If we knew what it was we were doing, it would not be called research, would it?
— Albert Einstein

University of Alberta

Explicit Robust Model Predictive Control and Its Applications

by

Danlei Chu ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**

in

Controls

Department of Electrical and Computer Engineering

Edmonton, Alberta
Fall 2006

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-23003-9
Our file Notre référence
ISBN: 978-0-494-23003-9

Abstract

Introduced over two decades ago, model predictive control (MPC) is nowadays arguably the most widely accepted advanced control design technique in control of industrial processes. It is featured by invoking system input and output constraints into system regulation and guaranteeing the closed-loop stability for nominal MPC systems. However, researchers currently notice that two main barriers hinder the further development of MPC: one is the ubiquitous model uncertainties of industrial processes; and the other is the implementation efficiency of MPC controllers. To improve its adaptability, this thesis proposes a novel MPC scenario – explicit robust model predictive control.

One of the contributions of this thesis is to separate MPC optimization from online implementation, and convert MPC design into multiple parametric subquadratic programming (mp-SQP). It is shown that the analytic solution to mp-SQP problems can be represented by a set of piece-wise affine functions associated with state space partitions. Consequently, online MPC implementation is simplified as an affine function evaluation. Thanks to a novel prediction pattern introduced in this thesis, no high order uncertain terms occur in the MPC optimization, and the critical challenge of finite horizon robust MPC, high computational complexity, is solved skillfully.

As a natural extension of explicit RMPC, robust moving horizon state observation (RMHSO) is also covered in this thesis. The essential point that distinguishes RMHSO from conventional state observation is that RMHSO explicitly combines physical state constraints with the robust observer formulation. This thesis develops two offline RMHSO algorithms, namely, RMHSO with the forward open-loop prediction and RMHSO with the recursive closed-loop prediction. Roughly speaking, the former is less time-consuming than the latter, but the later is less memory-consuming than the former.

Keywords. MPC, robust MPC, robust moving horizon state estimation, stability, robustness, recursive closed-loop prediction, affine function control.

Acknowledgements

This thesis is the outcome of my interaction with my supervisors and colleagues in the past four years. Without their invaluable suggestion and inspiration, I have no way to change the "tedious" Ph.D. research into a "joyful" experience.

First of all, I wish to express my deep gratitude to my supervisors, Dr. Tongwen Chen and Dr. Horacio J. Marquez. They not only helped me to find out an interesting and promising research topic at the very beginning of my study, but also encouraged me to keep at it whenever I could not see the light at the end of the tunnel. It is no doubt that if a student can receive a heuristic guidance, it does not take long for him/her to become professional in his/her area.

I would also like to thank my colleagues, Dr. Huijun Gao and Dr. Guofeng Zhang. Our heated discussion on the key trends in the next generation of control, keep me thinking about the future MPC directions. Their comments are very helpful to finalize Chapter 8.

Finally, a special thanks goes to all members in the advanced control lab because my fellow office colleagues made my Ph.D. study at the University of Alberta unforgettable.

Dedicated to my mom, dad, and sister
for their moral and financial support over the many, many years.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Model predictive control (MPC), also known as moving horizon control (MHC), originated in the late seventies, and has developed considerably since then. Due to the potential to incorporate system constraints into controller design, MPC has attracted extensive attention in both academia and industry. This chapter discusses the principle of MPC, the basic elements of MPC, and some open topics involved in MPC. The major contributions of this thesis are also summarized at the end of this chapter.

1.1 A survey on model predictive control (MPC)

The first version of MPC, model predictive heuristic control (MPHC), was introduced by Richalet in 1976, and later summarized in an Automatica paper [84]. The paper signalled the birth of a novel, advanced control methodology, namely, model predictive control, which formulates controller design as an optimization problem and explicitly represents system's physical constraints by optimization programming. Currently, more than 3000 commercial MPC implementations span in different industries as varied as petro-chemicals, food processing, automotives, aerospace, metallurgy, and pulp and paper processing [28, 76]. This section provides an overview of MPC characteristics.

1.1.1 The principle of MPC

The principle of MPC schemes can be demonstrated by a typical tracking problem. Consider a discrete-time nonlinear system

$$y(k) = f_y\left(U_{k-b\to k}, Y_{k-a\to k-1}, k\right), \tag{1.1}$$

where $k \in \mathbb{Z}^+$ is the time variable, and $u(k) \in \mathbb{R}^p$ and $y(k) \in \mathbb{R}^q$ are inputs and outputs, respectively. $U_{k-b\to k} \in \mathbb{R}^{p\cdot b}$ and $Y_{k-a\to k-1} \in \mathbb{R}^{q\cdot a}$ are stacked inputs and outputs with $U_{k-b\to k} := [u(k-b), \cdots, u(k)]$ and $Y_{k-a\to k-1} := [y(k-a), \cdots, y(k-1)]$. a, $b \in \mathbb{Z}^+$ stand for the delay factors of moving-average and auto-regressive terms. The control objective here is to design a control policy

$$u(k) = f_u(r, U_{k-b\to k-1}, \ Y_{k-a\to k}, k), \tag{1.2}$$

so that $y(k)$ is able to follow a prespecified reference input r ($r \in \mathbb{R}^q$). To this end, an MPC regulator is formulated as an open-loop optimization problem,

$$\begin{aligned}
&\min_{u(k|k),\cdots u(k+N_u-1|k)} \quad J \\
\text{s.t.} \quad & J = \sum_{i=1}^{N_p} \|y(k+i|k) - r(k+i|k)\|_{Q_i}^2 + \sum_{i=0}^{N_u-1} \|u(k+i|k)\|_{R_i}^2, \\
& u(k+i|k) \in \mathcal{A}_u, \ y(k+i|k) \in \mathcal{A}_y, \ y(k|k) = y_k, \\
& u(k+i|k) = u(k+N_u|k) \text{ if } N_u < i \le N_p,
\end{aligned} \tag{1.3}$$

where N_p and N_u are the prediction and control horizons; Q_i and R_i are the input and output weighting matrices. The notation $\|\cdot\|_P^2$ denotes the weighted 2-norm of a vector, namely $\|v\|_P^2 := v^T P v$. $\mathcal{A}_u$ and $\mathcal{A}_y$ are the admissible input and output sets determined by system's physical constraints. For simplicity, we define them by a series of element-wise linear inequalities, i.e., the slab sets

$$\begin{aligned} \mathcal{A}_u &:= \{u \in \mathbb{R}^p |\ u_{\min} \preceq u \preceq u_{\max},\ u_{\min},\ u_{\max} \in \mathbb{R}^p\}, \\ \mathcal{A}_y &:= \{y \in \mathbb{R}^q |\ y_{\min} \preceq y \preceq y_{\max},\ y_{\min},\ y_{\max} \in \mathbb{R}^q\}, \end{aligned} \tag{1.4}$$

where "$\preceq$" is the element-wise inequality sign, meaning $u \preceq u_{\max} \Leftrightarrow u_j \leq u_{\max,\ j}$ for $\forall\ j$ (the index of the vector elements). From (1.1), the future output $y(k+i|k)$ can be expressed in terms of past inputs and outputs, and future manipulated inputs,

$$y(k+i|k) = \begin{cases} f_y\left(U_{k-b+i\to k-1}, U_{k|k\to k+i|k}, Y_{k-a+i\to k-1}, k\right), & \text{if } 0 < i < a, \\ f_y\left(U_{k-b+i\to k-1}, U_{k|k\to k+i|k}, k\right), & \text{if } a \leq i \leq N_p, \end{cases} \tag{1.5}$$

where $U_{k-b+i\to k-1}$ and $Y_{k-a+i\to k-1}$ are past data, and $U_{k|k\to k+i|k} := [u(k|k), \cdots, u(k+i|k)]$ stacks future data. Inserting (1.5) into (1.3), the objective function is recast into

$$J = f_J\left(U_{k-b+i\to k-1}, U_{k|k\to k+i|k}, Y_{k-a\to k-1}, k\right). \tag{1.6}$$

The derivation from (1.5) to (1.6) is the so-called "prediction" and the procedure leads to the notion of MPC. By optimizing the stacked input $U_{k|k\to k+N_u|k}$, sending the first element $u(k|k)$ to the real process, and iterating the same procedure, we can obtain an online MPC regulator, which is capable of driving the output $y(k)$ approaching the reference $r(k)$ in the sense of minimal 2-norms of tracking errors. Obviously, this procedure consists of four steps: 1) modelling controlled systems, 2) predicting future signals, 3) optimizing manipulated inputs, and 4) implementing optimal inputs. Actually, some traditional control algorithms can be converted possibly into such a framework, too. For example, constrained linear quadratic regulation (CLQR) [92] can be regarded as a special case of MPC with $k = 0$, $N_u = N_p = \infty$.

Several excellent MPC survey papers can be found in the literature, for example, [5, 29, 34, 67, 80], among others.

1.1.2 Basic elements of MPC

MPC has four basic elements: model structure, objective function, constraint form, and prediction and control horizons.

- **Model structure**

  Researchers are used to using model structure as a criterion to classify different MPC algorithms, and also based on different model structures, they have developed various MPC algorithms. In current academic papers, researchers always employ state space (SS) model to represent system dynamics, but in commercial implementations, practitioners utilize several other models for MPC application. For example, model predictive heuristic control (MPHC) uses the finite impulse response model (FIR); dynamic matrix control (DMC) uses the finite step response model (FSR); predictive functional control (PFC) uses the time-invariant discrete state space model; and general predictive control (GPC) uses the auto-regressive integrated moving average plus exogenous input model (ARIMAX). The model structure is tightly related to computational complexity and signal's prediction accuracy. Especially for robust MPC design, it is critical to choose an appropriate model structure to describe the system dynamics in the presence of both internal and external uncertainties.

- **Objective function**

  MPC mostly defines an objective function in the form of the summation of the weighted 2-norms of input and output/state deviations from the desired steady state, since a quadratic objective facilitates the closed-loop stability analysis of resulting MPC systems. By defining the objective as a Lyapunov candidate function and regulating the convergence of the candidate along system trajectories, MPC is able to utilize the principle of optimality [2] and combine the stability analysis with online optimization. Consequently, the resulting closed-loop MPC system is stable if the MPC controller is feasible. With the development of more and more elaborate forms of MPC, some other objective structures have emerged in the MPC literature. As an example, a nonlinear MPC for a cyclopentenol reactor adopts the production of cyclopentene as the objective function which maintains the reactor's optimal yield points

[1]. Moreover, robust MPC sometimes adopts a mixed $1/\infty$ norm function as the objective which facilitates the calculation of uppers bound of suboptimal problems [4].

- **Constraint form**

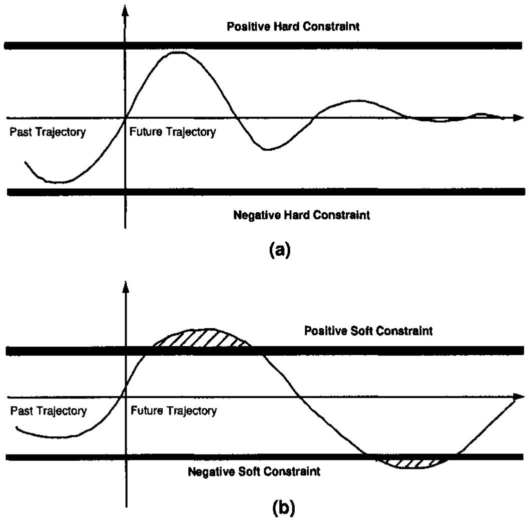


Figure 1.1: Hard and soft constraints

MPC usually defines input and output constraints in the form of element-wise linear inequalities and these inequalities form the admissible input and output polyhedrons with MPC programming. We can easily determine these polyhedrons from physical limitations of each individual input and output channel. For example, the MPC of a boiler system, the output steam temperature is bounded by upper- and lower- bounds. Therefore, the maximal and minimal temperatures of each boiler's outputs form an admissible output set. There are two types of constraints in MPC implementation: the constraint which is strictly inviolated is called hard constraints; and the constraint for which small violations are acceptable is called soft constraints. Fig. 1.1 illustrates the difference between these two types of constraints. The shaded regions in Part (b) show the violation of soft constraints which must be penalized in the objective functions. To scale the violation of soft constraints, slack variables are introduced and penalized as optimization variables. By tuning the weighting matrices associated with slack variables, we can maintain the violation of soft constraints in an acceptable region.

- **Prediction horizon and control horizon**

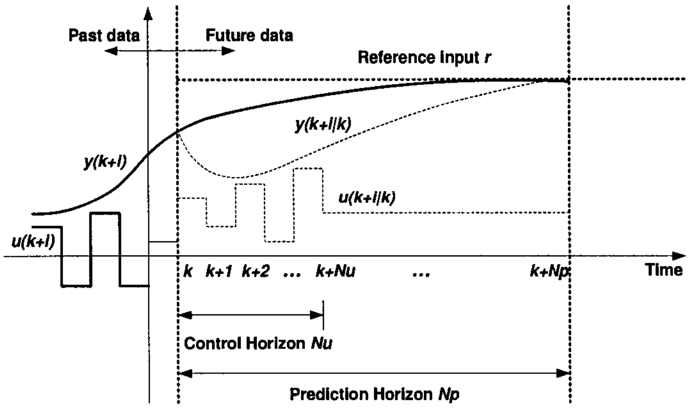


Figure 1.2: Prediction and control horizons

Fig. 1.2 illustrates the concepts of the prediction horizon and the control horizon. The dashed lines in Fig. 1.2 form a dynamic horizon window. In MPC schemes, future outputs of a period N_p, called the prediction horizon, are first predicted, and then manipulated inputs over a period N_u, called the control (or input) horizon, are optimized. Driving the first element of optimized inputs into real processes and shifting both N_p and N_u one step forward, we can realize "moving horizon." This is why MPC is also referred to as moving-horizon control. From the different settings of prediction and control horizons, MPC is divided into finite horizon MPC (FH-MPC) and infinite horizon MPC (IH-MPC). Roughly speaking, the former is superior to the latter in the sense of feasibility and flexibility; but the latter is better in the sense of stability and computational complexity.

In industrial applications, horizons are implemented in several different ways: N_y is realized by multiple moving (MM) or coincidence points (CP); and N_u is realized by multiple moving or single moving (SM). Actually, the different types of horizons correspond to the different objective criteria. Table 1.1 explains the relationship between the objective function and the type of horizons. For ease of notation, here we assume that the objective function is formulated in the form of weighted 2-norms. Fig. 1.3 shows the definition of coincidence points. Note here that the reference input r in Part (b) is regulated offline.

Table 1.1: The realization of horizons

Horizons	Objective function
MM N_y, MM N_u	$J = \sum_{i=1}^{N_p} \Vert y(k+i\vert k) - r(k+i\vert k)\Vert_{Q_i}^2 + \sum_{i=0}^{N_u-1} \Vert u(k+i\vert k)\Vert_{R_i}^2$
MM N_y, SM N_u	$J = \sum_{i=1}^{N_p} \Vert y(k+i\vert k) - r(k+i\vert k)\Vert_{Q_i}^2 + \Vert u(k\vert k)\Vert_{R_0}^2$
CP N_y, MM N_u	$J = \sum_{j} \Vert y(k+i\vert k) - r(k+i\vert k)\Vert_{Q_i}^2 + \sum_{i=0}^{N_u-1} \Vert u(k+i\vert k)\Vert_{R_i}^2$ (j is the index of coincidence points)
CP N_y, SM N_u	$J = \sum_{j} \Vert y(k+i\vert k) - r(k+i\vert k)\Vert_{Q_i}^2 + \Vert u(k\vert k)\Vert_{R_0}^2$

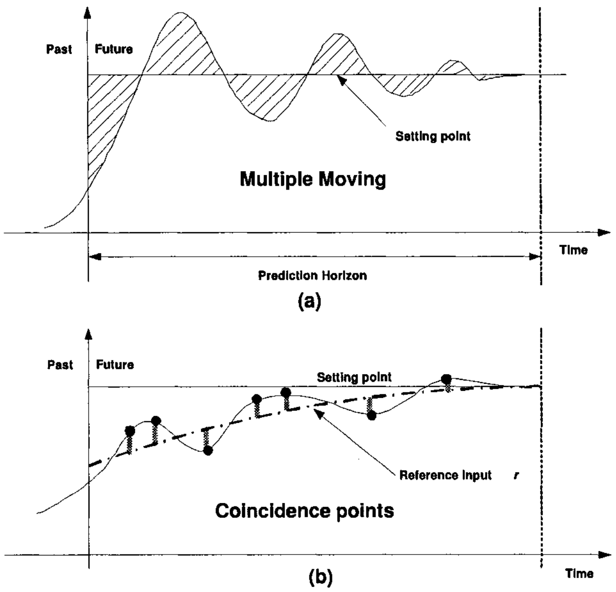


Figure 1.3: Multiple moving and coincidence points

The above four elements are very important for MPC design; and they tightly related to performance of resulting MPC systems, i.e., stability, feasibility, computational complexity, implementation efficiency, and aggressiveness and conservativeness.

1.1.3 Open topics in MPC

Although MPC has been widely accepted in industry, it is still far from enjoying a complete theoretical analysis. For example, the polices of tuning parameters are still open issues, and currently these parameters are determined by trial-and-error. Moreover, some advanced MPC, e.g., nonlinear MPC (NMPC) and robust MPC (RMPC), have only recently begun to be discussed in literature. In this subsection, some open topics involved in MPC are extensively addressed.

- Closed-loop stability

  Consider a system defined by

$$x(k+1) = f(x(k),\ u(k)). \tag{1.7}$$

  To realize some control objective, the cost is formulated by minimizing

$$V(x,k,u) = \sum_{i=0}^{N_p-1} W(x(k+i|k), u(k+i|k)) + F(x(k+N_p|k)), \tag{1.8}$$

  where $W(\cdot)$ denotes the predicted internal energy. If $W(\cdot)$ is defined in the form of (1.3), (1.8) implies that the weighting matrices R_i and Q_i are time-invariant. $F(\cdot) \geq 0$ is called the terminal cost, and its linear counterpart can be expressed by

$$F(x(k+N_p|k)) = \|x(k+N_p|k)\|_P^2 \ \ (P \text{ is a positive symmetric matrix}).$$

  Let $V^o(\cdot)$ denote the optimal value of $V(\cdot)$, corresponding to the optimal input sequence $U^o_{k\to k+N_u-1}$ with

$$U_{k\to k+N_u-1} := \begin{bmatrix} u(k|k) & \cdots & u(k+N_p-1|k) \end{bmatrix}.$$

  In the context, the superscript "o" stands for optimal solutions. Setting $V^o(\cdot)$ to be a Lyapunov function at instant k, and then by the principle of optimality [2] we have

$$\begin{aligned} & V^o_{k\to k+N_p}(x(k), k, u_o(k|k)) \\ = \ & W^o(x(k), u_o(k)) + V^o_{k+1\to k+N_p}(x(k+1), (k+1), u_o(k+1)), \end{aligned}$$

where the total objective is separated into the initial internal energy and the rest of the piece objective. The notation $V^o_{k+i \to k+j}(\cdot)$ stands for the piece objective spanning from the predicted instant $(k+i)$ to instant $(k+j)$ $(0 \leq i < j \leq N_p)$. Thus the difference between the Lyapunov functions at horizon k and $(k+1)$ is

$$\begin{aligned} & V^o_{N_p}\left(x\left(k+1\right),\left(k+1\right),u_o\left(k+1|k+1\right)\right) - V^o_{N_p}\left(x\left(k\right),k,u_o\left(k|k\right)\right) \\ = \; & F^o\left(x\left(k+N_p+1\right)\right) + W^o\left(x\left(k+N_p\right),u\left(k+N_p\right)\right) \\ & -F^o\left(x\left(k+N_p\right)\right) - W\left(x\left(k\right),u_o\left(k\right)\right). \end{aligned}$$

Because $W(x\left(k\right),\; u_o\left(k|k\right))$ is a nonnegative function, the closed-loop stability of the MPC system in (1.7) can be guaranteed by adding the extra terminal constraint

$$F^o\left(x\left(k+N_p+1\right)\right) + W^o\left(x\left(k+N_p\right),u\left(k+N_p\right)\right) - F^o\left(x\left(k+N_p\right)\right) < 0. \tag{1.9}$$

How to define the internal energy function $W(\cdot)$ and the terminal cost $F(\cdot)$ to satisfy the condition in (1.9) becomes pivotal to the stability analysis of the resulting MPC systems. Moreover, the condition in (1.9) is only effective for nominal MPC design, but for RMPC systems, it is unreasonable to derive the optimal solution $V^o(\cdot)$ and keep using Bellman's principle of optimality. Consequently, we can not easily derive a similar condition to (1.9) for RMPC. How to guarantee the closed-loop stability for RMPC systems still remains an open problem [63].

- Tuning parameters

  From Section 1.1.2, we know that weighting matrices are effective approaches to MPC tuning, and the major usage of weighting matrices is to adjust relative priorities of penalized variables in an objective function. Moreover, to facilitate closed-loop stability, a terminal weighting is separated from the output weighting Q_i, and constructed to guarantee the shrinking of a selected Lyapunov candidate. However, how to choose the proper terminal weighting to satisfy closed-loop stability as well as feasibility is a nontrivial problem. On the other hand, MPC always fixes weighting matrices as constant for individual penalty terms in objective functions, except for the terminal weighting for stability. The fixed weightings facilitate the stability and feasibility analysis but possibly impair closed-loop dynamics. Although there are some software packages that support varying weightings, e.g., MATLAB-MPC Toolbox [7], the trend of the weighting's change has to be determined by trial-and-error. As another effective tuning parameters, horizon length impacts on MPC dynamics dramatically too. From industrial implementations, we know that dynamics of closed-loop MPC systems are normally dependent on the difference between the prediction horizon N_p and the control horizon N_u instead of their individual values. Increasing the difference results in faster responses but impairs closed-loop stability of resulting MPC systems. Conversely, stability is improved, but performance is not. From another point of view, the freedom of an optimization problem is dependent on the number of manipulated variables and hard constraints. By increasing the horizon length of the variables which are not bounded by any hard constraints, the feasibility of algorithms may be improved. The rules mentioned above are derived only from MPC implementations. The theoretical analysis on MPC schemes with varying horizons remains an open problem.

- System uncertainties and robust MPC

  To improve flexibility, researchers recently extended nominal MPC into the area of robust MPC (RMPC), which incorporates system internal uncertainties (modelling perturbation) and external uncertainties (input/output disturbances) into controller design. However, a number of barriers exist in RMPC. The major difficulty comes from the computational complexity of future state/output predictions. As an example, for systems with linear discrete state space models and a perturbed system $A-$ matrix, high-order uncertain terms appear in the expressions of predicted signals. Normally it is difficult to generalize the effects of these terms on MPC online optimization and implementation. Therefore, for a successful RMPC algorithm, it becomes critical to construct a proper framework describing the characteristics of these high-order uncertain factors. Over the past several years, various strategies have been developed: Langson *et al.* proposed an uncertain "tube" to maintain controlled trajectories inside of the tube under an associated piecewise affine control policy [49]. Park and Jeong modified system parameter perturbations into the structured uncertainties bounded by a parametric increment rate [73]. Casavola *et al.* kept using the traditional norm-bounded uncertainties in the feedback loop and took advantage of the robustness analysis tool developed by Primbs and Nevistic [75] to realize robust moving horizon control [15]. Fukushima and Bitmead constructed a comparison model for the worst-case analysis and combined it with a robust Lyapunov function to simplify quadratic programming with uncertain terms [31]. Wang and Rawlings developed a convex hull set for all possible system uncertain terms and used a family of the subsystems (no uncertain terms) in the structure of the node-branch-tree to predict future variables [105]. Although these algorithms facilitate state/output predictions, to some extent, they increase the computational complexity and hinder the effectiveness of the RMPC implementation. Furthermore, few of them are capable of incorporating both system internal uncertainties and external disturbances with system regulation. The discussion on RMPC with internal and external uncertainties is covered by [71, 72, 68], but these papers postulate that external disturbances were constant unknown variables, which is not the case in most real processes. How to attenuate both internal uncertainties and external disturbances without aggravating the computational complexity remains an open problem.

- Implementation efficiency and explicit MPC

  The efficiency of online MPC implementation is another major barrier for advanced MPC algorithms, e.g., RMPC and NMPC. This point can be understood from the "tetralogy"of traditional RMPC: determine initial parameters, perform state/output predictions, optimize stacked input sequence, and implement first optimal input. For online MPC, all of the four steps have to be completed within one sampling period. Considering the nature of computational complexity, it takes a long time to complete the whole procedure. This limitation restricts the application of advanced MPC in industry. Evidence of this is that most existing RMPC algorithms are applicable only to slow systems; otherwise, some additional work has to be done to avoid the risk of inconsistent implementation [66, 99, 104]. Furthermore, to guarantee the feasibility of algorithms, system time constants should be no larger than the prediction horizon [81]. The additional requirement leads to further computational burdens (References [43, 44] show that the minimal prediction horizon $\underline{N}_y$ can be determined to ensure the feasibility of algorithms). A natural strategy to overcome this barrier is to employ offline RMPC, namely offline calculation of optimal inputs and online implementation of manipulated inputs. Offline or explicit MPC, however, remains an open problem, especially for robust cases.

1.2 The major contributions of the thesis

This thesis introduces a novel prediction pattern to reduce the computational complexity of RMPC and proposes the recursive closed-loop prediction pattern to simplify multiple-horizon prediction. By taking advantage of convex optimization techniques, the thesis accomplishes RMPC programming offline and separates RMPC optimization from online implementation. MPC implementation in this thesis essentially becomes function evaluation so that the implementation efficiency is improved dramatically. The thesis defines a novel multiple-parametric sub-quadratic programming (mp-SQP) problem, and by iterating mp-SQP it achieves robust MPC through a series of piece-wise affine functions of current state measurement associated with state space partitions (called *Critical Regions* in Reference [11]). The explicit RMPC presented in this thesis can guarantee asymptotic closed-loop stability of resulting MPC systems and by setting two tuning variables, namely terminal weighting and terminal feedback gain, it is capable of adjusting the tradeoff between system robustness and performance. This thesis also discusses a nontrivial problem: given admissible output sets, how to derive the admissible state set. The problem is solved by two approaches: piece-wise linear norm of output disturbances [12] and polyhedral Voronoi sets of admissible states [13]. Chapter 5 discusses the details relative to the explicit RMPC in recursive closed-loop prediction.

In the current literature, researchers always formulate MPC in state space models. Therefore, to obtain state feedback, state observers are necessary for MPC systems with unmeasured or partially unavailable states. To overcome this limitation, this thesis proposes two robust moving horizon state observer (MHSO) algorithms in Chapter 6, which are the extension of explicit RMPC. These two algorithms are able to handle system's nonlinear uncertainties and state's physical constraints. By employing open-loop and closed-loop prediction, the algorithms convert robust MHSO to an mp-SQP problem; and meanwhile they guarantee the convergence of observation errors by solving an algebraic Riccati equation and a semi-definite optimization problem. Chapter 6 also introduces another robust observer which formulates the design as a *Maximizing Determinant* (MAXDET) problem. The major feature of this approach is that the observer guarantees the convergence of the observer error in the sense of Lyapunov and ensures the observer error bounded by an ellipsoidal invariant set. Therefore, regarding the bounded error as modelling uncertainties, we can easily associate robust observer with RMPC formulation.

Chapter 3 describes a finite horizon RMPC (FH-RMPC) algorithm using linear matrix inequality (LMI) techniques; this is an important complement of offline finite horizon RMPC discussed in Chapter 5. The motivation for this work comes from the seminal paper published by Kothare *et al.* in 1996 which solved the problem of infinite horizon robust MPC (IH-RMPC) using LMIs [45]. Compared with IH-RMPC, FH-RMPC has more tuning freedom and can deal with more general uncertainty structures. To capture modelling uncertainties and facilitate future state/output prediction, the thesis constructs a moving average system matrix for system uncertainties, and it is pivotal for this algorithm. By adding two additional terminal cost constraints, FH-RMPC guarantees the closed-loop stability of resulting MPC systems if the optimization problem for FH-RMPC is feasible. From simulation examples, we can see that FH-RMPC using LMIs is more flexible and reliable than IH-RMPC using LMIs.

1.3 Outline of the thesis

Chapter 2 investigates four typical MPC algorithms: dynamic matrix control (DMC) [23, 24], model algorithm control (MAC) [37], predictive functional control (PFC)

[82], and general predictive control (GPC) [21, 22]. These four algorithms are adopted in various commercial packages and are able to demonstrate most of the characteristics of nominal MPC.

Chapter 3 discusses RMPC algorithms using LMIs. It includes two parts: IH-RMPC and FH-RMPC. The former was proposed by Kothare *et al.* in 1996; the latter is the algorithm proposed in this thesis [19]. This chapter first reviews two types of structured uncertainties which are widely used in MPC design, and then discusses the impact of uncertainties on computational complexity and the prediction of future signals. The closed-loop stability issues for both algorithms are also covered in this chapter: the former utilizes the invariant set theorem to drive the convergence of system trajectories and the latter develops two extra terminal cost constraints for the convergence of Lyapunov candidacy functions.

Chapter 4 reviews explicit model predictive control for nominal systems, i.e., no internal or external uncertainties are included in the MPC formulation. It first introduces a type of optimization problem, multiple-parametric quadratic programming (mp-QP) [26], and then shows that explicit MPC may be converted into an mp-QP problem. The control policies of nominal explicit MPC are represented by a set of affine functions associated with state space partitions. All techniques covered in this chapter are quite new (developed after 2002), and they are the basis for Chapter 5.

Chapter 5 is the essential part of this thesis, i.e., explicit robust model predictive control using recursive closed-loop prediction. Although explicit robust MPC is based on nominal explicit MPC, it is not a direct extension of the latter. Chapter 5 solves three key problems associated with explicit robust MPC, including admissible state sets, recursive closed-loop prediction, and asymptotic closed-loop robust stability. To solve these problems, it proposes a new type of optimization problem, multiple-parametric sub-quadratic programming (mp-SQP). Here the letter "*S*" is added to distinguish it from existing mp-QP problems.

Chapter 6 is a natural extension of robust MPC. We know that the optimal policies of explicit robust MPC are piece-wise affine functions of the current state measurement associated with some state space partition. In the case that states are unmeasurable or partially unmeasurable, it is necessary to combine the controller design with state estimation. Chapter 6 first presents several difficulties involved in robust observer design: 1) the system is corrupted by internal and external disturbances; 2) the state constraints have to be incorporated with state observation; and 3) to preserve the advantages of offline MPC (with its low implementation cost), an offline robust state observation approach is indispensable. Chapter 6 addresses these three challenges in the sequel.

Chapter 7 considers industrial applications of the algorithms developed in this thesis. It uses a SYNSIM model to evaluate the effectiveness of our algorithms. SYNSIM is a software package developed by researchers at the University of Alberta and engineers at the Syncrude Canada Ltd. (SCL) [86]. It is a simulation for the utility plant of SCL in Fort McMurray, Alberta, Canada, which is a co-generation system composed of boilers, turbines, headers, and let-down sub-systems. Here, we try to design an MPC controller for the boiler plus header sub-systems, i.e., a master controller. The combined system model is first identified using the MATLAB system identification toolbox [52], and then an explicit robust MPC regulator is developed based on the identified model. Finally, the master controller is able to demonstrate the effectiveness of the algorithms proposed in this thesis.

Chapter 8 discusses some future research topics on MPC.

1.4 Notation and symbols

- $\mathbb{S}_{+}^{n}$ $(\mathbb{S}_{++}^{n})$ denotes the space of symmetric nonnegative (positive) definite $n \times n$ matrices, and $\mathbb{D}_{+}^{n}$ $(\mathbb{D}_{++}^{n})$ stands for the space of diagonal nonnegative (positive) matrices.

- $\|X\|_P^2 := X^T P X$ denotes the weighted 2-norm of a matrix X, where $P \in \mathbb{S}_{+}^{n}$. $\underline{\sigma}(X)$, $\bar{\sigma}(X)$ are minimal and maximal singular values of X.

- $x(k+i)$ denotes the predicted states over the kth prediction horizon, similar to the definitions of $u(k+i)$ and $y(k+i)$, i.e., $x(k+i) := x(k+i|k)$ without special indication.

- x_j is the jth element of a vector x, X_j is the jth row of a matrix X, and X_{ij} is the ijth element. The superscript ‘ o’ stands for the corresponding optimal or sub-optimal solution, e.g., x^o.

- $\preceq$ ($\prec$) and $\succeq$ ($\succ$) denote the generalized element-wise (strict) inequality signs, i.e., $x \preceq x_{\max} \Leftrightarrow x_j \leq x_{\max,\, j}$ for $\forall\ j$.

- $\hat{x}\,(k-N+i)$ denotes the ith predicted estimation over the kth prediction horizon given the initial value $\hat{x}\,(k-N)$, i.e., $\hat{x}\,(k-N+i) := \hat{x}\,(k-N+i|k)$ for ease of notation.

- The sign of "$\cdot$" is defined as independent variables of a function, whose definition can be inferred from contexts, e.g., $f\,(x,k)$ is sometimes written as $f(\cdot)$ without special indication.

- $f_{k_1\to k_2}$ denotes the sequence of $\{f(k_1), \ \cdots, \ f(k_2)\}$, similar to the definitions of $u_{k_1\to k_2}$, $x_{k_1\to k_2}$ and $e_{k_1\to k_2}$.

1.5 Acronyms

ARE Algebraic Riccati Equation

ARIMAX Auto-Regressive Integrated Moving Average plus eXogenous input model

CLQR Constrained Linear Quadratic Regulation

CR Critical Region

CSTR Continuous Stirred Tank Reactor

DMC Dynamic Matrix Control

EMPC Explicit Model Predictive Control

ERMPC Explicit Robust Model Predictive Control

FH-MPC Finite Horizon Model Predictive Control

FIR Finite Impulse Response

FSR Finite Step Response

GEVP Generalized Eigenvalue Programming

GPC Generalized Predictive Control

IDCOM Identification and Command

IH-MPC Infinite Horizon Model Predictive Control

LMI Linear Matrix Inequality

LQR Linear Quadratic Regulation

MAC Model Algorithm Control

MAXDEX Maximizing Determinant programming

MHC Moving Horizon Control

MHSE Moving Horizon State Estimation

MHSO Moving Horizon State Observer

MIMO Multiple Input Multiple Output

MPC Model Predictive Control

MPHC Model Predictive Heuristic Control

mp-QP multi-parametric Quadratic Programming

mp-SQP multi-parametric Sub-Quadratic Programming

NMPC Nonlinear Model Predictive Control

PFC Predictive Functional Control

QP Quadratic Programming

RMHSO Robust Moving Horizon State Observer

RMPC Robust Model Predictive Control

SCL Syncrude Canada Ltd.

SDP Semidefinite Programming

SOCP Second Order Cone Programming

Chapter 2

Model predictive control algorithms

This chapter reviews four of the most popular MPC algorithms: DMC, MAC, PFC, and GPC. Dynamic Matrix Control (DMC) is the first commercial MPC package. It uses an identification and Command (IDCOM) software [37] to achieve system identification and control optimization at the same time. Model Algorithm Control (MAC) uses a similar scenario to that of DMC, but makes two innovations: an impulse response model replaces the step response model, and an approximated function replaces the fixed reference input. Predictive functional control (PFC) is developed for fast linear and nonlinear processes. It proposes the concept of coincidence points along horizon windows and constructs a linear combination of parameterized basis functions as optimal manipulated inputs. General predictive control (GPC) is the first version of stochastic MPC. It uses an ARIMAX model to perform both future output and disturbance prediction. Diophantine equations are employed to facilitate future disturbance prediction.

2.1 Dynamic matrix control

Before stating the mathematical details, we choose to list the major features of DMC:

- Use the step response model to describe system dynamics.
- Employ a quadratic performance objective over a finite prediction horizon to penalize the deviations between outputs and prespecified set points.
- Assume that the output disturbances are constant. The difference between the current output measurement and the current predicted output is implemented as future disturbances along all horizons.
- Convert DMC programming into a QP problem. It is possible to obtain an explicit solution of DMC problems in the case of no input and output disturbances.

Besides the above features, DMC inherits some disadvantages:

- Works with only asymptotically stable systems.
- Cannot handle systems with large internal and external uncertainties (modelling uncertainties is called as internal uncertainties and input/output disturbances is referred to as external uncertainties in this thesis).
- May be impractical for multiple-input-multiple-output (MIMO) systems with high dimensions because step response matrices for an MIMO system are memory consuming.
- Require that all controlled outputs be measured.

DMC can be formulated as a QP problem as follows:

$$J = \min_{\Delta u(k|k),\ \cdots,\ \Delta u(k+N_u-1|k)} \sum_{i=1}^{N_p} \|r - y\,(k+i|k)\|_{Q_i}^2 + \sum_{i=0}^{N_u-1} \|\Delta u\,(k+i|k)\|_{R_i}^2\,, \quad (2.1)$$

subject to

$$\begin{aligned} y\left(k+i|k\right) = & \sum_{j=1}^{i} s_j \Delta u\left(k+i-j|k\right) + \sum_{j=i+1}^{N} s_j \Delta u\left(k+i-j\right) \\ & + d(k+i|k), \end{aligned} \tag{2.2}$$

$$u\left(k+i|k\right) = u\left(k-1\right) + \sum_{j=0}^{i} \Delta u\left(k+j|k\right), \tag{2.3}$$

$$\Delta u\left(k+i|k\right) = \Delta u\left(k+N_u|k\right) \text{ if } N_u < i \leq N_p,$$

$$\sum_{i=1}^{N_p} C_{y,i}^{l} y(k+i|k) + \sum_{i=0}^{N_u-1} C_{u,i}^{l} u(k+i|k) + C^l \leq 0, \ (0 \leq l \leq N_c). \tag{2.4}$$

N_p and N_u stand for the prediction horizon and control horizon. $\Delta u\left(k+i|k\right)$, $y\left(k+i|k\right)$, and $d(k+i|k)$ are the predicted input derivation, predicted output and future disturbance over the kth horizon, respectively. $Q_i \in \mathbb{S}_+$ and $R_i \in \mathbb{S}_+$ are weighting matrices, and s_j is a step response, i.e.,

$$S := \begin{bmatrix} s_1, & \cdots, & s_N \end{bmatrix}. \tag{2.5}$$

The sequence S is usually obtained by system identification, and pre-stored in a computer for output prediction. For simplicity, here we first consider the step responses for a single-input-single-output system (SISO). The truncation scalar N of the step response satisfies

$$N \geq N_p + 1 \geq N_u + 1.$$

The future disturbances are assumed to be constant along all horizons, satisfying

$$d(k+i|k) = d(k|k) = y(k) - y(k|k),$$

where $y(k)$ is the output measurement at instant k, and $y(k|k)$ is the predicted output derived by

$$y(k|k) = \sum_{j=1}^{N} s_j \Delta u\left(k-j\right).$$

Note that both $y(k)$ and $y(k|k)$ can be calculated offline, so $d(k+i|k) \equiv d(k)$ is also calculated offline. Eqs. (2.2) - (2.3) express the future outputs and inputs, and (2.4) describes the future input/output constraints. The coefficients of (2.4), namely

$C^l_{y,i}$, $C^l_{u,i}$, C^l, are constant, l is the index of the constraints, and N_c is the number of constraints. Rewriting (2.2) in the form of stacked matrices, we have

$$Y_{k+1\to k+N_p} = M_f \Delta U_{k|k\to k+N_u-1|k} + M_p \Delta U_{k-N+1\to k-1} + \mathbf{1}\cdot d(k|k) \tag{2.6}$$

where

$$\begin{aligned} M_f &:= \begin{bmatrix} s_1 & 0 & 0 & \cdots & 0 \\ s_2 & s_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{N_u} & s_{N_u-1} & s_{N_u-2} & \cdots & s_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{N_p} & s_{N_p-1} & s_{N_p-3} & \cdots & s_{N_p-N_u+1}+\cdots+s_1 \end{bmatrix}, \\ M_p &:= \begin{bmatrix} s_N & s_{N-1} & s_{N-3} & \cdots & \cdots & s_2 \\ 0 & s_N & s_{N-2} & \cdots & \cdots & s_3 \\ \vdots & \vdots & \vdots & \cdots & \cdots & \vdots \\ 0 & 0 & s_N & \cdots & \cdots & s_{N_u+1} \\ \vdots & \vdots & \vdots & \ddots & \cdots & \vdots \\ 0 & 0 & 0 & S_N & \cdots & s_{N_p+1} \end{bmatrix}. \end{aligned} \tag{2.7}$$

In (2.7), "**1**" stands for a full-one vector with an appropriate dimension. $\Delta U_{k|k\to k+N_u-1|k}$ is the stacked matrix with predicted input derivations to be optimized. $\Delta U_{k-N+1\to k-1}$ is the matrix of past data and pre-stored in a computer. M_f reflects the future system dynamics and M_p reflects the past ones. Due to the physical meaning of M_f and M_p (dynamic matrices), the algorithm is referred to as DMC.

In the same fashion, the objective in (2.1) and the constraints in (2.4) can be recast into the form of stacked matrices. Consequently, DMC programming is converted into a quadratic programming (QP) problem.

Conclusion 2.1 *The manipulated inputs of DMC programming can be optimized by a QP problem, namely*

$$\Delta U^o_{k|k\to k+N_u-1|k} := \left[\Delta u^o(k|k) \quad \Delta u^o(k+1|k) \quad \cdots \quad \Delta u^o(k+N_u-1|k)\right]^T,$$

is the solution to a QP problem,

$$J = \min_{\Delta U_{k|k\to k+Nu-1|k}} \left\|R - Y_{k\to k+N_p}\right\|^2_{\mathcal{Q}} + \left\|\Delta U_{k|k\to k+N_u-1|k}\right\|^2_{\mathcal{R}}, \tag{2.8}$$

subject to

$$
\begin{aligned}
&Y_{k\rightarrow k+N_p} = M_f \Delta U_{k|k\rightarrow k+N_u-1|k} + M_p \Delta U_{k-N+1\rightarrow k-1} + \mathbf{1}\cdot d(k),\\
&C^l_{y,1\rightarrow N_p} Y_{k\rightarrow k+N_p} + C^l_{u,0\rightarrow N_u-1} U_{k|k\rightarrow k+N_u-1|k} + C^l \leq 0 \ (0 \leq l \leq N_c).
\end{aligned}
$$

where

$$
\begin{aligned}
&\mathcal{Q} := \mathrm{diag}(Q_1,\ \cdots,\ Q_{N_p}),\ \ \mathcal{R} := \mathrm{diag}(R_1,\ \cdots,\ R_{N_u}),\\
&C^l_{y,1\rightarrow N_p} := \begin{bmatrix} C^l_{y,1} & \cdots & C^l_{y,\ N_p} \end{bmatrix},\ C^l_{u,1\rightarrow N_u} := \begin{bmatrix} C^l_{u,0} & \cdots & C^l_{u,\ N_u-1} \end{bmatrix}.
\end{aligned}
$$

The above discussion just concentrates on SISO systems, but it can easily extended to MIMO systems, using the stacked step response matrix to replace the scalar s_i, i.e., setting

$$
S_{stack}(i) = \begin{bmatrix} s_{11}(i) & \cdots & s_{1p}(i) \\ \vdots & \ddots & \vdots \\ s_{q1}(i) & \cdots & s_{qp}(i) \end{bmatrix}. \tag{2.9}
$$

Here p is the dimension of inputs and q is the dimension of outputs. From (2.9), it can be seen that DMC is memory consuming for MIMO systems with high dimensions.

2.2 Model algorithm control

Model algorithm control (MAC) is a variant of model predictive heuristic control (MPHC), and marketed widely under the software package IDCOM-M (M is used to distinguish it from a SISO version of IDCOM) [37]. It differs from DMC in the following aspects:

- Controlled systems are modelled using impulse responses.
- Penalized variables in objective functions are in terms of u instead of Δu.
- MAC set the control horizons equal to the predictive horizon, and output weighting $Q_i \equiv I$ and $R_i \equiv \lambda I$. Therefore, the tuning parameters N_u, Q_i, and R_i are replaced by a positive scalar λ.
- The reference trajectory is a smooth approximation from the previous predicted output towards the prespecified setting point,

$$
w(k+i|k) = \alpha w(k+i-1|k) + (1-\alpha) r \ \ (1 \leq i \leq N_p), \text{ with } w(k) = y(k). \tag{2.10}
$$

α is a tuning parameter bounded by $[0,\ 1)$ and is able to adjust the tradeoff between aggressiveness and conservativeness of the MAC design.

Because MAC uses the impulse response to perform output prediction, the future output of an MAC system over the kth horizon can be expressed by

$$y\left(k+i|k\right) = \sum_{j=1}^{i} h_j u\left(k+i-j|k\right) + \sum_{j=i+1}^{N} h_j u\left(k+i-j\right) + d(k+i|k), \tag{2.11}$$

and the future constant disturbance is

$$d\left(k+i|k\right) = d\left(k|k\right) = y\left(k\right) - \sum_{j=1}^{N} h_j u\left(k-j\right), \tag{2.12}$$

where

$$H := \begin{bmatrix} h_1 & \cdots & h_N \end{bmatrix}\ \left(N \geq N_p + 1\right). \tag{2.13}$$

Similar to DMC, H is the impulse sequence and derived by system identification packages. In the same fashion, MAC can be converted into a QP problem.

Conclusion 2.2 *The manipulated inputs of MAC programming can be optimized by a QP problem; namely*

$$U^o_{k|k\to k+N_u-1|k} := \begin{bmatrix} u^o\left(k|k\right) & u^o\left(k+1|k\right) & \cdots & u^o\left(k+N_p-1|k\right) \end{bmatrix},$$

is the solution to a QP problem,

$$J = \min_{U_{k|k\to k+Np-1|k}} \left\|W_{k+1\to k+N_p} - Y_{k+1\to k+N_p}\right\|^2 + \lambda \left\|U_{k|k\to k+N_p-1|k}\right\|^2, \tag{2.14}$$

$$\begin{aligned} s.t. \qquad & Y_{k+1\to k+N_p} = H_f U_{k|k\to k+N_u-1|k} + H_p U_{k-N+1\to k-1} + \mathbf{1}\cdot d(k|k), \\ & C^l_{y,1\to N_p} Y_{k\to k+N_p} + C^l_{u,0\to N_p-1} U_{k|k\to k+N_p-1|k} + C^l \leq 0\ (0 \leq l \leq N_c). \end{aligned}$$

where

$$W_{k+1\to k+N_p} := [w(k+1|k),\ \cdots,\ w(k+N_p|k)]\ \textit{(Stacked reference inputs)}$$

$$H_f := \begin{bmatrix} h_1 & 0 & \cdots & 0 \\ h_2 & h_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ h_{N_p} & h_{N_p-1} & \cdots & h_1 \end{bmatrix},\ H_p := \begin{bmatrix} h_N & \cdots & h_{N-i} & \cdots & h_2 \\ 0 & \cdots & h_{N-j} & \cdots & h_3 \\ \vdots & \cdots & \ddots & \cdots & \vdots \\ 0 & \cdots & h_N & \cdots & h_{N_p+1} \end{bmatrix}.$$

Comparing the problems in (2.8) and (2.14), we can see that MAC and DMC share a similar setup, but have different tuning parameters. The influence of the tuning parameter α on the shape of reference trajectories can be illustrated by Fig. 2.1. The shape of the reference determines the speed of system responses approaching the setting point. From implementations, α is a more direct and intuitive tuning parameter than weighting matrices and prediction/control horizons. Similar to the analysis of DMC, the above discussion focuses on only SISO systems. To extend MAC to the MIMO cases, what we need to do is only replacing the stacked impulse response h_i by

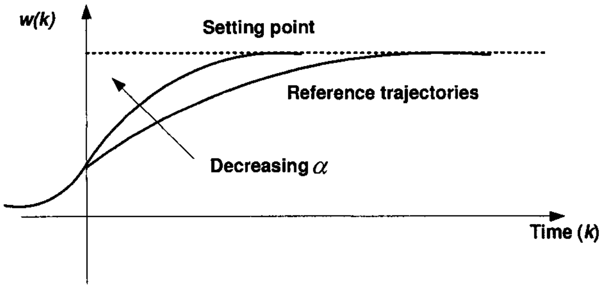


Figure 2.1: Influence of α on the reference trajectory

$$H_{stack}(i) = \begin{bmatrix} h_{11}(i) & \cdots & h_{1p}(i) \\ \vdots & \ddots & \vdots \\ h_{q1}(i) & \cdots & h_{qp}(i) \end{bmatrix}. \tag{2.15}$$

Moreover, if there are no constraints imposed on controlled processes, MAC can be solved explicitly by least square optimization, i.e.,

$$U^o_{k|k\to k+N_u-1|k} = -(H_f^T H_f + \lambda I)^{-1} H_f^T (W_{k+1\to k+N_p} - D_{past}),$$

where $D_{past} := H_p U_{k-N+1\to k-1} + \mathbf{1}\cdot d(k)$ denotes the past data.

2.3 Predictive functional control

Predictive functional control (PFC) was proposed by Richalet in 1992 for the cases of fast linear/nonlinear processes [82]. It has several distinctive characteristics superior to both DMC and MAC:

- The objective function is evaluated at coincidence points along predictive horizons.
- The manipulated inputs are constructed by a linear combinations of parameterized basis functions.
- A state space model is employed to describe process behavior.
- The reference trajectory is regulated by a linear function with the power terms of the tuning parameter α.
- It can be extended easily to nonlinear systems.

In past decades, PFC has been extensively studied and applied to various plants [83, 95, 102].

Consider a system

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ y(k) &= Cx(k), \end{aligned} \tag{2.16}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$ and $y \in \mathbb{R}^q$ stand for state, input and output, respectively. A, B, C are constant matrices with appropriate dimensions. The control policy is constructed as a linear combination of parameterized basis functions, i.e.,

$$u(k+i|k) = \sum_{j=1}^{N_B} \lambda_j(k) M_j(i), \tag{2.17}$$

where $M_j(i)$ stands for a basis function and $\lambda_j(k)$ is the corresponding coefficient over the kth horizon. "j" is the index of the basis functions and N_B is the number of the basis functions. $M_j(i)$ is always represented by polynomial functions, e.g., a set of eligible candidates are

$$M_1(i) = 1,\ M_2(i) = (i-\tau),\ \cdots,\ M_{N_B}(i) = (i-\tau)^{N_B - 1}, \tag{2.18}$$

where τ is a time constant. For simplicity, here we just focus on SISO systems, but PFC is possible to extend to MIMO systems. The objective function is evaluated at the coincidence points along a prediction horizon, i.e.,

$$J = \min_{\lambda_1(k),\ \cdots,\ \lambda_{N_B}(k)} \sum_i [y(k+i|k) - w(k+i|k)]^2, \qquad (i = N_1,\ N_2,\ \cdots,\ N_r) \tag{2.19}$$

where $(i = N_1,\ N_2,\ \cdots,\ N_r)$ is a set of coincidence points. "i" is the index of coincidence points and N_r is the largest index of coincidence points (refer to Fig. 1.3 for the definition of coincidence points). $w(k+i|k)$ is the predicted reference trajectory satisfying

$$w(k+i|k) = r - \alpha^k(r - y(k)). \tag{2.20}$$

In order to obtain smooth manipulated inputs, a quadratic factor of the form of $\mu(\Delta u(k+i|k))^2$ may enter in (2.19), similar to the objective in (2.14). From the system model in (2.16), the predicted output $y(k+i|k)$ can be expressed by the current state measurement $x(k)$ and future inputs,

$$y\,(k+i|k) = CA^i x(k) + \sum_{l=1}^{i} CA^{i-l}Bu(k+l-1|k). \tag{2.21}$$

From (2.17) and (2.18), we know that the predicted inputs are a linear combination of the basis functions $B_j(i)$. Therefore, (2.21) can be rewritten as

$$\begin{aligned} y\,(k+i|k) &= CA^i x(k) + \sum_{l=1}^{i}(CA^{i-l}B\sum_{j=1}^{N_B}\lambda_j(k)M_j(l-1)) \\ &= CA^i x(k) + \sum_{j=1}^{N_B}\lambda_j(k)y_{M_j}(i), \end{aligned} \tag{2.22}$$

where

$$y_{M_j}(i) := \sum_{l=1}^{i} CA^{i-l}BM_j(l-1), \tag{2.23}$$

is called the response of the basis function M_j. Rewriting (2.22) in the form of stacked matrices, it derives

$$y\,(k+i|k) = CA^i x(k) + Y_M(i)\Lambda(k), \tag{2.24}$$

where

$$\Lambda(k) := \begin{bmatrix}\lambda_1(k) & \cdots & \lambda_{N_B}(k)\end{bmatrix}^T,\ \ Y_M(i) := \begin{bmatrix}y_{M_1}(i) & \cdots & y_{M_{N_B}}(i)\end{bmatrix}^T$$

Inserting (2.24) and (2.20) into (2.19), the objective is recast into

$$J = \min_{\Lambda(k)} \sum_{i}^{N_r} ((1-\alpha^k)(CA^i x(k) + Y_M(i)\Lambda(k) - r))^2.$$
$$(i = N_1,\ N_2,\ \cdots,\ N_r) \tag{2.25}$$

So, if there are no input and output constraints, the explicit solution to (2.25) can be obtained by

$$\Lambda^o(k) = 2(1-\alpha^k)\mathcal{Y}^T\mathcal{Y}(\mathcal{R} - \mathcal{X}),$$

where

$$\mathcal{Y} = \begin{bmatrix} Y_M(i) \\ \vdots \\ Y_M(N_r) \end{bmatrix},\ \mathcal{R} = \begin{bmatrix} r \\ \vdots \\ r \end{bmatrix},\ \mathcal{X} = \begin{bmatrix} CA^i x(k) \\ \vdots \\ CA^{N_r} x(k) \end{bmatrix}. \tag{2.26}$$

In summary, the optimal manipulated input at instant k is

$$u^o(k) = \mathcal{M}(0)\Lambda^o(k),\ \text{where}\ \mathcal{M}(0) := \begin{bmatrix} M_1(0) & \cdots & M_{N_B}(0) \end{bmatrix}. \tag{2.27}$$

Here, we just concentrate on the cases of unconstrained PFC. Actually combining with the condition in (2.4), the constrained PFC can be also converted into a QP problem.

Conclusion 2.3 *The optimal coefficients of the basis functions in constrained PFC problems can be derived by a QP problem; namely*

$$\Lambda^o(k) := \begin{bmatrix} \lambda_1(k) & \cdots & \lambda_{N_B}(k) \end{bmatrix}^T,$$

is the solution to a QP problem:

$$J = \min_{\Lambda(k)} \sum_{i}^{N_r} ((1-\alpha^k)(CA^i x(k) + Y_M(i)\Lambda(k) - r))^2, \tag{2.28}$$
$$s.t. \qquad C^l_{y,i\to N_r}(\mathcal{X} + \mathcal{Y}\Lambda) + C^l_{u,i\to N_r}\mathcal{M}\Lambda + C^l \le 0\ (0 \le l \le N_c),$$

where the stacked matrices $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{M}$ are defined in (2.26) and (2.27). The manipulated input at instant k can be defined by

$$u^o(k) = \mathcal{M}(0)\Lambda^o(k).$$

2.4 Generalized model predictive control

The development of MPC has two branches: Deterministic MPC and Stochastic MPC [67]. All algorithms mentioned above belong to the family of deterministic MPC. In 1987, Clarke *et al.* originated GPC and it became the first stochastic MPC algorithm [21, 22]. The original version of GPC only treated stable SISO systems, but after a short time, it was extended to MIMO systems [94]. At present, GPC can even deal with uncertain systems with parameter perturbation [9]. Because extensively studying GPC is outside the scope of this thesis, this section only discusses the GPC for stable SISO systems.

Consider a system represented by an ARIMAX model [53],

$$A(q)y(k) = q^{\tau}B(q)u(k-1) + \frac{e(k)}{\Delta}, \tag{2.29}$$

where τ is the pure-delay of the system, $q = z^{-1}$ is the backward shift operator, and $\Delta = 1 - q$ is the difference operator. $y(k)$ and $u(k)$ are output and input, respectively, and $e(k)$ is the white noise with zero mean. A, B are two polynomials in terms of q,

$$\begin{aligned} A(q) &= 1 + a_1 q + a_2 q^2 + \cdots + a_{na} q^{na}, \\ B(q) &= b_0 + b_1 q + b_2 q^2 + \cdots + b_{nb} q^{nb}. \end{aligned} \tag{2.30}$$

The objective function for GPC is given by

$$J = \min_{\Delta u(1),\ \cdots,\ \Delta u(N_u)} \sum_{i=1}^{N_p} \lambda_y(i)(y(k+i|k) - w(k+i))^2 + \sum_{i=1}^{N_u} \lambda_u(i)(\Delta u(k+i-1|k))^2 \tag{2.31}$$

where $\lambda_y(i)$ and $\lambda_u(i)$ are the scalar weightings. $w(k+i)$ is the reference trajectory defined in (2.10). To solve the problem in (2.31), $y(k+i|k)$ is predicted by a set of recursive Diophantine equations, i.e., constructing a Diophantine equation pair $(M_i(q),\ N_i(q))$ by

$$1 = M_i\,(q)\,\Delta A(q) + q^i N_i(q). \tag{2.32}$$

Because the pair $(\Delta A(q),\ q^i)$ is co-prime, (2.32) uniquely determines $(M_i\,(q),\ N_i(q))$, satisfying

$$\begin{aligned} M_i\,(q) &= M_{i,0} + M_{i,1} q + M_{i,2} q^2 + \cdots + M_{i,i-1} q^{i-1}, \\ N_i(q) &= N_{i,0} + N_{i,1} q + N_{i,2} q^2 + \cdots + N_{i,\ na}\ q^{na}, \end{aligned} \tag{2.33}$$

where i is the index of recursive Diophantine equations. Note that the order of $N_i(q)$ equals to that of $A(q)$ and the order of $M_i(q)$ equals to $(i-1)$. By the method of undetermined coefficients, it is easy to obtain $M_i\,(q)$ and $N_i(q)$ from (2.32). Multiplying $M_i\,(q)\,\Delta$ to both sides of (2.29), we have

$$y(k+i|k) = N_i(q)y(k) + M_i(q)B(q)\Delta u(k+i-\tau-1|k) + M_i(q)e(k+i|k). \quad (2.34)$$

Note that the order of $M_i(q)$ is $(i-1)$, so that the last term in (2.34) expresses the future noise which is never predicable. Rewriting (2.34), the predicted output is expressed by

$$y(k+i|k) = G_i(q)\Delta u(k+i-\tau-1|k) + N_i(q)y(k). \qquad (2.35)$$

The coefficients of $G_i(q)$ and $N_i(q)$ can be obtained iteratively, i.e.,

$$g_{i+1,i+j} = g_{i,i+j} + N_{j,0}b_j \qquad (j = 0, 1, \cdots, nb),$$

where $g_{i+1,i+j}$ denotes the $(i+j)$th coefficient of the polynomial $G_{i+1}(q)$ over the $(i+1)$th iteration. The previous i coefficients of $G_{i+1}(q)$ have to be determined by the method of undetermined coefficients. Refer to [14] for the details.

Another challenge of GPC is the effect of the pure-delay factor τ. Due to the existence of τ, $\Delta u(k+i-\tau-1|k)$ can be either future data or past data. Rewriting the problem in (2.31) in the form of stacked matrices, we have

$$J = \min_{\Delta U}\ ||G(\Delta U) + Ny(k) - W||^2_{\Lambda_y^2} + ||\Delta U||^2_{\Lambda_u^2} \qquad (2.36)$$

where

$$G\ :\ = \begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_N & g_{N-1} & \cdots & g_0 \end{bmatrix},\ \hat{G} := \begin{bmatrix} (G_{\tau+1}(q) - g_0)q^{-1} \\ (G_{\tau+2}(q) - g_0 - g_1 q)q^{-2} \\ \vdots \\ (G_{\tau+N}(q) - g_0 - \cdots - g_{N-1}q^{N-1})q^{-N} \end{bmatrix},$$

$$N\ :\ = \begin{bmatrix} N_{\tau+1}(q) \\ N_{\tau+2}(q) \\ \vdots \\ N_{\tau+N}(q) \end{bmatrix},\ \Delta U := \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N-1) \end{bmatrix},\ W := \begin{bmatrix} w(k+\tau+1) \\ w(k+\tau+2) \\ \vdots \\ w(k+\tau+N) \end{bmatrix},$$

and $\Lambda_y := \text{diag}(\lambda_y(k+\tau+1), \cdots \lambda_y(k+\tau+N))$ and $\Lambda_u := \text{diag}(\lambda_u(k+\tau+1), \cdots \lambda_u(k+\tau+N))$. Eq. (2.36) is a standard QP problem; therefore, if there are no constraints imposed on the controlled system, the optimal input can be derived from the following conclusion.

Conclusion 2.4 *The unconstrained GPC problem can be solved by quadratic programming, and the optimal stacked input equals to*

$$\Delta U^o = (G^T \Lambda_y^2 G + \Lambda_u^2)^{-1} G^T \Lambda_y^2 (W - Ny(k)). \tag{2.37}$$

Because G has full column rank and both Λ_y and Λ_u are non-singular diagonalized weightings, the composition $(G^T \Lambda_y^2 G + \Lambda_u^2)$ is certainly invertible, i.e., the problem in (2.36) is always feasible.

In the above discussion, we assume that there are no auto-regressive terms of measurement disturbances, i.e., the measurement noises are assumed to be white noises. Actually GPC can be extended to the case of the systems with colored noises. Two operations are necessary for such an extension:

1. Replace the model in (2.29) by

$$\begin{aligned} A(q)y(k) &= q^{\tau} B(q)u(k-1) + \frac{C(q)}{\Delta} e(k), \\ C(q) &= 1 + c_1 q^1 + c_2 q^2 + \cdots + C_{nc} q^{nc}. \end{aligned}$$

2. The Diophantine equation pairs should be derived by

$$C(q) = M_i\,(q)\,\Delta A(q) + q^i N_i(q). \tag{2.38}$$

Because GPC is not the focus of this thesis, we will not pursue these topics here. An extensive discussion of GPC can be found in [21, 22, 14, 59].

2.5 Conclusions

This chapter surveys the conventional MPC algorithms: DMC, MAC, PFC, and GPC. These four algorithms are adopted in various commercial packages and are able to demonstrate most of the characteristics of nominal MPC.

Chapter 3

Robust model predictive control Using LMIs

Robust MPC refers to the MPC schemes which incorporate system uncertainties with the MPC formulation. With different horizon settings, robust MPC is divided into infinite horizon robust MPC (IH-RMPC) and finite horizon robust MPC (FH-RMPC). In this chapter, we introduce two RMPC schemes: IH-RMPC using linear matrix inequalities (LMIs) and FH-RMPC using LMIs. From theoretical analysis, it can be seen that the former is superior in the sense of stability and complexity, but the latter is better in flexibility and feasibility.

3.1 System uncertainties

System uncertainties have two major sources: modelling uncertainties (called as internal uncertainties in this thesis) and input/output disturbances (referred to as external uncertainties). The former is usually led by parameter perturbation and modelling mismatch, and the latter is possibly derived from measurement noises and unmeasured inputs and outputs. Fig. 3.1 shows a classical MPC feedback system corrupted by internal and external uncertainties. In the figure, the MPC block provides manipulated inputs from measured outputs and measured disturbances, but excludes unmeasured outputs and measurement noises from the MPC formulation. MPC schemes with the framework of Fig. 3.1 are called nominal MPC, which assumes system models with 100% fitness and predicted outputs with 100% accuracy. If the system in Fig. 3.1 is corrupted by serious internal or external uncertainties, MPC regulators normally can not achieve pre-specified control performance. Example 3.1 illustrates the impact of internal and external uncertainties upon closed-loop MPC dynamics.

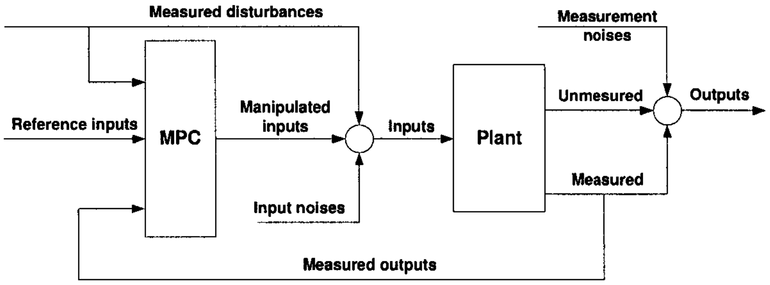


Figure 3.1: Nominal MPC systems with internal and external uncertainties.

Example 3.1 *Consider a system*

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + d(k) \end{aligned} \tag{3.1}$$

where $x \in \mathbb{R}^2$ stands for the state, $u \in \mathbb{R}^2$ the manipulated input, $y \in \mathbb{R}^2$ the output, and $d \in \mathbb{R}^2$ the disturbance. The dynamic matrix A is composed of two parts: the nominal value $\bar{A}$ and the time-varying perturbation $\Delta_A(k)$, that is $A = \bar{A} + \Delta_A(k)$

with

$$\bar{A} = \begin{bmatrix} 0.9719 & -0.0013 \\ -0.034 & 0.8628 \end{bmatrix}, \text{ and } ||\Delta_A(k)||_2 \leq 1.$$

Other parameters are given by

$$B = \begin{bmatrix} -0.0839 & 0.0232 \\ 0.0761 & 0.4144 \end{bmatrix}, \; C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; ||d(k)||_\infty \leq 1.$$

Set the initial condition equal to $x_o = [1,1]^T$. The control objective is to drive the state to converge to the origin along the state trajectories. We assume that system (3.1) is uncertainty-free and design a nominal MPC regulator, by setting $\Delta_A(k)$ and $d(k)$ equal to zeros. Fig. 3.2 illustrates the influence of horizon length on closed-loop dynamics. By increasing the difference between the prediction horizon and the control horizon, we can derive faster responses but get more aggressive inputs (N_p = 4 for the solid lines with dots and N_p = 2 for the pure solid lines; fixing $N_u \equiv 1$). Fig. 3.3 illustrates the influence of weighting matrices on closed-loop dynamics. For solid lines, we set

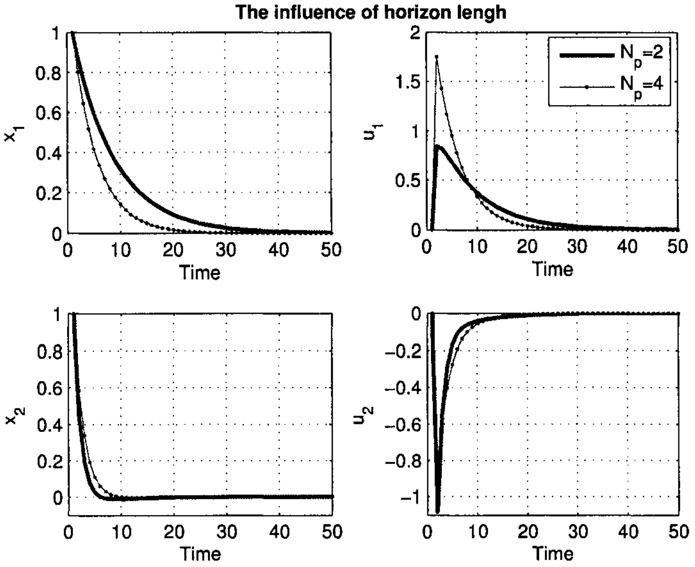


Figure 3.2: The influence of horizon length

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } R = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}.$$

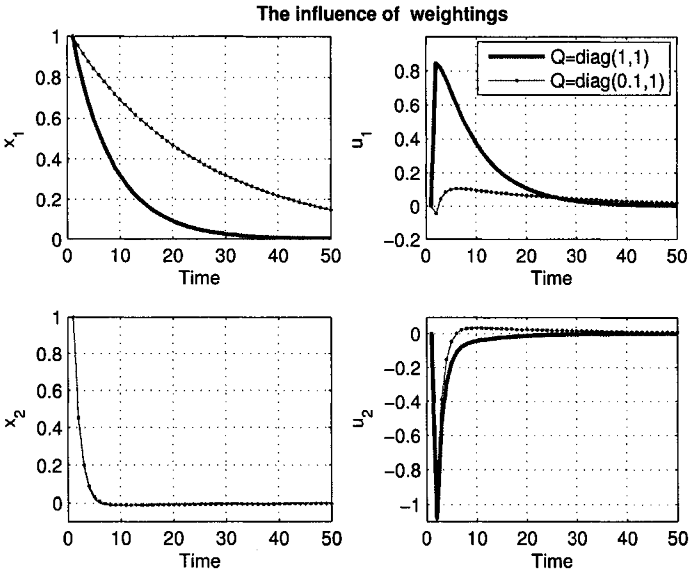


Figure 3.3: The influence of weighting matrices

and for solid lines with dots, we set

$$Q = \begin{bmatrix} 0.1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } R = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}.$$

It can be seen that the weighting element on x_1 is reduced and the trajectory of x_1 becomes much slower. Therefore, by tuning the weighting matrices, we can affect the penalized variables in objective functions.

To demonstrate the influence of system uncertainties, we design another two nominal MPC controllers with different settings: 1) use the MATLAB function "rand" to simulate the external uncertainty $d(k)$ and set $\Delta_A(k) \equiv 0$, and 2) use "rand" to create the internal uncertainty $\Delta_A(k)$ and keeps $d(k) \equiv 0$. The closed-loop responses with $d(k) \neq 0$ and $\Delta_A(k) \equiv 0$ are shown in Fig. 3.4. It can be seen that both the state trajectories and manipulated inputs are corrupted by noise and can not approach steady states, although the magnitudes of the state and input vibrations are not very large. This fact is consistent with the experience of MPC applications: if the system is impaired by small external disturbance, nominal MPC may still work, but not with large disturbances. Setting $\Delta_A(k) \neq 0$ and $d(k) \equiv 0$, Fig. 3.5 shows the trajectories of the states and inputs in the presence of internal

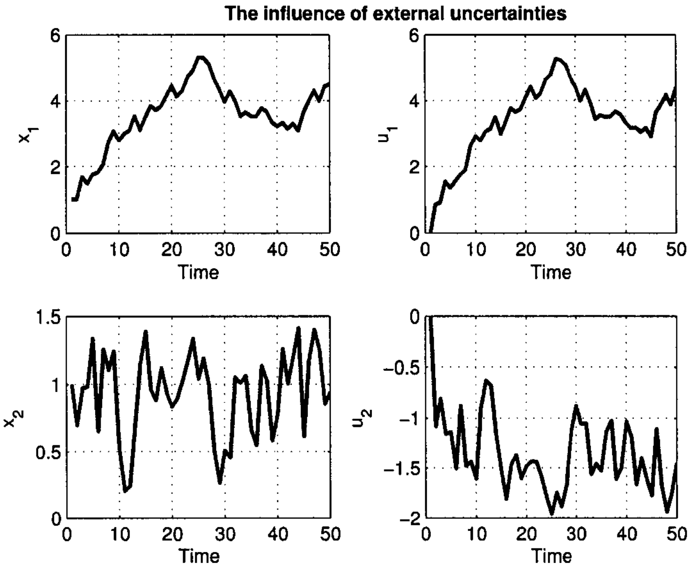


Figure 3.4: The influence of external uncertainties

disturbances. In this case, all trajectories become divergent and approach to infinity after 50 *seconds simulation. So nominal MPC has very poor robustness for internal uncertainties.*

From Example 3.1, we can draw the following two conclusions.

Conclusion 3.1 *By tuning horizon length and weighting matrices in objective functions, we can adjust the tradeoff between the aggressiveness and conservativeness of MPC design. This fact is consistent with the conclusions in Chapter 1.*

Conclusion 3.2 *Nominal MPC is not for uncertain systems with large internal uncertainties or external uncertainties. To improve MPC flexibility, it is necessary to incorporate system uncertainties with MPC formulation, i.e., developing a new class of MPC schemes, robust MPC (RMPC).*

3.2 Linear matrix inequality

A linear matrix inequality (LMI) has the form

$$F(x) = F_0 + \sum_{i=1}^{n} x_i F_i \geq 0, \tag{3.2}$$

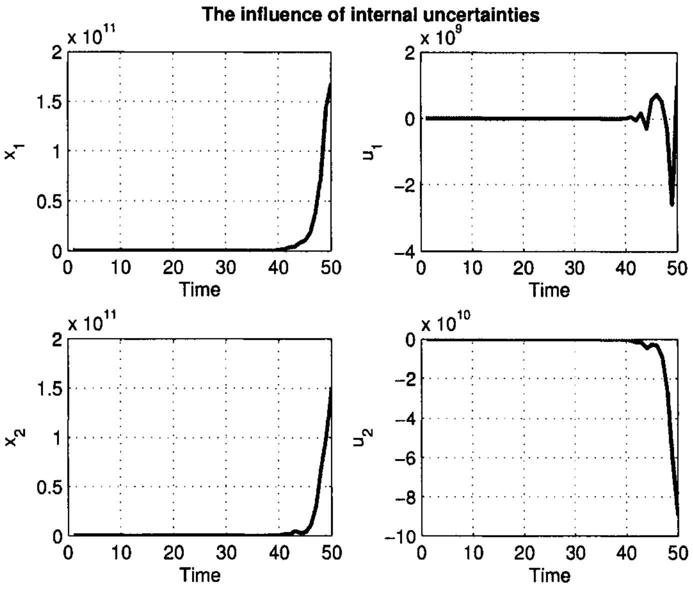


Figure 3.5: The influence of internal uncertainties

where $x \in \mathbb{R}^n$ is the unknown variable and $F_i \in \mathbb{S}^n$ $(i = 1, \cdots, n)$, namely, F_i is symmetric is the given matrix. Eq. (3.2) is equivalent to $v^T F(x) v \geq 0$, for $\forall\ v \in \mathbb{R}^n$ and $v \neq 0$. Actually, the condition in (3.2) is satisfied if and only if all eigenvalues of $F(x)$ are nonnegative, i.e.,

$$\lambda_{\min}(F(x)) \geq 0.$$

LMIs are widely used in system analysis and control [12]. Given a linear objective, LMIs define a set of optimization problems, namely semi-definite programming (SDP) which is extensively used in this chapter. Both IH-RMPC and FH-RMPC are possibly formulated as an SDP problem in the form

$$\begin{aligned} & \min\ c^T x \\ s.t. \quad & F_0 + \sum_{i=1}^{n} x_i F_i \geq 0, \end{aligned} \tag{3.3}$$

where c is a constant vector with an appropriate dimension.

3.2.1 Variants of SDP

Several popular convex optimization problems can be converted into SDP [100].

- Linear programming (LP):

$$\begin{aligned} \min_{x} \quad & c^T x \\ s.t. \quad & Ax \preceq 0. \end{aligned}$$

- Second-order cone programming (SOCP) [54]:

$$\begin{aligned} \min_{x} \quad & c^T x \\ s.t. \quad & ||A_i x + b_i||_2 \leq d_i^T x + f_i, \ (i = 1, 2, \cdots, n). \end{aligned}$$

- Convex quadratic programming (CQP) [13]:

$$\begin{aligned} \min_{x} \quad & x^T Q_0 x + b_0^T x + c_0 \ \ (Q_0 \geq 0) \\ s.t. \quad & x^T Q_i x + b_i^T x + c_i \leq 0 \ (i = 1, 2, \cdots, n, \text{ and } Q_i \geq 0). \end{aligned}$$

- Minimal generalized eigenvalue programming (GEVP) [12]:

$$\begin{aligned} \min_{x,\ \lambda} \quad & \lambda \ (\lambda > 0) \\ s.t. \quad & F_1(x) < \lambda F_2(x) \end{aligned}$$

  where $F_1(x)$ and $F_2(x)$ are two LMIs.

- Maximal determinant programming (MAXDET) [101]:

$$\begin{aligned} \max_{P,\ t_1,\ t_2} \quad & \det(P) \\ s.t. \quad & P \leq t_1 P_1 \ (P_1 \geq 0), \\ & P \leq t_2 P_2 \ (P_2 \geq 0), \\ & 0 \leq t_1 \leq 1, \\ & 0 \leq t_2 \leq 1. \end{aligned}$$

Both GEVP and MAXDET will be used in Chapter 6 for robust observer design and Example 3.2 gives detailed explanation for MAXDET problems.

Example 3.2 *[57] Given two ellipsoidal sets*

$$\begin{aligned}\mathcal{E}_1 &= \{x|x^TP_1x \leq 1,\ P_1 \geq 0\},\\ \mathcal{E}_2 &= \{x|x^TP_2x \leq 1,\ P_2 \geq 0\},\end{aligned}$$

find another ellipsoidal set $\mathcal{E}=\{x|x^TPx \leq 1,\ P_1 \geq 0\}$ *with the smallest possible volume that contains the union of* $\mathcal{E}_1$ *and* $\mathcal{E}_2$*. This is a standard MAXDET problem.*

3.2.2 LMI lemmas

From the following LMI lemmas, the above convex optimization problems can be converted into SDP, and then by utilizing the recently developed strategy, interior-point programming [69], SDP can be solved numerically and efficiently.

Lemma 3.1 *[12](Schur complements) The linear matrix inequality*

$$\begin{bmatrix} Q(x) & S(x) \\ S^T(x) & R(x)\end{bmatrix} > 0, \tag{3.4}$$

where $Q(x) = Q^T(x),\ R(x) = R^T(x)$*, and* $S(x)$ *are affine functions of* x*, is equivalent to*

$$R(x) > 0,\ Q(x) - S(x)R^{-1}(x)S^T(x) > 0.$$

Proof: Performing congruent transformations, we have

$$\begin{aligned}&\begin{bmatrix} I & -S(x)R^{-1}(x) \\ 0 & I\end{bmatrix}\begin{bmatrix} Q(x) & S(x) \\ S^T(x) & R(x)\end{bmatrix}\begin{bmatrix} I & 0 \\ -R^{-1}(x)S^T(x) & I\end{bmatrix}\\ = &\begin{bmatrix} Q(x) - S(x)R^{-1}(x)S^T(x) & 0 \\ 0 & R(x)\end{bmatrix}.\end{aligned}$$

Since

$$\det\begin{bmatrix} I & -S(x)R^{-1}(x) \\ 0 & I\end{bmatrix} = 1 \neq 0,$$

Eq. (3.4) is satisfied if and only if

$$\begin{bmatrix} Q(x) - S(x)R^{-1}(x)S^T(x) & 0 \\ 0 & R(x)\end{bmatrix} > 0.$$

Therefore Lemma 3.1 is proven. ■

Lemma 3.2 *[12](S-Procedure) Let* $F_0, \cdots, F_p$ *be quadratic functions of the variable* $\xi \in \mathbb{R}^n$ *and*

$$F_i(\xi) := \xi^TT_i\xi + 2u_i^T\xi + v_i,\ i = 0,\ 1,\ \cdots,\ p, \tag{3.5}$$

where $T_i \in \mathbb{S}^n$, u_i and v_i are all constant matrices. The condition on $F_0, \cdots, F_p$,

$$F_0(\xi) \geq 0 \text{ for } \forall \xi \text{ such that } F_i(\xi) \geq 0 \; (i = 1, \cdots, p),$$

is satisfied if $\exists\ \tau_1 \geq 0, \cdots, \tau_p \geq 0$ such that for $\forall \xi$

$$F_0(\xi) - \sum_{i=1}^{p} \tau_i F_i(\xi) \geq 0.$$

Proof: The proof is straightforward and is omitted here.

Lemma 3.3 *[106] Let X, Y be real constant matrices of compatible dimensions. Then*

$$X^T Y + Y^T X \leq \varepsilon X^T X + \frac{1}{\varepsilon} Y^T Y \tag{3.6}$$

holds for any $\varepsilon > 0$.

Proof: The proof follows from the condition

$$(\sqrt{\varepsilon} X^T - \frac{1}{\sqrt{\varepsilon}} Y^T)(\sqrt{\varepsilon} X - \frac{1}{\sqrt{\varepsilon}} Y) \geq 0.$$

■

Lemma 3.4 *[39](Robust LMIs) Let $T_1 = T_1^T$, T_2, T_3, and T_4 be real matrices of appropriate dimensions. Then* $\det(I - T_4\Delta) \neq 0$ *and*

$$T_1 + T_2 \Delta \left(I - T_4 \Delta\right)^{-1} T_3 + T_3^T \left(I - T_4 \Delta\right)^{-T} \Delta^T T_2^T \geq 0 \tag{3.7}$$

for every Δ, $\|\Delta\| = \bar{\sigma}(\Delta) \leq 1$, if and only if $\|T_4\| \leq 1$ and there exists a scalar $\tau \geq 0$ such that

$$\begin{bmatrix} T_1 - \tau T_2 T_2^T & T_3^T - \tau T_2 T_4^T \\ T_3 - \tau T_4 T_2^T & \tau\left(I - T_4 T_4^T\right) \end{bmatrix} \geq 0.$$

Proof. Let T_2 and T_3 be non-zero (the proof is straightforward if either of them is zero). Pre- and post-multiplying z^T and z to each term in (3.7), we have

$$z^T T_1 z + z^T T_2 \Delta \left(I - T_4 \Delta\right)^{-1} T_3 z + z^T T_3^T \left(I - T_4 \Delta\right)^{-T} \Delta^T T_2^T z \geq 0, \tag{3.8}$$

where z is a non-zero vector with an appropriate dimension. Define

$$\xi := \left(I - T_4 \Delta\right)^{-T} \Delta^T T_2^T z. \tag{3.9}$$

Then (3.8) can be rewritten as

$$\begin{bmatrix} z \\ \xi \end{bmatrix}^T \begin{bmatrix} T_1 & T_3^T \\ T_3 & 0 \end{bmatrix} \begin{bmatrix} z \\ \xi \end{bmatrix} \geq 0. \tag{3.10}$$

Pre-multiplying both sides of (3.9) by $(I - T_4\Delta)^T$, we get

$$\xi = \Delta^T \left(T_4\xi + T_2^T z \right).$$

For simplicity, set $p = T_4\xi + T_2^T z$ and consequently $\xi = \Delta^T p$. Then from the condition $\|\Delta\| = \bar{\sigma}(\Delta) \leq 1$, we derive

$$\xi^T \xi = p^T \Delta \Delta^T p \leq p^T p.$$

Thus

$$\left(T_4\xi + T_2^T z \right)^T \left(T_4\xi + T_2^T z \right) - \xi^T \xi \geq 0,$$

equivalently,

$$\begin{bmatrix} z \\ \xi \end{bmatrix}^T \begin{bmatrix} T_2 T_2^T & T_2 T_4^T \\ T_4 T_2^T & T_4 T_4 - I \end{bmatrix} \begin{bmatrix} z \\ \xi \end{bmatrix} \geq 0. \tag{3.11}$$

Using the S-procedure, (3.11) is satisfied if and only if

$$\begin{bmatrix} T_1 & T_3^T \\ T_3 & 0 \end{bmatrix} - \tau \begin{bmatrix} T_2 T_2^T & T_2 T_4^T \\ T_4 T_2^T & T_4 T_4 - I \end{bmatrix} \geq 0, \tag{3.12}$$

where τ is a positive scalar. The key idea of this lemma is employing τ to replace the norm-bounded uncertain matrix Δ. Simplify (3.12) to complete the proof. ■

Based on the above lemmas we can formulate both IH-RMPC and FH-RMPC as an SDP problem.

3.3 IH-RMPC using LMIs

IH-RMPC is motivated by constrained LQR (CLQR) [92]. It is different from CLQR, however, on two aspects: IH-RMPC is dynamic feedback control (CLQR is static), and IH-RMPC is able to handle system internal uncertainties (CLQR can not).

3.3.1 CLQR using LMIs

Infinite horizon MPC (IH-MPC) is an extension of CLQR and infinite horizon robust MPC (IH-RMPC) is an extension of IH-MPC. After formulating CLQR as an SDP problem, we can understand the essentials of IH-MPC and IH-RMPC using LMIs.

Consider a system

$$x(k+1) = Ax(k) + Bu(k), \tag{3.13}$$

where the state $x(k) \in \mathbb{R}^n$ and the input $u(k) \in \mathbb{R}^p$. A and B are constant matrices with compatible dimensions. Given the initial condition $x(0)$, design a control law $u(k)$ so that the state approaches the origin. The objective function in the CLQR problem can be formulated as

$$J = \sum_{k=0}^{\infty} ||x(k)||_Q^2 + ||u(k)||_R^2, \tag{3.14}$$

where $Q \in \mathbb{S}_{++}^n$ and $R \in \mathbb{S}_{++}^p$. Contrary to the conventional approach for LQR which derives an analytic solution to $u(k)$ by solving an algebraic Riccati equation (ARE), here we will use the lemmas discussed above to convert CLQR into an SDP problem. Set the control law in the form of static feedback, i.e., $u(k) = Fx(k)$ and F is a static feedback gain. Assume that there exists a matrix $P \in \mathbb{S}_{++}^n$ satisfying

$$x^T(k+i+1)Px(k+i+1) - x^T(k+i)Px(k+i) \leq -(||x(k+i)||_Q^2 + ||u(k+i)||_R^2), \tag{3.15}$$

Summing (3.15) from $i = 1$ to $i = \infty$, we have

$$x^T(\infty)Px(\infty) - x^T(0)Px(0) \leq -J. \tag{3.16}$$

If the resulting closed-loop system for (3.13) is stable, $x(\infty)$ must be zero and result in

$$J \leq x^T(0)Px(0) \leq \gamma, \tag{3.17}$$

where γ is a positive scalar and is regarded as an upper bound of the objective in (3.14),

$$\sum_{k=0}^{\infty} ||x(k)||_Q^2 + ||u(k)||_R^2 \leq \gamma.$$

Replacing $u(k)$ by $Fx(k)$, (3.15) is rewritten as

$$(A+BF)^T P(A+BF) - P + Q + F^T RF > 0. \tag{3.18}$$

Left- and right-multiplying $X := P^{-1}$ on the both sides of each term in (3.18), and then applying Schur complements, (3.18) becomes

$$\begin{bmatrix} X & * & * & * \\ AX+BY & X & * & * \\ Q^{1/2}X & 0 & I & * \\ R^{1/2}X & 0 & 0 & I \end{bmatrix} > 0, \tag{3.19}$$

where the symbol "$*$" stands for symmetric terms in the matrix. Set $Y = FX$. Applying Schur complements to (3.17) too, we derive

$$\begin{bmatrix} \gamma & * \\ x(0) & X \end{bmatrix} \geq 0. \tag{3.20}$$

Therefore, the CLQR problem is solved by

$$\begin{array}{cl} \min\limits_{\gamma,\, X,\, Y} & \gamma \\ s.t. & \text{Eqs. (3.19) and (3.20) hold,} \end{array} \tag{3.21}$$

and the feedback gain $F = YX^{-1}$. Note that here we omit a discussion on the constraints of CLQR which are easily added into Problem (3.21).

Conclusion 3.3 *From the above operation, CLQR is converted into an SDP problem in (3.21). From the condition in (3.15), the resulting CLQR feedback system is asymptotically stable if (3.21) is feasible.*

Conclusion 3.4 *The constraint in (3.19) is the function of the initial state $x(0)$. If replacing $x(0)$ by $x(k)$ and iterating the problem in (3.21), CLQR design is extended into IH-MPC.*

Conclusion 3.5 *Extending Conclusion 3.4 one step forward by incorporating system uncertainties into MPC formulation, IH-MPC becomes IH-RMPC. In this case, the objective in (3.21) degenerates to a sub-optimization problem, i.e., IH-RMPC can be formulated as*

$$\begin{array}{cl} \min\limits_{\gamma,\, X,\, Y} & \max\limits_{\Delta(k)} \gamma \\ s.t. & \textit{Eqs. (3.19) and (3.20),} \end{array} \tag{3.22}$$

where $\Delta(k)$ is the composition of internal and external uncertainties, and γ is the upper-bound of the objective in the form of (3.14).

The challenge of IH-RMPC is to develop a structured uncertainty $\Delta(k)$ in a way that captures model uncertainties and facilitates the calculation of the upper bound γ.

3.3.2 Structured system uncertainties

In 1996, Kothare *et al.* published a successful IH-RMPC algorithm [45], and in this paper two kinds of structured uncertainties were considered, namely polytopic uncertainties and structured uncertainties in the feedback loop.

1. Polytopic uncertainties.

   Consider a system

   $$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k), \\ y(k) &= Cx(k), \end{aligned} \tag{3.23}$$

   where $A(k)$, $B(k)$ stand for the time-varying dynamic matrix and input matrix, and their time-varying properties are results of modelling uncertainties. There exists a convex set Ω containing all the possibilities of $A(k)$ and $B(k)$, i.e.,

   $$\Omega = \mathrm{Co}\{[A_1, B_1],\ [A_2,\ B_2],\ \cdots,\ [A_L, B_L]\}, \tag{3.24}$$

   where "Co" denotes to the convex hull, such that if $[A(k), B(k)] \in \Omega$, then

   $$[A(k), B(k)] = \sum_{i=1}^{L} \lambda_i [A_i, B_i],\ \sum_{i=1}^{L} \lambda_i = 1 \text{ and } \lambda_i \geq 0. \tag{3.25}$$

   Ω defines a set of polytopic modelling uncertainties.

2. Structured uncertainties in the feedback loop.

   This type of modelling uncertainties can be represented by

   $$\begin{aligned} x(k+1) &= Ax(k) + B_1u(k) + B_2\theta(k), \\ y(k) &= C_1x(k) + D_1\theta(k), \\ q(k) &= C_2x(k) + D_2u(k), \\ \theta(k) &= \Delta(k)q(k), \end{aligned} \tag{3.26}$$

   where $\theta(k)$ is the unknown input due to modelling uncertainties, and $\Delta(k)$ is the system matrix for structured uncertainties, which is block-diagonalized and has all block entries norm-bounded by 1, i.e.,

   $$||\Delta_i(k)||_2 = \bar{\sigma}(\Delta_i(k)) \leq \lambda_i,\ i = 1, 2, \cdots, r,\ k \geq 0, \tag{3.27}$$

where $\bar{\sigma}(\cdot)$ is the maximum singular value and λ_i is the corresponding upper bound scalar. Fig. 3.6 depicts this framework. Consequently,

$$\sum_{j=0}^{k} \theta_i^T(j)\,\theta_i(j) \leq \sum_{j=0}^{k} q_i^T(j)\,q_i(j). \tag{3.28}$$

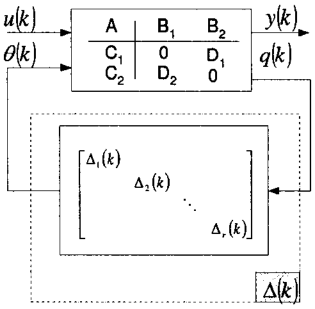


Figure 3.6: Structured uncertainties in the feedback loop.

Proved by Packard and Doyle [70], a number of control systems with modelling uncertainties can be recast in the framework of (3.26). Moreover, Reference [12] shows that the structured uncertainties in the feedback loop in (3.26) can be reformulated as the polytopic uncertainties in (3.24), according to

$$\Omega = \{[A + B_2\Delta(k)C_2, \quad B + B_2\Delta(k)D_2]\}.$$

3.3.3 Algorithms

From Conclusion 3.5, we know that the main challenge of IH-RMPC is to derive the upper bound γ in the presence of the uncertainties $\Delta(k)$. Reference [45] proposes two theorems to solve IH-RMPC problems with polytopic uncertainties and structured uncertainties in the feedback loop. The objective function is defined in the form of (3.14), i.e.,

$$\min_{u(k|k),\, \cdots,\, u(k+N_u|k)} \max_{\Delta(k)} \sum_{i=k}^{\infty} ||x(k+i|k)||_Q^2 + ||u(k+i)||_R^2. \tag{3.29}$$

Note that $N_p = \infty$, N_u is a fixed value, and when $i \geq N_u$, $u(k+i|k) \equiv u(k+N_u|k)$.

Theorem 3.1 *[45] **(IH-RMPC for systems with polytopic uncertainties)** Let the uncertainty set Ω be defined by the polytope in (3.24). Then the state feedback matrix F in the control law $u(k+i|k) = Fx(k+i|k)$ $(i \geq 0)$ that minimizes the upper bound $V(x(k|k))$ of the robust performance objective function at instant k is given by*

$$F = YX^{-1},$$

where $X > 0$ and Y are the optimal solutions (if they exist) to an SDP problem,

$$\min_{\gamma,\ X,\ Y} \quad \gamma \quad (\gamma > 0), \tag{3.30}$$

$$s.t. \quad \begin{bmatrix} \gamma & x(k|k)^T \\ x(k|k) & X \end{bmatrix} \geq 0, \tag{3.31}$$

$$\begin{bmatrix} X & * & * & * \\ A_jX + B_jY & X & * & * \\ Q^{1/2}X & 0 & I & * \\ R^{1/2}Y & 0 & 0 & I \end{bmatrix} \geq 0 \ (j = 1, 2, \cdots L). \tag{3.32}$$

Proof: In the same way to derive CLQR using LMIs, the sub-optimization of IH-RMPC can be converted into an SDP problem. Replacing the pair $(A,\ B)$ in (3.19) by $(A(k),\ B(k))$, we derive the first constraint for IH-RMPC

$$\begin{bmatrix} X & * & * & * \\ A(k)X + B(k)Y & X & * & * \\ Q^{1/2}X & 0 & I & * \\ R^{1/2}X & 0 & 0 & I \end{bmatrix} > 0. \tag{3.33}$$

Notice that the pair $(A(k),\ B(k))$ is time-varying and bounded by the convex hull Ω in (3.24). $X = P^{-1}$ and $Y = FX$ are similar to the symbols in (3.19). Obviously, any pair $(A(k),\ B(k))$ in Ω can satisfy the condition in (3.33) if and only if (3.32) is satisfied. Meanwhile, to guarantee γ is an upper bound of the objective in (3.29), the condition in (3.31) is needed. ■

Theorem 3.2 *[45] **(IH-RMPC for systems with structured uncertainties in the feedback loop)** Let the uncertainty set Ω be defined by the structured uncertainties in (3.26). Then the state feedback matrix F for IH-RMPC is given by*

$$F = YX^{-1},$$

where $X > 0$ and Y are the optimal solutions (if they exist) to an SDP problem,

$$\min_{\gamma,\ Q,\ Y,\ \Lambda} \quad \gamma \quad (\gamma > 0), \tag{3.34}$$

$$s.t. \quad \begin{bmatrix} \gamma & x\,(k|k)^T \\ x\,(k|k) & X \end{bmatrix} \geq 0, \tag{3.35}$$

$$\begin{bmatrix} Q & * & * & * & * \\ R^{1/2}Y & I & 0 & 0 & 0 \\ Q^{1/2}X & 0 & I & 0 & 0 \\ C_2X + D_2Y & 0 & 0 & \Lambda & 0 \\ AX + BY & 0 & 0 & 0 & X - B_2\Lambda B_2^T \end{bmatrix} \geq 0, \tag{3.36}$$

where

$$\Lambda = \begin{bmatrix} \lambda_1^{-1}I_{n_1} & & & \\ & \lambda_2^{-1}I_{n2} & & \\ & & \ddots & \\ & & & \lambda_2^{-1}I_{nr} \end{bmatrix} > 0. \tag{3.37}$$

Proof: Inserting the model in 3.26 into 3.15 and replacing $x(k{+}i{+}1|k)$ by $x(k{+}i|k)$, we have

$$\begin{bmatrix} x(k+i|k) \\ \theta(k+i|k) \end{bmatrix}^T \begin{bmatrix} ||A + B_1F||_P^2 - P + Q + ||F||_R^2 & * \\ B_2P(A + B_1F) & ||B_2||_P^2 \end{bmatrix} \begin{bmatrix} x(k+i|k) \\ \theta(k+i|k) \end{bmatrix}^T \leq 0. \tag{3.38}$$

Moreover, from the block diagonal uncertainties in (3.27) - (3.28), we have

$$\begin{bmatrix} x(k+i|k) \\ \theta(k+i|k) \end{bmatrix}^T \begin{bmatrix} -||C_2 + D_2F||_I^2 & * \\ 0 & I \end{bmatrix} \begin{bmatrix} x(k+i|k) \\ \theta(k+i|k) \end{bmatrix}^T \leq 0, \tag{3.39}$$

where I denotes the identity matrix with an appropriate dimension. Performing S-procedure, (3.38) is satisfied if and only if $\exists\ \lambda_1, \cdots, \lambda_r > 0$ such that

$$\begin{bmatrix} ||A + B_1F||_P^2 - P + Q + ||F||_R^2 + ||C_2 + D_2F||_\Lambda^2 & * \\ B_2P(A + B_1F) & ||B_2||_P^2 - \Lambda \end{bmatrix} \leq 0, \tag{3.40}$$

where Λ is defined in (3.37). Performing congruent transformation to (3.40) by the factor

$$\begin{bmatrix} X & 0 \\ 0 & X \end{bmatrix} \ (X := P^{-1}),$$

and then applying Schur complements to the result, we obtain (3.36). Similar to Theorem 3.1, the constraint in (3.35) is imposed for upper bounding γ. Theorem 3.2 is then proven. ■

Conclusion 3.6 *Because the summation of (3.15) goes from $i = 0$ to $i = \infty$, state prediction in the presence of uncertainties is skillfully avoided. This strategy is the key in IH-RMPC formulation. The fixed length of the prediction horizon, however, limits the tuning freedom of RMPC.*

Conclusion 3.7 *Because of the condition in (3.15), it is easy to prove that the resulting IH-RMPC system is closed-loop stable, associated with the Lyapunov function $x^T P x$ and $P = X^{-1}$. Eq. (3.15) defines an ellipsoidal invariant set,*

$$\mathcal{E}_k = \{x(k+i) \mid x(k+i)^T P^{-1} x(k+i) \leq \gamma,\ P \geq 0\}, \tag{3.41}$$

and this invariant set also guarantees the feasibility of IH-RMPC.

3.3.4 Input and output constraints

Imposing 2-norm or ∞-norm hard constraints in the problems of Theorems 3.1, 3.2, Kothare *et al.* developed constrained IH-RMPC.

- Input 2-norm constraints (energy constraints):

  Consider an input 2-norm constraint in the form of

$$||u(k+i|k)||_2 \leq u_{\max,2},\ i = 0,\ \cdots,\ N_u. \tag{3.42}$$

  Also, from (3.41), we know that the current state $x(k)$ determines an ellipsoidal invariant set,

$$\mathcal{E}_0 = \{x \mid x(k+i)^T X x(k+i) \leq \gamma\}. \tag{3.43}$$

  Therefore,

$$\begin{aligned} \max_{0 \leq i \leq N_u} ||u(k+i|k)||_2^2 &= \max_i ||Fx(k+i|k)||_2^2 \\ &\leq \bar{\sigma}^2(YX^{-1/2})\gamma^2, \end{aligned} \tag{3.44}$$

  From (3.42) and (3.44), the input 2-norm constraint in (3.42) is rewritten as

$$u_{\max,2}^2 I - (\gamma Y)^T X^{-1} (\gamma Y) \geq 0. \tag{3.45}$$

  Applying Schur complements, (3.45) is equivalent to

$$\begin{bmatrix} u_{\max,2}^2 I & * \\ \gamma Y & X \end{bmatrix} \geq 0, \tag{3.46}$$

  which is an LMI and easily combined with Problem (3.30) or (3.34).

- Input ∞-norm constraints (peak constraints):

  Consider an input ∞-norm constraints in the form of

$$\max_{i,\ j} |u_j(k+i|k)| \le u^j_{\max,\infty},\ i = 0,\ \cdots,\ N_u,\ j = 1,\ \cdots,\ p. \tag{3.47}$$

  Replacing u by the feedback gain $F = YX^{-1}$, we have

$$\begin{aligned} \max_{i,\ j} |u_j(k+i|k)|^2 &= \max_i |(YX^{-1}x(k+i|k))_j|^2 \\ &= \max_i |(YX^{-1/2})_j(X^{-1/2}x(k+i|k))|^2 \\ &\le (Y^TX^{-1}Y)_{jj}\gamma^2, \end{aligned} \tag{3.48}$$

  where the notation $(\cdot)_j$ stands for the jth row of a matrix, and $(\cdot)_{jj}$ is the (jj)th element. From (3.47) and (3.48), the input ∞-norm constraint in (3.47) is rewritten as

$$Z - \gamma^2 Y^T X^{-1} Y \ge 0, \text{ with } (Z)_{jj} \le u^j_{\max,\infty}, \tag{3.49}$$

  where $Z \in \mathbb{S}^p_+$ is an unknown matrix. Applying Schur complements, (3.49) is equivalent to

$$\begin{bmatrix} Z & * \\ \gamma Y & X \end{bmatrix} \ge 0, \text{ with } (Z)_{jj} \le u^j_{\max,\infty}, \tag{3.50}$$

  which is an LMI constraint.

- Output 2-norm constraints (energy constraints):

  Here, we only consider the case with polytopic uncertainties. It is easy to extend the results to cases with structured uncertainties in the feedback loop.

  Consider an output 2-norm constraint

$$||y(k+i|k)||_2 \le y_{\max,2},\ i = 1,\ \cdots,\ \infty. \tag{3.51}$$

  Because of the prediction horizon $N_p = \infty$, the condition in (3.51) is satisfied if and only if

$$||y(k+i|k)||_2 \le y_{\max,2}, \tag{3.52}$$

  and

$$||y(k+i+1|k)||_2 \le y_{\max,2}. \tag{3.53}$$

It is obvious that (3.52) is equivalent to

$$\begin{bmatrix} u^2_{\max,2} I & * \\ \gamma C & X \end{bmatrix} \geq 0. \tag{3.54}$$

From the model in (3.23), (3.53) can be rewritten as

$$\begin{bmatrix} y^2_{\max,2} I & * \\ \gamma C(A(k)X + B(k)Y) & X \end{bmatrix} \geq 0. \tag{3.55}$$

Obviously, any pair $[A(k),\ B(k))]$ in the convex hull Ω guarantees the condition in (3.55) if and only if

$$\begin{bmatrix} y^2_{\max,2} I & * \\ \gamma C(A_i X + B_i Y) & X \end{bmatrix} \geq 0,\ i = 1,\ 2,\ \cdots,\ L, \tag{3.56}$$

which form a set of LMI constraints.

- Output ∞-norm constraints (peak constraints):

  Similar to output 2-norm constraints, here we only consider cases with polytopic uncertainties.

  Consider an output ∞-norm constraint

$$\max_{i,\ j} |y_j(k+i|k)| \leq y^j_{\max,\infty},\ i = 0,\ \cdots,\ \infty,\ j = 1,\ \cdots,\ q. \tag{3.57}$$

  Similar to the analysis for output 2-norm constraints, (3.57) implies two LMI conditions,

$$\begin{bmatrix} Z_1 & * \\ \gamma C & X \end{bmatrix} \geq 0,\ \text{with } (Z_1)_{jj} \leq y^j_{\max,\infty}, \tag{3.58}$$

$$\begin{bmatrix} Z_2 & * \\ \gamma C(A_i X + B_i Y) & X \end{bmatrix} \geq 0,$$
$$\text{with } (Z_1)_{jj} \leq y^j_{\max,\infty},\ i = 1,\ \cdots,\ L, \tag{3.59}$$

  where $Z_1,\ Z_2 \in \mathbb{S}^p_+$ are the unknown matrices to be penalized in the objective in (3.30).

Corollary 3.1 *For systems with polytopic uncertainties in (3.23), the optimal input $u(k|k)$ of constrained IH-RMPC with both the input/output energy constraints and input/output peak constraints, can be solved by the SDP problem in (3.30) with the* *additional LMI constraints in (3.46), (3.50), (3.54), (3.56), (3.58), and (3.59). X, Y, γ, Z, Z_1, and Z_2 are the optimization variables, and*

$$u(k|k) = YX^{-1}x(k),$$

where $x(k)$ is the current state measurement.

Corollary 3.2 *For systems with structured uncertainties of the form (3.26), the optimal input $u(k|k)$ of constrained IH-RMPC with both input energy constraints and input peak constraints, can be solved by the SDP problem in (3.30) with the additional LMI constraints in (3.46) and (3.50). X, Y, γ, Λ and Z are the optimization variables, and*

$$u(k|k) = YX^{-1}x(k),$$

where $x(k)$ is the current state measurement.

3.4 FH-RMPC using LMIs

To preserve the numerical efficiency of LMIs and improve the tuning freedom of IH-RMPC, finite horizon robust model predictive control (FH-RMPC) using LMIs is developed. A moving average system matrix [16] is used to capture modelling uncertainties and facilitate future state prediction. Two additional terminal cost constraints in the form of LMIs are constructed to guarantee the closed-loop stability of FH-RMPC. Besides the horizons N_p, N_u, the terminal weighting Q_{N_p} (another tuning parameter) is constructed to adjust the tradeoff between closed-loop stability and resulting dynamics. The robust LMI theorem [35, 56], namely Lemma 3.4, is utilized in the FH-RMPC formulation. The moving average system matrix, called uncertainty block, is weighted and norm-bounded by one, which is consistent with the conditions of the robust LMI theorem. Paralleling the system nominal model with the uncertainty block, we develop an FH-RMPC framework, which reflects the influence of high order uncertain terms on the FH-RMPC formulation and facilitates state predictions as well. From the properties of robust LMIs, FH-RMPC using LMIs is finally recast into an SDP problem and solved numerically using several existing software packages, e.g., MATLAB LMI-Toolbox [32].

3.4.1 Finite horizon nominal MPC using LMIs

Consider a nominal model,

$$x\left(k+1\right)=Ax\left(k\right)+Bu\left(k\right),\ y\left(k\right)=Cx\left(k\right), \tag{3.60}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the input vector and $y \in \mathbb{R}^q$ is the output vector. A, B, and C are constant matrices of compatible dimensions. To obtain the nominal MPC for step tracking, the objective function of input $u\left(\cdot|k\right)$ and state measurement $x\left(k\right)$ over a horizon window is defined by

$$J=\sum_{i=1}^{N_p-1}||r-y(k+i|k)||_Q^2+\sum_{i=0}^{N_u-1}||u\left(k+i|k\right)||_R^2+||r-y\left(k+N_p|k\right)||_{Q_{N_p}}^2, \tag{3.61}$$

where r is the reference input, and Q, R, Q_{N_p} are the output, input and terminal weightings, respectively. Based on the model in (3.60), the predicted states can be expressed by:

$$x\left(k+i|k\right) = \begin{cases} A^ix\left(k\right)+A^{i-1}Bu\left(k|k\right)+\cdots+Bu\left(k+i-1|k\right), \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{if } 1\leq i\leq N_u, \\ A^ix\left(k\right)+A^{i-1}Bu\left(k|k\right)+\cdots+A^{i-N_u+1}Bu\left(k+N_u-2|k\right) \\ +\left(A^{i-N_u}B+\cdots+B\right)u\left(k+N_u-1|k\right), \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{if } N_u< i\leq N_p. \end{cases} \tag{3.62}$$

Rewrite the objective function in (3.61) in the form of augmented matrices [59] and derive

$$J=\ \left(\mathcal{R}-\mathcal{Y}\left(k\right)\right)^T\mathcal{Q}\left(\mathcal{R}-\mathcal{Y}\left(k\right)\right)+\mathcal{U}^T\left(k\right)\mathcal{R}\mathcal{U}\left(k\right), \tag{3.63}$$

where the augmented vectors are given by

$$\begin{aligned} \mathcal{U}\left(k\right) &= [\ u^T\left(k|k\right)\ \ u^T\left(k+1|k\right)\ \ \cdots\ \ u^T\left(k+N_u-1|k\right)\ ]^T, \\ \mathcal{Y}\left(k\right) &= [\ y^T\left(k+1|k\right)\ \ y^T\left(k+2|k\right)\ \ \cdots\ \ y^T\left(k+N_p|k\right)\ ]^T, \\ \mathcal{T} &= [\ r^T\ \ r^T\ \ \cdots\ \ r^T\ ]^T, \end{aligned} \tag{3.64}$$

and the augmented weightings are given by

$$\mathcal{Q}=\operatorname{diag}(Q,Q,\cdots,Q,Q_{N_p}),\ \mathcal{R}=\operatorname{diag}(R,R,\cdots,R). \tag{3.65}$$

Inserting the predicted states in (3.62) into (3.60) from $i=1$ to $i=N_p$, and utilizing the augmented vectors and weightings in (3.64) and (3.65), we can express the predicted output sequence $\mathcal{Y}(k)$ in terms of the current state $x(k)$,

$$\mathcal{Y}(k) = \mathcal{C}\mathcal{A}x(k) + \mathcal{C}\mathcal{B}\mathcal{U}(k), \tag{3.66}$$

where

$$\begin{aligned}
\mathcal{A} &= \begin{bmatrix} A \\ \vdots \\ A^{N_u} \\ \vdots \\ A^{N_p} \end{bmatrix}, \mathcal{B} = \begin{bmatrix} B & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ A^{N_u-1}B & A^{N_u-2}B & \cdots & B \\ \vdots & \vdots & \vdots & \vdots \\ A^{N_p-1}B & A^{N_p-2}B & \cdots & \begin{matrix}(A^{N_p-N_u}B \\ +\cdots+B)\end{matrix} \end{bmatrix}, \\
\mathcal{C} &= \begin{bmatrix} C & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C \end{bmatrix}.
\end{aligned} \tag{3.67}$$

Substituting (3.66) into (3.63), and defining an auxiliary positive scalar t, the nominal MPC can be solved by minimizing a linear objective,

$$\begin{aligned}
& J_o = \min_{t,\, \mathcal{U}(k)} \; t \\
s.t. \quad & t \geq J, \\
& J = (\mathcal{T} - (\mathcal{C}\mathcal{A}x(k) + \mathcal{C}\mathcal{B}\mathcal{U}(k)))^T \mathcal{Q} \\
& \quad \cdot(\mathcal{T} - (\mathcal{C}\mathcal{A}x(k) + \mathcal{C}\mathcal{B}\mathcal{U}(k))) + \mathcal{U}^T(k)\mathcal{R}\mathcal{U}(k),
\end{aligned} \tag{3.68}$$

where J_o is the optimal value of the objective J and the scalar t is an upper bound of J. Applying Schur complements to the constraint in (3.68), we convert the nominal MPC into a SDP problem.

Conclusion 3.8 *For nominal MPC with step-tracking, the optimal control sequence $\mathcal{U}(k)$ over a horizon starting at instant k, if exists, can be calculated by solving an SDP problem,*

$$\begin{aligned}
& J_o = \min_{t,\, \mathcal{U}(k)} \; t \\
s.t. \quad & t > 0, \\
& \begin{bmatrix} t & (\mathcal{T} - (\mathcal{C}\mathcal{A}x(k) + \mathcal{C}\mathcal{B}\mathcal{U}(k)))^T & \mathcal{U}^T(k) \\ * & \mathcal{Q}^{-1} & 0 \\ \mathcal{U}(k) & 0 & \mathcal{R}^{-1} \end{bmatrix} \geq 0,
\end{aligned} \tag{3.69}$$

where $x(k)$ is the state measurement at instant k.

3.4.2 Finite horizon robust MPC using LMIs

As in Section 3.3, the first step in the robust MPC synthesis is to configure a system framework to represent the influence of modelling uncertainties on controller design while capturing system dynamics. FH-RMPC sets both the prediction horizon N_p and the control horizons N_u by finite integers, so it becomes inevitable to perform state/output predictions. Eq. (3.62) provides an approach to future state calculation of nominal MPC systems. In the same fashion, we can perform state predictions in the presence of modelling uncertainties. However, if there exist uncertain terms in matrix A, the high order factors of uncertainties will appear in the expression of predicted states, which are notorious for the MPC formulation. This barrier motivates us to construct a new framework to represent the uncertain factors in matrix A: the nominal version of controlled systems paralleling a moving average uncertain matrix.

1. **A framework for modelling uncertainties**

   Fig. 3.7 shows the framework adopted by FH-RMPC. It is composed of the nominal model of the controlled system and a moving average uncertain matrix. Here we assume that C is known precisely and the states are fully measurable, so that the system is regarded as a transformation from inputs to states and then to outputs. In Fig. 3.7, Δ_k stands for the modelling uncertainties over the prediction horizon starting at instant k. It is weighted and

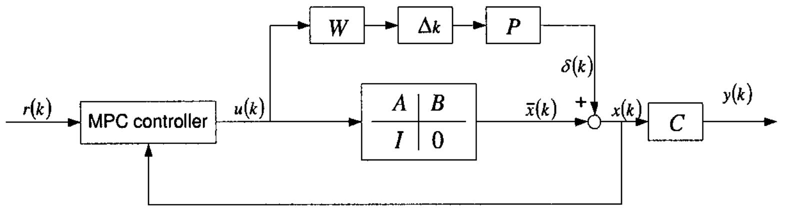


Figure 3.7: An FH-RMPC feedback system

norm-bounded by one, and W and P are weighting matrices, i.e.,

$$\Delta_k = \begin{bmatrix} \Delta_k(k,k) & 0 & \cdots & 0 \\ \Delta_k(k+1,k) & \Delta_k(k+1,k+1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \Delta_k(k+N_p,k) & \Delta_k(k+N_p,k+1) & \cdots & \Delta_k(k+N_p,k+N_p) \end{bmatrix}, \tag{3.70}$$

with $\|\Delta_k\| = \bar{\sigma}(\Delta_k) \leq 1$. To simplify formulation, we assume that predicted state $x(k+i|k)$ is independent of the previous modelling uncertainties due to the monotonicity of the prediction horizon. Taking advantages of such an assumption, the controller design may be significantly simplified .

2. **Convert FH-RMPC into a QP problem**

   Based on the uncertainty block defined in (3.70), perform the state predictions. The key point here is to exploit the monotonicity of the prediction horizon. At every prediction horizon starting at instant k, predictions are independent of the previous horizon uncertainty block Δ_{k-1}. Here the nominal model is given by

$$\bar{x}(k+i+1|k) = A\bar{x}(k+i|k) + Bu(k+i|k), \tag{3.71}$$

   and the uncertain term $\delta(k)$ led by modelling uncertainties can be calculated by

$$\delta(k+i|k) = \sum_{j=k}^{k+i} \hat{\Delta}(k+i,j)\, u(j|k), \tag{3.72}$$

   where the uncertainty matrix $\hat{\Delta}$ is defined, for convenience, as

$$\hat{\Delta} = P\Delta_k W.$$

   From (3.71) and (3.70), we have

$$\begin{aligned} x(k+i+1|k) &= \bar{x}(k+i+1|k) + \delta(k+1+i|k) \\ &= A\bar{x}(k+i|k) + Bu(k+i|k) \\ &\quad + \sum_{j=k}^{k+1+i} \hat{\Delta}(k+1+i,j)\, u(j|k). \end{aligned} \tag{3.73}$$

It is obvious that

$$\begin{aligned} x\left(k+i|k\right) &= \bar{x}\left(k+i|k\right)+\delta\left(k+i|k\right) \\ &= \bar{x}\left(k+i|k\right)+\sum_{j=k}^{k+i}\hat{\Delta}\left(k+i,j\right)u\left(j|k\right). \end{aligned} \tag{3.74}$$

Substituting $\bar{x}\left(k+i|k\right)$ in (3.74) into (3.73), we derive

$$\begin{aligned} x\left(k+1+i|k\right) &= Ax\left(k+i|k\right)+Bu\left(k+i|k\right)+\sum_{j=k}^{k+1+i}\hat{\Delta}\left(k+1+i,j\right)u\left(j|k\right) \\ &\quad -A\sum_{j=k}^{k+i}\hat{\Delta}\left(k+i,j\right)u\left(j|k\right). \end{aligned} \tag{3.75}$$

The predicted output satisfies

$$y\left(k+i|k\right)=Cx\left(k+i|k\right). \tag{3.76}$$

To illustrate the procedure of the state predictions, we implement the first two steps, namely the calculations of $x(k+1|k)$ and $x(k+2|k)$,

$$\begin{aligned} x(k+1|k) &= Ax(k)+Bu(k|k)+\sum_{j=k}^{k+1}\hat{\Delta}(k+1,j)u(j|k) \\ &\quad -A\hat{\Delta}(k,k)u(k|k), \end{aligned} \tag{3.77}$$

$$\begin{aligned} x(k+2|k) &= Ax(k+1|k)+Bu(k+1|k) \\ &\quad +\sum_{j=k}^{k+2}\hat{\Delta}(k+2,j)u(j|k)-A\sum_{j=k}^{k+1}\hat{\Delta}(k+1,j)u(j|k). \end{aligned} \tag{3.78}$$

Substituting (3.77) into (3.78), we have

$$\begin{aligned} x(k+2|k) &= A^2x(k)+ABu(k|k)+Bu(k+1|k) \\ &\quad +\sum_{j=k}^{k+2}\hat{\Delta}(k+2,i)u(j|k)-A^2\hat{\Delta}(k,k)u(k|k). \end{aligned} \tag{3.79}$$

Without loss of generality, we can assume that the uncertainty block Δ_k is strictly causal, hence the first element of uncertainty block $\Delta_k\left(k,k\right)=0$, consequently, $\hat{\Delta}\left(k,k\right)=0$ (weightings P and W are block diagonal matrices).

So we can derive the common expression for the predicted state:

$$x(k+i|k) = \begin{cases} A^{i}x(k) + A^{i-1}Bu(k|k) + \cdots + Bu(k+i-1|k) \\ + \sum\limits_{j=k}^{k+i} \hat{\Delta}(k+i,j)\,u(j|k), \qquad \text{if } 1 \le i \le N_u - 1, \\ \\ A^{i}x(k) + A^{i-1}Bu(k|k) + \cdots \\ + A^{i-N_u+1}Bu(k+N_u-2|k) + \cdots \\ + \left(A^{i-N_u}B + \cdots + B\right)u(k+N_u-1|k) \\ + \sum\limits_{j=k}^{k+N_u-1} \hat{\Delta}(k+i,j)\,u(j|k) \\ + \sum\limits_{j=k+N_u}^{k+i} \hat{\Delta}(k+i,j)\,u(k+N_u-1|k), \text{ if } N_u \le i \le N_p. \end{cases} \tag{3.80}$$

Rewrite the predicted states as an augmented matrix

$$\begin{bmatrix} x(k+1|k) \\ \vdots \\ x(k+N_u|k) \\ \vdots \\ x(k+N_p|k) \end{bmatrix} = \begin{bmatrix} A \\ \vdots \\ A^{N_u} \\ \vdots \\ A^{N_p} \end{bmatrix} x(k) + \begin{bmatrix} B & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ A^{N_u-1}B & \cdots & AB & B \\ \vdots & \cdots & \vdots & \vdots \\ A^{N_p-1}B & \cdots & A^{N_p-N_u+1}B & A^{N_p-N_u}B + \cdots + B \end{bmatrix} \begin{bmatrix} u(k|k) \\ \vdots \\ u(k+N_u-1|k) \end{bmatrix}$$
$$+ \begin{bmatrix} \hat{\Delta}(k+1,k) & \hat{\Delta}(k+1,k+1) & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ \hat{\Delta}(k+N_u,k) & \hat{\Delta}(k+N_u,k+1) & \cdots & \hat{\Delta}(k+N_u,k+N_u-1) + \cdots + \hat{\Delta}(k+N_u,k+N_u) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\Delta}(k+N_p,k) & \hat{\Delta}(k+N_p,k+1) & \cdots & \hat{\Delta}(k+N_u,k+N_u-1) + \cdots + \hat{\Delta}(k+N_p,k+N_p) \end{bmatrix}$$
$$\times \begin{bmatrix} u(k|k) \\ \vdots \\ u(k+N_u-1|k) \end{bmatrix}. \tag{3.81}$$

Here we define two auxiliary matrices M_l and M_r as the left- and right-multipliers of the uncertainty block $\hat{\Delta}$, namely

$$M_l := \begin{bmatrix} 0 & I_1 & 0 & \cdots & 0 \\ 0 & 0 & I_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I_1 \end{bmatrix}, \text{ and } M_r := \begin{bmatrix} I_2 & 0 & 0 & 0 & 0 \\ 0 & I_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I_2 & 0 \\ 0 & 0 & \cdots & 0 & I_2 \\ \vdots & \cdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & I_2 \end{bmatrix}, \qquad (3.82)$$

where both $I_1 \in \mathbb{R}^{n \times n}$ and $I_2 \in \mathbb{R}^{m \times m}$ are identity matrices. In the terms of M_l and M_r, the uncertainty block Δ_k defined in (3.70) represents the uncertain terms in (3.81). Using the notation in (3.64) and (3.67), we can stack the expressions in (3.76) and (3.80) from $i = 1$ to $i = N_p$ and derive

$$\begin{aligned} \mathcal{X}(k) &= \mathcal{A}x(k) + \mathcal{B}\mathcal{U}(k) + M_l P \Delta_k W M_r \mathcal{U}(k), \\ \mathcal{Y}(k) &= \mathcal{C}\mathcal{X}(k), \end{aligned} \qquad (3.83)$$

where $\mathcal{X}(k)$ is the augmented, predicted state vector with

$$\mathcal{X}(k) := [\ x^T(k+1|k) \quad x^T(k+2|k) \quad \cdots \quad x^T(k+N_p|k)\ ]^T.$$

In the way of the nominal MPC, we have also formulated FH-RMPC as a QP problem.

Conclusion 3.9 *A finite horizon robust MPC system can be represented by its corresponding nominal model in parallel with a weighed unity-norm uncertainty block. Based on such a framework, robust step tracking control, or, step tracking in the presence of modelling uncertainties, can be achieved by solving a robust semi-definite optimization problem (if solutions exist) with uncertain matrix constraints:*

$$J_o = \min_{t,\, \mathcal{U}(k)} t, \qquad (3.84)$$

subject to

$$t > 0,$$

$$\begin{aligned}\max_{\Delta_k} J &\leq t \; (with \; \|\Delta_k\| = \bar{\sigma}(\Delta_k) \leq 1), \\ J &= (\mathcal{T} - \mathcal{Y}(k))^T \mathcal{Q} (\mathcal{T} - \mathcal{Y}(k)) + \mathcal{U}^T(k) \mathcal{R} \mathcal{U}(k), \\ \mathcal{X}(k) &= \mathcal{A}x(k) + \mathcal{B}\mathcal{U}(k) + M_l P \Delta_k W M_r \mathcal{U}(k), \\ \mathcal{Y}(k) &= \mathcal{C}\mathcal{X}(k), \end{aligned} \tag{3.85}$$

where $\mathcal{T}$ is the augmented reference input with

$$\mathcal{T} := [r^T \;\; r^T \;\; \cdots \;\; r^T \;]^T,$$

and t is an upper bound of the objective J.

Note that if inserting (3.83) into (3.85), the objective J can be represented by

$$\begin{aligned} J = \; & (\mathcal{T} - (\mathcal{C}\mathcal{A}x(k) + \mathcal{C}\mathcal{B}\mathcal{U}(k) + \mathcal{C}M_l P \Delta_k W M_r \mathcal{U}(k)))^T \mathcal{Q} \\ & (\mathcal{T} - (\mathcal{C}\mathcal{A}x(k) + \mathcal{C}\mathcal{B}\mathcal{U}(k)) + \mathcal{C}M_l P \Delta_k W M_r \mathcal{U}(k)) \\ & + \mathcal{U}^T(k) \mathcal{R} \mathcal{U}(k). \end{aligned} \tag{3.86}$$

3. **An FH-RMPC algorithm using LMIs**

We have formulated FH-RMPC into the robust QP problem in (3.85). Due to the presence of modelling uncertainties, (3.86) involves the uncertain terms of Δ_k. Therefore, we cannot apply Schur complements and use existing software packages to solve Problem (3.85) numerically. In order to overcome such a barrier, the robust LMI theorem, namely Lemma 3.4, is utilized. The pivotal idea of Lemma 3.4 is using an auxiliary positive scalar τ to convert robust LMIs into standard LMI constraints. Consequently, we can recast the robust QP problem in (3.84) for FH-RMPC into an SDP problem.

Theorem 3.3 *The FH-RMPC design for step-tracking control is solvable by an SDP problem:*

$$J_o = \min_{t,\, \mathcal{U}(k),\, \tau} \; t,$$

subject to

$$t > 0, \;\; \tau \geq 0,$$

and

$$\begin{bmatrix} t & (\mathcal{T}-\mathcal{CA}x(k)-\mathcal{CBU}(k))^T & \mathcal{U}^T(k) & (WM_r\mathcal{U}(k))^T \\ * & \mathcal{Q}^{-1}-\tau\mathcal{C}M_lP(\mathcal{C}M_lP)^T & 0 & 0 \\ * & * & \mathcal{R}^{-1} & 0 \\ * & * & * & \tau I \end{bmatrix} \geq 0, \qquad (3.87)$$

where $\mathcal{T}$ is augmented reference input, $\mathcal{U}(k)$ is predicted input sequence, and $\mathcal{Q}$ and $\mathcal{R}$ are weighting matrices, defined in (3.64) and (3.65). The augmented matrices $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, M_l, and M_r are constructed in (3.67) and (3.82).

Proof: Applying Schur complements and rewriting the constraints in (3.85), we have

$$\begin{bmatrix} t & (\mathcal{T}-\mathcal{CA}x(k)-\mathcal{CBU}(k))^T-(\mathcal{C}M_lP\Delta_kWM_r\mathcal{U}(k))^T & \mathcal{U}^T(k) \\ * & \mathcal{Q}^{-1} & 0 \\ * & * & \mathcal{R}^{-1} \end{bmatrix} \geq 0. \qquad (3.88)$$

Separating the certain and uncertain terms in (3.88)

$$\mathcal{T}_1 - \begin{bmatrix} 0 & (\mathcal{C}M_lP\Delta_kWM_r\mathcal{U}(k))^T & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ \mathcal{C}M_lP\Delta_kWM_r\mathcal{U}(k) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \geq 0, \qquad (3.89)$$

and rewriting (3.89), we have

$$\mathcal{T}_1 - \begin{bmatrix} (WM_r\mathcal{U}(k))^T \\ 0 \\ 0 \end{bmatrix} \Delta_k^T \begin{bmatrix} 0 & (\mathcal{C}M_lP)^T & 0 \end{bmatrix}$$

$$- \begin{bmatrix} 0 \\ \mathcal{C}M_lP \\ 0 \end{bmatrix} \Delta_k \begin{bmatrix} WM_r\mathcal{U}(k) & 0 & 0 \end{bmatrix} \geq 0, \qquad (3.90)$$

where

$$\mathcal{T}_1 = \begin{bmatrix} t & (\mathcal{T}-\mathcal{CA}x(k)-\mathcal{CBU}(k))^T & \mathcal{U}^T(k) \\ * & \mathcal{Q}^{-1} & 0 \\ * & * & \mathcal{R}^{-1} \end{bmatrix}. \qquad (3.91)$$

Setting

$$\mathcal{T}_2 = \begin{bmatrix} 0 \\ -\mathcal{C}M_lP \\ 0 \end{bmatrix}, \; \mathcal{T}_3 = \begin{bmatrix} WM_r\mathcal{U}(k) & 0 & 0 \end{bmatrix}, \text{ and } \mathcal{T}_4 = 0,$$

and recasting (3.90) into the form of (3.7), we form a robust LMI in the structure of Lemma 3.4,

$$T_1 - T_3^T \Delta_k^T T_2^T - T_2 \Delta_k T_3 \geq 0 \Leftrightarrow \begin{bmatrix} T_1 - \tau T_2 T_2^T & T_3^T \\ T_3 & \tau I \end{bmatrix} \geq 0. \tag{3.92}$$

Therefore, FH-RMPC for step tracking control is converted into an SDP problem. ■

Theorem 3.3 provides an effective approach for solving FH-RMPC problems for robust step tracking control. By adjusting the length of the prediction horizon N_p or/and the control horizon N_u, different requirements of the pre-specified performance may be satisfied. From the previous theoretical analysis, if N_p and N_u are large enough (for example $N_p = N_u = \infty$), we can always find a Lyapunov function to guarantee closed-loop stability of RMPC without any terminal constraints. However for the FH-RMPC case, if both N_p and N_u are finite, terminal cost constraints have to be imposed to guarantee the robust stability of resulting FH-RMPC systems.

3.4.3 Terminal cost constraints

In 1988, Keerthi and Gilbert first proposed a method which employed the objective function of MPC systems as a Lyapunov function to solve the nominal stability problem [43]. Later the same approach was used for nonlinear systems [62]. In this section, we will employ a similar idea and develop terminal cost constraints to guarantee robust stability of FH-RMPC systems.

Without loss of generality, here we set $N_p = N_u$, otherwise we can enforce

$$u(k+i|k) = u(k+i|k), \text{ if } N_u \leq i < N_p.$$

For ease of notation, we denote $e(k+i|k) := y(k+i|k) - r(k+i|k)$. Consider a quadratic function

$$V(x(k+i|k)) = e(k+i|k)^T \Phi e(k+i|k) = ||Cx(k+i|k) - r||_\Phi^2, \ \Phi > 0, \tag{3.93}$$

of state measurement $x(k)$, $k > 0$. Let

$$V(x(k+i+1|k)) - V(x(k+i|k)) < -(||e(k+i|k)||_Q^2 + ||u(k+i|k)||_R^2), \tag{3.94}$$

and consequently,

$$\begin{aligned} & V(x\left(k+N_p|k\right) - V\left(x\left(k+N_p-1|k\right)\right)) \\ < \;& -(||e\left(k+N_p-1|k\right)||_Q^2 + ||u\left(k+N_p-1|k\right)||_R^2). \end{aligned} \tag{3.95}$$

Summing (3.94) and (3.95) from $i = 0$ to $i = N_p$, we get

$$V\left(x\left(k+N_p|k\right)\right) - V\left(x\left(k|k\right)\right) < -J - ||e\left(k\right)||_Q^2 + ||e\left(k+N_p|k\right)||_{Q_{Np}}^2.$$

Employ $V\left(x\left(k\right)\right)$ as a Lyapunov function satisfying

$$V\left(x\left(k\right)\right) > t + ||e\left(k\right)||_Q^2 - ||e\left(k+N_p|k\right)||_{Q_{Np}}^2 + V\left(e\left(k+N_p|k\right)\right), \tag{3.96}$$

where t is the upper bound of objective J defined in (3.84). Then $\tilde{V}\left(k\right) : \mathbb{R}^n \to \mathbb{R}$, the difference of Lyapunov functions of $x\left(k+1\right)$ and $x\left(k\right)$, can be expressed as

$$\begin{aligned} \tilde{V}\left(k\right) &:= V\left(x\left(k+1\right)\right) - V\left(x\left(k\right)\right) \\ &< V\left(x\left(k+1\right)\right) - t - ||e\left(k\right)||_Q^2 \\ &\quad + ||e\left(k+N_p|k\right)||_{Q'_{Np}}^2 - V\left(x\left(k+N_p|k\right)\right). \end{aligned} \tag{3.97}$$

In order to derive closed-loop asymptotic stability, we should guarantee that the right hand side of (3.97) is negative, i.e.,

$$||e\left(k+1\right)||_\Phi^2 - t - ||e\left(k\right)||_Q^2 + ||e\left(k+N_p|k\right)||_{Q_{Np}}^2 - ||e\left(k+N_p|k\right)||_\Phi^2 < 0. \tag{3.98}$$

From (3.80), we know that if $u\left(k|k\right)$, the first element of input sequence $\mathcal{U}\left(k\right)$ is sent to the real process, the state measurement at instant $(k+1)$ can be expressed as

$$x\left(k+1\right) = Ax\left(k\right) + Bu\left(k|k\right) + \hat{\Delta}\left(k+1,k\right)u\left(k|k\right),$$

and consequently

$$e\left(k+1\right) = CAx\left(k\right) + CBu\left(k|k\right) + C\hat{\Delta}\left(k+1,k\right)u\left(k|k\right) - r. \tag{3.99}$$

Introduce two constant matrices E_1 and E_2 such that

$$\hat{\Delta}\left(k+1,k\right) = E_1\hat{\Delta}E_2 = E_1 M_l P \Delta_k W M_r E_2, \tag{3.100}$$

with

$$E_1 = \begin{bmatrix} 0 & I & 0 & \cdots & 0 \end{bmatrix}, \text{ and } E_2 = \begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix}^T.$$

Inserting (3.99) and (3.100) into (3.98), we get

$$\begin{aligned}&||CAx\left(k\right)+CBu\left(k|k\right)-r+CE_1M_lP\Delta_kWM_rE_2u\left(k|k\right)||_\Phi^2-t\\&-||Cx\left(k\right)-r||_Q^2+e^T\left(k+N_p|k\right)\left(Q_{N_p}-\Phi\right)e\left(k+N_p|k\right)<0\quad.\end{aligned}\tag{3.101}$$

So if the inequalities

$$\begin{aligned}&||Cx\left(k\right)-r||_Q^2+t-||CAx\left(k\right)+CBu\left(k|k\right)\\&-r+CE_1M_lP\Delta_kWM_rE_2u\left(k|k\right)||_\Phi^2>0,\end{aligned}\tag{3.102}$$

$$\Phi-Q_{N_p}\geq 0,\tag{3.103}$$

hold simultaneously, we can guarantee the condition in (3.101). Applying Schur complements and the property of the robust LMI theorem (Lemma 3.4), we can recast (3.102) into

$$\begin{bmatrix}||Cx\left(k\right)-r||_Q^2+t & * & *\\ CAx\left(k\right)+CBu\left(k|k\right)-r & X-\lambda_1CE_1M_lP\left(CE_1M_lP\right)^T & *\\ WM_rE_2u\left(k|k\right) & 0 & \lambda_1I\end{bmatrix}>0,\tag{3.104}$$

where $X=\Phi^{-1}$ and λ_1 is a positive scalar. Then left- and right-multiplying X to both sides of each term in (3.103) and defining a small non-negative scale κ, which is selected as a tuning scalar of $\left(Q_{N_p}+\kappa I\right)$, we have

$$X-X\left(Q_{N_p}+\kappa I\right)X\geq 0\tag{3.105}$$

It is obvious that if $\kappa\to 0$, (3.105) is equivalent to (3.103). Apply Schur complements to Eq. (3.105) and derive

$$\begin{bmatrix}X & X\\ X & \left(Q_{N_p}+\kappa I\right)^{-1}\end{bmatrix}\geq 0.\tag{3.106}$$

Combined with (3.106), (3.104) forms a sufficient condition to (3.98), which is designed for asymptotical stability of the resulting closed-loop FH-RMPC system.

Meanwhile, in order to use $V\left(x\left(k\right)\right)$ as a Lyapunov function candidate, we design another LMI to guarantee (3.96). To this end, taking advantage of the condition in (3.103), we derive a sufficient condition to (3.96)

$$||e\left(k\right)||_{Q_{N_p}}^2-t-||e\left(k\right)||_Q^2-||e\left(k+N_p|k\right)||_\Phi^2>0.\tag{3.107}$$

From (3.80), $x\left(k+N_p|k\right)$ is expressed as

$$e\left(k+N_p|k\right)=CA^{N_p}x\left(k\right)+CE_3\mathcal{B}\mathcal{U}\left(k\right)+CE_3M_lP\Delta_kWM_r\mathcal{U}\left(k\right)-r, \qquad (3.108)$$

where $E_3=\begin{bmatrix}0\cdots & 0 & 0 & I\end{bmatrix}$. Substituting (3.108) into (3.107), applying Schur complements and using the property of the robust LMI theorem, we get

$$\begin{bmatrix}\left\|Cx\left(k\right)-r\right\|_{Q_{N_p}}^2-\left\|Cx\left(k\right)-r\right\|_Q^2-t & * & * \\ CA^{N_p}x\left(k\right)+CE_3\mathcal{B}\mathcal{U}\left(k\right)-r & X-\lambda_2CE_3M_lP\left(CE_3M_lP\right)^T & * \\ WM_r\mathcal{U}\left(k\right) & 0 & \lambda_2I\end{bmatrix}>0, \qquad (3.109)$$

where λ_2 is a positive scalar.

Theorem 3.4 *To achieve step tracking performance for the FH-RMPC system defined in Fig. 3.7, the manipulated input* $u^o\left(k\right)=E_4\mathcal{U}^o\left(k|k\right)$, $k>0$, *can be obtained by minimizing the following optimization problem,*

$$J_o=\min_{\mathcal{U}(k)}\ t,$$

subject to (3.87), (3.104), (3.106), and (3.109), where X, λ_1 *and* λ_2 *are variables of LMIs for terminal cost constraints, and* E_4 *is a truncation matrix, given by*

$$E_4=\begin{bmatrix}I & 0 & \cdots & 0\end{bmatrix}.$$

The closed-loop system is guaranteed asymptotically stable if the optimal input sequences

$$\mathcal{U}^o\left(k\right)=\left[\begin{array}{cccc}u^o\left(k|k\right)^T & u^o\left(k+1|k\right)^T & \cdots & u^o\left(k+N_u-1|k\right)^T\end{array}\right]^T,\ k>0,$$

exist.

Proof. From Theorem 3.3 we know that the SDP problem in (3.84) can be solved by minimizing the linear objective in (3.87). Meanwhile, combined with constraints (3.104), (3.106), and (3.109), the quadratic function of $e\left(k\right)$

$$V\left(x\left(k\right)\right)=e^T\left(k\right)\Phi e\left(k\right),$$

can be regarded as a Lyapunov function, and it is convergent with MPC iteration. Therefore, by adding auxiliary constraints (3.104), (3.106), and (3.109) into the optimization problem defined in (3.87), we can guarantee the resulting FH-RMPC regulator to be asymptotically stable, associated with the Lyapunov function $V\left(x\left(k\right)\right)$. ■

3.5 A simulation example

Consider a classical angular positioning system proposed by Kwakernaak and Sivan in 1972 [47]. The system model is

$$\begin{aligned} \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} &= \begin{bmatrix} 1 & 0.1 \\ 0 & 1-0.1\alpha \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 0.787 \end{bmatrix} u(k), \\ y(k) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(k), \end{aligned} \tag{3.110}$$

where $\alpha \in [0.1, 10]$ reflects the uncertain coefficient of viscous friction in the physical structure. From the approaches discussed in [45], an IH-RMPC controller for the structured uncertainties in the feedback loop is first designed. Comparing with the FH-RMPC controllers proposed in this chapter, it can be seen that the FH-RMPC controllers have better tracking performance and smaller overshoots (Fig. 3.8). Here the tuning parameters are set as: $r = 1$, $Q = I$, $Q_{N_p} = I$, $R = 0.00002I$, $P = I$, $N_u = 3$, and $W = 0.1$. The simulation length equals 50. For the simulation results in Fig. 3.8, we set $\alpha = 0.7$ (nominal value $\bar{\alpha} = 0.495$).

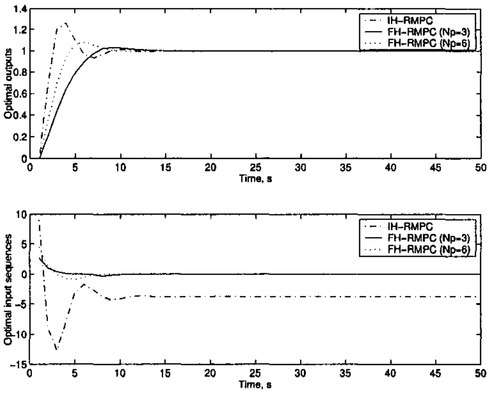


Figure 3.8: IH-RMPC controller (dash-dotted) and FH-RMPC controllers: $N_p = 3$ (solid) and $N_p = 6$ (dotted)

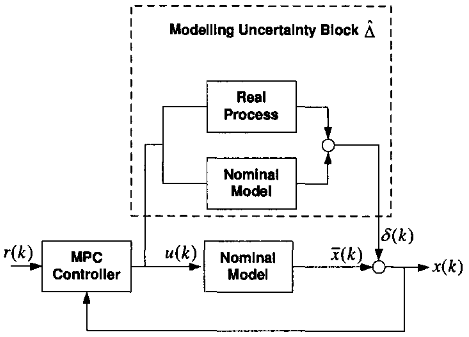


Figure 3.9: Modelling uncertainty reconfiguration

In order to reconfigure the system in (3.110) into the framework of Fig. 3.7, we can take advantages of the method described in Fig. 3.9, using the difference between the nominal model and real process to derive uncertainty block $\hat{\Delta}$.

We now increase and decrease the uncertain term α to its upper and lower bounds, i.e., setting $\alpha = 10$ and $\alpha = 0.1$, respectively. In the same fashion, we design an IH-RMPC controller again. We find that it takes a very long time to reach the steady-state value and serious ripples occur for IH-RMPC, and therefore figures are not presented here. Fig. 3.10 shows the simulation results based on FH-RMPC controllers with the different control horizons. It can be seen that FH-RMPC achieves the prespecified tracking under the worst conditions. From this point, the FH-RMPC algorithm proposed in this chapter has better robustness properties that IH-RMPC. Similar to nominal MPC controllers, FH-RMPC controllers also possess the property that if increasing the difference between N_p and N_u, the overshoot of performance decreases; meanwhile system responses become slower.

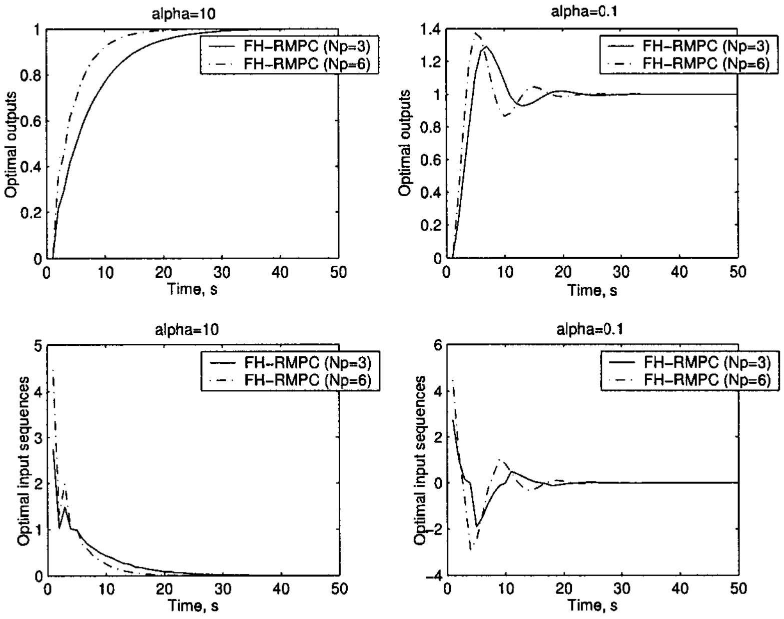


Figure 3.10: FH-RMPC controllers with the extreme uncertainty values: $N_p = 3$ (solid) and $N_p = 6$ (dash-dotted)

As discussed above, closed-loop stability is a challenge for FH-RMPC design. By imposing several extra terminal cost constraints, we can guarantee that the closed-loop stability of resulting FH-RMPC systems. Fig. 3.11 demonstrates the influence of the imposed terminal cost constraints on the system performance with the different prediction horizons N_p. Here we set $\alpha = 0.8$ and $N_u = 3$. It can be seen that the terminal cost constraints attenuate the input and output peaks, but give rise to slower responses. Fig. 3.12 demonstrates the influence of the terminal cost constraints on the system performance with different terminal weightings Q_{N_p}. We reset $\alpha = 0.9$ and $N_p = 3$, and keep $N_u = 3$. In the figures, solid lines (no cost constraints) are derived from Theorem 3.3 and dash-dotted lines from Theorem 3.4. It can be seen that for the controlled system, even though we do not impose extra terminal cost constraints, the FH-RMPC algorithm can still come to closed-loop stability.

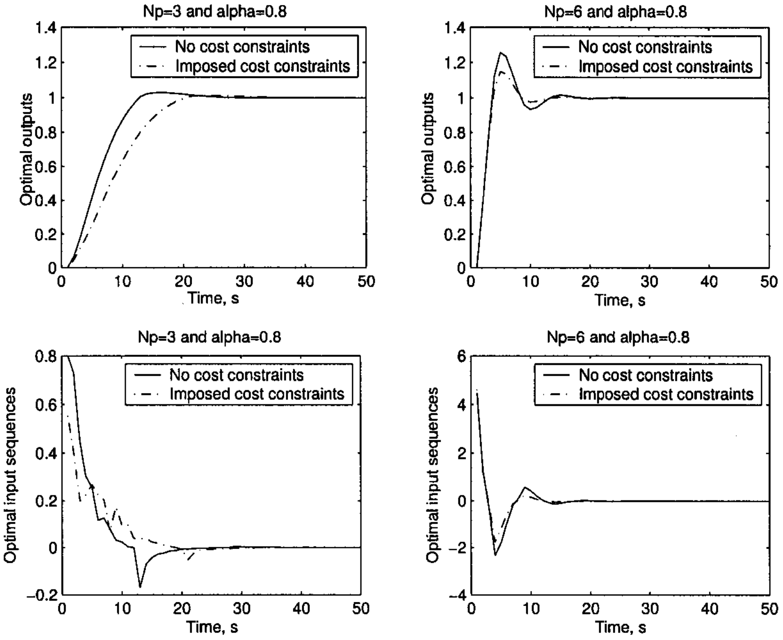


Figure 3.11: Influence of terminal cost constraints: no cost constraints (solid) and imposed cost constraints (dash-dotted)

All the simulations were performed on a PC with a Pentium 4 processor, 512MB RAM, using the software LMI Control Toolbox [32] in the MATLAB window's environment. Table 3.1 shows that the on-line computational cost can be reduced by FH-RMPC, compared with IH-RMPC. In the table, the numbers within parentheses are the average time to compute $u^o(k)$ over every prediction horizon, and the other is the total time with the simulation length equal to 50 ($N_p = N_u = 3$).

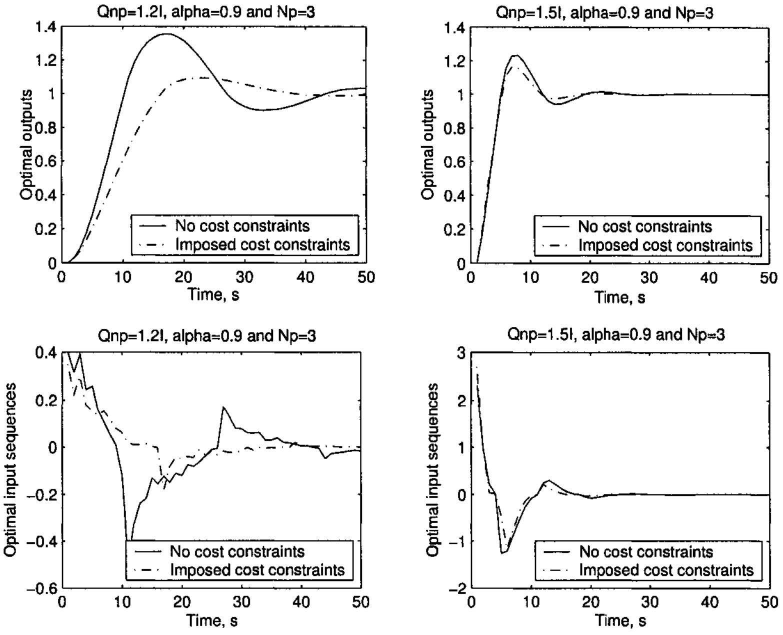


Figure 3.12: Influence of terminal weightings: no cost constraints (solid) and imposed cost constraints (dash-dotted)

Table 3.1: Time cost of the online computation

Uncertainty factor α	0.7	0.8	0.1	0.99
IH-RMPC controller (s)	4.366 (0.087)	5.049 (0.101)	–	–
FH-RMPC controller without terminal cost constraints (s)	3.956 (0.079)	3.986 (0.079)	3.695 (0.074)	3.736 (0.074)

3.6 Conclusions

In this chapter, we reviewed the background mathematics on system uncertainties and LMIs and introduced a successful IH-RMPC algorithm which was superior in numerical efficiency and closed-loop stability. After this, FH-RMPC, which is proposed in this thesis, was extensively discussed. Two important topics for FH-RMPC were covered: how to achieve robust step tracking control by FH-RMPC, and how to guarantee the closed-loop stability of resulting FH-RMPC systems. Taking advantage of the properties of a robust LMI theorem (Lemma 3.4), the conventional *min-max programming* was converted into an SDP problem. Compared with IH-RMPC, FH-RMPC has more tuning freedom, better control performance, and faster online implementation. All formulations mentioned in this chapter are based on an assumption: the controlled system has fully measurable states. How to remove this assumption is left to Chapter 6. The content of this chapter is summarized in our publication [19].

Chapter 4

Explicit model predictive control

This chapter investigates Bemporad's work — explicit model predictive control (EMPC). EMPC is featured by offline optimization and online implementation. Different from conventional MPC algorithms which provide nonlinear implicit functions as control policy, EMPC derives the expressions of manipulated inputs by a set of piece-wise affine functions associated with state space partitions.

4.1 Introduction

From previous chapters, we know that the MPC scenario is a tetralogy composed of system initialization, future signal prediction, online optimization, and controller implementation. With this scenario, MPC has to perform both optimization and implementation online and accomplish them at the same time. Considering the nature of computational complexity, though some modelling tricks may possibly simplify signal prediction, it takes a long time to finish the whole procedure of MPC formulation. This limitation hinders the MPC application to fast processes, e.g., aircraft control. To solve this problem, currently researchers are used to employing saturation elements plus anti-windup strategies [8] to regulate fast constrained systems, especially for industrial plants with integral control units.

4.1.1 Anti-windup control

The principle of anti-windup control can be demonstrated by an industrial continuously stirred tank reactor (CSTR) with a conventional proportional-integral-differential (PID) controller and an input saturation unit. In order to illustrate the influence of the undesired side effect known as "windup," we first consider the CSTR system without anti-windup compensators.

Example 4.1 *Consider an industrial CSTR system with a first order plus dead time (FOPDT) model:*

$$G(s) = \frac{e^{-0.2s}}{s+1}.$$

It is controlled by a PID compensator:

$$C(s) = 3.03(1 + \frac{1}{0.41s} + \frac{0.1s}{1+0.1s}).$$

Fig. 4.1 shows the Simulink diagram for this system. In order to satisfy input/output physical constraints, an input saturation unit is added into Fig. 4.1.

Experiments are conducted under four different conditions, namely, setting the saturation unit equal to $[-1.8, 1.8]$, $[-1.6, 1.6]$, $[-1.4, 1.4]$, *and* $[-1.2, 1.2]$, *respectively. Keep the reference input* r *equal to* 1. *The closed-loop responses of the CSTR without anti-windup compensators are shown in Fig. 4.2, which includes the trajectories of the system output* y *(upper-left part), the integrator output* y_{int} *(upper-right*

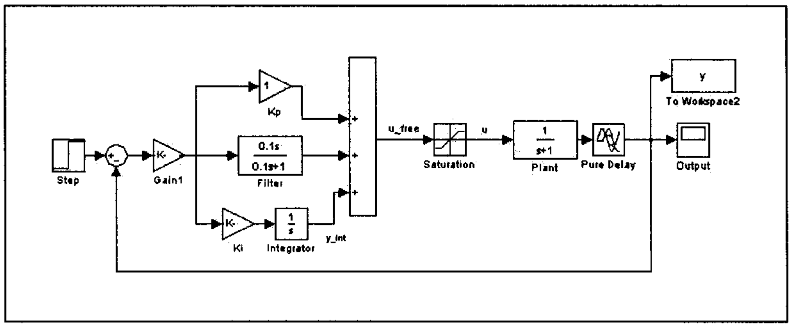


Figure 4.1: A CSTR system with an input saturation unit

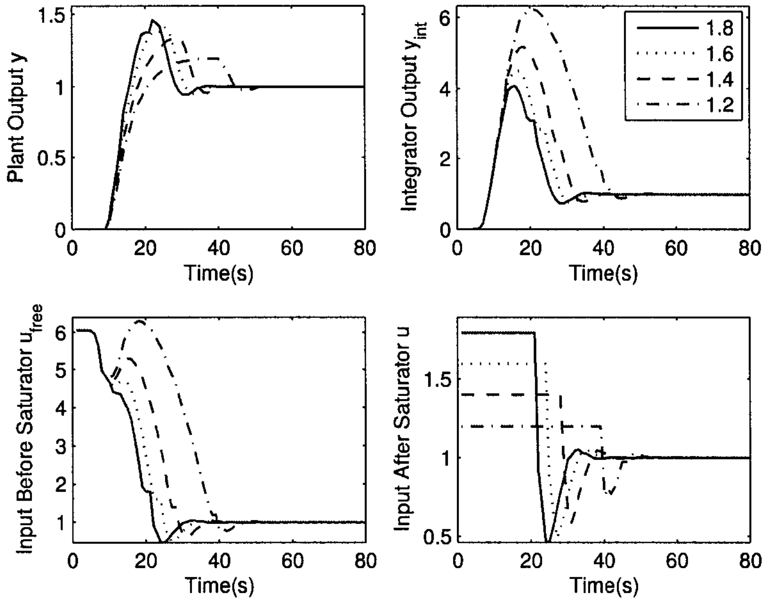


Figure 4.2: The closed-loop responses without anti-windup compensators

part), the input before the saturation block u_{free}(lower-left part), and the input after the saturation block u (lower-right part). From the curves of y_{int}, it can be seen that the undesired side effect, namely, windup, occurs, i.e., with the smaller admissible input set, the trajectory of the integrator output has a larger peak value as well as a larger peak time.

Conclusion 4.1 *[8] The effect of windup can be explained by the fact that when the* *control signal saturates the actuator, a further increase in the control signal will not lead to a faster response of the system.*

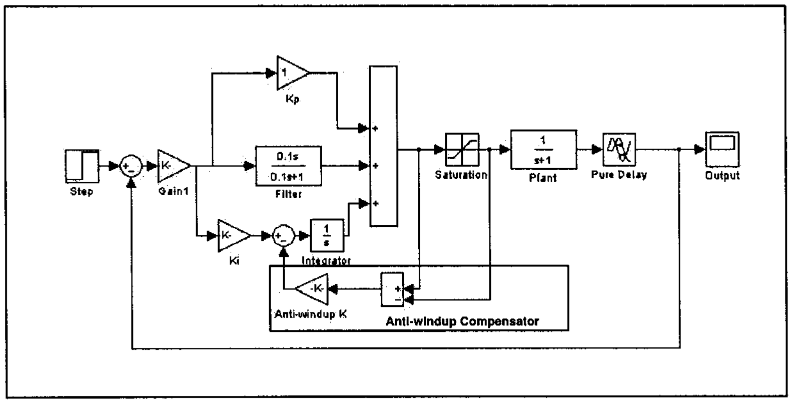


Figure 4.3: A CSTR system with an anti-windup compensator

To eliminate the windup, an anti-windup compensator is added in Fig. 4.3, and Fig. 4.4 shows the trajectories of the CSTR system with the anti-windup compensator. It can be seen that the side effect, windup, is effectively alleviated. The anti-windup compensator in the form of Fig. 4.3 is called *tracking anti-windup.* In industrial applications, there are other types of anti-windup control, e.g., conditional integration anti-windup, limited integrator anti-windup, and modified tracking anti-windup [90]. All these anti-windup controllers, however, suffer from a drawback: anti-windup gains have to be determined by trial-and-error. For example, in Fig. 4.3, the gain K is set to 5 by trial-and-error. However, if $K = 1.5$, we cannot obtain satisfactory performance. This fact inspires researchers to derive an explicit MPC algorithm and take advantage of the tuning superiority and the potential to handle system physical constraints.

4.1.2 A framework for explicit model predictive control

Fig. 4.5 is the framework for classical online MPC schemes in which the optimization block and implementation block are combined together, and the manipulated inputs are provided by the implementation block associated with online optimization. Here, the system is represented by a state space model; the system state is

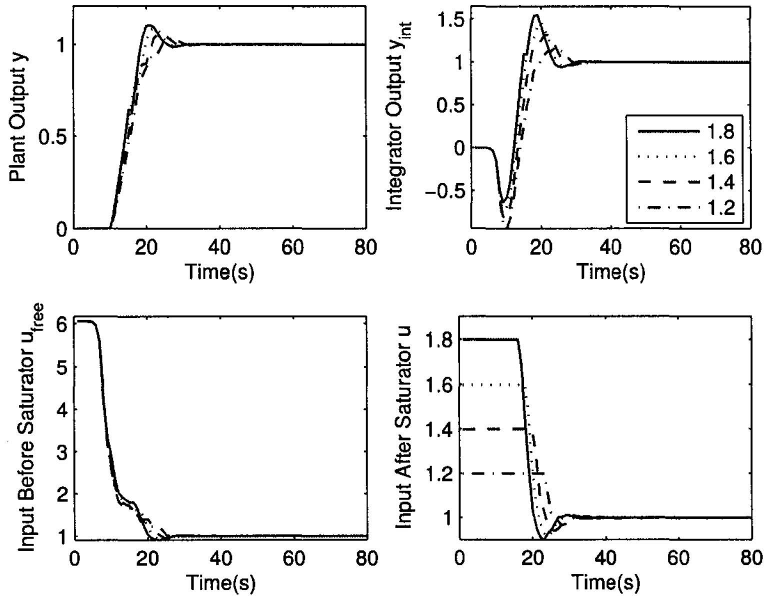
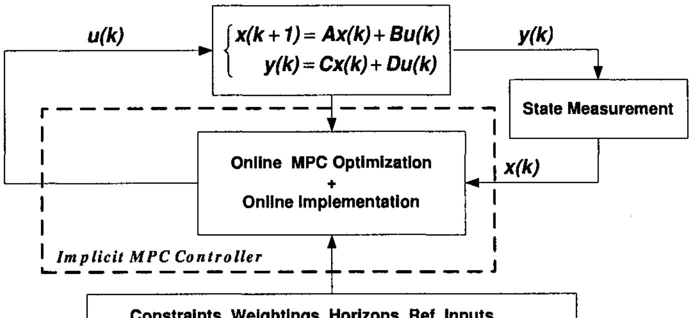


Figure 4.4: The closed-loop responses with anti-windup compensators

assumed to be fully measurable. Different from online MPC, Fig. 4.6 illustrates the



u(k)
$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases}$
y(k)
State Measurement
Online MPC Optimization
+
Online Implementation
x(k)
Implicit MPC Controller
Constraints, Weightings, Horizons, Ref. Inputs, ...

Figure 4.5: A classical framework of online MPC

framework of offline MPC, in which the optimization and implementation blocks are separated and the optimization block is independent of the state measurement $x(k)$ and the manipulated input $u(k)$. The optimization block communicates with the implementation block via two components: state space partitions and offline control functions. In 2002, Bemporad *et al.* realized the state space partition by a set of *critical polytopic regions* and the offline control policy by a series of piece-wise affine functions [6].

The offline MPC strategy can be summarized into four steps: 1) converting MPC into an optimization problem, 2) solving the optimization problem offline, 3) partitioning the state space, and 4) evaluating the optimal manipulated inputs online. State space partitions and piece-wise affine functions are pre-stored in a computer and called by implementation blocks later. The implementation block first determines the initial critical region based on the current state measurement $x(k)$ and then sends it to the control function block to evaluate the optimal input $u(k)$. After that, the optimal input $u(k)$ is re-sent back to the implementation block for online implementation. In this fashion, the MPC implementation is simplified as a function evaluation, and consequently implementation efficiency is improved dramatically. The framework in Fig. 4.6 proposes two challenges for explicit MPC: how to determine the offline functions for explicit MPC and how to perform the state partition to cover the whole state space.

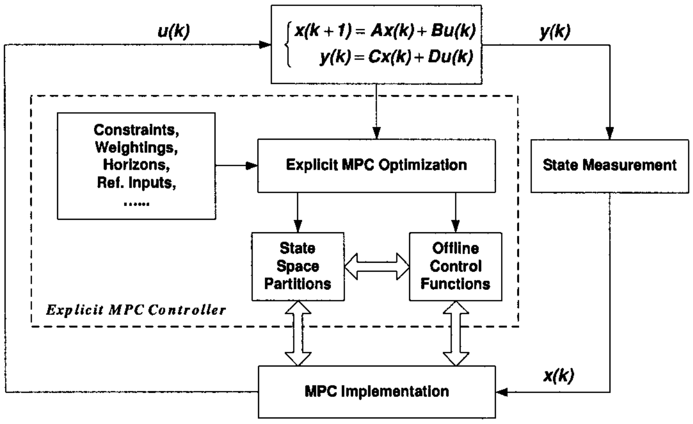

u(k)
$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases}$
y(k)
Constraints, Weightings, Horizons, Ref. Inputs, ...
Explicit MPC Optimization
State Measurement
State Space Partitions
Offline Control Functions
Explicit MPC Controller
x(k)
MPC Implementation

Figure 4.6: A framework of offline MPC

4.2 Multiple-parametric quadratic programming

Multiple-parametric programming refers to a class of optimal problems that seek the characteristics of the optimal solutions for a full range of multiple-parameter values associated with traditional objective functions. At present, two types of parametric programming have been widely studied: multiple-parametric linear programming (mp-LP) [11] and multiple-parametric quadratic programming (mp-QP) [98]. Based on these two types of multiple-parametric programming, explicit MPC laws are possibly obtained by a set of piecewise affine functions associated with state space partitions. Using several recently developed multiple-parametric programming toolboxes, e.g., the multiple-parametric toolbox (MPT) [46] and the hybrid toolbox [3], we can accomplish the analysis and visualization of explicit MPC. In this thesis, the mp-QP technique is extensively employed.

Definition 4.1 *The optimal problem in the form of*

$$\begin{aligned} &\min_z \frac{1}{2} z^T H z + \theta^T F^T z \\ s.t. \quad & G_z z \preceq G_c + G_\theta \theta, \ \theta \in \mathcal{A}_\theta, \end{aligned} \tag{4.1}$$

is defined as multiple-parametric quadratic programming (mp-QP), where $\theta \in \mathbb{R}^n$ is the dynamic parametric vector, and $z \in \mathbb{R}^m$ is the optimization variable. $H \in \mathbb{S}^m_{++}$ and all other matrices are constant with appropriate dimensions.

It can be shown that a possible solution to the problem in (4.1) is a set of piece-wise affine functions associated with a parameter partition, i.e.,

$$z = K_i \theta + g_i, \ \theta \in \mathcal{A}^i_\theta, \tag{4.2}$$

$$\mathcal{A}_\theta = \bigcup_i \mathcal{A}^i_\theta \ \ i = 1, 2, \cdots N_{CR}. \tag{4.3}$$

Eq. (4.2) defines a set of piece-wise affine functions, and (4.3) indicates a partition of the parameter admissible region $\mathcal{A}_\theta$, where i is the index of partition regions and N_{CR} is the number of partition regions. To derive the affine solutions in (4.2), we first define a Lagrange multiplier λ and convert the constrained mp-QP problem in (4.1) into an unconstrained one. In other words, a barrier function to (4.1) is given by

$$J = \frac{1}{2} z^T H z + \theta^T F^T z + \lambda^T (G_z z - G_c - G_\theta \theta + p) \tag{4.4}$$

where $p \in \mathbb{R}^{N_c}$ is a slack variable and N_c is the number of constraints in (4.1). From the first-order Karush-Kuhn-Tucker (KKT) theorem [13], the optimal conditions to the barrier function in (4.4) are given by

$$Hz + F\theta + G_z^T\lambda = 0, \tag{4.5}$$

$$(G_z z - G_c - G_\theta\theta + p)^T\lambda = 0, \tag{4.6}$$

$$\lambda \succeq 0, \quad p \succeq 0. \tag{4.7}$$

Motivated by the properties of optimization duality, the Lagrange multiplier λ can be divided into two parts, namely $\lambda_N = 0$ (the nonactive multiplier) and $\lambda_A \neq 0$ (the active multiplier). Obviously, $\lambda_N = 0 \Longleftrightarrow p_1 \neq 0$ ($\lambda_A \neq 0 \Longleftrightarrow p_2 = 0$), and $\lambda = [\lambda_N^T,\ \lambda_A^T]^T$ ($p = [p_1^T,\ p_2^T]^T$). From (4.5), we have

$$z = -H^{-1}(F\theta + \tilde{G}_z^T\lambda_A), \tag{4.8}$$

$$\tilde{G}_z z - \tilde{G}_c - \tilde{G}_\theta\theta = 0, \tag{4.9}$$

where $\{\tilde{G}_z,\ \tilde{G}_c,\ \tilde{G}_\theta\}$ is a linear combination of the active constraints in (4.6), and $\tilde{G}_z$ has a full-row rank. Insert (4.8) into (4.9) and derive

$$\lambda_A = -(\tilde{G}_z H^{-1}\tilde{G}_z^T)^{-1}(\tilde{G}_z H^{-1}F + \tilde{G}_\theta)\theta - (\tilde{G}_z H^{-1}\tilde{G}_z^T)^{-1}\tilde{G}_c. \tag{4.10}$$

Note that (4.10) is an affine function. Due to $H \in \mathbb{S}_{++}^m$ and $\tilde{G}_z$ with a full-row rank, the inverse of $(\tilde{G}_z H^{-1}\tilde{G}_z^T)$ does exist. Replacing λ_A from (4.10) and inserting it into (4.8), we have

$$\begin{aligned} z &= (-H^{-1}F + H^{-1}\tilde{G}_z^T(\tilde{G}_z H^{-1}\tilde{G}_z^T)^{-1}(\tilde{G}_z H^{-1}F + \tilde{G}_\theta))\theta \\ &\quad + H^{-1}\tilde{G}_z^T(\tilde{G}_z H^{-1}\tilde{G}_z^T)^{-1}\tilde{G}_c \\ &:= K_i\theta + g_i. \end{aligned} \tag{4.11}$$

Obviously, (4.11) shares the same structure as (4.2). Meanwhile, to guarantee the KKT conditions, we need

$$G_z z \preceq G_c + G_\theta\theta, \ \lambda_A \succ 0. \tag{4.12}$$

Eq. (4.12) defines a polytopic region, which is called as a *critical region* in this thesis. The critical region $\mathcal{A}_\theta^i$ is expressed as

$$\begin{aligned} \mathcal{A}_\theta^i := &\{\theta \in \mathbb{R}^n | (G_z K_i - G_\theta)\theta \preceq G_c - G_z g_i, \\ &(\tilde{G}_z H^{-1}\tilde{G}_z^T)^{-1}(\tilde{G}_z H^{-1}F + \tilde{G}_\theta)\theta \prec -(\tilde{G}_z H^{-1}\tilde{G}_z^T)^{-1}\tilde{G}_c\}. \end{aligned} \tag{4.13}$$

Conclusion 4.2 *The explicit solutions to the mp-QP problem in (4.1) can be expressed by a set of piece-wise affine functions associated with critical regions, i.e.,*

$$z = K_i\theta + g_i, \;\; \theta \in \mathcal{A}_\theta^i,$$

where $\mathcal{A}_\theta^i$ is the ith element of a partition of the admissible parameter region $\mathcal{A}_\theta$.

Conclusion 4.3 *Eqs. (4.11) and (4.13) propose three challenges for the parameter region's partition: 1) how to determine a combination $\{\tilde{G}_z, \; \tilde{G}_c, \; \tilde{G}_\theta\}$ based on the constraints in (4.1) , 2) how to guarantee that the combination $\{\tilde{G}_z, \; \tilde{G}_c, \; \tilde{G}_\theta\}$ have a full row-rank, and 3) how to guarantee that the partition elements $\mathcal{A}_\theta^i$s be disjointed and their union compose the whole admissible parameter region $\mathcal{A}_\theta$, i.e.,*

$$\mathcal{A}_\theta = \bigcup_i \mathcal{A}_\theta^i, \quad (i = 1, 2, \cdots N_{CR}) \tag{4.14}$$

where N_{CR} is the number of partition elements.

4.3 The partition of the parameter admissible region

This section deals with the challenges mentioned in Conclusion 4.3. Knowledge of the geometric algorithms of multi-parametric programming is employed. In 2000, Dua and Pistikopoulos proposed an effective approach for the partition of a polytopic region which is defined by a set of element-wise affine inequalities. Based on [26], we determine a satisfactory partition of the $\mathcal{A}_\theta$ in (4.14).

Theorem 4.1 *[6] Let $Y \subseteq R^n$ be a polyhedron, and*

$$CR_0 := \{x \in Y \mid Ax \preceq b\} \tag{4.15}$$

be a polyhedral subset of Y with $CR_0 \neq \emptyset$. Also let

$$R_i = \{x \in Y \;\mid A_i x \succ b_i, \; and \; A_j x \preceq b_j \; (\forall j < i)\}, \;\; (i = 1, \cdots, m) \tag{4.16}$$

where $m := \dim(b)$, and let

$$CR_{rest} := \bigcup_{i=1}^{m} R_i. \tag{4.17}$$

Then (i) $CR_{rest} \cup CR_0 = Y$; and (ii) $CR_0 \cap R_i = \emptyset$ and $R_i \cap R_j = \emptyset$ for $\forall j \neq i$, i.e., $\{CR_0, \; R_1, \cdots, \; R_m\}$ is a partition of Y.

Proof: (1) From the definition in (4.17), CR_{rest} can be expressed by

$$\begin{aligned} CR_{rest} &= \{x \in Y \mid A_i x \succ b_i,\ i = 1, \cdots, m\}, \\ &= \{x \in Y \mid Ax \succ b,\ i = 1, \cdots, m\}. \end{aligned}$$

Obviously $CR_{rest} \cup CR_0 = Y \subseteq R^n$.

(2) From the definition, R_i violates at lease one condition of the element-wise inequalities in (4.15), and therefore $CR_0 \cap R_i = \emptyset$. Also, R_i and R_j violate the different inequalities in (4.15) if $j \neq i$, so that $R_i \cap R_j = \emptyset$ for $\forall j \neq i$. Because CR_{rest} is the union of R_i disjointed with CR_0 and Y is the union of CR_{rest} and CR_0, we can say that $\{CR_0,\ R_1, \cdots,\ R_m\}$ is a partition of Y. ■

The idea behind Theorem 4.1 can be demonstrated by Example 4.2.

Example 4.2 *Consider a slab set Y defined by two element-wise inequalities as shown in Fig. 4.7-Part I, i.e.,*

$$Y := \{x \in \mathbb{R}^2 \mid -0.5 \leq x_1 \leq 1,\ -0.1 \leq x_2 \leq 1\}.$$

The initial critical region CR_0 can be defined by four element-wise inequalities, namely $C_1, \cdots, C_4$.

$$CR_0 := \left\{ x \in Y \;\middle|\; \begin{array}{r} -2x_1 + 2x_2 \leq 1 \ \ (C_1), \\ -2x_1 - 2x_2 \leq -1 \ \ (C_2), \\ 5x_1 - 8x_2 \leq -1 \ \ (C_3), \\ 0.5x_1 + x_2 \leq 1 \ \ (C_4), \end{array} \right\}. \tag{4.18}$$

CR_0 is superimposed upon Y as shown in Fig. 4.7-Part II. If C_1 is violated and $x \in Y$ is kept (no definitions on C_2, C_3 and C_4), the partition R_1 is created, as shown in Fig. 4.7-Part III. If C_1 is satisfied but C_2 is violated (no definition on C_3 and C_4), the partition R_2 is created, as shown in Fig. 4.7-Part IV. If C_1 and C_2 are satisfied but C_3 is violated (no definition on C_4), the partition R_3 is constructed, as shown in Fig. 4.7-Part V. If C_1, C_2, and C_3 is satisfied but C_4 is violated, the partition R_4 is finally produced, as shown in Fig. 4.7-Part VI. Note that $R_i \cap R_j = \emptyset$, $CR_0 \cap R_i = \emptyset$, and the partition $\{CR_0,\ R_1, \cdots,\ R_4\}$ composes the whole polyhedron Y.

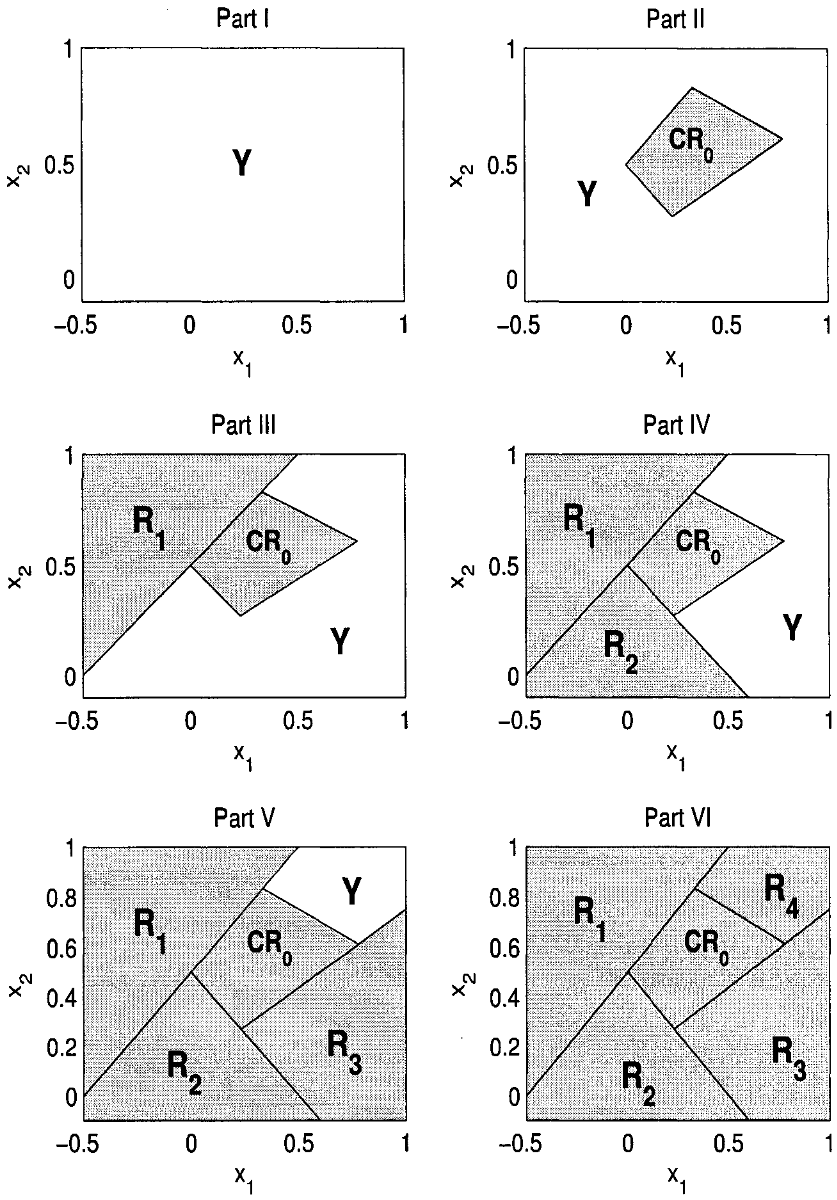


Figure 4.7: A partition of the polyhedron region Y

Fig. 4.7 illustrates the method discussed in Theorem 4.7. The partition elements are disjointed, and the union of these elements composes the given region Y. Eq. (4.13) defines a critical region corresponding to the optimal solution in (4.11). By setting $\mathcal{A}_\theta^i$ as the initial critical region CR_0, deriving the possible partition $\{CR_0, R_1, \cdots, R_{N_{CR}}\}$, and searching for the optimal solutions in the rest of the regions $\{R_1, \cdots, R_{N_{CR}}\}$, we can derive the explicit solutions to the mp-QP problem in (4.1).

4.4 Offline model predictive control

From the discussion on mp-QP and parameter space partition, we can develop an offline model predictive control law. Here we assume that the controlled system is given by a state space model, and neither internal nor external uncertainties are included in formulations, i.e, this section focuses on nominal explicit model predictive control.

4.4.1 Problem definition

Consider a discrete-time linear system

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ y(k) &= Cx(k), \end{aligned} \tag{4.19}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, and $y \in \mathbb{R}^q$ are the state, input, and output, respectively. A, B and C are constant matrices with compatible dimensions. The input and output constraints are defined by a set of element-wise inequalities,

$$\begin{aligned} \mathcal{A}_u &:= \{u \in \mathbb{R}^m |\ u_{\min} \preceq u \preceq u_{\max},\ u_{\min},\ u_{\max} \in \mathbb{R}^m\}, \\ \mathcal{A}_y &:= \{y \in \mathbb{R}^q |\ y_{\min} \preceq y \preceq y_{\max},\ y_{\min},\ y_{\max} \in \mathbb{R}^q\}, \end{aligned} \tag{4.20}$$

where $\mathcal{A}_u$ and $\mathcal{A}_y$ are the admissible input and output sets. $u_{\min}$ ($u_{\max}$) and $y_{\min}$ ($y_{\max}$) are constant vectors composed by input and output lower (upper) bounds.

Definition 4.2 *The design of an explicit MPC regulator for the system in (4.19)* *is a QP problem,*

$$\begin{aligned}
&\min_{U} \quad J && (4.21)\\
s.t. \quad & J = ||x(k+N_p)||_P^2 + \sum_{i=0}^{N_p-1} ||x(k+i)||_Q^2 + ||u(k+i)||_R^2, \\
& u(k+i) \in \mathcal{A}_u, \; y(k+i) \in \mathcal{A}_y, \\
& x(k+i+1) = Ax(k+i) + Bu(k+i), \\
& y(k+i) = Cx(k+i), \\
& u(k+i) = Fx(k+i) \; \textit{if} \; N_u < i \le N_p - 1, \\
& 0 < N_u \le N_p - 1, \; Q \in \mathbb{S}_{++}^n, \quad R \in \mathbb{S}_{++}^m.
\end{aligned}$$

where $P \in \mathbb{S}_{++}^n$ and F are terminal weighting and terminal feedback gain. The pair (A, B) is stabilizable and $(Q^{1/2}, A)$ is detectable. Assume an initial condition equal to x_o. The control objective is to drive the state trajectory converging to the origin.

Because we are developing an offline MPC law, the problem in (4.21) can be defined over any horizon window. For ease of notation, $x(k+i)$ denotes $x(k+i|k)$, similarly for $u(k+i)$ and $y(k+i)$.

4.4.2 Closed-loop stability

To guarantee closed-loop stability, the objective function (4.21) is chosen as a Lyapunov candidacy function. So the Lyapunov function at instant k can be expressed by

$$V(k) = ||x(k+N_p)||_P^2 + \sum_{i=0}^{N_p-1} ||x(k+i)||_Q^2 + ||u(k+i)||_R^2. \qquad (4.22)$$

From (4.22), the difference of the Lyapunov functions between $V(k+1)$ and $V(k)$ is given by

$$\begin{aligned}
\tilde{V} &:= V(k+1) - V(k) \\
&= ||x(k+N_p+1)||_P^2 + ||x(k+N_p)||_Q^2 + ||u(k+N_p)||_R^2 \\
&\quad -||x(k+N_p)||_P^2 - (||x(k)||_Q^2 + ||u(k+N_p)||_R^2).
\end{aligned}$$

Therefore if

$$||x(k+N_p+1)||_P^2 + ||x(k+N_p)||_Q^2 + ||u(k+N_p)||_R^2 - ||x(k+N_p)||_P^2 = 0, \quad (4.23)$$

the resulting closed-loop explicit MPC system is asymptotically stable since the initial penalized terms $(||x(k)||_Q^2 + ||u(k+N_p)||_R^2)$ are positive. Inserting (4.19) into (4.23) and replacing $u(k+N_p)$ by the terminal feedback gain F, i.e., setting $u(k+N_p) = Fx(k+N_p)$,

$$(A+BF)^T P(A+BF) + Q + F^T RF - P = 0. \tag{4.24}$$

is derived. Eq. (4.24) is an algebraic Riccati equation (ARE), and if the pair (A, B) is stabilizable and ($Q^{1/2}$, A) detectable, (4.24) is feasible given any terminal weighting $P \in \mathbb{S}_{++}^n$ and terminal feedback gain F. Incorporating the solutions to the ARE in (4.24) with the explicit MPC formulation in (4.21), closed-loop stability of explicit MPC is obtained.

4.4.3 An mp-QP problem for explicit MPC

The optimization problem in (4.21) can be possibly converted into an mp-QP problem, and then based on the solutions in (4.11) and (4.13), explicit MPC is obtained by a set of piece-wise affine functions associated with state space partitions. The objective in (4.21) may be rewritten in the form of stacked matrices, i.e.,

$$J = x^T(k)Qx(k) + X^T \mathcal{Q} X + U^T \mathcal{R} U, \tag{4.25}$$

where

$$\begin{aligned} \mathcal{Q} &:= \text{diag}\,(Q, \cdots Q, P), \qquad (4.26)\\ \mathcal{R} &:= \text{diag}\,(R, \cdots, R,\ (N_p - N_u)R),\\ X &:= [x^T(k+1),\ \cdots,\ x^T(k+N_p)]^T\\ U &:= [u^T(k),\ \cdots,\ u(k+N_u)]^T. \end{aligned}$$

Performing state and output prediction based on the model in (4.19) gives

$$\begin{aligned} X &= \mathcal{A}x(k) + \mathcal{B}U, \qquad (4.27)\\ Y &= \mathcal{C}X, \qquad (4.28) \end{aligned}$$

where $\mathcal{C} := \text{diag}\,(C, \cdots, C)$ and

$$\mathcal{A} := \begin{bmatrix} A \\ \vdots \\ A^{N_u} \\ \vdots \\ A^{N_p} \end{bmatrix}, \ \mathcal{B} := \begin{bmatrix} B & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ A^{N_u-1}B & A^{N_u-2}B & \cdots & B \\ \vdots & \vdots & \vdots & \vdots \\ A^{N_p-1}B & A^{N_p-2}B & \cdots & (A^{N_p-N_u}B + \cdots + B) \end{bmatrix}. \tag{4.29}$$

Inserting (4.27) into (4.25), we have

$$\begin{aligned} J &= U^T(\mathcal{B}^T \mathcal{Q} \mathcal{B} + \mathcal{R})U + 2x^T(k)\mathcal{A}^T \mathcal{Q} \mathcal{B} U + x^T(k)(Q + \mathcal{A}^T \mathcal{Q} \mathcal{A})x(k) \\ &= \frac{1}{2} U^T H U + x^T(k) F^T U + \Xi, \end{aligned} \tag{4.30}$$

where $H := 2(\mathcal{B}^T \mathcal{Q} \mathcal{B} + \mathcal{R}) \in \mathbb{S}_{++}^{m \cdot N_u}$ and $F := 2\mathcal{B}^T \mathcal{Q} \mathcal{A}$. Ξ is a square term of $x(k)$ and independent of the optimization variable U. From the stacked matrices in (4.26) and (4.29), explicit MPC can be reformulated as an mp-QP problem.

Theorem 4.2 *The explicit MPC regulator for system (4.19) constrained by the element-wise inequalities in (4.20) is an mp-QP problem.*

Proof: The proof follows immediately from the objective in (4.30). The optimization problem in (4.21) is equivalent to minimizing $\hat{J}$,

$$\hat{J} = \frac{1}{2} U^T H U + x^T(k) F^T U, \tag{4.31}$$

where $x(k)$ is the vector parameter and U is the stacked optimization variable. Note that Ξ is independent of the optimization variable U. From the expressions of predicted states and outputs in (4.27) and (4.28), the element-wise inequalities in (4.20) can be rewritten as

$$\underline{U} \preceq U \preceq \bar{U}, \ \underline{Y} \preceq \mathcal{C}\mathcal{A}x(k) + \mathcal{C}\mathcal{B}U \preceq \bar{Y}, \tag{4.32}$$

where $\underline{U}$ $(\bar{U})$ and $\underline{Y}$ $(\bar{Y})$ are the stacked input's and output's lower-bound (upper-bound) vectors, namely $\underline{U} := [u_{\min}^T, \cdots, u_{\min}^T]$ and $\underline{U} \in \mathbb{R}^{m \cdot N_u}$. Augmenting the constraint conditions in (4.32), we have

$$G_U U \preceq G_c + G_x x(k), \tag{4.33}$$

where

$$G_U := \begin{bmatrix} I \\ -I \\ \mathcal{C}\mathcal{B} \\ -\mathcal{C}\mathcal{B} \end{bmatrix}, \; G_c := \begin{bmatrix} \bar{U} \\ -\underline{U} \\ \bar{Y} \\ -\underline{Y} \end{bmatrix}, \text{ and } G_x := \begin{bmatrix} 0 \\ 0 \\ -\mathcal{C}\mathcal{A} \\ \mathcal{C}\mathcal{A} \end{bmatrix}.$$

Combining (4.31) with (4.33), the design of explicit MPC is converted into an mp-QP problem,

$$\min_U \frac{1}{2}U^T HU + x^T(k)F^T U \tag{4.34}$$

$$s.t. \quad G_U U \preceq G_c + G_x x(k), \; x(k) \in \mathcal{A}_x, \tag{4.35}$$

where $\mathcal{A}_x$ is an admissible state set which is normally derived from the physical conditions of a system. In the case that there is no definition on $\mathcal{A}_x$, we can define a closed polyhedron acting as $\mathcal{A}_x$ with

$$\mathcal{A}_x := \{x \in R^n \mid E_l x \preceq E_r\}. \tag{4.36}$$

The problem in (4.34) is an mp-QP problem with the multiple-parameter $x(k)$. ■

From Sections 4.2 and 4.3, it can be seen that the solution to (4.34) is a set of piece-wise affine functions associated with state space partitions.

Theorem 4.3 *Let a linear combination of the active constraints $\{\tilde{G}_U, \tilde{G}_c, \tilde{G}_x\}$ out of $\{G_U, G_c, G_x\}$ ($\tilde{G}_U$ has a full-row rank), and the initial critical region $CR_0 \subseteq \mathcal{A}_x$ be determined by $\{\tilde{G}_U, \tilde{G}_c, \tilde{G}\}$. The optimal control law U for the problem in (4.34) is defined by a set of functions of $x(k)$ associated with $\mathcal{A}_x$'s partitions.*

Proof: The proof follows directly from (4.11) and (4.13). The solutions to (4.34) can be expressed by

$$\begin{aligned} U_i &= (-H^{-1}F + H^{-1}\tilde{G}_U^T(\tilde{G}_U H^{-1}\tilde{G}_U^T)^{-1}(\tilde{G}_z H^{-1}F + \tilde{G}_x))x(k) \\ &\quad +H^{-1}\tilde{G}_U^T(\tilde{G}_U H^{-1}\tilde{G}_U^T)^{-1}\tilde{G}_c \\ &:= K_i x(k) + g_i, \end{aligned} \tag{4.37}$$

and the associated state space partition is

$$\begin{aligned} CR_0 := \; & \{x(k) \in \mathcal{A}_x \mid (G_U K_i - G_x)x(k) \preceq G_c - g_i, \\ & (\tilde{G}_U H^{-1}\tilde{G}_U^T)^{-1}(\tilde{G}_U H^{-1}F + \tilde{G}_x)x(k) \prec -(\tilde{G}_U H^{-1}\tilde{G}_U^T)^{-1}\tilde{G}_c\}. \end{aligned} \tag{4.38}$$

The active constraints $\{\tilde{G}_U, \tilde{G}_c, \tilde{G}_x\}$ can be determined by solving a linear programming (LP) problem which will be discussed in Section 4.4.4. Implement Theorem 4.1 and derive the $\mathcal{A}_x$'s partition $\{CR_0, R_1, \cdots, R_{N_c}\}$. N_c is the number of independent inequalities in (4.35). By searching for the optimal input functions U_i within the rest regions $\{R_1, \cdots, R_{N_c}\}$, we can finally derive a set of affine functions U_i and the associated critical regions CR_i. ■

Notice that Theorem 4.3 assumes that a combination of the active constraints $\{\tilde{G}_U, \tilde{G}_c, \tilde{G}_x\}$ exists, and the matrix $\tilde{G}_U$ has a full-row rank. The proof of Theorem 4.3 states that the combination $\{\tilde{G}_U, \tilde{G}_c, \tilde{G}_x\}$ can be determined by solving an LP problem. Here, a new problem is proposed: *how to define an LP problem and set up the initial critical region CR_0?*

4.4.4 A combination of the active constraints

To start searching U_i, we need a combination of the active constraints in (4.35), $\{\tilde{G}_U, \tilde{G}_c, \tilde{G}_x\}$, and the matrix $\tilde{G}_U$ must have a full-row rank. To this end, an initial searching parameter $x_0(k) \in \mathcal{A}_x$ is constructed. $x_0(k)$ is chosen by a point as close to the center of the polyhedron $\mathcal{A}_x$ as possible in the sense of 2-norms and satisfies the constraints in (4.35) as well. To determine $x_0(k)$, an LP problem is constructed

$$\max_{x_0(k),\ U,\ \delta} \quad \delta \tag{4.39}$$

$$\begin{aligned} s.t. \quad & E_l^i x + \delta \|E_l^i\|_2 \preceq E_r^i, \qquad (4.40) \\ & G_U U \preceq G_c + G_x x_0(k), \end{aligned}$$

where E_l^i denotes the ith row of the matrix E_l in (4.36), the same as for E_r^i. From Constraint (4.40), it can be seen that δ is the distance from the point $x_0(k)$ to each bound of the polyhedron $\mathcal{A}_x$. Therefore, $x_0(k)$ is the Chebychev center and δ is the associated Chebychev radius.

After determining the value of $x_0(k)$, we send it back to the problem in (4.35) and solve this mp-QP problem. Set the solution to (4.35) U^o. Then a combination $\{\tilde{G}_U, \tilde{G}_c, \tilde{G}_x\}$ can be determined by $x_0(k)$ and U^o, i.e.,

$$\tilde{G}_U U \preceq \tilde{G}_c + \tilde{G}_x x_0(k). \tag{4.41}$$

Because H is positive and symmetric, the combination $\{\tilde{G}_U, \tilde{G}_c, \tilde{G}_x\}$ is uniquely determined.

Remark 4.1 *If the optimal solution to (4.39) $\delta^o \leq 0$, this LP problem is infeasible. To handle these cases, widen the polyhedron $\mathcal{A}_x$ to a larger region.*

Remark 4.2 *There is a degenerated case of the condition in (4.41): the optimal pair $(x_0(k), U^o)$ does not activate any constraints in (4.35). In this case, the optimal solutions to (4.34) are simplified dramatically. No active constraints mean that the Lagrange multiplier λ_A in (4.10) uniquely equals to zeros ($\lambda \succ 0$), so that the optimization variable z in (4.8) is simplified as*

$$z = -H^{-1}F\theta.$$

In the same fashion, (4.37) becomes

$$\begin{aligned} U_i &= -H^{-1}Fx(k) \\ &:= K_i x(k), \end{aligned} \tag{4.42}$$

and the associated partition is

$$CR_0 := \{x(k) \in \mathcal{A}_x \mid (G_U K_i - G_x)x(k) \preceq G_c - g_i\}. \tag{4.43}$$

To store and visualize the optimal solutions to an mp-QP problems, MATLAB Hybrid Toolbox is employed in this thesis.

4.5 A simulation example

Hybrid Toolbox is a numerical solver for multiple-parametric programming. It was developed by Bemporad in 2005. The current version is 1.0.12 - Feb 23, 2006, which is available on the website: http://www.dii.unisi.it/~bemporad/. This package is developed under the MATLAB environment.

Consider the nominal version of Example 3.1,

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 0.9719 & -0.0013 \\ -0.034 & 0.8628 \end{bmatrix} x(k) + \begin{bmatrix} -0.0839 & 0.0232 \\ 0.0761 & 0.4144 \end{bmatrix} u(k), \\ y(k) &= x(k), \end{aligned} \tag{4.44}$$

where $x \in \mathbb{R}^2$, $u \in \mathbb{R}^2$, and $y \in \mathbb{R}^2$ are the state, input, and output respectively. The input and output constraints are given by

$$\mathcal{A}_u \;:=\; \{u \in \mathbb{R}^2 \mid -0.5 \cdot \mathbf{1} \preceq u \preceq 0.5 \cdot \mathbf{1}\}, \tag{4.45}$$

$$\mathcal{A}_y \;:=\; \{y \in \mathbb{R}^2 \mid -\infty \cdot \mathbf{1} \preceq y \preceq \infty \cdot \mathbf{1}\}. \tag{4.46}$$

Eq. (4.46) indicates that no output constraints are imposed on the system in (4.44). The control objective is to develop an explicit MPC controller to drive the state from the initial point $x_0 = [2,\ 1]^T$ to the origin along the state trajectories. The following parameters are used in the explicit MPC design

$$\begin{aligned} P &= \begin{bmatrix} 3.7897 & -0.0581 \\ -0.0581 & 1.2928 \end{bmatrix},\ K = \begin{bmatrix} 2.4543 & -0.2984 \\ -0.4042 & -1.3949 \end{bmatrix}, \\ Q &= I,\ R = 0.1I,\ x_0 = [2, 1]^T,\ u_0 = 0. \end{aligned}$$

With $N_p = 3$ and $N_u = 2$, the MPC law is

$$u = \begin{cases} \begin{bmatrix} 2.9960 & -0.6694 \\ -0.3662 & -1.0335 \end{bmatrix} x & \text{if} \quad \begin{bmatrix} 5.9920 & -1.3389 \\ -0.7324 & -2.0669 \\ -5.9920 & 1.3389 \\ 0.7324 & 2.0669 \end{bmatrix} x \preceq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \\ & \qquad\qquad \text{(Region \#1)} \\ \begin{bmatrix} -0.5 \\ -0.5 \end{bmatrix} & \text{if} \quad \begin{bmatrix} 2.9255 & -1.6709 \\ 0.2595 & -1.9595 \end{bmatrix} x \preceq \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \\ & \qquad\qquad \text{(Region \#2)} \\ \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} & \text{if} \quad \begin{bmatrix} -2.9255 & 1.6709 \\ -0.2595 & 1.9595 \end{bmatrix} x \preceq \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \\ & \qquad\qquad \text{(Region \#3)} \\ \begin{bmatrix} -0.5 \\ 0.5 \end{bmatrix} & \text{if} \quad \begin{bmatrix} -0.3681 & 2.7795 \\ 33.2688 & -19.0014 \end{bmatrix} x \preceq \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \\ & \qquad\qquad \text{(Region \#4)} \\ \begin{bmatrix} 0.5 \\ -0.5 \end{bmatrix} & \text{if} \quad \begin{bmatrix} -33.2688 & 19.0014 \\ 0.3681 & -2.7795 \end{bmatrix} x \preceq \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \\ & \qquad\qquad \text{(Region \#5)} \\ \text{(...Continued on the next page)} & \end{cases}$$

$$u = \begin{cases} \begin{bmatrix} 0 & 0 \\ 0.1522 & -1.1493 \end{bmatrix} x + \begin{bmatrix} -0.5 \\ 0.0865 \end{bmatrix} & \text{if} \quad \begin{bmatrix} 0.3681 & -2.7795 \\ -0.2595 & 1.9595 \\ 5.9920 & -1.3389 \end{bmatrix} x \preceq \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}, \\ & \qquad (\text{ Region } \#6) \\ \begin{bmatrix} 0 & 0 \\ 0.1522 & -1.1493 \end{bmatrix} x + \begin{bmatrix} 0.5 \\ -0.0865 \end{bmatrix} & \text{if} \quad \begin{bmatrix} 0.2595 & -1.9595 \\ -0.3681 & 2.7795 \\ -5.9920 & 1.3389 \end{bmatrix} x \preceq \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}, \\ & \qquad (\text{ Region } \#7) \\ \begin{bmatrix} 2.6890 & -1.5358 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0.4192 \\ -0.5 \end{bmatrix} & \text{if} \quad \begin{bmatrix} 33.2688 & -19.0014 \\ -2.9255 & 1.6709 \\ -0.7324 & -2.0669 \end{bmatrix} x \preceq \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}, \\ & \qquad (\text{ Region } \#8) \\ \begin{bmatrix} 2.6890 & -1.5358 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} -0.4192 \\ 0.5 \end{bmatrix} & \text{if} \quad \begin{bmatrix} 2.9255 & -1.6709 \\ -33.2688 & 19.0014 \\ 0.7324 & 2.0669 \end{bmatrix} x \preceq \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}. \\ & \qquad (\text{ Region } \#9) \end{cases} \tag{4.47}$$

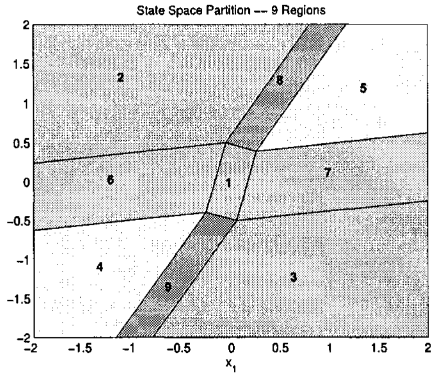


Figure 4.8: The state space partition with $N_p = 3$ and $N_u = 2$

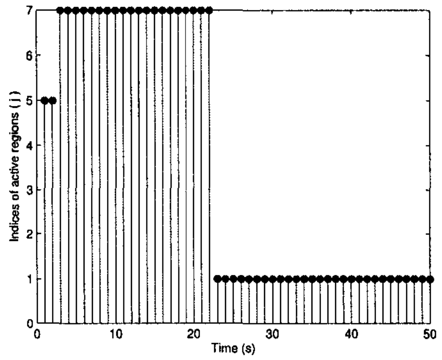


Figure 4.9: The transition of active regions along state trajectories $N_p = 3$ and $N_u = 2$

From 4.8, it can be seen that the state space $\mathcal{A}_x$ is divided into 9 regions corresponding to the control laws in (4.47). To execute the space partitions in (4.39) and (4.40), the polyhedron $\mathcal{A}_x$ is defined by

$$\mathcal{A}_x := \{x \in \mathbb{R}^2 \mid -2 \cdot \mathbf{1} \preceq x \preceq 2 \cdot \mathbf{1}\}. \tag{4.48}$$

Fig. 4.9 indicates the active regions along the state trajectories. It can be seen that the states start from Region #5, transit Region #7, enter into Region #1, and finally converge to the origin. If both the prediction horizon and the control horizon increase to 4, the state space $\mathcal{A}_x$ is separated into 53 regions, which are shown in Fig. 4.10. Fig. 4.11 indicates the active regions along the state trajectories with $N_u = N_p = 4$. In this case, the states attain more active region transitions.

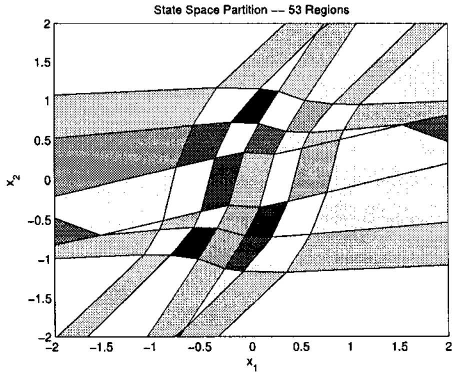


Figure 4.10: The state space partition with $N_p = 4$ and $N_u = 4$

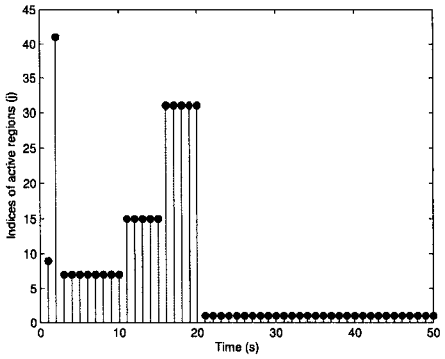


Figure 4.11: The transition of active regions along state trajectories with $N_p = 4$ and $N_u = 4$.

To compare the control performance of explicit MPC with different horizon length, we put the trajectories of states and inputs with the different N_p and N_u in the same windows (see Fig. 4.12). Roughly speaking, the explicit MPC regulator with $N_p = 3$ and $N_u = 2$ derives a more aggressive control than one with both N_p and N_u equal to 4. Also, we plot out the phase planes with the different horizon settings in Fig. 4.13. To demonstrate the implementation efficiency, we compare the time-cost of both offline (explicit) MPC and online MPC in Table 4.1. It can be seen that although offline MPC takes more time on optimization, it improves the implementation efficiency dramatically. Actually, offline MPC spends most time on state space partition and visualization. Online MPC needs less time for optimization than offline MPC, but its implementation takes much more time than offline MPC because the optimization and implementation are combined in online MPC. In a word, for fast processes, offline MPC is more practical than online MPC.

4.6 Conclusions

This chapter converts the offline MPC design into an mp-QP problem, and consequently the control law can be possibly represented by a set of affine functions

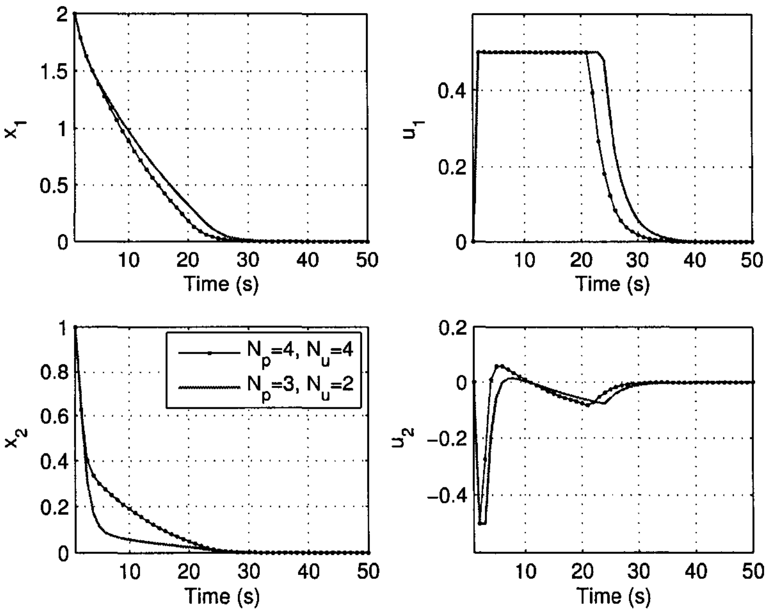


Figure 4.12: The trajectories of states and inputs with different horizons

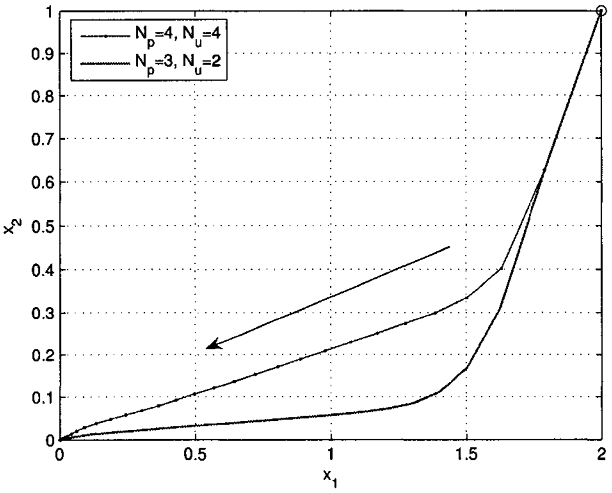


Figure 4.13: The phase plane of states and inputs with different horizons

associated with state space partition. All the results in this chapter can be easily extended to 1-norm and ∞-norm objective functions [4]. Multiple-parametric linear

Table 4.1: Time-cost of online MPC and offline MPC

MPC Types	Online MPC		Offline MPC	
Horizons	Opt.	Imp.	Opt.	Imp.
$N_p = 4$, $N_u = 4$	0.760000	0.760000	2.105000	0.110000
$N_p = 3$, $N_u = 2$	0.491000	0.491000	0.681000	0.110000

programming is constructed and can be solved in the same manner. For the explicit MPC algorithms covered in this chapter, two assumptions are critical: the system model must be precise and no internal/external disturbances are considered in MPC formulations. How to remove this limitation is left for Chapter 5, which deals with explicit robust model predictive control, the core of this thesis.

Chapter 5

Explicit robust model predictive control

This chapter develops an explicit robust MPC (ERMPC) algorithm for constrained MIMO systems with internal and external uncertainties. By proposing a novel prediction pattern, namely recursive closed-loop prediction, ERMPC is converted into multiple-parametric sub-quadratic programming (mp-SQP) and consequently, only one-step state prediction is necessary for ERMPC formulation with arbitrary horizons. It is shown that the optimal solution to mp-SQP problems is the piece-wise affine functions associated with corresponding piece objectives and state critical regions. Asymptotic closed-loop stability of resulting ERMPC systems is guaranteed by a terminal weighting and a terminal feedback gain; and by introducing two tuning variables, the algorithm is capable of adjusting the tradeoff between system performance and robustness. The state admissible set, as a nontrivial problem, is also constructed by two approaches: piece-wise linear norms and polyhedral Voronoi sets.

5.1 Problem definition

A system with structured internal and external uncertainties and bounded output disturbances is given by

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + f(\Delta(k),\ x(k),\ k), \\ y(k) &= Cx(k) + C_d d(k), \end{aligned} \tag{5.1}$$

where $x(k) \in \mathbb{R}^n$ stands for the state, $u(k) \in \mathbb{R}^m$ for the input, and $y(k) \in \mathbb{R}^q$ for the output. A, B, C, and C_d are all constant matrices with appropriate dimensions. The pair of $(A,\ B)$ is stabilizable. $d(k) \in \mathbb{R}^l$ is a combination of input and output disturbances, satisfying

$$d^T(k) W_d d(k) \leq 1,\ W_d \in \mathbb{S}^l_+. \tag{5.2}$$

$f(\cdot)$ is a time-varying nonlinear function with uncertain terms in the form of

$$||f(\Delta(k),\ x(k),\ k)||_2 \leq \mu \|x(k)\|_2, \tag{5.3}$$

which represents system internal uncertainties $\Delta(k)$ and external disturbances $d(k)$. The structure of (5.3) is widely used in perturbed systems, where μ gauges the bound of system uncertainties [109]. A more specified structure of $f(\cdot)$ is

$$|f_i(\Delta(k),\ x(k),\ k)| \leq \alpha_i \left|w_i^T x\right|,\ i = 1, 2, \cdots, n, \tag{5.4}$$

where w_i is a linear weighting vector in $\mathbb{R}^n$ and $\alpha_i > 0$ is used to scale the uncertain effect on each channels. Through routine algebraic manipulation, one can show that (5.4) corresponds to (5.3) with

$$\|f(\cdot)\|_2 \leq (\text{Trace } (\Gamma W_x W_x^T \Gamma))^{1/2} \|x\|_2,$$

where

$$\Gamma = \text{diag}(\alpha_i, \cdots,\ \alpha_n) \text{ and } W_x = [w_1^T,\ \cdots, w_n^T]^T \text{ (state weighting)}.$$

Moreover, other widely used structured uncertainties can be also converted into (5.4), such as structured uncertainties in the feedback loop (used in Chapter 3) [45]. Given the system

$$x(k+1) = (A + T_w \Delta(k) C) x(k), \tag{5.5}$$

where T_w is non-singular, and $\Delta(k)$ is the internal time-varying uncertainties in the feedback loop with the structure of

$$\Delta(k) = \mathrm{diag}(\Delta_1(k), \cdots, \Delta_n(k)), \text{ and } \bar{\sigma}(\Delta_i(k)) \leq \alpha_i.$$

Setting $x = T_w z$ and performing the similarity transformation to (5.5), we obtain

$$z(k+1) = T_w^{-1} A T_w z(k) + f(\Delta(k), z(k)),$$

where $f(\Delta(k), z(k)) := \Delta(k) C T_w z(k)$. Obviously,

$$|f_i(\Delta(k), z(k))| \leq \alpha_i |(CT_w)_i z|, \ i = 1, 2, \cdots, n.$$

In this chapter, we will assume that the system uncertainties and disturbances obey the constraints of (5.2) and (5.3). Compared with (4.19), the model in (5.1) describes both internal and external uncertainties as well as system dynamics.

The robust regulation problem to system (5.1) is first considered, i.e., driving the initial state $x(0)$ to converge to the origin in the presence of uncertain terms of $f(\cdot)$ and $d(k)$. Tracking problems, sometimes referred to as offset-free control, are discussed in Section 5.3.2. Here we introduce system input and output constraints based on practical operations of the system.

Definition 5.1 *The admissible input set $\mathcal{A}_u$ and output set $\mathcal{A}_y$ of system (5.1) are polyhedral regions defined by generalized element-wise inequalities,*

$$\begin{aligned} \mathcal{A}_u &:= \{u \in \mathbb{R}^m | \ u_{\min} \preceq u \preceq u_{\max}, \ u_{\min}, \ u_{\max} \in \mathbb{R}^m\}, \\ \mathcal{A}_y &:= \{y \in \mathbb{R}^q | \ y_{\min} \preceq y \preceq y_{\max}, \ y_{\min}, \ y_{\max} \in \mathbb{R}^q\}, \end{aligned} \tag{5.6}$$

where $u_{\min}$ ($y_{\min}$) and $u_{\max}$ ($y_{\max}$) are constant vectors, composed of corresponding channel's upper- and low-bounds. If there are no definitions over the jth input channel constraints, set $u_{\min,\ j} = -\infty$ and $u_{\max,\ j} = +\infty$. Similar rules are also imposed on output constraints.

Given (5.6) and associated by the uncertainty definitions in (5.2) and (5.3), the admissible state set can be derived from two approaches: the piece-wise linear norm of output disturbances and a Chebychev polyhedron with perturbed bounds. Obviously, how to get the admissible state set is a nontrivial problem.

5.1.1 Admissible state sets

From (5.1) and (5.6), we have

$$y_{\min} - C_d d(k) \preceq Cx(k) \preceq y_{\max} - C_d d(k). \tag{5.7}$$

After determining the bounds of $|C_{d,\ j} d(k)|$ $(j = 1, \cdots, q)$, i.e., the piecewise linear norm of $C_d d(k)$, we can use a polyhedral region as the admissible state set.

Definition 5.2 *A piecewise linear norm $\|\cdot\|_{pl}$ of a vector $z \in \mathbb{R}^n$ is defined by*

$$||z||_{pl} = \max_{i=1,\cdots,q} |a_i^T z|, \tag{5.8}$$

where $a_i \in \mathbb{R}^n$ is a column linear weighting.

Although the value of a piecewise linear norm is not easy to calculate, it can be approximated by the quadratic norm as $||z||_\Psi$ $(\Psi \in \mathbb{S}^n_{++})$. The book [12] provides a practical approach to compute the piecewise linear norm of lower-dimensional signals.

Lemma 5.1 *[12] For any $P > 0$, there exists some constant $\alpha \geq 1$ such that the quadratic norm defined by $||z||_\Psi := \sqrt{z^T \Psi z} = ||\Psi^{1/2} z||$ satisfies*

$$1/\sqrt{\alpha}||z||_\Psi \leq ||z||_{pl} \leq \sqrt{\alpha}||z||_\Psi \text{ (for } \forall z). \tag{5.9}$$

To approximate $||z||_{pl}$ by $||z||_\Psi$, the optimal α^o and Ψ^o can be calculated by eigenvalue programming (EVP),

$$\begin{aligned} &\min \quad \alpha \\ &s.t. \quad \zeta_j^T \Psi \zeta_j \leq \alpha, \ (j = 1, 2, \cdots, L) \\ &\qquad \begin{bmatrix} \Psi & a_i \\ a_i^T & \alpha \end{bmatrix} \geq 0, \ (i = 1, 2, \cdots, q), \end{aligned}$$

where $\zeta_1, \cdots, \zeta_L$ are the vertices of the unit ball $\mathcal{B}_{pl}$ of $||z||_{pl}$, and $\mathcal{B}_{pl} := \{z \mid ||z||_{pl} \leq 1\} = Co\{\zeta_1, \cdots, \zeta_L\}$.

From the point of view of computational complexity, it is obvious that the number of vertices L can grow exponentially in q and n, so that Lemma 5.1 is not practical for signals with high-order dimensions. Generally speaking, if l, the order of $d(k)$, is a small scalar, we can easily obtain $||C_d d(k)||_{pl}$, namely the bound of $|C_{d,\ j} d(k)|$ $(j = 1, \cdots, q)$ from (5.9).

Theorem 5.1 *The admissible state set $\mathcal{A}_x$ for the perturbed system of (5.1) with structured uncertainties (5.2) and (5.3) is defined by generalized element-wise inequalities,*

$$y_{\min} + \mathbf{1}\gamma \preceq Cx(k) \preceq y_{\max} - \mathbf{1}\gamma, \tag{5.10}$$

where $\gamma := \sqrt{\alpha}\bar{\sigma}\left(\Psi^{1/2}W^{-1/2}\right)$ and $\mathbf{1}$ denotes the constant vector whose all components equal to one. α and Ψ are constant parameters of the approximation to $\|C_d d(k)\|_{pl}$, derived from Lemma 5.1. $y_{\min}$ and $y_{\max}$ are output physical limitations.

Proof: From Lemma 5.1, we can get the values of Ψ and α satisfying $|C_{d,\ j}d(k)| \leq \sqrt{\alpha}||\Psi^{1/2}d(k)||$ for $\forall\ j = 1, \cdots, q$. From the bound of the weighted 2-norm of the disturbances in (5.2), it can be seen

$$||\Psi^{1/2}d(k)||_2 = ||\Psi^{1/2}W^{-1/2}W^{1/2}d(k)||_2 \leq \bar{\sigma}(\Psi^{1/2}W^{-1/2}).$$

Therefore, the theorem is proven. ■

Considering the limitation of the piece-wise linear norm approximation, Theorem 5.1 cannot solve the admissible state set with high-dimensional disturbances. To remove such a limitation, *Chebychev polyhedra* and *Voronoi sets* are introduced.

The element-wise inequalities of (5.7) determine a polyhedral set $\mathcal{H}(d(k))$. Due to the terms of $d(k)$, $\mathcal{H}(\cdot)$ is not just a single polyhedron, instead it stands for a family of polyhedra with the perturbed bounds. To guarantee all states satisfying the physical requirements in the presence of disturbances, we try to figure out the intersection set of all possible elements of $\mathcal{H}(\cdot)$, denoted by $\mathcal{A}_x$. Concluding from convex optimization, we know that $\mathcal{A}_x$ is a *Voronoi set*, and the corresponding radius is *Chebychev radius* [13].

Definition 5.3 *Let $x_1, \cdots, x_k \in \mathbb{R}^n$. Consider the set of points that are closer (in Euclidean norm) to x_0 than all x_i in the measurement of Euclidean norm, say,*

$$V := \{x \in \mathbb{R}^n \mid ||x - x_o||_2 \leq ||x - x_i||_2,\ i = 1, \cdots, k\}.$$

V is called a Voronoi set round x_o with respect to $x_1, \cdots, x_k$, and x_o is the Chebychev center.

Theorem 5.2 *The admissible state set $\mathcal{A}_x$ for the perturbed system of (5.1) with structured uncertainties (5.2) and (5.3), is defined by a Voronoi set*

$$\mathcal{A}_x := \{x(k) \in \mathbb{R}^n \mid ||x(k) - x^o||_2 \leq ||x(k) - x_i||_2\}, \tag{5.11}$$

where x^o is the Chebychev center of polyhedron $\mathcal{A}_x$, and x_i $(i = 1, \cdots, 2q)$ are mirror images of x^o with respect to the corresponding bounds, given by

$$x_i = x^o + \frac{2\delta^o}{||C_i||_2}(C_i)^T \ (\ i = 1, \cdots, q), \tag{5.12}$$

$$x_i = x^o - \frac{2\delta^o}{||C_i||_2}(C_i)^T \ (\ i = q+1, \cdots, 2q), \tag{5.13}$$

where δ^o is the Chebychev radius.

Proof: Expand (5.11), we have $E_l x \preceq E_r$ where

$$E_l = 2\begin{bmatrix} (x_1 - x^o)^T \\ \vdots \\ (x_{2q} - x^o)^T \end{bmatrix} \text{ and } E_r = \begin{bmatrix} (x_1^T x_1 - x^{oT} x^o) \\ \vdots \\ (x_{2q}^T x_{2q} - x^{oT} x^o)^T \end{bmatrix}. \tag{5.14}$$

So the condition in (5.11) defines a polyhedral set which is constant with our previous discussion — $\mathcal{A}_x$ is the intersection of all possible $\mathcal{H}(\cdot)$. The center of the intersection x^o is solvable by a sub-optimization problem, i.e., minimizing the region of $\mathcal{A}_x$ with respect to disturbance $d(k)$, and then maximizing the *Chebychev ball* contained in $\mathcal{A}_x$. The implicitness of this operation can be illustrated by Fig. 5.1. After determining the coordinates of the center x^o and corresponding mirror points x_i with respect to bounds, we are able to write down the expression of $\mathcal{A}_x$. From (5.7) and the geometric formulation of the distance between an internal point x and the boundary hyperplanes, the sub-optimization problem can be defined as

$$\max_x \min_d \ \delta \tag{5.15}$$

$$s.t. \quad C_j x + \delta \, ||C_j||_2 \leq y_{\max,\, j} - C_{d,\, j} d(k), \tag{5.16}$$

$$-C_j x + \delta \, ||C_j||_2 \leq -y_{\min,\, j} + C_{d,\, j} d(k), \tag{5.17}$$

$$||d(k)||_{W_d}^2 \leq 1, \tag{5.18}$$

$$j = 1, \cdots, q \text{ and } \delta > 0,$$

where δ is the distance from $x \in \mathcal{A}_x$ to the boundary hyperplanes. To get an approximation to the solutions of (5.15), we can tighten the right hand sides of (5.16)

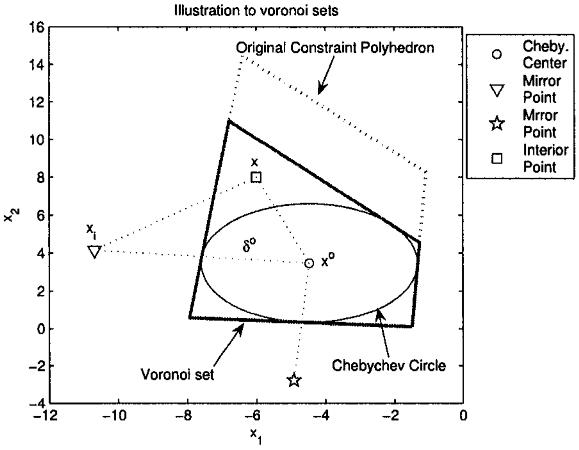


Figure 5.1: Illustration of Voronoi sets

and (5.17) from the Cauchy-Schwarz inequality and condition (5.18). Consequently, the operation converts problem (5.15) into a linear program. Obviously, optimal solution x^o is a Chebychev center, and δ^o is a Chebychev radius. Given the expression of all x_i from radius δ^o, $x_1, \cdots, x_{2q}$ associated with x^o defines a *Voronoi set* which in fact, is the Voronoi set for the intersection of $\mathcal{H}(d(k))$. ■

In the following, we use a simple 2-dimension system to explain the effects of $d(k)$ on $\mathcal{A}_x$ geometrically, meanwhile show the computation of $\mathcal{A}_x$ based on the piecewise linear norm of $d(k)$ and a *Voronoi set.*

Example 5.1 *Set the system output matrix is*

$$C = \begin{bmatrix} 0.5 & 2.5 \\ 0.6 & 0.7 \end{bmatrix},$$

and the disturbance output matrix $C_d = 0.3I$*. Admissible outputs are bounded by a slab set* $[-\mathbf{1},\ \mathbf{1}]$*, i.e.,* $-\mathbf{1} \preceq Cx + C_d d \preceq \mathbf{1}$*. With the perturbations of* $d(k)$*,* $\mathcal{H}(d(k))$ *keeps fluctuating. Performing the above operation, we can create the admissible polyhedral state set by a Voronoi set with the Chebychev center* x^o *and radius* δ^o *(see Fig. 5.2). If using the method given by Theorem 5.1, we can also get another* $\mathcal{A}_x$ *shown in Fig. 5.3.*

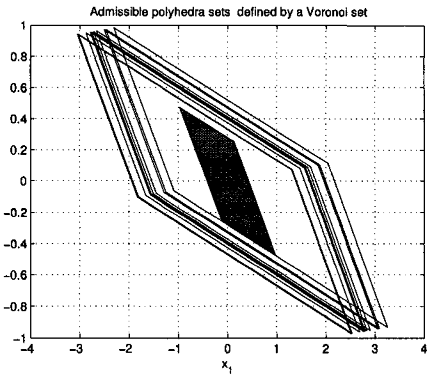


Figure 5.2: Admissible state set defined by a Voronoi set.

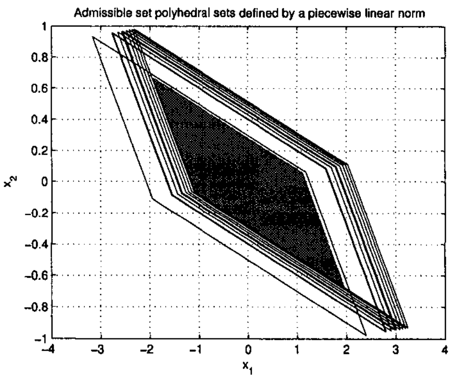


Figure 5.3: Admissible state polyhedral set derived by the piecewise linear norm of $d(k)$.

Given the definitions of state space model (5.1) and admissible input and state sets, we can define the constrained finite horizon robust MPC as follows:

Definition 5.4 *The design of a robust MPC regulator for the perturbed system in (5.1) with structured uncertainties (5.2), (5.3), and (5.4) is a constrained sub-optimization problem,*

$$\min_{U} \max_{f(\cdot)} J_{k \to k+N_p} \tag{5.19}$$

$$\begin{aligned} J_{k \to k+N_p} &= \|x(k+N_p)\|_P^2 + \sum_{i=0}^{N_p-1} \|x(k+i)\|_Q^2 + \|u(k+i)\|_R^2, \qquad (5.20) \\ & \quad s.t. \;\; u(k+i) \in \mathcal{A}_u, \quad x(k+i) \in \mathcal{A}_x, \\ & \quad x(k+i+1) = Ax(k+i) + Bu(k+i) + f(x(k+i), \Delta(k+i), k+i), \\ & \quad u(k+i) = Fx(k+i) \;\; if \;\; N_u < i \le N_p, \\ & \quad 0 < N_u \le N_p - 1, \end{aligned}$$

where $Q \in \mathbb{S}_+^n$ and $R \in \mathbb{S}_+^m$ are weightings, and $P \in \mathbb{S}_+^n$ and F are the terminal weighing and the terminal feedback gain, respectively. Pair (A, B) is stabilizable and $(Q^{1/2}, A)$ detectable.

Parameters P and F are constructed to guaranteed closed-loop stability and discussed in the next subsection.

Remark 5.1 *Compared with Definition 4.2, Definition 5.4 incorporates both internal and external uncertainties with explicit MPC formulation, and the objective is degenerated to a sub-optimization problem from optimization one.*

5.1.2 Closed-loop robust stability

To combine the stability issue with the MPC formulation together, we define the objective function $J_{k \to k+N_p}$ as a Lyapunov candidacy function,

$$V(k) = ||x(k+N_p)||_P^2 + \sum_{i=0}^{N_p-1} ||x(k+i)||_Q^2 + ||u(k+i)||_R^2,$$

and then the difference of the Lyapunov functions $V(k)$ and $V(k+1)$ can be expressed as

$$\begin{aligned} \tilde{V} \;:=\; & ||x((k+N_p+1)||_P^2 - ||x(k+N_p)||_P^2 \\ & + (||x(k+N_p)||_Q^2 + ||u(k+N_p)||_R^2) - (||x(k)||_Q^2 + ||u(k)||_R^2). \end{aligned}$$

So if

$$||x(k+N_p+1)||_P^2 - ||x(k+N_p)||_P^2 + ||x(k+N_p)||_Q^2 + ||u(k+N_p)||_R^2 \leq 0, \quad (5.21)$$

the closed-loop robust MPC system is asymptotically stable. Inserting (5.1) into (5.21) and replacing $u(k+N_p)$ by $Fx\,(k+N_p)$, we have

$$\check{x}^T((A+BF)^T P (A+BF) + Q + F^T RF - P)\check{x} + 2\check{x}^T (A+BF)^T P\check{f} + \check{f}^T P \check{f} \leq 0,$$

where $\check{x} := x\,(k+N_p)$ and $\check{f}$ denote terminal uncertainties. So the above inequality is necessary to the conditions

$$\begin{aligned} (A+BF)^T P (A+BF) + Q + F^T RF - P + \nu\check{Q} \;&=\; 0, & (5.22) \\ 2\check{x}^T (A+BF)^T P\check{f} + \check{f}^T P\check{f} - \check{x}^T \nu\check{Q}\check{x} \;&\leq\; 0, & (5.23) \end{aligned}$$

where $\check{Q} \in \mathbb{S}_+^n$ is introduced to assist system stability analysis, meanwhile $\nu > 0$ can be regarded as a scaling parameter. Obviously, (5.22) is an algebraic Riccati equation which guarantees the feasibility of terminal weighting P and terminal feedback F, given the arbitrary tuning parameters $\check{Q}$ and ν. The feasibility of condition (5.23) plays a critical role on system asymptotical stability.

Theorem 5.3 *The perturbed system (5.1) with uncertainties bounded by (5.2) and (5.3) is asymptotically stable if*

$$\begin{bmatrix} \nu\check{Q} - I & -(A+BF)^T P \\ -P(A+BF) & \mu^{-2}I - P \end{bmatrix} \geq 0, \tag{5.24}$$

where P and F is the solutions to the algebraic Riccati equation (5.22). $\nu > 0$ and $\check{Q} \in \mathbb{S}^n_+$ are stability tuning parameters.

Proof: From (5.3) and the *S-procedure* of LMIs, it can be seen that inequality (5.23) holds only if

$$2\check{x}^T (A+BF)^T P\check{f} + \check{f}^T P\check{f} - \check{x}^T \nu\check{Q}\check{x} + \check{x}^T\check{x} - \mu^{-2}\check{f}^T\check{f} \leq 0,$$

equivalently,

$$\begin{bmatrix} \check{x} \\ \check{f} \end{bmatrix}^T \begin{bmatrix} \nu\check{Q} - I & -(A+BF)^T P \\ -P(A+BF) & \mu^{-2}I - P \end{bmatrix} \begin{bmatrix} \check{x} \\ \check{f} \end{bmatrix} \geq 0,$$

which is necessary to (5.24). Theorem 5.3 is proven. ■

Theorem 5.3 offers a criterion to test whether system (5.1) with uncertainties (5.2) and (5.3) is asymptotically stable when the bound of disturbances μ is given. Conversely, given the variables P, F, ν, and $\check{Q}$ which can guarantee closed-loop stability, we can derive the upper bound of μ from condition (5.23), namely the bound of robustness.

Corollary 5.1 *The upper bound of the robustness parameter μ in system (5.1) with structured uncertainties (5.2) and (5.3) can be determined by*

$$\begin{aligned} &\max_{\bar{\mu}} \ \bar{\mu} \quad (\bar{\mu} > 0), \\ s.t. \quad &\begin{bmatrix} \nu\check{Q} - I & -(A+BF)^T P \\ * & \bar{\mu}^{-2}I - P \end{bmatrix} > 0, \end{aligned} \tag{5.25}$$

where P and F is the solutions to the algebraic Riccati equation in (5.22), and $\nu > 0$ and $\check{Q} \in \mathbb{S}^l_+$ are stability tuning parameters.

Remark 5.2 *By an auxiliary scalar $\tau > 0$, problem (5.25) can be easily converted into a semi-definite optimization problem, i.e., setting $\tau := \bar{\mu}^{-2}$ and changing the objective to minimize τ. Then the upper bound of robustness $\bar{\mu}^o$ equals to $1/\sqrt{\tau}$.*

Apparently, the program of finding a robustness bound is an SDP problem, and the optimal solution $\bar{\mu}^o$ is tightly relative to the selection of tuning variables $\check{Q}$ and ν. Therefore selecting the different values of $\check{Q}$ and ν, we can adjust the conservability of the RMPC regulator, and achieve a satisfactory tradeoff between system performance and robustness.

5.2 Robust MPC in the recursive closed-loop prediction

RMPC in the recursive closed-loop prediction was initially proposed by Lee and Yu in 1997 [50]. Its crucial difference from traditional RMPC problems lies in the prediction pattern. The algorithm first optimizes manipulated input $u\left(k+i+1\right)$ by a sub-optimization problem, and then $u\left(k+i\right)$. The same sub-optimization is iterated N_p times and the length of prediction horizon N_p is exactly determined by the number of iteration steps, instead of the number of state predictions. There is no need to perform multiple-step predictions, and no high-order uncertain terms to appear in robust MPC formulation, which is one of the notorious barriers of robust MPC design. Because of the nature of the recursive closed-loop prediction, the optimal value of piece objective $J_{k+i+1\to N_p}$ will be a part of the expression of objective $J_{k+i\to N_p}$. This property emphasizes the effect of future predicted feedback gains on the current predicted feedback regulation, and thus giving rise to a more flexible MPC regulator.

5.2.1 Mathematical formulation

The total objective $J_{k\to k+N_p}$ is divided into N_p pieces, and the first piece to be optimized is $J_{k+N_p-1\to k+N_p}$. From the stability analysis above, we have

$$\begin{aligned} J^o_{k+N_p-1\to k+N_p} &= \max_{f(\cdot)} \; ||x(k+N_p)||^2_P + \; ||x(k+N_p-1)||^2_Q \\ &\quad +||u(k+N_p-1)||^2_R, && (5.26) \\ \text{s.t.} \quad & x(k+N_p) = (A+BF)x(k+N_p-1) + f\left(\cdot\right), && (5.27) \\ & u(k+N_p-1) = Fx(k+N_p-1). && (5.28) \end{aligned}$$

Note that the terminal manipulated input $u^o(k+N_p-1) = Fx(k+N_p-1)$ is regulated by solving an algebraic Riccati equation, instead of MPC formulation.

Replacing $x(k+N_p)$ by $x(k+N_p-1)$ from model (5.27), we have

$$J^o_{k+N_p-1\to k+N_p} = h_1(x(k+N_p-1)) \text{ and } u^o(k+N_p-1) = g_1(x(k+N_p-1)), \quad (5.29)$$

where the subscript "1" of h and g denotes the horizon length. Set $J^o_{k+N_p-1\to k+N_p}$ as a term of piece objective $J_{k+N_p-2\to k+N_p}$, and then optimize manipulated input $u\,(k+N_p-2)$ in the recursive closed-loop prediction pattern,

$$\begin{aligned} J^o_{k+N_p-2\to k+N_p} \quad = \quad & \min_{u(k+N_p-2)} \; \max_{f(\cdot)} \; (||x(k+N_p-2)||^2_Q + \\ & ||u(k+N_p-2)||^2_R + h_1(x(k+N_p-1))), && (5.30) \\ \text{s.t.} \quad & x(k+N_p-1) = Ax(k+N_p-2) \\ & +Bu(k+N_p-2) + f(\cdot), && (5.31) \\ & x(k+N_p-1) \in \mathcal{A}_x, \; u(k+N_p-2) \in \mathcal{A}_u. \end{aligned}$$

We first assume that a pair of analytic (explicit) solutions to (5.30) is available, and let

$$J^o_{k+N_p-2\to k+N_p} = h_2(x(k+N_p-2)) \text{ and } u^o(k+N_p-2) = g_2(x(k+N_p-2)). \quad (5.32)$$

Comparing (5.30) and (5.26), if $h_1\,(\cdot)$ is a quadratic function of $x\,(k+N_p-1)$, they share exactly the same structure, except that the predicted state is pushed one-step backwards. This property will play a pivotal role later on. Iterate the operation of (5.30) to each piece objective recursively until we reach the optimization of total objective $J_{k\to k+N_p}$,

$$J_{k\to k+N_p} = \min_{u(k)} \; \max_{f(\cdot)} \; (||x\,(k)\,||^2_Q + ||u\,(k)\,||^2_R + h_{N_p-1}(x(k+1))), \quad (5.33)$$

subject to state space model (5.1) and the admissible state and input sets. Replace $x\,(k+1)$ by $x\,(k)$, the current manipulated input $u^o\,(k)$ is finally created by

$$J^o_{k\to k+N_p} = h_{N_p}(x\,(k)) \text{ and } u^o\,(k) = g_{N_p}(x\,(k)), \quad (5.34)$$

After sending $u^o\,(k) = g_{N_p}(x\,(k))$ to the real process, we finish the design of RMPC in the recursive closed-loop prediction. Generally speaking, the above closed-loop

RMPC schemes can be summarized as a recursive optimization problem,

$$
\begin{aligned}
J^o_{k\to k+N_p} \quad = \quad & \min_{u(k)} \max_{f(\cdot)} \left(\left|\left|x\left(k\right)\right|\right|^2_Q + \left|\left|u\left(k\right)\right|\right|^2_R + \cdots + \right. \\
& \min_{\tilde{u}(k+N_p-2)} \max_{f(\cdot)} (||x(k+N_p-2)||^2_P + ||u(k+N_p-2)||^2_Q + \\
& \max_{f(\cdot)} ||x(k+N_p)||^2_P + ||x(k+N_p-1)||^2_Q \\
& +||Fx(k+N_p-1)||^2_R)))). \qquad (5.35)
\end{aligned}
$$

Remark 5.3 *Robust MPC in the recursive closed-loop prediction avoids multiple-step state/output predictions; this reduces the computational complexity dramatically. But how to get the analytic (explicit) solutions to (5.32) and (5.34) is critical. Determining the explicit solutions to constrained suboptimal quadratic programming is a nontrivial issue.*

Remark 5.4 *The optimal solutions* $\{g_1(x\left(k+N_p-1\right)),\ \cdots,\ g_{N_p}(x\left(k\right))\}$ *should be piece-wise linear or piece-wise affine functions of predicted states, and piece objectives*

$$\{J^o_{k+N_p-1\to k+N_p},\ \cdots,\ J^o_{k\to k+N_p}\}$$

should be quadratic functions of predicted states. Otherwise we can not get the uniform structure of piece objectives. Apparently, these are two big challenges for explicit robust MPC.

5.2.2 Explicit solutions to the piece objective of $J_{k+i\to k+N_p}$

Definition 5.5 *The optimal problem in the form of*

$$
\begin{aligned}
& \min_{z} \max_{\Delta(k)} \frac{1}{2} z^T H_z z + \theta\left(k\right)^T H_{z\theta} z + \frac{1}{2} \theta\left(k\right)^T H_\theta \theta\left(k\right), \\
s.t. \quad & G_z z \preceq G_c + G_\theta \theta\left(k\right), \\
& \theta\left(k+1\right) = \Theta(\theta\left(k\right),\ \Delta\left(k\right)),
\end{aligned}
$$

is defined as multiple-parametric sub-quadratic programming (mp-SQP), where $\theta\left(k\right) \in \mathbb{R}^n$ *is the dynamic parameter vector,* $z \in \mathbb{R}^m$ *is the optimal variable and* $\Delta\left(k\right)$ *is the predefined structured uncertainty.* $H_z \in \mathbb{S}^m_{++}$, $H_\theta \in \mathbb{S}^n_{+}$, *and all other matrices are constant with appropriate dimensions.* $\Theta\left(\cdot\right)$ *is a piece-wise linear or affine function.*

Obviously, one-step robust MPC in the recursive prediction is an mp-SQP problem. We first assume that $h_{N_p-i-1}(\cdot)$ is a quadratic function of $x(k+i+1)$,

$$h_{N_p-i-1}(\cdot) = ||x(k+i+1)||^2_{P_{Np-i-1}} + Z_{N_p-i-1}x(k+i+1) + O_{N_p-i-1}, \tag{5.36}$$

where $P_{N_p-i-1} \in \mathbb{S}^n_+$. Z_{N_p-i-1} and O_{N_p-i-1} are constant matrices. Substitute $x(k+i+1)$ by $x(k+i)$, insert the result into (5.36), and then derive

$$\begin{aligned} h_{N_p-i-1}(\cdot) &= ||A\breve{x} + B\breve{u} + f(\cdot)||^2_{P_{Np-i-1}} + Z_{N_p-i-1}(A\breve{x} + B\breve{u} + f(\cdot)) + O_{N_p-i-1} \\ &\leq ||A\breve{x} + B\breve{u}||^2_{\hat{P}_{Np-i-1}} + Z_{N_p-i-1}(A\breve{x} + B\breve{u}) + \hat{O}_{N_p-i-1}, \end{aligned} \tag{5.37}$$

where

$$\begin{aligned} \hat{O}_{N_p-i-1} &:= \frac{1}{4}Z_{N_p-i-1}Z^T_{N_p-i-1} + O_{N_p-i-1} + (\mu^2 + 2\mu^2\bar{\sigma}^2(P_{N_p-i-1}))\breve{x}^T\breve{x}, \\ \hat{P}_{N_p-i-1} &:= 2P_{N_p-i-1}. \end{aligned}$$

μ is the robustness bound defined in (5.3). For ease of notation, $\breve{x} := x(k+i)$ and $\breve{u} := u(k+i)$. Based on the assumption on $h_{N_p-i-1}(\cdot)$ in (5.36), the following theorem is given:

Theorem 5.4 *One-step robust MPC in the recursive prediction for perturbed system (5.1) with structured uncertainties (5.2) and (5.3) can be converted into an mp-QP problem.*

Proof: From inequality (5.37), it can be seen that

$$\begin{aligned} &(||x(k+i)||^2_Q + ||u(k+i)||^2_R + h_{N_p-i-1}(x(k+i+1))) \\ \leq\ & ||\breve{x}||^2_Q + ||\breve{u}||^2_R + ||A\breve{x} + B\breve{u}||^2_{\hat{P}_{Np-i-1}} \\ & + Z_{N_p-i-1}(A\breve{x} + B\breve{u}) + \hat{O}_{N_p-i-1}. \end{aligned} \tag{5.38}$$

Updating $J_{k+i\rightarrow k+N_p}$ by its upper bound (5.38), we have

$$J^o_{k+i\rightarrow k+N_p} = \min_{\breve{u}} \frac{1}{2}\breve{u}^T H_{\breve{u}}\breve{u} + \breve{x}^T H_{\breve{u}\breve{x}}\breve{u} + \hat{Z}\breve{u} + H_{\breve{x}\hat{o}}, \tag{5.39}$$

$$s.t. \quad \breve{u} \in \mathcal{A}_u,\ x(k+i+1) \in \mathcal{A}_x, \tag{5.40}$$

where

$$\begin{aligned} H_{\breve{x}\hat{o}} &:= (\breve{x}^T(A^T\hat{P}_{N_p-i-1}A+Q)\breve{x}+Z_{N_p-i-1}A\breve{x}+\hat{O}_{N_p-i-1}), \\ H_{\breve{u}\breve{x}} &:= (2A^T\hat{P}_{N_p-i-1}B), \\ H_{\breve{u}} &:= 2(B^T\hat{P}_{N_p-i-1}B+R)\in\mathbb{S}^m_{++}. \end{aligned}$$

B has a full-column rank and $\hat{Z} := Z_{N_p-i-1}B$ for ease of notation. To satisfy the constraint of (5.40), we need $x\,(k+i+1)$ satisfying

$$E_l x\,(k+i+1) \preceq E_r.$$

E_l and E_r are structured matrices of $\mathcal{A}_x$. Therefore it is required that

$$E_l A\breve{x} + E_l B\breve{u} \preceq E_r - E_l f\,(\cdot)\,.$$

From the condition in (5.3) and the Chebychev polyhedron $\mathcal{H}$ associated to $\mathcal{A}_x$, we can easily derive the bound of $E_l f\,(\cdot)$, denoted by $\bar{f}$ and $\underline{f}$. Set

$$f_b = \max(|\bar{f}|,\ |\underline{f}|),$$

so that $E_lA\breve{x}+E_lB\breve{u} \preceq E_r - f_b$. Consequently the optimal problem of piece objective $J_{k+i\rightarrow k+N_p}$ can be solved by minimizing a quadratic function

$$\begin{aligned} J^o_{k+i\rightarrow k+N_p} &= \min_{\breve{u}} \frac{1}{2}\breve{u}^T H_{\breve{u}}\breve{u} + \breve{x}^T H_{\breve{u}\breve{x}}\breve{u} + \hat{Z}\breve{u} + H_{\breve{x}\hat{o}}, && (5.41) \\ s.t. &\quad G_{\breve{u}}\breve{u} \preceq G_{\breve{c}} + G_{\breve{x}}\breve{x}, && (5.42) \end{aligned}$$

where

$$G_{\breve{u}} := \begin{bmatrix} E_lB \\ 0 \\ I \\ -I \end{bmatrix},\ G_{\breve{c}} := \begin{bmatrix} E_r - f_b \\ E_r \\ u_{\max} \\ -u_{\min} \end{bmatrix},\ \text{and } G_{\breve{x}} := \begin{bmatrix} -E_lA \\ -E_l \\ 0 \\ 0 \end{bmatrix}. \qquad (5.43)$$

Apparently, (5.41) is an mp-QP problem. ■

Theorem 5.5 *The analytic (explicit) solutions to the mp-QP problem in (5.41) are the piece-wise affine functions of $\breve{x}$, over the corresponding state critical regions $\mathcal{A}^j_{\breve{x}}$, where index j denotes the jth critical region within the admissible state set $\mathcal{A}_x$.*

Proof: Taking advantages of the Lagrange multiplier $\lambda \succeq 0$, we can convert the constrained mp-QP problem in (5.41) into an unconstrained mp-QP problem. Motivated by the properties of optimization duality, we separate λ into two parts, i.e., $\lambda_N = 0$ (nonactive constraints) and $\lambda_A \succ 0$ (active constraints), where $\lambda = [\lambda_N^T, \lambda_A^T]^T$. Suppose that there exists a combination of active constrains $\tilde{G}_{\breve{u}}$, $\tilde{G}_{\breve{c}}$, and $\tilde{G}_{\breve{x}}$ out of the constraints in (5.42) and the rows of $\tilde{G}_{\breve{u}}$ are linear independent. Then from the first-order Karush-Kuhn-Tucker (KKT) theorem [12], the optimal active multiplier λ_A can be represented by

$$\lambda_A = -(\tilde{G}_{\breve{u}} H_{\breve{u}}^{-1} \tilde{G}_{\breve{u}}^T)^{-1} (\tilde{G}_{\breve{u}} H_{\breve{u}}^{-1} H_{\breve{u}\breve{x}}^T + \tilde{G}_{\breve{x}}) \breve{x} - (\tilde{G}_{\breve{u}} H_{\breve{u}}^{-1} \tilde{G}_{\breve{u}}^T)^{-1} (\tilde{G}_{\breve{c}} + \tilde{G}_{\breve{u}} H_{\breve{u}}^{-1} \hat{Z}^T), \tag{5.44}$$

and the corresponding optimal input $\breve{u}$ is

$$\begin{aligned} \breve{u} =& (-H_{\breve{u}}^{-1} H_{\breve{u}\breve{x}}^T + H_{\breve{u}}^{-1} \tilde{G}_{\breve{u}}^T (\tilde{G}_{\breve{u}} H_{\breve{u}}^{-1} \tilde{G}_{\breve{u}}^T)^{-1} (\tilde{G}_{\breve{u}} H_{\breve{u}}^{-1} H_{\breve{u}\breve{x}}^T + \tilde{G}_{\breve{x}})) \breve{x} + \\ & H_{\breve{u}}^{-1} \tilde{G}_{\breve{u}}^T (\tilde{G}_{\breve{u}} H_{\breve{u}}^{-1} \tilde{G}_{\breve{u}}^T)^{-1} (\tilde{G}_{\breve{c}} + \tilde{G}_{\breve{u}} H_{\breve{u}}^{-1} \hat{Z}^T) - H_{\breve{u}}^{-1} \hat{Z}^T \\ :=& S_{N_p-i}^j \breve{x} + K_{N_p-i}^j, \end{aligned} \tag{5.45}$$

which is a piece-wise affine function of $\breve{x}$. To guarantee the conditions of KKT, we need

$$G_{\breve{u}}(S_{N_p-i}^j \breve{x} + K_{N_p-i}^j) \preceq G_{\breve{c}} + G_{\breve{x}} \breve{x} \text{ and } \lambda_A \succeq 0. \tag{5.46}$$

Consequently, we give the expression of $\mathcal{A}_x^j$ as

$$\begin{aligned} \mathcal{A}_x^j :=& \{\breve{x} \in \mathbb{R}^n \mid G_{\breve{u}}(S_{N_p-i}^j \breve{x} + K_{N_p-i}^j) \leq G_{\breve{c}} + G_{\breve{x}} \breve{x}, \\ & (\tilde{G}_{\breve{u}} \tilde{H}_{\breve{u}}^{-1} \tilde{G}_{\breve{u}}^T)^{-1} \left(\tilde{G}_{\breve{u}} H_{\breve{u}}^{-1} H_{\breve{u}\breve{x}}^T + \tilde{G}_{\breve{x}}\right) \breve{x} + (\tilde{G}_{\breve{u}} \tilde{H}_{\breve{u}}^{-1} \tilde{G}_{\breve{u}}^T)^{-1} (\tilde{G}_{\breve{c}} + \tilde{G}_{\breve{u}} H_{\breve{u}}^{-1} \hat{Z}^T) \preceq 0\}. \end{aligned} \tag{5.47}$$

In the case that there are no active constraints out of the conditions in (5.42), i.e., $\tilde{G}_{\breve{u}}$, $\tilde{G}_{\breve{c}}$, and $\tilde{G}_{\breve{x}}$ do not exist, (5.45) and condition (5.47) degenerate to ($\lambda \equiv 0$),

$$\breve{u} = -H_{\breve{u}}^{-1} H_{\breve{u}\breve{x}}^T \breve{x} - H_{\breve{u}}^{-1} \hat{Z}^T := S_{N_p-i}^j \breve{x} + K_{N_p-i}^j, \tag{5.48}$$

$$G_{\breve{u}} \breve{u} - G_{\breve{c}} - G_{\breve{x}} \breve{x} \prec 0, \tag{5.49}$$

which results in the second case of the explicit solutions to the mp-QP problem in (5.41),

$$\breve{u} = S_{N_p-i}^j \breve{x} + K_{N_p-i}^j \; (\forall \breve{x} \in \mathcal{A}_x^j), \tag{5.50}$$

where $\mathcal{A}_x^j := \{\breve{x} \in \mathbb{R}^n \mid G_{\breve{u}} \breve{u} - G_{\breve{c}} - G_{\breve{x}} \breve{x} \prec 0\}$. Both expressions (5.45) and (5.50) are piece-wise affine functions. ■

There is a shortcut to determine $\tilde{G}_{\breve{u}}$, $\tilde{G}_{\breve{c}}$, and $\tilde{G}_{\breve{x}}$, a combination of active constraints out of (5.42). Set x^o as the Chebychev center of the admissible state set $\mathcal{A}_x$ and insert x^o into the constraints in (5.42). We can uniquely determine an optimal solution $\breve{u}^o$, consequently we can find a set of constraints $\bar{G}_{\breve{u}}\breve{u} = \bar{G}_{\breve{c}} + \bar{G}_{\breve{x}}\breve{x}$. Then choose a combination of active constraints with the possible maximal full-row rank to act as $\tilde{G}_{\breve{u}}$, $\tilde{G}_{\breve{c}}$, and $\tilde{G}_{\breve{x}}$ out of $\bar{G}_{\breve{u}}$, $\bar{G}_{\breve{c}}$ and $\bar{G}_{\breve{x}}$. Based on such a combination, we can derive a critical region $\mathcal{A}_x^j$. One may ask: how to explore the rest space $R_{\mathcal{A}_x^j} := \mathcal{A}_x - \mathcal{A}_x^j$ and generate the new critical region $\mathcal{A}_x^i$? A practical method has been discussed in Section 4.3 (Chapter 4), which can guarantee that the union of all critical regions $\mathcal{A}_x^j$s cover the entire polyhedron $\mathcal{A}_x$, i.e., any point in the admissible state set corresponds to a control policy. Refer to Section 4.3 for details.

Remark 5.5 *Theorem 5.5 concludes that the optimal solution of the manipulated input $\breve{u}$ is a piece-wise affine function of $\breve{x}$, over the corresponding state critical region $\mathcal{A}_x^j$. The result is consistent to Remark 5.4.*

5.2.3 Explicit solutions to the total objective $J_{k\to k+N_p}$

In order to get $J^o_{k\to k+N_p}$ in the recursive closed-loop prediction, $J_{k\to k+N_p}$ is separated into N_p pieces and $J_{k+N_p-1\to k+N_p}$ is the first piece objective to be optimized. From (5.26), (5.3), and (5.22), it shows

$$J^o_{k+N_p-1\to k+N_p} \le ||x(k+N_p-1)||^2_{P_1}, \tag{5.51}$$

where $P_1 := ((A+BF)^T P (A+BF) + P - \nu\check{Q} + 2\mu^2\bar{\sigma}^2(P)I)$. Based on (5.51), the piece objective $J_{k+N-2\to k+N_p}$ can be defined as

$$\begin{aligned} J_{k+N_p-2\to k+N_p} &= \min_{u(k+N_p-2|k)} \max_{f(\cdot)} (||x(k+N_p-1)||^2_{P_1} + \\ &\qquad ||x(k+N_p-2)||^2_Q + ||u(k+N_p-2)||^2_R), \ (P_1 \in \mathbb{S}^n_+ \quad (5.52) \\ &s.t. \quad x(k+N_p-1) \in \mathcal{A}_x, \ u(k+N_p-2) \in \mathcal{A}_u. \quad (5.53) \end{aligned}$$

Set $x(k+N_p-2) \in \mathcal{A}_x$ as a parameter vector, so (5.52) is an mp-SQP problem. From Theorems 5.4 and 5.5, the mp-SQP problem in (5.52) can be converted into mp-QP and the corresponding optimal solution $u^o(x(k+N_p-1))$ is a piece-wise affine function, satisfying

$$u^o(k+N-2) = g_2(x(\cdot)) = S_2^j x(k+N-2) + K_2^j, \ \forall x(\cdot) \in \mathcal{A}^j_{x(\cdot)}.$$

Therefore,

$$\begin{aligned}
J^o_{k+N_p-2\to k+N_p} &= h_2(x(\cdot)) = ||x(k+N_p-1)||^2_{P_1} + ||x(k+N_p-2)||^2_Q + \\
&\quad ||S^j_2 x(k+N_p-2) + K^j_2||^2_R, \\
&\leq ||x(k+N_p-2)||^2_{P_2} + Z_2 x\,(k+N_p-2) + O_2, \qquad (5.54)
\end{aligned}$$

where

$$\begin{aligned}
P_2 &:= ||A+BS^j_2||^2_{2P_1} + Q + ||S^j_2||^2_R + (3\bar{\sigma}^2(P_1))\mu^2 I, \\
Z_2 &:= 2K^{jT}_2(B^T P_1(A+BS^j_2) + RS^j_2), \\
O_2 &:= ||BK^j_2||^2_{2P_1} + ||K^j_2||^2_R. \qquad (5.55)
\end{aligned}$$

So the optimization problem of piece objective $J_{k+N_p-3\to k+N_p}$ is updated to

$$\begin{aligned}
J_{k+N_p-3\to k+N_p} &= ||x(k+N_p-3)||^2_Q + ||u(k+N_p-3)||^2_R + h_2(x(\cdot)) \\
&= ||x(k+N_p-3)||^2_Q + ||u(k+N_p-3)||^2_R + \\
&\quad ||x(k+N_p-2)||^2_{P_2} + Z_2 x(k+N_p-2) + O_2. \qquad (5.56)
\end{aligned}$$

Notice that (5.54) shares the exactly same structure of assumption (5.36), which is imposed upon Theorem 5.5.

Theorem 5.6 *The optimal solution to the mp-SQP problem for piece objective* $J_{k+i\to k+N_p}$, $J^o_{k+i\to k+N_p} = h_{N_p-i-1}(\cdot)$, *is a quadratic function satisfying*

$$h_{N_p-i}(\cdot) = ||x\,(k+i)\,||^2_{P_{N_p-i}} + Z_{N_p-i}x\,(k+i) + O_{N_p-i},$$

where

$$\begin{aligned}
P_{N_p-i} &:= ||A+BS^j_{N_p-i}||^2_{2P_{N_p-i+1}} + Q + ||S^j_{N_p-i}||^2_R + (3\bar{\sigma}^2(P_{N_p-i+1}))\mu^2 I, \\
Z_{N_p-i} &:= 2K^{jT}_{N_p-i}(B^T P_{N_p-i+1}(A+BS^j_{N_p-i}) + RS^j_{N_p-i}), \\
O_{N_p-i} &:= ||BK^j_{N_p-i}||^2_{2P_{N_p-i+1}} + ||K^j_{N_p-i}||^2_R.
\end{aligned}$$

$S^j_{N_p-i}$ *and* $K^j_{N_p-i}$ *are the optimal solutions to piece objective* $J_{k+i+1\to k+N_p}$, *and superscript 'j' is the index to the corresponding critical region.*

Proof: Apparently, iterating the operation of (5.54) and (5.56) in the recursive pattern till instant $(k+i)$, we can derive that the sub-optimal solution $J^o_{k+i\to k+N_p}$ is a quadratic function of $x(k+i)$ over parameters expressions of P_{N_p-i}, Z_{N_p-i}, and O_{N_p-i}. ■

Based on Theorem 5.6, we remove the assumption imposed on Theorem 5.5. In a word, we overcome two challenges of RMPC in the recursive closed-loop prediction proposed in Remark 5.4.

5.2.4 Controller implementation

By iterating Theorems 5.4, 5.5, and 5.6, the final optimal inputs $u^o(k)$ can be presented by a series of piece-wise affine functions of $x(k)$, associated with $J^o_{k\to k+N_p}(x(k))$ and critical region $\mathcal{A}^j_{x(k)}$ $(1 \leq j \leq s)$. Here s denotes the number of state set partitions. Due to non-uniqueness of partitions at different prediction loops, although in the same loop the partition is disjointed, it is possible that one *critical region* $\mathcal{A}^j_{x(k)}$ corresponds to more than one expressions of $u^o(k)$ and $J^o_{k\to k+N_p}(x(k))$, i.e.,

$$\begin{aligned} u^o(k) &:= \{u^{o,\,1}(x(k)),\ u^{o,\,2}(x(k)),\ \cdots\}, \\ J^o_{k\to k+N_p}(\cdot) &:= \{J^{o,\,1}_{k\to k+N_p,}(\cdot),\ J^{o,\,2}_{k\to k+N_p}(\cdot),\ \cdots\}. \end{aligned}$$

In this case, we just evaluate all $J^{o,\,j}_{k\to k+N_p}(\cdot)$ candidates, and send $u^{o,\,j}(k)$ which leads to the smallest $J^{o,\,j}_{k\to k+N_p}(\cdot)$ to the real process. This procedure can be illustrated by a CSTR system in Section 5.4.2 – the control for an industrial MIMO system.

5.3 Algorithms of robust MPC

We can generate an efficient off-line RMPC, which just performs one-step predictions within one computation loop but can realize the functions of RMPC with the arbitrary horizon length. The algorithm has more tuning freedom over offline infinite horizon RMPC (IH-RMPC).

Algorithm 1

1. Given the perturbed system in (5.1) with structured uncertainties (5.3) and bounded disturbances (5.2), derive the state admissible set $\mathcal{A}_x$ from system input/output physical characteristics.

2. Execute closed-loop stability analysis. Select the eligible tuning variables $\check{Q}$ and ν to obtain the satisfactory tradeoff between the closed-loop performance and robustness. Solve an algebraic Riccati equation and get the terminal weighting P and the terminal feedback gain F. Derive the optimal expressions of g_1 and h_1, and set $i = N_p - 2$ initially.

3. Solve the mp-SQP problem for piece objective $J_{k+i \rightarrow k+N_p}$. Store the expressions of the optimal solutions of $g_{N_p-i}(\cdot)$, $h_{N_p-i}(\cdot)$, and the corresponding $\mathcal{A}_x$ partition. Set $i = i - 1$.

4. Check whether $i = 0$. If yes, store optimal solution $u^o(k) = g_{N_p}(x(k))$, $J^o_{k \rightarrow k+N_p} = h_{N_p}(x(k))$, and $\mathcal{A}_x$ partition $\{\mathcal{A}^1_{x(k)}, \cdots, \mathcal{A}^s_{x(k)}\}$. Purge the memories for intermediate variables $g_{N_p-i}(\cdot)$, $h_{N_p-i}(\cdot)$, and other partitions. If $0 < i \leq N_{p-2}$, go to Step 3.

5. Exit the loop. Send the expressions of $u^o(k)$ and $J^o_{k \rightarrow k+N_p}$ to the evaluation block of a real precess and prepare for controller implementation.

6. From the state measure of $x(k)$, locate the state position. Supposing $x(k) \in \mathcal{A}^j_{x(k)}$, evaluate all of $J^{o,\,j}_{k \rightarrow k+N_p}(x(k))$ candidates and send $u^{o,\,j}(k)$ which leads to the smallest $J^{o,\,j}_{k \rightarrow k+N_p}(x(k))$ to the real process.

7. If $||x(k)|| < \varepsilon$, exist. Otherwise update $x(k)$ to $x(k+1)$ and go to Step 6. "ε" is a prespecified positive scalar and $||x(k)||$ is the proper norm of $x(k)$ as the measurement rule of system performance.

8. End procedure.

5.3.1 Computational complexity analysis

The maximal order of the uncertain terms of predicted states/outputs equals to 1, so the computational complexity is dramatically reduced. From the analysis of [6], we know that n_s, the number of partitions of $\mathcal{A}_x$, is dependent on the state dimension n, the number of regulated variables n_u ($n_u = n_u \times (N_p - 1)$), and the number of the combinations of active constraints n_c. Set the dimension of the Lagrange multipliers equal to n_λ and that of the active multipliers be n_{λ_A}. The worst case of n_c equals to

$$\bar{n}_c = \begin{pmatrix} n_\lambda \\ n_{\lambda_A} \end{pmatrix} = \frac{n_{\lambda_A}!}{n_\lambda! \left(n_\lambda - n_{\lambda_A}\right)!}. \tag{5.57}$$

Considering the existence of *Case II* in Theorem 5.5 and the probability of the adjacent partitions $\mathcal{A}_x^j$s associated with the same feedback gains (consequently these $\mathcal{A}_x^j$s can be combined), n_c is much less than $\bar{n}_c$.

Now let us think about the iteration times of mp-SQP optimization with the horizon N_p. Because of the nature of RMPC formulation in the recursive closed-loop prediction, we perform Theorems 5.4, 5.5 and 5.6 $(N_p - 2)$ times and solve an algebraic Riccati equation. Within each optimization loop, if the partition number of $\mathcal{A}_x$ is n_s, then the total optimization number is equal to

$$n_{tol} = (N_p - 2) \times n_s. \tag{5.58}$$

The number of the required intermediate variables $g_{N_p-i-1}(\cdot)$, $h_{N_p-i}(\cdot)$, and partitions equals to $n_{int} = 3n_s(N_p - 2)$. Here we just discuss the worst case of computational complexity. In fact, the real values of n_{tol} and n_{int} are much less than those of the worst case. From the complexity analysis, it is obvious that this strategy is time-consuming and memory-consuming. Fortunately, one point should be noticed that this procedure is performed offline. In the procedure of online implementation, only a function evaluation block is necessary. The expressions of $u^o(k)$, $J^o_{k\to k+N_p}(x(k))$, and the associated state space partitions are stored on the field spot and all other intermediate variables are rejected. Therefore, sacrificing the offline computational complexity and dramatically improving online implementation is a valuable strategy.

5.3.2 Offset-free robust MPC

In Section 5.1, we mentioned that by solving linear or quadratic programming, problems of robust offset-free control can be converted into a robust regulation problem. In this sub-section, we will elaborate on this point.

Because of the presence of uncertainties and disturbances, it is unrealistic to force the terminal state or output to follow the prespecified reference r without any static errors. Therefore we just expect to manipulate the terminal states or outputs, namely the state x_s and output y_s in the best way of certain measurement policies.

Since small perturbations are inevitable, here we propose a QP problem to calculate x_s, static input u_s and nominal disturbance $\bar{d}$:

$$\begin{aligned}
\left[x_s,\ u_s,\ \bar{d}\right] &:= \arg\min_{x_s,\ u_s,\ \bar{d}} \|e_s\|^2_{Q_s}, && (5.59)\\
s.t.\ & \bar{d}^T W_d \bar{d} \leq 1, && (5.60)\\
& y_s = CAx_s + CBu_s + C_d\bar{d}, && (5.61)\\
& e_s := y_s - r,\ u_s \in \mathcal{A}_u \text{ and } y_s \in \mathcal{A}_y, && (5.62)
\end{aligned}$$

where e_s is for the static tracking error and $Q_s \in \mathbb{S}^q_+$ is the objective weighting. (5.59) is defined as a constrained QP problem and there exist lots of solvers to such a problem. Moreover implementing Schur complements, the problem can be easily converted into a generalized eigenvalue problem (EVP) [12], for which LMI solvers exist for solutions.

Based on the values of u_s, x_s, y_s and $\bar{d}$, we can perform the similarity transformation to system (5.1) and derive a shifted system,

$$\begin{aligned}
\tilde{x}(k+1) &= A\tilde{x}(k) + B\tilde{u}(k) + f(\cdot),\\
\tilde{y}(k) &= C\tilde{x}(k) + C_d\tilde{d}(k), && (5.63)
\end{aligned}$$

where $\tilde{x}(k) := x(k) - x_s$, similar to the definitions of $\tilde{u}(k)$, $\tilde{y}(k)$, and $\tilde{d}(k)$. Therefore, to achieve the offset-free control to the system in (5.1) is equivalent to regulate $\tilde{x}(k)$, the state of (5.63), to the origin.

5.4 Simulation examples

Here we will use two simulation examples to demonstrate the effectiveness of explicit RMPC in the recursive closed-loop prediction: a double integrator system and a linearized continuous-stirred-tank-reactor (CSTR) system (a 2-by-2 system). The Hybrid Toolbox is used again to visualize the state space partition.

5.4.1 Double integrator system

The double integrator can be represented by a state space model with two eigenvalues located at 1,

$$x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} d(k), \qquad (5.64)$$

where $d(k)$ is a random variable simulated by Matlab function "*rand*," and satisfies $\|d(k)\|_2 \leq 0.1\,\|x(k)\|_2$ (internal uncertainties). In the sequel, we first indicate the effects of uncertainties on nominal MPC algorithm, and then design a RMPC regulator in the recursive closed-loop prediction to suppress $d(k)$. A nominal MPC regulator is created by the approach developed by Bemporad [6]. The initial conditions are given by

$$\begin{aligned} P &= \begin{bmatrix} 2.6235 & 1.6296 \\ 1.6296 & 2.6457 \end{bmatrix}, \; F = \begin{bmatrix} -0.6136 & -1.6099 \end{bmatrix}, \\ Q &= I, \; R = 0.01, \; N_u = \; N_y = 2, \; x_0 = [-2, 1]^T, \; u_0 = 0, \\ \mathbf{-1} &\preceq \; u(t) \preceq \mathbf{1}, \; -\inf \preceq y(t) \preceq \inf \; \text{(no output/state constraints)}. \end{aligned} \tag{5.65}$$

Fig. 5.4 shows the state space partition, overlapped by the state convergent trajectory. It can be seen that the terminal states keep oscillating around the origin and the amplitude is quite large. Fig. 5.5 gives more illustrative results: The input keeps switching from the upper-bound to the lower-bound, consequently leading to the oscillation and big overshoot in states. Therefore, nominal explicit MPC cannot suppress external uncertainties.

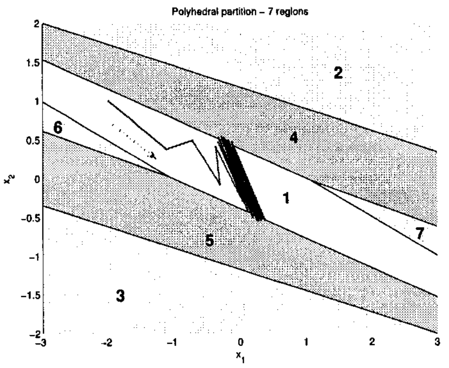


Figure 5.4: State space partition, overlapped by the state trajectory (nominal MPC)

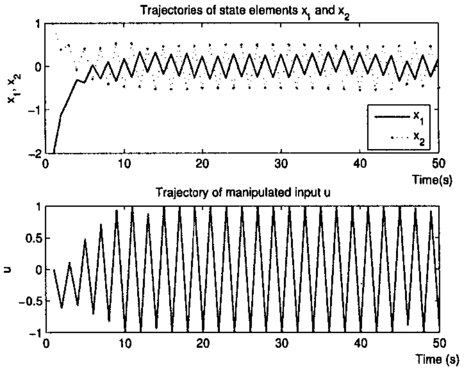


Figure 5.5: The trajectories of states and input (nominal MPC)

Now design the robust explicit MPC in the recursive closed-loop prediction.

Based on Theorem 5.3, initial conditions can be determined as,

$$\begin{aligned} P &= \begin{bmatrix} 7.7477 & 4.9537 \\ 4.9537 & 8.1698 \end{bmatrix}, \hat{Q} = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 1.2 \end{bmatrix}, \\ F &= \begin{bmatrix} -0.6056 & -1.6044 \end{bmatrix}, \ \nu = 2, \end{aligned} \tag{5.66}$$

and keep other parameters the same as in (5.65). Given the values of $\hat{Q}$ and ν, we derive the upper bound of system robustness, $\bar{\mu} = 0.1549$ (if $\nu = 1.5$, $\bar{\mu} = 0.1433$). From this point of view, our affine offline RMPC have 54.9% (43.3% if $\nu = 1.5$) robustness margin, due to original $\mu = 0.1$. Because $N_p = 2$, we need to perform Theorems 5.4, 5.5 and 5.6 in two loops and solve an algebraic Riccati equation to derive the optimal piece objective $J^o_{k+2 \to k+3}$. Eq. (5.67) lists the critical intermediate parameters. The constant terms $O_{N_p - i}$ are omitted because their value does not affect the optimal solutions to piece objectives (refer to (5.55)). By solving piece objective $J_{k+1 \to k+3}$, the state space is partitioned into 3 critical regions $\mathcal{A}^j_{x(k+1)}$ (see Fig. 5.6). Therefore, optimizing $J_{k \to k+3}$ in all $\mathcal{A}_{x(k+1)}$, we derive three different partitions of $\mathcal{A}_{x(k)}$ which are shown in Figs. 5.7 - 5.9.

$$\begin{aligned} P_{N_p-1} &= \begin{bmatrix} 13.8286 & 9.4977 \\ 9.4977 & 13.8507 \end{bmatrix}, \ Z_{N_p-1} = \begin{bmatrix} 0 & 0 \end{bmatrix} \\ P_{N_p-2,1} &= \begin{bmatrix} 46.9773 & 46.8064 \\ 46.8064 & 112.9749 \end{bmatrix}, P_{N_p-2,2} = \begin{bmatrix} 46.9961 & 46.8526 \\ 46.8526 & 113.0884 \end{bmatrix} \\ P_{N_p-2,3} &= \begin{bmatrix} 46.9773 & 46.8064 \\ 46.8064 & 112.9749 \end{bmatrix}, \ Z_{N_p-2,1} = \begin{bmatrix} 0 & 0 \end{bmatrix}, \\ Z_{N_p-2,2} &= \begin{bmatrix} -18.9954 & -46.6968 \end{bmatrix}, \ Z_{N_p-2,3} = \begin{bmatrix} 18.9954 & 46.6968 \end{bmatrix}. \end{aligned} \tag{5.67}$$

Using the policy proposed in Section 5.2.4, implement optimal input $u^o(k)$. The performance of the resulting closed-loop system is shown in Figs. 5.10 and 5.11. It is obvious that RMPC is capable of suppressing internal uncertainty $d(k)$ completely. Using a laptop with Pentium 4 processor and 512MB Ram, the simulation only costs $(0.37 + 0.33 + 0.31 + 0.27) = 1.28$ seconds for offline-optimization (simulation length equals to 50). Therefore, the algorithm is quite efficient.

To illustrate the precision of our algorithm, we use an existing offline IH-RMPC algorithm, Bisection RMPC to design the controller for system (5.64) [103]. The essential idea of Bisection RMPC is creating a serious of ellipsoidal invariant sets, which correspond to a serious of structured matrices Q_i and feedback gains F_i, and

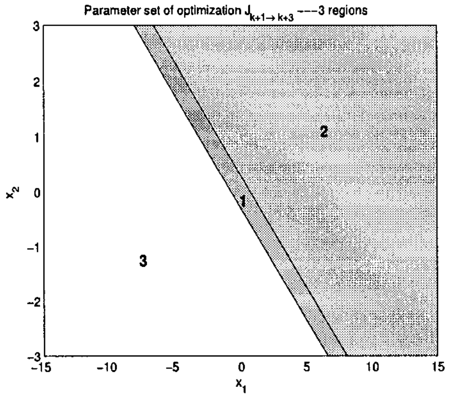


Figure 5.6: State space partition for piece objective $J_{k+1 \to k+3}$ (Loop I)

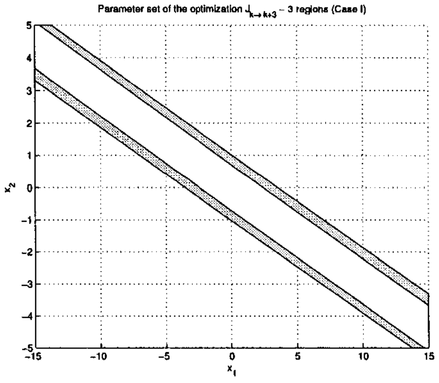


Figure 5.7: State space partition for piece objective $J_{k \to k+3}$ (Case I, Loop II)

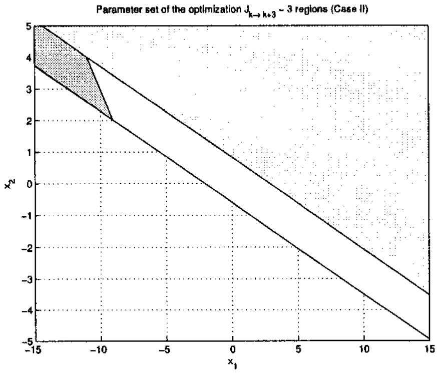


Figure 5.8: State space partition for piece objective $J_{k \to k+3}$ (Case II, Loop II)

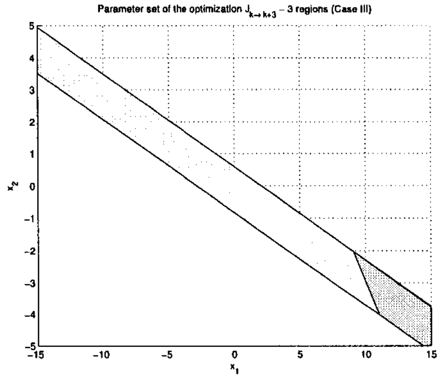


Figure 5.9: State space partition for piece objective $J_{k \to k+3}$ (Case III, Loop II)

forcing these ellipsoidal sets to shrink along the state trajectories. This procedure is completed offline and both Q_i and F_i compose an data table. By searching the table in the bisection manner, we can realize online implementation. But one of disadvantages is inherited from its origination — online infinite horizon RMPC (IH-RMPC) [45]: The fixed control horizon N_u and prediction horizon N_y, equal to infinite, weaken the tuning freedom of RMPC strategies. Moreover this method constructs a looking-up table which, unfortunately, leads to another three disadvantages:

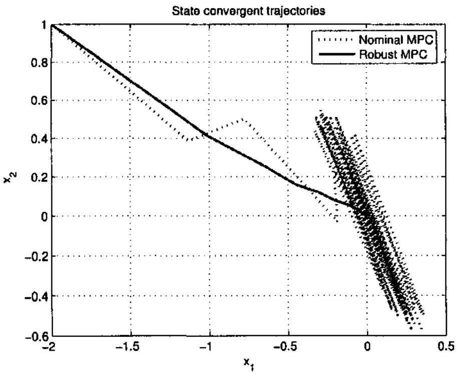


Figure 5.10: State phase planes of Nominal MPC and Robust MPC

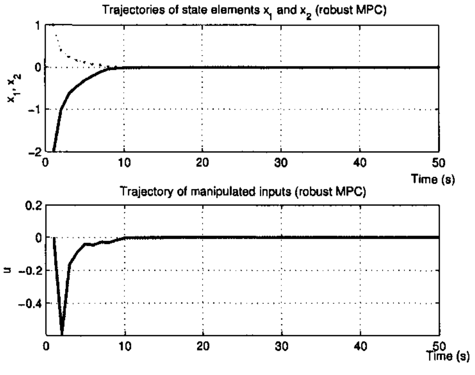


Figure 5.11: The trajectories of states and inputs (Robust MPC)

1. The size of the table is dependent on the selection of initial states. Therefore, in order to satisfy general cases, the table should be of a quite large scale.

2. While performing implementation, every iteration loop only sends an approximation solution to the objective to the real process. Consequently, the method can not take full advantage of the potential of RMPC.

3. It is just capable to handle convergent disturbances $d(k)$, i.e.,$\lim_{k\to\infty} d(k) = 0$. This is a quite conservative assumption.

Fig. 5.12 shows the shrinking invariant sets, superposed by the state phase plane. From Figs. 5.13 - 5.15, it can be seen that the optimal performance derived from affine RMPC (our method) is better than that of Bisection RMPC.

5.4.2 Linearized CSTR system

Consider a first order CSTR (an industrial MIMO plant), where chemical species A react to form species B: $A \rightarrow B$. Fig. 5.16 illustrates the physical structure of the system, where C_{Ai} is the input concentration of a key reactant A, C_A is the output concentration of A, T is reaction temperature, and T_c is the cooling medium temperature [93]. The dynamics of this process can be expressed as

$$\begin{aligned} \frac{dC_A}{dt} &= a_{11}C_A + a_{12}T + b_{11}T_c + b_{12}C_{Ai} \ , \\ \frac{dT}{dt} &= a_{21}C_A + a_{22}T + b_{21}T_c + b_{21}C_{Ai} \ , \end{aligned} \qquad (5.68)$$

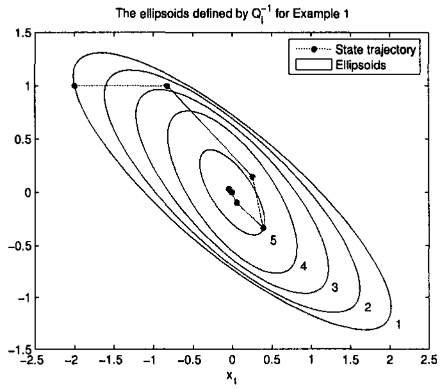


Figure 5.12: The shrinking invariant sets, derived by Bisection RMPC)

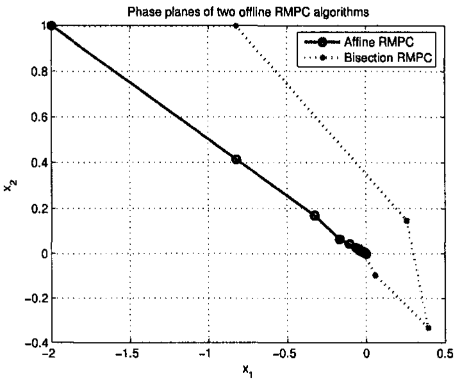


Figure 5.13: Comparison of phase planes derived by two offline RMPC algorithms.

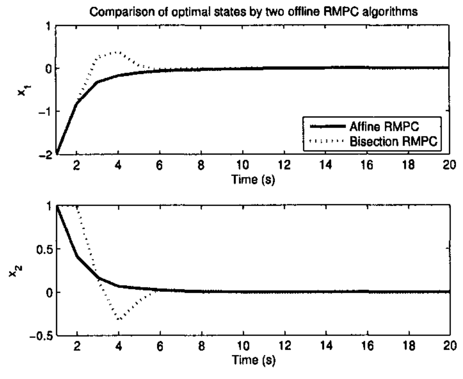


Figure 5.14: Optimal states derived by two offline RMPC algorithms

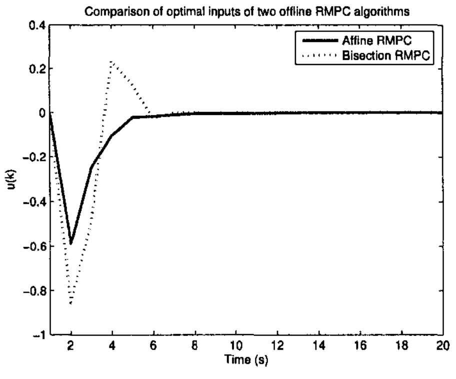


Figure 5.15: Comparison of optimal inputs derived by two offline RMPC algorithms.

Set the system state $x = [C_A,\ T]^T$, the input $u = [T_c,\ C_{Ai}]^T$, and then the discretized state space model is given by

$$\begin{aligned} x(k+1) &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} x(k) + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} u(k), \\ y(k) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} d(k), \end{aligned} \tag{5.69}$$

where $d \in [-0.1, 0.1]$ is an unknown time-varying disturbance. By experiments, we find some uncertain terms inherent in parameters a_{11} and a_{22}. The nominal values

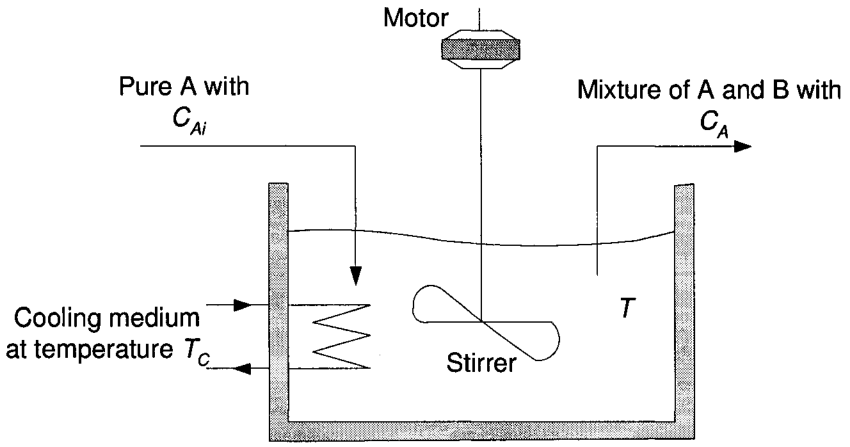


Figure 5.16: A continuous stirred-tank reactor.

are

$$\begin{aligned}
\bar{A} &= \begin{bmatrix} \bar{a}_{11} & \bar{a}_{12} \\ \bar{a}_{21} & \bar{a}_{22} \end{bmatrix} = \begin{bmatrix} 0.9719 & -0.0013 \\ -0.0340 & 0.8628 \end{bmatrix}, \\
B &= \begin{bmatrix} \bar{b}_{11} & \bar{b}_{12} \\ \bar{b}_{21} & \bar{b}_{22} \end{bmatrix} = \begin{bmatrix} -0.0839 & 0.0232 \\ 0.0761 & 0.4144 \end{bmatrix}, \\
B_d &= \begin{bmatrix} bd_1 & bd_2 \end{bmatrix}^T = \begin{bmatrix} 0.4349 & -0.0018 \end{bmatrix}^T,
\end{aligned}$$

and real values are $a_{11} = \bar{a}_{11} + \tilde{a}_{11}$, $a_{22} = \bar{a}_{22} + \tilde{a}_{22}$ where $\tilde{a}_{11}$, $\tilde{a}_{22}$ are both time-varying variables bounded in the range $[-0.1,\ 0.1]$ (simulated by "*rand*" function in programming). Recasting system (5.69) into the form of the structured model uncertainties defined in (5.3) and (5.4), so that we have

$$x(k+1) = \bar{A}x(k) + Bu(k) + f(x(k),\ \tilde{a}_{11},\ \tilde{a}_{22}), \tag{5.70}$$

where

$$\|f(\cdot)\| \leq 0.1 \|x(k)\|_2 .$$

The system constraints are given by: $u_{\min} = -u_{\max} = [-1,\ -1]^T$, and $y_{\min} = -y_{\max} = [-1,\ -1]^T$. We can base on the piece-wise linear norm of $d(k)$ or Voronoi sets to derive the admissible state set as follows:

$$\mathcal{A}_x := \{x \in \mathbb{R}^n |\ M_l x \preceq M_r\} \text{ (Using the approach of Voronoi sets)},$$

where

$$M_l = \begin{bmatrix} 3.6 & 0 \\ -3.6 & 0 \\ 0 & 3.6 \\ 0 & -3.6 \end{bmatrix}, \; M_r = \begin{bmatrix} 3.24 \\ 3.24 \\ 3.24 \\ 3.24 \end{bmatrix}.$$

Fig. 5.17 shows the admissible state polyhedron.

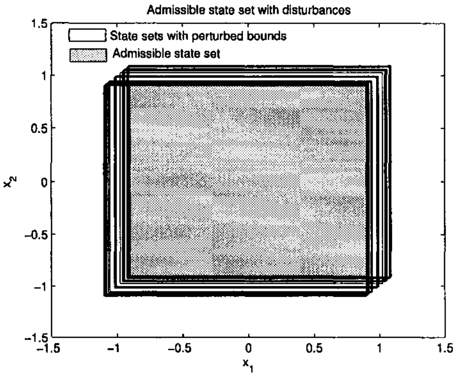


Figure 5.17: Admissible state set with disturbances

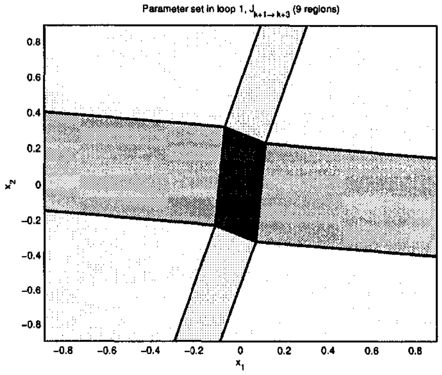


Figure 5.18: State space partitions in Loop I

The initial conditions are prespecified as

$$\begin{aligned} P &= \begin{bmatrix} 9.1123 & 0.0053 \\ 0.0053 & 4.5746 \end{bmatrix}, \; F = \begin{bmatrix} 4.4878 & -0.3994 \\ -0.7974 & -1.7797 \end{bmatrix}, \\ Q &= 2I, \; R = 0.1I, \; N_u = N_y = 2, \; x_0 = [0.4, 0.4]^T, \; u_0 = 0, \\ \hat{Q} &= \begin{bmatrix} 2 & 0.1 \\ 01 & 2.2 \end{bmatrix}, \; \nu = 1 \text{ (tuning parameters).} \end{aligned} \tag{5.71}$$

P and F are the terminal weighting and terminal feedback gain. With the tuning parameters $\hat{Q}$ and ν, we can derive the robustness bound $\bar{\mu} = 0.1642$ (robustness margin equals to 64.2%). As in the same fashion of Section 5.4.1, total objective $J_{k\to k+3}$ is optimized in two loops: Loop I is for piece objective $J_{k+1\to k+3}$ associated with $J^o_{k+2\to k+3}$; and Loop II is for $J_{k\to k+3}$ associated with $J^o_{k+1\to k+3}$. The critical intermediate parameters are given in Table 5.1 (here subscripts stand for the number of loops). In loop I, the admissible state set is partitioned into 9 regions (Fig. 5.18) so that the partition in Loop II has 9 possibilities (Fig. 5.19).

It can be seen that the union of all the partitions covers the whole admissible state space $\mathcal{A}_x$. That is to say that we can guarantee the feasibility of our algorithm.

Table 5.1: Intermediate parameters for the Loop II optimization of CSTR offline MPC

1	$P_1 = \begin{bmatrix} 11.8077 & -0.1521 \\ -0.1521 & 4.0773 \end{bmatrix}, \quad Z_1 = \begin{bmatrix} 0 & 0 \end{bmatrix}$		
2	$P_{2,1} = \begin{bmatrix} 19.3932 & -0.1970 \\ -0.1970 & 13.1584 \end{bmatrix}$ $Z_{2,1} = \begin{bmatrix} 0 & 0 \end{bmatrix}$	$P_{2,2} = \begin{bmatrix} 30.5212 & -0.4242 \\ -0.4242 & 14.4559 \end{bmatrix}$ $Z_{2,2} = \begin{bmatrix} 2.2654 & 2.3489 \end{bmatrix}$	$P_{2.3} = \begin{bmatrix} 30.5212 & -0.4242 \\ -0.4242 & 14.4559 \end{bmatrix}$ $Z_{2,3} = \begin{bmatrix} -2.2654 & -2.3489 \end{bmatrix}$
	$P_{2,4} = \begin{bmatrix} 30.5212 & -0.4242 \\ -0.4242 & 14.4559 \end{bmatrix}$ $Z_{2,4} = \begin{bmatrix} 1.6748 & -3.4690 \end{bmatrix}$	$P_{2,5} = \begin{bmatrix} 30.5212 & -0.4242 \\ -0.4242 & 14.4559 \end{bmatrix}$ $Z_{2,5} = \begin{bmatrix} -1.6748 & 3.4690 \end{bmatrix}$	$P_{2,6} = \begin{bmatrix} 30.5077 & -0.5572 \\ -0.5572 & 13.1456 \end{bmatrix}$ $Z_{2,6} = \begin{bmatrix} 2.0108 & -0.1596 \end{bmatrix}$
	$P_{2,7} = \begin{bmatrix} 30.5077 & -0.5572 \\ -0.5572 & 13.1456 \end{bmatrix}$ $Z_{2,7} = \begin{bmatrix} -2.0108 & 0.1596 \end{bmatrix}$	$P_{2,8} = \begin{bmatrix} 21.6074 & 2.1097 \\ 2.1097 & 13.7356 \end{bmatrix}$ $Z_{2,8} = \begin{bmatrix} -1.3396 & -2.6121 \end{bmatrix}$	$P_{2,9} = \begin{bmatrix} 21.6074 & 2.1097 \\ 2.1097 & 13.7356 \end{bmatrix}$ $Z_{2,9} = \begin{bmatrix} 1.3396 & 2.6121 \end{bmatrix}$

More specifically, in Fig. 5.19, Partition 2 is symmetric to Partition 3 with respect to the origin, and it is similar to Partitions 4 & 5, Partitions 6 & 7, and Partitions 8 & 9. Therefore it is possible that one state measurement $x(k)$ corresponds to more than one optimal solutions $u^o(k)$ and suboptimal solutions $J^o_{k\to k+3}$. In this case, we just select the smallest $J^{o,\ j}_{k\to k+3}$ (where "j" denotes the index of the smallest variable) and the corresponding $u^{o,\ j}(k)$ as control signals for implementation. From Fig. 5.19, we can figure out that the total number of partitions equals to

$$13 + 12 \times 2 + 1 \times 2 + 11 \times 2 + 12 \times 2 = 85. \tag{5.72}$$

But after performing combination, we find that the state space partition has only 14 regions. In order to demonstrate the different control results of RMPC on system internal uncertainties and external disturbances, the simulation is separately performed under two conditions: 1. only system internal uncertainties $\tilde{a}_{11}$ and $\tilde{a}_{22}$ are considered; 2. both internal uncertainties $\tilde{a}_{11}$, $\tilde{a}_{22}$ and external disturbances $d(k)$ are considered. Figs. 5.20 - 5.21 show the results under Condition I, and Figs. 5.22 - 5.23 are the results under Condition II. From the figures, we can say that RMPC in the recursive closed-loop prediction can eliminate the internal uncertainties completely, which is consistent to the results of double integrator. But for external uncertainties, the controller can confine the output perturbations within a small region, but can not suppress 100%. Figs. 5.21 and 5.23 illustrate the state convergent trajectories under both Condition I and Condition II, where the vertical axis ($z - axis$) indexes the number of partitions. Apparently, from these two figures, we can see that during implementation, manipulated inputs keep jumping over $\{u^{o,1},\ u^{o,\ 2},\ \cdots\}$. The whole procedure is completed within 8.3 seconds. Thus, RMPC application in the

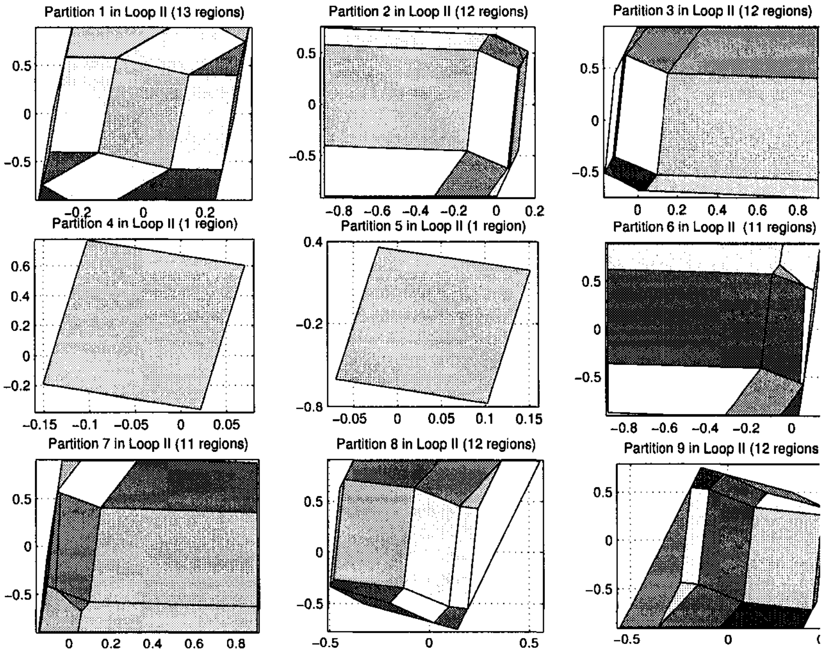
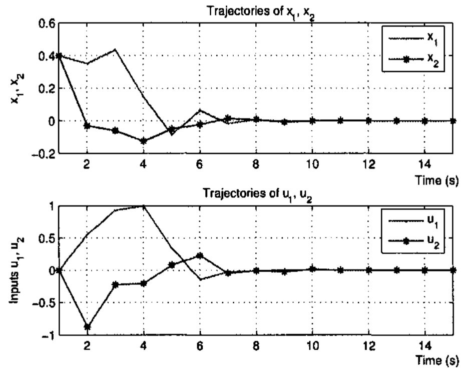


Figure 5.19: State space partitions in Loop II (x-axis is x_1 and y-axis is x_2)



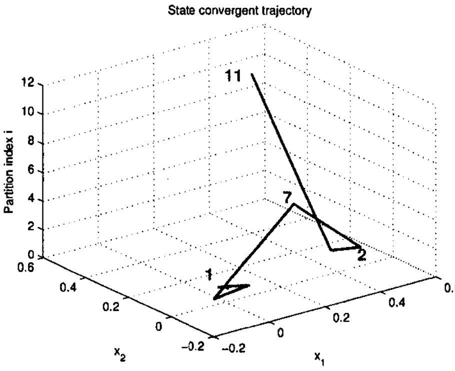


Figure 5.20: The trajectories of state elements x_1,x_2 (no output disturbances)

Figure 5.21: State convergent trajectory (without output disturbances)

CSTR industrial system is efficient, flexible, and reliable.

In the same fashion, we use the bisection RMPC algorithm to control the CSTR system in (5.69). To illustrate the different effects of internal and external uncer-

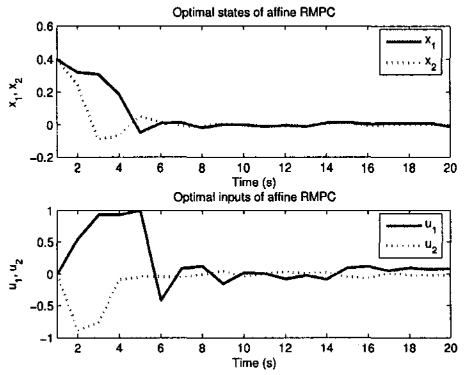


Figure 5.22: The trajectories of optimal states and inputs (with output disturbances)

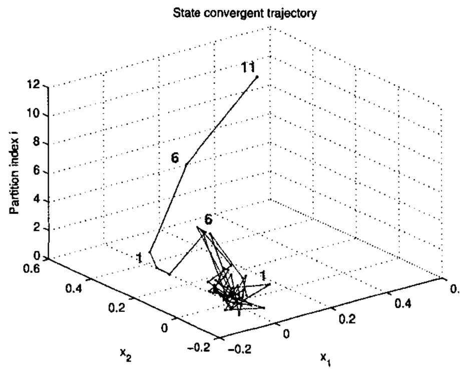


Figure 5.23: State convergent trajectory (with output disturbances)

tainties on system dynamics, we first set $d(k) = 0$ and perform controller design by the approaches. Fig. 5.24 shows the shrinking ellipsoidal sets for the CSTR system.

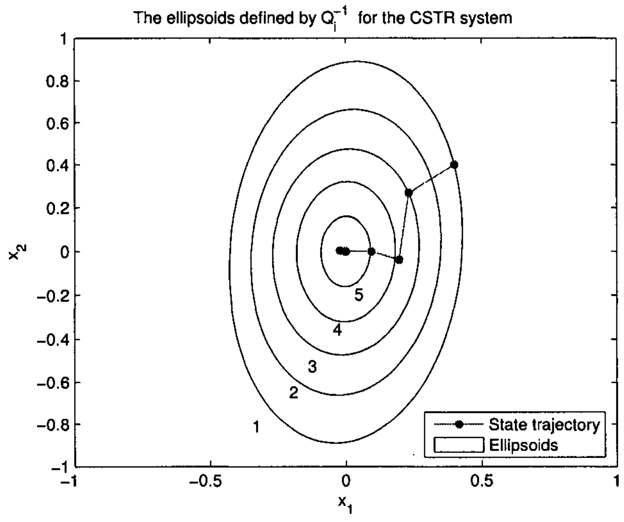


Figure 5.24: The shrinking ellipsoidal sets for the CSTR system

From the comparison results of Figs. 5.25 - 5.26, we can see that affine RMPC (our algorithm) derives better control performance, although it is not extreme improvement. However, if we set $d(k) \neq 0$ and repeat the same process, we will get more

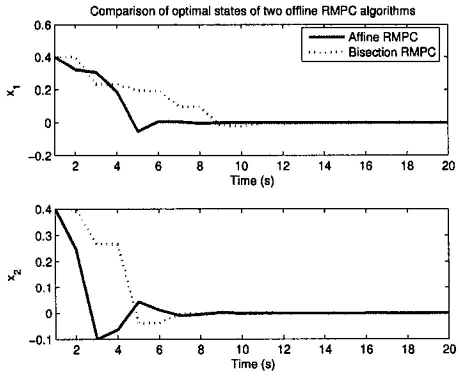


Figure 5.25: Optimal states derived by two algorithms disregarding $d(k)$ (CSTR)

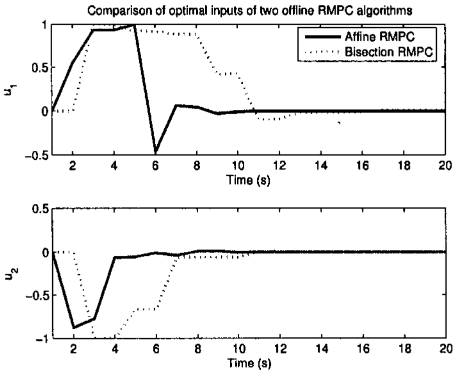


Figure 5.26: Optimal inputs derived by two algorithms disregarding $d(k)$.

illustrative results (Figs. 5.27 - 5.28). The bisection RMPC cannot generate stable control any more, but our method still result to acceptable performance.

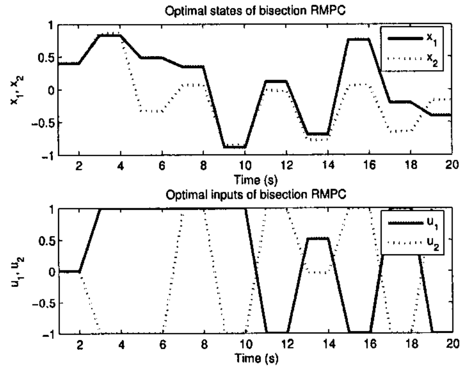


Figure 5.27: Unstable optimal states and inputs derived by bisection robust MPC (CSTR)

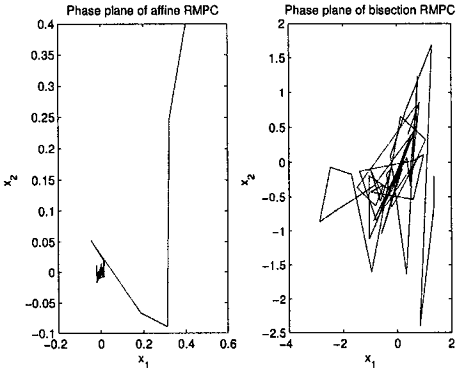


Figure 5.28: Comparison of phase planes derived by two algorithms in the presence of $d(k)$.

5.5 Conclusions

In this chapter, we developed a robust MPC algorithm with an arbitrary prediction horizon for constrained systems with structured uncertainties and bounded disturbances. It is featured by four aspects: 1. the manipulated input is optimized offline and implemented online, so that the controller is flexible and efficient. 2. The closed-loop asymptotic stability can be guaranteed automatically, and by fully partitioning the admissible state set, feasibility of the algorithm is also can be guaranteed. 3. Two fabricated tuning variables $\check{Q}$ and ν are capable of adjusting the tradeoff between system closed-loop performance and robustness. 4. The unique prediction pattern, recursive closed-loop prediction, dramatically reduces the computational complexity of robust MPC formulations. Based on the one-step prediction, it is able to construct robust MPC with the arbitrary horizon length, but one-step prediction is sufficient.

Using mp-SQP techniques, the explicit (analytic) solution to robust MPC can be established by the piece-wise affine functions of state measurement $x(k)$, associated with corresponding critical regions $\mathcal{A}^j_{x(k)}$. The regions are determined by a series of element-wise inequalities and their union covers the whole admissible state region, i.e., each point in the admissible state set corresponds to a control policy. Some novel mathematical strategies are introduced into robust MPC areas, including construction of the tuning parameters $\check{Q}$ and ν, reconfiguration of structured uncertainties, piece-wise linear norm and Voronoi sets, and the uniform structure of piece objectives. The simulation examples illustrate that the algorithm is efficient, reliable, and flexible. It is capable of eliminating internal uncertainties completely and reducing the output disturbances dramatically. The offline affine robust MPC can be applied to different kinds of fast or slow industrial MIMO systems. This chapter is summarized in our publication [18].

Chapter 6

Robust state observer

Chapter 3 developed an FH-RMPC algorithm using LMIs and Chapter 5 constructed an explicit FH-RMPC approach using recursive closed-loop prediction. These method, however, assume that the system states are fully measurable, which obviously is not always the case. To remove this limitation, Chapter 6 focuses on robust observer design in the presence of system internal and external uncertainties. Two approaches are proposed in the sequel: one formulates robust observer design as a Maximize Determinant Optimization problem and employs the principle of invariant sets to reduce the computational complexity; and the other converts the design into an mp-SQP problem and leads to robust moving horizon state observers (RMHSOs) with both open-loop and closed-loop prediction. Both approaches can guarantee the convergence of estimation errors in the sense of Lyapunov. The former ensures the estimation errors converging to an ellipsoidal invariant set along system trajectories. The latter constructs two tuning parameters, namely arrival weighting and the arrival estimation gain, to adjust the tradeoff between observer's robustness and stability.

6.1 Robust observer using MAXDET programming

State observer theory has been widely used in many branches of science and engineering and there exists a rich collection of state observer design methods and algorithms. The Luenberger observer [58] and the Kalman filter [42], as two most successful observer strategies, were developed for deterministic systems and stochastic systems, respectively. The former is limited to systems with accurate models neglecting both internal and external uncertainties; the latter considers external uncertainties as white noises, but modelling errors in Kalman filtering systems often lead to poor performance [74]. To incorporate modelling uncertainties with state observer, robust observer design has received considerable attention in the past decade, and different kinds of robust observers were published, e.g., unknown input observer (UIO) [108], spectrum assignment observer [110], LMI based observer [51], high-gain robust observer [60], and input-output observer [61]. However, few of them incorporate both internal and external uncertainties with observer design.. In this section, we will convert observer design into a Maximize Determinant (MAXDET) optimization problem and involve both internal and external uncertainties in observer design. Refer to Section 3.2.1 for the definition of MAXDET programming.

6.1.1 Formulation

Consider a system with structured internal uncertainties and external disturbances

$$\begin{aligned} x(k+1) &= Ax(k)+Bu(k)+B_\theta\theta(k)+B_d d(k), \\ \xi(k) &= T_1 x(k)+T_2 u(k), \\ \theta(k) &= \Delta_\theta(k)\xi(k), \\ y(k) &= Cx(k)+C_d d(k), \end{aligned} \tag{6.1}$$

where $x(k)\in\mathbb{R}^n$ stands for the state, $u(k)\in\mathbb{R}^m$ for the input, and $y(k)\in\mathbb{R}^q$ for the output. $d(k)\in\mathbb{R}^l$ is a combination of input, output, and state disturbances satisfying

$$d^T(k)W_d d(k) < 1, \tag{6.2}$$

where W_d is a positive symmetric matrix. A, B, B_θ, T_1, T_2, C, B_d, and C_d are all constant matrices of appropriate dimensions, and $\Delta_\theta(k)$ represents the time-varying internal uncertainties in the feedback loop, for which the maximal singular value is bounded by 1, so

$$\theta^T(k)\,\theta(k) < \xi^T(k)\,\xi(k)\,. \tag{6.3}$$

We first make an assumption:

A1) *Matrix C_d has a full column rank. In other words, the dimension of the disturbance $d(k)$ is no more than that of the output $y(k)$, namely $l \leq q$.*

Based on A1), it is possible to represent unknown disturbances in the terms of the outputs and states in (6.1),

$$d(k) = \Phi_1\left(y(k) - Cx(k)\right), \tag{6.4}$$

where $\Phi_1 = (C_d^T C_d)^{-1} C_d^T$. The goal of the observer design is to develop a filter which can provide the state approximation $\hat{x}(k)$ from the current or/and past output and input data and guarantee the observer error, $e(k) = x(k) - \hat{x}(k)$, as small as possible in some criterion. The observer design can be described as follows:

Given an ellipsoidal set $\mathcal{E}_{e(k-1)}$ with respect to the old estimation $\hat{x}(k-1)$,

$$\mathcal{E}_{e(k-1)} = \{e(k-1) \mid e(k-1)^T P_{k-1} e(k-1) < 1 \text{ and } P_{k-1} > 0\}, \tag{6.5}$$

where $P_{k-1} \in \mathbb{S}_+^n$, there exists another ellipsoidal set $\mathcal{E}_{e(k)}$ for the current state estimation $\hat{x}(k)$ and the condition

$$(x(k) - \hat{x}(k))^T P_k (x(k) - \hat{x}(k)) < 1, \tag{6.6}$$

holds for some $P_k \in \mathbb{S}_+^n$, and also the volume of $\mathcal{E}_{e(k)}$, with the condition $\mathcal{E}_{e(k)} \subseteq \mathcal{E}_{e(k-1)}$, should be as small as possible. Fig. 6.1 gives a graphical interpretation of the concept of the robust state observer design in the sense of convergent ellipsoidal invariant sets. Let the state space model of the robust observer as

$$\hat{x}(k) = A\hat{x}(k-1) + Bu(k-1) + L(k-1)(y(k-1) - C\hat{x}(k-1)), \tag{6.7}$$

where $\hat{x}(k-1)$ is the state at instant $(k-1)$, namely past state estimation, and $L(k-1)$ is the observer gain at instant $(k-1)$. Consequently, the estimation error at instant k is

$$\begin{aligned} e(k) \;&=\; Ax(k-1) + B_\theta\theta(k-1) + B_d d(k-1) - A\hat{x}(k-1) \\ &\quad + L(k-1)(C\hat{x}(k-1) - y(k-1)). \end{aligned} \tag{6.8}$$

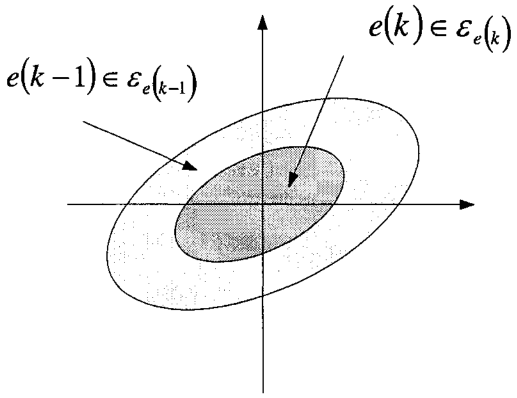

$e(k) \in \varepsilon_{e(k)}$
$e(k-1) \in \varepsilon_{e(k-1)}$

Figure 6.1: Ellipsoidal invariant sets for the robust observer errors at instant $(k-1)$ and k

Define a stacked vector v to facilitate the robust observer design [55] with

$$v := \begin{bmatrix} x^T(k-1) & \theta^T(k-1) & d^T(k-1) & 1 \end{bmatrix}^T.$$

From (6.3), (6.3), and (6.8), express the observer errors $e(k-1)$ and $e(k)$, the bounded disturbances $d(k-1)$ and $d(k)$, the auxiliary variables $\theta(k)$ and $\xi(k)$, and the output $y(k)$ in terms of v,

$$\begin{aligned}
e(k-1) &= \begin{bmatrix} I & 0 & 0 & -\hat{x}(k-1) \end{bmatrix} v = \Gamma_{e(k-1)}v, && (6.9)\\
e(k) &= \begin{bmatrix} A & B_\theta & B_d & -A\hat{x}(k-1) & + L(k-1) \cdot \end{bmatrix.\\
&\quad \left. (C\hat{x}(k-1) - y(k-1)) \right] v = \Gamma_{e(k)}v, && (6.10)\\
\theta(k-1) &= \begin{bmatrix} 0 & I & 0 & 0 \end{bmatrix} v = \Gamma_\theta v, && (6.11)\\
\xi(k-1) &= \begin{bmatrix} T_1 & 0 & 0 & T_2u(k-1) \end{bmatrix} v = \Gamma_\xi v, && (6.12)\\
d(k-1) &= \begin{bmatrix} 0 & 0 & I & 0 \end{bmatrix} v = \Gamma_{d(k-1)}v, && (6.13)\\
d(k) &= \left[-\Phi_1CA \quad -\Phi_1CB_\theta \quad -\Phi_1CB_d \right.\\
&\quad \left. \Phi_1y(k) - \Phi_1CBu(k-1) \right] v = \Gamma_{d(k)}v, && (6.14)\\
1 &= \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} v = \Gamma_1 v. && (6.15)
\end{aligned}$$

Note that all transformation matrices are known in (6.9) - (6.15), except the matrix $\Gamma_{e(k)}$ which is the unknown function of $L(k-1)$. Given the condition in (6.6), we have

$$\left(\Gamma_{e(k)}v\right)^T P_k\Gamma_{e(k)}v < \left(\Gamma_1 v\right)^T \Gamma_1 v, \tag{6.16}$$

if given

$$\begin{aligned}
(\Gamma_{e(k-1)}v)^T P_{k-1}\Gamma_{e(k-1)}v &< (\Gamma_1 v)^T\Gamma_1 v \quad \text{(from (6.5))}, && (6.17)\\
(\Gamma_{d(k-1)}v)^T W_d\Gamma_{d(k-1)}v &< 1 \qquad\qquad \text{(from (6.2))}, && (6.18)\\
\left(\Gamma_d v\right)^T W_d\Gamma_d v &< 1 \qquad\qquad \text{(from (6.2))}, && (6.19)\\
\left(\Gamma_\theta v\right)^T \Gamma_\theta v &< \left(\Gamma_\xi v\right)^T \Gamma_\xi v \quad \text{(from (6.3))}. && (6.20)
\end{aligned}$$

Implementing the S-procedure, the condition in (6.16) does hold if

$$\begin{aligned}
&\Phi_2 - \Gamma_{e(k)}^T P_k\Gamma_{e(k)}\\
> \; &\gamma_1(\Phi_2 - \Gamma_{e(k-1)}^T P_{k-1}\Gamma_{e(k-1)}) + \gamma_2(\Phi_2 - \Gamma_{d(k-1)}^T W_d\Gamma_{d(k-1)})\\
&\gamma_3(\Phi_2 - \Gamma_d^T W_d\Gamma_d) + \gamma_4(\Gamma_\xi^T\Gamma_\xi - \Gamma_\theta^T\Gamma_\theta),
\end{aligned} \tag{6.21}$$

where $\Phi_2 = \Gamma_1^T\Gamma_1$, and γ_1, γ_2, γ_3, and γ_4 are all auxiliary positive scalars. P_k and P_{k-1} are both positive-definite matrices where P_{k-1} is known with respect to the ellipsoid $\mathcal{E}_{e(k-1)}$. Notice that in (6.21) both variables P_k and $\Gamma_{e(k)}$ are unknown. So (6.21) is not a linear matrix inequality. Setting $M = P_k^{1/2}$ and $N = P_k^{1/2}L\,(k-1)$, it denotes

$$P_k^{1/2}\Gamma_{e(k)} = [\; MA \quad MB_\theta \quad MB_d \quad -MA\hat{x}\,(k-1) + N(Cx\,(k-1) - y\,(k-1)) \;]. \tag{6.22}$$

By Schur complements, (6.21) can be converted into

$$\begin{bmatrix} \Pi_{11} & \Pi_{21}^T \\ \Pi_{21} & I \end{bmatrix} > 0, \tag{6.23}$$

where

$$\begin{aligned}
\Pi_{11} :=\; & \Phi_2 - \gamma_1\left(\Phi_2 - \Gamma_{e(k-1)}^T P_{k-1}\Gamma_{e(k-1)}\right) - \gamma_2\left(\Phi_2 - \Gamma_{d(k-1)}^T W_d\Gamma_{d(k-1)}\right)\\
& -\gamma_3\left(\Phi_2 - \Gamma_d^T W_d\Gamma_d\right) - \gamma_4\left(\Gamma_\xi^T\Gamma_\xi - \Gamma_\theta^T\Gamma_\theta\right),\\
\Pi_{21} :=\; & P_k^{1/2}\Gamma_{e(k)} = \left[\; MA \quad MB_\theta \quad MB_d \right.\\
& \left. -MA\hat{x}\,(k-1) + N\left(Cx\,(k-1) - y\,(k-1)\right)\right].
\end{aligned} \tag{6.24}$$

As we know that increasing the diameter of an ellipsoid $\mathcal{E}_{e(k)}$, namely the longest axis of the ellipsoid, results in increasing its volume. Meanwhile the diameter is proportional to the maximal eigenvalue of P_k^{-1}, $\bar{\lambda}(P_k^{-1})$ [12]. Therefore, minimizing $\bar{\lambda}(P_k^{-1})$, namely maximizing the minimal eigenvalue of $P_k^{1/2}$ (denoted by $\underline{\lambda}(P_k^{1/2})$), the structure matrix P_k defines the smallest ellipsoid containing the estimation errors $e\,(k)$. This idea leads to a minimizing eigenvalue problem (EVP) (refer to Section 3.2.1).

Theorem 6.1 *In the presence of model uncertainties $\Delta_\theta\,(k)$ and bounded unmeasured disturbance $d\,(k)$, the state estimation of system (6.1) can be obtained by solving an SDP problem,*

$$\begin{aligned} & \min_{L(k-1),\ \lambda,\ M,\ N,\ \gamma_i} (-\lambda)\ \ (i = 1,\ 2,\ 3,\ 4), && (6.25)\\ s.\ t. \quad & \begin{bmatrix} \Pi_{11} & \Pi_{21}^T \\ \Pi_{21} & I \end{bmatrix} > 0, \\ & \lambda - \bar{\lambda}(P_{k-1}^{1/2}) > 0, \\ & M - \lambda I > 0, \gamma_4 > 0, \quad M > 0, \\ & \gamma_1 > 0, \quad \gamma_2 > 0, \ \gamma_3 > 0, \end{aligned}$$

where the symbols M, N are defined in (6.22), Π_{11}, Π_{21} in (6.24), and γ_1, γ_2, γ_3, γ_4 are unknown scalars. $\bar{\lambda}(P_{k-1}^{1/2})$ stands for the maximal eigenvalue of the square root of P_{k-1}, the structure matrix for the past estimation error ellipsoid $\mathcal{E}_{e(k-1)}$. The the time-varying state observer gain at instant $(k-1)$, namely $L\,(k-1)$, can be calculated by

$$L\,(k-1) = M^{-1}N,$$

and the estimated state at instant k satisfies

$$\hat{x}\,(k) = A\hat{x}\,(k-1) + Bu\,(k-1) + L\,(k-1)\,(y\,(k-1) - C\hat{x}\,(k-1))\,.$$

Proof: Following immediately from (6.23), we can compute the feasible solution of the structure matrix P_k by minimizing the volume of ellipsoidal set $\mathcal{E}_{e(k)}$, equivalently maximize the determinant of P_k. It derives a standard MAXDET optimization problem of P_k. To reduce the computation complexity, define the minimal eigenvalue of $(P_k)^{1/2}$as λ. We know that

$$\lambda \propto \mathbf{det}^{-1}(P_k^{-1}) \text{ (equivalently } \lambda \propto \mathbf{Vol}^{-1}(\mathcal{E}_{e(k)})),$$

where symbol "$\propto$" and notation **Vol**$(*)$ stand for "*Proportional to*" and "*Volume*", respectively. Therefore, the optimal P_k can be solved by,

$$\begin{aligned} & \min\ (-\lambda) \\ \text{s. t.} \quad & M - \lambda I > 0,\ \lambda > 0,\ (M = P_k^{1/2}), \\ & \text{Equation (6.23).} \end{aligned} \tag{6.26}$$

The constraint $\lambda - \bar{\lambda}(P_{k-1}^{1/2}) > 0$ is designed to guarantee observer's stability in the sense of Lyapunov. P_{k-1} is the structure matrix of the ellipsoid $\mathcal{E}_{e(k-1)}$ containing the past observer errors. As we know, the diameter of $\mathcal{E}_{e(k)}$ is proportional to $\bar{\lambda}\left(P_k^{-1}\right)$, similar to $\underline{\lambda}\left(P_k\right)$. Therefore if $\underline{\lambda}(P_k) > \bar{\lambda}(P_{k-1})$, equivalently $\lambda > \bar{\lambda}(P_{k-1}^{1/2})$, $\mathcal{E}_{e(k)}$ will always be contained in $\mathcal{E}_{e(k-1)}$, and consequently, the observer is stable. Add the constraint $\lambda - \bar{\lambda}(P_{k-1}^{1/2}) > 0$ to (6.26), and Theorem 6.1 is proven. ■

Remark 6.1 *Theorem 6.1 derives a dynamic observer gain $L(k-1)$, which is a time-varying nonlinear function of the past input $u(k-1)$, the past output $y(k-1)$, the past estimation $\hat{x}(k-1)$, and the current output $y(k)$.*

Remark 6.2 *By setting the constraint $\lambda - \bar{\lambda}(P_{k-1}^{1/2}) > 0$, we can guarantee the stability of the robust state observer in the sense of Lyapunov. To control the convergent rate, we construct a tuning scalar $\alpha > 0$ with*

$$\lambda > \alpha \underline{\lambda}(P_{k-1}^{1/2})$$

to replace the constraint $\lambda - \bar{\lambda}(P_{k-1}^{1/2}) > 0$ in problem (6.25). The larger α is chosen, the faster convergent trajectory derives.

Remark 6.3 *Because the volume of the ellipsoidal set $\mathcal{E}_{e(k)}$ is proportional to the determinant of P_k^{-1}, minimizing the volume of $\mathcal{E}_{e(k)}$ is equivalent to maximizing the determinant of P_k. Thus, robust observer design is possibly converted into an MAXDET optimization problem [107]. Using the existing solver YAPLMI [57], MAXDET can be solved numerically. However, compared with the EVP problem in (6.25), YAPLIM needs more time to get a feasible solutions to the MAXDET programming for robust observer design.*

6.1.2 A numerical example

Consider a system given by

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 0.5 & 0.7 \\ 0.1 & 0.6+\alpha(k) \end{bmatrix} x(k) + \begin{bmatrix} 0.1 \\ 0.2+\beta(k) \end{bmatrix} u(k) + \begin{bmatrix} 0.1 \\ 0 \end{bmatrix} d(k), \\ y(k) &= \begin{bmatrix} 2 & 1 \end{bmatrix} x(k) + 0.1\, d(k), \end{aligned} \tag{6.27}$$

where $\alpha(k) \in [-0.05,\ 0.05]$, $\beta(k) \in [-0.04,\ 0.04]$, and $d(k) \in [-1,\ 1]$. Suppose that the initial conditions $x(0) = [1,\ 0]^T$ and $\hat{x}(0) = [0,\ 0]^T$. Reforming the model in (6.27) into the structure of (6.1), we have

$$T_1 = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix},\ T_2 = \begin{bmatrix} 0 \\ 0.04 \end{bmatrix},\ B_\theta = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix},\ \text{and } B_d = \begin{bmatrix} 0.1 \\ 0 \end{bmatrix}.$$

Set $W_d = 0.9$ for the disturbance invariant set and $P_0 = 0.1I$ relative to the ellipsoid $\mathcal{E}_{e(0)}$ for the initial observe error $e(0)$. Figs. 6.2 and 6.3 show the trajectories of the estimated states with $u(k) \equiv 1$. The estimation and real states match each other well. Fig. 6.4 is the trajectories of the observer errors which are bounded within a neighborhood around the origin. Here the uncertain terms $\alpha(k)$, $\beta(k)$, and unknown disturbance $d(k)$ are simulated by MATLAB function "*rand.*" Fig. 6.5 illustrates the development of ellipsoidal invariant sets for observer errors (only several ellipsoids are presented for a clearer figure).

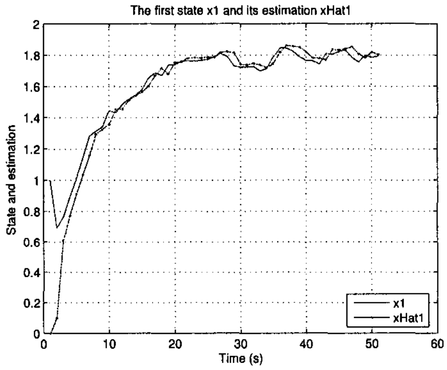


Figure 6.2: The trajectories of x_1 and $\hat{x}_1$.

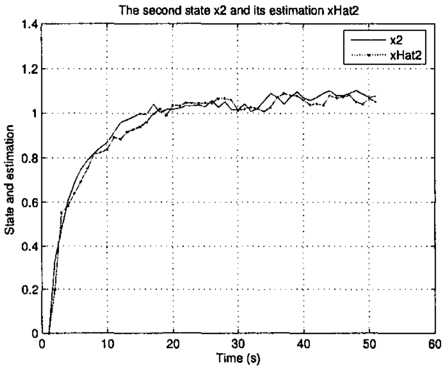


Figure 6.3: The trajectories of x_2 and $\hat{x}_2$.

Comparing with the estimation resulted by YAPLIM in Figs. 6.6 - 6.7, our results are much better (Figs. 6.4 - 6.5).

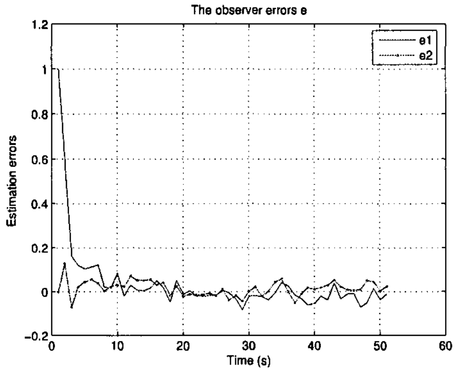


Figure 6.4: The trajectories of the estimation errors.

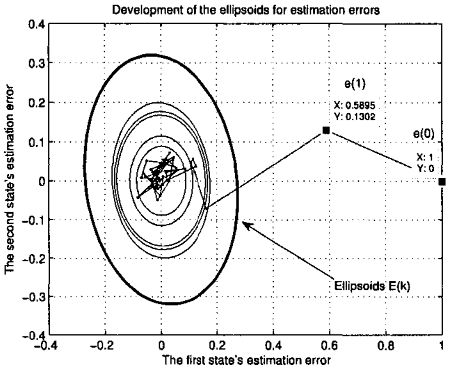


Figure 6.5: Development of the ellipsoid sets for the observer errors, $\mathcal{E}(k)$ ($k = 3, 5, 10, 15, 35, 50$)

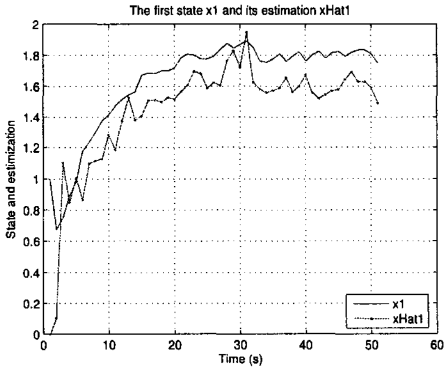


Figure 6.6: The trajectories of x_1 and $\hat{x}_1$ by the YAPLMI.

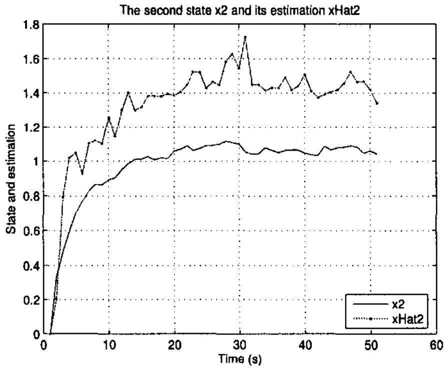


Figure 6.7: The trajectories of x_2 and $\hat{x}_2$ by the YAPLMI.

6.2 Moving horizon state observer

Although the above algorithm has the ideal properties of stability and convergence, it does not combine the practical issues with theoretical analysis, e.g., system physical constraints, computational complexity, and implementation efficiency. To obtain a new observation method with a wider scope of applications, moving horizon state observer (MHSO) was proposed by reformulating the design as an optimization problem [41, 87].

MHSO is motivated by the full information state observer which, however, suffers from the curse of increasing dimension [27]. Different from full information state observer, MHSO includes only the most recent measurement and defines the problem within a fixed prediction horizon, so that the dimension of problems is fixed and determined by the length of the horizon. This idea originates from FH-MPC [80, 67, 63] and utilizes a similar scenario. Similar as FH-MPC, an iterative loop of MHSO is composed of four steps: determining initial parameters, predicting future states, solving an optimal problem, and then updating state observation [64, 78, 79]. Because of the potential to handle state constraints, MHSO witnessed wide applications to different physical systems in the past decade. For example, the state observer of biomass concentration in CHO animal cell cultures is successfully developed in the fashion of MHSO [36, 77]. Arrival cost, as one of the fundamental concepts, is proposed to summarize the effort of the past data ahead of current prediction horizon [87]. It can be shown that if we can compute the explicit solution to arrival cost, stability can be easily guaranteed by solving an algebraic Riccati equation. For an instance, the Kalman filter, as a special case of MHSO with free state constraints and unit prediction horizon, can achieve stability in this way. For general cases, however, computing an explicit solution still remains an open problem [38]. In 2001, Rao *et al.* proposed a sufficient condition for the stability of MHSO employing the approximation of arrival cost [78]; but one assumption was critical: the system must have a precise model. To remove such a limitation, in this thesis we propose an extended MHSO, namely robust MHSO (RMHSO) for the systems with both internal uncertainties and external disturbances. Also, the importance of RMHSO can be seen from another point of view: RMHSO is critical for explicit MPC systems whose states are unmeasured or partially unavailable (refer to Chapter 5). Because of the nature of offline MPC, it is mandatory to combine state physical constraints with robust observer formulation. Otherwise there is no way to implement offline controllers with bounded state space partitions [6, 4, 18]. To preserve the superiority of offline MPC (for instance, small implementation cost), the associated observer should be offline optimized and online implemented as well. The aim of this section is to develop an offline MHSO algorithm in the presence of internal uncertainties and external disturbances.

6.3 Robust moving horizon state observer

Consider a system modelled by

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + f(x(k), d(k), k), \\ y(k) &= Cx(k) + v(k). \end{aligned} \tag{6.28}$$

Here $x(k) \in \mathbb{R}^n$ stands for the state, $u(k) \in \mathbb{R}^m$ the input, $y(k) \in \mathbb{R}^q$ the output, $v(k) \in \mathbb{R}^q$ the measurement error, and $d(k) \in \mathbb{R}^l$ a combination of input and state disturbances. To simplify design, we firstly assume $v(k) \equiv 0$ and postpone the discussion on the case of $v(k) \neq 0$ to Section 6.6.3. A and B are constant matrices with appropriate dimensions. We assume that the output matrix C has a full row-rank and the pair $(C,\ A)$ is observable. Suppose that the states and disturbances obey the conditions

$$x(k) \in \mathcal{A}_x,\ v(k) \in \mathcal{A}_v,\ \text{and}\ d(k) \in \mathcal{A}_d, \tag{6.29}$$

where $\mathcal{A}_x(\mathcal{A}_v)$ is the admissible state (noise) set defined by a set of generalized element-wise inequalities, and $\mathcal{A}_d$ is the admissible disturbance set defined by an ellipsoidal invariant set,

$$\mathcal{A}_x := \{x \in \mathbb{R}^n |\ x_{\min} \preceq x \preceq x_{\max},\ x_{\min},\ x_{\max} \in \mathbb{R}^n\}, \tag{6.30}$$

$$\mathcal{A}_v := \{v \in \mathbb{R}^q |\ v_{\min} \preceq v \preceq v_{\max},\ v_{\min},\ v_{\max} \in \mathbb{R}^q\}, \tag{6.31}$$

$$\mathcal{A}_d := \{d \in \mathbb{R}^p \mid d^T(k) W_d d(k) \leq 1, W_d \in \mathbb{S}^l_+\}. \tag{6.32}$$

The nonlinear term $f : \mathcal{A}_x \times \mathcal{A}_d \times \mathbb{R}^+ \to \mathbb{R}^n$ reflects the composition of internal and external uncertainties satisfying

$$||f(x(k), d(k), k)||_2 \leq \kappa\ (\kappa \geq 0). \tag{6.33}$$

In fact, many structured internal and external uncertainties can be reformulated into the form of (6.33).

Case I — external uncertainties: The function $f(k)$ is explicitly expressed by

$$f(x(k), d(k), k) = B_d d(k),$$

where $d(k) \in \mathcal{A}_d$ and B_d is the a constant matrix. From the definition in (6.30), one has

$$||B_d d(k)||_2 = ||B_d W_d^{-1/2} W_d^{1/2} d(k)||_2 \leq \bar{\sigma}(B_d W_d^{-1/2}),$$

which is in the form of (6.33).

Case II — internal uncertainties: The widely used structured uncertainties in the feedback loop [45] can be also converted into (6.33). Consider the system

$$x(k+1) = (A + W_L \Delta(k) W_R) x(k), \tag{6.34}$$

where $\Delta(k) = \text{diag}(\Delta_1(k), \cdots, \Delta_r(k))$ and $\bar{\sigma}(\Delta_i(k)) \leq \alpha_i$. W_L and W_R are constant scaling matrices and W_L is invertible. Performing the similarity transformation to (6.34) and setting $x(k) = W_L\ z(k)$, we have

$$z(k+1) = W_L^{-1} A W_L z(k) + \Delta(k) W_R W_L z(k).$$

Because $x(k) \in \mathcal{A}_x$, there exists a constant term ξ such that $||z(k)||_2 \leq \xi$. Denote $f(x(k), d(k), k) := \Delta(k) W_R W_L z(k)$, and consequently

$$||f(x(k), d(k), k)||_2 \leq (\max_i \alpha_i) \bar{\sigma}(W_R W_L) \xi,$$

which is also in the form of (6.33).

To proceed the further discussion, we first assume $v(k) \equiv 0$ and focus on the system

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + f(x(k), d(k), k), \\ y(k) &= Cx(k), \end{aligned} \tag{6.35}$$

where $x(k) \in \mathcal{A}_x$ and $||f(\cdot)||_2 \leq \kappa$, to design a robust moving horizon state observer.

6.3.1 Formulation of RMHSO

Based on the state space model in (6.35), the observer is defined as

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + Bu(k) + \hat{f}(k), \\ \hat{y}(k) &= C\hat{x}(k), \end{aligned} \tag{6.36}$$

where $\hat{x}(k) \in \mathbb{R}^n$ is the estimated state, $\hat{y}(k) \in \mathbb{R}^q$ the estimated output, and $\hat{f}(k) \in \mathbb{R}^n$ the estimated disturbance. Given the model in (6.36) and past estimated state $\hat{x}(k-N)$, we predict the intermediate observation $\hat{x}(k-N+i)$,

$$\hat{x}(k-N+i) = A^i\hat{x}(k-N)+\sum_{j=0}^{i-1} A^{i-1-j}Bu(k-N+j)+\sum_{j=0}^{i-1} A^{i-1-j}\hat{f}(k-N+j), \quad (6.37)$$

where $i \in [0,N]$ is the index of estimated signals. The sequence $\hat{x}_{k-N\to k}$ is the observation components over the kth prediction horizon. $\hat{x}(k-N)$ is the initial condition of the kth prediction horizon and is optimized by the $(k-N)$th prediction horizon. Obviously, if we can optimize the estimated sequence $\hat{f}^o_{k-N\to k-1}$, the current estimated state $\hat{x}(k)$ can to be solved by

$$\hat{x}(k) = A^N\hat{x}(k-N) + \sum_{j=0}^{N-1} A^jBu(k-1-j) + \sum_{j=0}^{N-1} A^j\hat{f}(k-1-j). \quad (6.38)$$

Retain the value of $\hat{x}(k)$, reject intermediate observation $\hat{x}_{k-N\to k-1}$, and repeat the above procedure. Finally, we can obtain the state observation at any instant k. Eq. (6.38) shows the essential difference between MHSO and MPC regulation. MPC predicts future states/outputs based on current measurements. After determining the optimal input sequence $u_{k\to k+N-1}$, it retains the first element $u(k)$ and rejects the intermediate solutions, including predicted states and the rest of optimal inputs. But MHSO employs a different policy: it is based on the past observation $\hat{x}(k-N)$ to predict intermediate observation till the current observation $\hat{x}(k)$. After determining the optimal sequence $\hat{f}_{k-N\to k-1}$, it calculates $\hat{x}^o(k)$ and then rejects all of the intermediary variables $\hat{x}_{k-N\to k-1}$. Essentially, MPC performs the optimization loop in a forward manner, but MHSO does it in the backward way which, fortunately, consistent with the nature of closed-loop prediction [50]. The rule in (6.38) makes it straightforward to convert MHSO design into an optimization problem with recursive closed-loop prediction.

Definition 6.1 *The design of robust MHSO for the system with internal and ex-*

ternal uncertainties in (6.35) is a constrained optimization problem,

$$\min_{\hat{f}_{k-N\to k-1}} \quad J_{k-N\to k} \tag{6.39}$$

$$\begin{aligned} s.t. \quad & J_{k-N\to k} = \|C\hat{x}(k) - y(k)\|^2_{Q_0} \\ & + \sum_{j=k-N}^{k-1} \|C\hat{x}(j) - y(j)\|^2_Q + \left\|\hat{f}(j)\right\|^2_R, \end{aligned} \tag{6.40}$$

$$\begin{aligned} & \hat{x}(k-N+i) = A^i\hat{x}(k-N) + \sum_{j=0}^{i-1} A^{i-1-j}Bu(k-N+j) \\ & \qquad + \sum_{j=0}^{i-1} A^{i-1-j}\hat{f}(k-N+j) \quad (1 \le i \le N), \\ & \hat{f}(k-1) = L(C\hat{x}(k-1) - y(k-1)), \quad \hat{x}(k-N+i) \in \mathcal{A}_x, \end{aligned}$$

where $Q \in \mathbb{S}^q_+$ and $R \in \mathbb{D}^n_{++}$ are weightings. $Q_0 \in \mathbb{S}^q$ and L are the arrival weighing and the arrival observer gain, respectively, which are constructed for robust observer stability. Pair (C, A) is observable.

From (6.35) and (6.36), we can write down the model of observation errors,

$$e(k+1) = Ae(k) + \hat{f}(k) - f(x(k), d(k), k), \tag{6.41}$$

where $e(k) := \hat{x}(k) - x(k)$. Therefore, the robust stability of state observers is converted into a problem on the convergence of $e(k)$ in the presence of the uncertain term $f(\cdot)$ in (6.41).

Definition 6.2 *The observer in (6.36) is stable for the system with internal and external uncertainties in (6.35), if for any $\bar{\varepsilon} > 0$ there exists a number $\bar{\delta} > 0$ and a positive integer T such that if $||e(0)|| \le \bar{\delta}$ and $\hat{x}(0) \in \mathcal{A}_x$, then $||e(k)|| \le \bar{\varepsilon}$ and $\hat{x}(k) \in \mathcal{A}_x$ for all $k \ge T$. The admissible state set $\mathcal{A}_x$ and observation error dynamics are given in (D1.1) and (6.41), respectively.*

6.3.2 Robust observation stability

To guarantee the stability of the robust observer in (6.41), we employ the objective function (6.40) as a Lyapunov candidacy function, so that we have the Lyapunov functions $V(k) := J_{k-N\to k}$ and $V(k+1) := J_{k-N+1\to k+1}$. From (6.40), the difference of the Lyapunov functions is given by

$$\begin{aligned} \tilde{V} &= V(k+1) - V(k) \\ &= ||e(k+1)||^2_{\hat{Q}_0} + ||e(k)||^2_{\hat{Q}} + ||\hat{f}(k)||^2_R \\ &\quad -||e(k)||^2_{\hat{Q}_0} - ||e(k-N)||^2_{\hat{Q}} - ||\hat{f}(k-N)||^2_R, \end{aligned} \tag{6.42}$$

where $\hat{Q}_0 = C^T Q_0 C$ and $\hat{Q} = C^T Q C$. In Definition 6.1, we propose the arrival observer gain L, satisfying

$$\hat{f}(k) = L(C\hat{x}(k) - y(k)) = LCe(k). \tag{6.43}$$

Inserting (6.41) and (6.43) into the difference of the Lyapunov functions in (6.42), we have

$$\tilde{V} = ||e(k)||^2_{Q_{tot}} + ||f(\cdot)||^2_{\hat{Q}_0} - 2e(k)^T(A+LC)^T\hat{Q}_0 f(\cdot) - ||e(k-N)||^2_{\hat{Q}} - ||\hat{f}(k-N)||^2_R,$$

where

$$Q_{tot} := (A+LC)^T\hat{Q}_0(A+LC) + \hat{Q} + (LC)^T R(LC) - \hat{Q}_0.$$

To guarantee stability, we need $\tilde{V} \leq 0$, i.e.,

$$\begin{aligned} &||e(k)||^2_{Q_{tot}} + ||f(\cdot)||^2_{\hat{Q}_0} - 2e(k)^T(A+LC)^T\hat{Q}_0 f(\cdot) - \\ &||e(k-N)||^2_{\hat{Q}} - ||\hat{f}(k-N)||^2_R, + \nu||e(k)||^2_P - \nu||e(k)||^2_P \leq 0, \end{aligned} \tag{6.44}$$

where $\nu > 0$ and $P \in \mathbb{S}^n_{++}$ are the tuning parameters and critical to the robustness of RMHSO. We have a pair of sufficient conditions to (6.44)

$$(A+LC)^T\hat{Q}_0(A+LC) + \hat{Q} + (LC)^T R(LC) - \hat{Q}_0 + \nu P = 0, \tag{6.45}$$

$$||f(\cdot)||^2_{\hat{Q}_0} - 2e(k)^T(A+LC)^T\hat{Q}_0 f(\cdot) - ||\hat{f}(k-N)||^2_R - \nu||e(k)||^2_P \leq 0. \tag{6.46}$$

Note that $||e(k-N)||^2_{\hat{Q}}$ is the initial observation error which is positive and omitted here. Apparently, (6.45) is an algebraic Riccati equation with the unknown variables of the arrival observer gain L and the transformed arrival weighting $\hat{Q}_0$. It is not hard to derive Q_0 based on the solution of (6.45),

$$Q_0 = (CC^T)^{-1}C\hat{Q}_0 C^T(CC^T)^{-1}, \; \hat{Q}_0 \in \mathbb{S}^n_+. \tag{6.47}$$

Note that we assume that C has full row-rank, so the pseudo-inverses $(CC^T)^{-1}C$ and $C^T(CC^T)^{-1}$ exist. It can be seen that no matter what tuning parameters ν and

P are chosen, we always can derive L and Q_0 from (6.45) and (6.47). Consequently, the feasibility of (6.46) plays a critical role on robust stability analysis.

Theorem 6.2 *The observer in (6.36) is robust stable for the constrained system with internal and external uncertainties in (6.35) if the arrival weighting Q_0 and the arrival observer error L are determined by the Riccati equation in (6.45) and the estimated disturbance $\hat{f}(k-N)$ is solved by minimizing the following linear program,*

$$\begin{aligned} & \min \ \varepsilon, \\ s.t. \quad & \begin{bmatrix} \nu P & (A+LC)^T \\ (A+LC) & \varepsilon \hat{Q}_0 \end{bmatrix} \geq 0, \end{aligned}$$

and satisfies

$$|row(R^{1/2})\hat{f}(k-N)| \geq \frac{1}{n}(1+\varepsilon)^{1/2}\bar{\sigma}(\hat{Q}_0)\kappa,$$

where $row(R^{1/2}) := [R_{11,}^{1/2} \cdots , R_{nn}^{1/2}]$ is a row vector composed of the diagonal elements of R, and κ is the uncertainty bound defined in (6.33).

Proof: Following the conditions in (6.45) and (6.46) and applying Lemma 3.3, we have

$$\begin{aligned} & -2e(k)^T(A+LC)^T\hat{Q}_0 f(\cdot) \\ \leq \ & \varepsilon f^T(\cdot)\hat{Q}_0 f(\cdot) + \frac{1}{\varepsilon}e(k)^T(A+LC)^T\hat{Q}_0(A+LC)e(k) \qquad (\varepsilon > 0). \end{aligned}$$

Therefore (6.46) is necessary to

$$\begin{aligned} & \frac{1}{\varepsilon}e(k)^T(A+LC)^T\hat{Q}_0(A+LC)e(k) - \nu||e(k)||_P^2 \\ & +(1+\varepsilon)f^T(\cdot)\hat{Q}_0 f(\cdot) - ||\hat{f}(k-N)||_R^2 \leq 0. \end{aligned} \tag{6.48}$$

So if the conditions

$$\nu P - \frac{1}{\varepsilon}(A+LC)^T\hat{Q}_0(A+LC) \geq 0, \tag{6.49}$$

$$||\hat{f}(k-N)||_R^2 \geq (1+\varepsilon)f^T(\cdot)\hat{Q}_0 f(\cdot), \tag{6.50}$$

are satisfied simultaneously, the condition in (6.48) is obtained. To minimize $||\hat{f}(k-N)||_R^2$, we minimize the positive scalar ε. Consequently, (6.49) can be recast into an SDP problem. Performing Schur complements, we have

$$\begin{aligned} & \min \ \varepsilon, \\ s.t. \quad & \begin{bmatrix} \nu P & (A+LC)^T \\ (A+LC) & \varepsilon \hat{Q}_0 \end{bmatrix} \geq 0. \end{aligned} \tag{6.51}$$

Eq. (6.50) is equivalent to

$$\sum_{i=1}^{n}(\sqrt{R_{ii}}\hat{f}_i(k-N))^2 \geq (1+\varepsilon)f^T(\cdot)\hat{Q}_0 f(\cdot).$$

As we know the condition

$$\sum_{i=1}^{n}(\sqrt{R_{ii}}\hat{f}_i(k-N))^2 \geq \frac{1}{n}(\text{row}(R)\hat{f}\,(k-N))^2,$$

so that a sufficient condition to (6.50) is

$$|\text{row}(R^{\frac{1}{2}})\hat{f}\,(k-N)\,| \geq \frac{1}{n}(1+\varepsilon)^{\frac{1}{2}}\bar{\sigma}(\hat{Q}_0)\kappa. \tag{6.52}$$

Note $R \in \mathbb{D}^n_{++}$ and $\hat{f}(k-N) \in \mathbb{R}^n$. Theorem 6.2 is then proven. ■

Remark 6.4 *The feasibility of the semi-definition optimization problem in (6.51) is strongly related to the selection of the tuning parameters ν and P. Roughly speaking, if we choose an appropriate pair of ν and P (large enough), the robust stability can be always satisfied.*

Remark 6.5 *After determining the values of $\hat{f}_i\,(k-N)$, Q_0, and L, we can calculate the upper bound of κ to satisfy both conditions (6.49) and (6.50). The upper bound of κ reflects the robustness of our algorithm, i.e., by adjusting the values of ν and P, we can adjust the tradeoff between the performance and stability of our robust observers. This fact is similar as Corollary 5.1*

6.4 RMHSO using open-loop prediction

From the above discussion, we know that RMHSO design can be converted into a quadratic program, and associated with Theorem 6.2, the robust stability is guaranteed. From Chapter 4, we know that nominal MPC can be reformulated as an mp-QP regulation. The solution to mp-QP is a set of piece-wise affine functions associated with state space partitions. In this section, we first employ the idea of nominal explicit MPC to develop an open-loop RMHSO, and then extend the design to an RMHSO with the recursive closed-loop prediction in the next section. From (6.37), we can predict the N step coming observations $x_{k-N+1\to k}$, so that the objective $J_{k-N\to k}$ can be rewritten as an expression of $\hat{x}(k-N)$,

$$J_{k-N\to k} = ||C\hat{x}(k-N)-y\,(k-N)\,||_Q^2+||\mathcal{C}\mathcal{A}\hat{x}(k-N)+\mathcal{C}\mathcal{B}U+\mathcal{C}\mathcal{B}_F F-Y||_{\mathcal{Q}}^2+||F||_{\mathcal{R}}^2, \tag{6.53}$$

where the augmented matrices are given by

$$\begin{aligned}
\mathcal{A} &= \begin{bmatrix} A \\ \vdots \\ A^N \end{bmatrix},\ \mathcal{B} = \begin{bmatrix} B & \cdots & 0 \\ \vdots & \ddots & \vdots \\ A^{N-1}B & \cdots & B \end{bmatrix},\ \mathcal{B}_F = \begin{bmatrix} I & \cdots & 0 \\ \vdots & \ddots & \vdots \\ A^{N-1} & \cdots & I \end{bmatrix}, \\
U &= [u(k-N)^T, \cdots, u(k-1)^T]^T,\ F = [\hat{f}(k-N)^T, \cdots, \hat{f}(k-1)^T]^T, \\
Y &= [Y(k-N+1)^T, \cdots, Y(k-1)^T,\ Y(k)]^T, \\
\mathcal{C} &= \text{diag}(C, \cdots, C),\ \mathcal{Q} = \text{diag}(Q, \cdots, Q,\ Q_0),\ \mathcal{R} = \text{diag}(R, \cdots, R).
\end{aligned} \tag{6.54}$$

Proceeding further, (6.53) becomes a standard mp-QP problem,

$$J_{k-N\to k} = \frac{1}{2}F^T\Theta F + \hat{x}^T\,(k-N)\,\Xi F + (\mathcal{C}\mathcal{B}U - Y)^T\Pi F + \mathcal{W}, \tag{6.55}$$

where

$$\Theta = 2((\mathcal{C}\mathcal{B}_F)^T\mathcal{Q}(\mathcal{C}\mathcal{B}_F) + \mathcal{R}),\ \Xi = (\mathcal{C}\mathcal{A})^T\mathcal{Q}(\mathcal{C}\mathcal{B}_F),\ \text{and } \Pi = \mathcal{Q}\mathcal{C}\mathcal{B}_F.$$

$\mathcal{W}$ is the residue term independent of F and is determined by the variables in (6.54). Notice that $\mathcal{Q}$ is the matrix of the arrival weighting Q_0 which is fundamental to closed-loop stability, and $\Theta \in \mathbb{S}_{++}$.

Theorem 6.3 *The optimal estimated disturbance vector F in (6.54) is determined by an mp-QP problem with element-wise inequality and equality constraints.*

Proof: Employing the notation in (6.54), the constraint $\hat{x}_{k-N+1\to k} \in \mathcal{A}_x$ can be explicitly expressed by

$$\mathcal{X}_{\min} \preceq \mathcal{A}\hat{x}(k-N) + \mathcal{B}U + F \preceq \mathcal{X}_{\max}, \tag{6.56}$$

where $\mathcal{X}_{\min} := [x_{\min}^T, \cdots, x_{\min}^T]^T$ and $\mathcal{X}_{\max} := [x_{\max}^T, \cdots, x_{\max}^T]^T$. For closed-loop stability, the condition in (6.52) has to be satisfied,

$$\Psi F \preceq -(n+n\varepsilon)^{1/2}\bar{\sigma}(\hat{Q}_0)\kappa, \tag{6.57}$$

where $\Psi := (-1)^{\alpha}[\text{row}(R^{\frac{1}{2}}),\ 0,\ \cdots,\ 0]$ ($\alpha = 0$ or 1). The arrival observer gain L is determined by an algebraic Riccati equation, so

$$\begin{aligned}
\hat{f}(k-1) &= LC\hat{x}(k-1) - y(k-1) \\
&= LCA^{N-1}\hat{x}(k-N) + LCA^{N-2}Bu(k-1) + \cdots + LCBu(k-2) \\
&\quad + LCA^{N-2}\hat{f}(k-N) + LCA^{N-3}\hat{f}(k-N+1) \\
&\quad + \cdots + LC\hat{f}(k-2), \qquad (6.58)
\end{aligned}$$

and equivalently

$$\Gamma_F F = \Gamma_x \hat{x}(k-N) + \Gamma_U U. \qquad (6.59)$$

Combining (6.57) with (6.56), we have an element-wise inequality constraint,

$$G_1 F \preceq G_2 \hat{x}(k-N) + G_3, \qquad (6.60)$$

where

$$G_1 = \begin{bmatrix} I \\ -I \\ \Psi \end{bmatrix},\ G_2 = \begin{bmatrix} -\mathcal{A} \\ \mathcal{A} \\ 0 \end{bmatrix},\ G_3 = \begin{bmatrix} \mathcal{X}_{\max} - \mathcal{B}U \\ -\mathcal{X}_{\min} + \mathcal{B}U \\ -(n+n\varepsilon)^{1/2}\bar{\sigma}(\hat{Q}_0)\kappa \end{bmatrix}.$$

Imposed by constraints (6.57) and (6.60), the design of RMHSO in (6.55) is converted into a constrained mp-QP problem,

$$\begin{aligned}
J^o_{k-N\to k} &= \min_F\left(\frac{1}{2}F^T\Theta F + \hat{x}^T(k-N)\Xi F + (\mathcal{C}\mathcal{B}U - Y)^T\Pi F + \mathcal{W}\right), \\
s.t. &\quad \Gamma_F F = \Gamma_x\hat{x}(k-N) + \Gamma_U U, \\
&\quad G_1 F \preceq G_2\hat{x}(k-N) + G_3. \qquad (6.61)
\end{aligned}$$

Theorem 6.3 is then proven. ■

Theorem 6.4 *The analytic (explicit) solutions to the mp-QP problem in (6.61), which is defined for RMHSO using open-loop prediction, are piece-wise affine functions of $\hat{x}(k-N)$, over the corresponding state critical region $\mathcal{A}_x^j$, where j denotes the jth partition within the admissible state set $\mathcal{A}_x$.*

Proof: Taking advantages of two Lagrange multipliers $\lambda_1 \succeq 0$, $\lambda_2 \succ 0$, and a slack variable μ, (6.61) can be converted into an unconstrained program. From the first-order Karush-Kuhn-Tucker (KKT) theorem, the optimal conditions to (6.61) are known as

$$\begin{aligned} \Theta F + \Xi^T \hat{x}\,(k-N) + \Pi^T(\mathcal{C}\mathcal{B}U - Y) + G_1^T\lambda_1 + \Gamma_F^T\lambda_2 &= 0, && (6.62)\\ (G_1F - G_2\hat{x}\,(k-N) - G_3 + \mu)^T\lambda_1 &= 0, && (6.63)\\ \Gamma_F F - \Gamma_x\hat{x}\,(k-N) - \Gamma_U U &= 0. && (6.64) \end{aligned}$$

From the properties of optimization duality, λ_1 is divided into two parts, namely $\lambda_N = 0$ (nonactive constraints, $\mu > 0$) and $\lambda_A \succ 0$ (active constraints, $\mu = 0$), where $\lambda_1 = [\lambda_N^T,\ \lambda_A^T]^T$. From (6.62) we have

$$-\Theta^{-1}\Xi^T\hat{x}\,(k-N) - \Theta^{-1}\Pi^T(\mathcal{C}\mathcal{B}U - Y) - \Theta^{-1}\tilde{G}_1^T\lambda_A - \Theta^{-1}\Gamma_F^T\lambda_2 = F, \quad (6.65)$$

$$\tilde{G}_1F - \tilde{G}_2x\,(k-N) - \tilde{G}_3 = 0, \quad (6.66)$$

where $\tilde{G}_1$, $\tilde{G}_2$, and $\tilde{G}_3$ is a combination of the active constraints out of G_1, G_2, and G_3, with a full-row rank. Inserting (6.65) into (6.34), we have

$$\lambda_2 = G_{\lambda\lambda}\lambda_A + G_{x\lambda_2}\hat{x}\,(k-N) + G_{\lambda_2}, \quad (6.67)$$

where

$$\begin{aligned} G_{\lambda\lambda} &:= -(\Gamma_F\Theta^{-1}\Gamma_F^T)^{-1}\Gamma_F\Theta^{-1}\tilde{G}_1^T, && (6.68)\\ G_{x\lambda_2} &:= -(\Gamma_F\Theta^{-1}\Gamma_F^T)^{-1}(\Gamma_F\Theta^{-1}\Xi^T + \Gamma_x),\\ G_{\lambda_2} &:= -(\Gamma_F\Theta^{-1}\Gamma_F^T)^{-1}(\Gamma_F\Theta^{-1}\Pi^T(\mathcal{C}\mathcal{B}U - Y) + \Gamma_U U). \end{aligned}$$

Note that $\lambda_2 \succ 0$ corresponds to the equality constraints and Γ_F has full row-rank (refer to (6.59)). Inserting (6.67) and (6.65) into (6.66), we finally derive the explicit solution to λ_A

$$\lambda_A = G_{x\lambda_A}\hat{x}\,(k-N) + G_{\lambda_A}, \quad (6.69)$$

where

$$\begin{aligned} G_{x\lambda_A} &:= -(\tilde{G}_1\Theta^{-1}\tilde{G}_1^T + \tilde{G}_1\Theta^{-1}\Gamma_F^T G_{\lambda\lambda})^{-1}(\tilde{G}_1\Theta^{-1}\Xi^T\\ &\quad +\tilde{G}_1\Gamma_x + \tilde{G}_2 + \tilde{G}_1\Theta^{-1}\Gamma_F^T G_{x\lambda_2}),\\ G_{\lambda_A} &:= -(\tilde{G}_1\Theta^{-1}\tilde{G}_1^T + \tilde{G}_1\Theta^{-1}\Gamma_F^T G_{\lambda\lambda})^{-1}(\tilde{G}_1\Theta^{-1}\Pi^T(\mathcal{C}\mathcal{B}U - Y)\\ &\quad +\tilde{G}_1\Gamma_U U + \tilde{G}_3 + \tilde{G}_1\Theta^{-1}\Gamma_F^T G_{\lambda_2}). \end{aligned}$$

It is obvious that $(\tilde{G}_1\Theta^{-1}\tilde{G}_1^T + \tilde{G}_1\Theta^{-1}\Gamma_F^T G_{\lambda\lambda}) \in \mathbb{S}_{++}$ (replacing $G_{\lambda\lambda}$ by (6.68)). From (6.67) and (6.69), we can conclude that the optimal solution F^o is an affine function of $\hat{x}\,(k-N)$, i.e.,

$$F = G_{xF}\hat{x}\,(k-N) + G_F, \tag{6.70}$$

where

$$\begin{aligned} G_{xF} &:= -\Theta^{-1}\Xi^T - (\Theta^{-1}\tilde{G}_1^T + \Theta^{-1}\Gamma_F^T G_{\lambda\lambda})G_{x\lambda_A} - \Theta^{-1}\Gamma_F^T G_{x\lambda_2}, \\ G_F &:= -\Theta^{-1}\Pi^T(\mathcal{C}\mathcal{B}U - Y) - (\Theta^{-1}\tilde{G}_1^T + \Theta^{-1}\Gamma_F^T G_{\lambda\lambda})G_{\lambda_A} - \Theta^{-1}\Gamma_F^T G_{\lambda_2}. \end{aligned}$$

To guarantee $\lambda_1 \succeq 0$ and satisfy the constraints imposed on estimated states, we need

$$G_1 F - G_2\hat{x}\,(k-N) - G_3 \preceq 0, \tag{6.71}$$

$$G_{x\lambda_A}\hat{x}\,(k-N) + G_{\lambda_A} \succeq 0, \tag{6.72}$$

where F is derived in (6.70). Eqs. (6.71) and (6.72) define a critical region $\mathcal{A}_x^j$ inside the admissible set $\mathcal{A}_x$. From the above discussion, we conclude that the optimal solution to (6.61) is an affine function of $\hat{x}(k-N)$ corresponding to the region $\mathcal{A}_x^j$. Theorem 6.4 is proven. ■

Theorem 6.4 succeeds in converting the design of RMHSO into an mp-QP problem and makes it possible to utilize existing solvers to obtain the partitions of the critical region $\mathcal{A}_x$ and the optimal solution F^o, e.g., the MATLAB-HYBRID Toolbox. Due to the existence of equality constraints, the mp-QP problem is quite complicated and the optimal solutions of F and $\mathcal{A}_x^j$ are memory-consuming. This fact impairs the implementation efficiency of RMHSO, one of the essentials of offline observation schemes. Therefore, we consider: *Is it possible to use the closed-loop prediction strategies to get simpler solutions (because only one-step prediction is necessary) and reduce the number of necessary parameters?*

Remark 6.6 *Theorem 6.4 solves the mp-QP problem with both element-wise inequality and equality constraints. Currently, how to solve this kind of problems remains an open problem.*

6.5 RMHSO using recursive closed-loop prediction

From Chapter 5, we know that recursive closed-loop prediction is able to utilize one-step prediction to simulate robust explicit MPC with an arbitrary prediction horizon. Therefore, we apply the similar prediction pattern to RMHSO design and formulate the problem in (6.39) as iteratively programming, i.e.,

$$
\begin{aligned}
J^o_{k-N\to k} \; = \; & \min_{\hat{f}(k-N)} (||C\hat{x}(k-N) - y(k-N)||^2_{Q_0} + ||\hat{f}(k-N)||^2_R + \qquad (6.73) \\
& (\min_{\hat{f}(k-N+1)} ||C\hat{x}(k-N+1) - y(k-N+1)||^2_Q \\
& +||\hat{f}(k-N+1)||^2_R + \cdots + (\min_{\hat{f}(k-1)} (||C\hat{x}(k-1) - y(k-1)||^2_Q \\
& +||\hat{f}(k-1)||^2_R + ||C\hat{x}(k) - y(k)||^2_{Q_0})))), \\
s.t. \quad & \hat{x}\,(k-i+1) = A\hat{x}\,(k-i) + Bu\,(k-i) + \hat{f}\,(k-i)\,, \quad i = 1, \,\cdots, N, \\
& \hat{f}\,(k-1) = L(C\hat{x}\,(k-1) - y(k-1))\;, \quad \hat{x}_{k-N+1\to k} \in \mathcal{A}_x, \\
& (-1)^\alpha(\text{row}(R^{1/2})\hat{f}\,(k-N)) \geq (n+n\varepsilon)^{1/2}\bar{\sigma}(\hat{Q}_0)\kappa \;\; (\alpha = 0 \text{ or } 1).
\end{aligned}
\tag{6.74}
$$

In other words, the intermediate piece objective $J_{k-N+i\to k}$ can be represented by

$$
J_{k-i\to k} = ||C\hat{x}(k-i) - y(k-i)||^2_Q + ||\hat{f}(k-i)||^2_R + J^o_{k-i+1\to k}. \tag{6.75}
$$

Similar as explicit robust MPC, the same prediction pattern in (6.73) is iterated N times for RMHSO, so that the prediction length of RMHSE is determined by the number of iteration loops. This feature enables us to implement RMHSO with an arbitrary horizon by implementing one-step prediction.

Remark 6.7 *Eq. (6.73) derives a recursive optimization problem and takes the advantage of closed-loop prediction. Meanwhile, it however proposes two challenges: how to derive the expression of the optimal piece objective $J^o_{k-i+1\to k}$ in terms of predicted observation $\hat{x}(k-i+1)$; and how to guarantee the expression of $J^o_{k-i+1\to k}$ to be a quadratic (or linear) function and remain the uniform structure for all piece objectives $J_{k-i\to k}$.*

6.5.1 Piece objective $J_{k-i\to k}$

From the above discussion, we know that two equality constraints are imposed on the arrival (terminal) observer gain and the initial observer gain. So when we choose the different value of i, the piece objective $J_{k-N+i\to k}$ $(1 \le i \le N-2)$ is associated with the different number of constraints. Two cases are discussed here:

Case I: Optimize the total objective $J_{k-N\to k}$. In this case, two constraints are imposed on both $\hat{x}\,(k-N+1)$ and $\hat{f}(k-N)$,

$$x_{\min} \preceq \hat{x}\,(k-N+1) \preceq x_{\max}, \tag{6.76}$$

$$(-1)^{\alpha}\mathrm{row}(R^{1/2})\breve{f} \preceq -(n+n\varepsilon)^{1/2}\bar{\sigma}(\hat{Q}_0)\kappa, \tag{6.77}$$

where (6.76) is a physical constraint and (6.77) is constructed for robust stability. Using $\hat{x}(k-N)$ to replace $\hat{x}(k-N+1)$ in (6.76), we have

$$\breve{f} \preceq x_{\max} - (A\breve{x} + B\breve{u}) \text{ and } -\breve{f} \preceq (A\breve{x} + B\breve{u}) - x_{\min}, \tag{6.78}$$

where $\breve{x}$, $\breve{u}$, and $\breve{f}$ represent the current signals for ease of notation, namely $\breve{x} := x(k-N)$ and similar to others. Stacking (6.77) and (6.78), we derive an element-wise inequality constraint for the total objective $J_{k-N\to k}$,

$$G_{\breve{f}}\breve{f} \preceq G_{\breve{c}} + G_{\breve{x}}\breve{x}, \tag{6.79}$$

where

$$G_{\breve{f}} := \begin{bmatrix} I \\ -I \\ (-1)^{\alpha}\mathrm{row}(R^{1/2}) \end{bmatrix},\ G_{\breve{c}} := \begin{bmatrix} x_{\max} - B\breve{u} \\ B\breve{u} - x_{\min} \\ -(n+n\varepsilon)^{1/2}\bar{\sigma}(\hat{Q}_0)\kappa \end{bmatrix} \text{ and } G_{\breve{x}} := \begin{bmatrix} -A \\ A \\ 0 \end{bmatrix}. \tag{6.80}$$

Case II: If $2 \le i \le N-2$, the initial estimated disturbance $\hat{f}(k-N)$ does not appear in the piece objective $J_{k-i\to k}$, so that the constraint in (6.79) is simplified as

$$G'_{\breve{f}}\,\breve{f} \preceq G'_{\breve{c}} + G'_{\breve{x}}\breve{x}, \tag{6.81}$$

where

$$G'_{\breve{f}} := \begin{bmatrix} I \\ -I \end{bmatrix},\ G'_{\breve{c}} := \begin{bmatrix} x_{\max} - B\breve{u} \\ B\breve{u} - x_{\min} \end{bmatrix} \text{ and } G'_{\breve{x}} := \begin{bmatrix} -A \\ A \end{bmatrix}. \tag{6.82}$$

Comparing (6.79) and (6.81) with (6.61), it can be seen that the constraints for RMHSO using closed-loop prediction are much simpler. For the cases of $i \neq N$, there is no need to consider the constraints imposed on $\hat{f}(k-N)$. As a result, we avoid the computational burden derived by the mixture of the augmented inequality and equality constraints. Following (6.75), the optimization of piece objective $J_{k-i\to k}$ turns out to be

$$\begin{aligned} J^o_{k-i+1\to k} &= \min_{\hat{f}(k-i)} J_{k-i\to k}, \quad (2 \le i \le N-1) \\ s.t. \quad & \hat{x}\left(k-i+1\right) = A\hat{x}\left(k-i\right) + Bu\left(k-i\right) + \hat{f}\left(k-i\right), \\ & \hat{x}\left(k-i+1\right) \in \mathcal{A}_x. \end{aligned} \tag{6.83}$$

Note that the problem in (6.83) excludes the case of total objective $J_{k-N\to k}$. We first assume that $J^o_{k-i+1\to k}$ is a quadratic function,

$$J^o_{k-i+1\to k} = ||\hat{x}(k-i+1)||^2_{Q_{i-1}} + \Gamma_{i-1}\hat{x}(k-i+1) + \Psi_{i-1}. \tag{6.84}$$

Inserting (6.84) into (6.75) and using $\hat{x}(k-i)$ to replace $\hat{x}(k-i+1)$, we have

$$J_{k-i\to k} = \frac{1}{2}\breve{f}^T H_{\breve{f}}\breve{f} + \breve{x}^T H_{\breve{f}\breve{x}}\breve{f} + Z_{\breve{f}}\breve{f} + H_{\breve{x}\Psi}, \tag{6.85}$$

where

$$\begin{aligned} H_{\breve{f}} &= 2Q_{i-1} + 2R, \quad H_{\breve{f}\breve{x}} = 2A^T Q_{i-1}, \;\; Z_{\breve{f}} = 2\breve{u}B^T Q_{i-1} + \Gamma_{i-1}, \\ H_{\breve{x}\Psi} &= ||C\breve{x} - \breve{y}||^2_{\bar{Q}} + ||A\breve{x} + B\breve{u}||_{Q_{i-1}} + \Gamma_{i-1}(A\breve{x} + B\breve{u}) + \Psi_{i-1}. \end{aligned} \tag{6.86}$$

Here $\breve{f} := \hat{f}(k-i)$ for the current signal. Notice that $H_{\breve{f}} \in \mathbb{S}^n_{++}$ and $H_{\breve{x}\Psi}$ is independent of $\breve{f}$, i.e., irrelevant to $J^o_{k-i\to k}$. From the definition of the piece objective $J_{k-i\to k}$ in (6.85), we can convert (6.83) into an mp-QP problem with the element-wise inequality constraints. Setting the different value of index i, the mp-QP problem for piece objective $J_{k-i\to k}$ is iterated $N-2$ times.

Remark 6.8 *Given the assumption on the quadratic form of $J^o_{k-i+1\to k}$, the mp-QP problem for the piece-objective $J_{k-i\to k}$ is given by*

$$\begin{aligned} J^o_{k-i+1\to k} &= \min_{\hat{f}(k-i)} J_{k-i\to k}, \quad (2 \le i \le N) \\ s.t. \quad & J_{k-i\to k} = \frac{1}{2}\breve{f}^T H_{\breve{f}}\breve{f} + \breve{x}^T H_{\breve{f}\breve{x}}\breve{f} + Z_{\breve{f}}\breve{f} + H_{\breve{x}\Psi}, \\ & G'_{\breve{f}}\,\breve{f} \preceq G'_{\breve{c}} + G'_{\breve{x}}\breve{x}. \end{aligned} \tag{6.87}$$

where $\breve{f}$ is the optimization variable and $\breve{x}$ is the multi-parameter vector (the current state observation).

Theorem 6.5 *The analytic (explicit) solutions to the piece-objective $J_{k-i\to k}$ defined in (6.87) are piece-wise affine functions of $\breve{x}$, over the corresponding state critical region $\mathcal{A}_x^j$, where index j denotes the jth critical region within the admissible state set $\mathcal{A}_x$.*

Proof: The proof is similar as that of Theorem 6.4, and to save space, here we only give the expression of the optimization solutions to (6.87), namely the piece-wise affine functions of $\breve{x}$ associated with the state critical regions $\mathcal{A}_x^j$. The optimal estimated disturbance is

$$\begin{aligned}\breve{f} = &(-H_{\breve{f}}^{-1}H_{\breve{f}\breve{x}}^T + H_{\breve{f}}^{-1}\tilde{G}_{\breve{f}}^T(\tilde{G}_{\breve{f}}H_{\breve{f}}^{-1}\tilde{G}_{\breve{f}}^T)^{-1}(\tilde{G}_{\breve{f}}H_{\breve{f}}^{-1}H_{\breve{f}\breve{x}}^T + \tilde{G}_{\breve{x}}))\breve{x}+\\ &H_{\breve{f}}^{-1}\tilde{G}_{\breve{f}}^T(\tilde{G}_{\breve{f}}H_{\breve{f}}^{-1}\tilde{G}_{\breve{f}}^T)^{-1}(\tilde{G}_{\breve{c}} + \tilde{G}_{\breve{f}}H_{\breve{f}}^{-1}\hat{Z}_{\breve{f}}^T) - H_{\breve{f}}^{-1}Z_{\breve{f}}^T\\ :=& L_i^j\breve{x} + O_i^j,\end{aligned} \tag{6.88}$$

where L_i^j can be regarded as the current observer gain corresponding to the jth critical region $\mathcal{A}_x^j$. The critical region $\mathcal{A}_x^j$ is

$$\begin{aligned}\mathcal{A}_x^j :=&\{\breve{x} \in \mathcal{A}_x|\ (L_i^j\breve{x} + O_i^j)\breve{f} \leq G_{\breve{c}} + G_{\breve{x}}\breve{x},\\ &(\tilde{G}_{\breve{f}}\tilde{H}_{\breve{f}}^{-1}\tilde{G}_{\breve{f}}^T)^{-1}\left(\tilde{G}_{\breve{f}}H_{\breve{f}}^{-1}H_{\breve{f}\breve{x}}^T + \tilde{G}_{\breve{x}}\right)\breve{x} + (\tilde{G}_{\breve{f}}\tilde{H}_{\breve{f}}^{-1}\tilde{G}_{\breve{f}}^T)^{-1}(\tilde{G}_{\breve{c}} + \tilde{G}_{\breve{f}}H_{\breve{f}}^{-1}Z_{\breve{f}}^T) \preceq 0\}.\end{aligned} \tag{6.89}$$

In the case that there are no active constraints out of the conditions (6.81), i.e., the row-independent combination $\{\tilde{G}'_{\breve{f}},\ \tilde{G}'_{\breve{c}},\ \tilde{G}'_{\breve{x}}\}$ do not exist, (6.88) and (6.89) degenerate to

$$\begin{aligned}\breve{f} = -H_{\breve{f}}^{-1}H_{\breve{f}\breve{x}}^T\breve{x} - H_{\breve{f}}^{-1}Z_{\breve{f}}^T := L_i^j\breve{x} + O_i^j,\\ G_{\breve{f}}\breve{f} - G_{\breve{c}} - G_{\breve{x}}\breve{x} \prec 0,\end{aligned}$$

which result in the second case of the explicit solutions to the mp-QP problem in (6.87),

$$\breve{f} = L_i^j\breve{x} + O_i^j\ \left(\forall\breve{x} \in \mathcal{A}_x^j\right), \tag{6.90}$$

where $\mathcal{A}_x^j := \{\breve{x} \in \mathcal{A}_x \mid G_{\breve{f}}\breve{f} - G_{\breve{c}} - G_{\breve{x}}\breve{x} \prec 0\}$. Obviously, the analytic (explicit) solutions to the mp-QP problem defined in (6.87) are piece-wise affine functions of $\breve{x}$. Theorem 6.5 is then proven. ■

Remark 6.9 *Theorem 6.5 offers the explicit solutions to the piece objective $J_{k-i\to k}$. The solutions are much simpler than those of Theorem 6.4 (open-loop MHSO). However, Theorem 6.5 is built on the assumption that $J^o_{k-i+1\to k}$ must be quadratic.*

Remark 6.10 *Replacing all of the parameters $\{G'_{\breve{f}}, G'_{\breve{c}}, G'_{\breve{x}}\}$ by $\{G_{\breve{f}}, G_{\breve{c}}, G_{\breve{x}}\}$, we can derive the optimal solutions to the last step iteration, i.e., the calculation of the total objective $J_{k-N\to k}$.*

6.5.2 Offline RMHSO using closed-loop prediction

The purpose of this subsection is to remove the assumption on $J^o_{k-i+1\to k}$ (Remark 6.9) and construct the affine solutions to $J_{k-N\to k}$. Note that the arrival observer gain L is determined by solving an algebraic Riccati equation, therefore, the number of the optimization variables is $N-1$ instead of the length of the prediction horizon N. The first piece objective to be optimized is $J_{k-2\to k}$ instead of $J_{k-1\to k}$.

Theorem 6.6 *The optimal solution to the piece objective $J_{k-i+1\to k}$ is a quadratic function of the observation $\hat{x}(k-i+1)$.*

Proof: The proof is same as that of Theorem 5.6. To save space, here we only write down the explicit expression of the optimal solution to the piece objective $J_{k-i+1\to k}$,

$$J^o_{k-1\to k} = ||\hat{x}(k-1)||^2_{Q_1} + \Gamma_1\hat{x}(k-1) + \Psi_1, \tag{6.91}$$

where

$$\begin{aligned}
Q_1 &:= C^TQC + L_1^TRL_1 + (CA+CL_1)^TQ_0(CA+CL_1),\\
\Gamma_1 &:= -2y^T(k-1)\,QC + 2O_1^TRL_1 + 2(CBu\,(k-1) + CO_1 - y\,(k))^TQ_0(CA+CL_1),\\
\Psi_1 &:= ||y(k-1)||^2_Q + ||O_1||^2_R + ||CBu(k-1) + CO_1 - y(k)||^2_{Q_0},\\
L_1 &:= LC, \text{ and } O_1 := -Ly(k-1).
\end{aligned}$$

And

$$J^o_{k-i+1\to k} = ||\hat{x}(k-i+1)||^2_{Q_{i-1}} + \Gamma_{i-1}\hat{x}(k-i+1) + \Psi_{i-1}, \tag{6.92}$$

where

$$\begin{aligned}
Q_{i-1} &:= C^TQC + L_{i-1}^TRL_{i-1} + (A + L_{i-1})^TQ_{i-2}(A + L_{i-1}), \\
\Gamma_{i-1} &:= -2y^T(k-i+1)\,QC + 2O_{i-1}^TRL_{i-1} \\
&\quad +(2(Bu\,(k-i+1) + O_{i-1})^TQ_0 + \Gamma_{i-2})(A + L_{i-1}), \\
\Psi_{i-1} &:= ||y(k-i+1)||_Q^2 + ||O_{i-1}||_R^2 + ||Bu(k-i+1) \\
&\quad +O_{i-1}||_{Q_0}^2 + \Gamma_{i-2}(Bu(k-i+1) + O_{i-1}) + \Psi_{i-2}.
\end{aligned}$$

Note that Γ_{i-1} and O_{i-1} are the expressions of $y\,(k-i+1)$ and $u(k-i+1)$. In other words, Γ_{i-1} and O_{i-1} collect the information of past inputs and outputs. ■

Remark 6.11 Γ_{i-1} *collects past input and output information, and also* Γ_{i-1} *is a term of* $Z_{\hat{f}_{(k-i)}}$ *which influences the observer gain* L_i *and the admissible state partitions (refer to (6.86)). So the optimal* L_i*, equivalent to* $\hat{f}_{(k-i)}$*, is the composition of past inputs and outputs.*

Combining Theorems 6.5 and 6.6, we can derive the optimal solutions of all piece objectives $J_{k-i\to k}$ $(1 \leq i \leq N)$ and the corresponding observer gain L_i^o. Consequently, the current state observation can be obtained by

$$\hat{x}(k) = A^N\hat{x}(k-N) + \sum_{j=0}^{N-1} A^jBu(k-1-j) + \sum_{j=0}^{N-1} A^j\hat{f}(k-1-j). \qquad (6.93)$$

6.6 Algorithms of RMHSO

From the above discussion, RMHSO is converted into a set of mp-QP problems. A series of offline observer polices are developed to reduce offline computational burden and facilitate online implementation. To perform state predictions, the initial state observation $\hat{x}(k-N)$ is necessary. How to setup the initial conditions of RMHSO is covered in this section.

6.6.1 The initial setup

We will use the full information state observer to determine the sequence $\hat{x}_{1\to N}$, i.e., the initial setup of RMHSO. Here the problem is given as follows:

$$
\begin{aligned}
\hat{f}_{0\to i} &:= \arg \min_{\hat{f}_{0\to i}} \|C\hat{x}(i) - y(i)\|^2_{Q_0} + \sum_{j=0}^{i-1} \|C\hat{x}(j) - y(j)\|^2_Q + \left\|\hat{f}(j)\right\|^2_R, \qquad (6.94)\\
s.t. \quad & \hat{x}(i) = A^i\hat{x}(0) + A^{i-1}Bu(0) + \cdots Bu(i-1) + A^{i-1}\hat{f}(0) + \cdots + \hat{f}(i-1),\\
& \hat{x}(k+i) \in \mathcal{A}_x \ (0 < i \leq N).
\end{aligned}
$$

It can be seen that the dimension of (6.94) is increasing while collecting more input and output data, but because the horizon length N is not too large, the full information state observer is still practical and effective. Fig. 6.8 illustrates the integration of the full information state observer and a RMHSO. In the figure, the trajectory of $\hat{x}(k)$ is composed of two segments: one spans from the initial instant to instant N, and the other starts at instant $N+1$ and proceeds to future. The two shadowed regions represent the moving horizon windows which are shifted one step ahead while iteratively implementing RMHSO. The thin solid line shows the optimal trajectory derived by full information observer, the dark solid is obtained by RMHSO, and the dot-dashed line simulates the optimization of RMHSO whose prediction horizon windows are shifted one-step ahead.

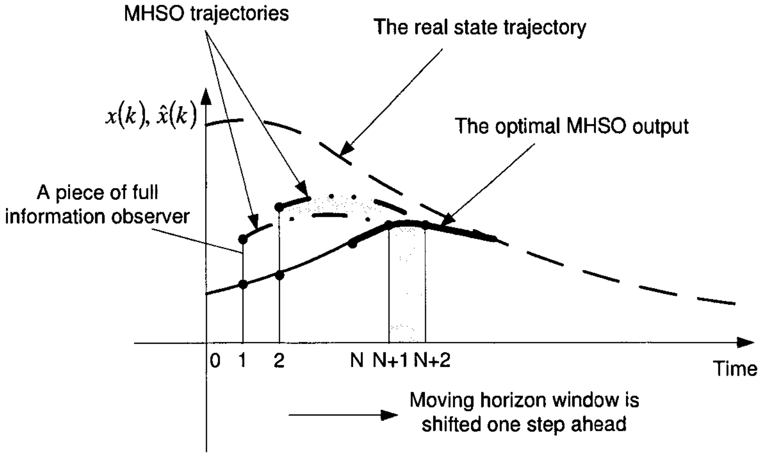

MHSO trajectories
The real state trajectory
$x(k), \hat{x}(k)$
The optimal MHSO output
A piece of full information observer
0
1
2
N
N+1
N+2
Time
Moving horizon window is shifted one step ahead

Figure 6.8: The theory of MHSO design

6.6.2 Algorithms

Based on Theorems 6.3-6.4 and Theorems 6.5-6.6, we can develop the open-loop and closed-loop RMHSO, respectively. The two pairs of theorems are both featured by offline optimization and online implementation, so that they can be associated with explicit RMPC design.

Algorithm I (Open-loop MHSO)

1. Setup the initial observation $\hat{x}_{1 \to N}$ based on full information state observer and store the optimal solutions (refer to the problem in (6.94)).

2. Execute closed-loop robust stability analysis. Choose eligible tuning parameters ν and P, and solve a Riccati equation and an semi-definition program to derive Q_0, L_1, and constraints imposed on estimated disturbance $\hat{f}(k-N)$ in Theorem 6.2.

3. Define augmented matrices $\mathcal{A}$, $\mathcal{B}$, $\mathcal{B}_F$, $\mathcal{C}$, $\mathcal{Q}$, $\mathcal{R}$, Θ, Ξ, and Π in (6.54). Form the mp-QP objective and the constraint parameters G_1, G_2, G_3, Γ_F, Γ_x, and Γ_U (memory-consuming) for open-loop MHSO in (6.60).

4. Stack the input/output measurements U, Y and online partition the admissible state set $\mathcal{A}_x$, i.e., deriving $\{\tilde{G}_1^j, \tilde{G}_2^j, \tilde{G}_3^j\}$. "$j$" is the index of the state space partitions.

5. Derive the optimal sequence $\hat{f}_{k-N \to k}$ based on auxiliary matrices $G_{\lambda\lambda}$, $G_{x\lambda_2}$, G_{λ_2}, $G_{x\lambda_A}$, G_{λ_A}, G_{xF}, and G_F defined in (6.67) - (6.70) (memory-consuming).

6. Implement $\hat{f}_{k-N \to k}$ from (6.38) and derive the current optimal state observation $\hat{x}(k)$. Purge the memories for intermediate matrices, partitions, and optimal sequences $\hat{f}_{k-N \to k}$.

7. If $k > t$, exit. Otherwise update U, Y and go to Step 4. Here "t" is the prespecified observation length.

In Algorithm I, all steps before Step 4 are completed offline – offline optimization, and all steps from Step 4 are done online – online implementation. This procedure is different from that of offline MPC whose state space partitions are performed offline.

Algorithm II (RMHSO with the recursive closed-loop prediction)

1 Steps 1 and 2 are same to those of Algorithm I.

3 Derive the optimal expressions of L_1 and O_1, and store the parameters of $J_{k-1\to k}$, i.e., Q_1, Γ_1, and Ψ_1 in (6.91). Set $i = 2$, the index of recursive optimization loops.

4 Define the optimal solutions to the mp-QP problem of the piece objective $J_{k-i\to k}$, i.e., $\hat{f}_i^j$, O_i^j, Q_i^j, Γ_i^j, and Ψ_i^j in (6.92), and store the corresponding state space partitions $\{\mathcal{A}_i^1, \cdots, \mathcal{A}_i^{N_p}\}$ where N_p is the number of partitions.

5 Identify the active partition from the set $\{\mathcal{A}_i^1, \cdots, \mathcal{A}_i^{N_p}\}$, based on the measurements $u(k-i)$, $y(k-i)$, and $y(k-i+1)$. Suppose that the jth partition is active. Keep $\hat{f}_i^j$, O_i^j, Q_i^j, Γ_i^j, and Ψ_i^j, and purge the memories for other optimal solutions corresponding to the partitions but $\mathcal{A}_i^j$. Set $i = i+1$.

6 Check whether $i = N$, if yes store $\hat{f}_{k-N\to k}^j$ and reject all other intermediate solutions. Otherwise go to Step 4.

7 Implement the optimal observer gain $\hat{f}_{k-N\to k}^j$ from (6.93) and derive the current optimal state observation $\hat{x}(k)$. Purge the memories for intermediate matrices, partitions, and optimal sequences $\hat{f}_{k-N\to k}$.

8 If $k > t$, exit. Otherwise go to Step 3.

Remark 6.12 *Comparing Algorithms I and II, the former costs more memories for intermediate solutions, and also the augmented matrices may lead to some feasibility problems. The latter utilizes recursive optimization and reduces the computational cost but two level iterative loops may lower the implementation efficiency.*

6.6.3 RMHSO to systems with measurement noises

In the above discussion, we assume the measurement noise $v(k) = 0$, i.e., we use model (6.35) instead of (6.28) for the open-loop and closed-loop RMHSO design.

However, $v(k)$ is ubiquitous in real plants, and how to incorporate $v(k)$ with RMHSO design is a nontrivial problem. Motivated by [68, 72], this problem can be solved by introducing a noise model. For a simple case, we can just rewrite the system in (6.28) as

$$\begin{aligned} z(k+1) &= \underline{A}x(k) + \underline{B}u(k) + B_f f(x(k), d(k), k), \\ y(k) &= \underline{C}z(k), \end{aligned} \tag{6.95}$$

where $z(k) := [x^T(k),\ v^T(k)]^T$,

$$\underline{A} = \begin{bmatrix} A & 0 \\ 0 & A_d \end{bmatrix},\ \underline{B} = \begin{bmatrix} B \\ 0 \end{bmatrix},\ B_f = \begin{bmatrix} I \\ B_{fd} \end{bmatrix} \text{ and } \underline{C} = [C,\ I].$$

So we can proceed with the above discussion based on model (6.95), and use the different value of Q to tune the observer performance. Because of the limitation of space, here we choose not to discuss how to derive matrices A_d and B_{fd}. For the interested, please refer to [68, 72] for details.

6.7 A simulation example for RMHSO

The system is given by

$$\begin{aligned} x(k+1) &= (A + \delta_A(k))x(k) + B_d w(k) \\ y(k) &= Cx(k), \end{aligned}$$

where $\delta_A(k)$ and $w(k)$ represent system's internal and external uncertainties, respectively. The system parameters are known as

$$A = \begin{bmatrix} 0.99 & 0.2 \\ -0.1 & 0.3 \end{bmatrix},\ B_d = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ C = [1,\ 3],$$

and both internal and external uncertainties are bounded by 0.5, i.e.,

$$-0.5 \le w(k) \le 0.5, \text{ and } \bar{\sigma}(\delta_A(k)) \le 0.5.$$

To reflect the different influence of internal uncertainties and external disturbances, we perform the simulations under two conditions: (1) set $\delta_A(k) = 0$ and call the random function in MATLAB to simulate $w(k)$ in order to demonstrate the influence of external disturbances, e.g., "*rand*"; (2) set $\delta_A(k) \neq 0$, and call "*rand*" to create both $w(k)$ and $\delta_A(k)$, i.e., simulate the combined internal uncertainty and external disturbance. Reformulate the uncertainties into the form of (6.33), we can derive the uncertainty bounds for two cases, $\kappa_1 = 0.5$ and $\kappa_2 = 1.25$. To guarantee stability, the arrival weighting Q_0, the arrival observer gain L, and the initial estimated disturbance $\hat{f}(k-N)$ are determined by solving an algebraic Riccati equation and a semi-definite optimization problem. The related parameters are given in Table 6.1. Set the prediction horizon $N = 3$. Two algorithms are employed in the sequel, namely, the open-loop RMHSO and closed-loop RMHSO with recursive optimization. Fig. 6.9 is the simulation results for the observers under Condition 1. We find that under Condition 1, both the RMHSO algorithms and the nominal MHSO can work well. The left two columns in Table 6.2 list the means and variances of the observation errors derived by the three different types of MHSO. It can be seen that offline RMHO (our algorithms) are better than nominal MHSO, but the improvement is not remarkable. So we repeat the stimulation again and set a nonzero

Table 6.1: Simulation parameters of offline RMHSO

Two cases	$\delta_A(k) = 0$	$\delta_A(k) \neq 0$
Q, R, P, ν,	I, $3I$, $4I$, 0.8	I, $4I$, $4I$, 0.8
Q_0	1.3897	1.4073
ε	0.1664	0.1453
$L_1(LC)$	$\begin{bmatrix} 0.6254 & 0.1514 \\ -0.0014 & 0.2526 \end{bmatrix}$	$\begin{bmatrix} 0.5751 & 0.1437 \\ 0.0109 & 0.2395 \end{bmatrix}$
$\hat{f}(k-N)$	$(-1)^{\alpha}\mathbf{1}\hat{f}(k-N) \leq 1.6473$	$(-1)^{\alpha}\mathbf{1}\hat{f}(k-N) \leq 3.5608$

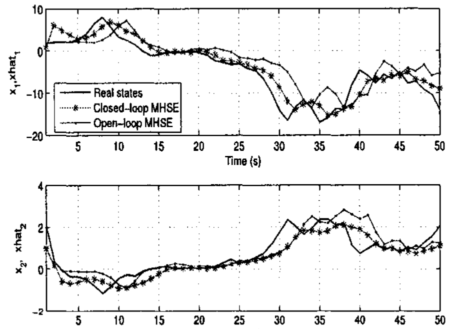


Figure 6.9: Comparison of observers with external uncertainties

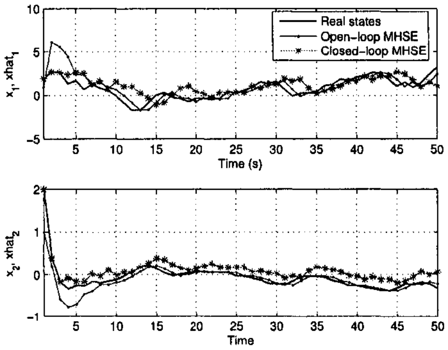


Figure 6.10: Comparison of observers with internal uncertainties

Table 6.2: Means and Variances for RMHSO errors

	Means	Variances	Means	Variances
Nominal MHSO	$[1.8129, 0.1200]$	$[13.3990, 0.3042]$	–	–
Open-loop MHSO	$[1.0577, -0.1820]$	$[6.2067, 0.1341]$	$[0.2850, 0.1709]$	$[0.0097, 1.0624]$
Closed-loop MHSO	$[0.7126, -0.1864]$	$[5.0215, 0.1880]$	$[0.1739, -0.0513]$	$[0.8863, 0.0325]$

internal disturbance $\delta_A(k)$. Under Condition 2, we find that MHSO becomes unstable, so that in Table 6.2 right columns do not give the means and variances for this case. But offline RMHSO still works well. Fig. 6.10 illustrates the dynamics of both open-loop and closed-loop RMHSO.

All simulations are performed using a laptop with a Pentium 4 processor and a 512MB-RAM. From Figs. 6.9 and 6.10, it is hard to say whether the closed-loop RMHSO gives better observation than open-loop RMHSO. But we can compare the simulation time-costs and memory-costs. Keeping the simulation length equal to 50, the open-loop RMHSO costs 8.4810 seconds and its data file takes 11KB of capacity, but for the closed-loop one, time cost increases to 16.6330 seconds (two level iterations) and date file decreases to 1KB. The simulation results are consistent with the theoretical analysis.

6.8 Conclusions

In this chapter, we developed two types of robust observers for systems with both internal uncertainties and external disturbances, namely robust state observer using MAXDET programming and robust moving horizon state observer (RMHSO) using mp-SQP. Two prediction patterns are employed for the RMHSO: forward open-loop prediction and recursive closed-loop prediction.

The open-loop RMHSO converts observer design into an mp-QP problem imposed by element-wise inequality and equality constraints. Although the equality constraints lead to more computational complexity, the optimal solutions are a set of piece-wise affine functions of the initial state observation. The closed-loop RMHSO constructs a novel recursive optimization pattern and realizes multiple-loop observations with one one-step necessary prediction. Comparing these two algorithms, it can be seen that the former suffers from high offline computational burden and needs more memories for intermediate parameters, it however leads to faster online implementation. The latter does not mix up the inequality constraints with equality constraints in each optimization loops so that it requires a smaller amount of memory for intermediate parameters, but closed-loop RMHSO spends a longer time on observer implementation. The contents of this chapter are summarized in our publication [20].

Chapter 7

Industrial applications

The aim of this chapter is to verify the effectiveness of the explicit robust model predictive control (ERMPC) and moving horizon state observation (MHSO) in industrial applications. To this end, ERMPC and MHSO are applied to the SYNSIM model for the co-generation system regulation.

SYNSIM is a simulation package developed by researchers of the University of Alberta and Engineers in the Syncruded Canada Ltd. (SCL). It is working under the Matlab-Simulink environment and based on the field plants owned and operated by the SCL in Fort Mcmurray, AB, Canada [86]. It is a complicated, nonlinear simulation package, but an effective tool to test the controllers for co-generation systems.

This chapter is composed of two parts: Part I identifies the model of the loop from the firing rate to the 900# header in SYNSIM; and Part II utilizes the identified model to design a master controller.[1]

[1] In this chapter, the master controller is referred to as the feedback controller for the loop from the firing rate to the 900# header pressure.

7.1 System identification

The plant to be studied is an industrial co-generation system, owned and operated by the Syncrude Canada Ltd. in Fort McMurray, AB, Canada. This plant is an integrated energy facility consisting of a boiler subsystem, a header subsystem, a letdown subsystem, and an electricity generating subsystem. The boiler subsystem produces steam by three utility boilers, three CO boilers, and two one-through steam generators (OTSGs). The header subsystem receives steam from boilers and stores it in the different headers that operate at different pressures, namely the 900# header, the 600# header, the 150# header, and the 50# header. The number here indicates the pressure of stored steam. For example, the steam in 900# header has the pressure of 900 *psi.* Through the header subsystem, steam is distributed to the electrical subsystem and then transformed to electricity by steam turbines. The letdown subsystem is used to convert steam from one pressure to another. Four types of valves exist in the letdown subsystem, i.e., 900# to 600# valve, 600# to 150# valve, 600# to 50# valve, and 150# to 50# valve. Fig. 7.1 illustrates the interconnection of the subsystems and indicates the loop from the firing rate to the 900# header pressure (in dashed lines) which is called as the master loop in this chapter. From experimental facts, it can be seen that the 900# header pressure is critical to the quality of steam production, and fortunately it can be regulated by the firing rate. To maintain the pressure around 6.306 Mpa or 900 Psi (static operating point), we have to guarantee that the firing rate settles down upon 0.7117. In this chapter, we will first identify the model of the master loop, and then design a master controller for pressure regulation. The master loop input, namely the firing rate, has two physical constraints, i.e., the saturation limit and derivative limit,

$$0 < u < 1, \ -0.16/60 < \Delta u < 0.16/60, \tag{7.1}$$

where Δu denotes the derivative. In the sequel, the constraints in (7.1) are incorporated with the master controller design.

We assume that the master loop can be represented by a state space model

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w(k), \\ y(k) &= Cx(k) + Du(k) + v(k), \end{aligned} \tag{7.2}$$

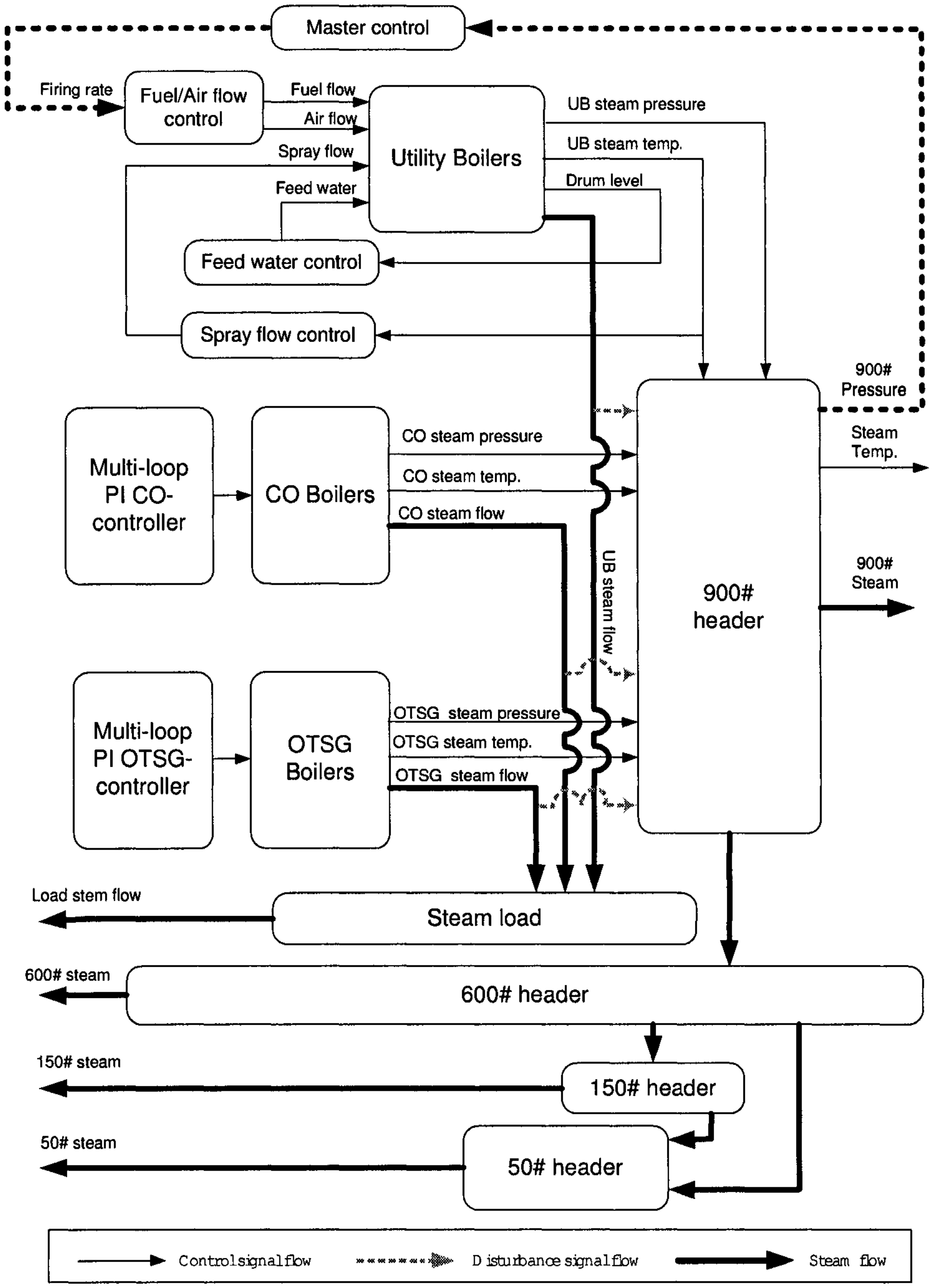


Figure 7.1: The co-generation system

where $x(k)$, $y(t)$, and $u(k)$ stand for the state, output, and input respectively. The matrices $(A,\ B,\ C,\ D)$ are constant with appropriate dimensions. The direct feedthrough matrix D is usually equal to zero due to time delay from the firing rate to the 900# header pressure. $w(k)$ is the process disturbance, and $v(k)$ is the measurement noise. They are independent random sequences with zero mean and zero covariance; thus

$$\mathrm{Cov}(w(k)) = R_w,\ \mathrm{Cov}(v(k)) = R_v, \text{ and } \mathrm{Cov}(w(k),\ v(k)) = 0. \tag{7.3}$$

We assume that the pair $(C,\ A)$ is observable and the pair $(A,\ BR_w^{1/2})$ is controllable. In the sequel, we first choose a step excitation to identify the approximate time constant and time-delay factor of the system (7.2), and then use these parameters to design a pseudorandom binary sequence (PRBS) for state space identification.

7.1.1 Second order plus dead-time identification

For chemical processes, it has been recognized that their dynamics may in general be simplified to a first order plus dead time (FOPDT) system or a second order plus dead time (SOPDT) system. From step tests, we find that the trajectory of the 900# header pressure has both overshoot and oscillation. Therefore the order of the master loop should be a second- or higher-order system. Here we employ the SOPDT identification method to derive an approximation to the model in (7.2). References [40, 97] proposed an effective SOPDT algorithm to determine the static gain K, time-delay factor T_d, and time-constant τ, graphically. Suppose that an SOPDT system has the form of

$$G_o(s) = \frac{Ke^{-T_d s}}{\tau^2 s^2 + 2\xi\tau s + 1}, \tag{7.4}$$

where ξ denotes the damping ratio. The step response of System (7.4) can be illustrated by Figs. 7.2 and the mathematical expression of the output trajectory is given by

$$\begin{aligned} y(t) &= Ku(t-T_d)(1 - e^{-(t-T_d)}\omega_n\xi(\frac{\xi}{\sqrt{1-\xi^2}}\sin(\sqrt{1-\xi^2}\omega_n(t-T_d)) \\ &\quad +\cos(\sqrt{1-\xi^2}\omega_n(t-T_d)))), \end{aligned} \tag{7.5}$$

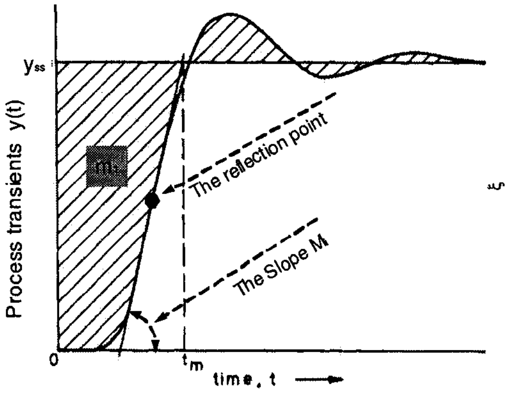


Figure 7.2: SOPDT system identification

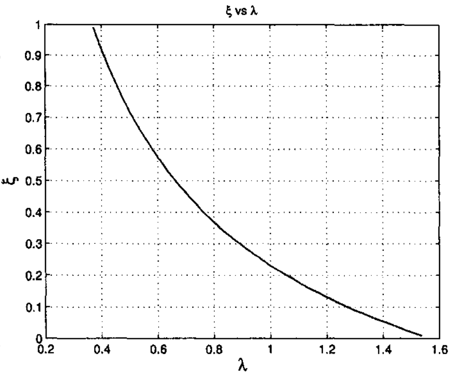


Figure 7.3: The graph of ξ vs λ

where $\omega_n = 1/\tau$ is the fundamental frequency. From Fig. 7.2, we can define a tuning parameter λ satisfying

$$\lambda = (t_m - m_1)M_i. \tag{7.6}$$

Here, t_m denotes the time instant when the tangent line crossing the inflection point of the output trajectory Fig. 7.2 first intersects with the static output y_{ss}. m_1 is the area of shaded regions in Fig. 7.2 in which the area takes positive value if the trajectory is below y_{ss}; otherwise, it takes negative value. M_i is the slope of the tangent line crossing the inflection point. By reading the process transient response, we can determine the value of λ, which is tightly related to the characteristic parameter ξ via

$$\lambda = \frac{\cos^{-1}\xi}{\sqrt{1-\xi^2}}\exp(\frac{-\xi}{\sqrt{1-\xi^2}}\cos^{-1}\xi). \tag{7.7}$$

Seemingly, by solving (7.7) we can derive the value of ξ from λ. However, Eq. (7.7) provides finite number of roots mapping from λ to ξ unless $0 < \xi < 1$. It can be shown that by decreasing the value of λ from $\frac{\pi}{2}$ to e^{-1}, the value of ξ increases from 0 to 1 monotonically. Fig. 7.3 shows the relationship between λ and ξ in the region of $\xi \in [0,\ 1]$. Therefore, after deriving the value of λ from Fig. (7.2), we can read out the value of ξ from Fig. 7.3. After that, the characteristic parameters ω_n and T_d can be determined by

$$\omega_n = \frac{1}{\tau} = \frac{\cos^{-1}\xi}{\sqrt{1-\xi^2}(t_m - m_1)}, \tag{7.8}$$

$$T_d = m_1 - \frac{2\xi}{\omega_n}. \tag{7.9}$$

Eqs. (7.6), (7.8), and (7.9) provide a simple and effective way to identify an SOPDT system.

Table 7.1 lists the static operating point of the co-generation system. In a small region around the operating point, we perform SOPDT identification. Skipping the identification details, the SOPDT model in (7.4) is finally derived as

Table 7.1: The working point of the co-generation system

BFW Inlet Temperature (^oC)	141.3
FW Flow (kg/s)	89.8415
Fuel Flow Rate (kg/s)	4.9826
Air Flow Rate (kg/s)	92.2323
Spray Flow Rate (kg/s)	2.6021
Firing Rate (%)	71.17
Drum Temperature (^oC)	500
Drum Level (m)	1
Drum Pressure (Mpa)	7.0186
Head 900# Pressure (Mpa)	6.3060
UB Flow Rate (kg/s)	277.3304
CO Boiler Flow Rate (kg/s)	185.3823
OTSG Flow Rate (kg/s)	47.9204
The number of boilers	3UB, 3CO, 3OTSG
Total UB Load (kg/s, $kpph$)	277.4275 (2200)
Firing Rate (%)	71.17
Ratio of Fuel Flow Rate to Firing rate	7
Ratio of Air Flow Rate to Fuel Flow Rate	18

$$G_o(s) = \frac{1.609}{12.57s^2 + 1.002s + 1}, \tag{7.10}$$

where $\tau = \frac{1}{\varpi_n} = 3.5449\,\text{min}$, $k = 1.609$, and $\xi = 0.1413$. From (7.10), we can plot out the simulated step-response and the real step-response in the same window shown in Fig. 7.4. Although the SOPDT model in (7.10) is not accurate enough for the master controller design, it provides a criterion to design PRBS excitations for state space identification. From experiential equations [96], the approximate

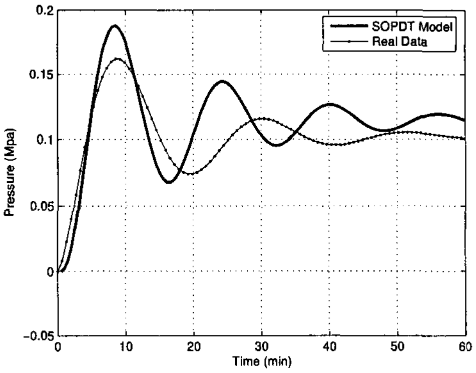


Figure 7.4: The step responses

sampling period T_s of the master loop, i.e., the sampling rate of an appropriate PRBS, can be determined by:

$$\begin{aligned} T_s &= \left(\frac{1}{10} \sim \frac{1}{20}\right)\tau = \left(\frac{1}{10} \sim \frac{1}{20}\right) \times 3.5449 \times 60 \\ &= 7 \sim 22\,\text{sec}\,. \end{aligned} \tag{7.11}$$

Moreover, the crossing frequency ϖ_b of the master loop is

$$\varpi_b = \frac{1}{\tau} = \frac{1}{3.5449 \times 60} = 0.0047,$$

and the Nyquist frequency ϖ_N of the master loop is

$$\varpi_N = \frac{\pi}{T_s} = \frac{\pi}{12} = 0.2618.$$

From the definition of a PRBS frequency band [53], we can derive the upper bound of the PRBS frequency band k_2 by

$$k_2 = k_1 \frac{\varpi_b}{\varpi_N} = k_1 \frac{0.0047}{0.2618} \approx (0.02 \sim 0.08)\,,\ \text{as}\ k_1 = 2 \sim 4.$$

Remark 7.1 *Using the MATLAB commands*

$$\begin{aligned} ul &= idinput(60000,\ \text{'}rbs\text{'},\ [0,\ 0.04],\ [-0.7117 \times 0.04,\ 0.7117 \times 0.04]), \\ u2 &= idinput(60000,\ \text{'}rbs\text{'},\ [0,\ 0.02],\ [-0.7117 \times 0.025,\ 0.7117 \times 0.025]), \\ u3 &= idinput(60000,\ \text{'}rbs\text{'},\ [0,\ 0.1],\ [-0.7117 \times 0.04,\ 0.7117 \times 0.04]), \\ &\ \ \ldots \end{aligned} \tag{7.12}$$

we can create a series of PRBS sequences for the state space identification of the master loop. Note that the value of “0.7117” is the static operating point of the firing rate in Table 7.1.

7.1.2 A state space model

Based on Remark 7.1, we can create a series of the PRBS excitations for the state space identification. From experiments, it can be seen that the u_2 in (7.12) derives the best identification result. Sending u_2 to the master loop, it derives a set of input and output data. Using MATLAB commands, we can stack the input and output data together and derive an “iddata” z, which is shown in Fig. 7.5. From z, the best identification of the master loop is a state space model with 5 order (see Fig. 7.6). This 5th order system is given by

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Ke(k), \\ y(k) &= Cx(k) + Du(k) + e(k), \end{aligned} \tag{7.13}$$

where

$$A = \begin{bmatrix} 0.9984 & -0.0357 & 0.0040 & -0.0008 & 0.0012 \\ 0.0546 & 0.9148 & 0.1155 & -0.0493 & 0.0045 \\ -0.0286 & 0.20445 & 0.6499 & 0.3814 & -0.0164 \\ 0.0088 & -0.1487 & 0.3305 & 0.4402 & -0.3272 \\ 0.0019 & 0.0041 & -0.0112 & 0.0160 & 0.3939 \end{bmatrix}, \; B = \begin{bmatrix} 0.0055 \\ -0.9652 \\ 2.9157 \\ -3.9774 \\ -4.2840 \end{bmatrix},$$

$$C = [\ 206.8800 \quad -3.6375 \quad 0.3470 \quad -0.0655 \quad 0.0547\ ], \; D = 0,$$

$$K = [\ 0.0080 \quad -0.1054 \quad 0.0226 \quad -0.0315 \quad -0.1360\ ]^T.$$

Note that the pair (A, B) in (7.13) is controllable, and (C, A) is observable. In programming, the “iddata” z is divided into two groups, namely $z := [ze,\ zv]$. ze is used for system identification and zv for model validation. From the data zv, we perform model validation in several ways, namely by output fitness (Fig. 7.7), residue analysis (Fig. 7.8), step responses (Fig. 7.9), and spectrum analysis (Fig. 7.10). Moreover, we check the zeros & poles of the model in (7.13) (see Fig. 7.11) and the DFT for the input signal u_2 (see Fig. 7.12). From Figs. 7.7 - 7.12, we can say that the model (7.13) is accurate enough for the master controller design.

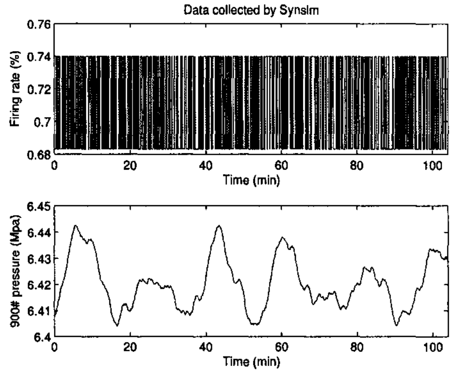
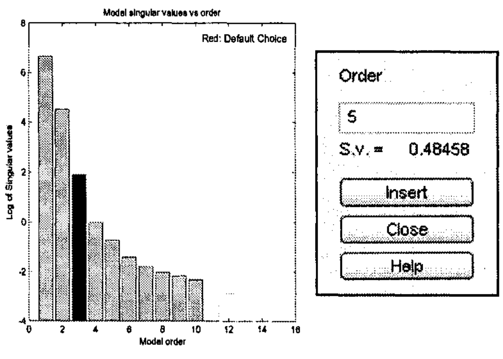


Figure 7.5: The PRBS excitation and the corresponding response



Order

5

S.V. = 0.48458

Figure 7.6: Model structure selection

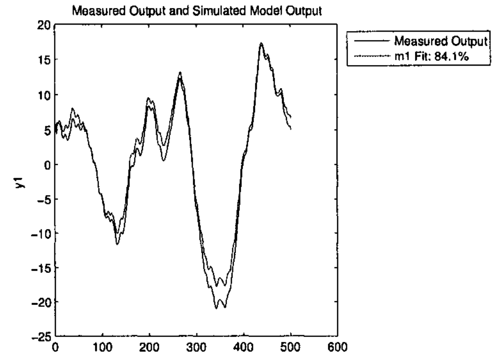


Figure 7.7: The output fitness

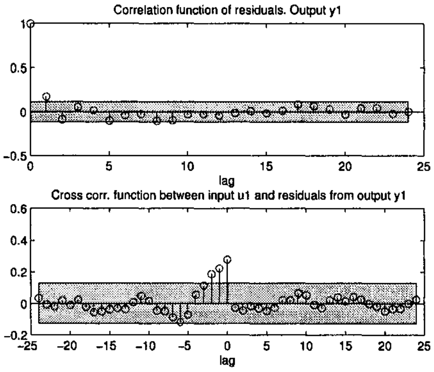


Figure 7.8: The residue analysis

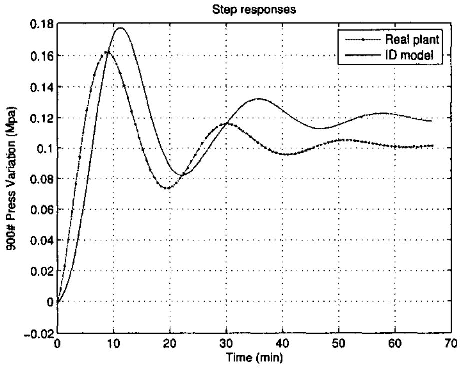


Figure 7.9: The step responses

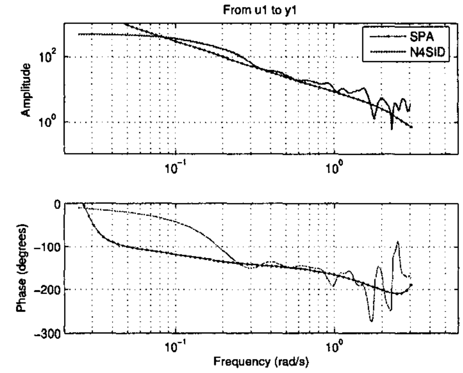


Figure 7.10: The spectrum analysis

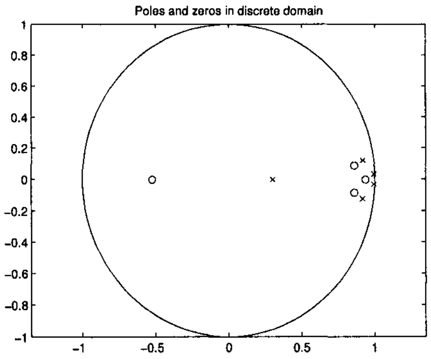


Figure 7.11: The Bode plots of the real process and identified model

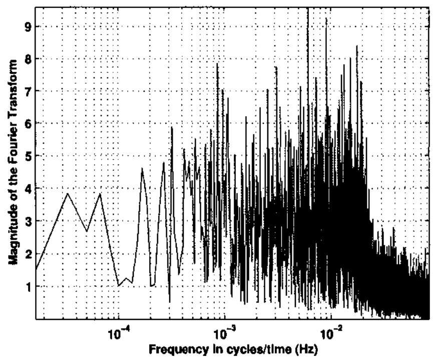


Figure 7.12: The DFT for the input u_2

7.2 The master controller design

A proportional-integral (PI) master controller exists in SYNSIM to regulate the firing rate and maintain the 900# header pressure in an acceptable region. Fig. 7.13 shows the simulink diagram of this controller. The function of the anti-windup block in Fig. 7.13 is to handle the physical constraints of the firing rate in Eq. (7.1). Chapter 4 has shown that anti-windup strategies have two critical limitations: 1). the parameters of anti-windup controllers have to be chosen by trial-and error; 2).

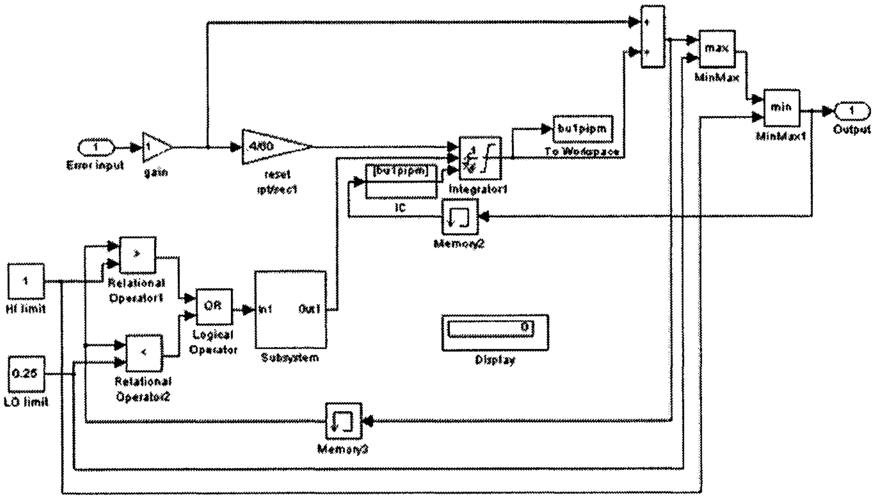


Figure 7.13: The simulink diagram for the PI master controller

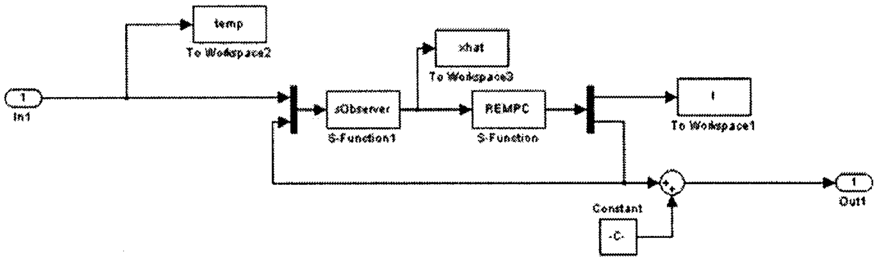


Figure 7.14: The simulink diagram for the explicit MPC master controller

in the presence of large disturbances, the parameters have to be continuously adjusted to attenuate system unstable behaviors. Concerning these disadvantages, in this section we choose explicit robust MPC to design an analytic master controller. The identified model in (7.13) is used for state prediction and offline optimization. Note that the states in (7.13) are constructed by MATLAB commands and they do not have physical meaning. Therefore, the states are unmeasured in the master controller design. We have to use the algorithms developed in Chapter 6 to design a moving horizon state observer and incorporate the observer with MPC formulation. In the sequel, the analytic master controller and the state observer are first designed, and then integrated with SYNSIM together to evaluate control performance. Fig. 7.14 shows the Simulation block for the analytic master controller. In Fig.7.14, both the controller and the observer are realized by S-function programming which facilitates the analytic master controller. Setting the initial admissible state set as

$$\mathcal{A}_x := \{x \in \mathbb{R}^5 | \ -10 \cdot \mathbf{1} \preceq x \preceq 10 \cdot \mathbf{1}\} \tag{7.14}$$

where $\mathbf{1} \in \mathbb{R}^5$ denotes the full-one vector. The analytic master controller is stored by a MATLAB structure variable "*expcon*1",

$$\boxed{\begin{array}{ll} expcon1 = & \\ & H: \ [31 \times 5 \ \text{double}] \\ & K: \ [31 \times 1 \ \text{double}] \\ & F: \ [27 \times 5 \ \text{double}] \\ & G: \ [27 \times 1 \ \text{double}] \\ & i1: \ [9 \times 1 \ \text{double}] \\ & i2: \ [9 \times 1 \ \text{double}] \\ & thmin: \ [5 \times 1 \ \text{double}] \\ & thmax: \ [5 \times 1 \ \text{double}] \\ & nr: \ 9 \\ & nu: \ 3 \\ & npar: \ 5. \end{array}} \tag{7.15}$$

In (7.15), the fields H and K store the parameters of $\mathcal{A}_x$'s partitions; and the fields F and G store the parameters of the feedback affine functions. $i1$ and $i2$ are the indices of the critical regions; and nr is the number of critical regions. By accessing the elements of H, K, F, G, $i1$, and $i2$, we can explicitly express the control polices for the master controller. Due to the limitation of space, here we just give the expression of the control policy in region #1, i.e.,

$$\begin{array}{ll} & u(k) = \begin{bmatrix} 0.555 & -0.069 & 0.069 & 0.069 & 0.069 \\ 0.242 & -0.191 & 0.104 & 0.087 & 0.139 \\ 0 & 0 & -0.043 & 0.015 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 0.005 \\ 0.003 \\ 0 \end{bmatrix}, \\ \text{if} & \begin{bmatrix} 15.470 & -5.701 & 4.238 & 2.194 & 5.847 \\ 11.605 & -4.122 & 3.907 & 2.072 & 2.732 \end{bmatrix} x(k) \preceq \begin{bmatrix} 17.955 \\ 13.237 \end{bmatrix} \quad \text{(Region \#1)}. \end{array} \tag{7.16}$$

From the value of nr, we know that there are 9 critical regions in $\mathcal{A}_x$. Thus, the firing rate is finally regulated by 9 affine functions in the structure of (7.16). The parameters used to design the analytic master controller are listed as follows:

$$\begin{aligned} P &= \begin{bmatrix} 74.518 & 10.731 & 16.117 & 10.470 & -2.948 \\ 10.731 & 39.210 & 25.861 & 13.875 & -3.275 \\ 16.117 & 25.861 & 22.252 & 12.407 & -3.184 \\ 10.470 & 13.875 & 12.407 & 8.481 & -1.957 \\ -2.948 & -3.275 & -3.184 & -1.957 & 1.549 \end{bmatrix}, \quad F = \begin{bmatrix} -0.162 \\ 0.048 \\ -0.035 \\ -0.013 \\ -0.057 \end{bmatrix}^T, \\ Q &= I_5, \ R = 0.1, \ N_u = 3, \ N_y = 5, \ x_0 = [0.4, 0.4]^T, \ u_0 = 0, \\ x_0 &= [0.040, \ 0.024, \ 0.207, \ -0.017, \ -0.080]^T \\ \hat{Q} &= I_5, \ \ \nu = 0.1 \ \text{(tuning parameters)}. \end{aligned} \tag{7.17}$$

where I_5 denotes the 5×5 identity matrix and other notation was defined in (5.71). We do not provide the figures for $\mathcal{A}_x$'s partitions because it is hard to visualize a section in high-dimensional spaces. However, we can still write down all mathematical expressions of critical regions in the form of (7.16).

In Fig.7.14, there are two customized blocks "sObserver" and "ERMPC". The former is associated with the S-function "s-Observer.m" for state observation; and the latter is associated with the S-function "ERMPC.m" for firing rate regulation. Replacing the PI master controller (Fig. 7.13) in the SYNSIM by the analytic master controller (Fig. 7.14), we can compute the trajectories of the estimated states (see Figs. 7.15) and the trajectories of the firing rate and the 900# header pressure (see Figs. 7.17 - 7.18). The block "ERMPC" has two external parameters "*expconl*" and "*us*," namely two flags of the corresponding S-function. "*expconl*" is the structure variable for the analytic master controller; and "*us*" is the static input which is extensively discussed in Section 5.3.2. In Figs. 7.17 - 7.18, the solid lines are the input and output of the analytic master controller and the dashed line are those of the PI controllers. Although the analytic master controller results in a larger overshoot, it gives faster responses. Fig. 7.16 illustrates the transition of $\mathcal{A}_x$'s critical regions, where y-axis is the index of the critical regions.

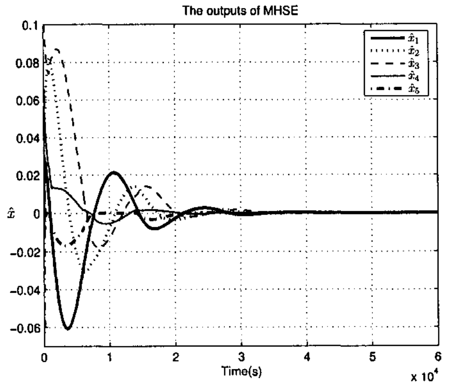


Figure 7.15: The trajectories of estimated states

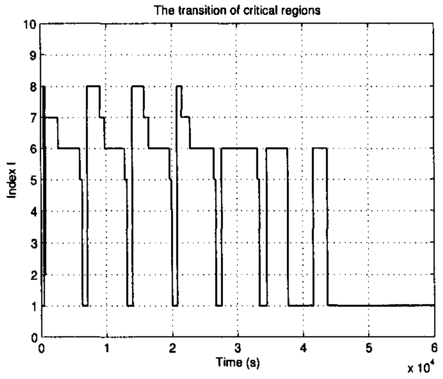


Figure 7.16: The transition of critical regions

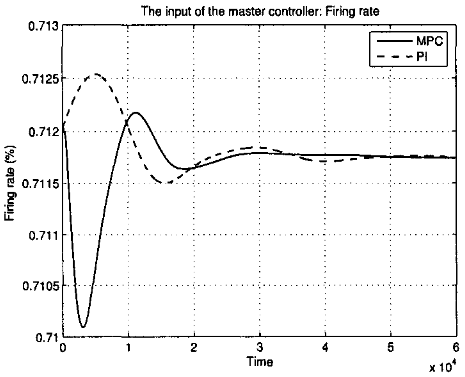


Figure 7.17: The trajectories of the firing rate

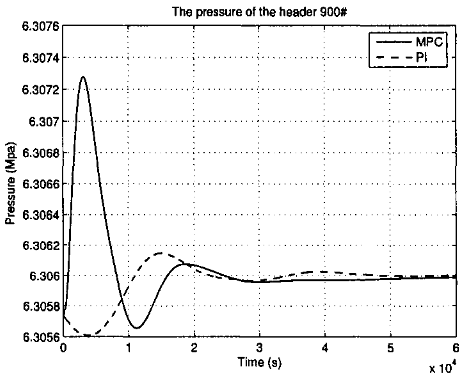


Figure 7.18: The trajectories of the 900# header pressure

7.3 Conclusions

This chapter applies the proposed algorithms of this thesis, explicit robust MPC and moving horizon state observation, to the SYNSIM model. The results were simulated by "SYNSIM", a very accurate, high order model of the plants, whose predictions very accurately correlate with actual plant measurement. By using the MATLAB identification toolbox, in this chapter we first identify a state space model of the master loop. Based on the identified model, an analytic master controller and a moving horizon state observer which work together to replace the existing PI master controller, were design. To facilitate the debugging of the analytic master controller, both the MPC controller and MHSO are realized by S-function programming and integrated with SYNSIM systematically. From simulation results, we can see that the proposed algorithms of this thesis are practical and effective, and may be applied to constrained industrial systems.

Chapter 8

Conclusions and suggestions for future research

In this chapter, the main contributions of this thesis are summarized and some suggestions for future research on MPC are outlined. The future research on MPC includes Dynamic Output-feedback MPC, Quantized MPC, Hybrid MPC, Time-delay MPC, Moving Horizon Fault Detection, Optimal MPC Horizons, Explicit GPC, 2-Dimensional MPC, etc.

8.1 Conclusions

The central ideas behind this thesis is to perform recursive closed-loop prediction and multiple-parametric sub-quadratic programming to separate MPC optimization from online implementation. In this fashion, we can achieve the FH-RMPC with an arbitrary horizon through only one-step forward state prediction. The main contributions of this thesis are summarized below:

FH-RMPC using LMIs

- A moving average system matrix was constructed to capture the modelling uncertainties and facilitate future state prediction, and FH-RMPC was achieved by using linear matrix inequality techniques.
- The terminal cost constraints were invoked to guarantee the closed-loop stability of resulting FH-RMPC systems.
- The robust LMI theorem was used to remove the existence of norm-bounded uncertain matrices of the LMIs constraints, which prevented the online optimization of FH-RMPC objectives. Consequently, the FH-RMPC design was converted into a semi-definite optimization problem.
- The details can be founded in Section 3.4.

Admissible state set

- The nontrivial problem, how to determine the admissible state set given input/output constraints and the bounds of uncertainties, was solved.
- The admissible state set for the system with norm-bounded uncertainties was derived by the piece-wise linear norm of output disturbances. Also, it was shown that the piece-wise linear norm of low-dimensional signals can be approximated by a weighted quadratic norm.
- To overcome the dimensional limitation of piece-wise linear norms, the approach of voronoi sets was developed for the systems with high-dimensional

disturbances. It was shown that the admissible state set can be also constructed by the voronoi set of perturbed state polyhedra associated with a Chebychev center and a Chebychev radius.

- The details can be founded in Section 5.1.1.

Offline robust model predictive control

- The offline robust model predictive control algorithm was developed to improve the implementation efficiency and reduce the computational complexity. The control policy of robust MPC was optimized by a set of piece-wise affine functions associated with the state space partition. As a result, the online implementation of explicit MPC regulator was simplified as function evaluation.

- A recursive closed-loop prediction pattern was introduced. By iteratively optimizing the piece objectives in the backward direction, only one-step state prediction was sufficient for FH-RMPC. No high-order uncertain terms occurred in the RMPC formulation, and sequentially two challenges of the offline RMPC formulation were solved, i.e., how to derive the explicit solutions to the piece objectives, and how to guarantee the uniform structure of the piece objectives.

- The closed-loop stability of the offline robust MPC was guaranteed by solving an algebraic Riccati equation and an LMI feasible problem. By constructing two tuning parameters, namely the terminal feedback gain and the terminal weighting, the proposed algorithm was capable of adjusting the tradeoff between the robustness and closed-loop stability of the resulting MPC system.

- Offset-free robust MPC was also discussed. By setting the static state, the static input, and the nominal disturbance, the offset-free control was converted into a regulation problem based on the shifted system model.

- The feasibility issue of the proposed algorithm was illustrated by state space partitions.

- The details can be founded in Chapter 5.

Robust moving horizon state observation

- Existing MHSO algorithms were extended into systems with internal uncertainties and external disturbances. Taking advantage of rewinding closed-loop prediction and multiple parametric optimization, two offline robust MHSO algorithms were developed.
- The state trajectory of MHSO was composed of two segments: one spanned from the initial point to the instant N (the length of prediction horizon), and the other started at instant $(N + 1)$ and proceeded to future. It was shown that the first segment can be determined by a full information state observer, and the other can be optimized by a set of piece-wise affine functions.
- Two proposed offline MHSOs were compared with each other in the sense of time-cost and memory-cost. It was shown that the offline MHSO with open-loop forward prediction cost more memory for intermediate optimization variables, and, however, the offline MHSO with closed-loop rewinding prediction spent a longer time on observer implementation.
- The robust MHSO with measurement noises was extensively discussed
- In addition to the offline robust MHSO, a robust state observer using LMIs was also considered. From the principle of invariant sets, the robust state observer using LMIs was formulated as an MAXDET optimization problem. It was shown that the convergence of the observer errors can be guaranteed by a set of shrinking ellipsoidal invariant sets.
- The details can be founded in Chapter 6.

Implementation efficiency and physical applications of MPC are two core factors considered while conducting this thesis, and they will also usher in the future MPC research.

8.2 Future research topics

Some possible future research topices are outlined below:

Problem 8.1 (Explicit observer-based RMPC) Observer-based MPC is an advisable choice for constrained MIMO systems with unmeasurable or partially unavailable states. In 2004, Roset and Nijmeijer proposed a nonlinear observer for nominal MPC systems [89]. The proposed algorithm stacked the states of the controlled system and corresponding nonlinear observer, and then based on the augmented system performed online optimization and online implementation. This scenario is practical and effective for nominal MIMO systems. Considering implementation efficiency and system uncertainties, however this algorithm may not be applied to fast systems in the presence of internal or external uncertainties. In previous chapters, we successfully constructed explicit robust MPC and offline RMHSO. In the same fashion, this approach can be used to develop novel explicit observer-based model predictive control.

Problem 8.2 (Dynamic output-feedback MPC) Besides observer-based MPC, dynamic output-feedback is an alternative to regulating constrained MIMO systems in the presence of unmeasured or partially unavailable states. Given a system

$$P : \left\{ \begin{array}{rcl} x(k+1) & = & Ax(k) + B_w w(k) + Bu(k), \\ z(k) & = & C_z x(k) + D_{zw} w(k) + D_z u(k), \\ y(k) & = & Cx(k) + D_w w(k), \end{array} \right. \tag{8.1}$$

where u is the manipulated input, w is the exogenous input, y is the measured output, and z is the system output, the output-feedback controller for the system P can be formulated as

$$K : \left\{ \begin{array}{rcl} x_c(k+1) & = & A_c x_c(k) + B_c y(k), \\ u(k) & = & C_c x_c(k) + D_c y(k). \end{array} \right. \tag{8.2}$$

The system matrices

$$\left[\begin{array}{c|c} A_c & B_c \\ \hline C_c & D_c \end{array} \right]$$

are unknown variables to be calculated by MPC formulations. From [91] and [45], we know that the state-feedback IH-MPC can be solved by iterating an constrained LQR problem. In the same fashion, by setting the prediction horizon equal to infinity, dynamic output-feedback MPC can be easily converted into online semi-definite programming. Due to the limitation of tuning freedom, it is much better to extend infinite horizon dynamic output–feedback MPC to the case of finite horizons. However, for FH-MPC with unknown system matrices, state prediction and online optimization are two major challenges. Thanks to the rewinding closed-loop prediction pattern, which was extensively used in Chapters 5 and 6, we think that the output-feedback FH-MPC may be solved by offline multiple-parametric quadratic programming.

Problem 8.3 (Quantized MPC) Quantizer is an essential element for industrial distributed control systems (DCSs) in which MPC is widely utilized. Researchers have pointed out that the influence of quantizers on closed-loop systems may be significant. A stable controlled system may exhibit limit-cycles and chaotic behaviors after quantized control [25, 65]. A quantized feedback control system can be represented by

$$P \ : \ \begin{cases} x(k+1) = Ax(k) + Bu(k), \\ y(k) = Cx(k) + Du(k), \end{cases} \tag{8.3}$$

$$K \ : \ \begin{cases} u(k) = f(v(k)), \\ v(k) = g(x(k)), \end{cases} \tag{8.4}$$

where P is the controlled system and K is the quantized controller. $f(\cdot)$ represents the quantized feedback and $g(\cdot)$ stands for the unquantized feedback. In MPC applications, MPC regulators are always pre-stored in a computer by optimization blocks (online MPC) or affine function blocks (offline MPC). Considering byte limitation and the influence of encoders and decoders, it is crucial to incorporate quantizer behaviors with MPC formulations. Reference [30] developed a sector bounded approach to quantized feedback design, i.e.,

$$Q : f(v) = \begin{cases} u_i & \text{if } \ \frac{1}{1+\delta}u_i < v \le \frac{1}{1-\delta}u_i, \ v > 0, \\ 0 & \text{if } \ v = 0, \\ -f(-v) & \text{if } \ v < 0, \end{cases}$$

where ρ is the quantization density and

$$\delta = \frac{1-\rho}{1+\rho}.$$

Associated with Q, the quantized feedback system can be converted into an uncertain system with the bounded uncertain output matrix Δ, i.e.,

$$x(k+1) = Ax(k) + B(1+\Delta)v(k), \quad \Delta \in [-\delta, \ \delta]. \tag{8.5}$$

Based on model (8.5) and utilizing the proposed algorithms in this thesis, we may derive novel quantized MPC.

Problem 8.4 (Hybrid MPC) As we discussed above, MPC is widely used in distributed control systems (DCSs), and in DCSs MPC is usually designed in the discrete-time domain but implemented in the continuous-time domain. Moreover, quantizers, samplers, and holders are essential components of a DCS network, and through quantizers, samplers, and holders, manipulated inputs and sampled outputs may be stored as logical variables in a computer. The logical manipulated inputs and outputs may reduce the communication cost between signals and MPC blocks, and consequently facilitate online optimization and online implementation. With the expectation of better MPC performance, researchers start to incorporate the behaviors of quantizer, sampler, and holder with MPC formulations and develop hybrid MPC schemes. In the past two to three years, hybrid MPC have attracted extensive attention of researchers.

Problem 8.5 (MPC for time-delay systems) To improve control performance, it is a natural idea to choose a more precise model to behave like the real process. It can be shown that many processes have the aftereffect phenomenon. Especially, for communication and field network systems, time-delay is very common and may lead to serious effects on system dynamics. Reference [85] provided a systematic survey on time-delay systems. A general time-delay system can be introduced by

$$\begin{aligned} x(k+1) &= f(x(k-\theta),\ k,\ u(k-\theta)), \qquad (8.6)\\ y(k) &= g(x(k-\theta),\ k,\ u(k-\theta)),\\ x(\theta) &= \vartheta(\theta),\ -t_0 \le \theta \le h - t_0, \end{aligned}$$

where θ is the time-varying delay factor bounded by h. $x(k-\theta)$ and $u(k-\theta)$ are the functions of the delay factor θ, and stand for the system's input and output respectively. If designing the manipulated input $u(k)$ (delay-independent) from MPC formulations, we proposes a new type of advanced MPC, namely MPC for time-delay systems [48].

Problem 8.6 (Moving horizon fault detection) Fault detection is an important problem in process engineering. Detecting faults effectively and maintaining processes in controllable regions can help avoid abnormal events and reduce productivity loss. Process faults may occur in the sensor side and/or the actuator side. Therefore, a system with faults is formulated as

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ \hat{y}(k) &= Cx(k) \\ u(k) &= \hat{u}(k) + f_u(k) \\ y(k) &= \hat{y}(k) + f_y(k) \end{aligned} \tag{8.7}$$

where $\hat{u}(k)$ and $\hat{y}(k)$ are the fault-free input and output. $u(k)$ and $y(k)$ are the real input (the actuator output) and the real output (the sensor output). $f_u(k)$ and $f_y(k)$ stand for the actuator fault and the sensor fault, respectively. The aim of fault detection and isolation (FDI) is to estimate the value of f_u and f_y. In literature, f_u and f_y are sometimes referred to as primary residual vectors (PRVs). Also, from f_u and f_y we can construct a set of structured residual vectors (SRVs) to facilitate fault isolation [17]. Roughly speaking, FDI problems can be regarded as the estimation of unknown inputs and outputs. Since we successfully developed the robust state estimation by utilizing moving horizon schemes in Chapter 6. By using the similar idea, we may achieve moving horizon fault detection in the presence of system internal and external uncertainties.

Problem 8.7 (RMPC with the optimal prediction horizon) The computational complexity of offline MPC is tightly related to the length of prediction horizons. Therefore, it is a crucial (but still open) issue to optimize MPC horizon length and obtain a satisfactory tradeoff between computational complexity and design aggressiveness. Roughly speaking, a smaller horizon reduces the number of optimization loops, but deteriorates the stability of feedback control systems. The problem of online RMPC with varying horizons may be also considered in our future research.

Problem 8.8 (Explicit GPC) GPC is the most popular stochastic MPC strategy in industry, and it has been extended to MIMO systems in the presence of internal and external uncertainties [10]. If GPC systems have no internal and external uncertainties, we can derive the explicit solution to GPC, which has been shown in Eq. (2.37). But for constrained GPC, it is still open to derive an analytic solution.

Problem 8.9 (2-dimensional MPC) 2-dimensional systems have the practical and theoretical importance in process analysis and control. As one of widely used 2-dimensional systems, Roesser system can be expressed as

$$\begin{aligned} \begin{bmatrix} x_h(i+1,\ j) \\ x_v(i,\ j+1) \end{bmatrix} &= A \begin{bmatrix} x_h(i,\ j) \\ x_v(i,\ j) \end{bmatrix} + Bu(i,\ j), \\ y(i,j) &= C \begin{bmatrix} x_h(i,\ j) \\ x_v(i,\ j) \end{bmatrix} + Du(i,\ j), \end{aligned} \tag{8.8}$$

where $x_h(i+1,\ j)$ and $x_v(1,\ j+1)$ stand for the horizon state and vertical state, respectively [88, 33]. Based on the model in (8.8), we may extend the conventional MPC algorithms into 2-dimensional systems. Similar to dynamic output-feedback MPC, state prediction and online optimization are two barriers for 2-dimensional MPC.

Besides the above problems, some other MPC issues related to convex optimization, which may reduce computational complexity, may be also considered in our future research.

Bibliography

[1] F. Allgower, R. Findeisen, and Z. K. Nagy. Nonlinear model predictive control: From theory to application. *J. Chin. Inst. Chem. Engrs.*, 3:299–315, 2004.

[2] R. Bellman. *Dynamic programming.* Princeton University Press, New Jersey, 1957.

[3] A. Bemporad. *Hybrid Toolbox - A Free MATLAB Toolbox.* Available: http://www.dii.unisi.it/ bemporad/, 2005.

[4] A. Bemporad, F. Borrelli, and M. Morari. Model predictive control based on linear programming - the explicit solution. *IEEE Tran. on Automatic Control*, 47:1974–1985, 2002.

[5] A. Bemporad and M. Morari. Robust model predictive control: A survey. In *Proc. of European Control Conf.*, pages 939–944, Porto, Portugal, 2001.

[6] A. Bemporad, M. Morari, V. Dua, and N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38:3–20, 2002.

[7] A. Bemporad, N. L. Richer, and M. Morari. *Model Predictive Control Toolbox.* Mathworks Inc., Natick, MA, 2004.

[8] C. Bohn and D. P. Atherton. An analysis package comparing pid anti-windup strategies. *IEEE Control Systems*, 15:34–40, 1995.

[9] J. Bordeneuve-Guibe and C. Vaucoret. Robust multivariable predictive control – an application to an industrial test stand. *Systems and Control Letters*, 21:54–65, 2001.

[10] J. Bordeneuve-Guibe and C. Vaucoret. Robust multivariable predictive control: an application to an industrial test stand. *IEEE Tran. Control Systems Magazine*, 21(2):54–65, 2001.

[11] F. Borrelli, A. Bemporad, and M. Morari. Geometric algorithm for multiparametric linear programming. *Journal of optimization theory and applications*, 118:515–540, 2003.

[12] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in Control Theory.* SIAM, Philadelphia, 1994.

[13] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge university press, Cape Town, South Africa, 2004.

[14] E. F. Camacho and C. Bordons. *Model Predictive Control.* Springer, London, 1999.

[15] A. Casavola, D. Famularo, and G. Franze. Robust constrained predictive control of uncertain norm-bounded linear systems. *Automatica*, 40:1865–1876, 2004.

[16] T. Chen and B. Francis. *Optimal sampled-data control systems.* Springer, New York, 1995.

[17] E. Chow and A. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Tran. Automatic Control*, 29:603–614, 1984.

[18] D. Chu, T. Chen, and H. J. Marquez. Explicit robust model predictive control using recursive closed-loop prediction. *International Journal of Robust and Nonlinear Control*, 16:519–546, 2006.

[19] D. Chu, T. Chen, and H. J. Marquez. Finite horizon robust model predictive control with terminal cost constraints. *IEE Part D: Control Theory and Application*, 153:156–166, 2006.

[20] D. Chu, T. Chen, and H. J. Marquez. Robust moving horizon state observer. *Submitted to International Journal of Control*, 2006.

[21] D. W. Clarke, C. Mohtadi, and P. S. Tuffs. Generalized predictive control—part i the basic algorithm. *Automatica*, 23:137–148, 1987.

[22] D. W. Clarke, C. Mohtadi, and P. S. Tuffs. Generalized predictive control—part ii extensions and interpretations. *Automatica*, 23:149–160, 1987.

[23] C. R. Culter and B. L. Ramaker. Dynamic matrix control - a computer control algorithm. In *AIChE National Meeting*, Houston, TX, 1979.

[24] C. R. Culter and B. L. Ramaker. Dynamic matrix control - a computer control algorithm. In *Proceedings of the Joint Automatic Control Conference*, Prague, the Czech Republic, 1980.

[25] D. F. Delchamps. Stabilizing a linear system with quantized state feedback. *IEEE Tran. on Automatic Control*, 35:916–924, 1990.

[26] V. Dua and E.N. Pistikopoulos. Algorithms for the solution of multiparametric mixed integer linear programming problems. *Annals of operations research*, 99:123–139, 2000.

[27] P. Findeisen. Moving horizon state estimation of discrete time systems. Master's thesis, Department of Chemical and Biological Engineering, University of Wisconsin-Madison, 1997.

[28] R. Findeisen, F. Allgower, M. Diehl, L. Magni, and Z. Nagy. Tutorical: Model predictive control technology. In *IFAC Workshop on Nonlinear Model Predictive Control: Introduction & Current Topics*, pages 3–12, Prague, the Czech Republic, 2005.

[29] J. B. Froisy. Model predictive control: Past, present and future. *ISA Transactions*, 33:235–243, 1994.

[30] M. Fu and L. Xie. The sector bounded approach to quantized feedback control. *IEEE Tran. Automatic Control*, 50:1698–1711, 2005.

[31] H. Fukushima and R. R. Bitmead. Robust constrained predictive control using comparison model. *Automatica*, 41:97–106, 2005.

[32] P. Gahinet, A. Nemirovski, A. J. Lamb, and M. Chilali. *LMI Control Toolbox: For Use with MATLAB.* Mathworks Inc., Natick, MA, 1995.

[33] H. Gao, J. Lam, S. Xu, and C. Wang. Stability and stabilization of uncertain 2d discrete systems with stochastic perturbation. *Multidimensional Systems and Signal Processing*, 16, 2005.

[34] C. E. Garcia, D. M. Prett, and M. Morari. Model predictive control: Theory and practice - a survey. *Automatica*, 25:335–348, 1989.

[35] L. E. Ghaout and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM J. Matrix Anal. Appl.*, 18:1035–1064, 1997.

[36] H. V. Gonzalez, J. M. Flaus, and G. Acuna. Moving horizon state estimation with global convergence using interval techniques: Application to biotechnological pocesses. *Journal of Process Control*, 13:325–336, 2003.

[37] P. Grosdidier, J. B. Froisy, and M. Hamman. *IFAC workshop on Model based predictive control, chapter The IDCOM-M controller.* Pergamon press, Oxford, 1988.

[38] E. L. Haseltine and J. B. Rawlings. Critical evaluation of extended kalman filtering and moving horizon estimation. *Ind. Eng. Chem. Res.*, 44:2451–2460, 2005.

[39] B. Hu and A. Linnemann. Toward infinite-horizon optimality in nonlinear model predictive control. *IEEE. Trans. on Automatica Control*, 47:679–683, 2002.

[40] C. Huang and M. Huang. Estimation of the second-order parameters from the process transient by simple calculation. *Industrial & Engineering Chemistry Research*, 32(1):228–230, 1993.

[41] J. B. Rawlings K. R. Muske and J. H. Lee. Receding horizon recursive state estimation. In *Proc. American Control Conference*, pages 900–904, San Francisco, California, 1993.

[42] R. E. Kalman. A new approach to linear filtering and prediction problems. *Tran. of the ASME - Journal of Basic Engineering*, 82:35–45, 1960.

[43] S. S. Keerthi and E. G. Gilbert. Optimal infinite horizon feedback laws for a general class of constrainted discrete time systems: Stability and moving-horizon approximations. *Journal of Optimization Theory and Application*, 57:265–293, 1988.

[44] E.C. Kerrigan and J.M. Maciejowski. Invariant sets for constrained nonlinear discrete-time systems with application to feasibility in model predictive control. In *Proc. of the 39th IEEE conference on decision and control*, pages 855–860, Sydney, Australia, 2000.

[45] M. V. Kothare, V. Balakrishman, and M. Morari. Roust constrained model predictive control using linear matrix inequalities. *Automatica*, 32:1361–1379, 1996.

[46] M. Kvasnica, P. Grieder, and M. Baotić. *Multi-Parametric Toolbox (MPT) - A Free MATLAB Toolbox.* Available: http://control.ee.ethz.ch/ mpt/, 2004.

[47] H. Kwakernaak and R. Sivan. *Linear optimal control systems.* Wiley-interscience, New York, 1972.

[48] W. H. Kwon, Y. S. Lee, and S. H. Han. General receding horizon control for linear time-delay systems. *Automatica*, 40:1603–1611, 2004.

[49] W. Langson, I. Chryssochoos, S. Rakovic, and D. Q. Mayne. Robust model predictive control using tubes. *Automatica*, 40:125–133, 2004.

[50] J. H. Lee and Z. Yu. Worst-case formulations of model predictive control for systems with bounded parameters. *Automatica*, 33:763–781, 1997.

[51] C. Lien. Robust observer-based control of systems with state perturbations via lmi approach. *IEEE Tran. on Automatic Control*, 49:1365–1370, 2004.

[52] L. Ljung. *System Identification Toolbox: For Use with MATLAB.* Mathworks Inc., Natick, MA, 1995.

[53] L. Ljung. *System Identification - Theory For the User, 2nd ed.* PTR Prentice Hall, N.J., 1999.

[54] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its applications*, 284:193–228, 1998.

[55] J. Löfberg. *Linear Model Predictive Control: Stability and Robustness.* PhD thesis, Department of Electrical Engineering, Linköpings university, Sweden, 2001.

[56] J. Löfberg. Minimax mpc for systems with uncertain input gain — revisited. Technical report, Automatic Control Group in Linköpings University, July 2001.

[57] J. Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[58] D. G. Luenberger. An troduction to observers. *IEEE Trans. on Automatic Control*, 16:596–602, 1971.

[59] J. M. Maciejowski. *Predictive Control with Constraints.* Prentice Hall, England, 2002.

[60] M. S. Mahmouda and H. K. Khalil. Robustness of high-gain observer-based nonlinear controllers to unmodeled actuators and sensors. *Automatica*, 38:361–369, 2002.

[61] H. J. Marquez and M. Riaz. Robust state observer design with application to an industrial boiler system. *Control Engineering Practice*, 13:713–728, 2005.

[62] D. Q. Mayne and H. Michalska. Receding horizon control of non-linear systems. *IEEE Tran. on Automatic Control*, 35:614–824, 1990.

[63] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

[64] H. Michalska and D. Q. Mayne. Moving horizon observers and observer-based control. *IEEE Tran. on Automatic Control*, 40:995–1006, 1995.

[65] R. K. Miller, A. N. Michel, and J. A. Farrel. Quantizer effects on steady state error specifications of digital control systems. *IEEE Tran. Automatic Control*, 34:651–654, 1989.

[66] S. Mo and J. Billingsley. Fast-model predictive control of multivariable systems. *IEE proceedings control theory and applications*, 137:364–366, 1990.

[67] M. Morari and J. H. Lee. Model predictive control: Past, present and future. *Computer and Chemical Engineering*, 23:667–682, 1999.

[68] K. R. Muske and T. A. Badgwell. Disturbance modeling for offset-free linear model predictive control. *Journal of Process control*, 12:617–632, 2002.

[69] Y. Nesterov and A. Nemirovskii. *Interior-point polynominal algorithms in convex programming.* SIAM Studies in Applied Mathematics, Philadelphia, 1994.

[70] A. Packard and J. Doyle. The complex structured singular value. *Automatica*, 29:71–109, 1993.

[71] G. Pannocchia. Disturbance models for offset-free model predictive control. *AIChE Journal*, 49:426–437, 2003.

[72] G. Pannocchia. Robust disturbance modeling for model predictive control with application to multivariable ill-conditioned processes. *Journal of Process control*, 13:693–701, 2003.

[73] P. Park and S. C. Jeong. Constrained rhc for lpv systems with bounded rates of parameter variations. *Automatica*, 40:865–872, 2004.

[74] I. Peterson and D. C. McFarlane. Optimal guaranteed cost control and filtering for uncertain linear systems. *IEEE Tran. on Automatic Control*, 39:1971–1977, 1994.

[75] J.A. Primbs and V. Nevistic. A framework for robustness analysis of constrained finite receding horizon control. *IEEE Tran. on Automatic Control*, 45:1828–1838, 2000.

[76] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764, 2003.

[77] T. Raissi, N. Ramdani, and Y. Candau. Bounded error moving horizon state estimator for non-linear continuous-time systems: Application to a bioprocess system. *Journal of Process Control*, 15:537–545, 2005.

[78] C. V. Rao, J. B. Rawlings, and J. H. Lee. Constrained linear state estimation - a moving horizon approach. *Automatica*, 37:1619–1628, 2001.

[79] C. V. Rao, J. B. Rawlings, and D. Q. Mayne. Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. *IEEE Tran. on Automatic Control*, 48:246–258, 2003.

[80] J. B. Rawlings. Tutorical: Model predictive control technology. In *Proc. of the American Control Conference*, pages 662–676, San Diego, California, 1999.

[81] J. B. Rawlings and K. R. Muske. Stability of constrained receding horizon control. *IEEE Tran. on Automatic Control*, 38:1512–1516, 1993.

[82] J. Richalet. *Pratique de la commande predictive.* Hermes Publicat Sciences, France, 1992.

[83] J. Richalet. Industrial applications of model-based predictive control. *Automatica*, 29:1251–1274, 1993.

[84] J. Richalet, A. Rault, J. L. Testud, and J. Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14:413–428, 1978.

[85] J. Richard. Time-delay systems: an overview of some recent advances and open problems. *Automatica*, 39:1667–1694, 2003.

[86] R. D. Rink, A. White, and R. Leung. *SYNSIM: A Computer Simulation Modle for the Mildred Lake Steam/Electrical System of Syncrude Canada Ltd.* University of Alberta, Edmonton, Canada, 1996.

[87] D. G. Robertson, J. H. Lee, and J. B. Rawlings. A moving horizon-based approach for least-squares state estimation. *AIChE. Journal*, 42:2209–2224, 1996.

[88] R. P. Roesser. A discrete state-space model for linear image processing. *IEEE Tran. Automatic Control*, 20, 1975.

[89] B. Roset and H. Nijmeijer. Observer-based model predictive control. *International of Control*, 77:1452–1462, 2004.

[90] L. Rundqwist. *Anti-Reset Windup for PID Controllers.* PhD thesis, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1991.

[91] C. Scherer, P. Cahinet, and M. Chilali. Multiobjective output-feedback control via lmi optimization. *IEEE Tran. Automatic Control*, 42:896–911, 1997.

[92] P. O. M. Scokaert and J. B. Rawlings. Constrained linear quadratic regulation. *IEEE Tran. on Automatic control*, 43:1163–1169, 1998.

[93] D.E. Seborg, T.F. Edgar, and D.A. Mellichamp. *Process dynamics and control.* Wiley series in chemical engineering, New York, USA, 1989.

[94] S. L. Shah, C. Mohtadi, and D. W. Clarke. Multivariable adaptive control without a prior knowledge of the delay matrix. *Systems and Control Letters*, 9:295–306, 1987.

[95] I. Skrjanc and D. Matko. Predictive functional control based on fuzzy model for heat-exchanger pilot plant. *IEEE Tran. on fuzzy systems*, 8:705–712, 2000.

[96] S.Shah and A.K. Tangirala. *System identification course notes.* Dept. of chemical engineering, University of Alberta, Edmonton, Canada, 2003.

[97] K. R. Sundaresan, C. C. Prasad, and P. R. Krishnaswamy. Evaluating parameters from process transients. *Industrial & Engineering Chemistry Process Design and Development*, 17(3):237–241, 1978.

[98] P. Tondel, T. A. Johansen, and A. Bemporad. An algorithm for multiparametric quadratic programming and explicit mpc solutions. In *IEEE Conference on Decision and Control*, pages 1199 – 1204, Oriando, Florida, 2001.

[99] J. G. VanAntwerp and R. D. Draatz. Fast model predictive control of sheet and film processes. *IEEE Tran. on Control Systems Technology*, 8:408–417, 2000.

[100] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.

[101] L. Vandenberghe, S. Boyd, and S. P. Wu. Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 19:499–533, 1998.

[102] A. Vivas and P. Poignet. Predictive functional control of a parallel robot. *Control Engineering Practice*, 13:863–874, 2005.

[103] Z. Wan and M. V. Kothare. An efficient off-line formulation of robust model predictive control using linear matrix inequalities. *Automatica*, 39:837–846, 2003.

[104] L. Wang and F. Wan. Structured neural networks for constrained model predictive control. *Automatica*, 37:1235–1243, 2001.

[105] Y. Wang and J. Rawlings. A new robust model predictive control method i: theory and computation. *Journal of process control*, 14:231–247, 2004.

[106] Y. Wang, L. Xie, and C. de Souza. Robust control of a class of uncertain systems. *System and Control Letters*, 19:139–149, 1992.

[107] S. Wu, L. Vandenberghe, and S. Boyd. *MAXDET - Software for Determinant Maximization Problems: User's Guide.* Electrical Engineering Department, Stanford University, Natick, MA, 1996.

[108] Y. Xiong and M. Saif. Unknown disturbance inputs estimation based on a state functional observer design. *Automatica*, 39:1389–1398, 2003.

[109] W. Yan and J. Lam. On quadratic stability of systems with structured uncertainty. *IEEE Tran. on Automatic Control*, 46:1799–1805, 2001.

[110] P. Zitek. Anisochronic state observers for hereditary systems. *International Journal of Control*, 71:581–599, 1998.