+1\,|\,k) \\ \vdots \\ u(k+N-1\,|\,k) \end{bmatrix}}_{g(k)} \tag{7}$$

where x, d, B and u are elements of Equation (3) already discussed. $\tilde{x}$ is a hyper vector of the state vectors, representing the number of vehicles standing at each controlled link of

the intersections. c is a hyper vector combination of the previous state vector and d. The disturbance d is considered measured and constant during the horizons of kth step. Hence it is multiplied by the value of the current horizon. $\widetilde{B}$ is a lower triangular hyper matrix including the matrix B. g is a hyper vector of the control input vectors (green times), $k = 1,2...$ a is the discrete time index, and N is the length of the MPC horizon.

The MPC algorithm needs the current values of the states at each control interval which means the exact knowledge of the numbers of vehicle. However the states can not be directly measured only estimated using appropriate measurement system (e.g. loop detectors) and estimation algorithm. A possible realization for state estimation was published in paper of Vigos et al. (2007) which is based on the well-known Kalman Filter algorithm (Welch & Bishop, 1995). The estimation error is neglected in the paper.

The elements of B are the combinations of turning rates and saturation flow as discussed in Section 3.1. Saturation flow is not measurable hence a standard value is determined ($S = 0.5[veh/\sec]$). Usually the values of turning rates are also considered constant. Nevertheless, in practice the turnings vary around the nominal rates. Thus a continuous estimation may be applied to ameliorate the MPC algorithm. A possible way to estimate turning rates is to use a finite back stepped state observer, e.g. Moving Horizon Estimation (MHE) method (Kulcsár et al., 2005).

Several choices of the objective function in the optimization literature have been reported. In this chapter we consider the following quadratic cost function characterized by the weighted system states and control inputs:

$$J(k) = \frac{1}{2}\left\{\tilde{x}^T(k) Q \tilde{x}(k) + g^T(k) R g(k)\right\} \rightarrow \min \tag{8}$$

where $Q \succ 0$ and $R \succ 0$ are scalar weighting matrices. Q and R have appropriately chosen tuning parameters in their diagonals. The weightings reflect that the control input variation is lightly punished compared to the state variation. The selection of the appropriate weightings is important, because this could influence (especially the end-point weight) the stability of the closed loop (Kwon & Pearson, 1978). To solve this minimization problem several mathematical software can be applied which provide built-in function for quadratic constrained optimization. The solution of optimization problem (8) leads to the minimization of the vehicle queues waiting for crossing intersections. The control input green time is defined corresponding to the states of intersection branches representing a fully adaptive traffic management.

Different stability proofs exist for receding horizon control algorithms. Maciejowski (2002), Rawlings & Muske (1993) or Mayne et al. (2000) offer different methodological approaches. However the urban traffic is a special case. It is ensured that the system will not turn instable because of the hard physical constraints coming from the network geometry. Accordingly, there is a natural saturation in the system. The states can never grow boundlessly. The instability can appear only if there is an oversaturation in the network. To solve this problem we intend to apply the results of the invariant set theory (Blanchini & Miani, 2007) in the future. It is also has to be noted that if we choose a traffic area to control we do not deal with the traffic outside of the boundary of the network. Obviously the sphere of control action is also an important question in traffic management.

5.2 Test network for simulation

To test MPC technology in urban traffic management we choose a real-world test area situated in the 10th district of Budapest. The test network includes seven neighboring intersections (Figure 4.).

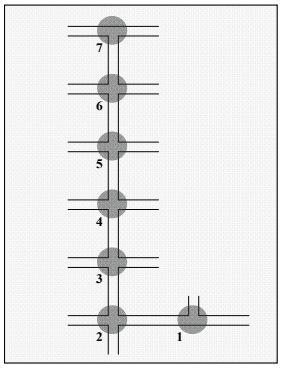


Fig. 4. Schematic representation of the test network consisting of seven junctions

The dimension of the system is 36 which means that we intend to control 36 links. This area is suitable for testing our new control system since the included road stretches have a heavy traffic volume in rush hours. The current traffic management system is offline. The seven junctions are controlled individually. Three of them use fix time signal plan. In the other four intersections detectors help the controllers. They can slightly modify their fix programs. The current control is effective but only in case of normal traffic flow. If the volume of vehicles increases extremely, the system cannot manage the situation and traffic becomes congested before the stop lines. The biggest problem is that the controllers work locally and independently. Our new control design, however, takes the seven junctions into consideration as a real network.

As the MPC cost function (8) represents a quadratic optimization problem the control input was calculated using the built-in *quadprog* function of MATLAB.

5.3 Simulation results

To prove the applicability of the MPC based control design it was compared with the current control system of the test network, which is a partly adaptive control strategy.

The same input traffic volumes were set for both simulations. We used volume data for which the traffic lights were originally designed. The simulation provided similar results for both strategies as we expected. This means the current system is correctly designed, and manages non-extreme traffic flow with good results.

To test the effectiveness of the two systems in case of heavier traffic we generated more intensive traffic flow during the simulation. The original input volumes were increased by 10% in the network. This simulation showed different results to the previous case. The current system could manage the traffic less efficiently compared with the MPC based control system. The simulation time was 1 hour long. The results are presented in Table 1. All important traffic parameters changed in a right way. The new system can provide a very effective control in the test network.

Parameter	OLD STRATEGY	MPC based strategy	Variation
Total travel time per vehicle [sec]	114	96	↓ 16%
Average speed [km/h]	20.6	24.9	↑ 21%
Average delay time per vehicle [sec]	68	56	↓ 18%
Average number of stops per vehicles	3.8	3.1	↓ 18%

Table 1. Average simulation results of the test network

At the same time these simulations were run in a reduced environment. We diminished the number of junctions in the test network from seven to four. Namely the traffic lights at junctions 4., 5., 6. (see Figure 4.) work totally offline. The capacities of these locations increased apparently. So only the junctions 1., 2., 3., and 4. were kept in order to focus on the comparison of the two adaptive strategies.

Parameter	Old strategy	MPC based strategy	Variation
Total travel time per vehicle [sec]	105	96	↓ 9%
Average speed [km/h]	20.5	23.5	↑ 15%
Average delay time per vehicle [sec]	64	52	↓ 19%
Average number of stops per vehicles	1.2	1.2	0%

Table 2. Average simulation results of the test network with design input volumes

Parameter	Old strategy	MPC based strategy	Variation
Total travel time per vehicle [sec]	110	96	↓ 13%
Average speed [km/h]	18.4	23.6	↑ 28%
Average delay time per vehicle [sec]	71	52	↓ 27%
Average number of stops per vehicles	1.5	1.2	↓ 20%

Table 3. Average simulation results of the test network with 10% augmentation of the design input volumes

Alike above, the behavior of the reduced network was analyzed with normal and heavier input traffic volumes. The results ameliorated in both cases (see Table 2. and 3.). The simulation time was 2 hours long.

The aim of the MPC based control is the minimization of the number of vehicles waiting at the stop line. The current system cannot adapt to the increased volume. The average queue length grew strongly during the simulations. However, the MPC strategy is able to manage heavier traffic situations real-time. Figure 5. represents the effectiveness of our system. It shows the variation of average queue lengths in the network.

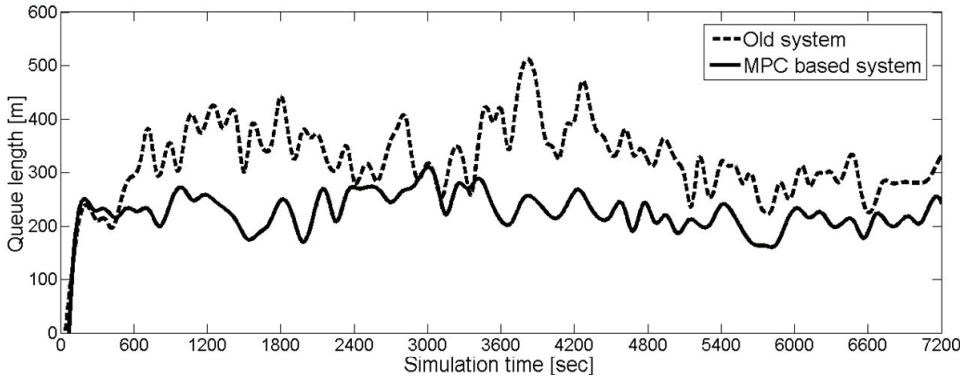


Fig. 5. The variation of average queue lengths in the two different control cases

6. Distributed traffic management system based on MPC

The classical scheme for adaptive road traffic management structure is based on control center which processes and computes all signal control for the network. Another method for the control system architecture is the decentralized and distributed control scheme. This approach has numerous economical and technological advantages.

In this section we present a distributed control system scheme for urban road traffic management. The control algorithm is based on MPC involving Jacobi iteration algorithm to solve constrained and nonlinear programming problem. The distributed control design was also simulated and tested.

6.1 The MPC cost function

We refer to the results of Section 5.1. Substituting $\tilde{x}(k)$ and $g(k)$ in Equation (8) one arrives to:

$$J(k)=\frac{1}{2}g^T\left(q\tilde{B}^T\tilde{B}+rI\right)g+qc^T\tilde{B}g+\frac{1}{2}qc^Tc=\frac{1}{2}g^T\Phi g+\beta^T g+\gamma \tag{9}$$

where q and r are constants coming from the diagonal of the scalar matrices Q and R. As γ is a constant term, finally one has the objective function to minimize:

$$J(k)=\frac{1}{2}g^T(k)\Phi g(k)+\beta^T(k)g(k)\rightarrow \min \tag{10}$$

where Φ is constant matrix as it contains the combination of constant turning rates, saturation rates and fixed tuning parameters. At the same time β contains varying values coming from the current dynamics of the traffic area.

6.2 Multivariable nonlinear programming to solve MPC problem

The solution of the MPC cost function (10) represents a multivariable nonlinear problem subject to linear constraints. It formulates a standard quadratic optimization problem (Bertsekas & Tsitsiklis, 1997):

$$J(k)=\frac{1}{2}g^T\Phi g+\beta^T g\rightarrow \min$$
$$\text{s. t. } Fg-h\leq 0 \tag{11}$$

where matrix inequality $Fg\leq h$ incorporates the constraints (4), (5) and (6) already discussed in Section 3.1.
If Φ is a positive semi definite matrix, (11) gives a convex optimization problem (Boyd & Vanderberghe, 2004). Otherwise one has to use the singular value decomposition method to Φ which results a convex problem. This means a linear transformation to the original problem (11).
Using the duality theory (Bertsekas & Tsitsiklis, 1997) the primal problem can be formulated into Lagrange dual standard form. The basic idea in Lagrangian duality is to take the constraints into account by augmenting the objective function with a weighted sum of the constraint functions. We define the Lagrangian associated with the problem as:

$$L(g,\lambda)=J(k)+\lambda^T(Fg-h) \tag{12}$$

We refer to λ_i as the Lagrange multiplier associated with the *i*th inequality constraint of (11). The dual function is defined as the minimum value of the Lagrangian function. This can be easily calculated by setting gradient of Lagrangian to zero (Boyd & Vanderberghe, 2004). This yields an optimal green time vector (16) which minimizes the primal problem. Hence one arrives to the dual of the quadratic programming problem:

$$J_{DUAL}(k)=\frac{1}{2}\lambda^T P\lambda+w^T\lambda\rightarrow \min$$
$$\text{s. t. } \lambda\geq 0 \tag{13}$$

where P and w are coming from the original problem:

$$P=F\Phi^{-1}F^T \tag{14}$$
$$w=F\Phi^{-1}\beta+h \tag{15}$$

It is shown that if λ^* provides optimal solution for the $J_{DUAL}(k)$ problem then

$$g^*=-\Phi^{-1}\left(\beta+F^T\lambda^*\right) \tag{16}$$

gives also an optimal solution for the primal problem (Rockafellar, 1970).
The dual problem has a simple constraint set compared with the primal problem's constraints. Hence expression (13) represents a standard minimization problem over nonnegative orthant.
A very efficient method, the Jacobi iteration was found to solve the optimization problem. Since Φ is a positive semi definite matrix the *j*th diagonal element of P, given by

$$p_{jj} = f_j^T \Phi^{-1} f_j \tag{17}$$

is positive. This means that for every j the dual cost function is strictly convex along the jth coordinate. Therefore the strict convexity is satisfied and it is possible to use the nonlinear Jacobi algorithm. Because the dual objective function is also quadratic the iteration can be written explicitly. Taking into account the form of the first partial derivative of the dual cost

$$w_j + \sum_{k=1}^{n} p_{jk} \lambda_k \tag{18}$$

the method is given by:

$$\lambda_j(t+1) = \max\left\{0, \lambda_j(t) - \frac{\kappa}{p_{jj}}\left(w_j + \sum_{k=1}^{n} p_{jk}\lambda_k(t)\right)\right\}, \; j = 1, \ldots, n \tag{19}$$

Where $\kappa \succ 0$ is the stepsize parameter which should be chosen sufficiently small and some experimentation may be needed to obtain the appropriate range for κ.
The importance of this method, over its efficiency, is the ability to satisfy the positivity since equation (19) excludes negative solution for λ. Thus, during the MPC control process at each (kth) step the optimal green times can be directly calculated from equation (16) after solving the problem (11).

6.3 Realization of MPC based distributed traffic management system

The economical and technological innovation of the above described control method is represented by the state-of-the-art control design and the optional decentralized realization at the same time.
Generally, the architectures of traffic control systems can be central, decentralized, or mixed. The central management architecture is a frequent strategy based on a central processor which controls all signal controllers in the transportation network. Decentralized and mixed control systems are not so common applications yet. However they have many advantages and represent a new way in traffic control technology. Decentralized management systems carry a higher performance since they can distribute their computations between the traffic controllers. As well as they represent a higher operation safety because of their structural redundancy. Some of these distributed realizations are for example SCATS (Wolshon & Taylor, 1999) or Utopia (UTOPIA, 2010).
The distributed technology can be used in any road traffic network which is equipped with adequate signal controllers and detectors, as well as communication between controllers is also required.
Since the solution of the Jacobi algorithm (19) is an iteration process the computers can distribute their calculations during the operation cycle. Therefore it is suitable for the distributed realization of the MPC problem. Considering a large traffic network the following practical system realization can be applied. Firstly we define the nodes represented by the red cubes on Figure 6. The nodes are the head traffic controllers which participate in the resolution procedure. Every node covers a few intersections (traffic

controllers) which do not participate in the computation. The distributed control network is represented by Figure 6.

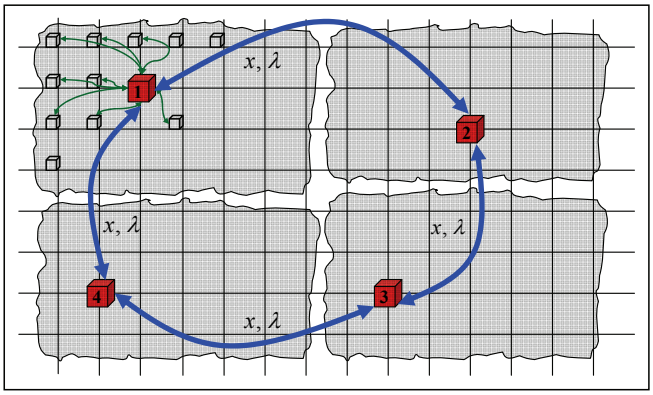


Fig. 6. Distributed MPC control in urban traffic network

The distributed computation is executed during the operation cycle as follows:

1. Communication: At the end of the *k*th cycle all traffic controllers send their measurements (number of vehicles) to their node.
2. Communication: The head controllers share the measurement data with the other nodes.
3. Calculation (which is not the third step practically since it can be started parallel with step 1.): Node 1. starts iteration procedure. After some predefined iteration steps it forwards their computational results to the next node and so on. The transmitted data is the currently calculated vector λ_s where $s = 1,2,...,\kappa$ (κ represents the final iteration step number specified previously).
4. Communication: When the last node finishes the computation (which means practically that $s = \kappa$) it shares the optimal result (λ^*) with the other nodes.
5. Calculation: The head controllers calculate their final calculation. Using Equation (16) the nodes do not need to execute the whole multiplication. But only the specified part of g^* which contains the optimal green times of their traffic controllers.
6. Communication: Finally the nodes pass the optimal green times to every traffic controllers for the next *(k+1)* cycle.

If one wishes to control small traffic network with a few intersections the distributed solution is not certainly required. The calculation of one Jacobi iteration step means simple multiplications and additions of scalars. In case of few states (number of the controlled links) a single controller's performance is sufficient to compute all signal sets of the network. Thus the system is working with redundancy which can be very useful at the same time. The controllers can continuously check their operation comparing their computation results. On the other hand, if one of the signal controllers fails in the network the system can go on with safe functioning.

However with the growth of the number of the system states the computational demand increases quadratically. Therefore larger network requires the distributed solution of the

MPC control. Certainly the solution method is also largely depends on the performance of the actual signal controllers and the communication system.

6.4 Simulation of MPC based distributed control algorithm

To verify the designed control system scheme a closed loop simulation environment was created (Section 4.). The traffic network used for the simulation is equivalent with the one applied in Section 5.2.

As discussed before the appropriate setting for the stepsize parameter κ requires some practical experimentation. This value strongly influences the performance of the calculation. Convergence can be shown when $\kappa = n^{-1}$. However this value may lead to an unnecessarily slow rate of convergence for some problems.

The values in Table 4. shows the variation of the number of steps to achieve convergence. In case of our test network the smallest value with convergence was $\kappa = n^{-0.0525}$.

κ	Number of steps
n^{-1}	6000
$n^{-0.5}$	1000
$n^{-0.1}$	200
$n^{-0.0525}$	150

Table 4. The variation of the number of steps to achieve convergence

We also compared the computation times of the applied methods. Using the *quadprog* function of MATLAB the computation time was about 20 seconds. Conversely the Jacobi algorithm required less than 1 second on average which means 20 times faster calculation. It has to be noted that the Jacobi algorithm was not tested in a distributed way. However even with some communication time the Jacobi iteration is more efficient. On the one hand in our test network the number of states was quite few. The distributed solution is not needed. On the other hand the distributed realization is highly dependent on the current system configuration (measurement accuracy, communication speed, etc.).

6.5 Vehicle priority management in MPC based urban control

The design of an adaptive traffic control system comes with the desirable demand to incorporate vehicle priority management as well. Therefore an additional feature of the designed system is the ability to manage priority.

The scope of the priority management has to be specified as some special vehicle classes (e.g. emergency vehicles) have top-level priority. Therefore they do not need any help from traffic lights to cross the intersections anytime. Our control deals vehicles which are favored compared to the others but not by all means. Vehicles of the public transport are typically of this sort. However one may differentiate the levels of importance even between public vehicles (e.g. an overland bus compared to a local bus).

To operate such system these vehicles have to be able to communicate with the traffic controllers. If a preferred vehicle arrives to any junction of the network it may be automatically indicated by the traffic controller through radio frequency. Its stage can be handled with priority getting maximum green time as possible in every cycle until the

vehicle will not leave the intersection. It means practically that the cost function is dynamically modified by the system weights depending of the presence of any preferred vehicles. Accordingly for the sake of immediate reaction the given junction falls out of the scope of the coordinated traffic control until the vehicle will not leave the intersection. However it can be considered as disturbance.
We refer to the original MPC cost function (8) where Q is a diagonal weighting matrix:

$$Q = \begin{bmatrix} q_1 & & & \\ & q_2 & & \\ & & \ddots & \\ & & & q_n \end{bmatrix} \tag{20}$$

Each diagonal element tunes a state (queue length of controlled links). If there is no preferred vehicle in the scope of control: $q_1 = q_2 = \ldots = q_n$. By online modifying the weight q_i (according to the preferred vehicle's direction) one can assure priority. The measure of the modification of q_i depends on the current level of priority. In practice, the appropriate choice of the weights is an empirical process as it strongly depends on the junction's properties.

7. Future work: Robust MPC in urban traffic management

As future work we introduce the problem of robustness in urban traffic management. In Section 3.1 all disturbances in the state space model were considered as known (measured) values and all possible uncertainties were neglected. These assumptions were taken by practical reasons. However for more precise traffic modeling these factors can be involved in the control scheme determining upper and lower bounds of the uncertainties. This implies the use of a suitable robust control method as well.
The simplest approach to represent disturbances in the system is the bounded unknown external additive disturbance. In this case an additive term appears in the LTI state space model. This approach can deal with state disturbances. As a part of the Ph.D thesis of Löfberg (2003) a Minimax MPC is presented which can be eligible for traffic systems too.
Another possibility to model the uncertainties is the polytopic paradigm. The system matrices $A(k)$ and $B(k)$ of an LTV state space description can be defined by a prespecified polytopic set:

$$\Omega = Co\{[A_1\ B_1], [A_2\ B_2], \ldots, [A_L\ B_L]\} \tag{21}$$

where Co devotes to the convex hull and L is the number of the vertices. Matrix $A(k)$ can be used to express uncertainties of the states. In practice it means for example parking places along the road or non-controlled junctions in the network which result unmeasured state variation. Matrix $B(k)$ can be used to represent uncertainties of the saturation flow rates which are also non-measurable parameters. For polytopic system Kothare et al. (1996) provide an efficient Minimax MPC solution which can be potentially applied in urban traffic management as well.
There is another factor which can be taken into consideration in robust traffic control. In Section 5.1 the demands (d) intending to enter the network were assumed constant and

measured disturbances. In effect they vary continuously. Therefore for fully exact solution varying demands should be considered in the MPC cost function.

8. Conclusion

This chapter introduced the aspects of MPC applied in urban traffic management. As the urban traffic is a complex system having special attributes the appropriate traffic model had to be discussed in details as well. At the same time MPC technology is suitable to control such complex system optimally and real-time. The main control aim was the optimal and coordinated control which can be satisfied. The applicability was demonstrated by several simulations. Furthermore a distributed technology was presented which can be very useful in practice particularly in large traffic network. As an additional feature of MPC based system we showed that an optional vehicle priority management can be easily implemented in the control design. Finally we introduced the possibility of the robust control in urban traffic which is a planned research scope in the future.

9. Acknowledgement

This work is connected to the scientific program of the "Development of quality-oriented and harmonized R+D+I strategy and functional model at BME" project. This project is supported by the New Hungary Development Plan (Project ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002) and by the Hungarian Scientific Research Fund (OTKA) through grant No. CNK 78168 and by the János Bólyai Research Scholarship of the Hungarian Acadamy of Sciences which are gratefully acknowledged.

10. References

Aboudolas, K.; Papageorgiou, M. & Kosmatopoulos, E. (2009). *Store-and-forward based methods for the signal control problem in large-scale congested urban road networks,* Transportation Research Part C: Emerging Technologies, 17:163_174, doi:10.1016/j.trc.2008.10.002

Bellemans, T.; De Schutter, B. & De Moor, B. (2002). *Model predictive control with repeated model fitting for ramp metering.* Singapore, Proceedings of the Fifth IEEE Intelligent Transportation Systems Conference, doi:10.1109/ITSC.2002.1041221

Bertsekas, D. P. & Tsitsiklis, J. N. (1997). *Parallel and distributed computation: Numerical methods*. ISBN 1-886529-01-9, 731 pages

Blanchini, F. & Miani, S. (2007). *Set-Theoretic Methods in Control*. ISBN:0817632557, Birkhäuser, Boston

Boyd, S. & Vanderberghe, L. (2004). *Convex optimization*. Cambridge University Press, ISBN 0 521 83378 7

Diakaki, C.; Papageorgiou, M. & McLean, T. (1999). *Application and evaluation of the integrated traffic-responsive urban corridor control strategy In-TUC in Glasgow.* In CD-ROM of the 78th Annual Meeting of the Transportation Research Board, number 990310, Washington, D.C., USA

Farges, J. L.; Henry, J. J. & Tufal, J. (1983). *The PRODYN real-time traffic algorithm.* In 4th IFAC Symposium on Transportation Systems, pages 307_312

Gartner, N. H. (1983). *OPAC: A demand-responsive strategy for traffic signal control.* Transportation Research Record, 906:75_84

Gazis, D. C. & Potts, R. B. (1963). *The oversaturated intersection.* In: Proceedings of the Second International Symposium on Traffic Theory, London, UK, pp. *221–237.*

Hegyi, A.; De Schutter, B. & Hellendoorn, J. (2003). *MPC-based optimal coordination of variable speed limits to suppress shock waves in freeway traffic.* Denver, USA, American Control Conference.

Hunt, P. B.; Robertson, D. L. & Bretherton, R. D. (1982). *The SCOOT on-line traffic signal optimisation technique.* Traffic Engineering & Control, 23:190_192

Kwon, H. K. & Pearson, A. E. (1978). *On Feedback Stabilization of Time-Varying Discrete Linear System*. IEEE Trans. Autom. Control, 23_3_, pp. 479–481

Maciejowski, J. M. (2002): *Predictive Control with Constraints*. Harlow, England: Prentice Hall

MATLAB, (2010). The MathWorks Inc., available on http://www.mathworks.com

Rawlings, J. B. & Muske, K. R. (1993). *The Stability of Constrained Receding Horizon Control*. IEEE Trans. Autom. Control, 38_10_, pp. 1512– 1516., doi:10.1109/9.241565

Mayne, D. Q.; Rawlings, J. B.; Rao, C. V. & Scokaert, P. O. (2000). *Constrained Model Predictive Control: Stability and Optimality.* Automatica, 36_6_, pp. 789–814., doi:10.1016/S0005-1098(99)00214-9

Kothare, M. V.; Balakrishnan, V. & Morari, M. (1996). *Robust Constrained Model Predictive Control using Linear Matrix Inequalities*. Automatica, vol. 32, no. 10, pp. 1361--79., doi:10.1016/0005-1098(96)00063-5

Kulcsár, B.; Varga, I. & Bokor, J. (2005). *Constrained Split Rate Estimation by Moving Horizon*. 16th IFAC World Congress Prague, Czech Republic

Lowrie, P. R. (1982). *The Sydney co-ordinated adaptive traffic system – principles, methodology logrithms*. In: International Conference on Road Traffic Signaling, London, pp. 67–70

Löfberg, J. (2003). *Minimax approaches to robust model predictive control*. Ph.D thesis, Linköping University, Sweden

Roca, V. (2005). *User Manual for the VISSIM COM interface*, PTV AG.

Rockafellar, R. T. (1970). *Convex analysis*, Princeton University Press

Sen, S. & Head, L. (1997). *Controlled optimization of phases at an intersection.* Transportation Science, 31:5_17, doi: 10.1287/trsc.31.1.5

Tettamanti, T. & Varga, I. (2010). *Distributed traffic control system based on model predictive control.* Periodica Polytechnica ser. Civil. Eng., 54

Tettamanti, T.; Varga, I.; Kulcsár, B. & Bokor, J. (2008). *Model predictive control in urban traffic network management.* 16th IEEE Mediterranean *Conference* on Control and AutomationIn CD ISBN:978-1-4244-2505-1 pp. 1538-1543, Ajaccio, Corsica, France

UTOPIA, (2010). Swarco AG., available on http://www.swarco.com

Vigos, G.; Papageorgiou, M. & Wang, Y. (2007). *Real-time estimation of vehicle-count within signalized links*. Transportation Research Part C, doi: 10.1016/j.trc.2007.06.002

VISSIM, (2010). PTV AG., available on http://www.ptvag.com

Webster, F. V. & Cobbe, B. M. (1966). *Traffic Signals.* Technical Paper 56. HMSO, London

Welch, G. & Bishop, G. (1995). *An Introduction to the Kalman Filter.* University of North Carolina at Chapel Hill, TR95-041

Wiedemann, R. (1974). *Simulation des Straßenverkehrsflusses*, Schriftenreihe des Instituts für Verkehrswesen der Universität Karlsruhe, Heft 8

12

Off-line model predictive control of dc-dc converter

Tadanao Zanma and Nobuhiro Asano
Mie University
Japan

1. Introduction

Control systems with switching modes in which different dynamics are assigned are called hybrid dynamical systems and are being actively researched (1–6). The continuous behavior in the hybrid dynamical system is expressed generally by differential or difference equations, while the discrete behavior is described by logics or state machines such as automata. If a system can be regarded as a hybrid dynamical system, both continuous and discrete properties can be dealt with concurrently. Therefore, a hybrid dynamical system has the ability to represent many systems as a single model without dividing into separate continuous and discrete systems.

Power electronic circuits can also be regarded as hybrid dynamical systems as they share both continuous and discontinuous behaviors(7–14). The continuous behavior of current or voltage in such a system is subject to passive elements such as resistance, capacitance and inductance, whereas the discontinuous element of switching devices such as MOSFETs and IGBTs yields an on-off signal that is essentially discrete.

A conventional method currently being used for the control of dc-dc converters is PWM (Pulse Width Modulation) with triangular wave. The average output voltage is controlled by PWM, which determines on-off switching timing by employing relatively high carrier frequency. However, the reference may vary in the half period of triangular wave carrier if the carrier frequency is lowered to decrease switching loss for saving energy. Then, the average voltage can no longer approximate the voltage reference. One possible reason is that the control frequency is determined by the carrier frequency only. Another reason may be that the PWM method focuses only on the average output characteristic and excludes switching property. Therefore, a novel method is desired for dc-dc converters by considering switching property explicitly as hybrid dynamical systems.

For synthesis of the hybrid dynamical system, various approaches have been proposed. Specifically, modeling and synthesis based on mixed logical dynamical (MLD) systems has much potential since the formulation is similar to the linear discrete time state space representation(19). The solution of the design is obtained by solving an optimization problem with the help of model predictive control (MPC)(16; 17). It derives the optimal input to minimize an estimation of a given cost function by predicting controlled variables for an MLD system. Specifically, the problem is reduced to a mixed-integer linear or quadratic programming (MILP or MIQP) problem. The method is expected to achieve better control performance than that achieved by conventional methods when applied to the output control of a power

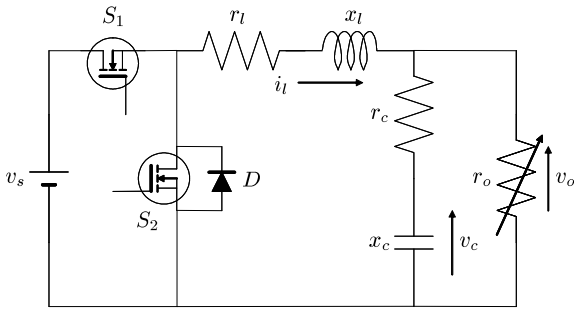


Fig. 1. Topology of the step-down dc-dc converter.

converter. However, it is difficult to solve the optimal problem online because of the computation burden caused by the control period of power converters being considerably short compared to that in mechanical or process control systems.
This paper proposes a control method using the MPC for the output control problem of the dc-dc converter. The considered system is described as an MLD system form. In our work(14), one control period is divided into N submodels. Thus, additional auxiliary variables are needed. In addition, the state variable among the submodels is handled as an averaged one. The method in this paper, however. requires no averaging. The explicit switching law is given as a direct gate signal for the switching devices. Moreover, it is emphasized that a quadratic cost function which was not adopted in a previous work(14) is addressed in this paper so that not only the tracking error but also the switching losses can be considered. The proposed control method achieves quick tracking to the reference in transient state, while keeping the switching frequency as small as possible in steady state. To verify the effectiveness of the proposed method, numerical simulations and experimental results are illustrated.
This paper is organized as follows. In Section 2, a step-down dc-dc converter and MLD system are introduced. Next, the optimization problem for the control is described. Following several simulation results, Section 3 proposes a modified control method taking into account the computation delay. Then, experimental results are shown in Section 4. Finally, Section 5 concludes this paper. In the Appendix, formulation of constraints and transformation to mp-MIQP are explained.

2. Preliminaries

In this section, a step-down dc-dc converter is considered as an example of power electronic circuits. After the formulation, an MLD system(19) and multi-parametric MIQP (mp-MIQP)(18) are reviewed.

2.1 Step-down dc-dc converter

The circuit topology of the step-down dc-dc converter is shown in Fig. 1. The dc-dc converter controls the load voltage v_o with on-off switches S_1 and S_2. The resistance r_o expresses the load. The equivalent series resistance of the capacitor and the internal resistance of the inductor are denoted by r_c and r_l, respectively, while x_l and x_c represent inductance and capacitance of the low-pass filtering stage, respectively. Switches S_1 and S_2 cannot be conducted simultaneously. Together with diode D, switch S_2 provides a path for the inductor current i_l

regardless of whether it is positive or negative. The continuous time state-space representation of the dc-dc converter shown in Fig. 1 is given by

$$\dot{x}(t) = A_c x(t) + B_c u(t), \tag{1}$$

$$y(t) = C_c x(t), \tag{2}$$

where $x(t) = \begin{bmatrix} i_l(t) & v_o(t) \end{bmatrix}'$. Denoted by $i_l(t)$ and $v_o(t)$ are the inductor current and output voltage, respectively. The matrices A_c, B_c and C_c are given by $A_c = \begin{bmatrix} -\frac{r_l}{x_l} & -\frac{1}{x_l} \\ \frac{r_o}{r_c+r_o}(\frac{1}{x_c} - \frac{r_c r_l}{x_l}) & -\frac{r_o}{r_c+r_o}(\frac{1}{x_c r_o} + \frac{r_c}{x_l}) \end{bmatrix}$, $B_c = \begin{bmatrix} \frac{1}{x_l} \\ \frac{r_c r_o}{x_l(r_c+r_o)} \end{bmatrix}$ and $C_c = \begin{bmatrix} 0 & 1 \end{bmatrix}$, respectively. Eqs. (1) and (2) are sampled by T_s. Hereafter, the discrete time is described anew as t. Thus, the considered system is recast as follows.

$$x(t+1) = Ax(t) + Bu(t), \tag{3}$$

$$y(t) = Cx(t), \tag{4}$$

where $A = e^{A_c T_s}$, $B = \int_0^{T_s} e^{A_c \tau} d\tau B_c$ and $C = C_c$. Note that the value of input is limited to either 0 or v_s, which can be rewritten as follows.

$$(\forall t)\ u(t) \in \{0, v_s\}. \tag{5}$$

2.2 Representation by MLD system(19)

A mixed logical dynamical (MLD) system is described by a linear dynamical equation with linear mixed-integer inequalities so that discrete properties included in the process can be introduced into the system using logical variables. One advantage is that the logical formula can be described with linear inequalities and model predictive control can be applied.

The model of the dc-dc converter is rewritten to the MLD system representation. The auxiliary δ of 0-1 variable is introduced as a new input variable to describe the discrete variable. The variable is associated as follows.

$$[\delta(t) = 1] \rightarrow [z(t) = v_s], \tag{6}$$

$$[\delta(t) = 0] \rightarrow [z(t) = 0], \tag{7}$$

where $z(t)$ is,

$$0 \leq z(t) \leq v_s. \tag{8}$$

Eqs. (6) and (7) indicate that $z(t) = v_s$ if $\delta(t) = 1$, whereas $z(t) = 0$, otherwise. By replacing Eqs. (6) and (7) with their equivalent linear inequalities,

$$E_1 \delta(t) + E_2 z(t) \leq E_3 u(t) + E_4 x(t) + E_5, \tag{9}$$

where,

$$E_1 = \begin{bmatrix} 0 & v_s & -v_s & 0 \end{bmatrix}', \tag{10}$$

$$E_2 = \begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix}', \tag{11}$$

$$E_3 = E_4 = O, \tag{12}$$

$$E_5 = \begin{bmatrix} v_s & 0 & 0 & 0 \end{bmatrix}'. \tag{13}$$

is obtained. Inequality (9) reflects that $z(t) = v_s$ if $\delta(t) = 1$ whereas $z(t) = 0$ if $\delta(t) = 0$. Namely, $\delta(t)$ can be considered as the state of the switch: $\delta(t) = 1$ if the switch is on, $\delta(t) = 0$ otherwise. Note that $z(t)$ in inequality (8) is an apparent continuous auxiliary variable.
As a result, Eqs. (3), (4) and (5) can be transformed into an MLD system consisting of one standard linear discrete time state space representation and linear inequalities associated with the constraints on the system,

$$x(t+1) = Ax(t) + Bz(t), \tag{14}$$

$$y(t) = Cx(t), \tag{15}$$

$$\text{subject to Eq. (9).} \tag{16}$$

2.3 Multi-parametric MIQP(18)

Multi-parametric MIQP (mp-MIQP) is a type of MIQP(18) parameterized by multiple parameters. The mp-MIQP parameterized by state x of the system is described as follows.

$$\min_{\nu} \ \nu' H \nu + 2x' F \nu + x' Y x + 2C_f \nu + 2C_x x, \tag{17}$$

$$\text{subject to } G\nu \leq W + Ex, \tag{18}$$

where ν is

$$\nu = \begin{bmatrix} \Delta' & \Xi' \end{bmatrix}', \tag{19}$$

$$\Delta = \begin{bmatrix} \delta_0 & \dots & \delta_{N_p-1} \end{bmatrix}', \tag{20}$$

$$\Xi = \begin{bmatrix} z_0 & \dots & z_{N_p-1} \end{bmatrix}'. \tag{21}$$

In Eqs. (20) and (21), the predictive horizon in MPC is denoted by N_p.
If solved, the optimal solution of mp-MIQP is given as the piece-wise affine state feedback form. Namely, the explicit control law parameterized by the state x is obtained as follows.

$$\nu = K_i x + h_i \quad \text{if } x \in X_i, \tag{22}$$

where X_i ($i = 1, 2, \dots$) are regions partitioned in the state space, and K_i and h_i are the corresponding constant matrices and vectors, respectively. As Eq. (22) is available off-line, the optimal input is determined online according to the state measured at each sampling.

3. Numerical simulation and revision of control method

In this section, the effectiveness of the method proposed in the previous section and the Appendix is shown by applying it to the output control of the dc-dc converter shown in Fig. 1. The control objective is to achieve quick tracking to the reference in transient state with minimal switching in steady state. For the purpose, mp-MIQP is exploited.

3.1 Simulation condition and state partition

The circuit and control parameters for simulation are listed in Tables 1 and 2, respectively.
Let us consider Eqs. (14) to (16) as the model for the dc-dc converter shown in Fig. 1. In Eq. (45), $\tilde{H}$ and L are first set as zeros. Then, the setting of these matrices imply that focus is only on tracking performance. The state partition obtained by off-line model predictive control, (mp-MIQP) and its enlarged view are shown in Fig. 2. In each region of Fig. 2, the optimal input sequence is assigned. The figure of state partition shown in Fig. 2 is generated

Table 1. Circuit parameters

source voltage v_s	5.0 [V]
inductance x_l	20 [μH]
internal resistance r_l	25 [mΩ]
capacitance x_c	2.2 [mF]
equivalent series resistance r_c	60[mΩ]
load resistance r_o	1[Ω]

Table 2. Control parameters

control period T_s	10 [μs]
predictive horizon N_p	1, 3, 5
upper limit $i_{l,\max}$	8.0 [A]
reference value v_{ref}	2.0 [V]

using of Multi-Parametric Toolbox(20). In Fig. 2, the number of state partitions is limited to at most 2^{N_p}. Each partition is specified by linear inequalities. In each partition, the solution of mp-MIQP given by Eq. (22) is assigned. To investigate to which partition it belongs, the state $\begin{bmatrix} i_l & v_o \end{bmatrix}'$ at each sampling can be performed simply since the obtained state partition is constructed by linear inequalities. Focus on the white region at the right bottom corner in Fig. 2. Whenever the state $\begin{bmatrix} i_l & v_o \end{bmatrix}'$ enters the region, switch S_1 shown in Fig. 1 is forced to turn off since the constraint about the inductor current given by Eq. (37) can no longer be satisfied.

3.2 Consideration of delay for computation of state distinction

Figs. 3 and 4 show simulation results for $N_p = 3$ and $N_p = 5$, respectively. Note that the method described in the Appendix is utilized for each of the calculations. Figs. 3 and 4, also indicate that the output voltage is kept at the specified value 2.0 [V] in steady state, while the inductor current does not exceed its limit of 8[A]. In the simulation, the computation time of state distinction for optimal input is assumed to be negligible. Little difference exists between

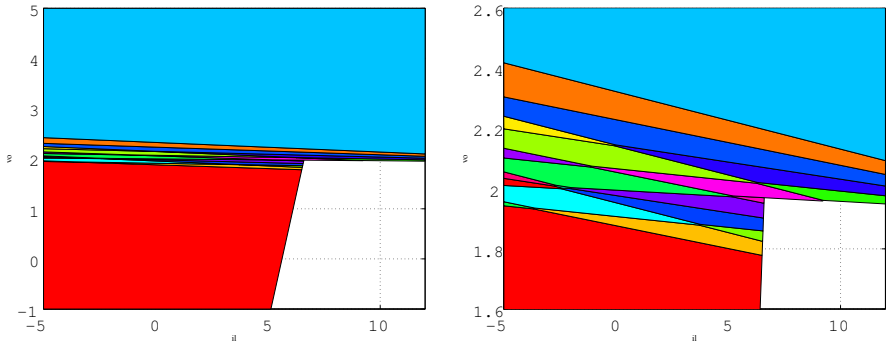


Fig. 2. State partition for $N_p = 5$ (left: whole, rigtht: closeup).

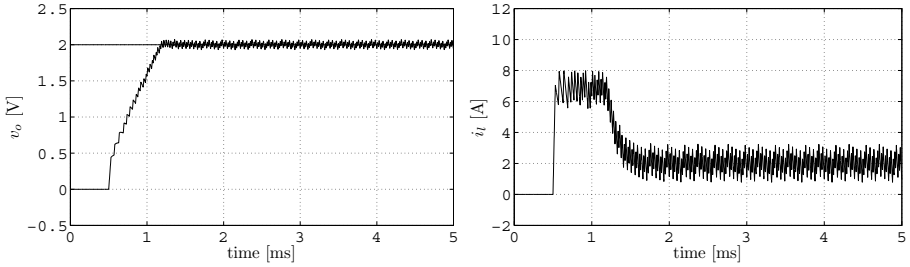


Fig. 3. Simulation result in case computation delay is negligible for $N_p = 3$ (left: v_o, right: i_l).

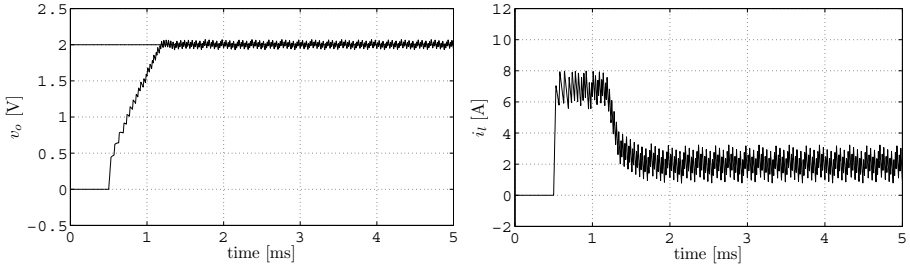


Fig. 4. Simulation result in case computation delay is negligible for $N_p = 5$ (left: v_o, right: i_l).

the two outputs shown in Figs. 3 and 4. In other words, the performance is almost identical for $N_p = 3$ and $N_p = 5$ as long as the computation time is minimal.
On the other hand, as described later in the next section, the computation time should be considered. because of the effects of various factors such as DSP performance and the number of state partitions. In preliminary experiments, 5 [μs] and 8 [μs] for $N_p = 3$ and $N_p = 5$, respectively, are obtained as average computation delay. Using the values, we set the delay for determination of the switching signal after measurement of the state in the simulation. Figs. 5 and 6 illustrate the simulation results under the assumption that the computation delay is not negligible, i.e., the delay is assumed to exist for the computation. From Figs. 5 and 6, the switching intervals that exceed 20[¯s] can be seen. Thus, the ripple effect increases as the

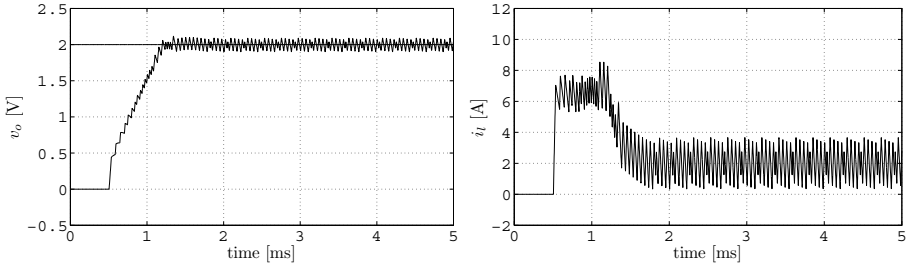


Fig. 5. Simulation result in case computation delay is 5 [μs] for $N_p = 3$ (left: v_o, right: i_l).

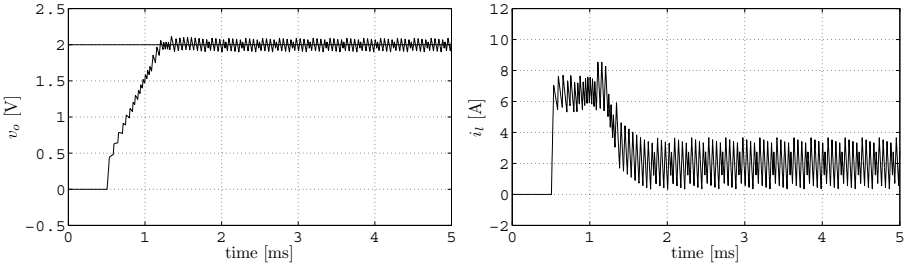


Fig. 6. Simulation result in case computation delay is 8 [μs] for $N_p = 5$ (left: v_o, right: i_l).

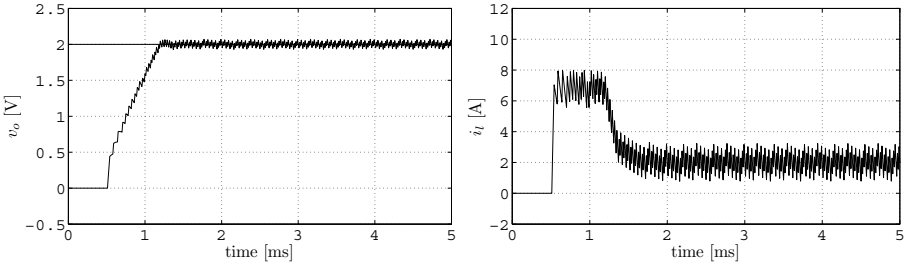


Fig. 7. Simulation result with consideration of computation time for $N_p = 5$ (left: v_o, right: i_l).

difference widens between the value of the measured state and that of the input which is determined after the delay.

3.3 Modification of control method

In the method proposed(21) in the Appendix, input is applied after examination of the region in which the state belongs. However, as mentioned above, the performance is not necessarily satisfactory due to the computation delay even if the horizon is small. Therefore, the control method should be slightly modified in order to consider the computation delay so that performance is not degraded. Specifically, instead of the first one, the second element of the optimal input sequence is applied to the system at the beginning of the next control period. In addition, the first element of the optimal input sequence has to be used as that given at the last sampling. In other words, the first element is not solved but is set as that given at the last period, i.e., in the modified control method, δ_0 and z_0 in Eqs. (20) and (21), respectively, are given in advance as the constants of the last optimized input sequence, not solved as the optimized variables. Note that the modified control method requires $N_p > 1$ due to the structure. Fig. 7 depicts the simulation result by the modified method above mentioned. Compared with Fig. 6, the result shown in Fig. 7 is improved in the sense that the ripple is reduced in steady state.

4. Experimental result

In this section, we show the effectiveness of the modified proposed method(21) through experiments. In addition, the effectiveness for consideration of the switching loss is demonstrated.

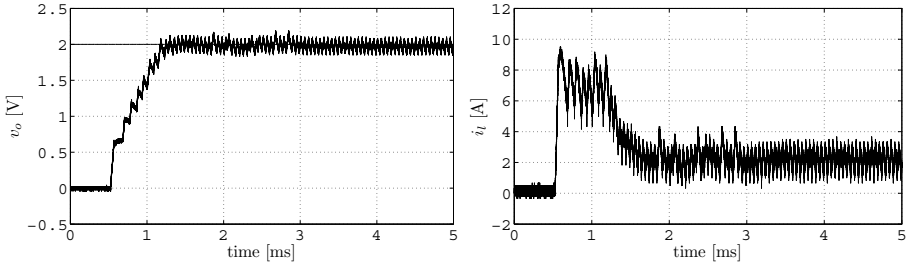


Fig. 8. Experimental result without consideration of computation delay (left: v_o, right: i_l).

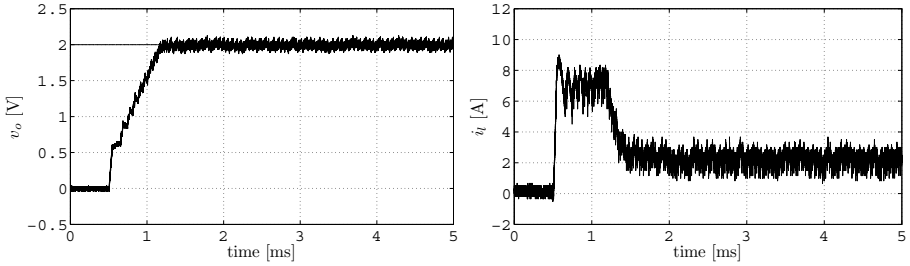


Fig. 9. Experimental result with consideration of computation delay (left: v_o, right: i_l).

The experiments are carried out on a DSP (Texas Instruments TMS3200C/F2812, operating frequency: 150 [MHz], AD-converter: 12 [bit], conversion time: 80 [ns]).

4.1 Comparison of proposed method(21) and its modified method

Fig. 8 shows the experimental result obtained without considering the computation delay for state distinction for $N_p = 5$. Similar to simulation results shown in Fig. 4, many switchings are described with intervals exceeding 20 [μs] although the control period is 10 [μs]. The reason for the results is that the state transits to another which is not the predictive one, due to the computation delay. Therefore, the computation delay for state distinction should be considered in the experiments. Fig. 9 shows the experimental result upon consideration of the computation delay. Note that the results shown in Fig. 9 are obtained by the modified control method mentioned in the previous section.
Compared with the results shown in Fig. 8, the ripple effect is reduced as shown in Fig. 9. This reduction occurs because the computation delay is considered in the latter result. Thus, the effectiveness of the modified control method in Subsection 3.3 is demonstrated.

4.2 Consideration of switching loss

The shorter the control period, the more the switching losses tend to increase, as do the number of switchings. In the proposed method, the switching loss can be considered by incorporating it into the cost function. This can be achieved by setting $Q = qI_{N_p-1}$ where $q = 10^{-3}$ in Eq. (42). The experimental result is shown in Fig. 10. From Fig. 10, the output voltage is tracked to the voltage reference even though the term to reduce switching is added into the cost function. Fig. 10 also shows that the inductor current does not severely exceed the limit

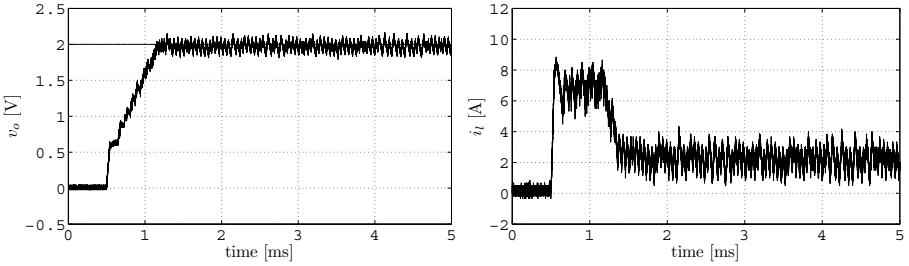


Fig. 10. Experimental result with consideration of computation delay and the switching loss for $N_p = 5$ (left: v_o, right: i_l).

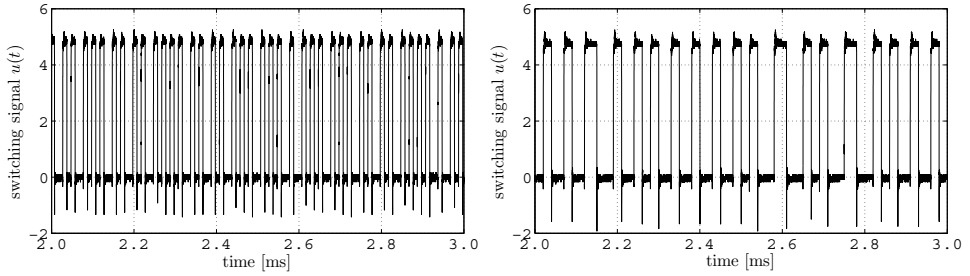


Fig. 11. Experimental result of switching signal without/with consideration of the switching loss for $N_p = 5$ (left: without, that in Fig. 9, right: with, that in Fig. 10).

of 8 [A]. Fig. 11 shows the switching signals for Figs. 9 and 10. From the right of Fig. 11, the switching frequency is reduced by considering the switching loss in the cost function given by Eq. (45). Thus, both tracking performance and switching loss can be considered simultaneously in the proposed method.

5. Conclusions

In this paper, a novel control method for the dc-dc converter has been proposed. The dc-dc converter has been modeled as a mixed logical dynamical (MLD) system since it has the ability to combine continuous and discrete properties. For the control, a model predictive control (MPC) based method has been introduced. The optimization problem has been solved as a multi-parametric off-line programming problem. The result has been obtained as the state space partition which makes the implementation feasible. As a result, computation time is shortened without deteriorating control performance. Finally, it has been demonstrated that the output voltage has been tracked to the reference at the expense of tracking performance by introducing the term to reduce the switching in the cost function. In some cases, other factors such as resistance loss in r_l shown in Fig. 1 may need to be considered, although the cost function given by Eq. (28) considers only the tracking performance and switching loss. Note, however, that the factors represented as linear and/or quadratic forms of the state variable can be incorporated into the cost function.
Further research includes robustness analysis in implementation and investigation of performance for different cost functions as mentioned above.

Acknowledgment

We are grateful to the Okasan-Kato Foundation. We also thank Professor Manfred Morari, Ph.D, Sébastien Mariéthoz, Ph.D, Andrea Beccuti, Ph.D, of ETH Zurich for valuable comments and suggestions.

Here, the proposed method(15) is reviewed in brief.

MIQP derives the values that minimize an estimation of a given cost function under constraints given by inequalities and/or equalities concerning integer variables. The MIQP for Eqs. (14) to (16) is given as follows.

$$\min_{\nu_t} \ \nu_t' S_1 \nu_t + 2(S_2 + x(t)' S_3)\nu_t, \tag{23}$$

$$\text{subject to } F_1 \nu_t \leq F_2 + F_3 x(t), \tag{24}$$

where ν_t is

$$\nu_t = \begin{bmatrix} \Delta_t' & \Xi_t' \end{bmatrix}', \tag{25}$$

$$\Delta_t = \begin{bmatrix} \delta(0|t) & \dots & \delta(N_p - 1|t) \end{bmatrix}', \tag{26}$$

$$\Xi_t = \begin{bmatrix} z(0|t) & \dots & z(N_p - 1|t) \end{bmatrix}'. \tag{27}$$

To derive the optimal input sequence for Eqs. (14) to (16), the following cost function is set.

$$\begin{aligned} J(x(t), \Delta_t, \Xi_t) = &\sum_{k=1}^{N_p} \|y(k|t) - v_{\text{ref}}\|_2^2 \\ &+ \Delta_t' \tilde{H} \Delta_t + 2L\Delta_t, \end{aligned} \tag{28}$$

where v_{ref} denotes the constant voltage reference. In Eq. (28), the first term is associated with the tracking performance whereas the switching loss can be also considered in the second and third terms. Eq. (28) is rewritten as the general MIQP form of Eqs. (23) in order to solve the minimization problem. By Eqs. (14) and (15), $y(k|t)$ which is the predictive output k steps ahead of t is described as follows.

$$\begin{aligned} y(k|t) &= C(A^k x(t) + \sum_{j=0}^{k-1} A^{k-j-1} B z(j)) \\ &= C(A^k x(t) + G_k \Xi_k), \end{aligned} \tag{29}$$

where $G_k = \begin{bmatrix} A^{k-1}B & A^{k-2}B & \dots & B \end{bmatrix}$. By substituting Eq. (29) for Eq. (28), the minimization problem for Eq. (28) is formalized as follows.

$$\begin{aligned} \min_{\Delta_t, \Xi_t} \Bigg(&\sum_{k=1}^{N_p} \Xi_t' G_k' C' C G_k \Xi_t - 2 \sum_{k=1}^{N_p} v_{\text{ref}}' C G_k \Xi_t \\ &+ 2 \sum_{k=1}^{N_p} x(t)' A'^k C' C G_k \Xi_t + \Delta_t' \tilde{H} \Delta_t + 2L\Delta_t \Bigg). \end{aligned} \tag{30}$$

Note that the irrelative terms for the minimization problem are omitted in Eq. (30). Connected with Eq. (23), the optimization problem of Eq. (30) is transformed as

$$\min_{\Delta_t, \Xi_t} \begin{bmatrix} \Delta_t \\ \Xi_t \end{bmatrix}' S_1 \begin{bmatrix} \Delta_t \\ \Xi_t \end{bmatrix} + 2(S_2 + x(t)' S_3) \begin{bmatrix} \Delta_t \\ \Xi_t \end{bmatrix}, \tag{31}$$

where S_1, S_2 and S_3 are,

$$S_1 = \begin{bmatrix} \tilde{H} & O \\ O & \sum_{k=1}^{N_p} G_k' C' C G_k \end{bmatrix} \in \mathbb{R}^{2N_p \times 2N_p}, \tag{32}$$

$$S_2 = \begin{bmatrix} L & -\sum_{k=1}^{N_p} v_{\text{ref}}' C G_k \end{bmatrix} \in \mathbb{R}^{1 \times 2N_p}, \tag{33}$$

$$S_3 = \begin{bmatrix} O & \sum_{k=1}^{N_p} A_k' C' C G_k \end{bmatrix} \in \mathbb{R}^{2 \times 2N_p}, \tag{34}$$

respectively.
Let us rewrite the constraint as the general form like inequality (24). Recall that only two discrete inputs are permitted in the considered system. The constraint represented by Eq. (9) is also transformed as

$$\tilde{F}_1 \begin{bmatrix} \Delta_t \\ \Xi_t \end{bmatrix} \leq \tilde{F}_2 + \tilde{F}_3 x(t), \tag{35}$$

where $\tilde{F}_1$, $\tilde{F}_2$ and $\tilde{F}_3$ are, respectively,

$$\begin{aligned} \tilde{F}_1 &= \begin{bmatrix} E_1 & & O & E_2 & & O \\ & \ddots & & & \ddots & \\ O & & E_1 & O & & E_2 \end{bmatrix} \in \mathbb{R}^{4N_p \times 2N_p}, \\ \tilde{F}_2 &= \begin{bmatrix} E_5 \\ \vdots \\ E_5 \end{bmatrix} \in \mathbb{R}^{4N_p}, \quad \tilde{F}_3 = \begin{bmatrix} E_4 & E_4 \\ \vdots & \vdots \\ E_4 & E_4 \end{bmatrix} \in \mathbb{R}^{4N_p \times 2}. \end{aligned} \tag{36}$$

The constraints imposed on the inductor current limitation is are necessary to prevent damage to the switching device from excessive current. More specifically, if the predictive inductor current at $t+1$, i.e., $i_l(1|t)$, exceeds its limit, $i_{l,\max}$, then the switch is forced to be off. Such an additional condition can be described as

$$[i_l(1|t) > i_{l,\max}] \rightarrow [\delta(0) = 0]. \tag{37}$$

Transformed into the inequality, Eq. (37) is described as

$$i_l(1|t) - i_{l,\max} \leq M(1 - \delta(0)), \tag{38}$$

where M is the admissible upper limit of i_l. Since $x = \begin{bmatrix} i_l & v_o \end{bmatrix}'$, replaced the first row of A and the first element of B with A_1 and b_1, respectively, $i_l(1|t)$ is recast as,

$$i_l(1|t) = \begin{bmatrix} a_{11} & a_{12} \end{bmatrix} x(t) + b_1 z(0), \tag{39}$$

where $\begin{bmatrix} a_{11} & a_{12} \end{bmatrix}$ is the first row of A. Consequently, using Eq. (39), inequality (38) can be expressed as

$$M\delta(0) + b_1 z(0) \leq (M + i_{l,\max}) - \begin{bmatrix} a_{11} & a_{12} \end{bmatrix} x(t). \tag{40}$$

Add Eq. (40) as a new constraint to the last row of Eq. (36), then Eq. (36) is modified as follows.

$$\begin{aligned} F_1 &= \begin{bmatrix} & & & & \tilde{F_1} & & & \\ M & 0 & \dots & 0 & b_1 & 0 & \dots & 0 \end{bmatrix}, \\ F_2 &= \begin{bmatrix} \tilde{F_2} \\ M + i_{l,\max} \end{bmatrix}, \; F_3 = \begin{bmatrix} \tilde{F_3} \\ A_1 \end{bmatrix}. \end{aligned} \tag{41}$$

The switching loss can also be considered in the second and third terms in Eq. (28). In Eq. (28), for example, $L = O$ and $\tilde{H}$ is set with $Q \succeq 0$ as follows.

$$\tilde{H} = (\Pi_1 - \Pi_2)' Q (\Pi_1 - \Pi_2), \tag{42}$$

where Π_1 and Π_2 are, respectively,

$$\Pi_1 = \begin{bmatrix} 0 & \\ \vdots & I_{N_p-1} \\ 0 & \end{bmatrix} \in \mathbb{R}^{(N_p-1)\times N_p}, \tag{43}$$

$$\Pi_2 = \begin{bmatrix} & 0 \\ I_{N_p-1} & \vdots \\ & 0 \end{bmatrix} \in \mathbb{R}^{(N_p-1)\times N_p}. \tag{44}$$

Note that when $\tilde{H}$ and L are set above, the estimation of the cost function of Eq. (28) increases in response to the number of switchings required. Therefore, the switching loss can be reduced depending on Q in Eq. (42).

If the cost function is described, the optimal input sequence can be derived. However, it is impractical to apply it to the considered dc-dc converter with a short control period since the computation requires much solution time for every control period. Then, the method above is transformed into mp-MIQP so that solving the optimization problem on-line is no longer necessary. Eq. (28) is adopted as the cost function again for mp-MIQP. Then, Eq. (28) is described as follows.

$$\begin{aligned} & J(x, \Delta, \Xi) \\ &= \sum_{k=1}^{N_p} \Xi' G_k' C' C G_k \Xi + 2 \sum_{k=1}^{N_p} x' A'^k C' C G_k \Xi \\ &\quad + \sum_{k=1}^{N_p} x' A'^k C' C A^k x - 2 \sum_{k=1}^{N_p} v_{\mathrm{ref}}' C G_k \Xi \\ &\quad - 2 \sum_{k=1}^{N_p} v_{\mathrm{ref}}' C A^k x + \Delta' \tilde{H} \Delta + 2 L \Delta, \end{aligned} \tag{45}$$

where $\Delta = \begin{bmatrix} \delta_0 & \dots & \delta_{N_p-1} \end{bmatrix}$ and $\Xi = \begin{bmatrix} z_0 & \dots & z_{N_p-1} \end{bmatrix}$. Associated with Eq. (17), the optimization problem of Eq. (45) is transformed as follows.

$$\begin{aligned} \min_{\Delta,\Xi} \; & \begin{bmatrix} \Delta \\ \Xi \end{bmatrix}' H \begin{bmatrix} \Delta \\ \Xi \end{bmatrix} + 2x' F \begin{bmatrix} \Delta \\ \Xi \end{bmatrix} + x' Y x \\ & + 2 C_f \begin{bmatrix} \Delta \\ \Xi \end{bmatrix} + 2 C_x x, \end{aligned} \tag{46}$$

where $\tilde{H} = S_1$, $F = S_3$ and $C_f = S_2$, respectively. Note that there exists a clear difference between notations of ν_t and ν. The former is utilized for MIQP while the latter is used for mp-MIQP. The others are

$$Y = \sum_{k=1}^{N_p} A'^k C' C A^k, \tag{47}$$

$$C_x = -\sum_{k=1}^{N_p} v'_{\text{ref}} C A^k. \tag{48}$$

The constraints are given by

$$F_1 \begin{bmatrix} \Delta \\ \Xi \end{bmatrix} \leq F_2 + F_3 x. \tag{49}$$

Transformed as above, the optimization problem is solved offline as mp-MIQP. Then, the result is employed for on-line control.

6. References

[1] *Hybrid systems* I, II, III, IV, V, Lecture Notes in Computer Science, 736, 999, 1066, 1273, 1567, New York, Springer-Verlag, 1993 to 1998.

[2] "Special issue on hybrid control systems," *IEEE Trans. Automatic Control*, Vol. 43, No. 4, 1998.

[3] "Special issue on hybrid systems," *Automatica*, Vol. 35, No. 3, 1999.

[4] "Special issue on hybrid systems," *Systems & Control Letters*, Vol. 38, No. 3, 1999.

[5] "Special issue hybrid systems: Theory & applications", *Proc. IEEE*, Vol. 88, No. 7, 2000.

[6] T. Ushio, "Expectations for Hybrid Systems," *Systems, Control and Information*, Vol. 46, No. 3, pp. 105–109, 2002.

[7] S. Almer, H. Fujioka, U. Jonsson, C. Y. Kao, D. Patino, P. Riedinger, T. Geyer, A. G. Beccuti, G. Papafotiou, M. Morari, A. Wernrud and A. Rantzer, "Hybrid Control Techniques for Switched-Mode DC-DC Converters Part I: The Step-Down Topology," *Proc. ACC* , pp. 5450–5457, 2007.

[8] A. G. Beccuti, G. Papafotiou, M. Morari, S. Almer, H. Fujioka, U. Jonsson, C. Y. Kao, A. Wernrud, A. Rantzer, M. Baja, H. Cormerais, and J. Buisson, "Hybrid Control Techniques for Switched-Mode DC-DC Converters Part II: The Step-Up Topology," *Proc. ACC*, pp. 5464–5471, 2007.

[9] A. G. Beccuti, G. Papafotiou, R. Frasca and M. Morari, "Explicit Hybrid Model Predictive Control of the dc-dc Boost Converter," *Proc. IEEE PESC*, pp. 2503–2509, 2007.

[10] I. A. Fotiou, A. G. Beccuti and M. Morari, "An Optimal Control Application in Power Electronics Using Algebraic Geometry," *Proc. ECC*, pages 475–482, July 2007.

[11] R. R. Negenborn, A. G. Beccuti, T. Demiray, S. Leirens, G. Damm, B. D. Schutter and M. Morari, "Supervisory Hybrid Model Predictive Control for Voltage Stability of Power Networks," *Proc. ACC*, pp. 5444–5449, 2007.

[12] A. G. Beccuti, G. Papafotiou and M. Morari, "Optimal control of the buck dc-dc converter operating in both the continuous and discontinuous conduction regimes," *Proc. IEEE CDC*, pp. 6205–6210, 2006.

[13] T. Geyer, G. Papafotiou, M. Morari, “On the Optimal Control of Switch-Mode DC-DC Converters," *Hybrid Systems: Computation and Control*, Vol. 2993, pp. 342–356, Lecture Notes in Computer Science, 2004.

[14] G. Papafotiou, T. Geyer, M. Morari, “Hybrid Modelling and Optimal Control of Switch-mode DC-DC Converters," *IEEE Workshop on Computers in Power Electronics (COMPEL)*, pp. 148–155, 2004.

[15] K. Asano, K. Tsuda, A. Bemporad, M. Morari, “Predictive Control for Hybrid Systems and Its Application to Process Control," *Systems, Control and Information*, Vol. 46, No. 3, pp. 110–119, 2002.

[16] M. Ohshima, M. Ogawa, “Model Predictive Control –I– Basic Principle: history & present status," *Systems, Control and Information*, Vol. 46, No. 5, pp. 286–293, 2002.

[17] M. Fujita, M. Ohshima, “Model Predictive Control –VI– Model Predictive Control for Hybrid Systems," *Systems, Control and Information*, Vol. 47, No. 3, pp. 146–152, 2003.

[18] F. Borrelli, M. Baotic, A. Bemporad, M. Morari, “An efficient algorithm for computing the state feedback optimal control law for discrete time hybrid systems," In *Proc. ACC*, pp. 4717–4722, 2003.

[19] A. Bemporad, M. Morari, “Control of systems integrating logic, dynamics, and constraints," *Automatica*, Vol. 35, No. 3, pp. 407–427, 1999.

[20] M. Kvasnica, P. Grieder, M. Boatić and F. J. Christophersen, “Multi-Parametric Toolbox (MPT)," Institut für Automatik, 2005.

[21] N. Asano, T. Zanma and M. Ishida, “Optimal Control of DC-DC Converter using Mixed Logical Dynamical System Model," *IEEJ Trans. IA*, Vol. 127, No. 3, pp. 339–346, 2007.

13

Nonlinear Predictive Control of Semi-Active Landing Gear

Dongsu Wu, Hongbin Gu and Hui Liu
Nanjing University of Aeronautics and Astronautics
China

1. Introduction

When airplane touches down and taxis on uneven runways with high speed, there is heavy ground impact and huge vertical load to the airframe. To improve safety and make passengers comfortable during landing, an effective landing gear capable of absorbing impact energy as much as possible is indispensable for modern airplane. Besides the basic function of reducing impact loads, the landing gear must also allow sufficient maneuverability during ground operation, which leads to conflicting requirements in terms of the suspension system (Krüger, 2000). Traditional landing gear consists of tires and passive shock absorbers, which can only be optimized before leaving factory to ensure the landing gear having a fairly good performance in particular design operational conditions, typically hard landings. However, due to its fixed structure, the passive shock absorber cannot always work well on various ground conditions and operational conditions. A heavy landing or a coarse runway may lead to significant deterioration of its performance, which is harmful to the fatigue life of the landing gear and of the airframe.

Active control and semi-active control are widely used approach in the field of construction vibration control and vehicle suspension control. Compared with passive control, active and semi-active control has excellent tunable ability due to their flexible structure. Active control needs an external hydraulic source to supply energy for the system. The main drawback of active control approach is that its structure is very complex and the external energy may lead to instability of the system. The semi-active approach (Fig.1) modifies the damping characteristics by changing the size of the orifice area and does not introduce any external energy. Studies by Karnopp (Karnopp, 1983) for automotive applications also suggest that the efficiency of semi-active dampers is only marginally lower than of a fully active system, provided that a suitable control concept is used. In consideration of its simple structure and high reliability, semi-active control approach could be a better choice for landing gear systems.

The main component of semi-active landing gear system is a tunable oleo-pneumatic shock absorber, which contains multidisciplinary and highly nonlinear dynamics. It is not an easy task to design an effective controller for such complex system. Krüger (Krüger, 2000) focuses his studies on optimization of taxiing performance of a semi-active landing gear. SIMPACK software is used to run simulation with a complete aircraft FEA model. Ghiringhelli builds a

complete aircraft landing simulation model in ADAMS software (Ghiringhelli et al., 2004). A semi-active PID control method is used to control the orifice area. His also studies sensitivity of the complete aircraft model to the variation of control parameters and compares the results obtained in the simulated drop tests between passive and semi-active approach (Ghiringhelli, 2000). Maemori et al. (Maemori et al., 2003) proposes an optimization method for a semi-active landing gear to handle variations in the maximum vertical acceleration of an aircraft during landing caused by the variation of the aircraft mass, which is always due to the variations in the number of passengers, and the amounts of cargo and fuel. Wang et al. (Wang et al., 1999) considers both taxiing and landing conditions. He uses a fuzzy controller to optimize the performance of the semi-active landing gear. But he does not consider the dynamics of the actuator. Mikulowski et al. (Mikulowski et al., 2008) discuss the application of piezo-actuators and magneto-rheological damper in the adaptive landing gear design. And there are some other researchers applying ER (Lou et al., 1993) or MR (Batterbee et al., 2007) technology in semi-active land gear system. All of the semi-active controllers designed above do not consider the actuator saturations (limited control amplitude and rate), which may lead to significant, undesirable deterioration in the closed-loop performance and even closed-loop instability.

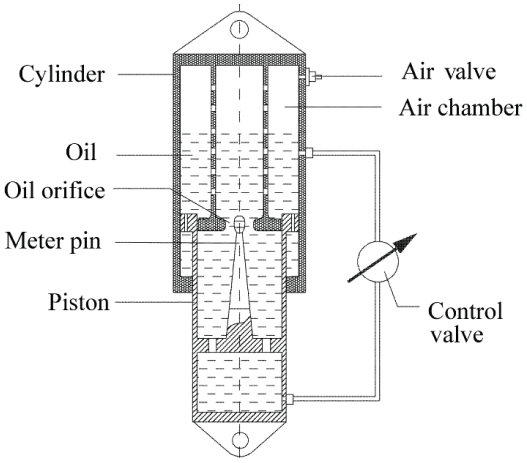


Fig. 1. Structure of Semi-Active Controlled Shock Absorber

Model predictive control refers to a class of control algorithms in which a dynamic model is used to predict and optimize control performance. The predictions are obtained from a dynamic model and the optimization problem is solved subject to constraints on input and output variables. So MPC is especially suited for constrained, digital control problems. Initially MPC has been widely used in the industrial processes with linear models, but recently some researchers have tried to apply MPC to other fields like automotive (Mehra et al., 1997) and aerospace (Hyochoong et al., 2004), and the nonlinear model is used instead of linear one due to the increasingly high demands on better control performance and rapidly developed powerful computing systems (Michael et al., 1998). To the semi-active landing gear control problem, the nonlinear model predictive control is a good choice considering its effectiveness to constrained control problems and continuously optimized performance. The goal of this paper is to introduce the design and the analysis of a nonlinear hierarchical

control strategy, for semi-active landing gear systems in civil and military aircrafts, based on predictive control strategies.

2. Dynamic Model of Semi-Active Landing Gear

The structure mass of landing gear is divided into sprung mass and non-sprung mass. Sprung mass defined in the figure includes the airframe, the cylinder etc. Non-sprung mass includes the piston rod, wheel etc. The tire is modelled as a simple spring and the tunable damping is realized by a variable size orifice which is controlled by a high-speed solenoid valve.

The governing dynamic equations of semi-active landing gear can be presented as the following:

$$m_s \ddot{z}_s = m_s g - F \tag{1}$$

$$m_u \ddot{z}_u = m_u g + F - P \tag{2}$$

Where mu is the unsprung mass, ms the sprung mass, z_u the displacement of unsprung part, z_s the displacement of sprung part, P the vertical force on the tire, F the semi-active damper shock strut force.

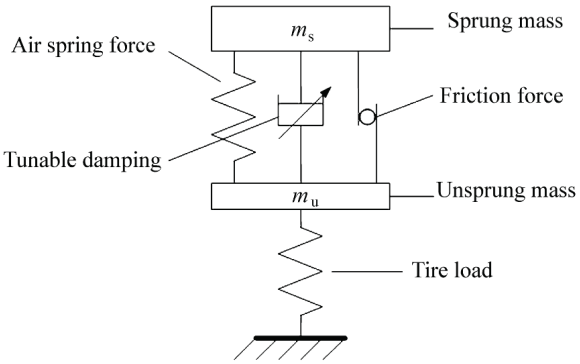


Fig. 2. System Model of Semi-Active Landing Gear

2.1 Shock Struct Force Model

Considering basics of the shock strut operation, a damping effect is produced by squeezing the compressed oil through the tunable orifice. In the pneumatic chamber, the enclosed air is compressed by the movement of the piston, which provides an air cushion spring. There is also friction produced between sliding parts. All these forces comprise the shock strut force (Yadav et al., 1991):

Oleo damping force:

$$F_{oil} = \frac{\rho A_o^{\ 3} sgn(\dot{z}_s - \dot{z}_u)}{2(A_d C_d)^2 \sqrt{1-(A_d / A_o)^2}} (\dot{z}_s - \dot{z}_u)^2 \tag{3}$$

Air spring force:

$$F_{air} = P_i A_a [\frac{V_0}{V_0 - A_a(z_s - z_u)}]^n - P_0 A_a \tag{4}$$

Friction force:

$$F_f = K_m F_{air} \tag{5}$$

Total axial force in the shock strut:

$$F = F_{air} + F_f + F_{oil} \tag{6}$$

where ρ is the oil density, P_i is the initial pneumatic pressure of air chamber, P_i is the atmospheric pressure, A_0 is the effective oil action area, A_a is the effective air action area, A_d is the tunable oil orifice area, C_d is the tunable oil orifice flow coefficient, V_0 is the initial volume of air chamber, K_m is the coefficient of kinetic friction.

2.2 Tire Force Model

The vertical force P on the tire is due to polytropic compression of air inside the tire. In order to simplify the mathematical model, the tire is treated as a linear spring here:

$$P = K_t z_u + C_t \dot{z}_u \tag{7}$$

where K_t is the stiffness coefficient of tire, and C_t the damp coefficient of tire.

2.3 Model of High Speed Solenoid Valve

Traditional solenoid valve (Fig.3) are simple in construction, rugged, relatively cheap to produce and have higher power-mass ratio, but they are not usually used for continuous and proportional control due to its high nonlinearity. Recently, some attempts are made in this kind of application using nonlinear control methods. According to our previous studies (Liu H. et al, 2008), we model a high speed solenoid valve by considering its mechanical, magnetic and electrical dynamics.

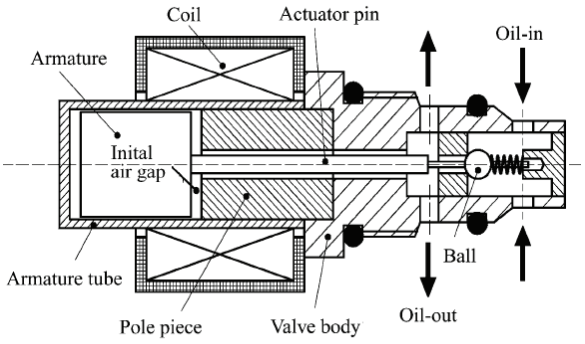


Fig. 3. High-speed Solenoid Valve's Structure

(a) The mechanical dynamics of solenoid valve can be expressed as below:

$$m_v\ddot{x}_v + C_s\dot{x}_v + (K_s + K_f)x_v + f_0 + f = F_v \tag{8}$$

where m_v is the total mass of movable parts including armature, actuator pin, etc., C_s viscous damping coefficient, K_s spring stiffness, K_f static flow coefficient, f_0 preloading force of spring, f Coulomb friction, and x_v movable part displacement and is proportional to oil orifice area A_d.

$$A_d = K_v x_v \tag{9}$$

where K_v is the proportionality coefficient.

(b) The magnetic dynamics of solenoid valve can be summarized as following:
The magnetomotive force is

$$\varepsilon_m = Ni = \Phi R_m \tag{10}$$

where N is the coil turns, i current, R_m total magnetic reluctance, and Φ total magnetic flux.
The electromagnetic force that acts on the armature of valve can be given by

$$F_v = \frac{\Phi_{air}^2}{2\pi\mu_0 r^2 \lambda^2} \tag{11}$$

where μ0 is the air permeability, r the radius of armature, λ the leakage coefficient of the main air gap, and Φ_{air} magnetic flux passing through the working air gap.

$$\Phi_{air} = \frac{R_o R_L}{R_o + R_L}\Phi \tag{12}$$

Ro and RL are corresponding to the magnetic reluctance of two part of magnetic flux paths. Due to the fact that λ, R_o, R_L, R_m are related to x_v, and according to Eq. (10-12), the magnetic equations of solenoid valve can be simplified as:

$$F_v = B(x_v)i^2 \tag{13}$$

where $B(x_v)$ is a function of x_v and represents nonlinear magnetic dynamics of valve. F_v depends on i^2, the square of electrical current. i is the control input for solenoid valve.

(b) Solenoid valve is also characterized by the electric equation:

$$V = Ri + i\frac{dL(x_v, i)}{dt} + L(x_v, i)\frac{di}{dt} \tag{14}$$

From the above equation, we can see that an inner loop to control current can be introduced to improve current input accuracy. According to (Malaguti et al., 2002), mechanical dynamics of solenoid valve is slow respect to electric one, so we obtain the simple electric equation.

$$V = Ri + L(x_{v0})\frac{di}{dt} \tag{15}$$

The inductance is supposed constant in the operating position and independent on current. And specific values of valve's parameters can be found in (Liu H. et al., 2008).

2.4 Full State Mode

Assigning the states as $x_1 = z_s - z_u$, $x_2 = \dot{z}_s - \dot{z}_u$, $x_3 = z_u$, $x_4 = \dot{z}_u$, $x_5 = x_v$, $x_6 = \dot{x}_v$, $x_7 = i$, and combining all the equations we obtain the full state model.

$$\dot{x}_1 = x_2; \tag{16}$$

$$\dot{x}_2 = \frac{1}{m_u}(K_t x_3 + C_t x_4) - \frac{m_s + m_u}{m_s m_u}F; \tag{17}$$

$$\dot{x}_3 = x_4; \tag{18}$$

$$\dot{x}_4 = g + \frac{1}{m_u}[F - K_t x_3 - C_t x_4]; \tag{19}$$

$$\dot{x}_5 = x_6; \tag{20}$$

$$\dot{x}_6 = \frac{1}{m_v}[B(x_5)x_7^2 - C_s x_6 - (K_s + K_f)x_5 - f - f_0]; \tag{21}$$

$$\dot{x}_7 = \frac{1}{L}(V - Rx_7); \tag{22}$$

where F, Fair and Foil can be expressed as following,

$$F = (1 + K_m)F_{air} + F_{oil} \tag{23}$$

$$F_{air} = P_i A_a (\frac{V_0}{V_0 - A_a x_1})^n - P_0 A_a \tag{24}$$

$$F_{oil} = \frac{\rho A_o{}^3 sgn(x_2)}{2(K_v x_5 C_d)^2 \sqrt{1-(K_v x_5 / A_o)^2}} x_2{}^2 \tag{25}$$

3. Design Objective of Oleopneumatic Shock Absorber in Landing Gear System

The tasks of aircraft landing gears are complex and lead to a number of sometimes contradictory requirements. At touchdown, the landing gear has to perform its task of absorbing the aircraft vertical energy via the shock absorber and the horizontal energy by the brakes. At taxiing, the landing gear has to carry the aircraft over taxiways and runways of varying quality. The requirements for absorption of a hard touch-down and for comfortable rolling lead to a design conflict.

3.1 Touchdown Phase

At touchdown phase, the design objective of oleo-pneumatic shock absorber is aimed at reducing the maximum vertical load level introduced at the fuselage attachment and producing a possibly "balanced" set of landing structural loads at touchdown. To get optimal structural load, the impact energy should be equally distributed with respect to the shock absorber stroke. So the optimal structural load during touchdown is a constant value:

$$F_{sao} = \frac{\int_0^{z_{cc}} F_{sa}(z)dz}{z_0} \tag{26}$$

F_{sao} can be estimated by the total energy to be absorbed at touchdown, including kinetic energy and potential energy in vertical direction, and the expected stroke of shock absorber which is generally 90%-95%of the maximal stroke (the work done by drag and lift are omitted).

It is hard for a conventional passive landing gear system to achieve this optimal target load. Semi-active landing gear system has a better performance due to its flexible structure, and is possible to reach the ideal effect if a suitable control method is used. Actually, stroke z_1 is needed to travel before structural load reaches F_{sao}, and this part of the gear compression cannot overly reduced (Ghiringhelli et al., 2004). If z_1 is too short, the gear stiffness will be large and thus the longitudinal spin-up loads will increase sharply. That will lead to the reduction of unitary efficiency. So a reasonable choice is to use passive control till the structural load reaches F_{sao}, and then change to semi-active control afterwards. That results F_{sas}, a sub-optimal structural load solution. By using this scheme, the unitary efficiency of a landing gear system can be achieved though the efficiency of the shock absorber is decreased.

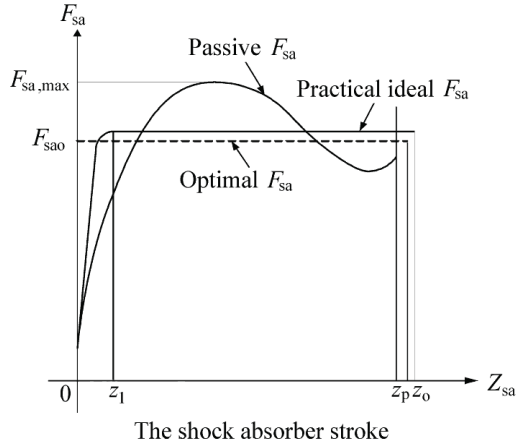


Fig. 4. Shock Absorber Efficiency Comparison

3.2 Taxiing Phase

At taxiing phase, the design objective of oleo-pneumatic shock absorber is aimed at filtering the unevenness of runway surface and providing a comfortable ground ride. It is expected that an aircraft rapidly returns to its original equilibrium state and have minimum vertical displacement when influenced by a runway excitation such as bump or cave. So the maximum vertical displacement of airframe over a test runway is an important criterion for shock absorber design. Another design criterion is root-mean-square (RMS) of airframe vertical acceleration by reason that ground induced vibrations become more and more of a problem as structures of modern aircraft become increasingly flexible. That will lead to shorten the fatigue life of the landing gear and of the airframe.
The RMS of acceleration is defined as follows

$$C_{RMS} = \sqrt{\frac{1}{t_e - t_0}\int_{t_0}^{t_e} (\ddot{z}_s - \ddot{z}_{sr})^2 dt} \tag{27}$$

with airframe's vertical acceleration $\ddot{z}_s$ and reference value for acceleration evaluation $\ddot{z}_{sr}$.

3.3 Transition From Touchdown to Taxiing

The damping required to successfully encounter oscillations has to considerably larger for taxiing than for touchdown because the oleo stroke velocity at taxiing is significantly smaller than at touchdown. So there exists a transition between touchdown phase and taxiing phase. For passive landing gear, a standard solution is the use of a double-stage air spring or a taxi valve. At low stroke velocities (taxiing), high damping factor is achieved, while at high stroke velocity (touchdown), the valve reduces its damping factor. For our semi-active landing gear, the transition from touchdown phase to taxiing phase is also monitored by

measurement of stroke velocity and the damping factor of the shock absorber is changed by variable-sized oil orifice.

3.4 Dual Mode Controller

Due to totally different design goal of landing gear during aircraft touchdown phase and taxiing phase, the semi-active controller should be able to switch from one mode to another. Thus a dual mode predictive controller will be proposed in the following sections. Fig.5 shows the structure of the dual mode controller.

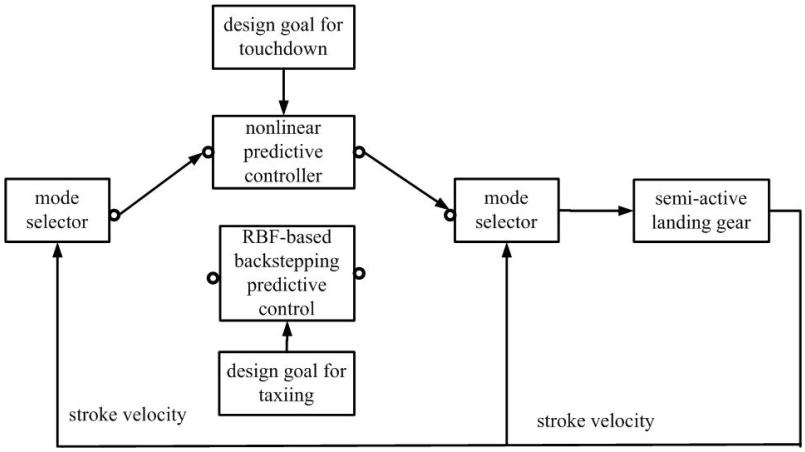


Fig. 5. Controller Switching Between Touchdown Phase and Taxiing Phase

4. Semi-Active Predictive Controller Design for Touchdown Phase

It is noted that the hydraulic dynamics, pneumatic dynamics and fast valve dynamics make controls design very difficult. In order to achieve the ideal objective, a proper semi-active control method should be applied. Considering the highly nonlinear behaviour of landing gear, the classical linear control theory will be useless. The advances of nonlinear control theory make it possible to transform certain types of nonlinear systems to linear system (Slotine et al., 1991).

4.1 Inverse Dynamics Controller

The semi-active landing gear dynamic model (eq.16-25) can be simplified as a following SISO nonlinear system:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})u \tag{28}$$

$$y = \boldsymbol{c}(\boldsymbol{x}) \tag{29}$$

Where, u is the system input which stands for actuator's driving voltage V, y is the system output which stands for the shock absorber force F.

To deal with strong nonlinearities, generally an input-output linearization can be adopted during the system synthesis process. The basic approach of input-output linearization is simply differentiating the output function y repeatedly until the input u appears, and then designing u to cancel the nonlinearity (Slotine et al., 1991). However, the nonlinearity cancelling can not be carried out here because the relative degree of the semi-active landing gear system is undefined,

Since the semi-active landing gear dynamic model consists of shock absorber's model and high-speed solenoid valve's model, we propose a cascade nonlinear inverse dynamics controller. First, an expected oil orifice area A_d for the shock absorber is directly computed by inversion of nonlinear model if control valve's limited magnitude and rate are omitted,

$$A_d = \sqrt{\frac{\rho A_0^3}{2x_2^2 C_d^2 (F_{sao} - F_{air} - K_m F_{air})}} \tag{30}$$

Then a nonlinear tracking controller for high-speed solenoid valve can be designed to follow the expected movable parts position of solenoid valve. However, the practical actuator has magnitude and rate limitations. The maximum adjustable open area of the valve is $7.4mm^2$ and switch frequency is $100Hz$. So the optimal performance is not achievable.

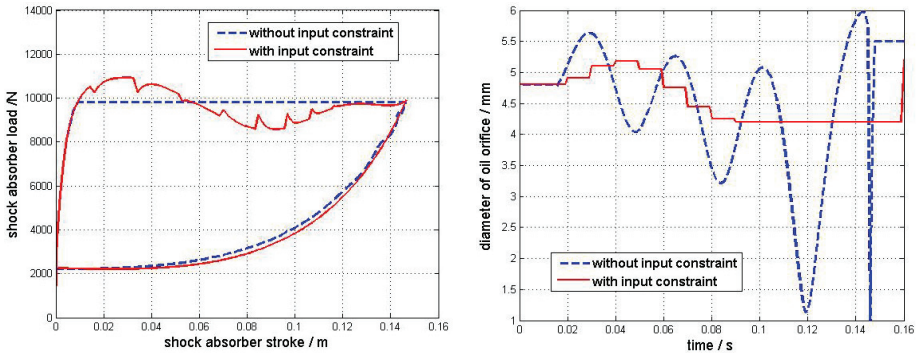


Fig. 6. Shock Absorber Efficiency and Control Input Comparison w/o Input Constraints

From the above figures, we can see that the high-speed solenoid valve's limited rate and magnitude have negative effects on the shock absorber if those input constraints are not considered during the controller synthesis process.

4.2 Nonlinear Predictive Controller

Model predictive control (MPC) is suitable for constrained, digital control problems. Initially MPC has been widely used in the industrial processes with linear models, but recently some researchers have tried to apply MPC to other fields like automotive and aerospace, and the nonlinear model is used instead of linear one due to the increasingly high demands on better control performance. However, optimization is a difficult task for nonlinear model predictive control (NMPC) problem. Generally a standard nonlinear programming method such as SQP is used. But it is the non-convex optimization method for constrained nonlinear

problem, thus global optimum can not be obtained. Furthermore, due to its high computational requirement, SQP method is not suitable for online optimization.

To the semi-active landing gear control problem, a nonlinear output-tracking predictive control approach (Lu, 1998) is adopted here considering its effectiveness to constrained control problems and real-time performance. The basic principle of this control approach is to get a nonlinear feedback control law by solving an approximate receding-horizon control problem via a multi-step predictive control formulation.

The nonlinear state equation and output equation are defined by eq. (28-29). And the following receding-horizon problem can be set up for providing the output-tracking control:

$$\min_{u} \boldsymbol{J}[\boldsymbol{x}(t), t, u] = \min_{u} \frac{1}{2} \int_{t}^{t+T} [e^{T}(\tau)\boldsymbol{Q}e(\tau) + u^{T}(\tau)\boldsymbol{R}u(\tau)]d\tau \tag{31}$$

subject to the state equations (28) and

$$e(t+T) = 0 \tag{32}$$

where $e(t) = y(t) - y_d(t)$.

Then we shall approximate the above receding-horizon control problem by the following multi-step-ahead predictive control formulation. Define $h = T/N$, with N is control number during the prediction horizon. The output $y(t+kh)$ is approximated by the first-order Taylor series expansion

$$y(t+kh) \approx y(t) + \boldsymbol{C}[\boldsymbol{x}(t)]\{\boldsymbol{x}(t+kh) - \boldsymbol{x}(t)\}, \ 1 \le k \le N \tag{33}$$

where $\boldsymbol{C} = \partial \boldsymbol{c}(\boldsymbol{x}) / \partial \boldsymbol{x}$. The desired output $y_d(t+kh)$ is predicted similarly by recursive first-order Taylor series expansions

$$y_d(t+h) \approx y_d(t) + h\dot{y}_d(t)$$

$$y_d(t+2h) \approx y_d(t+h) + h\dot{y}_d(t+h) \approx y_d(t) + h\dot{y}_d(t) + h[\dot{y}_d(t) + h\ddot{y}_d(t)]$$

where another first-order expansion $\dot{y}_d(t+h) \approx \dot{y}_d(t) + h\ddot{y}_d(t)$, then we have

$$y_d(t+kh) \approx y_d(t) + h[\sum_{i=0}^{k-1} (1+hp)^i p y_d(t)] \tag{34}$$

where $p = d/dt$ is the differentiation operator. Combining the predictions of $y(t+kh)$ and $y_d(t+kh)$, we obtain the prediction of the tracking error

$$e(t+kh) = y(t+kh) - y_d(t+kh) \approx h\{\sum_{i=0}^{k-1}[\boldsymbol{C}(\boldsymbol{I}+h\boldsymbol{F})^i]\boldsymbol{f} + \sum_{i=0}^{k-1}[C(I+h\boldsymbol{F})^i \boldsymbol{g}u(t+(k-1-i)h) - (1+hp)^i py_d(t)]\} \tag{35}$$

where $\boldsymbol{F}(\boldsymbol{x}) = \partial \boldsymbol{f}(\boldsymbol{x})/\partial \boldsymbol{x}$. Approximating the cost function by the trapezoidal rule, it can be written as a quadratic function

$$\bar{\boldsymbol{J}} = \frac{1}{2}\boldsymbol{v}^T \boldsymbol{H}(\boldsymbol{x})\boldsymbol{v} + \boldsymbol{r}^T(\boldsymbol{x})\boldsymbol{v} + \boldsymbol{q}(e, \boldsymbol{x}, y_d) \tag{36}$$

where $\boldsymbol{v} = col\{u(t), u(t+h), ..., u[t+(N+1)h]\}$.
The constraint (eq. (32)) is then expressed as $e(t+Nh) = 0$ which leads to

$$\boldsymbol{M}^T(\boldsymbol{x})\boldsymbol{v} = \boldsymbol{d}(e, \boldsymbol{x}, y_d) \tag{37}$$

where

$$\boldsymbol{M}^T = \boldsymbol{C}[(\boldsymbol{I}+h\boldsymbol{F})^{N-1}\boldsymbol{g}, \ldots, (\boldsymbol{I}+h\boldsymbol{F})\boldsymbol{g}, \boldsymbol{g}] \tag{38}$$

$$\boldsymbol{d} = -\frac{1}{h}e - \sum_{i=0}^{N-1}[\boldsymbol{C}(\boldsymbol{I}+h\boldsymbol{F})^i \boldsymbol{f} - (1+hp)^i py_d(t)] \tag{39}$$

Now the output-tracking receding-horizon optimal control problem is reduced to the problem of minimizing $\bar{\boldsymbol{J}}$ with respect to $\boldsymbol{v}$ subject to eq. (37), which is a quadratic programming problem. The closed-form optimal solution for this problem is

$$\boldsymbol{v} = -[\boldsymbol{H}^{-1} - \boldsymbol{H}^{-1}\boldsymbol{M}(\boldsymbol{M}^T\boldsymbol{H}^{-1}\boldsymbol{M})^{-1}\boldsymbol{M}^T\boldsymbol{H}^{-1}]\boldsymbol{r} + [\boldsymbol{H}^{-1}\boldsymbol{M}(\boldsymbol{M}^T\boldsymbol{H}^{-1}\boldsymbol{M})^{-1}]\boldsymbol{d} \tag{40}$$

Then the closed-loop nonlinear predictive output-tracking control law is

$$u(t; \boldsymbol{x}, N) = \boldsymbol{v}(1) \tag{41}$$

Unlike the input-output feedback linearization control laws, the existence of the proposed nonlinear predictive output-tracking control does not depend on the requirement that the system have a relative degree. And more important, the actuator's amplitude and rate constraints can be taken into account during the controller synthesis process.

4.3 Numerical Simulation

Based on the analysis described in previous sections, the numerical simulation of the semi-active landing gear system responses are derived using MATLAB environment. The prototype of the simulation model is a semi-active landing gear comprehensive

experimental platform we built, which can be reconfigured to accomplish tasks such as drop tests, taxi tests and shimmy tests. The sprung mass of this system is 405kg and the unsprung mass is 15kg. The other parameters of the simulation model can be found in (Wu et al, 2007). Fig.7 is the photo of the experiment system.

Fig. 7. Landing gear experiment platform

Three kinds of control methods including passive control, inverse dynamics semi-active control and nonlinear predictive semi-active control are used in the computer simulation. The fixed size of oil orifice for passive control is optimized manually under following parameters: sinking speed is 2 m/s and aircraft sprung mass is 405 kg. In the process of simulation, the sprung mass remains constant and the comparison is taken in terms of different sinking speed: 1.5 m/s, 2 m/s and 2.5 m/s. For passive control, the orifice size is fixed. From the Figs. 8-10 and Table 1, when system parameters such as sinking speed change, the control performance of the passive control decreases greatly, for the fixed orifice size in passive control is designed under standard condition.

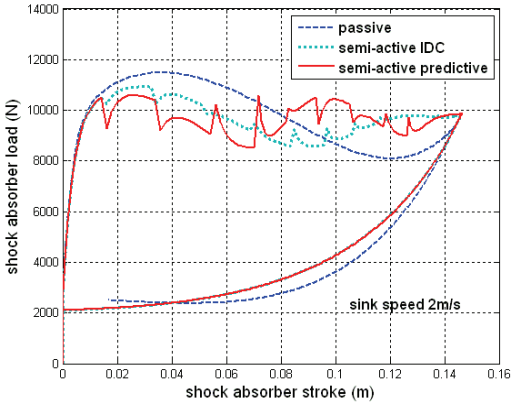


Fig. 8. Efficiency Comparison under Normal Condition

Conventional passive landing gear is especially optimized for heavy landing load condition, so the passive landing gear behaves even worse under light landing load condition. The performance of semi-active control is superior to that of passive one due to its tunable orifice size and nonlinear predictive semi-active control method has the best performance of all. Due to its continuous online compensation and consideration of actuator's constraints, nonlinear predictive semi-active control method can both increase the efficiency of shock absorber and make the output smoother during the control interval, which can effectively alleviate the fatigue damage of both airframe and landing gear.

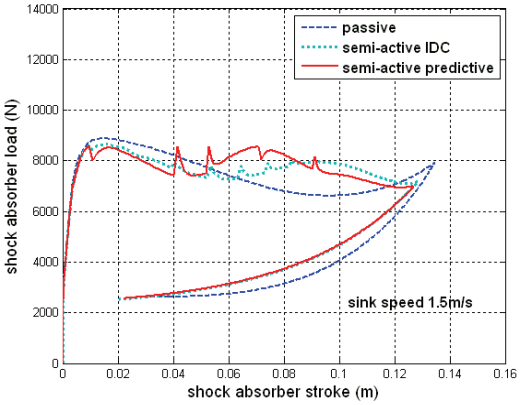


Fig. 9. Efficiency Comparison under Light Landing Load Condition

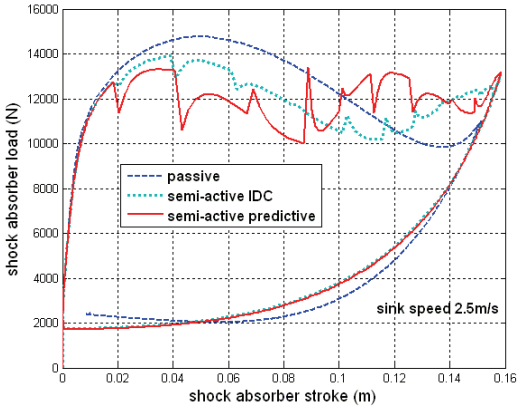


Fig. 10. Efficiency Comparison under Heavy Landing Load Condition

Control Method	Passive	Semi-Active IDC	Semi-Active Predictive
Efficiency/$(2.0m \cdot s^{-1})$	0.8483	0.8788	0.9048
Efficiency/$(1.5m \cdot s^{-1})$	0.8449	0.8739	0.9036
Efficiency/$(2.5m \cdot s^{-1})$	0.8419	0.8554	0.8813

Table 1. Comparison of shock absorber efficiency

4.4 Sensitivity Analysis

Sometimes system parameters such as sinking speed, sprung weight and attitude of aircraft at touch down may be measured or estimated with errors, which will lead to bias of estimation for optimal target load. But the controller should behave robust to withstand certain measurement or estimation errors within reasonable scope so that the airframe will not suffer from large vertical load at touch down.

Simulation of sensitivity analysis is conducted under the standard condition controller design: sinking speed is 2 m/s and aircraft sprung mass is 405 kg, introducing 10% errors for sinking speed and sprung mass individually. The actual sinking speed is measured by avionic equipments and the aircraft sprung mass is estimated by considering the weights of oil, cargo and passengers. The measurement and estimation errors will be less than the assumed maximal one.

From the above Figs.11,12 simulation results, it can seen that the reasonable measuring error of sinking speed has little effect on the performance of nonlinear predictive semi-active controller, whilst estimating error of sprung mass has side effect to the control performance and shock absorber efficiency decreases a little. To further improve the performance under mass estimating error, it is possible to either simply introduce measurement of aircraft mass or develop robust controller which is non-sensitive to estimating the error of aircraft sprung mass.

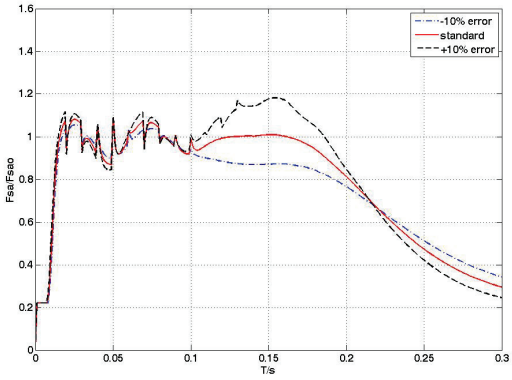


Fig. 11. Sensitivity to sink speed measuring error

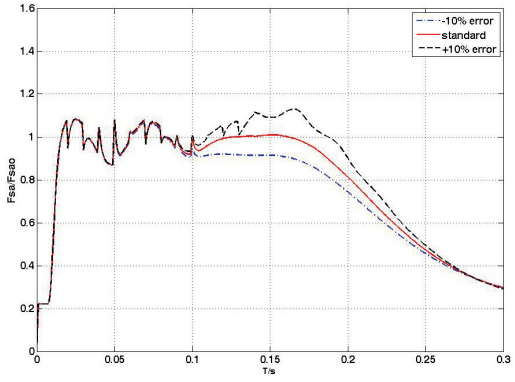


Fig. 12. Sensitivity to sprung mass estimating error

5. Semi-Active Predictive Controller Design for Taxiing Phase

In this section, we will propose a nonlinear predictive controller incorporating radial basis function network (RBF) and backstepping design methodology (Kristic et al., 1995) for semi-active controlled landing gear during aircraft taxiing.

5.1 Hierarchical Controller Structure

A hierarchical control structure which contains three control loops is adopted here. The outer loop determines the expected strut force of the semi-active shock absorber. At touchdown phase and taxiing phase, the computation of the expected strut force will be different due to different design objective. The middle loop is responsible for controlling of solenoid valve's mechanical and magnetic dynamics. The high speed solenoid valve contains high nonlinearity and can not be regulated by traditional linear controller i.e. PID. We develop a RBF network to approximate the nonlinear dynamics which can not be precisely modelled and adopt backstepping, a constructive nonlinear control design method to stabilize the whole nonlinear system. The inner loop is the current loop. It ensures stable tracking of commanded current that middle loop outputs.

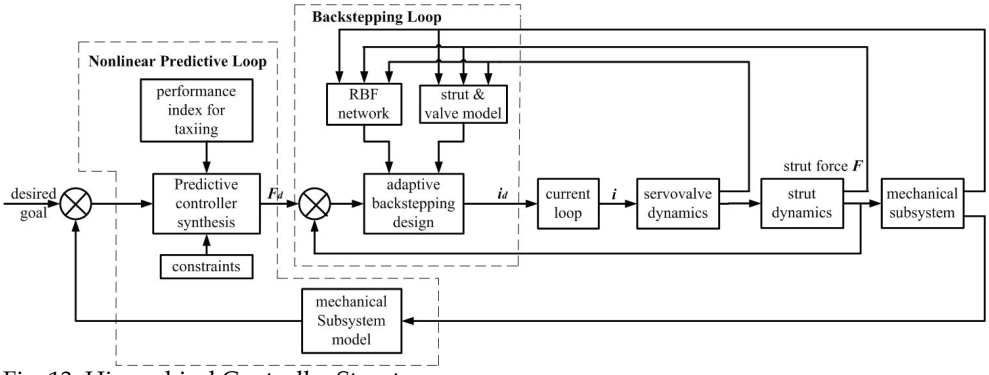


Fig. 13. Hierarchical Controller Structure

5.2 Background for RBF network

A RBF network is typically comprised of a layer of radial basis activation functions with an associated Euclidean input mapping. The output is then taken as a linear activation function with an inner product or weighted average input mapping.

In this paper, we use a weighted average mapping in the output node. The input-output relationship in a RBF with $\mathbf{x} = [x_1, \ldots, x_n]^T$ as an input is given by

$$\Psi(\mathbf{x}, \boldsymbol{\theta}) = \frac{\sum_{i=1}^{m} w_i \exp(-\left|\mathbf{x} - c_i\right|^2 / \gamma^2)}{\sum_{i=1}^{m} \exp(-\left|\mathbf{x} - c_i\right|^2 / \gamma^2)} = \boldsymbol{\theta}^T \boldsymbol{\xi}(\mathbf{x}) \tag{42}$$

where

$$\boldsymbol{\theta} = [w_1, \ldots, w_n]^T \tag{43}$$

$$\xi_i = \frac{\exp(-\left|\mathbf{x} - c_i\right|^2 / \gamma^2)}{\sum_{i=1}^{m} \exp(-\left|\mathbf{x} - c_i\right|^2 / \gamma^2)} \tag{44}$$

The RBF network is a good approximator for general nonlinear function. For a nonlinear function FN, we can express it using RBF network with the following form,

$$F_N = \boldsymbol{\theta}^T \boldsymbol{\xi} + \varepsilon = \hat{\boldsymbol{\theta}}^T \boldsymbol{\xi} + \tilde{\boldsymbol{\theta}}^T \boldsymbol{\xi} + \varepsilon \tag{45}$$

where θ is the vector of tunable parameters under ideal approximation condition, $\hat{\boldsymbol{\theta}}$ under practical approximation condition, $\tilde{\boldsymbol{\theta}}$ parameter approximation error, ε function reconstruction error.

5.3 Outer Loop Design

The function of the outer control loop is to produce a target strut force for semi-active shock absorber by using active control law. Then middle loop and inner loop controller will be designed to approximate the optimal performance that active controller achieves.

(a) Skyhook Controller

At the taxiing phase, the landing gear system acts like the suspension of ground vehicle. So we first adopt the most widely used active suspension control approach – the skyhook controller. At this control scheme the actuator generates a control force which is proportional to the sprung mass vertical velocity. The equation of skyhook controller can be expressed as the following form:

$$F_{sky} = K_{sky}(x_{1d} - x_1) - C_{sky}(x_2 + x_4) \tag{46}$$

In order to blend out low frequency components of the vertical velocity signal which results from the aircraft taxiing on sloped runways or long bumps, we modify it by adding high pass filter to the skyhook controller.

$$x_s = \frac{s}{s + w_k} x_1 \tag{47}$$

where w_k is roll off frequency of high pass filter. Thus we get the desired strut force.

$$F_d = K_{sky}(x_{1d} - x_1) - C_{sky}(x_2 + x_4) - K_{HP} x_s \tag{48}$$

where K_{HP} is a constant scale factor.

(b) Nonlinear Predictive Controller
Compare with traditional skyhook controller, model predictive controller is more suitable for constrained nonlinear system like landing gear system or suspension system. Input and state constraints can be incorporated into the performance index to achieve best performance.
The system model of outer loop controller is eq. (16-19), which can be expressed as follows:

$$\dot{\boldsymbol{x}}_a = \boldsymbol{f}(\boldsymbol{x}_a) + \boldsymbol{g}(\boldsymbol{x}_a)F_d \tag{49}$$

where $\boldsymbol{x}_a = [x_1, x_2, x_3, x_4]$, F_d is the control input and the output equation is $y = x_1$. Then a similar receding-horizon problem can be set up for providing the output-tracking control:

$$\min_{F_d} \boldsymbol{J}[\boldsymbol{x}_a(t), t, F_d] = \min_{F_d} \frac{1}{2}\int_t^{t+T} [e_a^T(\tau)\boldsymbol{Q}_a e_a(\tau) + F_d^T(\tau)\boldsymbol{R}_a F_d(\tau)]d\tau \tag{50}$$

subject to the state equations (49) and

$$e_a(t+T) = 0 \tag{51}$$

where $e_a(t) = x_1(t) - x_{1d}(t)$.
Following a similar synthesis process as in section 4.2, we can get a closed-loop nonlinear predictive output-tracking control law to achieve approximate optimal active control performance.

5.4 RBF-based Backstepping Design (Middle Loop)

In this section we propose a RBF-based backstepping method to complete the design of the semi-active controller. Stability proofs are given.
First we define the force tracking error as $e_1 = F_d - F$. Differentiate and substitute from Eq. (16-25),

$$\begin{aligned}
\dot{e}_1 &= \dot{F}_d - \dot{F} = \dot{F}_d - \frac{d}{dt}[(1+K_m)F_{air} + F_{oil}] \\
&= \dot{F}_d - \frac{d}{dx_5}\left(\frac{\rho {A_o}^3}{2(K_v x_5 C_d)^2\sqrt{1-(K_v x_5 / A_o)^2}}\right){x_2}^2\dot{x}_5 \\
&- \frac{\rho {A_o}^3}{(K_v x_5 C_d)^2\sqrt{1-(K_v x_5 / A_o)^2}} x_2\dot{x}_2 \\
&-(1+K_m)\frac{d}{dx_1}[P_i A_a(\frac{V_0}{V_0 - A_a x_1})^n - P_0 A_a]\dot{x}_1 \\
&= \dot{F}_d + G_1(x_2, x_5)x_6 + H_1(x_1, x_2, x_3, x_4, x_5)
\end{aligned}$$

where $G_1(x_2, x_5)$, $H(x_1, x_2, x_3, x_4, x_5)$ is the nonlinear functions related to the strut dynamics.

(a) First Step
Select the desired solenoid valve movable part velocity as

$$x_{6d} = -G_1^{-1}(H_1 + \dot{F}_d + k_1 e_1) \tag{52}$$

where k_1 is a design parameter. Then we get

$$\begin{aligned}\dot{e}_1 &= \dot{F}_d + G_1(x_2, x_5)x_{6d} + G_1(x_2, x_5)e_2 + H_1(x_1, x_2, x_3, x_4, x_5) \\ &= \dot{F}_d + G_1 e_2 - (H_1 + \dot{F}_d + k_1 e_1) + H_1 = G_1 e_2 - k_1 e_1\end{aligned}$$

where $e_2 = x_6 - x_{6d}$.
Consider the following Lyapunov function candidate

$$V_1 = \frac{1}{2} e_1^2$$

Differentiate V_1, thus we get

$$\dot{V}_1 = e_1 \dot{e}_1 = e_1 G e_2 - k_1 e_1^2$$

(b) Second Step
x_6 is not the true control input. We then choose $u = x_7^2$ as virtual input.
Differentiate e_2, we get

$$\dot{e}_2 = \dot{x}_6 - \dot{x}_{6d} = G_2 u - \frac{C_s}{m_v} x_6 + H_2 + W$$

where $G_2 = B(x_5)/m_v$, $H_2 = -f/m_v - \dot{x}_{6d}$ and $W = -[(K_s + K_f)x_5 + f_0]/m_v$.
Consider the following Lyapunov function candidate

$$V_2 = V_1 + \frac{1}{2} e_2^2 + \frac{1}{2} tr(\tilde{\boldsymbol{\theta}}_1^T \boldsymbol{\Gamma}_1^{-1} \tilde{\boldsymbol{\theta}}_1) + \frac{1}{2} tr(\tilde{\boldsymbol{\theta}}_2^T \boldsymbol{\Gamma}_2^{-1} \tilde{\boldsymbol{\theta}}_2)$$

where Γ_1 and Γ_2 are positive definite matrices. Differentiate V_2

$$\dot{V}_2 = \dot{V}_1 + e_2(G_2 u - \frac{C_s}{m_v} x_6 + H_2 + W) + tr(\tilde{\boldsymbol{\theta}}_1^T \boldsymbol{\Gamma}_1^{-1} \dot{\tilde{\boldsymbol{\theta}}}_1) + tr(\tilde{\boldsymbol{\theta}}_2^T \boldsymbol{\Gamma}_2^{-1} \dot{\tilde{\boldsymbol{\theta}}}_2)$$

Then we choose the control input:

$$u = -\hat{G}_2^{-1}(\hat{H}_2 - \frac{C_s}{m_v}x_{6d} + W + k_2 e_2 + G_1 e_1) \tag{53}$$

where k_2 is a design parameter, $\hat{G}_2 = \hat{\boldsymbol{\theta}}_1^{\,T}\boldsymbol{\xi}_1$ is the estimation of $G_2(x_2, x_5)$, $\hat{H}_2 = \hat{\boldsymbol{\theta}}_2^{\,T}\boldsymbol{\xi}_2$ is the estimation of $H(x_1, x_2, x_3, x_4, x_5)$. Thus we get

$$\dot{V}_2 = \dot{V}_1 + e_2\tilde{G}_2 u + e_2\varepsilon_1 u - \frac{C_s}{m_v}e_2 + e_2\tilde{H}_2 + e_2\varepsilon_2 - e_2k_2e_2 - e_2G_1e_1 + tr(\tilde{\boldsymbol{\theta}}_1^T\boldsymbol{\Gamma}_1^{-1}\dot{\tilde{\boldsymbol{\theta}}}_1) + tr(\tilde{\boldsymbol{\theta}}_2^T\boldsymbol{\Gamma}_2^{-1}\dot{\tilde{\boldsymbol{\theta}}}_2)$$

$$= \dot{V}_1 + e_2\tilde{\boldsymbol{\theta}}_1\boldsymbol{\xi}_1 u + e_2\tilde{\boldsymbol{\theta}}_2\boldsymbol{\xi}_2 - \tilde{\boldsymbol{\theta}}_1\boldsymbol{\xi}_1 u e_2 - \tilde{\boldsymbol{\theta}}_2\boldsymbol{\xi}_2 e_2 - e_2k_2e_2 - \frac{C_s}{m_v}e_2 + e_2\varepsilon_1 u + e_2\varepsilon_2 - e_2G_1e_1$$

$$+ tr[\tilde{\boldsymbol{\theta}}_1^T(\boldsymbol{\Gamma}_1^{-1}\dot{\hat{\boldsymbol{\theta}}}_1 + \boldsymbol{\xi}_1 u e_2)] + tr[\tilde{\boldsymbol{\theta}}_2^T(\boldsymbol{\Gamma}_2^{-1}\dot{\hat{\boldsymbol{\theta}}}_2 + \boldsymbol{\xi}_2 e_2)]$$

Choose the tuning law as:

$$\dot{\hat{\boldsymbol{\theta}}}_1 = -\boldsymbol{\Gamma}_1\boldsymbol{\xi}_1 u e_2, \quad \dot{\hat{\boldsymbol{\theta}}}_2 = -\boldsymbol{\Gamma}_2\boldsymbol{\xi}_2 e_2 \tag{54}$$

So we have

$$\dot{V}_2 = e_1G_1e_2 - k_1e_1^2 - k_2e_2^2 - e_2G_1e - \frac{C_s}{m_v}e_2 + e_2\varepsilon_1 u + e_2\varepsilon_{21}$$

$$= -k_1e_1^2 - k_2(e_2 - \frac{\varepsilon_1 u + \varepsilon_2 - C_s/m_v}{2k_2})^2 - \frac{(\varepsilon_1 u + \varepsilon_2 - C_s/m_v)^2}{4k_2} \le 0$$

Therefore, the system is stable and the error will asymptotically converge to zero.

5.5 Inner Loop Design

The function of the inner loop is to precisely tracking of solenoid valve's current. We apply a simple proportional control to the electrical dynamics as follows

$$V = K_c(x_{7d} - x_7) = K_c(\sqrt{u} - x_7) \tag{55}$$

where K_c is the controller gain.

The above three control loops represent different time scales. The fastest is the inner loop due to its electrical characteristics. The next is the middle loop. It is faster than the outer loop because the controlled moving part's inertial of the middle loop is much smaller than that of the outer loop.

5.6 Numerical Simulation

After touchdown, the taxiing process will last relatively a long time before aircraft stops. To simulate the road excitation of runway and taxiway, a random velocity excitation signal $w(t)$ is introduced into Eq. (18).

$$\dot{x}_3 = x_4 + w(t) \tag{56}$$

The simulation result is compared using airframe vertical displacement, which is one of the most important criterion for taxiing condition. Due to lack of self-tuning capability, the passive landing gear does not behave well and passes much of the road excitation to the airframe. That will be harmful for the aircraft structure and meanwhile make passages uncomfortable. The proposed semi-active landing gear effectively filters the unfriendly road excitation as we wish.

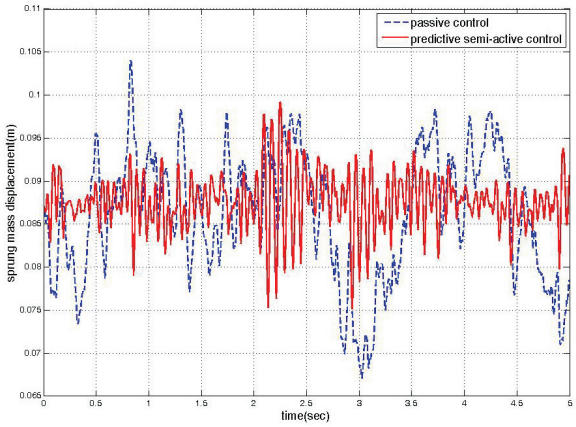


Fig. 14. System Response Comparison under Random Input

From the simulation results of both aircraft touch-down and taxiing conditions, we can see that the proposed semi-active controller gives the landing gear system extra flexibility to deal with the unknown and uncertain external environment. It will make the modern aircraft system being more intelligent and robust.

6. Conclusion

The application of model predictive control and constructive nonlinear control methodology to semi-active landing gear system is studied in this paper. A unified shock absorber mathematical model incorporates solenoid valve's electromechanical and magnetic dynamics is built to facilitate simulation and controller design. Then we propose a hierarchical control structure to deal with the high nonlinearity. A dual mode model predictive controller as an outer loop controller is developed to generate the ideal strut force on both touchdown and taxiing phase. And a systematic adaptive backstepping design method is used to stabilize the whole system and track the reference force in the middle and inner loop. Simulation results show that the proposed control scheme is superior to the traditional control methods.

7. References

Batterbee, D.; Sims, N. & Stanway, R. (2007). Magnetorheological landing gear: 1. a design methodology. *Smart Materials and Structures*, Vol. 16, pp. 2429-2440.

Ghiringhelli, L. G. (2000). Testing of semiactive landing gear control for a general aviation aircraft. Journal of Aircraft, Vol.7, No.4, pp.606-616.

Ghiringhelli, L. G. & Gualdi, S. (2004). Evaluation of a landing gear semi-active control system for complete aircraft landing. *Aerotecnica Missili e Spazio*, No. 6, pp. 21-31.

Hyochoong, B.& Choong-Seok, O. (2004). Predictive control for the attitude maneuver of a flexible spacecraft. *Aerospace Science and Technology*, Vol. 8, No. 5, pp. 443-452.

Karnopp, D. (1983). Active damping in road vehicle suspension systems. *Vehicle System Dynamics*, Vol.12, No. 6, pp.291-316.

Kristic, M.; Kanellakopoulos, I. & Kokotovic, P.V. (1995). *Nonlinear and Adaptive Control Design*, Wiley-Interscience, ISBN: 978-0-471-12732-1, USA.

Krüger, W. (2000). Integrated design process for the development of semi-active landing gears for transport aircraft, PhD thesis, University of Stuttgart.

Liu, H.; Gu, H. B. & Chen, D. W. (2008). Application of high-speed solenoid valve to the semi-active control of landing gear. *Chinese Journal of Aeronautics*, Vol.21, No.3, pp.232-240.

Liu, H. & Gu, H. B. (2008). Nonlinear model for a high-speed solenoid valve and its simulation. *Mechanical Science and Technology for Aerospace Engineering,* Vol.27, No.7, pp.866-870.

Lou, Z.; Ervin, R. & Winkler, C. (1993). An electrorheologically controlled semi-active landing gear. SAE Paper931403.

Lu, P. (1998). Approximate nonlinear receding-horizon control laws in closed form. *International Journal of Control*, Vol. 71, No.1, pp.19-34.

Maemori, K.; Tanigawa, N. & Koganei R (2003). Optimization of a semi-active shock absorber for aircraft landing gear, *Proceedings of ASME Design Engineering Technical Conference*, pp.597-603.

Malaguti, F. & Pregnolato, E. (2002). Proportional control of on/off solenoid operated hydraulic valve by nonlinear robust controller, *Proceedings of IEEE International Symposium on Industrial Electronics*, pp.415-419.

Mehra, R. K.; Amin, J. N. & Hedrick, K. J. (1997). Active suspension using preview information and model predictive control, *Proceedings of the 1997 IEEE International Conference on Control Applications*, pp. 860-865.

Michael, A. H. (1998). Nonlinear model predictive control: current status and future direction. *Computers and Chemical Engineering*, Vol.23, pp.187-202.

Mikulowski, G. & LeLetty, R. (2008). Advanced landing gears for improved impact absorption, *Proceedings of the 11th International conf. on New Actuators*, pp.175-194.

Slotine, J.E. & Li, W.P. (1991). *Applied Nonlinear Control,* Prentice Hall, ISBN: 0-13-040890-5, USA.

Wang, X. M. & Carl U. (1999). Fuzzy control of aircraft semi-active landing gear system, *AIAA 37th Aerospace Sciences Meeting and Exhibit.*

Wu, D. S.; Gu, H.B.; Liu, H. (2007). GA-based model predictive control of semi-active landing gear. *Chinese Journal of Aeronautics*, Vol.20, No.1, pp.47-54.

Yadav, D. & Ramamoorthy, R. P. (1991). Nonlinear landing gear behavior at touchdown. *Journal of Dynamic Systems, Measurement and Control*, Vol.113, No.12, pp.677-683.